Learning Pseudo-Riemannian Manifolds for Heterophilic Graphs via Graph Signature

Yun Young Choi[*†]
SolverX
Seoul, Republic of Korea
young@solverx.ai

Asung Kil[*]
SolverX
Seoul, Republic of Korea
askil@solverx.ai

Sun Woo Park[*]
Max Planck Institute for Mathematics
Bonn, Germany
s.park@mpim-bonn.mpg.de

Minho Lee
SolverX
Seoul, Republic of Korea
mhlee@solverx.ai

Seokhwan Kim
SolverX
Seoul, Republic of Korea
kshsh0405@solverx.ai

Abstract

Several widely used families of Graph Neural Networks carry a positive-definite geometric prior. Graph convolution read as heat diffusion needs a positive semi-definite operator, hyperbolic models place features on positive-definite Riemannian manifolds, and dot-product and GAT-style attention scores use positive-definite inner products. On heterophilic graphs, where edges often signify dissimilarity or structural repulsion, that prior is a mismatch: no positive-definite form can assign a negative squared distance to a direction the labels separate. To measure the mismatch, we introduce the *Graph Signature Index* ($s_{\mathcal{G}}$), the inertia of a class-compatibility matrix built from label topology alone, and we prove that no value of the edge or node homophily index in $(0, 1)$ rules out a negative eigenvalue. We then propose the *Pseudo-Riemannian Attention Network* (PRAT), whose attention logit is a signed squared pseudo-distance under a learnable diagonal metric initialized from $s_{\mathcal{G}}$. The design is diagnose-then-learn: the signature supplies a sign structure that unconstrained training keeps outside near-zero directions, and the projection learns the rest of the metric around it. Across fifteen node-classification benchmarks PRAT outperforms or matches the strongest heterophily-adapted models. A controlled ablation ties that performance to the sign structure itself: with architecture and protocol fixed, forbidding the metric to be indefinite costs accuracy on the graphs $s_{\mathcal{G}}$ marks as indefinite and leaves the others unchanged.

1 Introduction

Graph Neural Networks (GNNs) are the standard tool for relational learning [4]. Several of the most widely used families rest on a geometric prior rooted in Riemannian manifolds, typically Euclidean space, in which the metric tensor is strictly positive definite and connected nodes are therefore "close" and expected to share representations. The prior is explicit in three places. Graph convolution read as heat diffusion requires a positive semi-definite operator [5]; hyperbolic architectures such as HGCN [6] and HAT [47] embed features on positive-definite Riemannian manifolds; and dot-product [38] and GAT-style [39] attention use Euclidean inner products of projected features, which admit no negative squared distance. The assumption is not universal. Signed attention [1] and sheaf

[*]Equal contribution. Yun Young Choi, Asung Kil, and Sun Woo Park contributed equally.
[†]Corresponding author: young@solverx.ai.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

models [2] are natural counterexamples, and precisely because they work, the geometry deserves to be treated as a modeling choice rather than a default.

While this Riemannian inductive bias serves well for homophilic graphs, it creates a fundamental *geometric mismatch* when applied to heterophilic graphs, where edges often denote dissimilarity or structural repulsion [49, 27]. Heterophily-adapted GNNs address this architecturally with high-frequency spectral filters [11, 7, 14] or signed aggregation weights [1, 44, 41]. These are empirically effective, but the space they act in stays positive definite, so repulsion is expressed by distorting an embedding built for attraction [36].

We read heterophily geometrically, whereas the prevailing view is statistical. Non-Euclidean GNNs [6, 42] adapt curvature to hierarchical structure. We argue that some heterophilic graphs call instead for an indefinite metric, which admits *negative squared distances* [17] and so can sharpen across one class pair while smoothing across another. We diagnose the geometry the labels induce and then learn the metric inside it. The diagnostic is the *Graph Signature Index* ($s_{\mathcal{G}}$), a spectral invariant of the label topology. The model is the *Pseudo-Riemannian Attention Network* (PRAT), whose attention logit is a signed squared pseudo-distance under a diagonal metric initialized from $s_{\mathcal{G}}$ and learned thereafter.

2 Theoretical Framework: The Geometry of Heterophily

2.1 Graph Signature Index

Previous studies proposed measures of graph homophily, including the node homophily index h_V [27] and edge homophily index h_E [50] described in Appendix A.5. Their values, ranging between 0 and 1, quantify the likelihood that nodes with identical labels are connected by edges (see Appendix A.7). The graph signature index refines these indices by recording the signature of the eigenvalues of a real symmetric matrix constructed from a graph with node labels.

Definition 2.1. Let $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ be an undirected graph without self-loops. Let $l : \mathcal{V} \to L = \{1, 2, \cdots, |L|\}$ be the ground-truth node label function assigning each node to one of $|L|$ classes. The graph signature matrix S_G is the $|L| \times |L|$ real symmetric matrix whose (i, j)-th entry is given as

$$s_{i,j} := \frac{|\{(v, w) \in \mathcal{E} \mid l(v) = i, l(w) = j\}|}{|\{v \in \mathcal{V} \mid l(v) = i\}| \cdot |\{v \in \mathcal{V} \mid l(v) = j\}|} \tag{1}$$

The graph signature index of $\mathcal{G}$, denoted as $s_{\mathcal{G}}$, is the 3-tuple consisting of the number of positive, zero, and negative eigenvalues of $S_{\mathcal{G}}$.

Because $S_{\mathcal{G}}$ is a real symmetric matrix, every eigenvalue of $S_{\mathcal{G}}$ is real. Sylvester's law of inertia shows that the graph signature index is invariant under any invertible linear change of coordinates on $\mathbb{R}^{|L|}$.

Theorem 2.2 ([34])**.** *Let $S \in \mathbb{R}^{n \times n}$ be a real symmetric matrix. Then there exists a diagonal matrix $D \in \mathbb{R}^{n \times n}$ and an invertible matrix $W \in \mathbb{R}^{n \times n}$ such that $S = W^T D W$, where $(\cdot)^T$ denotes the transpose of a square matrix. The numbers of positive, zero, and negative eigenvalues of S are invariant under the change of coordinates.*

Each (i, j)-th entry of $S_{\mathcal{G}}$ is the expected probability that a node with a preassigned label j lies in the depth 1 unfolding tree rooted at a node with label i; see Appendix A.2. The signature of the eigenvalues of $S_{\mathcal{G}}$ can be used to determine whether graphs with discrete labels are heterophilic or homophilic. For example, graphs with two discrete node labels and $\mathcal{E} \neq \emptyset$ have three types of $s_{\mathcal{G}}$; see Appendix A.3. Appendix A.4 provides heuristics on the density of these three graph types.

The following theorem relates the edge homophily level h_E to the graph signature index $s_{\mathcal{G}}$.

Theorem 2.3 (Critical Homophily Threshold)**.** *Assume class i contains n_i nodes and $N = \sum_{i=1}^{|L|} n_i$. Under a proportional allocation model, the expected graph signature matrix $S_{\mathcal{G}}$ has eigenvalues:*

$$\lambda_1 > 0, \qquad \lambda_2 = \cdots = \lambda_{|L|} = \frac{2|\mathcal{E}| \left(h_E N^2 - \sum_{k=1}^{|L|} n_k^2\right)}{\sum_{k=1}^{|L|} n_k^2 \cdot \left(N^2 - \sum_{k=1}^{|L|} n_k^2\right)}. \tag{2}$$

The critical homophily threshold is given by: $h_E^* = \frac{\sum_{k=1}^{|L|} n_k^2}{N^2}$. For balanced classes where $n_i = n$ for all i, this reduces to $h_E^* = 1/|L|$.

While Theorem 2.3 reveals a correlation between edge homophily and graph signature under idealized conditions, homophily indices cannot fully characterize the spectral properties of $S_{\mathcal{G}}$. We formalize this limitation by showing that no value of h_E or h_V in $(0, 1)$ rules out a negative eigenvalue of $S_{\mathcal{G}}$.

Theorem 2.4 (Limitation of Homophily Indices)**.** *For every rational $h \in (0, 1)$, there exists a graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ with discrete node labels such that $h_E = h_V = h$ and $S_{\mathcal{G}}$ has a negative eigenvalue.*

2.2 Limitations of Riemannian GNNs

We reformulate the problem of analyzing heterophilic graphs using pseudo-Riemannian manifolds. A pseudo-Riemannian manifold $(\mathcal{M}, g)$ is a smooth manifold endowed with a metric tensor $g_p : T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$ which defines a non-degenerate symmetric bilinear form over the tangent space $T_p\mathcal{M}$ at each point $p \in \mathcal{M}$. A pseudo-Riemannian manifold is Riemannian if the tensor g_p is positive-definite for every point $p \in \mathcal{M}$. We keep this terminology for a symmetric bilinear form that is allowed to be degenerate. This is deliberate: the form on the hidden space is a pullback of rank at most $|L|$, hence degenerate whenever the hidden width exceeds the number of classes, and it depends on a hidden vector only through its image in $\mathcal{L}$. Non-degeneracy returns on $\mathcal{L}$ itself whenever $s_{\mathcal{G}}$ has no zero eigenvalue, and Theorem 2.5 counts only negative eigenvalues, so degeneracy does not affect it. We focus on the h-dimensional manifold $\mathcal{M}$ endowed with a metric tensor $g_p := S \in \mathbb{R}^{h\times h}$ for some fixed real symmetric matrix S at every point $p \in \mathcal{M}$, which defines the following quadratic form on tangent vectors $w_1, w_2 \in T_p\mathcal{M}$: $g_p(w_1, w_2) = w_1^T S w_2$ [24, 37].

Let $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ be a graph with continuous node attributes $X : \mathcal{V} \to \mathbb{R}^h$ and embedded node labels $l : \mathcal{V} \to \mathbb{R}^{|L|}$. We set the metric tensor over $\mathbb{R}^{|L|}$ to be the graph signature matrix $S_{\mathcal{G}}$ of the given graph $\mathcal{G}$. By doing so, we regard the graph $\mathcal{G}$ as a subspace of some smooth manifold $\mathcal{M}$ admitting a smooth function $f : \mathcal{M} \to \mathcal{L} \cong \mathbb{R}^{|L|}$, whose metric tensor is the pullback of metric tensor $f^* S_{\mathcal{G}}$. Given a metric tensor g on $\mathcal{L}$, its pullback f^*g on $\mathcal{M}$ is defined as

$$(f^*g)_p(v, w) := g_{f(p)}(T_p f(v), T_p f(w)).$$

Because $S_{\mathcal{G}}$ can have negative eigenvalues, $\mathcal{M}$ is a pseudo-Riemannian manifold, thereby covering a class of manifolds which are not Riemannian. The theorem below, proved in Appendix A.8, makes this observation explicit.

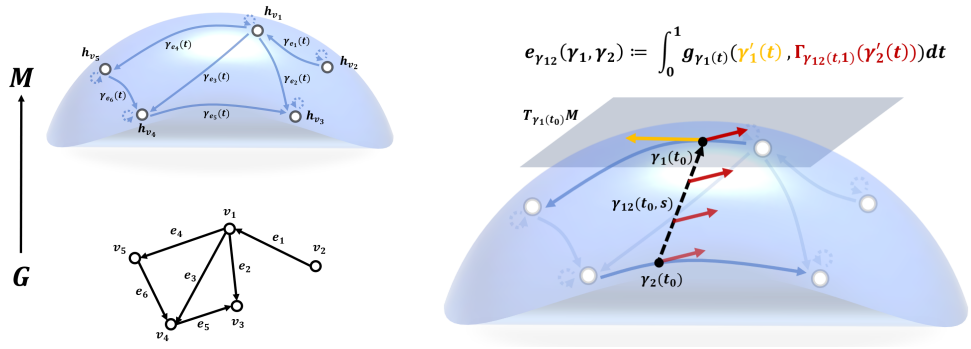
Theorem 2.5 (Geometric Mismatch)**.** *Let $\mathcal{M}$ and $\mathcal{L}$ be differentiable manifolds and $f : \mathcal{M} \to \mathcal{L}$ be a smooth map. If $S_{\mathcal{G}}$ defined on $\mathcal{L}$ is indefinite and the pushforward $T_p f : T_p\mathcal{M} \to T_{f(p)}\mathcal{L}$ is surjective for every $p \in \mathcal{M}$, then there exists no diffeomorphism $\phi : \mathcal{L} \to \mathcal{L}$ such that the pullback of the Euclidean metric I, the identity form, satisfies $(\phi \circ f)^* I = f^* S_{\mathcal{G}}$.*

Theorem 2.5 establishes a geometric mismatch between the Riemannian prior and graphs whose $S_{\mathcal{G}}$ is indefinite. It proves that this geometric mismatch is intrinsic and cannot be bridged by any smooth transformation of the Riemannian space. Surjectivity of the pushforward is what carries the indefiniteness through to the pullback: the image of $T_p f$ is then all of $T_{f(p)}\mathcal{L}$, so a negative direction of $S_{\mathcal{G}}$ has a preimage. Injectivity alone would not suffice, since an injective map can pull an indefinite form back to a degenerate one. In our architecture $T_p f$ is the projection $\mathbf{W}$ of Equation (4), surjective exactly when $\operatorname{rank} \mathbf{W} = |L|$. Every configuration we report has $h' > |L|$, so the hypothesis holds for almost every $\mathbf{W}$. Appendix A.8 discusses further implications for heterophilic graphs.

3 Method: Pseudo-Riemannian Attention Network (PRAT)

3.1 Pseudo-Riemannian Edge Attention using continuous features

We define the tangent attention scores of a finite graph $\mathcal{G}$, which generalize the notion of attention scores used in graph attention networks (GAT) and attention-transformers. The geometric structure endowed by the pseudo-metric tensor g allows us to compute the numerical relation between two edges of a finite directed graph $\mathcal{G}$ by comparing their derivatives. Given a curve $\gamma : [0, 1] \to \mathcal{M}$, the derivative of the curve at time $t = t_0$, denoted as $\gamma'(t_0)$, is an element of the tangent space $T_{\gamma(t_0)}\mathcal{M}$. Given two curves $\gamma_1, \gamma_2 : [0, 1] \to \mathcal{M}$, the two derivatives $\gamma_1'(t_0)$ and $\gamma_2'(t_0)$ may not lie in the same tangent space. To overcome this, we define a function $\gamma_{12} : [0, 1] \times [0, 1] \to \mathcal{M}$ which satisfies $\gamma_{12}(t, 0) = \gamma_2(t)$ and $\gamma_{12}(t, 1) = \gamma_1(t)$. For each time $t = t_0$, the parallel transport operator $\Gamma_{\gamma_{12}(t_0,1)} : T_{\gamma_2(t_0)}\mathcal{M} \to T_{\gamma_1(t_0)}\mathcal{M}$ moves any tangent vector $w_2 \in T_{\gamma_2(t_0)}\mathcal{M}$ to the tangent



(a) (b)

Figure 1: Figure 1(a) illustrates how one can map a directed graph $\mathcal{G}$ to a subspace of a pseudo-Riemannian manifold $(\mathcal{M}, g)$. The nodes of $\mathcal{G}$ correspond to points on $\mathcal{M}$, whereas the edges of $\mathcal{G}$ correspond to curves on $\mathcal{M}$. Figure 1(b) demonstrates how one may compute tangent attention scores between two edges of $\mathcal{G}$ using the pseudo-Riemannian metric g defined over the tangent space $T_{\gamma_1(t)}\mathcal{M}$ at each point of the curve representing an edge.

vector $\Gamma_{\gamma_{12}(t_0,1)}(w_2) \in T_{\gamma_1(t_0)}\mathcal{M}$ along the curve $\gamma_{12}(t_0, s) : \{t_0\} \times [0,1] \to \mathcal{M}$. After performing parallel transport, we define the tangent attention score between γ_1 and γ_2. Figure 1b illustrates how to compute a tangent attention score.

Definition 3.1. The tangent attention score between two curves $\gamma_1, \gamma_2 : [0,1] \to \mathcal{M}$ with a choice of a function $\gamma_{12} : [0,1] \times [0,1] \to \mathcal{M}$ is

$$e_{\gamma_{12}}(\gamma_1, \gamma_2) := \int_0^1 g_{\gamma_1(t)} \left(\gamma_1'(t), \Gamma_{\gamma_{12}(t,1)}(\gamma_2'(t))\right) dt.$$

By imposing suitable topological structures on finite directed graphs, we can compute a tangent attention score between two edges of a directed graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ (including self-loops) as a function of node features $X : \mathcal{V} \to \mathbb{R}^h$. See Appendix A.9 for details. We also choose a continuous function $X_\gamma : \mathcal{G} \to \mathcal{M}$ which extends the node feature function X. Given two edges $e_1, e_2 \in \mathcal{E}$ with the corresponding paths $\gamma_{e_1}, \gamma_{e_2} : [0,1] \to \mathbb{R}^h$, we define $\gamma_{e_1,e_2}(t,s) := s \cdot \gamma_{e_1}(t) + (1-s) \cdot \gamma_{e_2}(t)$. Tangent attention scores are symmetric in their two edges. Section A.9 gives the argument. We can quantify the differences among node features of $\mathcal{G}$ using these scores.

Definition 3.2. The tangent attention score matrix of $\mathcal{G}$, denoted as $e_{\mathcal{G},\gamma}$, is the $|\mathcal{E}| \times |\mathcal{E}|$ real symmetric matrix whose (e_1, e_2)-th entry for $e_1, e_2 \in \mathcal{E}$ is the tangent attention score $e_{\gamma_{e_1,e_2}}(\gamma_{e_1}, \gamma_{e_2})$.

