Geometric Perturbation Graph Neural Network for Heterophily Modeling

Sanghyun Jung*
sjung242@uwo.ca
Western University
London, Ontario, Canada

María Óskarsdóttir
University of Southampton
Southampton, United Kingdom
Reykjavik University
Reykjavik, Iceland

Cristián Bravo
Western University
London, Ontario, Canada

Abstract

Graph Neural Networks (GNNs) excel when applied over homophilic graphs but often struggle with heterophilic ones, where connected nodes carry dissimilar information. This degradation is typically caused by indiscriminate message passing that mixes heterogeneous signals, leading to feature oversmoothing. To tackle this problem, we propose the Geometric Perturbation Graph Neural Network (GP-GNN), a framework that decomposes neighborhood interactions into distinct geometric components: projections and orthogonal residuals. Our core operator treats message passing as a controlled perturbation of node identities, modulated by an edge-level gating mechanism. This allows the model to adaptively weight its geometric components, prioritizing feature consensus in homophilic regions while selectively emphasizing neighborhood differences in heterophilic regions. Unlike prior methods that impose fixed spectral biases, GP-GNN provides a unified mechanism that adapts to local neighborhood geometry in the feature space. Extensive experiments on various benchmarks demonstrate that GP-GNN consistently matches or outperforms state-of-the-art methods in transductive settings. Our results show that adaptive geometric perturbation provides a robust and flexible inductive bias that successfully generalizes across diverse graph regimes, ranging from high homophily to extreme heterophily. [1]

CCS Concepts

• **Computing methodologies** → **Neural networks**; **Network science**.

Keywords

Graph Neural Networks, Transductive Learning, Heterophily

1 Introduction

Graph-structured data arises in a wide range of real-world applications, including social networks, recommender systems, fraud detection, and scientific modeling [8, 35]. Graph Neural Networks (GNNs) have established themselves as the dominant paradigm for learning on such data by propagating and aggregating information along edges [9, 13]. However, even architectures such as graph attention networks (GAT) [32] and Graph Transformers (GT) [29] typically rely on scalar mixing coefficients. Since these weights often function as similarity-based filters, they reinforce the implicit assumption of homophily, where connected nodes are expected to share similar features [21, 40].

This reliance on similarity is frequently violated in heterophilic graphs, where edges encode complementary, functional, or adversarial relationships. For example, e-commerce graphs often link complementary products from disparate categories, such as a stand mixer and bread flour, rather than items with identical class labels [25]. Similarly, dating networks exhibit disassortative mixing by gender [16], while financial forensics reveals illicit actors connecting to licit entities to launder funds [34]. Functional heterophily also prevails in role-based systems, such as Q&A communities connecting information seekers with experts [25].

In response to these limitations, various architectures have been proposed to improve GNN performance under heterophily [1, 10, 40]. In general, these methods utilize spectral filters to emphasize high-pass components or introduce global-context mechanisms to reshape neighborhoods. However, most approaches still model the influence of the neighbors through scalar mixing coefficients applied to whole feature vectors, leaving the pairwise geometry of interactions implicit. Crucially, they rarely distinguish between neighbors that are negatively aligned with a target and those that are orthogonal yet complementary, even though these cases carry distinct semantic meanings in heterophilic graphs.

We argue that effective learning in diverse graph regimes requires making this pairwise geometry explicit. Standard aggregators typically treat dissimilar neighbors as generic noise to be attenuated, yet these neighbors often represent distinct structural phenomena. For example, in fraud networks, role complementarity may occur when a fraudulent account connects to a mule account, a middle-man used to launder funds. Though their features are dissimilar, the mule provides a functional context that reveals the illicit pathway. In contrast, negative alignment arises in adversarial settings, such as scam payments to benign customers. These customers may exhibit patterns in opposite directions to fraudulent behavior, providing a contrastive signal. Treating both as generic heterophilic noise is problematic because the mule serves as evidence of crime, while the customer represents a legitimate user to be protected. An effective architecture must therefore distinguish between signed alignment for adversarial contrast and orthogonal residuals for functional complementarity to preserve the unique structural signatures of each interaction.

Motivated by this, we introduce the *Geometric Perturbation Graph Neural Network* (GP-GNN). Its core operator, *Geometric Perturbation Convolution* (GP-Conv), constructs messages by: (i) decomposing interactions into a signed projection and an orthogonal residual; (ii) employing an edge-conditioned gating head to produce a geometric

[1] Our code is available at https://anonymous.4open.science/r/GPGNN-89CC.

GALOP Workshop at KDD 2026: Geometric Space, Architecture and Learning Objective for Large Pre-Trained Models, August 9, 2026, Jeju, Republic of Korea.
2026.

refinement vector; and (iii) injecting this refinement as a controlled perturbation anchored to the neighbor's identity to form the final message.

Concretely, GP-Conv constructs messages by performing a bidirectional geometric decomposition of each pair of neighbors. For each edge (j, i), the operator calculates the projection and residual of the source $\mathbf{h}_j$ representation relative to the target $\mathbf{h}_i$, and simultaneously the projection and residual of the target $\mathbf{h}_i$ relative to the source $\mathbf{h}_j$. This dual-perspective information is modulated by four distinct, symmetric gates derived from joint node transformations. By injecting this geometric refinement as a controlled perturbation anchored to the neighbor's identity $\mathbf{h}_j$, the model ensures that the propagated message encodes both the neighbor's semantic content and its specific relational orientation relative to the target node. A global scaling factor β constrains these updates, while a separate root transformation on the target identity ensures that node-specific features are preserved, effectively mitigating feature oversmoothing in deep architectures.

Our contributions are as follows. First, we propose GP-GNN, a framework that, to the best of our knowledge, is the first to explicitly decompose neighborhood signals into signed alignment and orthogonal residuals. Using an identity-anchored perturbation, we move beyond binary homophily-heterophily assumptions to distinguish between negative feature correlation and complementary structural information. Second, we introduce GP-Conv, a bidirectional operator that constructs reciprocal projection-residual decompositions from both endpoints of an edge. These geometric interactions are modulated by a symmetric gating mechanism. Third, we demonstrate that this geometric decomposition enables the model to preserve critical structural properties and localized functional roles, which are typically lost in standard GNNs. Finally, we evaluate GP-GNN across ten benchmarks that span both strongly heterophilic and standard highly homophilic regimes, showing that our method exceeds state-of-the-art performance in the former and matches it in the latter. Furthermore, our representational analysis quantifies the relative contribution share, that is, the proportion of the total message signal attributed to each geometric component. We find that as homophily decreases, the model autonomously shifts the representational weight from the projection components to the residual components. This systematic gating strategy allows GP-GNN to prioritize feature consensus in homophilic settings while using feature variance orthogonal to the target node's representation to prevent feature collapse in heterophilic graphs.

2 Related Work

Learning on heterophilic graphs. Heterophilic graphs violate the usual homophily assumption: connected nodes often possess dissimilar features and labels. Standard GNNs frequently underperform in such settings, sometimes falling below simple MLP baselines [40]. While some "good" heterophilic graphs remain solvable for vanilla GCNs [19], challenging cases have required specialized architectures. Spatial-domain methods modify message-passing by separating ego- and neighbor-embeddings (H_2GCN, FSGNN) [20, 40], learning class-compatibility matrices (CPGNN) [39], or utilizing large-scale adjacency information (LINKX) to capture structural patterns independently of features [16]. Spectral approaches learn adaptive graph filters beyond low-pass smoothing: GPR-GNN and BernNet use tunable polynomial filters to capture band-pass components [3, 10], while FAGCN and JacobiConv learn frequency responses to amplify high-frequency signals [1, 33]. More recently, adaptive multi-channel frameworks like ACM-GNN and GBK-GNN dynamically mix homophilic and heterophilic channels [5, 18], and global-context methods such as GloGNN introduce auxiliary global nodes to aggregate information from similar nodes across the graph [15]. Finally, geometric approaches such as Geom-GCN and Neural Sheaf Diffusion capture heterophily by embedding nodes into latent geometric spaces or sheaf structures where structurally similar nodes can exchange information without direct adjacency [2, 23].

In summary, a broad spectrum of spatial, spectral, and adaptive strategies have been developed to improve GNN performance under heterophily. However, these methods typically address heterophily as a global filtering problem or a node-level mixing task, often requiring auxiliary structural information or complex spectral tuning. While effective, they treat the neighborhood as an undifferentiated signal, lacking a primitive mechanism to resolve the specific geometric relationship between any two connected nodes. Furthermore, many specialized architectures designed for heterophily are prone to over-correcting, often leading to performance degradation in homophilic regimes where identity-based smoothing remains optimal [26]. This creates a representational bottleneck where diverse relational types, such as complementary versus adversarial links, are conflated into a single high-frequency channel.

Directional and geometry-aware message passing. Beyond homophily ratios and spectral filters, several works emphasize directionality and feature-space geometry. Dir-GNN [27] shows that treating edges as directed can improve performance by separately aggregating incoming and outgoing edges. Signed-graph GNNs encode antagonistic relationships by propagating messages differently over positive and negative edges [4, 12, 30]. A different line of work incorporates feature differences $(\mathbf{h}_j - \mathbf{h}_i)$ as contrastive signals; for instance, Difference Residual Graph Neural Networks (DRC) introduce difference-based residual connections that inject neighbor feature differences alongside standard aggregation [36]. Similarly, GDFGAT employs feature-difference-based attention for fraud detection [31]. However, these methods typically implement interaction as a monolithic contrastive channel. By reducing interactions to a single difference-based scalar or vector, these methods conflate diverse heterophilic signals like active opposition and complementarity. This lack of geometric resolution prevents the model from distinguishing between different types of non-homophilic relationships.