Using $e_{\mathcal{G},\gamma}$, we recover several classes of previously studied attention functions. The detailed version of the theorem stated below can be found in Appendix A.10.

Theorem 3.3. *There exist continuous extensions $X_\gamma : \mathcal{G} \to \mathcal{M}$ such that $e_{\mathcal{G},\gamma}$ respectively recovers dot-product attention, single-layer feedforward linear layer, and dynamic graph attention [38, 39, 3].*

3.2 Learnable Indefinite Metric

Because tangent attention scores encompass various types of previously studied attention scores, a wide class of novel graph attention networks can be constructed by carefully choosing the curves representing the edges of $\mathcal{G}$, choosing the symmetric matrix S, and choosing which subsets of entries of the tangent attention score matrix to use.

Building on this, our proposed *Pseudo-Riemannian Attention Network* (PRAT) architecture reformulates classical attention networks with tangent attention scores defined from the curve $\gamma_e^{[1]} : [0,1] \to \mathcal{M}$ as shown in Theorem 3.3 and Appendix A.10. The following equation governs the PRAT algorithm.

$$h_v^{(i+1)} := \bigoplus_{k=1}^{K} \sigma \left(\sum_{w \in \mathcal{N}_v \cup \{v\}} \alpha_{v,w}^{[i,k]} \mathbf{W_v}^{[i,k]} h_w^{(i)} \right), \qquad h_v^{(0)} := \psi_{enc}(X(v)) \tag{3}$$

For each $1 \leq k \leq K$,

$$\alpha_{v,w}^{[i,k]} := \frac{\exp(e(h_v^{(i)}, h_w^{(i)})^{[k]})}{\sum_{u \in \mathcal{N}_v \cup \{v\}} \exp(e(h_v^{(i)}, h_u^{(i)})^{[k]})}, \; e(h_v^{(i)}, h_u^{(i)})^{[k]} := \begin{cases} \Delta(h_{v,u}^{(i)})^\top \mathbf{D}^{[i,k]} \Delta(h_{v,u}^{(i)}) & \text{if } u \in \mathcal{N}_v \\ a^{[i,k]} (h_v^{(i)})^\top \mathbf{D}^{[i,k]} (h_v^{(i)}) & \text{if } v = u \end{cases}$$

where $\mathcal{N}_v$ is the set of nodes adjacent to the node v, $\Delta(h_{v,u}^{(i)})$ is equal to $h_v^{(i)} - h_u^{(i)}$ given $u \in \mathcal{N}_v$, and σ is the nonlinear activation function. $\psi_{enc} : \mathbb{R}^h \to \mathbb{R}^{h'}$ is a learnable function (e.g., multilayer perceptron) that maps node features to hidden states of dimension h'. The $a^{[i,k]}$ are learnable self-loop weights playing the role of a_v^2 in Equation (40), calibrated in Appendix F.3, and $\mathbf{D}^{[i,k]} \in \mathbb{R}^{h' \times h'}$ are symmetric matrices that serve as the metric tensor.

In Appendix B, we conceptualize the learning process of PRAT as a sequence of geometric transformations, from which we derive $\mathbf{D}^{[l,k]}$ for the k-th head at layer l as follows, where $\mathbf{W}^{[l,k]} \in \mathbb{R}^{|L| \times h'}$ is a learnable weight matrix, and $S^{|L|} \in \mathbb{R}^{|L| \times |L|}$ is a pseudo-metric on the latent manifold.

$$\mathbf{D}^{[l,k]} = (\mathbf{W}^{[l,k]})^\top S^{|L|} \mathbf{W}^{[l,k]}. \tag{4}$$

Following the theoretical motivation in Section 2.2, we instantiate the pseudo-metric $S^{|L|}$ as the graph signature matrix $S_\mathcal{G}$. The pullback then carries the label topology into the attention score. Substituting $S^{|L|} = S_\mathcal{G}$ into Equation (4) and performing the eigendecomposition $S_\mathcal{G} = Q^\top \Lambda_\mathcal{G} Q$, we can reformulate the induced metric tensor. By absorbing the orthogonal eigenvector matrix Q into the learnable weight matrix, i.e., defining $\mathbf{W}'^{[l,k]} := Q\mathbf{W}^{[l,k]}$, we arrive at:

$$\mathbf{D}^{[l,k]} = (\mathbf{W}^{[l,k]})^\top (Q^\top \Lambda_\mathcal{G} Q) \mathbf{W}^{[l,k]} = (Q\mathbf{W}^{[l,k]})^\top \Lambda_\mathcal{G} (Q\mathbf{W}^{[l,k]}) = (\mathbf{W}'^{[l,k]})^\top \Lambda_\mathcal{G} \mathbf{W}'^{[l,k]}. \tag{5}$$

In Equation (5), $\Lambda_\mathcal{G}$ is a diagonalized graph signature matrix which directly encodes the homophilic or heterophilic character of the graph. In practical semi-supervised settings, however, the ground-truth labels for the entire graph are inaccessible, so we compute the empirical graph signature $S_{\mathcal{G},\text{train}}$ and its eigenvalues $\Lambda_{\mathcal{G},\text{train}}$ on the subgraph induced by the training nodes. We initialize the diagonal matrix Λ using a normalized version of $\Lambda_{\mathcal{G},\text{train}}$, i.e. $\Lambda_{\text{init}} = \frac{\Lambda_{\mathcal{G},\text{train}}}{\text{std}(\Lambda_{\mathcal{G},\text{train}})} \cdot \frac{1}{\sqrt{|L|}}$, where std$(\cdot)$ is the standard deviation, set to 1 if zero. We leave these values free and treat the diagonal matrix Λ as a learnable parameter. This design allows the model to fine-tune the geometric prior derived from the training set and generalize it to the unseen test nodes. Labels enter only here, and the forward pass never reads one. What the initialization has to supply is an informed indefinite prior, not ground truth: Section J.2 shows that a signature estimated from k-means pseudo-labels on node features recovers the label-based accuracy on both indefinite benchmarks, and Section K shows how far the estimate and the accuracy survive when the training labels are few or wrong.

4 Experiments

4.1 Datasets and Baselines

Standard Benchmark Datasets. We first evaluate PRAT on node classification across eight standard benchmarks covering a broad spectrum of homophily levels. We use Texas and Cornell from WebKB [3], Chameleon and Squirrel [30], and Actor [35] as heterophilic datasets, and Cora, Citeseer, and PubMed [32, 45] as homophilic counterparts. Detailed statistics, including edge homophily levels (h_E) and graph signature indices ($s_\mathcal{G}$), are summarized in Table 1. The homophilic graphs here carry strictly positive signatures and the heterophilic ones other than Actor carry negative components. That association only motivates the design; the ablation in Table 5, not this correlation, shows that the sign structure produces the gain.

Platonov Benchmark Datasets. We additionally evaluate PRAT on seven datasets curated by Platonov et al. [28]: (i) the *filtered* versions of Chameleon and Squirrel, in which duplicate nodes causing severe train–test leakage in the original benchmarks are removed; and (ii) five large-scale heterophilic datasets, *roman-empire*, *amazon-ratings*, *minesweeper*, *tolokers* and *questions*, spanning 10K to 49K nodes They test PRAT at scale without the data-quality artifacts of the original Wikipedia networks. Statistics are in Table 2 and descriptions in Appendix C.

[3] `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb`

Table 1: Standard benchmark statistics (h_E, $s_{\mathcal{G}}$).

Dataset	Cora	PubMed	Citeseer	Chameleon	Squirrel	Actor	Cornell	Texas
Nodes	2,708	19,717	3,327	2,277	5,201	7,600	183	183
Edges	5,278	44,324	4,552	36,101	217,073	30,019	298	325
Classes	7	3	6	5	5	5	5	5
h_E	0.810	0.802	0.736	0.230	0.222	0.217	0.123	0.061
$s_{\mathcal{G}}$	(7,0,0)	(3,0,0)	(6,0,0)	(3,0,2)	(2,2,1)	(1,4,0)	(3,0,2)	(2,0,3)

Table 2: Platonov benchmark statistics [28].

Dataset	Cham.-filt.	Sq.-filt.	roman-empire	amazon-ratings	minesweeper	tolokers	questions
Nodes	890	2,223	22,662	24,492	10,000	11,758	48,921
Edges	8,854	57,134	32,927	93,050	39,402	519,000	153,540
Classes	5	5	18	5	2	2	2
h_E	0.235	0.220	0.046	0.380	0.683	0.595	0.840
$s_{\mathcal{G}}$	(3,0,2)	(2,1,2)	(6,7,5)	(3,2,0)	(1,1,0)	(2,0,0)	(1,1,0)

Baselines on Standard Benchmarks. We compare PRAT against a diverse set of architectures spanning multiple paradigms: (1) **Standard GNNs**: GCN [16], GIN [43], and GraphSAGE [12]; (2) **Attention-based models**: GAT [39], GATv2 [3], Scaled Dot-Product (SDP) Attention [38, 33], and Graph Transformer with and without ego-neighbor separation (GT, GT-sep) [33, 49]; (3) **Heterophily-adapted GNNs**: CPGNN [50], FAGCN [1], Geom-GCN [27], GPR-GNN [7], BernNet [14], and H_2GCN [49]; (4) **Geometric models**: Q-GCN [42] on a pseudo-hyperboloid and HAT [47] on the Poincaré ball.

Baselines on Platonov Benchmarks. On the Platonov benchmarks, we compare PRAT against the best of three ResNet variants and the 14 GNN baselines reported in Platonov et al. [28]. (1) **Standard GNNs**: GCN [16] and GraphSAGE [12]; (2) **Attention-based models**: GAT [39], GAT-sep [49], Graph Transformer (GT) [33], and GT-sep [49], where the "-sep" suffix denotes the ego- and neighbor-embedding separation trick of Zhu et al. [49]; (3) **Heterophily-adapted GNNs**: H_2GCN [49], CPGNN [50], GPR-GNN [7], FSGNN [23], GloGNN [19], FAGCN [1], GBK-GNN [10], and JacobiConv [40].

4.2 Experimental Setup

Standard Benchmarks. We evaluate performance using 10 splits. For heterophilic datasets, we adopt the fixed 48%/32%/20% train/validation/test splits provided by Pei et al. [27]. For homophilic datasets, we generate random 60%/20%/20% splits, avoiding the selection bias of a fixed set. All graphs are undirected with self-loops added, and the seed is fixed across splits. Baselines and PRAT share one sweep over a predefined range (hidden dimension, learning rate, dropout), and we report mean test accuracy with standard deviation at the best-validation step.[4] Detailed configurations are provided in Appendix F.1.

Platonov Benchmarks. For the seven Platonov benchmarks, we use the official training pipeline of Platonov et al. [28][5] as our experimental harness, where PRAT is implemented as a drop-in attention replacement inside the Platonov backbone. Data loading, the ten predefined splits, validation-based model selection and 10-split mean $\pm$ std reporting all follow Platonov et al. [28] exactly, so every deviation we report is across splits at a fixed seed. Section G.2 adds three seeds and finds seed variance the smaller of the two. PRAT is hyperparameter-tuned inside the same harness published by Platonov et al. [28]. Detailed configurations are provided in Appendix F.1.

Cost. A PRAT layer has GAT's asymptotic cost, since the metric it adds is diagonal and acts on a latent space whose dimension is the number of classes (Section E gives the expressions). Measured on the two largest graphs at equal backbone, width and depth, a training step takes 88.7 ms on questions against 91.2 for GAT and 175.2 for GT, and 26.7 ms on tolokers against 22.8 and 32.9 (Section E.3). At this scale, PRAT costs as much as GAT.

4.3 Main Results

Table 3 reports node classification accuracy on the eight standard benchmarks. PRAT is best on Actor and Texas, second on Squirrel, third on Chameleon and fourth on Cornell, and among the homophilic datasets it is best on Citeseer and third on PubMed; Section G.1 reports the pairwise tests, four of which are ties. PRAT attends over *1-hop* neighborhoods only, yet it exceeds the multi-hop spectral filters BernNet and GPR-GNN on Texas, Squirrel and Actor, so the margin there does not come from a larger receptive field. Section H reads the GT and geometric rows.

[4] For Geom-GCN, we report results from Yan et al. [44].

[5] `https://github.com/yandex-research/heterophilous-graphs`

Table 3: Performance of PRAT and baselines on benchmark datasets. The top 3 performances are highlighted in **First**, Second, and Third.

	Homophilic			Heterophilic				
	Cora	PubMed	Citeseer	Chameleon	Squirrel	Actor	Cornell	Texas
h_E	0.810	0.802	0.736	0.230	0.222	0.217	0.123	0.061
GCN	88.32±1.02	88.75±0.43	77.69±1.06	67.43±2.45	54.95±1.19	30.56±0.70	44.32±6.75	58.38±4.86
GIN	87.38±1.56	89.20±0.51	76.82±1.29	62.52±2.61	53.56±1.66	30.97±1.56	41.62±6.86	59.73±3.51
SAGE	88.80±1.04	89.77±0.45	77.66±0.81	64.91±1.81	43.92±1.87	36.17±1.26	69.46±8.64	81.35±3.72
GAT	88.32±1.09	88.87±0.41	77.10±1.27	69.67±1.67	55.18±2.13	31.61±1.99	50.54±7.55	64.59±6.67
GATv2	87.99±0.93	89.63±0.47	76.97±0.96	67.94±2.34	**59.79±1.83**	32.97±1.37	54.32±9.70	66.76±6.63
SDP	87.99±1.30	89.45±0.55	76.85±0.97	67.06±1.53	56.42±3.30	34.19±0.82	52.97±8.04	65.68±5.28
GT	87.64±0.94	89.10±0.50	76.76±1.31	68.71±2.35	56.83±1.40	35.70±0.97	66.22±4.81	76.76±6.14
GT-sep	87.92±1.31	89.13±0.45	77.69±1.27	60.59±1.78	44.07±1.29	35.86±0.67	70.00±2.97	77.03±5.59
CPGNN	87.24±1.07	89.78±0.48	76.83±1.00	56.18±2.93	42.34±1.07	33.95±1.08	64.05±6.63	77.57±7.55
FAGCN	88.75±1.38	89.03±0.55	77.46±1.11	65.07±1.56	49.55±1.92	36.51±1.02	60.27±6.40	71.89±7.17
Geom-GCN	85.35±1.57	89.95±0.47	78.02±1.15	60.00±2.81	38.15±0.92	31.59±1.15	60.54±3.67	66.76±2.72
GPR-GNN	89.43±1.06	90.01±0.60	77.76±0.83	68.18±1.92	51.46±1.34	35.38±0.98	74.59±4.12	83.51±5.78
BernNet	**89.67±1.04**	90.13±0.51	77.79±1.38	**70.77±1.63**	55.38±1.46	35.20±1.14	**76.76±3.68**	83.78±4.27
H_2GCN	89.61±1.15	**90.14±0.51**	76.86±1.35	59.30±2.12	35.55±1.58	34.72±1.18	74.86±4.68	82.43±5.28
Q-GCN	87.70±1.19	88.22±0.40	77.07±1.50	59.67±2.09	46.27±1.22	29.65±0.70	45.68±4.90	61.62±5.10
HAT	88.36±1.09	89.89±0.53	77.90±1.35	57.48±2.03	38.87±1.40	34.34±1.03	66.49±4.39	78.65±5.05
PRAT (ours)	88.42±1.17	90.03±0.47	**78.06±1.37**	68.73±1.74	56.96±1.51	**37.26±1.44**	71.89±4.71	**84.32±5.10**

Table 4: Performance on Platonov et al. [28] benchmarks. Accuracy (↑) is reported for roman-empire, amazon-ratings, Cham.-filt., Sq.-filt.; ROC AUC (↑) for minesweeper, tolokers, questions. †Best per dataset of the graph-agnostic ResNet of Platonov et al. [28] and its graph-aware variants ResNet+SGC and ResNet+adj.