GP-GNN departs from these ad-hoc strategies by introducing a bilateral geometric decomposition at the message level. Rather than relying on scalar mixing, we explicitly partition each interaction into signed alignments and orthogonal residuals from both the source-to-target $(j \to i)$ and target-to-source $(i \to j)$ perspectives. This allows the model to capture how node identities mutually reinforce or oppose one another. This bilateral operator provides a robust inductive bias that enables the model to navigate diverse graph regimes autonomously. As evidenced by our analysis of relative contribution shares—a metric quantifying the L_2 impact of

each geometric component (see Section 6.1 for visual evidence)—GP-GNN systematically reallocates its representational capacity from feature-based consensus in homophilic settings to structural contrast in heterophilic ones. Specifically, by utilizing variance that is orthogonal to the representations of both endpoints, the model preserves critical structural signatures and localized functional roles that are typically obscured by over-smoothing or generic high-pass filtering. This geometry-aware strategy allows GP-GNN to prevent feature collapse in complex heterophilic networks while maintaining stability in assortative regimes.

3 Preliminaries

Graph Neural Networks (GNNs) learn node representations by iteratively propagating and transforming information over the graph structure. Most GNNs follow the message-passing paradigm, in which node representations are updated based on information aggregated from local neighborhoods [7, 9, 13, 32].

Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, where $\mathcal{V}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of (directed or undirected) edges. We denote by $\mathbf{h}_i^{(l)} \in \mathbb{R}^d$ the representation of node $i \in \mathcal{V}$ in layer l. A generic GNN layer computes the representation of node i by first aggregating messages from its neighbors $\mathcal{N}(i)$:

$$\mathbf{m}_i^{(l)} = \text{AGGREGATE}^{(l)}\left(\left\{\text{MESSAGE}^{(l)}(\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}) \mid j \in \mathcal{N}(i)\right\}\right), \tag{1}$$

followed by an update step:

$$\mathbf{h}_i^{(l)} = \text{UPDATE}^{(l)}\left(\mathbf{h}_i^{(l-1)}, \mathbf{m}_i^{(l)}\right). \tag{2}$$

A prominent example is the Graph Convolutional Network (GCN) [13], which instantiates message passing using a degree-normalized adjacency operator:

$$\mathbf{H}^{(l)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)}\right), \tag{3}$$

where $\mathbf{H}^{(l-1)}$ and $\mathbf{H}^{(l)}$ denote the node-feature matrices (hidden representations) at layers $l-1$ and l, $\mathbf{W}^{(l)}$ is a learnable weight matrix and $\sigma(\cdot)$ is an element-wise nonlinearity. Moreover, $\tilde{\mathbf{A}} = \mathbf{A}+\mathbf{I}$ is the adjacency matrix with self-loops and $\tilde{\mathbf{D}}$ is its degree matrix.

This operator performs low-pass filtering, which is effective for homophilic graphs but fails in heterophilic settings. In the latter, indiscriminate averaging propagates misleading signals and drives embeddings toward neighborhood means, leading to representation collapse and oversmoothing as network depth increases [14, 22, 23, 40]. These limitations motivate architectures that disentangle neighborhood information to selectively control propagation rather than applying uniform smoothing. In the following section, we introduce such an architecture based on a geometric decomposition of pairwise interactions.

4 Method: Geometric Perturbation GNN (GP-GNN)

GP-GNN treats message passing as a controlled geometric perturbation of node identities to achieve robustness across the homophily-heterophily spectrum. We first detail an adaptive geometric decomposition that partitions pairwise signals into signed projections and orthogonal residuals, capturing nuanced structural relationships. We then introduce the Geometric Perturbation Convolution (GP-Conv), which employs an edge-conditioned gating mechanism to selectively modulate these components. Finally, we describe the backbone architecture, demonstrating how these layers integrate into a deep residual framework. By modeling mutual geometric relationships rather than unidirectional signal flow, GP-GNN preserves informative structural signals that standard GNNs typically suppress. To simplify notation, we omit the layer superscript (l) throughout this section unless otherwise specified.

4.1 Adaptive Geometric Decomposition

We take advantage of the principle of orthogonal decomposition to partition the source representation $\mathbf{h}_j$ into two fundamental source-centric geometric components relative to the target direction $\mathbf{h}_i$: a projection component (semantic alignment) and an orthogonal residual component.

Formally, we define the signed projection $\mathbf{p}_{j\to i}$ of the neighbor $\mathbf{h}_j$ onto the target $\mathbf{h}_i$, and its corresponding orthogonal residual $\mathbf{r}_{j\to i}$, as follows:

$$\mathbf{p}_{\mathbf{j}\to\mathbf{i}} = \frac{\mathbf{h}_j^\top \mathbf{h}_i}{\|\mathbf{h}_i\|_2^2 + \epsilon}\,\mathbf{h}_i, \qquad \mathbf{r}_{j\to i} = \mathbf{h}_j - \mathbf{p}_{j\to i}, \tag{4}$$

where $\epsilon > 0$ is a small constant for numerical stability.

This decomposition partitions the magnitude of $\mathbf{h}_j$ according to the Pythagorean identity:

$$\|\mathbf{h}_j\|_2^2 = \left\|\mathbf{p}_{j\to i}\right\|_2^2 + \left\|\mathbf{r}_{j\to i}\right\|_2^2 \quad (\text{up to } \epsilon). \tag{5}$$

The relative magnitudes of these terms quantify the structural relationship: a dominant projection suggests feature consensus (homophily), while a dominant residual highlights local contrast (heterophily).

Crucially, the scalar term in the projection is signed because it is proportional to the inner product $\mathbf{h}_j^\top \mathbf{h}_i$. A negative inner product indicates opposition ("repulsive" heterophily), where $\mathbf{h}_j$ is inversely aligned with $\mathbf{h}_i$, while a positive inner product indicates attraction (homophilic alignment). In contrast, the residual term $\mathbf{r}_{j\to i}$ isolates feature variance orthogonal to $\mathbf{h}_i$, capturing complementary or residual information that cannot be explained by the attributes of the target node. We apply the same construction symmetrically to obtain the target-centric components $\mathbf{p}_{i\to j}$ and $\mathbf{r}_{i\to j}$.

4.2 Geometric Perturbation Convolution

The GP-Conv layer implements an identity-anchored geometric perturbation mechanism. For each edge (j, i), the interaction between $\mathbf{h}_i$ and $\mathbf{h}_j$ defines a bilateral relational frame: GP-Conv performs a dual decomposition, partitioning the representation of the source relative to the target and simultaneously the representation of the target relative to the source. By preserving the relative magnitudes and signed alignment of these four components, the operator allows the gating mechanism to calibrate messages based on the mutual orientation relative magnitudes of the node representations.

Bilateral symmetric gating. For a directed message flow from source j to target i, via edge (j, i), GP-Conv computes a four-channel gating vector $\mathbf{g}_{ji} \in \mathbb{R}^4$ that modulates the intensity of each reciprocal component in the geometric decomposition. To ensure structural robustness and align with the mutual nature of

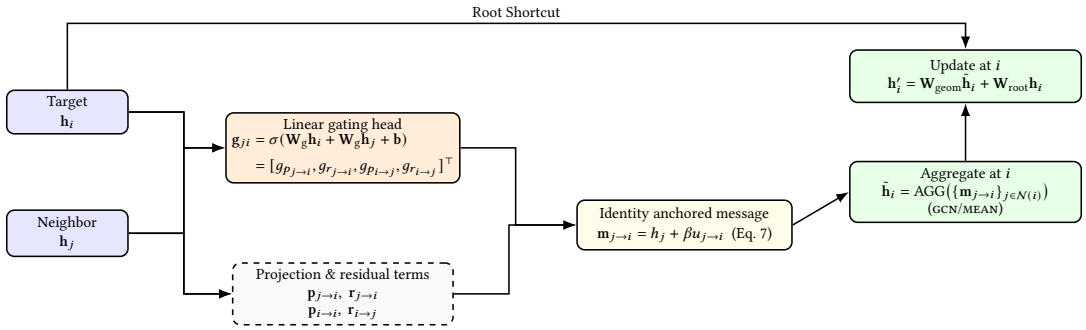

Figure 1: GP-Conv message passing. A linear gating head produces scalar weights from $(\mathbf{h}_i, \mathbf{h}_j)$, which modulate projection and residual components to construct an identity-anchored message $\mathbf{m}_{j\to i}$ (Eq. 7). Incoming messages are aggregated using AGG and combined with a root transform.

heterophilic relationships, we employ a symmetric joint gating mechanism, where the influence of a neighbor is invariant to edge orientation. The gates are computed as:

$$\mathbf{g_{ji}} = [g_{p_{j\to i}}, g_{r_{j\to i}}, g_{p_{i\to j}}, g_{r_{i\to j}}]^\top = \sigma\big(\mathbf{W}_g\mathbf{h_i} + \mathbf{W}_g\mathbf{h_j} + \mathbf{b}\big), \quad (6)$$

where $\mathbf{W}_g \in \mathbb{R}^{4\times d}$ is a learnable weight matrix and $\mathbf{b} \in \mathbb{R}^4$ is a bias vector. Using a single weight matrix for both endpoints enforces a joint structural bias, while the four distinct channels allow the model to independently weight the projections and residuals for both directions of the bilateral interaction. Although the four gating components are symmetric ($\mathbf{g_{ji}} = \mathbf{g_{ij}}$) with respect to the node pair, they modulate reciprocal geometric decompositions that are unique to the feature-space orientation of each endpoint. This ensures that while the nature of the interaction is mutually determined, the resulting feature updates are node-specific and context-aware, as the symmetric gates modulate reciprocal projections and residuals that are inherently anchored to each node's unique position in the feature space.