	Filtered Wikipedia		Large-scale heterophilic				
	Cham.-filt.	Sq.-filt.	roman-empire	amazon-ratings	minesweeper	tolokers	questions
h_E	0.235	0.220	0.046	0.380	0.683	0.595	0.840
ResNet†	41.01±4.54	38.37±1.99	73.90±0.51	51.83±0.57	70.88±0.90	80.70±0.97	75.81±0.96
GCN	40.89±4.12	39.47±1.47	73.69±0.74	48.70±0.63	89.75±0.52	83.64±0.67	76.09±1.27
SAGE	37.77±4.14	36.09±1.99	85.74±0.67	53.63±0.39	93.51±0.57	82.43±0.44	76.44±0.62
GAT	39.21±3.08	35.62±2.06	80.87±0.30	49.09±0.63	92.01±0.68	83.70±0.47	77.43±1.20
GAT-sep	39.26±2.50	35.46±3.10	88.75±0.41	52.70±0.62	**93.91±0.35**	83.78±0.43	76.79±0.71
GT	38.87±3.66	36.30±1.98	86.51±0.73	51.17±0.66	91.85±0.76	83.23±0.64	77.95±0.68
GT-sep	40.31±3.01	36.66±1.63	87.32±0.39	52.18±0.80	92.29±0.47	82.52±0.92	78.05±0.93
H_2GCN	26.75±3.64	35.10±1.15	60.11±0.52	36.47±0.23	89.71±0.31	73.35±1.01	63.59±1.46
CPGNN	33.00±3.15	30.04±2.03	63.96±0.62	39.79±0.77	52.03±5.46	73.36±1.01	65.96±1.95
GPR-GNN	39.93±3.30	38.95±1.99	64.85±0.27	44.88±0.34	86.24±0.61	72.94±0.97	55.48±0.91
FSGNN	40.61±2.97	35.92±1.32	79.92±0.56	52.74±0.83	90.08±0.70	82.76±0.61	**78.86±0.92**
GloGNN	25.90±3.58	35.11±1.24	59.63±0.69	36.89±0.14	51.08±1.23	73.39±1.17	65.74±1.19
FAGCN	41.90±2.72	**41.08±2.27**	65.22±0.56	44.12±0.30	88.17±0.73	77.75±1.05	77.24±1.26
GBK-GNN	39.61±2.60	35.51±1.65	74.57±0.47	45.98±0.71	90.85±0.58	81.01±0.67	74.47±0.86
JacobiConv	39.00±4.20	29.71±1.66	71.14±0.42	43.55±0.48	89.66±0.40	68.66±0.65	73.88±1.16
PRAT (ours)	**42.69±2.66**	39.60±2.25	**90.08±0.62**	**53.78±0.66**	92.24±0.68	**85.04±0.48**	78.04±1.19

Table 4 extends the comparison to Platonov et al. [28] benchmarks. PRAT is best or statistically tied on four of the seven datasets (Cham.-filt., roman-empire, amazon-ratings, tolokers), ranks 2nd on Sq.-filt., and sits 0.05pp below the third-placed model on minesweeper and 0.01pp below GT-sep at rank 3 on questions. Only two of those four are wins; the other two are ties: against SAGE on amazon-ratings a Welch test gives $p = 0.55$, and against FAGCN on filtered Chameleon $p = 0.52$ (Section G.1). Across the seven benchmarks PRAT therefore outperforms or remains competitive with all standard and heterophily-adapted GNNs reported by Platonov et al. [28]. The ResNet row is graph-aware on every dataset (ResNet+SGC or ResNet+adj) and outscores GCN on filtered Chameleon, yet PRAT leads it on all seven, by 1.23 to 21.36 points. On filtered Chameleon, where the duplicate nodes responsible for train–test leakage in the original benchmark are removed, the gain survives; removing that leakage shows the gain is not an artifact of it, though it does not by itself rule out memorization of other kinds. These results support our central thesis: the geometry of the latent space, not the depth of aggregation, is the dominant factor for learning under heterophily.

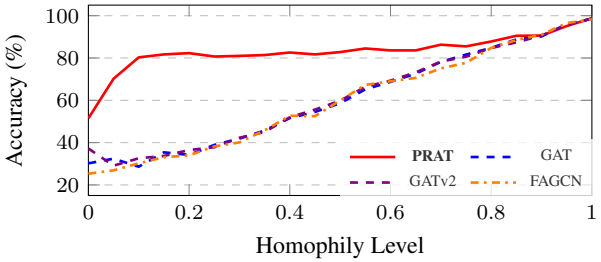


Figure 2: Accuracy across edge homophily levels, within the attention family.

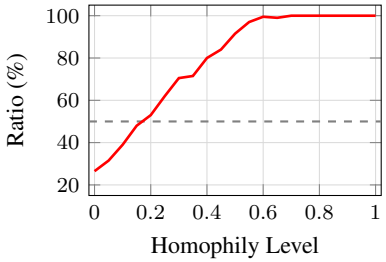
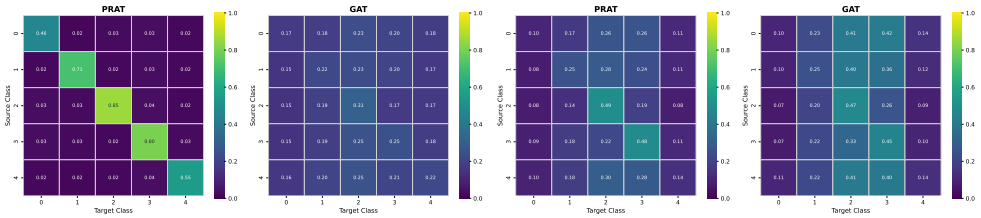


Figure 3: Ratio of positive eigenvalues of the learned metric in PRAT across varying homophily levels.



(a) $h_E = 0.1$ (heterophilic)

(b) $h_E = 0.9$ (homophilic)

Figure 4: Attention score distributions of PRAT and GAT. Each heatmap gives mean attention from source class (row) to target class (column).

4.4 Validation of Geometric Hypothesis Based on Synthetic Data

Generated graphs let us vary one structural property at a time. Nodes, edges, features and classes are held fixed across all of them, and every cell is five random $60/20/20$ splits. Section D gives the generator.

Varying Homophily level We generated synthetic graphs with edge homophily $h_E \in \{0, 0.05, \ldots, 1.0\}$ and evaluated classification accuracy. The comparison stays inside the attention family, so the receptive field is fixed and only the metric differs: GAT and GATv2 keep it positive definite, FAGCN signs the aggregation weights without touching the geometry, and PRAT makes the metric itself indefinite. In Figure 2, PRAT holds its accuracy across the sweep while the other three fall away together below $h_E = 0.5$. The gap to FAGCN averages 40.6 points over the nine levels at $h_E \leq 0.4$. The curves converge above $h_E = 0.8$, where a positive-definite metric is no longer a mismatch. Signed weights alone do not substitute for an indefinite metric.

Figure 3 tracks the ratio of positive eigenvalues in the learned metric: as homophily falls the negative share rises, so the signature the model settles on is not imposed but follows the data. On real graphs the learned Λ instead keeps the initial sign of every entry with $|\lambda| > 10^{-4}$ in $\Lambda_{\mathcal{G},\text{train}}$, which Section J.1 reports.

Figure 4 visualizes the attention scores of PRAT and GAT, aggregated by source-target label pairs across all edges. Unlike GAT, PRAT concentrates attention on same-class connections (the diagonal) at $h_E = 0.1$, suppressing dissimilar neighbors while keeping the node's own signal, but does not do so at $h_E = 0.9$. The metric changes how attention uses the same neighborhood.

Varying Graph Signature Index To investigate the influence of higher-order structural patterns beyond simple homophily, we generated synthetic graphs with a fixed homophily level ($h_E = 0.5$) but varying graph signature indices $s_{\mathcal{G}} \in \{(5, 0, 0), (4, 0, 1), (3, 0, 2)\}$.

In Figure 5, baseline performance fluctuates significantly with the graph signature index. Specifically, the standard deviation of accuracy across different signatures is 3.99 for GAT and 3.42 for GATv2. PRAT varies by 0.68. Two readings follow: PRAT is the less sensitive of the three to latent structure, and $s_{\mathcal{G}}$ separates graphs that a scalar h_E cannot, since these three share an h_E and behave differently.

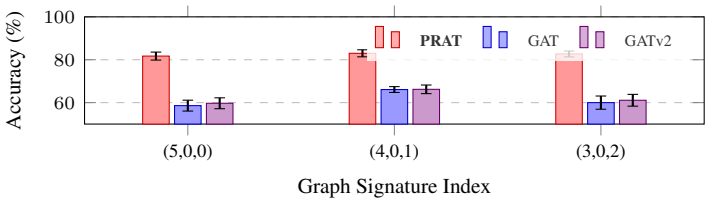


Figure 5: Performance comparison under fixed $h_E = 0.5$ across different graph signature indices.

4.5 Ablation Study

To evaluate the effectiveness of graph-aware metric initialization in PRAT, we ran an ablation with two variants: (1) PRAT_{pos}, whose learnable Λ is constrained to be positive definite, and (2) $\text{PRAT}_{\text{fixed}}$, whose Λ is frozen at its signature initialization.

Table 5: Ablation study results.

Dataset	PRAT	$\text{PRAT}_{\text{fixed}}$	PRAT_{pos}
Cora	88.42±1.17	**88.49±1.27**	88.18±1.10
PubMed	**90.03±0.47**	89.94±0.46	89.87±0.45
Citeseer	78.06±1.37	77.93±1.38	**78.09±1.45**
Chameleon	**68.73±1.74**	68.40±2.42	65.64±1.60
Squirrel	**56.96±1.51**	54.01±2.08	53.47±1.43
Actor	**37.26±1.44**	36.93±1.21	36.69±0.99
Cornell	**71.89±4.71**	55.15±8.97	57.58±8.87
Texas	**84.32±5.10**	74.89±5.70	76.78±2.90
Average	**71.96**	68.22	68.29

In Table 5 the gap between PRAT and PRAT_{pos} falls on the graphs whose signature has negative eigenvalues: Chameleon, Squirrel, Cornell and Texas all lose under the constraint, while Cora, PubMed and Citeseer, whose signatures are strictly positive, are unchanged. Section I.1 repeats the comparison on the Platonov suite with an exact paired test: both significant losses are on indefinite graphs and no graph without a negative eigenvalue changes significantly, while filtered Squirrel, whose negative eigenvalues sit near zero, ties. A negative signature is therefore necessary for a significant loss, and Section I.2 finds no significant rank correlation between n_- and what signed or high-pass baselines gain, so we offer $s_{\mathcal{G}}$ as a diagnostic and not as a predictor.

The two variants isolate different things. PRAT_{pos} removes the freedom to be indefinite and starts Λ at a scaled identity, so it measures the sign structure. $\text{PRAT}_{\text{fixed}}$ freezes Λ but not the metric, since $\mathbf{D} = (\mathbf{W}')^\top \Lambda \mathbf{W}'$ keeps adapting through the projection, so it measures whether the inertia has to be supplied at initialization. Its drop sits on the small graphs, 16.7 points on Cornell and 9.4 on Texas, where the training-split signature varies most; on the large Platonov graphs the two are within noise. Learning Λ earns its place where the signature is noisy, not where it is clean.

Which part of the signature carries the effect differs by graph. In Section J.1 the only control that loses decisively on roman-empire deletes the negative directions and keeps the magnitudes, 2.26 points at $p = 0.002$, while sign-only, sign-flipped and three random draws stay within 0.7. On filtered Chameleon the magnitudes matter too, since every random draw loses 1.63 to 2.62 points.

5 Related Works

GNNs for Heterophily. Standard GNNs assume homophily, acting as low-pass filters that smooth features across edges. Heterophily-adapted models answer this architecturally: Geom-GCN [27] and H_2GCN [49] separate aggregation from the immediate neighborhood, spectral methods such as GPR-GNN [7] and BernNet [14] learn polynomial filters that mix high- and low-frequency signals, and attention-based models use signed edge weights [1, 44] to represent repulsion. All of these operate within a Euclidean space.

Pseudo-Riemannian graph models. We are not the first to bring an indefinite geometry to graphs. Q-GCN [42] embeds node features on a pseudo-hyperboloid $\mathcal{Q}_\beta^{s,t}$ and convolves there through diffeomorphic exponential and logarithmic maps; Q-GT [18] carries the same ambient space into a transformer with linear attention. Three differences separate them from PRAT. First, where the indefinite form lives: in both it is the canonical scalar product of the manifold the nodes are embedded on, whereas in PRAT it is the attention logit itself, a signed squared pseudo-distance $\Delta^\top \mathbf{D} \Delta$. Q-GT's attention weights come from a positive-definite kernel, so its indefinite product never reaches a score. Second, how the signature is chosen: Q-GCN picks the number of time dimensions by a curvature

heuristic and a sweep, which its own conclusion names as an open limitation, and Q-GT selects its own signature (p, q), the counts of positive and negative dimensions, offline by matching a sectional-curvature distribution. Neither criterion reads the label topology, so neither can see heterophily, while $s_{\mathcal{G}}$ is built from that topology and Λ is learned from the spectrum of $S_{\mathcal{G},\text{train}}$. Third, coverage: Q-GCN's labeled evaluation is citation networks and Airport, Q-GT adds two non-homophilous social graphs, and neither reports the WebKB, Wikipedia or Platonov suites used here. We claim a learnable label-initialized indefinite metric inside attention together with a diagnostic for when it should help, not priority for pseudo-Riemannian geometry on graphs. ACM-GCN [21] is closest on the channel side, mixing aggregation, diversification and identity channels node-wise, which puts the sign in the channel domain rather than in a metric. Its published numbers use different splits, so we cite it as a counterpart, not a comparable row.

Geometric Graph Learning. Hyperbolic GNNs [6, 47] use constant negative curvature for hierarchical structure, and ARGNN [41] makes the metric node-wise and learnable. Both stay Riemannian ($g > 0$); only an indefinite metric admits a negative squared distance [17].

Graph Attention. Standard L_2 attention can be reformulated as an SDP and is biased accordingly [22]. DGAT [20] uses spectral distances for rewiring in a separate preprocessing step, and global Graph Transformers [46, 29, 9] reach long range at quadratic cost while remaining Euclidean. PRAT instead reads attention as a tangent-space interaction on a learnable manifold.

Statistical Foundations. The Generalised Random Dot Product Graph (GRDPG) [31] shows statistically that indefinite inner products are needed to model disassortative networks. Theorem 2.5 reaches a complementary conclusion from the geometric side. GRDPG gives a fixed spectral embedding. PRAT learns the indefinite metric end-to-end, so attention moves with it at every layer.

Conclusion

We examined the positive-definite geometric prior carried by diffusion, hyperbolic and dot-product attention models, and asked what it costs when the labels induce an indefinite form. Theorem 2.5 answers in one direction: where $S_{\mathcal{G}}$ is indefinite and the pushforward is surjective, no diffeomorphism of the target turns the Riemannian pullback into $f^* S_{\mathcal{G}}$. That is a statement about those model families and about graphs with a negative signature, not about heterophily as a whole.

The empirical picture matches that scope. The Graph Signature Index says which graphs carry a negative signature. In the paired Platonov ablation every significant loss from constraining the metric falls on such a graph, and no other graph changes significantly. The condition is necessary for a detectable loss. Several of our heterophilic benchmarks do not meet it, so the diagnosis, not heterophily as such, is a precondition of the indefinite metric's advantage.

Limitations and future work. The signature is an initialization prior rather than an architectural requirement, since the forward pass never reads labels and Λ is an ordinary learnable parameter. Even so, four limits are worth stating. It is computed from training-node labels, which ties the scheme as presented to (semi-)supervised node classification. Section J.2 weakens that, but a full self-supervised treatment, objective as well as initialization, is outside our scope. Inductive use needs no recomputation in the forward pass, yet signature shift between training and test graphs is open. Graph-level tasks are out of reach as the method stands, since $S_{\mathcal{G}}$ is undefined without node labels. Finally, $\operatorname{rank}(\mathbf{D}) \leq |L|$ bounds the structure the metric can carry when classes are few, and on the binary benchmarks the inertia reduces to a single bit. Dynamic curvature and directed graphs are natural next steps.

Acknowledgments and Disclosure of Funding

This research was supported by the "Advanced GPU Utilization Support Program" funded by the Government of the Republic of Korea (Ministry of Science and ICT). Sun Woo Park would like to sincerely thank Max Planck Institute for Mathematics for its kind and generous support. The authors declare no competing interests.

References

[1] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *AAAI*. AAAI Press, 2021.

[2] Cristian Bodnar, Francesco Di Giovanni, Benjamin Paul Chamberlain, Pietro Liò, and Michael M. Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. In *Advances in Neural Information Processing Systems*, volume 35, pages 18527–18541, 2022.

[3] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=F72ximsx7C1.

[4] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34 (4):18–42, 2017. doi: 10.1109/MSP.2017.2693418.

[5] Benjamin Paul Chamberlain, James Rowbottom, Maria Goronova, Stefan Webb, Emanuele Rossi, and Michael M Bronstein. Grand: Graph neural diffusion. *Proceedings of the 38th International Conference on Machine Learning, (ICML) 2021, 18-24 July 2021, Virtual Event*, 2021.

[6] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 4869–4880, 2019.

[7] 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.

[8] Yun Young Choi, Sun Woo Park, Youngho Woo, and U Jin Choi. Cycle to clique (cy2c) graph neural network: A sight to see beyond neighborhood aggregation. In *The eleventh international conference on learning representations*, 2022.

[9] Yun Young Choi, Sun Woo Park, Minho Lee, and Youngho Woo. Topology-informed graph transformer. In *Proceedings of the Geometry-grounded Representation Learning and Generative Modeling Workshop (GRaM)*, volume 251 of *Proceedings of Machine Learning Research*, pages 20–34. PMLR, 2024.

[10] Lun Du, Xiaozhou Shi, Qiang Fu, Xiaojun Ma, Hengyu Liu, Shi Han, and Dongmei Zhang. GBK-GNN: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In *Proceedings of the ACM Web Conference 2022 (WWW)*, 2022.

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

[12] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, 2017.

[13] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[14] Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *NeurIPS*, 2021.

[15] Feng Ji, Yanan Zhao, Kai Zhao, Hanyang Meng, Jielong Yang, and Wee Peng Tay. Rethinking graph neural networks from a geometric perspective of node features. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lBMRmw59Lk.

[16] 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.

[17] Marc Law and Jos Stam. Ultrahyperbolic representation learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1668–1678. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/123b7f02433572a0a560e620311a469c-Paper.pdf.

[18] Viet Quan Le and Cuong Van Ta. Pseudo-riemannian graph transformer. In *Advances in Neural Information Processing Systems*, volume 38, pages 15951–15983, 2025.

[19] 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 *International Conference on Machine Learning (ICML)*, 2022.

[20] Qincheng Lu, Jiaqi Zhu, Sitao Luan, and Xiao-Wen Chang. Representation learning on heterophilic graph with directional neighborhood attention, 2024. URL https://arxiv.org/abs/2403.01475.

[21] 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*, 2022.