Message construction. GP-Conv anchors the message at the neighbor identity $\mathbf{h}_j$ and injects an edge-conditioned geometric perturbation $\mathbf{u}_{j\to i}$ scaled by a fixed intensity β:

$$\mathbf{m}_{j\to i} = \mathbf{h}_j + \beta\, \mathbf{u}_{j\to i} \quad (7)$$

The perturbation $\mathbf{u}_{j\to i}$ is defined as the independently gated sum of the four bilateral components:

$$\mathbf{u}_{j\to i} = g_{p_{j\to i}}\, \mathbf{p}_{j\to i} + g_{r_{j\to i}}\, \mathbf{r}_{j\to i} + g_{p_{i\to j}}\, \mathbf{p}_{i\to j} + g_{r_{i\to j}}\, \mathbf{r}_{i\to j} \quad (8)$$

Here, β modulates the influence of geometric updates relative to the neighbor identity. By anchoring the message at $\mathbf{h}_j$, this additive structure acts as an edge-level residual connection. This provides a direct gradient path that prevents vanishing signals during the transformation of neighborhood features. Consequently, β stabilizes early training by ensuring that the complex geometric components, such as orthogonal residuals, perturb rather than replace the foundational feature consistency.

Aggregation and update. Node-wise updates are obtained by aggregating incoming messages from all neighbors with a configurable aggregation rule and combining the result with a root transform:

$$\tilde{\mathbf{h}}_i = \mathrm{AGG}\big(\{\mathbf{m}_{j\to i} : j \in \mathcal{N}(i)\}\big), \quad \mathbf{h}'_i = \mathbf{W}_{\mathrm{geom}}\tilde{\mathbf{h}}_i + \mathbf{W}_{\mathrm{root}}\mathbf{h}_i, \quad (9)$$

where the weight matrices $\mathbf{W}_{\mathrm{geom}}, \mathbf{W}_{\mathrm{root}} \in \mathbb{R}^{d\times d}$ are learnable.

We propose two instantiations in our implementation, AGG, as follows:

$$\mathrm{AGG}_{\mathrm{MEAN}}: \quad \tilde{\mathbf{h}}_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j\in\mathcal{N}(i)} \mathbf{m}_{j\to i}, \quad (10)$$

$$\mathrm{AGG}_{\mathrm{GCN}}: \quad \tilde{\mathbf{h}}_i = \sum_{j\in\mathcal{N}(i)} \frac{1}{\sqrt{d_i d_j}}\, \mathbf{m}_{j\to i}, \quad (11)$$

where d_i and d_j denote node degrees. While $\mathrm{AGG}_{\mathrm{MEAN}}$ provides a stable estimate of the neighborhood distribution, we include $\mathrm{AGG}_{\mathrm{GCN}}$ to take advantage of its symmetric normalization properties. This is particularly critical for high-degree nodes found in dense networks, as the $1/\sqrt{d_i d_j}$ scaling prevents high-degree hubs from dominating the aggregation.

Relationship to SAGE. When mean aggregation is employed and all gating coefficients vanish ($\mathbf{g}_{ji} = 0$), the perturbation $\mathbf{u}_{ji}$ disappears and the message reduces to the neighbor identity $\mathbf{m}_{j\to i} = \mathbf{h}_j$, rendering the update functionally equivalent to GraphSAGE [9]. However, GP-GNN departs from this paradigm by treating the neighbor identity as a stable geometric anchor for the message. While GraphSAGE mixes undifferentiated features through static linear transformations, GP-GNN utilizes the local feature geometry to resolve signals into specific directional components. By explicitly partitioning the interaction into signed projections and orthogonal residuals, the model autonomously balances feature consensus with structural contrast. This capability allows it to navigate complex heterophilic relationships where standard architectures typically collapse.

4.3 Backbone Architecture

GP-GNN follows a residual stacking strategy to enable deep feature extraction. Let d_{in} denote the input feature dimension, d the hidden dimension, and d_{out} the output dimension (e.g. number of classes/targets). Given the input features $\mathbf{x} \in \mathbb{R}^{d_{\mathrm{in}}}$, the forward pass is as follows:

(1) **Input projection.** $\mathbf{h}^{(0)} = \phi(\mathbf{W}_{\text{sem}}\mathbf{x})$, where $\mathbf{W}_{\text{sem}} \in \mathbb{R}^{d \times d_{\text{in}}}$ is a learnable projection matrix and ϕ is an activation function.
(2) **Residual stacking.** For $l = 0, \ldots, L-1$,

$$\mathbf{h}^{(l+1)} = \phi\Big(\text{GP-Conv}\Big(\mathbf{h}^{(l)}\Big)\Big) + \mathbf{h}^{(l)}.$$

(3) **Output head.** $\mathbf{z} = \mathbf{W}_{\text{out}}\mathbf{h}^{(L)}$, where $\mathbf{W}_{\text{out}} \in \mathbb{R}^{d_{\text{out}} \times d}$ is a learnable output matrix producing logits $\mathbf{z}$.

In summary, the architecture operates by iteratively perturbing node identities with bilateral geometric signals, effectively steering the node embeddings toward regions of the feature space that best separate heterogeneous classes while maintaining stable gradient flow via the residual connections.

5 Experimental Setup

Datasets and Tasks. We evaluated GP-GNN on ten diverse node classification benchmarks spanning both heterophilic and homophilic regimes. For heterophilic evaluation, we utilized the benchmarks proposed by [26]. These include: *Roman Empire*, a syntactic role prediction task on a chain-like dependency graph; *Amazon Ratings*, a co-purchasing network where product ratings are predicted from review-based embeddings; *Minesweeper*, an 100×100 grid graph that requires predicting mines based on neighbor counts; *Tolokers*, a dense crowdsourcing network for identifying banned users; and *Questions*, a medical question-answering network. To ensure balanced evaluation across graph types, we also include standard homophilic citation networks, *Cora*, *CiteSeer*, and *PubMed* [37], along with *Amazon Computers* and *Amazon Photo* co-purchase graphs [28]. Together, these datasets provide a rigorous testing ground with varying graph properties and class distributions.

We characterized the graph regimes using edge homophily (h) and adjusted homophily (h_{adj}), as reported in Table 1. Edge homophily represents the fraction of edges connecting nodes of the same class:

$$h = \frac{|\{(i,j) \in \mathcal{E} : y_i = y_j\}|}{|\mathcal{E}|}. \tag{12}$$

where $h \approx 1$ indicates strong homophily and $h \approx 0$ reflects strong heterophily.

Because h is sensitive to class imbalance, we also reported adjusted homophily [24] to account for expected homophily under random configurations:

$$h_{adj} = \frac{h - \sum_{k=1}^{C} \bar{p}(k)^2}{1 - \sum_{k=1}^{C} \bar{p}(k)^2}, \text{ and } \bar{p}(k) = \frac{\sum_{i:y_i=k} d(i)}{|\mathcal{E}|} \tag{13}$$

where C denotes the number of classes and $\bar{p}(\cdot)$ is the empirical degree-weighted class distribution.

Under adjusted homophily (h_{adj}), values greater than, equal to, or less than zero indicate label assortativity that is, respectively, higher than, independent of, or lower than expected by chance. For datasets like *Minesweeper* and *Questions*, h_{adj} is near zero despite high raw homophily (h), a discrepancy explained by their low normalized Shannon entropy (H). For instance, the extreme imbalance in *Questions* ($H = 0.19$) artificially inflates h without providing structural signal. We categorize the first five datasets in Table 1 as heterophilic because they lack meaningful positive assortativity. In this regime, simple neighborhood smoothing is insufficient, necessitating models that leverage feature-level contrast.

Table 1: Descriptive statistics for the benchmark datasets. We report the number of nodes (N), edges (E), average degree (d_{avg}), feature dimension (f), and number of classes (C). Edge homophily (h), adjusted homophily (h_{adj}), and normalized Shannon entropy (H) are provided to characterize the graph regimes and class balance.

Dataset	N	E	d_{avg}	f	C	h	h_{adj}	H
Heterophilic benchmarks								
Roman Empire	22,662	65,854	2.91	300	18	0.05	-0.05	0.91
Amz. Ratings	24,492	186,100	7.60	300	5	0.38	0.14	0.88
Minesweeper	10,000	78,804	7.88	7	2	0.68	0.01	0.72
Tolokers	11,758	1,038,000	88.28	10	2	0.60	0.09	0.76
Questions	48,921	307,080	6.28	301	2	0.84	0.02	0.19
Homophilic benchmarks								
Cora	2,708	10,556	3.90	1,433	7	0.81	0.77	0.94
CiteSeer	3,327	9,104	2.74	3,703	6	0.74	0.67	0.98
PubMed	19,717	88,648	4.50	500	3	0.80	0.69	0.97
Amz. Computers	13,752	491,722	35.76	767	10	0.78	0.68	0.81
Amz. Photo	7,650	238,162	31.13	745	8	0.83	0.79	0.93

Baselines. We compared GP-GNN against ten representative architectures, categorized by their structural assumptions. The **MLP** baseline treats nodes as independent and identically distributed samples, making it structure-agnostic. To evaluate the benefit of simple topological augmentation, we implemented **MLP+adj**, which concatenates raw node features $\mathbf{h}_i$ with their 1-hop neighborhood mean $\frac{1}{|\mathcal{N}(i)|}\sum_{j \in \mathcal{N}(i)} \mathbf{h}_j$ before passing them through the MLP backbone. This approach, inspired by the LINK architecture [38], provides localized structural context without the iterative smoothing characteristic of traditional GNNs.