[22] Liheng Ma, Soumyasundar Pal, Yingxue Zhang, Philip H. S. Torr, and Mark Coates. Plain transformers can be powerful graph learners. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=bEmDvP0fdv.

[23] Sunil Kumar Maurya, Xin Liu, and Tsuyoshi Murata. Simplifying approach to node classification in Graph Neural Networks. *Journal of Computational Science*, 62:101695, 2022.

[24] John Douglas Moore. Lectures on differential geometry. Lecture notes, University of California, Santa Barbara. https://web.math.ucsb.edu/~moore/riemanniangeometry.pdf, 2009.

[25] Sun Woo Park, Yun Young Choi, Dosang Joe, U Jin Choi, and Youngho Woo. The pwlr graph representation: A persistent weisfeiler-lehman scheme with random walks for graph classification. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pages 287–297. PMLR, 2022.

[26] Sun Woo Park, Yun Young Choi, U Jin Choi, and Youngho Woo. A functorial formulation of neighborhood aggregating deep learning. *arXiv preprint arXiv:2604.24672*, 2026.

[27] 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.

[28] 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.

[29] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

[30] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

[31] Patrick Rubin-Delanchy, Joshua Cape, Minh Tang, and Carey E. Priebe. A statistical interpretation of spectral embedding: the generalised random dot product graph. *Journal of the Royal Statistical Society Series B*, 84(4):1446–1473, 2022.

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

[33] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1548–1554. International Joint Conferences on Artificial Intelligence Organization, 2021. doi: 10.24963/ijcai.2021/214.

[34] J. J. Sylvester. Xix. a demonstration of the theorem that every homogeneous quadratic polynomial is reducible by real orthogonal substitutions to the form of a sum of positive and negative squares. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 4(23):138–142, 1852.

[35] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816, 2009.

[36] Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=7UmjRGzp-A`.

[37] Loring Tu. *Differential Geometry: Connections, Curvature, and Characteristic Classes*. Springer, 2017.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[39] 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`.

[40] Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *International Conference on Machine Learning (ICML)*, 2022.

[41] Xudong Wang, Tongxin Li, Chris Ding, and Jicong Fan. Adaptive riemannian graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

[42] Bo Xiong, Shichao Zhu, Nico Potyka, Shirui Pan, Chuan Zhou, and Steffen Staab. Pseudo-riemannian graph convolutional networks. In *Advances in Neural Information Processing Systems*, 2022.

[43] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=ryGs6iA5Km`.

[44] Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1287–1292. IEEE, 2022.

[45] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

[46] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

[47] Yiding Zhang, Xiao Wang, Chuan Shi, Xunqiang Jiang, and Yanfang Fanny Ye. Hyperbolic graph attention network. *IEEE Transactions on Big Data*, 2021.

[48] Yilun Zheng, Sitao Luan, and Lihui Chen. What is missing for graph homophily? disentangling graph homophily for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 37, pages 68406–68452, 2024.

[49] 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 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7793–7804. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/58ae23d878a47004366189884c2f8440-Paper.pdf.

[50] Jiong Zhu, Ryan A. Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K. Ahmed, and Danai Koutra. Graph neural networks with heterophily. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11168–11176, May 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17332.

A Theoretical analysis and proofs

A.1 Normalization of the Graph Signature Matrix

The (i, j)-th entry of $S_{\mathcal{G}}$ is defined as $s_{ij} = |E_{ij}|/(n_i \cdot n_j)$, where $|E_{ij}|$ is the number of ordered pairs (v, w) of adjacent nodes with $l(v) = i$ and $l(w) = j$ (so an edge inside class i is counted twice, which is the factor 2 of the diagonal entries in Appendix A.6), and n_i, n_j are the class sizes. This quantity is the *edge density* between classes i and j: the probability that a randomly chosen pair (v, w) with $l(v) = i$, $l(w) = j$ is connected. This normalization is standard in stochastic block models (SBMs) and carries a clear physical interpretation: positive eigenvalues correspond to over-connected class pairs (homophilic tendency), while negative eigenvalues indicate under-connected pairs (heterophilic tendency).

The signature does not depend on the normalization by $n_i \cdot n_j$. Write C for the matrix of raw counts $|E_{ij}|$ and $N_L := \operatorname{diag}(n_1, \ldots, n_{|L|})$. Since every class is nonempty, N_L is invertible, so $S_{\mathcal{G}} = N_L^{-1} C N_L^{-1}$ is congruent to C and has the same inertia by Theorem 2.2. The normalization acts on the eigenvalues instead. Those of C scale with class sizes, so class imbalance enters their magnitudes, while those of $S_{\mathcal{G}}$ are the spectrum of the connection-probability matrix above. Rescaled, they initialize Λ (Equation (5)), and Section J.1 shows that on filtered Chameleon their magnitudes matter beyond their signs. The critical homophily threshold in Theorem 2.3 is derived under this normalization, where the expected matrix takes the form $aI + b(J - I)$ with eigenvalues in closed form, and the resulting $h_E^* = \sum n_k^2/N^2$ explicitly accounts for class imbalance.

A.2 A different characterization of $S_{\mathcal{G}}$

Using the unfolding-tree perspective on local graph structure [25], we interpret entries of the graph signature matrix through depth 1 unfolding trees rooted at each node of $\mathcal{G}$.

Definition A.1. For any node $v \in \mathcal{V}$, we define the probability that a node with a given node label j is connected to a directed edge starting from node v as

$$\begin{aligned} p_{v,j} &:= \mathbb{P}\left[(v, w) \in \mathcal{E} \;\middle|\; \substack{w \in \mathcal{V} \\ l(w) = j}\right] \\ &= \frac{|\{w \in \mathcal{V} \mid l(w) = j, (v, w) \in \mathcal{E}\}|}{|\{w \in \mathcal{V} \mid l(w) = j\}|} \end{aligned} \tag{6}$$

Lemma A.2. *Given a graph $\mathcal{G}$, each matrix entry of $S_{\mathcal{G}}$ satisfies*

$$p_{i,j} = \frac{1}{|\{v \in \mathcal{V} \mid l(v) = i\}|} \sum_{\substack{v \in \mathcal{V} \\ l(v) = i}} p_{v,j} \tag{7}$$

Proof. We note that

$$\begin{aligned} p_{i,j} &:= \frac{1}{|\{v \in \mathcal{V} \mid l(v) = i\}|} \frac{|\{(v,w) \in \mathcal{E} \mid l(v) = i, l(w) = j\}|}{|\{v \in \mathcal{V} \mid l(v) = j\}|} \\ &= \frac{1}{|\{v \in \mathcal{V} \mid l(v) = i\}|} \frac{\sum_{\substack{v \in \mathcal{V} \\ l(v)=i}} |\{(v,w) \in \mathcal{E} \mid l(w) = j\}|}{|\{w \in \mathcal{V} \mid l(w) = j\}|} \\ &= \frac{1}{|\{v \in \mathcal{V} \mid l(v) = i\}|} \sum_{\substack{v \in \mathcal{V} \\ l(v)=i}} p_{v,j} \end{aligned} \tag{8}$$

□

In other words, each (i, j)-th entry of $S_{\mathcal{G}}$ is the expected value of the probability that a node with a preassigned label j lies in the depth 1 unfolding tree rooted at a node with label i.

A.3 An example of graph signature index for graphs with two discrete node labels

Theorem A.3 (2 Discrete Node Labels)**.** *Let $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ be an undirected graph without self-loops and with $\mathcal{E} \neq \emptyset$. Let $l : \mathcal{V} \to \{1, 2\}$ be the discrete node label function over the set of nodes of $\mathcal{G}$. Then the following correspondence holds between $s_{\mathcal{G}}$ and entries of $S_{\mathcal{G}} := (p_{i,j})_{1 \leq i,j \leq 2}$:*

$$\begin{aligned} s_{\mathcal{G}} = (2,0,0) &\iff p_{1,2} < \sqrt{p_{1,1}p_{2,2}} \\ s_{\mathcal{G}} = (1,1,0) &\iff p_{1,2} = \sqrt{p_{1,1}p_{2,2}} \\ s_{\mathcal{G}} = (1,0,1) &\iff p_{1,2} > \sqrt{p_{1,1}p_{2,2}} \end{aligned} \tag{9}$$

Proof. Let $S_{\mathcal{G}}$ be the graph signature matrix of $\mathcal{G}$. Its explicit form is given as

$$S_{\mathcal{G}} := \begin{pmatrix} p_{1,1} & p_{1,2} \\ p_{2,1} & p_{2,2} \end{pmatrix} \tag{10}$$

The characteristic polynomial of $S_{\mathcal{G}}$ is given by

$$(\lambda - p_{1,1})(\lambda - p_{2,2}) - p_{1,2}^2 = 0 \tag{11}$$

The eigenvalues of $S_{\mathcal{G}}$ are given by

$$\lambda = \frac{(p_{1,1} + p_{2,2}) \pm \sqrt{(p_{1,1} + p_{2,2})^2 - 4(p_{1,1}p_{2,2} - p_{1,2}^2)}}{2} \tag{12}$$

Since $\mathcal{E} \neq \emptyset$, the trace $p_{1,1} + p_{2,2}$ is positive whenever $p_{1,1}p_{2,2} \geq p_{1,2}^2$, as otherwise all three entries vanish. Hence, the graph signature $s_{\mathcal{G}}$ is determined by the sign of the term $p_{1,1}p_{2,2} - p_{1,2}^2$. □

A.4 Heuristics on heterophilic and homophilic graphs with two discrete node labels

The phase diagram for the graph $F : p_{1,2} - \sqrt{p_{1,1}p_{2,2}} = 0$ is shown in Figure 6, where the region below F characterizes graphs whose nodes with identical labels are more likely to be connected than those with distinct labels (i.e. homophilic graphs), whereas the region above F contains heterophilic graphs. A third region, defined by F, contains graphs whose nodes with identical labels are as likely to be connected as those with distinct labels. Note that the volume of the region below F over $0 \leq p_{1,1} \leq 1$ and $0 \leq p_{2,2} \leq 1$ is given by

$$\int_0^1 \int_0^1 \sqrt{p_{1,1}p_{2,2}} dp_{1,1} dp_{2,2} = \frac{4}{9}. \tag{13}$$

Consider the idealized model with 2 discrete node labels whose connectivity parameters $(p_{1,1}, p_{1,2}, p_{2,2})$ are uniformly distributed on $[0, 1]^3$, and let $P = (p_{i,j})$ be the symmetric 2×2 connectivity matrix, with inertia $\mathrm{In}(P)$. The integral computed above shows that

$$\mathbb{P}\left[\mathrm{In}(P) = (i,j,k)\right] = \begin{cases} \frac{5}{9} & \text{if } (i,j,k) = (1,0,1) \\ 0 & \text{if } (i,j,k) = (1,1,0) \\ \frac{4}{9} & \text{if } (i,j,k) = (2,0,0) \end{cases} \tag{14}$$

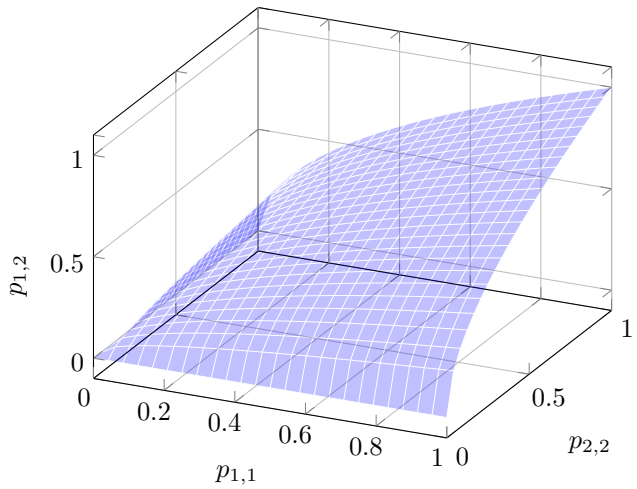


Figure 6: A phase diagram of graph signature index with two types of discrete node labels. The blue surface denotes the region where $F : p_{1,2} = \sqrt{p_{1,1}p_{2,2}}$.

A.5 Notion of homophily indices

Remark A.4. Earlier work measured homophily in undirected graphs $\mathcal{G}$ without self-loops by using the node homophily index defined as in [27]

$$h_V := \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \frac{|\{w \in \mathcal{V} \mid \substack{(v,w)\in\mathcal{E} \\ l(w)=l(v)}\}|}{\deg(v)}, \tag{15}$$

or by using the edge homophily index defined as in [50]:

$$h_E := \frac{|\{(u,v) \in \mathcal{E} \mid l(u) = l(v)\}|}{|\mathcal{E}|}. \tag{16}$$

A common way to distinguish homophilic from heterophilic graphs is to compute the homophily indices h_V and h_E and check whether their values are closer to 0 or 1. We demonstrate that the graph signature matrix allows one to closely analyze the relations among different node labels, a trait that numerical homophily indices do not necessarily share.

A.6 Proof of Theorem 2.3

Setup and Notation. Consider an undirected graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ with $|L| \geq 2$ class labels, where class i contains $n_i \geq 1$ nodes. Let $N = \sum_{i=1}^{|L|} n_i$ denote the total number of nodes. The graph has $|\mathcal{E}| > 0$ edges with edge homophily level $h_E \in [0, 1]$, partitioned into $h_E|\mathcal{E}|$ intra-class edges and $(1 - h_E)|\mathcal{E}|$ inter-class edges.

We adopt a *proportional allocation model*:

- Intra-class edges are distributed among classes proportionally to n_i^2 (the denominator of $s_{i,i}$ in Definition 2.1).
- Inter-class edges are distributed among class pairs (i, j) with $i \neq j$ proportionally to $n_i n_j$.

Define the following quantities for notational convenience:

$$\Sigma_2 := \sum_{k=1}^{|L|} n_k^2, \qquad \Sigma_\times := \sum_{i<j} n_i n_j = \frac{N^2 - \Sigma_2}{2}. \tag{17}$$

Expected Edge Counts. Under the proportional allocation model, the expected number of intra-class edges in class i is

$$\mathbb{E}[X_i] = h_E|\mathcal{E}| \cdot \frac{n_i^2}{\Sigma_2}, \tag{18}$$

and the expected number of inter-class edges between classes i and j (where $i \neq j$) is

$$\mathbb{E}[Y_{ij}] = (1 - h_E)|\mathcal{E}| \cdot \frac{n_i n_j}{\Sigma_\times} = \frac{2(1 - h_E)|\mathcal{E}| \cdot n_i n_j}{N^2 - \Sigma_2}. \tag{19}$$

Expected Signature Matrix Entries. The expected signature matrix $S_\mathcal{G} = \mathbb{E}[\widehat{S}_G]$ has entries given by Definition 2.1. For diagonal entries:

$$(S_\mathcal{G})_{ii} = \frac{2\mathbb{E}[X_i]}{n_i^2} = \frac{2h_E|\mathcal{E}| \cdot n_i^2/\Sigma_2}{n_i^2} = \frac{2h_E|\mathcal{E}|}{\Sigma_2} =: a. \tag{20}$$

For off-diagonal entries:

$$(S_\mathcal{G})_{ij} = \frac{\mathbb{E}[Y_{ij}]}{n_i n_j} = \frac{2(1 - h_E)|\mathcal{E}| \cdot n_i n_j/(N^2 - \Sigma_2)}{n_i n_j} = \frac{2(1 - h_E)|\mathcal{E}|}{N^2 - \Sigma_2} =: b. \tag{21}$$

Both diagonal and off-diagonal entries are constant across all indices, yielding the matrix structure:

$$S_\mathcal{G} = a \cdot I_{|L|} + b \cdot (J_{|L|} - I_{|L|}) = (a - b)I_{|L|} + bJ_{|L|}, \tag{22}$$

where $I_{|L|}$ is the $|L| \times |L|$ identity matrix and $J_{|L|}$ is the $|L| \times |L|$ all-ones matrix.

Eigenvalue Computation. The all-ones matrix $J_{|L|}$ has eigenvalue $|L|$ with eigenvector $\mathbf{1}_{|L|} = (1, \ldots, 1)^\top$, and eigenvalue 0 with multiplicity $|L| - 1$ for eigenvectors orthogonal to $\mathbf{1}_{|L|}$. Therefore, the first eigenvalue (with eigenvector $\mathbf{1}_{|L|}$) is

$$\lambda_1 = (a - b) + |L|b = a + (|L| - 1)b, \tag{23}$$

and the remaining eigenvalues (with eigenvectors in $\mathbf{1}_{|L|}^\perp$, multiplicity $|L| - 1$) are

$$\lambda_2 = \cdots = \lambda_{|L|} = a - b. \tag{24}$$

Substituting the definitions of a and b:

$$\begin{aligned} a - b &= \frac{2h_E|\mathcal{E}|}{\Sigma_2} - \frac{2(1 - h_E)|\mathcal{E}|}{N^2 - \Sigma_2} = 2|\mathcal{E}| \left(\frac{h_E}{\Sigma_2} - \frac{1 - h_E}{N^2 - \Sigma_2} \right) \\ &= 2|\mathcal{E}| \cdot \frac{h_E(N^2 - \Sigma_2) - (1 - h_E)\Sigma_2}{\Sigma_2(N^2 - \Sigma_2)}. \end{aligned} \tag{25}$$

Expanding the numerator:

$$h_E(N^2 - \Sigma_2) - (1 - h_E)\Sigma_2 = h_E N^2 - h_E\Sigma_2 - \Sigma_2 + h_E\Sigma_2 = h_E N^2 - \Sigma_2. \tag{26}$$

Therefore:

$$\lambda_2 = \cdots = \lambda_{|L|} = \frac{2|\mathcal{E}|(h_E N^2 - \Sigma_2)}{\Sigma_2(N^2 - \Sigma_2)}. \tag{27}$$

Critical Threshold. The sign of $\lambda_2, \ldots, \lambda_{|L|}$ is determined by the numerator $(h_E N^2 - \Sigma_2)$:

$$h_E N^2 - \Sigma_2 \gtreqless 0 \quad \Longleftrightarrow \quad h_E \gtreqless \frac{\Sigma_2}{N^2} = \frac{\sum_{k=1}^{|L|} n_k^2}{N^2}. \tag{28}$$

Thus, the critical threshold is:

$$h_E^* = \frac{\sum_{k=1}^{|L|} n_k^2}{N^2} = \sum_{k=1}^{|L|} \left(\frac{n_k}{N} \right)^2. \tag{29}$$

Positivity of λ_1. We verify that $\lambda_1 > 0$ for all valid parameter ranges:

$$\lambda_1 = a + (|L| - 1)b = \frac{2h_E|\mathcal{E}|}{\Sigma_2} + \frac{2(|L| - 1)(1 - h_E)|\mathcal{E}|}{N^2 - \Sigma_2}. \tag{30}$$

Since $h_E \in [0, 1]$, $|\mathcal{E}| > 0$, $\Sigma_2 > 0$, and $N^2 - \Sigma_2 = 2\sum_{i<j} n_i n_j > 0$ for $|L| \geq 2$, we have $\lambda_1 > 0$ always.