We implemented the foundational message-passing architectures **GCN** [13] and **GraphSAGE** [9] using the shared backbone in 4.3; for GraphSAGE, we utilized mean aggregation without neighbor sampling. We also included **GAT** [32], which uses additive attention to weigh neighbor contributions. For a more powerful attentive baseline, we implemented a **Graph Transformer (GT)** variant that employs scaled dot-product attention [29]. Our GT implementation is restricted to neighborhood-restricted message passing, where each node attends exclusively to its immediate neighbors. To ensure architectural parity, we omitted the traditional position-wise feed-forward block in the GT and applied a common backbone skeleton across GCN, GraphSAGE, GAT, and GT. This skeleton consists of an initial input projection followed by L layers of the specific operation augmented with additive skip connections. Notably, the GT was the only baseline where we incorporated Layer Normalization to maintain training stability.

We included several models engineered specifically for non-homophilic regimes. **LINKX** [16] processes node features and adjacency matrix through separate MLP branches before fusion, allowing it to capture structural patterns that standard aggregators might smooth over. **H$_2$GCN** [40] utilizes ego- and neighbor-embedding

Table 2: Main results for node classification. We report mean Macro-F1 (%) over 50 random splits. Nominal ranks: first, second, third. All GP-GNN results are significantly different from baselines at the $\alpha = 0.05$ level (paired bootstrap test) unless marked with *, which denotes a statistical tie.

Dataset	MLP	MLP+adj	GCN	GAT	SAGE	GT	LINKX	H_2GCN	FSGNN	GPR-GNN	GP-GNN
Heterophilic Datasets											
Roman Empire	54.25 ± 0.61	70.63 ± 0.83	81.49 ± 0.71	82.15 ± 0.67	85.44 ± 0.62	87.17 ± 0.62	53.75 ± 0.65	79.21 ± 0.51	74.13 ± 0.83	66.82 ± 0.66	87.83 ± 0.55
Amz. Ratings	42.70 ± 0.73	46.21 ± 0.78	45.43 ± 0.86	46.39 ± 0.69	47.58 ± 0.83	46.04 ± 0.84	46.82 ± 0.92	45.75 ± 0.86	46.33 ± 0.83	47.15 ± 0.77	47.81 ± 0.78
Minesweeper	47.55 ± 2.51	73.32 ± 1.24	78.29 ± 0.93	82.04 ± 0.72	82.98 ± 0.78*	80.61 ± 1.99	50.08 ± 1.07	77.25 ± 0.88	76.08 ± 1.02	73.78 ± 1.27	83.05 ± 0.86
Tolokers	59.64 ± 1.79	65.00 ± 2.91	71.64 ± 1.17	70.93 ± 1.66	70.74 ± 1.32	71.44 ± 1.05	66.00 ± 1.61	69.62 ± 1.59	71.20 ± 0.97	62.70 ± 3.06	72.25 ± 1.11
Questions	60.79 ± 1.18	59.47 ± 1.10	60.74 ± 1.15	60.86 ± 1.07	60.11 ± 1.04	59.68 ± 1.44	60.01 ± 1.25	60.75 ± 1.02	60.50 ± 1.07	60.76 ± 1.06	61.56 ± 0.96
Homophilic Datasets											
Cora	73.12 ± 1.76	84.57 ± 1.32	85.75 ± 1.55*	86.58 ± 1.43	85.56 ± 1.62*	85.71 ± 1.44*	82.20 ± 1.27	86.12 ± 1.38	85.71 ± 1.42*	86.36 ± 1.42	85.55 ± 1.61
CiteSeer	68.68 ± 1.62	71.63 ± 1.55	71.94 ± 1.64*	72.00 ± 1.66*	72.07 ± 1.53*	70.11 ± 1.66	69.36 ± 1.75	71.36 ± 1.81	70.32 ± 1.71	71.70 ± 1.65	72.20 ± 1.64
PubMed	87.37 ± 0.48	87.64 ± 0.43	88.57 ± 0.45	89.07 ± 0.42	88.75 ± 0.44	88.56 ± 0.47	86.84 ± 0.49	88.79 ± 0.38	88.51 ± 0.44	88.63 ± 0.45	88.92 ± 0.39
Amz. Computers	82.92 ± 0.92	89.06 ± 0.64	91.82 ± 0.57	91.94 ± 0.54	91.08 ± 0.60	91.24 ± 0.60	90.40 ± 0.62	91.27 ± 0.49	91.49 ± 0.54*	90.53 ± 0.67	91.46 ± 0.62
Amz. Photo	90.59 ± 0.67	94.21 ± 0.57	94.98 ± 0.58*	94.96 ± 0.56*	94.74 ± 0.64	94.88 ± 0.64*	94.31 ± 0.56	95.00 ± 0.54*	94.66 ± 0.63	94.74 ± 0.53	94.95 ± 0.49

separation combined with higher-order neighborhood concatenation to preserve structural information. **FSGNN** [20] is a decoupled GNN that performs fixed-scale aggregations followed by learnable hop-wise selection. Finally, **GPR-GNN** [3] employs a decoupled architecture where an MLP transforms features into logits, which are then processed by a learnable Generalized PageRank filter. Because these models rely on specialized concatenation or polynomial propagation, we did not utilize our additive skip-connection backbone for this group.

We used Optuna with the Tree-structured Parzen Estimator (TPE) sampler (100 trials per model-dataset pair) for hyperparameter optimization to identify the best configuration of each model. The search space includes hidden dimension $\in \{32, 64, 128, 256, 512\}$, the number of layers $\{1, \ldots, 5\}$, activation function $\{\text{ReLU}, \text{GeLU}\}$, dropout rates $\{0.0, 0.2, 0.4, 0.6\}$, learning rates $\{10^{-4}, 10^{-3}, 10^{-2}\}$, and weight decay $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. For GP-GNN, we implemented a hierarchical search to capture dependencies between structural components. The optimizer first selects an aggregation function $\in \{\text{GCN}, \text{Mean}\}$ and a geometric scaling option. Following these selections, the perturbation intensity β is determined via conditional branching. In the adaptive regime, β is tuned over $\{0.01, 0.05, 0.1, 0.2, 0.5\}$ to provide manual damping of the geometric signal. Alternatively, in the constrained regime, β is fixed at 1.0 if the chosen scaling method inherently manages signal intensity. This branching allows the model to adapt its perturbation strength to the dataset's specific homophily or heterophily levels. Full search spaces and resulting optimal hyperparameters are detailed in Appendix A.

All models were trained using the AdamW optimizer with a gradient clipping max norm of 1.0 and an early stopping patience of 100 epochs. The benchmark models were implemented in PyTorch Geometric [6] and trained on NVIDIA H100 GPUs using mixed-precision (AMP).

6 Main results

Table 2 summarizes the performance of GP-GNN across the ten benchmarks. We report the mean and standard deviation of the test Macro-F1 score over 50 random splits, a metric chosen to ensure robust evaluation on datasets with significant label imbalance. Statistical significance is assessed using a paired bootstrap test; we consider two models to be different with α significance if the $(1-\alpha)\%$ confidence interval of their paired performance differences excludes zero. In Table 2, an asterisk (*) denotes a statistical tie to GP-GNN, while the absence of a marker indicates that the performance of GP-GNN is significantly different at the $\alpha = 0.05$ level. For completeness, we provide a comparison of the model performance using accuracy in Appendix B, where GP-GNN demonstrates consistent superiority.

GP-GNN achieved state-of-the-art performance across heterophilic regimes, securing the top nominal rank on all five benchmarks. Notably, on *Roman Empire* (87.83%), *Tolokers* (72.25%), and *Questions* (61.56%), GP-GNN maintained a decisive statistical lead over every baseline. On *Roman Empire*, GP-GNN surpassed the Graph Transformer (GT) by 0.66% absolute F1. This suggests that the use of reciprocal projections and residuals captures structural dependencies more effectively than the content-based multi-head attention mechanisms in GT, which can struggle with specific structural patterns in heterophilic graphs. On *Minesweeper*, while GraphSAGE (82.98%) remained a competitive tie, GP-GNN still achieved the highest nominal mean. This demonstrates its ability to provide a nuanced inductive bias for dense graphs where traditional message-passing often fails to resolve informative signals.