Signature Classification. Combining the above results:

- If $h_E > h_E^*$: $\lambda_1 > 0$ and $\lambda_2 = \cdots = \lambda_{|L|} > 0$, so $s_{\mathcal{G}} = (|L|, 0, 0)$.
- If $h_E = h_E^*$: $\lambda_1 > 0$ and $\lambda_2 = \cdots = \lambda_{|L|} = 0$, so $s_{\mathcal{G}} = (1, |L| - 1, 0)$.
- If $h_E < h_E^*$: $\lambda_1 > 0$ and $\lambda_2 = \cdots = \lambda_{|L|} < 0$, so $s_{\mathcal{G}} = (1, 0, |L| - 1)$.

Reduction to Balanced Case. For balanced classes where $n_i = n$ for all $i \in \{1, \ldots, |L|\}$:

$$h_E^* = \frac{\sum_{k=1}^{|L|} n^2}{(|L|n)^2} = \frac{|L|n^2}{|L|^2 n^2} = \frac{1}{|L|}. \tag{31}$$

Probabilistic Interpretation. The critical threshold $h_E^* = \sum_{k=1}^{|L|} (n_k/N)^2$ equals the probability that two nodes chosen uniformly at random (with replacement) belong to the same class. This quantity provides a natural baseline:

- $h_E > h_E^*$: Edges preferentially connect intra-class nodes relative to random chance (*homophilic*).
- $h_E < h_E^*$: Edges preferentially connect inter-class nodes relative to random chance (*heterophilic*).

□

A.7 Proof of Theorem 2.4

Both homophily indices of a finite graph are rational: h_E is a ratio of edge counts and h_V is an average of ratios of integers. The construction below attains every rational number in $(0, 1)$, so Theorem 2.4 covers every value that h_E or h_V takes in $(0, 1)$. We prove it in the following stronger form, which also prescribes the number of classes.

Theorem A.5. *For every rational $h \in (0, 1)$ and every number of classes $|L| \geq 2$, there exists a graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ with node labels $l : \mathcal{V} \to L$ such that $h_E = h_V = h$ and $S_{\mathcal{G}}$ has a negative eigenvalue.*

Proof. We first record a sufficient condition that involves neither h_E nor h_V. Suppose that some class j has no intra-class edge and at least one edge to a class $i \neq j$, so that $s_{j,j} = 0$ and $s_{i,j} = s_{j,i} > 0$. For the standard basis vectors $e_i, e_j \in \mathbb{R}^{|L|}$ and $x = te_i - e_j$ with $t > 0$,

$$x^T S_{\mathcal{G}} x = t^2 s_{i,i} - 2t s_{i,j} + s_{j,j} = t\left(t s_{i,i} - 2 s_{i,j}\right), \tag{32}$$

which is negative for every sufficiently small $t > 0$. The smallest eigenvalue of $S_{\mathcal{G}}$ is at most the Rayleigh quotient $x^T S_{\mathcal{G}} x / x^T x$, hence negative.

Write $h = p/q$ with integers $1 \leq p < q$, and let $k = |L|$ and $m = k - 1$. Let $\mathcal{G}$ be the disjoint union of mq edges: mp of them join two nodes of class 1, and each of the other $m(q - p) \geq k - 1$ edges joins a node of class 1 to a node of a class $j \in \{2, \ldots, k\}$, with every such j used at least once. Every class is nonempty, so $S_{\mathcal{G}}$ is well defined, and $h_E = mp/(mq) = h$. Every node has degree 1: each of the $2mp$ endpoints of an intra-class edge has its only neighbor in its own class, and each of the other $2m(q - p)$ nodes has its only neighbor in another class, so $h_V = 2mp/(2mq) = h$. Class 2 has no intra-class edge and an edge to class 1, so $S_{\mathcal{G}}$ has a negative eigenvalue by the sufficient condition.

The signature is explicit. Class 1 has $n_1 = m(q + p)$ nodes and $mp \geq 1$ intra-class edges, so $s_{1,1} > 0$. Each node of a class $j \geq 2$ has its single edge to class 1, so $s_{j,1} = 1/n_1$ and $s_{j,j'} = 0$ for all $j, j' \geq 2$. Rows $2, \ldots, k$ of $S_{\mathcal{G}}$ therefore all equal $(1/n_1, 0, \ldots, 0)$, while row 1 has the entry $1/n_1$ in columns $2, \ldots, k$, so $S_{\mathcal{G}}$ has rank 2. One of its two nonzero eigenvalues is negative and their sum is $\operatorname{tr} S_{\mathcal{G}} = s_{1,1} > 0$, so $s_{\mathcal{G}} = (1, |L| - 2, 1)$. For $|L| = 2$,

$$S_{\mathcal{G}} = \begin{pmatrix} s_{1,1} & \frac{1}{q+p} \\ \frac{1}{q+p} & 0 \end{pmatrix}, \qquad s_{1,1} > 0, \qquad \det S_{\mathcal{G}} = -\frac{1}{(q+p)^2} < 0. \tag{33}$$

□

A.8 Proof and interpretations of Theorem 2.5

For a smooth map $f : \mathcal{M} \to \mathcal{L}$ between differentiable manifolds $\mathcal{M}$ and $\mathcal{L}$, the signature of the metric g over $\mathcal{L}$ and that of the pullback f^*g over $\mathcal{M}$ satisfy the following conditions:

$$\begin{aligned}
\text{Number of positive eigenvalues of } (f^*g)_p &\leq \text{Number of positive eigenvalues of } g_{f(p)} \\
\text{Number of negative eigenvalues of } (f^*g)_p &\leq \text{Number of negative eigenvalues of } g_{f(p)} \\
\text{Number of zero eigenvalues of } (f^*g)_p &= \dim_{\mathbb{R}} \text{Ker}(T_p f) + \dim_{\mathbb{R}} \left(\text{Im}(T_p f) \cap \text{Im}(T_p f)^{\perp_g}\right)
\end{aligned} \tag{34}$$

The second term in the last line is needed whenever g is indefinite. A linear map can be injective and still pull an indefinite form back to a degenerate one: for $A = (1, 1)^\top : \mathbb{R} \to \mathbb{R}^{1,1}$ with $g = \text{diag}(1, -1)$ we have $\ker A = 0$ yet $A^\top g A = 0$, so the nullity is 1. The image of A is a null line, contained in its own g-orthogonal complement, which is exactly what the second term counts. When g is positive definite that intersection is trivial and the identity reduces to the familiar $\dim \ker$. The two inequalities on the positive and negative counts hold regardless.

By Sylvester's law of inertia (Theorem 2.2), $\phi^* I$ is positive-definite. Hence, for any pushforward $T_p f : T_p\mathcal{M} \to T_{f(p)}\mathcal{L}$, the eigenvalues of the pullback $f^*(\phi^* I)$ satisfy:

$$\text{Number of negative eigenvalues of } (f^*(\phi^* I))_p = 0 \tag{35}$$

On the other hand, if $T_p f$ is surjective then $\text{Im}(T_p f) = T_{f(p)}\mathcal{L}$, so a negative direction of $S_\mathcal{G}$ has a preimage and the eigenvalues of the pullback $f^* S_\mathcal{G}$ satisfy:

$$\text{Number of negative eigenvalues of } (f^* S_\mathcal{G})_p > 0 \tag{36}$$

Surjectivity is what this step needs, and it is strictly stronger than full rank when $\dim \mathcal{M} < \dim \mathcal{L}$: an injective map into a higher-dimensional indefinite space can miss every negative direction, as the example above shows. A linear map is surjective exactly when its rank equals $\dim \mathcal{L}$, which requires $\dim \mathcal{M} \geq \dim \mathcal{L}$ and is not implied by it. In our setting $T_p f$ is the projection $\mathbf{W} \in \mathbb{R}^{|L| \times h'}$ of Equation (4), and every configuration we report has $h' > |L|$. The matrices of rank below $|L|$ are the common zero set of the $|L| \times |L|$ minors, a proper algebraic subset of $\mathbb{R}^{|L| \times h'}$ of measure zero, so the hypothesis holds for almost every $\mathbf{W}$, and almost surely at the random initialization. Therefore, there does not exist a diffeomorphism $\phi : \mathcal{L} \to \mathcal{L}$ which induces $(\phi \circ f)^* I = f^* S_\mathcal{G}$. This concludes the proof of Theorem 2.5.

We can also deduce the following three observations from Theorem 2.5.

- Discretization of graph diffusion processes [5], which inherently assumes that the underlying topological space with graphs as subspaces is a Riemannian manifold, may have limited capabilities in analyzing relations among nodes of graphs whose $S_\mathcal{G}$ has a negative eigenvalue.
- Continuous node attributes of graphs may not be beneficial in classifying node labels of heterophilic graph data sets, implying that utilizing only the connectivity structure between node labels could be advantageous. Empirical evidence also supports this observation, as shown in [50].
- Using adjacency matrices other than positive-definite symmetric matrices, such as graph laplacians, may boost results in classifying node labels of heterophilic graph data sets. Examples include using adjacency matrices multiplied with cosine similarities of node labels [44] or using sheaf graph laplacian operator, a type of block matrix which is not necessarily positive-definite (but positive-definite as a block matrix) [2].

A.9 Topological structure on directed finite graphs in preparation for Definition 3.2

The parallel transport $\Gamma_{\gamma_{e_1,e_2}(t_0,s)}$ for each $t = t_0$ is taken with respect to the flat connection of $\mathbb{R}^h$, so it translates vectors, preserving both their directions and lengths. Because the metric $g = S$ is constant, this connection is compatible with g even when S is degenerate, and it is the Levi-Civita connection of g when S is non-degenerate. Since Γ acts as the identity on $\mathbb{R}^h$ and S is symmetric, $e_{\gamma_{e_1,e_2}}(\gamma_{e_1}, \gamma_{e_2}) = e_{\gamma_{e_2,e_1}}(\gamma_{e_2}, \gamma_{e_1})$, which proves the commutative property of tangent attention scores.

This subsection describes suitable topological structures to impose on directed finite graphs so that we can rigorously define the tangent attention score matrix of a graph. We treat graphs with h-dimensional node features as subspaces of an h-dimensional connected pseudo-Riemannian manifold. Without loss of generality, let $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ be a finite directed graph with a node feature function $X : \mathcal{V} \to \mathbb{R}^h$. We may view undirected graphs as directed graphs by regarding each undirected edge between the node v and w as two reciprocal directed edges between them. Any graph $\mathcal{G}$ can be viewed as a 1-dimensional topological space with nodes as discrete points and edges as unit intervals $[0, 1]$ [26, 8]; see Chapter 0 of [13] for further discussion. This presupposition allows us to extend a function of node features $X : \mathcal{V} \to \mathbb{R}^h$ to a continuous function $X_\gamma : \mathcal{G} \to \mathcal{M}$ for some h-dimensional pseudo-Riemannian manifold $(\mathcal{M}, g)$, where each directed edge e from the source node v to the target node w, denoted as $e := (v, w)$, is represented by a smooth curve of our choice $\gamma_e : [0, 1] \to \mathcal{M}$ satisfying the constraint that $\gamma_e(0) = X(v)$ and $\gamma_e(1) = X(w)$. Figure 1a illustrates how one may regard a directed graph $\mathcal{G}$ as a subspace of $\mathcal{M}$.

A.10 Detailed description and proof of Theorem 3.3

This subsection shows, with proof, which choice of extension X_γ allows us to reconstruct a wide class of attention functions, including the dot-product attention from [38], the single-layer feedforward linear layer from GAT [39], and the dynamic graph attention from GATv2 [3]. Provided below is a detailed description of Theorem 3.3.

Theorem A.6. *Let $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ be a finite directed complete graph with self-loops, along with an h-dimensional node feature function $X : \mathcal{V} \to \mathbb{R}^h$. Fix a symmetric matrix $S \in \mathbb{R}^{h\times h}$ which endows $\mathbb{R}^h$ a pseudo-metric structure.*

Suppose that for each edge $e := (v, w) \in \mathcal{E}$, the curves $\gamma_e^{[i]} : [0, 1] \to \mathcal{M} \cong \mathbb{R}^h$ are given by:

$$\gamma_e^{[1]}(t) := \begin{cases} (1-t)\cdot X(v) + t\cdot X(w) & \text{if } v \neq w \\ \sqrt{3}\left(a_v t(t-1) + \frac{1}{\sqrt{3}}\right)\cdot X(v) & \text{if } v = w \end{cases} \tag{37}$$

$$\gamma_e^{[2]}(t) := \begin{cases} (1-t^2)\cdot X(v) + t^3\cdot X(w) & \text{if } v \neq w \\ X(v) + 6t(t-1)(5t-3)\cdot a_{v,1} + 10t(t-1)(2t-1)\cdot a_{v,2} & \text{if } v = w \end{cases} \tag{38}$$

$$\gamma_e^{[3]}(t) := \begin{cases} (1-t)\cdot X(v) + t\cdot X(w) + t(t-1)\cdot a_e & \text{if } v \neq w \\ X(v) + 3t(t-1)\cdot a_{v,3} & \text{if } v = w \end{cases} \tag{39}$$

for some fixed choices of set of scalars $\{a_v\}_{v\in\mathcal{V}}$ and h-dimensional real vectors $\{a_e\}_{e\in\mathcal{E}}, \{a_{v,i}\}_{v\in\mathcal{V}} \subset \mathbb{R}^h$. Then given a pair of edges $e_1 := (v_1, w_1), e_2 := (v_2, w_2)$, the (e_1, e_2)-th entries of the tangent attention score matrices $e_{\mathcal{G},\gamma^{[i]}}$ are equal to:

$$\left(e_{\mathcal{G},\gamma^{[1]}}\right)_{e_1,e_2} := \begin{cases} (X(w_1) - X(v_1))^T \cdot S \cdot (X(w_2) - X(v_2)) & \text{if } v_1 \neq w_1,\ v_2 \neq w_2 \\ 0 & \text{if } v_1 \neq w_1,\ v_2 = w_2 \\ 0 & \text{if } v_1 = w_1,\ v_2 \neq w_2 \\ a_{v_1} a_{v_2} X(v_1)^T \cdot S \cdot X(v_2) & \text{if } v_1 = w_1,\ v_2 = w_2 \end{cases} \tag{40}$$

$$\left(e_{\mathcal{G},\gamma^{[2]}}\right)_{e_1,e_2} := \begin{cases} \left(\frac{4}{3}X(v_1) - \frac{3}{2}X(w_1)\right)^T \cdot S \cdot X(v_2) & \\ \qquad - \left(\frac{3}{2}X(v_1) - \frac{9}{5}X(w_1)\right)^T \cdot S \cdot X(w_2) & \text{if } v_1 \neq w_1,\ v_2 \neq w_2 \\ X(v_1)^T \cdot S \cdot a_{v_2,1} + X(w_1)^T \cdot S \cdot a_{v_2,2} & \text{if } v_1 \neq w_1,\ v_2 = w_2 \\ a_{v_1,1}^T \cdot S \cdot X(v_2) + a_{v_1,2}^T \cdot S \cdot X(w_2) & \text{if } v_1 = w_1,\ v_2 \neq w_2 \\ 48 a_{v_1,1}^T \cdot S \cdot a_{v_2,1} + 30 a_{v_1,1}^T \cdot S \cdot a_{v_2,2} & \\ \qquad + 30 a_{v_1,2}^T \cdot S \cdot a_{v_2,1} + 20 a_{v_1,2}^T \cdot S \cdot a_{v_2,2} & \text{if } v_1 = w_1,\ v_2 = w_2 \end{cases} \tag{41}$$

$$\left(e_{\mathcal{G},\gamma^{[3]}}\right)_{e_1,e_2} := \begin{cases} (X(w_1) - X(v_1))^T \cdot S \cdot (X(w_2) - X(v_2)) & \\ \qquad + \frac{1}{3} a_{e_1}^T \cdot S \cdot a_{e_2} & \text{if } v_1 \neq w_1,\ v_2 \neq w_2 \\ a_{e_1}^T \cdot S \cdot a_{v_2,3} & \text{if } v_1 \neq w_1,\ v_2 = w_2 \\ a_{v_1,3}^T \cdot S \cdot a_{e_2} & \text{if } v_1 = w_1,\ v_2 \neq w_2 \\ 3 a_{v_1,3}^T \cdot S \cdot a_{v_2,3} & \text{if } v_1 = w_1,\ v_2 = w_2 \end{cases} \tag{42}$$

Proof. We prove the statement of the theorem for the curve $\gamma_e^{[1]} : [0,1] \to \mathcal{M}$. The remaining cases follow analogously. Let $e_1 = (v_1, w_1), e_2 = (v_2, w_2)$ be a pair of directed edges of a finite directed complete graph $\mathcal{G}$.