GP-GNN remains highly competitive across the entire homophily-heterophily spectrum, unlike many specialized models that sacrifice performance on homophilic benchmarks. On *CiteSeer* (72.20%), our model is nominally first, although it remains statistically tied to GAT and GraphSAGE. On the *Amazon* datasets, GP-GNN's performance is statistically indistinguishable from top-performing baselines like H_2GCN, GCN, and FSGNN. This stability suggests that the symmetric gating mechanism adapts dynamically: in homophilic settings, the gates prioritize feature consensus, yet retain the flexibility to distinguish signals at local class boundaries. While spectral methods like GPR-GNN rely on global filters, they often overlook localized nuances inherent in heterophilic structures. Similarly, while H_2GCN utilizes neighbor concatenation, it lacks the refined gating necessary to isolate informative signals from structural noise. In contrast, GP-GNN's bilateral decomposition adaptively transforms

signals based on the mutual geometric context of node pairs, providing a more effective inductive bias for diverse graph regimes, ranging from high homophily to extreme heterophily.

6.1 Analysis of Geometric Adaptation

To verify that the architectural components of GP-GNN adapt systematically to the graph structure, we analyzed the relationship between graph homophily and the model's relative message composition. We define the *Relative Contribution Share* ($S_{ji}^{(l,k)}$) to quantify this adaptation. Rather than relying solely on the gating coefficients $\mathbf{g}_{ji}$, this metric measures the L_2 norm of each gated component to reveal how the model partitions its total perturbation budget across the four bilateral components of alignment and contrast.

For a layer l and an edge (j, i), let $\mathbf{q}_k^{(l)}$ denote one of the four bilateral geometric components, where $k \in \{\mathbf{p}_{j\to i}, \mathbf{r}_{j\to i}, \mathbf{p}_{i\to j}, \mathbf{r}_{i\to j}\}$. We define $g_k^{(l)}$ as the corresponding scalar gate from the vector $\mathbf{g}_{ji}^{(l)}$ in layer l (Eq. 6). The relative share $S_{ji}^{(l,k)}$ for the component k in layer l is calculated as:

$$S_{ji}^{(l,k)} = \frac{\|g_k^{(l)} \mathbf{q}_k^{(l)}\|_2}{\sum_{k' \in \{p_{j\to i}, r_{j\to i}, p_{i\to j}, r_{i\to j}\}} \|g_{k'}^{(l)} \mathbf{q}_{k'}^{(l)}\|_2} \tag{14}$$

By focusing on the relative share of each component, the metric captures the functional shift in the message construction of the model across diverse graph regimes. This analysis allows us to observe how GP-GNN reallocates its representational capacity by transitioning from a reliance on feature consensus in homophilic environments to the extraction of local contrast in heterophilic ones.

Global Adaptive Trends. Figure 2 illustrates the relative contribution share in the final layer across ten benchmarks, highlighting how GP-GNN systematically reallocates its representational capacity. We evaluated this adaptation against two metrics: standard homophily (h), which measures edge-wise label consistency, and adjusted homophily (h_{adj}), which accounts for the expected similarity due to class imbalance. In both cases, we can observe a distinct architectural shift: the source-centric residual component ($\mathbf{r}_{j\to i}$, red dashed line) exhibits a negative slope ($\beta = -0.42$ for h), becoming the dominant message channel in heterophilic regimes. This suggests that the model prioritizes orthogonal feature variance to preserve local contrast where label consistency is low. In contrast, the source-centric projection component ($\mathbf{p}_{j\to i}$, blue dashed line) shows a positive slope ($\beta = 0.48$ for h), confirming that the model exploits feature consensus as graph assortativity increases. Although these trends are most pronounced under standard homophily (h), they remain consistent under adjusted homophily (h_{adj}), although with more moderate slopes ($\beta = -0.26$ for residuals). The target-centric components, $\mathbf{p}_{i\to j}$ (green) and $\mathbf{r}_{i\to j}$ (purple), remain comparatively stable, serving as a consistent structural prior. These regression slopes demonstrate that GP-GNN learns a geometric strategy that generalizes from disassortative graphs to highly homophilic networks.

Local Spatial Adaptation. This architectural adaptation is visually reflected in the localized behavior within 1-hop egonets (Figure 3). In homophilic environments such as PubMed (Figure 3, left), the message flow is dominated by projection components, indicating a priority for feature alignment to reinforce local consensus. The uniform intensity across neighbors suggests that the model utilizes broad agreement to stabilize class identities through smoothing.

In contrast, in the heterophilic Roman Empire dataset (Figure 3, right), the model reallocates the representational weight to residual components. As visualized, the projection share diminishes while residual gates selectively highlight specific in-edges to preserve node distinctness. This shift confirms that symmetric joint gating effectively distinguishes between semantic similarity and structural irregularity, using orthogonal residuals to prevent feature collapse and indiscriminate averaging. A detailed analysis of how these geometric priorities evolve across successive layers is provided in Appendix C, demonstrating the model's capacity for multi-stage signal refinement.

6.2 Ablation Study

To evaluate the individual contributions of the geometric components, we conducted a systematic ablation study across all benchmarks. We performed full hyperparameter tuning for each variant using the same optimization budget as the full model. The results, summarized in Table 3, compare the full GP-GNN architecture against three restricted variants:

- **(1) No Bidir.:** Removed target-centric components $\mathbf{p}_{\mathbf{i}\to\mathbf{j}}$ and $\mathbf{r}_{\mathbf{i}\to\mathbf{j}}$, reducing the message to a unidirectional flow from neighbor to target.
- **(2) No Resid.:** Disabled residuals ($g_{r_{j\to i}}, g_{r_{i\to j}} = 0$), forcing exclusive reliance on semantic alignment via projections.
- **(3) No Proj.:** Disabled projections ($g_{p_{j\to i}}, g_{p_{i\to j}} = 0$), ignoring collinear feature components.

Ablation findings confirmed the necessity of each geometric component. We included GraphSAGE (SAGE) as a baseline, representing the architecture when perturbations were deactivated. Removing bidirectional decomposition in Variant (1) caused a significant drop on *Questions*, indicating that target-centric context is particularly beneficial in graphs with extreme class imbalance and low entropy distributions, where resolving structural dependencies requires bilateral geometric perspective. On most other benchmarks, the unidirectional variant performs comparably, suggesting that bidirectionality provides selective rather than universal benefit. Variant (2) demonstrated that orthogonal residuals are critical for heterophily; its removal led to significant declines in *Roman Empire* and *Tolokers*. Similarly, the drop in Variant (3) on *Questions* (61.56 $\rightarrow$ 60.44) confirms that projections are necessary to filter informative signals from structural noise. Interestingly, on several homophilic benchmarks (e.g., *CiteSeer*, *PubMed*), ablated variants remained statistically tied (*) with the full model. This suggests that GP-GNN's symmetric gating dynamically simplifies its operations in homophilic regimes, utilizing its full geometric bilateral decomposition when heterophilic signals are present. GP-GNN significantly outperformed SAGE across almost all heterophilic datasets, validating the superior inductive bias of our geometric gating.

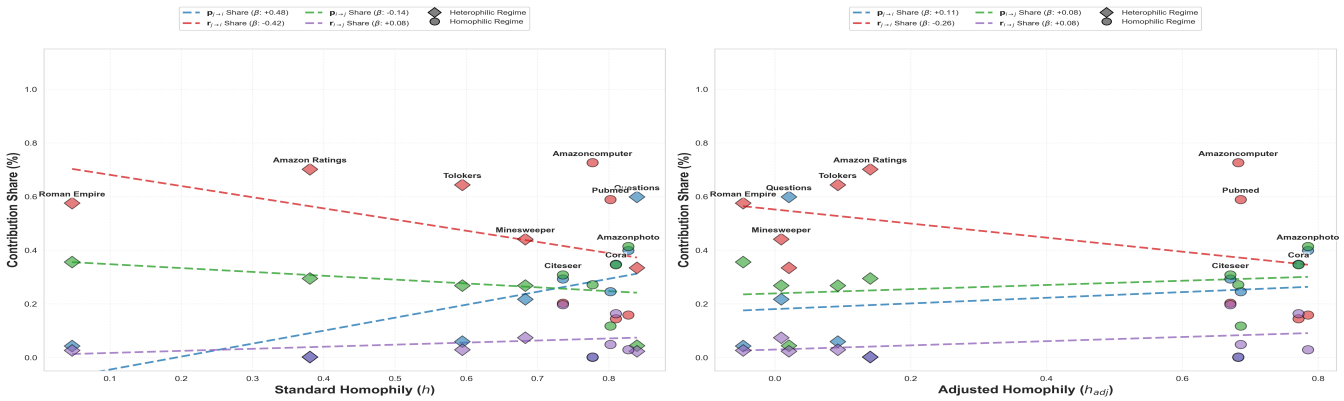

Figure 2: Geometric strategy adaptation. Comparison of the relative contribution shares across the homophily spectrum (h, h_{adj}). Data points reflect the average share in the final layer across all edges. The blue ($\mathbf{p}_{j\to i}$) and red ($\mathbf{r}_{j\to i}$) lines denote source-centric projection and residual shares, while green ($\mathbf{p}_{i\to j}$) and purple ($\mathbf{r}_{i\to j}$) show target-centric components. The plot shows that GP-GNN adaptively shift from feature-based consensus in homophilic regimes to structural contrast in heterophilic ones.

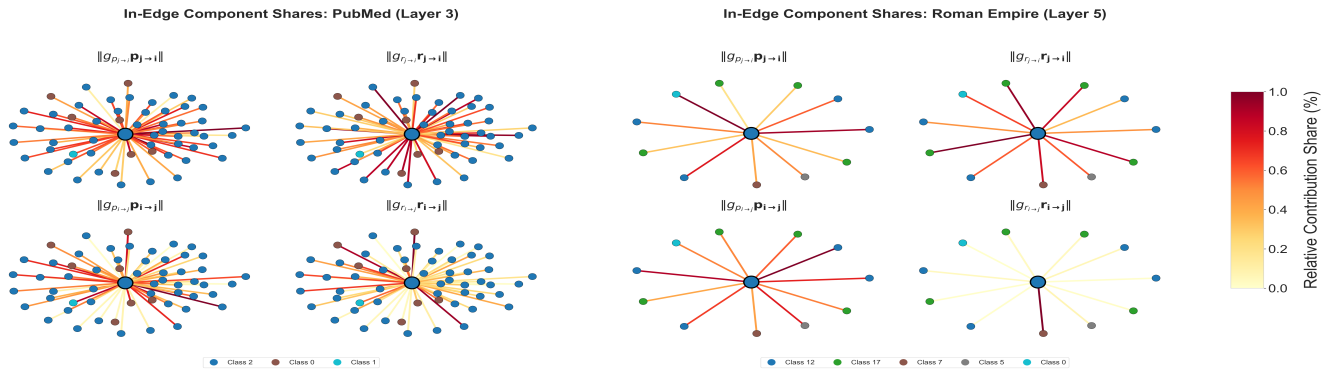

Figure 3: Spatial analysis of the relative contribution share ($S_{ji}^{(l,k)}$) for incoming edges within a 1-hop egonet. Left (*PubMed*): Feature consensus is achieved through dominant projection components ($k \in \{\mathbf{p}_{j\to i}, \mathbf{p}_{i\to j}\}$). Right (*Roman Empire*): The model reallocates representational capacity to residual components ($k \in \{\mathbf{r}_{j\to i}, \mathbf{r}_{i\to j}\}$) to maintain node distinctness in heterophilic regimes.

Table 3: Ablation study of GP-GNN components. We report mean Macro-F1 (%) over 50 random splits. All ablated variants significantly differ from the Full GP-GNN ($\alpha = 0.05$) via a paired bootstrap test except where marked with *, which denotes a statistical tie.

				Heterophilic Benchmarks					Homophilic Benchmarks				
Variant	**Proj.**	**Resid.**	**Bidir.**	**Roman Empire**	**Amz. Ratings**	**Minesweeper**	**Tolokers**	**Questions**	**Cora**	**CiteSeer**	**PubMed**	**Amz. Computers**	**Amz. Photo**
Full GP-GNN	✓	✓	✓	**87.83**	**47.81**	**83.05**	**72.25**	**61.56**	**85.55**	**72.20**	**88.92**	**91.46**	**94.95**
(1) No Bidir.	✓	✓	×	87.76*	47.72*	82.94*	72.54*	60.60	85.54*	71.86	89.04	91.44*	94.88*
(2) No Resid.	✓	×	✓	86.99	47.66*	82.95*	71.69	61.36*	85.88	72.01*	88.97*	91.37*	94.81
(3) No Proj.	×	✓	✓	87.36	47.70*	83.18*	72.62*	60.44	85.88	71.89*	88.96*	91.15	94.88*
SAGE	×	×	×	85.44	47.58	82.98*	70.74	60.11	85.56*	72.07*	88.75	91.08	94.74

7 Conclusion

We introduce GP-GNN, a message-passing framework that replaces indiscriminate neighborhood averaging with controlled geometric perturbations of node identities. By explicitly decomposing edge interactions into reciprocal projections and orthogonal residuals, GP-GNN provides a flexible mechanism for both feature alignment and the preservation of node distinctness. This architecture ensures robustness across the homophily-heterophily spectrum.

Our experimental results demonstrate that GP-GNN consistently matches or outperforms state-of-the-art methods. Our analysis confirms that the model adaptively manages these geometric components, prioritizing feature consensus in homophilic regions while leveraging orthogonal residuals to prevent feature collapse in heterophilic graphs. Spatial visualizations further verify that GP-Conv maintains critical class separations that are typically lost to over-smoothing in standard architectures.

Real-world graphs sometimes exhibit mixed homophily-heterophily structure — a fundamental challenge for graph foundation models [11, 17], which are typically pretrained under implicit homophilic assumptions. GP-Conv's edge-level geometric decomposition makes no global assumption about graph regime, adapting locally at each edge. This property positions GP-Conv as a candidate geometric primitive for graph foundation models that must generalize across diverse and structurally heterogeneous graph domains.

A promising direction is integrating GP-Conv as a geometry-aware message passing backbone for graph foundation models, where edge-level adaptivity across homophilic and heterophilic structures is essential for generalizing across diverse graph domains without retraining. Future research also focuses on extending the projection-residual decomposition to handle multi-relational edge attributes. Incorporating weighted graphs represents another key developmental path, as edge weights can dynamically scale the magnitude of geometric perturbations to reflect varying connection intensities. By reconciling feature-space orientation with explicit topological information, a more general geometric basis for graph learning can be developed.

Acknowledgements

The authors thank the Financial Wellness Lab and its industry partners for their support. The last author acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [Discovery Grant RGPIN-2020-07114]. This research was undertaken, in part, thanks to funding from the Canada Research Chairs program [CRC-2024-00192]. This work was enabled in part by support provided by Compute Ontario (computeontario.ca), Calcul Québec (calculquebec.ca), and the Digital Research Alliance of Canada (alliancecan.ca).

References

[1] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. 2021. Beyond Low-frequency Information in Graph Convolutional Networks. In *AAAI*. AAAI Press.

[2] Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Lió, and Michael Bronstein. 2022. Neural Sheaf Diffusion: A Topological Perspective on Heterophily and Oversmoothing in GNNs. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 18527–18541. https://proceedings.neurips.cc/paper_files/paper/2022/file/75c45fca2aa416ada062b26cc4fb7641-Paper-Conference.pdf

[3] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. arXiv:2006.07988 [cs.LG] https://arxiv.org/abs/2006.07988

[4] Tyler Derr, Yao Ma, and Jiliang Tang. 2018. Signed Graph Convolutional Networks. In *2018 IEEE International Conference on Data Mining (ICDM)*. 929–934. doi:10.1109/ICDM.2018.00113

[5] Lun Du, Xiaozhou Shi, Qiang Fu, Xiaojun Ma, Hengyu Liu, Shi Han, and Dongmei Zhang. 2022. GBK-GNN: Gated Bi-Kernel Graph Neural Networks for Modeling Both Homophily and Heterophily. arXiv:2110.15777 [cs.LG] https://arxiv.org/abs/2110.15777

[6] Matthias Fey and Jan Eric Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. arXiv:1903.02428 [cs.LG] https://arxiv.org/abs/1903.02428

[7] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for Quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) *(ICML'17)*. JMLR.org, 1263–1272.

[8] W.L. Hamilton. 2020. *Graph Representation Learning*. Morgan & Claypool Publishers.

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.

[10] Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. 2021. BernNet: learning arbitrary graph spectral filters via bernstein approximation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NIPS '21)*. Curran Associates Inc., Red Hook, NY, USA, Article 1091, 13 pages.

[11] Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. 2025. UniGraph: Learning a Unified Cross-Domain Foundation Model for Text-Attributed Graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1* (Toronto ON, Canada) *(KDD '25)*. Association for Computing Machinery, New York, NY, USA, 448–459. doi:10.1145/3690624.3709277

[12] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. 2019. Signed Graph Attention Networks. In *Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions*. Springer International Publishing, Cham, 566–577.

[13] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs.LG] https://arxiv.org/abs/1609.02907

[14] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence* (New Orleans, Louisiana, USA) *(AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, Article 433, 8 pages.

[15] Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. 2022. Finding Global Homophily in Graph Neural Networks When Meeting Heterophily. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 13242–13256.

[16] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser-Nam Lim. 2021. Large scale learning on non-homophilous graphs: new benchmarks and strong simple methods. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NIPS '21)*. Curran Associates Inc., Red Hook, NY, USA, Article 1598, 16 pages.

[17] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2024. One For All: Towards Training One Graph Model For All Classification Tasks. In *International Conference on Learning Representations*, B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (Eds.), Vol. 2024. 20188–20210.

[18] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. 2021. Is Heterophily A Real Nightmare For Graph Neural Networks To Do Node Classification? arXiv:2109.05641 [cs.LG] https://arxiv.org/abs/2109.05641

[19] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. 2023. Is Homophily a Necessity for Graph Neural Networks? arXiv:2106.06134 [cs.LG] https://arxiv.org/abs/2106.06134

[20] Sunil Kumar Maurya, Xin Liu, and Tsuyoshi Murata. 2022. Simplifying approach to node classification in Graph Neural Networks. *Journal of Computational Science* 62 (2022), 101695. doi:10.1016/j.jocs.2022.101695

[21] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. 2001. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology* 27 (2001), 415–444.

[22] Kenta Oono and Taiji Suzuki. 2021. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. arXiv:1905.10947 [cs.LG] https://arxiv.org/abs/1905.10947

[23] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-GCN: Geometric Graph Convolutional Networks. arXiv:2002.05287 [cs.LG] https://arxiv.org/abs/2002.05287

[24] Oleg Platonov, Denis Kuznedelev, Artem Babenko, and Liudmila Prokhorenkova. 2023. Characterizing graph datasets for node classification: homophily-heterophily dichotomy and beyond. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 25, 26 pages.