Suppose $v_1 \neq w_1$ and $v_2 \neq w_2$. Then we obtain that

$$\begin{aligned} \gamma_{e_1}^{'[1]}(t) &= X(w_1) - X(v_1) \\ \gamma_{e_2}^{'[1]}(t) &= X(w_2) - X(v_2) \end{aligned} \tag{43}$$

Because both derivatives are constant functions with respect to the variable t,

$$e_{\gamma_{e_1,e_2}}(\gamma_{e_1}^{[1]}, \gamma_{e_2}^{[1]}) = (X(w_1) - X(v_1))^T \cdot S \cdot (X(w_2) - X(v_2)) \,. \tag{44}$$

If $v_1 = w_1$ whereas $v_2 \neq w_2$, then

$$\begin{aligned} \gamma_{e_1}^{'[1]}(t) &= \sqrt{3} a_{v_1} (2t-1) \cdot X(v_1) \\ \gamma_{e_2}^{'[1]}(t) &= X(w_2) - X(v_2) \end{aligned} \tag{45}$$

which leads to

$$e_{\gamma_{e_1,e_2}}(\gamma_{e_1}^{[1]}, \gamma_{e_2}^{[1]}) = \left(\int_0^1 \sqrt{3} a_{v_1} (2t-1) dt \right) \cdot \left(X(v_1)^T \cdot S \cdot (X(w_2) - X(v_2)) \right) = 0. \tag{46}$$

Lastly, if $v_1 = w_1$ and $v_2 = w_2$, then

$$\begin{aligned} \gamma_{e_1}^{'[1]}(t) &= \sqrt{3} a_{v_1} (2t-1) \cdot X(v_1) \\ \gamma_{e_2}^{'[1]}(t) &= \sqrt{3} a_{v_2} (2t-1) \cdot X(v_2) \end{aligned} \tag{47}$$

which gives

$$\begin{aligned} e_{\gamma_{e_1,e_2}}(\gamma_{e_1}^{[1]}, \gamma_{e_2}^{[1]}) &= \left(\int_0^1 3 a_{v_1} a_{v_2} (2t-1)^2 dt \right) \cdot \left(X(v_1)^T \cdot S \cdot X(v_2) \right) \\ &= a_{v_1} a_{v_2} X(v_1)^T \cdot S \cdot X(v_2). \end{aligned} \tag{48}$$

□

Theorem 3.3 demonstrates that previously researched attention scores correspond to computing the tangent attention score between two directed edges where at least one of them is a self-loop at each node of $\mathcal{G}$. The dot product attention score from the encoder layer of attention-transformers [38] with identical query and key weight matrices is equal to the tangent attention score between two self-loops represented by $\gamma_e^{[1]}$, with $S = W^T W$ a symmetric matrix obtained from a product of invertible square weight matrix $W \in \mathbb{R}^{h \times h}$. With the same symmetric matrix S, the attention score of the single-layer feedforward linear layer in [39] can be obtained from the tangent attention score between a self-loop and a directed edge represented by $\gamma_e^{[2]}$. Lastly, by setting S to be the identity matrix I, and by choosing the fixed vector a_e as $a_e = \text{LeakyRelu}\left(W[X(v)||X(w)]\right)$ for each edge $e = (v, w) \in \mathcal{E}$ and every $a_{v,3}$ as the shared attention vector $\mathbf{a} \in \mathbb{R}^h$ of GATv2, where $[X(v)||X(w)] \in \mathbb{R}^{2h}$ is a concatenation of two node feature vectors, and $W \in \mathbb{R}^{h \times 2h}$ is a weight matrix, the tangent attention score between a self-loop and a directed edge represented by $\gamma_e^{[3]}$ is equal to the dynamic graph attention score constructed from [3].

B Geometric Framework of PRAT

In Figure 7, $\mathcal{H}^{(l)} \cong \mathbb{R}^{h'}$ denotes the hidden manifold at layer l, and $\mathcal{L} \cong \mathbb{R}^{|L|}$ is the common latent manifold corresponding to the node labels. The encoder $\psi_{\text{enc}} : \mathcal{M} \to \mathcal{H}^{(0)}$ calibrates node features to the initial hidden feature.

For each layer l, we posit the existence of a linear map $\varphi^{(l)} : \mathcal{H}^{(l)} \to \mathcal{L}$, which projects the hidden representation to the label space. While the final layer's map $\varphi^{(l_{\text{max}})}$ acts as the explicit readout function for classification, the intermediate maps $\varphi^{(l)}$ (for $l < l_{\text{max}}$) provide a critical geometric link:

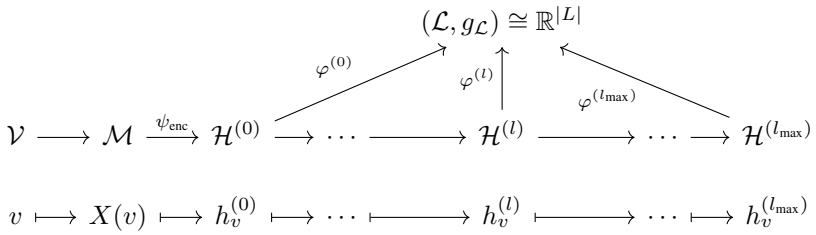


Figure 7: Geometric interpretation of PRAT. Each hidden manifold $\mathcal{H}^{(l)}$ is mapped to the common latent manifold $\mathcal{L}$ via $\varphi^{(l)}$.

they allow us to transport the intrinsic geometric structure of the latent manifold back to the hidden manifolds.

Let $g_{\mathcal{L}}$ be the pseudo-metric tensor defined on $\mathcal{L}$, represented by a symmetric matrix $S^{|L|}$. To ensure that the hidden representations at layer l respect the global structural prior of the latent space, we set the metric on $\mathcal{H}^{(l)}$ to the *pullback metric* induced by $\varphi^{(l)}$. Hence, the pseudo-metric on $\mathcal{H}^{(l)}$ is determined by pulling back $g_{\mathcal{L}}$ via $\varphi^{(l)}$:

$$g_{\mathcal{H}^{(l)}} = (\varphi^{(l)})^* g_{\mathcal{L}}. \tag{49}$$

Considering the multi-head attention mechanism where k denotes the head index, the projection $\varphi^{(l)}$ is parameterized by a weight matrix $\mathbf{W}_{\varphi}^{[l,k]}$. The induced metric tensor $\mathbf{D}^{[l,k]}$ for the k-th head at layer l is then derived as:

$$\mathbf{D}^{[l,k]} = (\mathbf{W}_{\varphi}^{[l,k]})^\top S^{|L|} \mathbf{W}_{\varphi}^{[l,k]}, \tag{50}$$

which is identical to the expression for $\mathbf{D}^{[l,k]}$ in Equation (4).

C Benchmark Datasets

We use fifteen real-world benchmarks in two groups: (i) eight *standard* benchmarks spanning homophilic and heterophilic regimes, and (ii) seven *Platonov benchmarks* [28] consisting of filtered Wikipedia networks and five large-scale heterophilic datasets.

Standard benchmarks.

Citation Networks (Cora, Citeseer, PubMed) Standard benchmark datasets representing citation relationships between scientific publications [32, 45]. Nodes correspond to documents, and edges denote citation links. For each node, the features are typically sparse bag-of-words vectors derived from the document text, and the label indicates the subject category of the paper.

Wikipedia Networks (Chameleon, Squirrel) These are page-page networks derived from Wikipedia [30]. Nodes represent Wikipedia articles, and edges indicate mutual hyperlinks between them. Node features are informative nouns extracted from the corresponding article, and the node label represents the average monthly traffic of the webpage.

Actor Co-occurrence Network (Actor) The Actor dataset (often referred to as Film) [35] is a subgraph of a film-director-actor-writer network. Nodes represent actors, and edges signify co-occurrence on the same Wikipedia page. Node features are derived from keywords on the actor's Wikipedia page, and node labels are determined by the words found on the webpage.

Webpage Networks (Texas, Cornell) Subsets of the WebKB dataset[6]. Nodes represent university webpages, and edges represent hyperlinks between them. For each node, the features are bag-of-words representations of the webpage content, and the node label denotes the category of the webpage.

[6] `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb`

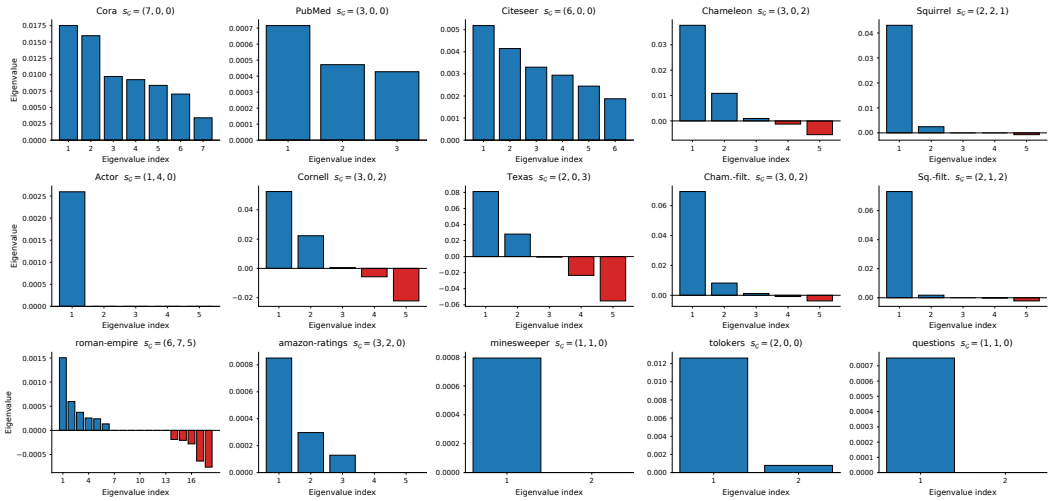


Figure 8: Eigenvalue distributions of the graph signature matrix of the full labeled graph (Definition 2.1) for all fifteen benchmark datasets (eight standard + seven Platonov), which give the signatures of Table 1 and Table 2. Eigenvalues with $|\lambda| < 1.0 \times 10^{-4}$ are rounded to zero for reporting.

Platonov benchmarks [28].

Filtered Wikipedia networks (Chameleon-filtered, Squirrel-filtered) Platonov et al. [28] identified that 57–61% of nodes in the original Chameleon and Squirrel datasets are exact duplicates with identical neighborhoods and labels, causing severe train–test leakage. The *filtered* versions remove these duplicate nodes, yielding 890 and 2,223 nodes respectively. Node features and class definitions follow the original Rozemberczki et al. [30] construction. These filtered benchmarks are widely adopted in subsequent heterophily research as a more reliable proxy for the underlying task.

roman-empire Each node corresponds to a non-unique word in the English Wikipedia article "Roman Empire," and edges connect words that are syntactically dependent or sequential in the article. Node features are GloVe-300 embeddings of the word, and the 18-class label is the syntactic role assigned by spaCy. With $h_E = 0.05$, this is the most extreme heterophilic dataset in the suite.

amazon-ratings A product co-purchase network from Amazon, where nodes are products (books, music CDs, DVDs, VHS tapes) and edges connect frequently co-purchased products. Node features are the mean of fastText embeddings of the product description, and the 5-class label is the average rating, binned into five ranges.

minesweeper A synthetic 100×100 grid graph from the Minesweeper game: nodes are cells, edges connect each cell to its eight surrounding neighbors, and 20% of cells are randomly designated as mines (the binary class). Node features are 7-dimensional one-hot encodings of the number of neighboring mines for non-mine cells (with mine cells masked); the resulting graph has very low-dimensional node features and a regular topological structure.

tolokers A worker-collaboration network from the Toloka crowdsourcing platform, where nodes are workers and edges connect workers who co-completed at least one task. Node features include task performance and profile statistics, and the binary label indicates whether a worker was banned. With over 519K edges over 11.8K nodes, tolokers is the densest graph in the suite.

questions A user-interaction network on the Yandex.Q question-answering site, where nodes are users and edges connect users who answered each other's questions during a one-year window. Node features are FastText embeddings of the user description, and the binary label indicates account activity at the end of the window. With 48,921 nodes, this is the largest graph in the suite.

Analysis on Graph Signature Index. Figure 8 illustrates the eigenvalue distributions of the graph signature matrices for all benchmark datasets, revealing distinct structural characteristics. Among the standard benchmarks, the **Actor** dataset stands out for its many zero eigenvalues. Its signature $s_{\mathcal{G}} = (1, 4, 0)$ indicates connectivity of numerical rank one at the 10^{-4} threshold, where all 5 classes share nearly identical edge density patterns, with 4 out of 5 label-space dimensions carrying negligible discriminative signal. PRAT leverages this by learning Λ values near zero for the corresponding dimensions during training, effectively projecting out uninformative directions and concentrating the metric on the single informative one. This "feature selection via metric learning" explains why PRAT achieves the best accuracy on Actor despite the absence of negative eigenvalues: the pseudo-Riemannian framework naturally accommodates zero eigenvalues, providing a mechanism to suppress noise that other methods aggregate indiscriminately.

D Synthetic Data Generation

Synthetic datasets are generated based on Model 1 proposed in Appendix B of [15]. According to this model, each node feature is initially sampled from a Bernoulli distribution, yielding binary values of 0 or 1. However, we observed that this formulation results in class labels being trivially predictable by GNNs, leading to near-perfect accuracy even in highly heterophilic regimes.

To address this limitation and make classification harder, we modify the original model's features. We add class-independent Gaussian noise of standard deviation 0.4 to each node feature, thereby transforming the discrete binary features into continuous values and increasing the difficulty of class discrimination. The added noise prevents perfect predictions while preserving the synthetic dataset's underlying structure.

For graph construction, given a target number of edges, a desired homophily level $h_E \in [0, 1]$, and a target graph signature index, we first compute the number of homophilic edges as $|\mathcal{E}_{homo}| = h_E \cdot |\mathcal{E}|$ and heterophilic edges as $|\mathcal{E}_{hetero}| = |\mathcal{E}| - |\mathcal{E}_{homo}|$. To ensure the resulting graph matches the specified signature index, these edge counts are strategically distributed across valid intra-class and inter-class pairs by a random search over at most $2{,}000$ symmetric candidates for a class-connectivity matrix that satisfies both the homophily and signature constraints. Homophilic edges are then generated by randomly sampling node pairs within the same class, while heterophilic edges are created by connecting nodes from different classes according to the determined connectivity matrix. Certain combinations of homophily levels and signature indices are mathematically incompatible; in such cases, synthetic data generation is restricted. This procedure precisely controls both the graph's homophily ratio and signature index, enabling systematic evaluation of model performance across diverse structural scenarios. Every graph is generated with seed 42, the five random splits use seeds 42 to 46, and every model is trained with seed 42.

Eigenvalue Evolution. Figure 9 illustrates the evolution of eigenvalues in the graph signature matrix across synthetic datasets generated with varying homophily levels. At the heterophilic extreme ($h_E = 0$), the graph signature index is $(1, 0, 4)$, indicating a predominance of negative eigenvalues. As h_E increases, these eigenvalues exhibit a continuous upward trend, eventually transitioning to a strictly positive regime. This trajectory closely mirrors the sign evolution of the learned metric observed in Figure 3. These findings provide empirical evidence that PRAT successfully leverages the underlying topological information regarding graph homophily.

E Computational Complexity Analysis

We analyze the computational complexity of PRAT compared to other attention-based models. Let N denote nodes, E edges, H hidden dimension, C classes, K attention heads, and L layers. H/K is the per-head dimension.

E.1 Time Complexity

Theorem E.1 (PRAT Layer Complexity)**.** *The time complexity of a single PRAT layer is* $\mathcal{O}(NH(H + KC) + E(H + KC))$.

Proof. PRAT has operations:

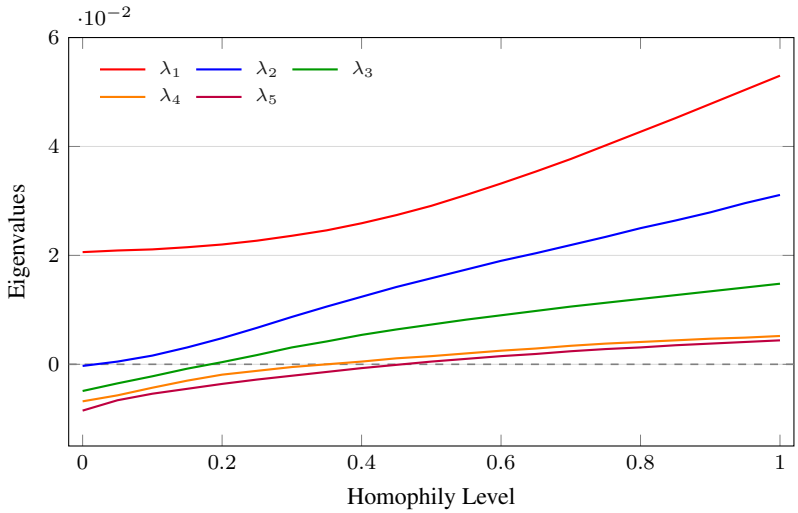


Figure 9: Evolution of eigenvalues of graph signature matrix across varying homophily levels. As homophily increases, eigenvalues change from mixed signs to positive values, indicating a shift from pseudo-Riemannian to Riemannian geometry.