[25] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. 2024. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? arXiv:2302.11640 [cs.LG]

[26] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. 2024. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? arXiv:2302.11640 [cs.LG] https://arxiv.org/abs/2302.11640

[27] Emanuele Rossi, Bertrand Charpentier, Francesco Di Giovanni, Fabrizio Frasca, Stephan Günnemann, and Michael Bronstein. 2023. Edge Directionality Improves Learning on Heterophilic Graphs. arXiv:2305.10498 [cs.LG] https://arxiv.org/abs/2305.10498
[28] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2019. Pitfalls of Graph Neural Network Evaluation. arXiv:1811.05868 https://arxiv.org/abs/1811.05868
[29] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. 2021. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. arXiv:2009.03509 https://arxiv.org/abs/2009.03509
[30] Rahul Singh and Yongxin Chen. 2022. Signed Graph Neural Networks: A Frequency Perspective. arXiv:2208.07323 [cs.LG] https://arxiv.org/abs/2208.07323
[31] An Tong, Bochao Chen, Zhe Wang, Jiawei Gao, and Chi Kin Lam. 2025. GDFGAT: Graph attention network based on feature difference weight assignment for telecom fraud detection. *PLoS One* 20, 5 (2025), e0322004.
[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. arXiv:1710.10903 [stat.ML] https://arxiv.org/abs/1710.10903
[33] Xiyuan Wang and Muhan Zhang. 2022. How Powerful are Spectral Graph Neural Networks. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 23341–23362.
[34] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I. Weidele, Claudio Bellei, Tom Robinson, and Charles E. Leiserson. 2019. Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. arXiv:1908.02591 [cs.SI] https://arxiv.org/abs/1908.02591
[35] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (Jan. 2021), 4–24. doi:10.1109/tnnls.2020.2978386
[36] Liang Yang, Weihang Peng, Wenmiao Zhou, Bingxin Niu, Junhua Gu, Chuan Wang, Yuanfang Guo, Dongxiao He, and Xiaochun Cao. 2022. Difference Residual Graph Neural Networks. In *Proceedings of the 30th ACM International Conference on Multimedia* (Lisboa, Portugal) *(MM '22)*. Association for Computing Machinery, New York, NY, USA, 3356–3364. doi:10.1145/3503161.3548111
[37] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. arXiv:1603.08861 https://arxiv.org/abs/1603.08861
[38] Elena Zheleva and Lise Getoor. 2009. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th International Conference on World Wide Web* (Madrid, Spain) *(WWW '09)*. Association for Computing Machinery, New York, NY, USA, 531–540. doi:10.1145/1526709.1526781
[39] Jiong Zhu, Ryan A. Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K. Ahmed, and Danai Koutra. 2021. Graph Neural Networks with Heterophily. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 12 (May 2021), 11168–11176. doi:10.1609/aaai.v35i12.17332
[40] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond homophily in graph neural networks: current limitations and effective designs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 653, 12 pages.

A Implementation Details and Hyperparameter Search

A.1 Hyperparameter search range

To ensure a fair comparison, all models were optimized using the Optuna framework with a search budget of 100 trials per model-dataset pair. We employed the Tree-structured Parzen Estimator (TPE) sampler to efficiently navigate the hyperparameter space. All models were trained for a maximum of 10,000 epochs using the AdamW optimizer, with an early stopping patience of 100 epochs based on validation accuracy. The comprehensive search space for both our proposed GP-GNN and the baseline models is detailed in Table 4.

A.2 Model hyperparameters

We report the specific hyperparameter configurations that yielded the optimal performance for GP-GNN and all baseline models. Following the TPE-based optimization process described in Section A.1, the configurations for learning rate (lr), weight decay (L_2), hidden dimensionality, and model-specific architectural parameters, such as the number of layers, attention heads, or propagation steps, were selected based on the highest validation macro F1-score. Table 5 details these configurations for heterophilic datasets, while Table 6 covers homophilic benchmarks.

B Performance comparison on accuracy

While our primary analysis focuses on Macro-F1 to ensure robust evaluation under class imbalance, we provide accuracy results in Table 7 for completeness. These results were obtained using the same hyperparameter configurations selected via validation Macro-F1.

GP-GNN maintains a strong performance profile under this metric, securing the top nominal rank on three of the five heterophilic benchmarks (*Roman Empire*, *Amazon Ratings*, and *Tolokers*). On *Questions*, GP-GNN achieves 97.22%, remaining statistically tied with the top-performing MLP baseline. This is particularly noteworthy given the dataset's extreme imbalance ($H = 0.19$, Table 1); whereas most GNNs struggle to significantly outperform majority-class baselines on such skewed data, GP-GNN remains highly competitive while simultaneously providing superior minority-class resolution (as seen in Table 2).

In homophilic settings, the accuracy results mirror the trends observed in the main text. GP-GNN remains statistically indistinguishable from, or superior to, specialized homophilic baselines like GAT and GCN on datasets such as *CiteSeer* and *Amazon Photo*. This cross-metric consistency confirms that GP-GNN's bilateral decomposition does not merely exploit the specificities of the Macro-F1 objective, but rather learns a fundamentally more robust representation of the graph signal that generalizes across different evaluation targets.

C Layer-wise geometric signatures

To investigate the internal mechanics of GP-GNN, we analyzed layer-wise geometric signatures for the homophilic *PubMed* and heterophilic *Roman Empire* datasets. Figure 4 reveals that GP-GNN functions as a hierarchical geometric filter, adaptively refocusing its attention across depths. Notably, the model learns to distinctively weight source-to-target ($j \to i$) flow against reciprocal target-to-source ($i \to j$) components.

In the homophilic regime (*PubMed*), the model maintains a stable reliance on the source-to-target projection ($\mathbf{p}_{j\to i}$), prioritizing feature alignment to identify shared class attributes. Conversely, the heterophilic regime (*Roman Empire*) exhibits a strategic evolution: while initial layers capture a broad mix of components, deeper layers increasingly prioritize the source-to-target residual ($\mathbf{r}_{j\to i}$). This suggests that as structural context integrates, deeper layers transition into high-pass filters to emphasize discriminative signals. Such multi-stage refinement confirms that the bilateral gating mechanism enables complex filtering by independently modulating the four geometric channels to suit the underlying graph regime.

Table 4: Hyperparameter search spaces for GP-GNN and baseline models.

Category	Hyperparameter	Search Space
Shared	Hidden Dimension	$\{32, 64, 128, 256, 512\}$
	Dropout	$\{0.0, 0.2, 0.4, 0.6\}$
	Learning Rate	$\{10^{-4}, 10^{-3}, 10^{-2}\}$
	Weight Decay (L_2)	$\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$
	Activation	{ReLU, GeLU}
GP-GNN	Scale Mode	{Full Structural, Refined}
	Perturbation Scale (β)	$\{1.0\}$ (Fixed) or $\{0.01, 0.05, 0.1, 0.2, 0.5\}$ (Refined)
	Aggregation Function	{GCN, Mean}
	Number of Layers	$[1, 5]$
Baselines	Number of Layers	$[1, 5]$ (GCN, SAGE, MLP, MLP+adj, GT)
	GPR-GNN Prop. Steps (K)	$[1, 10]$ (with PPR initialization)
	H2GCN / FSGNN Hops	$[1, 3]$
	LINKX Fusion Layers	$[1, 5]$
	LINKX Node/Edge MLP Layers	$[1, 5]$ (Individual depth per MLP)
	GAT / GT Attention Heads	$\{2, 4, 8\}$
Training	Optimizer	AdamW (with Gradient Clipping 1.0)
	Early Stopping	Patience: 100 epochs
	Max Epochs	10,000

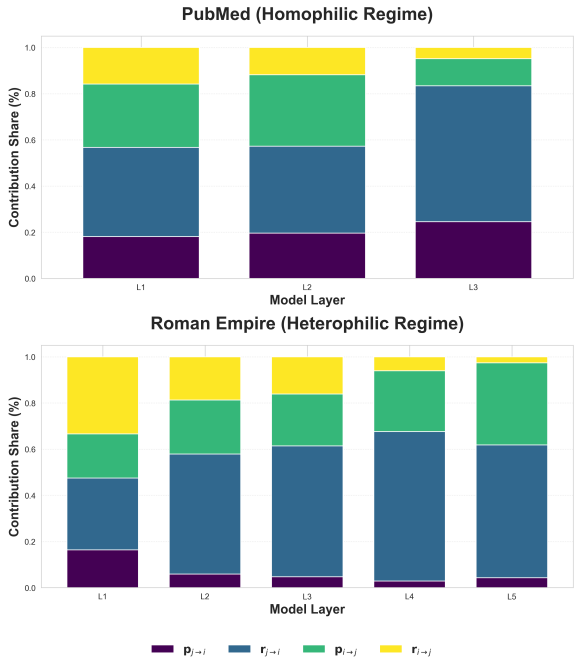

Figure 4: Layer-wise Geometric Signatures. Stacked bar charts illustrating the average contribution shares of the four bilateral geometric components across model layers. Each layer effectively provides a different filter focus.

Table 5: Optimal hyperparameter configurations for GP-GNN and baseline models across heterophilic benchmarks, determined via TPE-based search.