1. Latent space projection ($\mathbf{XW}$, $\mathbf{W} \in \mathbb{R}^{H \times KC}$): $\mathcal{O}(NHKC)$
2. Value projection ($\mathbf{XW}_V$, $\mathbf{W}_V \in \mathbb{R}^{H \times H}$): $\mathcal{O}(NH^2)$
3. Attention scores (difference, square, weighted sum per edge): $\mathcal{O}(EKC)$
4. Softmax and aggregation: $\mathcal{O}(EK + EH)$

□

Theorem E.2 (GAT/GATv2 Layer Complexity)**.** *The time complexity of GAT and GATv2 layers is* $\mathcal{O}(NH^2 + EH)$.

Proof. GAT computes $e_{ij} = \text{LeakyReLU}(\mathbf{a}^\top [\mathbf{Wh}_i \| \mathbf{Wh}_j])$:

1. Feature transform ($\mathbf{XW}$): $\mathcal{O}(NH^2)$
2. Attention computation (concatenation + linear per edge): $\mathcal{O}(EK(H/K)) = \mathcal{O}(EH)$
3. Softmax and aggregation: $\mathcal{O}(EK + EH)$

GATv2 applies LeakyReLU before the attention vector but has identical complexity.

□

Theorem E.3 (SDP Layer Complexity)**.** *The time complexity of Scaled Dot-Product attention is* $\mathcal{O}(3NH^2 + EH)$.

Proof. SDP computes $e_{ij} = (\mathbf{Q}_i \cdot \mathbf{K}_j)/\sqrt{d_k}$ with three projections:

1. Query, key, and value projections: $\mathcal{O}(3NH^2)$
2. Dot product per edge: $\mathcal{O}(EK(H/K)) = \mathcal{O}(EH)$
3. Softmax and aggregation: $\mathcal{O}(EK + EH)$

□

Table 6 summarizes the time complexity of PRAT and attention-based models. PRAT shares asymptotic complexity $\mathcal{O}(NH^2 + EH)$ with GAT and GATv2 since KC is typically small relative to H (e.g., K=4, C=5, H=64 gives KC=20 $<$ 64).

Table 6: Time complexity comparison per layer.

Model	Complexity
PRAT	$\mathcal{O}(NH(H+KC)+E(H+KC))$
GAT	$\mathcal{O}(NH^2+EH)$
GATv2	$\mathcal{O}(NH^2+EH)$
SDP	$\mathcal{O}(3NH^2+EH)$

E.2 Parameter Count

For a model with encoder, L attention layers, and decoder:

Theorem E.4 (Parameter Counts)**.** *The total parameters for each model are:*

$$P_{PRAT} = (F{+}1)H + L(H^2 + HKC + H + KC + K) + (H{+}1)C \tag{51}$$

$$P_{GAT} = (F{+}1)H + L(H^2 + 3H) + (H{+}1)C \tag{52}$$

$$P_{GATv2} = (F{+}1)H + L(H^2 + 3H) + (H{+}1)C \tag{53}$$

$$P_{SDP} = (F{+}1)H + L(3H^2 + 3H) + (H{+}1)C \tag{54}$$

Proof. Per-layer parameters:

- **PRAT**: $\mathbf{W} \in \mathbb{R}^{H\times KC}$ without bias, $\mathbf{W}_V \in \mathbb{R}^{H\times H}$ with bias, the C diagonal entries of the learnable metric held for each of the K heads, $\Lambda \in \mathbb{R}^{K\times C}$, self-loop scaling parameters $\{a\} \in \mathbb{R}^K$. Total: $HKC + H^2 + H + KC + K$.
- **GAT/GATv2**: $\mathbf{W} \in \mathbb{R}^{H\times H}$ and an output bias, with $\mathbf{a} \in \mathbb{R}^{2H}$ for GAT; GATv2 shares $\mathbf{W}$ between source and target and adds a bias to it, with $\mathbf{a} \in \mathbb{R}^H$. Total: $H^2 + 3H$.
- **SDP**: $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{H\times H}$ each with bias. Total: $3H^2 + 3H$.

Single layer MLP encoder and decoder contribute $(F{+}1)H$ and $(H{+}1)C$, respectively, for all models.

□

Table 7: Parameter count per layer ($H{=}64$, $C{=}5$, $K{=}4$).

Model	Formula	Count
PRAT	$H^2 + HKC + H + KC + K$	5,464
GAT	$H^2 + 3H$	4,288
GATv2	$H^2 + 3H$	4,288
SDP	$3H^2 + 3H$	12,480

Table 7 shows that PRAT has modestly more parameters than GAT and GATv2 because of its learnable metric components. Since $KC < H$ in practice, this overhead remains negligible. Moreover, PRAT is substantially more parameter-efficient than SDP attention, which requires separate query, key, and value projections.

E.3 Measured cost

The asymptotics above say that PRAT and GAT are in the same class. Table 8 shows what that means in practice, measured on the two largest graphs with the same backbone, width, depth and number of steps, so that the only difference is the attention module.

PRAT adds 80 parameters over GAT on questions, out of 2.83 million, and 32 on tolokers. The step time is 0.97 times GAT's on questions and 1.17 on tolokers, and peak memory is 1.03 and 1.20 times GAT's. On questions, PRAT and GAT are indistinguishable at that scale: single-step timing has

Table 8: Training cost per step and peak memory, measured on one A100 80GB with identical backbone, width and depth.

Dataset	Model	ms / step	peak memory (MiB)	parameters
tolokers (11,758 nodes)	PRAT	26.72	1084.4	336,689
	GAT	22.80	903.0	336,657
	GT	32.85	466.1	531,713
questions (48,921 nodes)	PRAT	88.73	3791.2	2,830,201
	GAT	91.23	3696.9	2,830,121
	GT	175.17	5285.5	6,734,081

run-to-run spread of the same order, so their timings do not establish that PRAT is faster. GT, whose attention is also restricted to graph neighbors, costs 1.92 times GAT on questions and needs 2.4 times the parameters.

F Experimental Details

Signature estimation. $S_{\mathcal{G},\text{train}}$ is Definition 2.1 applied to the subgraph induced by the training nodes, with self-loops removed: an edge counts only when both endpoints are training nodes, the class sizes in the denominator are training-node counts, and a class with no training node contributes a zero row and column. Λ_{init} is computed from the unrounded eigenvalues, with $\text{std}(\cdot)$ the population standard deviation over the $|L|$ eigenvalues. When all eigenvalues are equal, so that this deviation is zero, the division is skipped and $\Lambda_{\text{init}} = \Lambda_{\mathcal{G},\text{train}}/\sqrt{|L|}$; the identity initialization of Table 15 takes this path. The 10^{-4} rounding of Figure 8 is applied to the unnormalized eigenvalues and only for reported inertias.

F.1 Hyperparameters

Standard Benchmark Experiments. For experiments on the eight standard benchmarks, the hyperparameters of every model (PRAT and baselines) were selected via a hyperparameter sweep over the search ranges and distributions listed in Table 9, with the best configuration chosen by validation accuracy.

Table 9: Hyperparameter search ranges used for the standard benchmark experiments.

Hyperparameter	**Search Range**	**Distribution**
Model Architecture		
Hidden channels	64	Fixed
Number of layers	{1, 2, 3, 4}	Discrete
Attention heads	{1, 2, 4, 8, 16}	Discrete
Input dropout	$[0.0, 0.9]$	Uniform
Output dropout	$[0.0, 0.9]$	Uniform
Optimization		
Learning rate	$[10^{-4}, 10^{-1}]$	Log-uniform
Weight decay	$[10^{-5}, 10^{-2}]$	Log-uniform
Epochs	1,000	Fixed
Patience	100	Fixed

Platonov Benchmark Experiments. For experiments on the seven Platonov et al. [28] benchmarks (filtered Chameleon/Squirrel and the five large-scale heterophilic datasets), PRAT's hyperparameters were selected via a sweep starting from the official PRAT-compatible defaults of Platonov et al. [28] (hidden channels 512, layers $\in \{1, 2, 3, 4, 5\}$, lr 3×10^{-5}, dropout 0.2, $1{,}000$ training steps), and we report 10-split mean $\pm$ std on the official predefined splits. Table 10 lists the selected configurations.

Synthetic Data Experiments. When training models on synthetic data, we employed a fixed set of hyperparameters shown in Table 11 across all synthetic datasets and models. This controlled

Table 10: Selected PRAT configurations on the Platonov benchmarks, shared by Table 4 and the matched run of Section I.1. Every configuration uses LayerNorm, AdamW and training seed 42; warmup is a fraction of the training steps.

Dataset	Hidden	Layers	Heads	FFN mult.	Dropout	Learning rate	Weight decay	Steps	Warmup
Cham.-filt.	128	1	8	1	0.217	1.37×10^{-4}	3.28×10^{-2}	2,000	0.031
Sq.-filt.	512	1	8	0.5	0.627	1.59×10^{-4}	2.34×10^{-2}	500	0.042
roman-empire	512	5	8	1	0.2	3×10^{-4}	0	1,000	0
amazon-ratings	512	2	8	1	0.2	3×10^{-4}	10^{-3}	2,000	0.1
minesweeper	128	4	8	0.5	0.1	7×10^{-5}	10^{-3}	2,000	0
tolokers	128	4	4	2	0.1	3×10^{-4}	0	2,000	0
questions	512	5	8	0.5	0.2	10^{-4}	5×10^{-4}	1,000	0

Table 11: Configuration for synthetic data experiments.

Dataset Configuration		**Hyperparameters**	
Nodes	1,000	Hidden channels	64
Edges	2,000	Number of layers	2
Features	1,000	Attention heads	8
Classes	5	Input dropout	0.6
		Output dropout	0.6
		Learning rate	0.002
		Weight decay	0.0005
		Epochs	1,000
		Patience	100

setup ensures that any observed performance fluctuations are solely attributable to changes in the homophily level or graph signature index, thus facilitating a clear analysis of the behavioral trends and stability of each method.

F.2 Computing Resources

All experiments we conducted for this paper were run on a single NVIDIA A100 80GB GPU. The baseline rows of Table 4 are quoted from Platonov et al. [28], and the Geom-GCN results from Yan et al. [44]. No multi-GPU or multi-node training was used. One full hyperparameter sweep per dataset takes between 0.5 and 8 GPU-hours depending on dataset size, after which the selected configuration is rerun across multiple splits for the reported test mean / standard deviation. Final 10-split evaluations on the Platonov benchmarks (Table 4) require an additional ~ 0.5–2 GPU-hours per dataset. The total compute budget across the project, including preliminary and discarded configurations not reported in the paper, is conservatively estimated at $\sim$200 A100-hours.

F.3 Calibration of the Self-Loop Scaling Parameter

From the tangent attention framework in Appendix A.10, self-loop scores involve a scaling coefficient applied to the quadratic form $h_v^\top \mathbf{D} h_v$. We derive a principled initialization for the self-loop scaling parameter a by matching expected attention scores.

Pseudo-Label Representation. Assume hidden features encode noisy pseudo-labels in the latent space $\mathcal{L}$:

$$\mathbf{W}h_v = e_{\ell(v)} + \epsilon_v, \quad \epsilon_v \sim \mathcal{N}(0, \sigma^2 I_{|L|}), \tag{55}$$

where $\mathbf{W}$ is the weight matrix from Equation (4). $e_i \in \mathbb{R}^{|L|}$ is the one-hot encoding of label i and σ^2 quantifies the noise level in the representations.

Matching Condition. Using the graph signature matrix $S_{\mathcal{G}}$ as the metric tensor and the proportional allocation model introduced in Appendix A.6, matching the expected self-loop and edge scores yields, for $h_E > 0$:

$$a(1 + \sigma^2|L|) = \frac{2(1 - h_E)(h_E - h_E^*)}{h_E(1 - h_E^*)} + 2\sigma^2|L|, \tag{56}$$

where $h_E^* = \sum_{k=1}^{|L|}(n_k/N)^2$ is the critical homophily threshold from Theorem 2.3, with n_k denoting the number of nodes in class k. Solving for a:

$$a = \frac{\frac{2(1-h_E)(h_E-h_E^*)}{h_E(1-h_E^*)} + 2\sigma^2|L|}{1+\sigma^2|L|}. \tag{57}$$

Noise-Dominated Initialization. At the onset of training, before the network has learned label-discriminative representations, the hidden features are approximately uninformative about class labels, corresponding to the high-noise regime. In this regime, the structure-dependent term $\frac{2(1-h_E)(h_E-h_E^*)}{h_E(1-h_E^*)}$ contributes negligibly compared to the noise-driven term $2\sigma^2|L|$. Consequently, Equation (57) simplifies to:

$$a_{\text{init}} \simeq \frac{2\sigma^2|L|}{\sigma^2|L|} = 2. \tag{58}$$

This initial value is independent of both the homophily level h_E and the class distribution $\{n_k\}$, providing a universal, *structure-agnostic* initialization.

Separation of Concerns. This derivation reveals a principled division of roles between PRAT's two learnable components:

- **Metric tensor** Λ: Encodes the *geometric structure* of the graph. Initialized from the graph signature eigenvalues $\Lambda_{\mathcal{G},\text{train}}$, it captures whether the underlying manifold is Riemannian or pseudo-Riemannian, directly reflecting the homophilic or heterophilic nature of the graph.
- **Self-loop scaling** a: Controls the *signal-to-noise balance* in attention computation. Initialized at 2 corresponding to uninformative representations, it adapts during training as the network learns discriminative features.

G Statistical reliability

How the reported margins are tested, and how the spread divides between seeds and splits.

G.1 Statistical significance

Several margins in Table 3 and Table 4 are small relative to the split deviation, so we test them at two levels.

A Friedman test over all models of each table rejects equality of mean ranks on both suites, and PRAT holds the best mean rank on each: 2.63 against 2.88 for BernNet on the standard suite, with $\chi^2 = 54.46$, $p = 4.4 \times 10^{-6}$, and 1.86 against 4.57 for GT-sep on the Platonov suite, with $\chi^2 = 67.59$, $p = 1.2 \times 10^{-8}$. With eight and seven datasets the Nemenyi critical difference is 8.73 and 8.72 rank units at $\alpha = 0.05$, so this post-hoc test separates PRAT from the four and five lowest-ranked models, and the per-dataset tests carry the comparison with the strongest baselines.

Per dataset we use Welch's t where only a mean and a standard deviation are available, and an exact paired Wilcoxon signed-rank test where per-split values exist on both sides. PRAT beats GAT-sep on roman-empire at $p = 3.9 \times 10^{-5}$ and on tolokers at $p = 8.2 \times 10^{-6}$. We re-ran GAT-sep, the strongest baseline on roman-empire, in the identical pipeline, reproducing its 88.75 there, which allows a paired test on the shared splits that PRAT wins on all ten at $p = 0.002$.

In four comparisons PRAT has the higher mean but the difference is not significant at $\alpha = 0.05$, and we count these as ties: amazon-ratings against SAGE at $p = 0.55$, Citeseer against Geom-GCN at $p = 0.94$, filtered Chameleon against FAGCN at $p = 0.52$, and Actor against FAGCN at $p = 0.20$. The paired tests behind the ablation are given with Table 13 and Table 14.

G.2 Variation across training seeds

The tables vary data splits with the training seed fixed, so that a split is reproducible. To check that fixing the seed hides no variance, we reran the ten-split evaluation on the five large Platonov datasets with three training seeds, in one matched run that reproduces Table 4 to within 0.11 points.

Table 12: Ten-split mean under three training seeds, against the split-to-split deviation of the reported run.

Dataset	seed 42	seed 7	seed 123	seed-to-seed sd	split-to-split sd
roman-empire	90.16	90.08	89.98	0.09	0.56
amazon-ratings	53.79	53.56	53.39	0.20	0.59
minesweeper	92.25	92.02	91.34	0.47	0.68
tolokers	84.93	85.09	85.14	0.11	0.65
questions	78.06	78.08	77.96	0.07	1.00

In Table 12 each ten-split mean moves by under one point across the three seeds. On a fixed split, the deviation across the three seeds, pooled over the ten splits, is 0.25, 0.40, 0.50, 0.37 and 0.47 points in the order of the table, below the split-to-split deviation of a single run on every dataset, so the error bars we report capture the dominant source of variability.

H Graph Transformer and geometric baselines

Four rows of Table 3 answer two further questions: how the neighbor-only Graph Transformer used by Platonov et al. [28] behaves on these splits (GT and GT-sep), and how PRAT compares against models that already carry an indefinite or curved geometry (Q-GCN on a pseudo-hyperboloid, HAT on the Poincaré ball).

Two readings follow. First, GT is close on the two Wikipedia graphs, at 68.71 against 68.73 on Chameleon and 56.83 against 56.96 on Squirrel, and an exact paired test on the shared splits finds no significant difference ($p = 0.16$ and $p = 0.70$). GT differs from PRAT in its projections and layer architecture as well as in the metric, so it is not a geometry-matched control; the causal claim in this paper rests on the paired ablation of Section I.1, not on this table.

Second, among the geometric models the pattern is cleaner. On the three homophilic graphs, where no negative direction can be exploited, everything sits within two points. On the heterophilic half Q-GCN trails by 7.61 to 26.21 points and HAT, curved but positive definite, trails on every dataset here. Q-GCN also trails PRAT by 3.87 and 4.48 points on filtered Chameleon and filtered Squirrel, at 38.82 ± 2.28 and 35.12 ± 1.36.

I The sign-conditional ablation

What happens when the metric is forbidden to be indefinite, and what that does and does not license.

I.1 Three-way ablation on the Platonov suite

Table 5 runs the three-way ablation on the standard benchmarks. Table 13 repeats it on the Platonov suite and the filtered pair, inside one matched run whose PRAT rows reproduce Table 4 to within 0.24 points. Because the three variants share every split, an exact paired Wilcoxon signed-rank test over the ten splits is available, which a comparison of means alone is not.

Table 13: Three-way ablation on the Platonov benchmarks, grouped as in Table 4, with the graph signature index of each dataset. p is an exact paired Wilcoxon signed-rank test of PRAT against PRAT_{pos} over the ten shared splits; **bold** p marks significance at $\alpha = 0.05$. Among the three variants the best mean is in **bold**, and where means tie the smaller deviation takes it.

Dataset	$s_{\mathcal{G}}$	PRAT	PRAT_{pos}	$\text{PRAT}_{\text{fixed}}$	p
Cham.-filt.	(3,0,2)	**42.45±2.20**	40.39±2.73	42.00±2.66	**0.039**
Sq.-filt.	(2,1,2)	39.58±1.71	39.39±2.16	**40.22±1.59**	0.770
roman-empire	(6,7,5)	**90.16±0.56**	87.56±0.57	90.02±0.51	**0.002**
amazon-ratings	(3,2,0)	**53.79±0.59**	53.72±0.48	53.70±0.63	0.492
minesweeper	(1,1,0)	92.25±0.68	**92.28±0.56**	92.09±0.70	0.846
tolokers	(2,0,0)	84.93±0.65	84.83±0.72	**84.93±0.57**	0.131
questions	(1,1,0)	**78.06±1.00**	77.92±1.21	77.91±1.07	0.625

The gap between PRAT and PRAT_{pos} appears only where the signature has negative eigenvalues, and there it is significant: $+2.60$ points on roman-empire on all ten splits, and $+2.06$ on filtered Chameleon on eight of ten. On the four datasets with $n_- = 0$ the two variants differ by at most 0.14 points at $p \geq 0.13$, and the single nominal win for the constraint is minesweeper by 0.03 points at $p = 0.85$. Filtered Squirrel is the boundary case. Its training-split spectrum is nearly flat beyond $\lambda_1 = 0.1375$, the next three eigenvalues carrying 3.2%, 0.2% and -0.7% of it, so what the initialization supplies there is near-zero suppression rather than sign, and the positive-definite constraint costs 0.19 points at $p = 0.77$. A constraint that mattered for generic reasons would not localize this way.

$\text{PRAT}_{\text{fixed}}$ matches PRAT on six of the seven, the exception being minesweeper by 0.16 points at $p = 0.002$. Freezing Λ freezes the inertia and not the metric, since $\mathbf{D} = (\mathbf{W}')^\top \Lambda \mathbf{W}'$ keeps adapting through $\mathbf{W}'$, so these rows say that on large graphs the initialization is what has to supply the spectrum while the pullback is still learned. Where the training-split signature is noisy, on the small graphs of Table 5, learning Λ as well is what recovers 16.7 points on Cornell.

I.2 The signature as a diagnostic rather than a predictor

It is natural to ask whether n_- predicts how much a signed or high-pass model gains. Rank-correlating n_- from Table 1 against what each such baseline of Table 3 gains over GCN is positive throughout, at $\rho = +0.23$ for FAGCN, $+0.33$ for GPR-GNN, $+0.60$ for BernNet and $+0.33$ for their mean gain; none is significant at eight datasets, and Squirrel is the exception, where the three models lose 2.82 points on average despite $n_- = 1$.

We therefore present $s_{\mathcal{G}}$ as a diagnostic of the condition under which an indefinite mechanism should pay, not as a predictor of accuracy. The direct evidence for the diagnostic reading is the sign-conditional ablation of Section I.1, which manipulates the admissible sign structure with everything else held fixed and finds the effect exactly where the condition holds. Predicting the gain is hard for measures of this kind in general: Zheng et al. [48] report their own composite homophily score correlating with the graph-aware advantage at $p \geq 0.11$ over thirty-one graphs.

J What the initialization has to supply

Which part of the signature carries the effect, and whether ground-truth labels are needed to obtain it.

J.1 What the initialization has to supply

The ablation above shows that the freedom to be indefinite is worth something. It does not say which part of the signature carries it, the sign pattern or the magnitudes. Table 14 separates the two with four controls that keep the architecture fixed and change the initial diagonal; `abs_pos` also holds Λ positive definite during training, as PRAT_{pos} does.

Table 14: Initialization controls on the two indefinite benchmarks. The signature row is the reference: Δ is the loss against it and p an exact paired Wilcoxon over ten shared splits, with **bold** p marking significance at $\alpha = 0.05$. `abs_pos` keeps the magnitudes and deletes the negative directions; `sign` keeps the signs at a uniform magnitude; `flip` keeps the magnitudes and inverts the signs; `random` draws an indefinite diagonal of the same inertia at the signature's scale.

	roman-empire (6,7,5)			Cham.-filt. (3,0,2)		
Initialization	Accuracy	Δ	p	Accuracy	Δ	p
signature (PRAT)	90.16±0.56	–	–	42.45±2.20	–	–
`abs_pos`	87.91±0.61	2.26	**0.002**	41.64±2.50	0.81	0.275
`sign`	89.86±0.62	0.30	0.065	39.90±2.22	2.55	**0.008**
`flip`	89.96±0.64	0.21	0.084	39.44±3.20	3.01	**0.027**
`random` 1	89.53±0.31	0.63	**0.010**	40.81±2.77	1.63	**0.037**
`random` 2	90.16±0.47	0.00	1.000	39.83±2.40	2.62	**0.037**
`random` 3	89.58±0.42	0.58	**0.002**	40.11±3.52	2.34	0.055
identity	87.61±0.69	2.55	–	40.18±2.39	2.27	–
positive definite	87.56±0.57	2.60	**0.002**	40.39±2.73	2.06	**0.039**

The controls give different results on the two graphs, and the difference is informative. On roman-empire the only variant that loses decisively is `abs_pos`, which keeps the magnitude profile and removes the negative directions: it costs 2.26 points on all ten splits. Everything that retains negative directions, including sign-only, sign-flipped and all three random draws of matching inertia, stays within 0.7 points. The reason is visible in the spectrum: eleven of roman-empire's eighteen training-split eigenvalues are below a sixth of λ_1 in magnitude, so the profile is close to flat and the only information the signature carries there is that negative directions exist. On filtered Chameleon the spectrum is not flat, its second eigenvalue carrying 10.8% of the first against 3.2% on filtered Squirrel, and there the specific values matter: every random draw of matching inertia loses 1.63 to 2.62 points and sign-only loses 2.55.

An uninformed initialization is the other end of the scale. An identity initialization that is free to become indefinite but carries no signature information reaches 87.61 on roman-empire, level with the positive-definite variant at 87.56. Freedom alone therefore buys nothing; what has to be supplied is the negative directions and, where the spectrum is informative, their relative sizes as well.

The initialization also survives training. Figure 3 reports the converged positive-eigenvalue ratio on synthetic graphs. On real graphs we logged Λ throughout training. Between 97.2% and 100% of its entries keep their initial sign at convergence, the cosine to the initialization stays above 0.985, and roman-empire's negative-entry fraction holds between 0.38 and 0.42 while the magnitudes adapt. Every sign change falls on an entry initialized from a training-split eigenvalue within 10^{-4} of zero, which the convention of Figure 8 rounds to zero, so every entry whose training-split eigenvalue counts as positive or negative keeps its sign. On tolokers, whose signature is strictly positive, no entry becomes negative. The model does not drift back to a Riemannian solution when it is free to, which is what the diagnose-then-learn reading predicts.

J.2 Estimating the signature without labels

The signature is computed from training-node labels, but nothing in the forward pass reads a label. To test whether the labels are the essential ingredient or merely a convenient one, we re-estimate the signature from k-means pseudo-labels on node features, with the number of clusters set to the number of classes and no ground-truth label used in the estimate, and train as usual.

Table 15: Label-free initialization. The inertia is that of the eigenvalue vector each initialization is built from, before normalization, at the 10^{-4} rounding convention of Figure 8; the two signature estimates use the training nodes.

	roman-empire		Cham.-filt.	
Initialization	Accuracy	inertia	Accuracy	inertia
label-based signature	**90.16±0.56**	(8,4,6)	42.45±2.20	(3,0,2)
k-means pseudo-labels	90.10±0.48	(7,4,7)	**42.70±2.00**	(1,3,1)
uninformed identity	87.61±0.69	(18,0,0)	40.18±2.39	(5,0,0)

In Table 15 recovery is complete on both indefinite benchmarks, while the uninformed initialization loses 2.55 and 2.27 points. The feature-only estimate is not a reconstruction of the label signature: its inertia is $(7, 4, 7)$ against $(8, 4, 6)$ on roman-empire and $(1, 3, 1)$ against $(3, 0, 2)$ on filtered Chameleon, and it still recovers the accuracy. Taken with Section J.1, the picture is that what has to be supplied is the presence of negative directions together with near-zero suppression, not a faithful copy of the label spectrum, and that node features alone can supply it. A complete self-supervised treatment, the training objective as well as the initialization, is outside the scope of this paper.

K Behavior under sparse and noisy labels

How far the signature and the accuracy survive when training labels are few or wrong.

K.1 Stability of the signature under label sparsity and noise

Table 17 varies how many labels the model trains on. A separate question is how quickly the signature estimate itself degrades. We keep a random fraction p of the training nodes, or flip a fraction η of the

training labels to a random other class, recompute the signature and report the mean sign agreement of its eigenvalues against the full-training-label estimate over 100 repeats.

Table 16: Mean eigenvalue sign agreement of the estimated signature against the full-training-label signature, over 100 repeats.

	fraction of training nodes kept			fraction of training labels flipped		
Dataset	p=10%	p=25%	p=50%	η=10%	η=20%	η=30%
roman-empire	0.88	0.88	0.93	0.97	0.93	0.87
amazon-ratings	0.74	0.85	0.91	1.00	0.87	0.80
minesweeper	0.69	0.82	0.98	1.00	1.00	1.00
tolokers	0.99	1.00	1.00	1.00	1.00	0.96
questions	0.93	1.00	1.00	1.00	1.00	1.00
Chameleon	0.92	0.93	0.91	0.94	0.93	0.92
Squirrel	0.93	0.92	0.93	0.95	0.93	0.90
Actor	0.45	0.53	0.67	0.86	0.83	0.84
Cornell	0.11	0.53	0.85	0.93	0.90	0.87
Texas	0.34	0.67	0.94	0.86	0.81	0.81

Table 16 reports the agreement. Large graphs are stable at modest label fractions. Actor's low agreement sits in near-zero eigenvalues at the rounding threshold, where the dominant directions keep their sign, and the 183-node WebKB graphs lose the signature below a quarter of an already small training set. That is exactly where Table 17 shows Cornell losing its lead, with all three models inside 0.6 points at 10%, and it is why the learnable Λ matters most on small graphs.

What matters in the end is accuracy. Training PRAT from the corrupted initialization, with training labels left clean and only the initialization degraded, costs at most 0.11 points on roman-empire ($90.16 \to 90.05$ at $\eta = 10\%$) and 1.46 on filtered Chameleon ($42.45 \to 40.99$ at $p = 25\%$), both inside the split deviation, and every corrupted variant stays at or above the positive-definite ablation.

K.2 Training Fraction Sensitivity

We evaluate how PRAT's advantage depends on label availability by training on 10%, 20%, 40%, 60%, and 100% of the original training split on Cornell, Texas, Actor, and Cora, using each model's best hyperparameters and 10 data splits.

Table 17: Accuracy (%) at varying training fractions. Bold indicates the best model per dataset and fraction.

Dataset	Model	10%	20%	40%	60%	100%
Cornell	PRAT	44.05±4.53	**49.46±9.37**	**57.03±8.59**	**68.38±6.40**	**71.89±4.71**
	GAT	44.32±5.16	41.62±8.04	45.68±9.00	46.76±9.37	50.54±7.55
	GATv2	**44.59±5.44**	41.35±4.02	43.51±3.91	46.49±10.38	54.32±9.70
Texas	PRAT	**63.24±6.42**	**67.57±9.82**	**74.05±6.42**	**78.65±5.33**	**84.32±5.10**
	GAT	60.54±6.07	59.73±9.16	61.62±5.51	61.08±7.27	64.59±6.67
	GATv2	57.57±6.95	60.54±8.65	61.62±4.80	62.16±6.16	66.76±6.63
Actor	PRAT	**27.40±1.61**	**28.93±1.20**	**33.72±1.14**	**35.61±1.57**	**37.26±1.44**
	GAT	26.32±1.11	27.53±1.75	28.79±1.64	29.56±1.31	31.42±0.69
	GATv2	25.60±1.33	27.46±1.10	30.36±1.39	32.28±1.33	34.24±1.10
Cora	PRAT	74.24±14.97	83.09±2.12	85.27±1.16	86.61±1.14	**88.42±1.17**
	GAT	**80.11±1.99**	82.98±1.76	84.88±1.17	86.54±1.32	88.32±1.09
	GATv2	78.88±1.75	**83.26±1.98**	**85.62±1.26**	**86.74±1.06**	87.99±0.93

On heterophilic datasets, PRAT consistently outperforms GAT and GATv2, but the gap narrows with fewer labels (Texas: +19.7pp at 100% $\to$ +2.7pp at 10%). This reveals the mechanism behind PRAT's advantage: as $S_{\mathcal{G}}$ becomes noisier with fewer labels, initialization quality degrades and performance approaches that of standard attention.

On homophilic Cora, all models perform comparably at fractions $\geq 20\%$, as expected, since the positive-definite structure of Cora's $S_{\mathcal{G}}$ initializes PRAT's metric as positive semi-definite, offering no additional geometric advantage over GAT/GATv2. At 10%, PRAT shows higher variance (74.24±14.97%) due to one split where $S_{\mathcal{G}}$ computed from $\sim$163 training nodes across 7 classes produces a severely noisy initialization. This is consistent with our analysis: the $S_{\mathcal{G}}$ quality degrades

when training labels are extremely sparse relative to the number of classes, but performance recovers rapidly at 20% (83.09±2.12%).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are supported by the theoretical analysis in Section 2, the proposed PRAT method in Section 3, and empirical results in Section 4. The scope is limited to node classification on homophilic and heterophilic graph benchmarks.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations paragraph after the conclusion states four: the signature is initialized from training-node labels, which ties the method as presented to (semi-)supervised node classification; signature shift between training and test graphs is not studied; graph-level tasks are out of reach because $S_{\mathcal{G}}$ is undefined without node labels; and $\mathrm{rank}(\mathbf{D}) \leq |L|$ bounds the structure the metric can carry when classes are few. Appendix J.2 reports the label-free variant that weakens the first of these.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper states the assumptions and formal results in Section 2, including Theorems 2.2–2.5, and provides detailed proofs in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4.2 specifies the train/validation/test split, use of fixed and random splits, self-loop preprocessing, random seed control, and hyperparameter tuning protocol. Detailed configurations are provided in Appendix F.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide a repository at `https://github.com/voltwin-dev/PRAT_NeurIPS_2026`, containing the implementation of PRAT, scripts for reproducing the experiments, configuration files, license information, and instructions for the reported benchmarks. The released code is provided for research and non-commercial use only due to patent-related restrictions, while the datasets used in this work are publicly available benchmark datasets and cited in Section 4.1.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies dataset splits, preprocessing, random seeds, hyperparameter tuning, and evaluation metrics in Section 4.2, with additional configuration details in Appendix F.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All benchmark, ablation and appendix tables report the mean over the ten fixed evaluation splits with the standard deviation across those splits, so the error bars capture split variability at a fixed seed. Appendix G.1 adds a Friedman test across models on each suite and, per dataset, Welch's t-test and the exact paired Wilcoxon signed-rank test over the ten splits, and reports which comparisons are ties. Appendix G.2 separates seed variability from split variability by repeating the ten-split evaluation under three training seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments we conducted were run on a single NVIDIA A100 80GB GPU; the used computing resources are reported in Appendix F.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: The research appears to conform to the NeurIPS Code of Ethics and does not involve human subjects, private data, or high-risk deployment.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper does not include a separate discussion of societal impacts. PRAT is a foundational method for node classification on graphs and is not tied to a particular application or deployment, and we do not see a direct path from it to negative applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a high-risk pretrained model, scraped dataset, or system with a clear high-risk misuse scenario requiring special safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original sources of the benchmark datasets and baseline methods used in the experiments. The repository documents the licenses and terms of use for the datasets and baseline code assets, and states that the released PRAT implementation is provided under a research-only, non-commercial license due to patent-related restrictions.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release an implementation of PRAT with documentation, experiment scripts, configuration files, license information, and instructions for reproducing the reported results. The released PRAT code is made available for research and non-commercial use only due to patent-related restrictions.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects, so IRB approval or equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as an important, original, or non-standard component of the core method, experiments, or scientific contributions of this work.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.