Dataset	Model	Optimal Configuration
Roman Empire	FSGNN	hops: 3, lr: 0.01, hidden: 128, L_2: 10^{-4}, dropout: 0.6, activation: GeLU
	GAT	layer: 4, lr: 10^{-3}, hidden: 128, L_2: 10^{-5}, dropout: 0.2, activation: ReLU, heads: 8
	GCN	layer: 5, lr: 10^{-2}, hidden: 128, L_2: 10^{-4}, dropout: 0.6, activation: GeLU
	GPRGNN	steps: 7, lr: 10^{-2}, hidden: 256, L_2: 10^{-2}, dropout: 0.6, activation: GeLU
	GraphSAGE	layer: 5, lr: 10^{-2}, hidden: 256, L_2: 10^{-2}, dropout: 0.6, activation: GeLU
	H_2GCN	hops: 3, lr: 10^{-2}, hidden: 128, L_2: 10^{-2}, dropout: 0.6, activation: ReLU
	LINKX	fusion layer: 1, edge layer: 1, node layer: 4, lr: 10^{-4}, hidden: 512, L_2: 10^{-3}, dropout: 0.2, activation: ReLU
	MLP	layer: 5, lr: 10^{-2}, hidden: 512, L_2: 10^{-2}, dropout: 0.6, activation: ReLU
	MLP+adj	layer: 1, lr: 10^{-2}, hidden: 256, L_2: 10^{-2}, dropout: 0.6, activation: GeLU
	GP-GNN	layer: 5, lr: 10^{-3}, hidden: 512, L_2: 10^{-4}, dropout: 0.6, activation: ReLU
	GT	layer: 5, lr: 10^{-2}, hidden: 128, L_2: 10^{-2}, dropout: 0.6, activation: ReLU, heads: 8
Amz. Ratings	FSGNN	hops: 3, lr: 10^{-2}, hidden: 512, L_2: 10^{-5}, dropout: 0.6, activation: ReLU
	GAT	layer: 4, lr: 10^{-2}, hidden: 512, L_2: 10^{-4}, dropout: 0.2, activation: GeLU, heads: 8
	GCN	layer: 2, lr: 10^{-2}, hidden: 512, L_2: 10^{-4}, dropout: 0.6, activation: ReLU
	GPRGNN	steps: 2, lr: 10^{-3}, hidden: 512, L_2: 10^{-2}, dropout: 0.6, activation: ReLU
	GraphSAGE	layer: 5, lr: 10^{-2}, hidden: 512, L_2: 10^{-2}, dropout: 0.6, activation: GeLU
	H_2GCN	hops: 1, lr: 10^{-2}, hidden: 512, L_2: 10^{-5}, dropout: 0.4, activation: ReLU
	LINKX	fusion layer: 1, edge layer: 2, node layer: 4, lr: 10^{-4}, hidden: 256, L_2: 10^{-2}, dropout: 0.2, activation: ReLU
	MLP	layer: 1, lr: 10^{-3}, hidden: 512, L_2: 10^{-2}, dropout: 0.6, activation: ReLU
	MLP+adj	layer: 1, lr: 10^{-3}, hidden: 512, L_2: 10^{-4}, dropout: 0.6, activation: ReLU
	GP-GNN	layer: 3, lr: 10^{-2}, hidden: 512, L_2: 10^{-2}, dropout: 0.6, activation: GeLU
	GT	layer: 4, lr: 10^{-2}, hidden: 128, L_2: 10^{-5}, dropout: 0.2, activation: GeLU, heads: 4
Minesweeper	FSGNN	hops: 3, lr: 10^{-2}, hidden: 256, L_2: 10^{-4}, dropout: 0, activation: ReLU
	GAT	layer: 5, lr: 10^{-2}, hidden: 32, L_2: 10^{-5}, dropout: 0.2, activation: GeLU, heads: 8
	GCN	layer: 4, lr: 10^{-2}, hidden: 32, L_2: 10^{-4}, dropout: 0.0, activation: GeLU
	GPRGNN	steps: 4, lr: 10^{-2}, hidden: 64, L_2: 10^{-2}, dropout: 0.6, activation: GeLU
	GraphSAGE	layer: 5, lr: 10^{-2}, hidden: 32, L_2: 10^{-3}, dropout: 0.4, activation: GeLU
	H_2GCN	hops: 3, lr: 10^{-2}, hidden: 128, L_2: 10^{-5}, dropout: 0.0, activation: GeLU
	LINKX	fusion layer: 5, edge layer: 1, node layer: 3, lr: 10^{-2}, hidden: 128, L_2: 10^{-5}, dropout: 0.4, activation: GeLU
	MLP	layer: 4, lr: 10^{-4}, hidden: 64, L_2: 10^{-3}, dropout: 0.4, activation: ReLU
	MLP+adj	layer: 4, lr: 10^{-2}, hidden: 64, L_2: 10^{-2}, dropout: 0.0, activation: GeLU
	GP-GNN	layer: 5, lr: 10^{-2}, hidden: 64, L_2: 10^{-5}, dropout: 0.4, activation: GeLU
	GT	layer: 5, lr: 10^{-3}, hidden: 32, L_2: 10^{-4}, dropout: 0.4, activation: GeLU, heads: 8
Tolokers	FSGNN	hops: 2, lr: 10^{-2}, hidden: 128, L_2: 10^{-4}, dropout: 0.2, activation: ReLU
	GAT	layer: 5, lr: 10^{-2}, hidden: 128, L_2: 10^{-4}, dropout: 0.2, activation: ReLU, heads: 4
	GCN	layer: 5, lr: 10^{-2}, hidden: 512, L_2: 10^{-3}, dropout: 0.0, activation: ReLU
	GPRGNN	steps: 10, lr: 10^{-2}, hidden: 512, L_2: 10^{-4}, dropout: 0.0, activation: ReLU
	GraphSAGE	layer: 5, lr: 10^{-2}, hidden: 32, L_2: 10^{-5}, dropout: 0.2, activation: ReLU
	H_2GCN	hops: 2, lr: 10^{-2}, hidden: 128, L_2: 10^{-3}, dropout: 0.0, activation: GeLU
	LINKX	fusion layer: 2, edge layer: 3, node layer: 1, lr: 10^{-2}, hidden: 128, L_2: 10^{-5}, dropout: 0.4, activation: GeLU
	MLP	layer: 4, lr: 10^{-2}, hidden: 512, L_2: 10^{-2}, dropout: 0.0, activation: ReLU
	MLP+adj	layer: 5, lr: 10^{-3}, hidden: 256, L_2: 10^{-3}, dropout: 0.0, activation: GeLU
	GP-GNN	layer: 3, lr: 10^{-2}, hidden: 128, L_2: 10^{-3}, dropout: 0.6, activation: GeLU
	GT	layer: 4, lr: 10^{-2}, hidden: 32, L_2: 10^{-4}, dropout: 0.4, activation: GeLU, heads: 4
Questions	FSGNN	hops: 3, lr: 10^{-2}, hidden: 512, L_2: 10^{-2}, dropout: 0.2, activation: GeLU
	GAT	layer: 3, lr: 10^{-2}, hidden: 128, L_2: 10^{-3}, dropout: 0.4, activation: GeLU, heads: 2
	GCN	layer: 4, lr: 10^{-2}, hidden: 64, L_2: 10^{-3}, dropout: 0.0, activation: GeLU
	GPRGNN	steps: 3, lr: 10^{-2}, hidden: 512, L_2: 10^{-4}, dropout: 0.6, activation: ReLU
	GraphSAGE	layer: 5, lr: 10^{-2}, hidden: 64, L_2: 10^{-3}, dropout: 0.4, activation: GeLU
	H_2GCN	hops: 3, lr: 10^{-2}, hidden: 64, L_2: 10^{-4}, dropout: 0.2, activation: GeLU
	LINKX	fusion layer: 5, edge layer: 1, node layer: 5, lr: 10^{-4}, hidden: 512, L_2: 10^{-3}, dropout: 0.6, activation: ReLU
	MLP	layer: 5, lr: 10^{-2}, hidden: 512, L_2: 10^{-4}, dropout: 0.6, activation: ReLU
	MLP+adj	layer: 2, lr: 10^{-2}, hidden: 32, L_2: 10^{-2}, dropout: 0.4, activation: GeLU
	GP-GNN	layer: 5, lr: 10^{-4}, hidden: 512, L_2: 10^{-3}, dropout: 0.2, activation: ReLU
	GT	layer: 2, lr: 10^{-2}, hidden: 32, L_2: 10^{-5}, dropout: 0.6, activation: GeLU, heads: 2

Table 6: Optimal hyperparameter configurations for GP-GNN and baseline models across homophilic benchmarks, determined via TPE-based search.

Dataset	Model	Optimal Configuration
Cora	FSGNN	hops: 3, lr: 0.01, hidden: 512, L_2: 0.01, dropout: 0.6, activation: GeLU
	GAT	layer: 2, lr: 0.001, hidden: 64, L_2: 10^{-4}, dropout: 0.6, activation: GeLU, heads: 4
	GCN	layer: 3, lr: 0.001, hidden: 32, L_2: 10^{-5}, dropout: 0.6, activation: GeLU
	GPRGNN	steps: 10, lr: 0.01, hidden: 64, L_2: 10^{-4}, dropout: 0.2, activation: GeLU
	GraphSAGE	layer: 4, lr: 0.001, hidden: 128, L_2: 0.01, dropout: 0.4, activation: GeLU
	H_2GCN	hops: 2, lr: 0.01, hidden: 32, L_2: 10^{-5}, dropout: 0.4, activation: GeLU
	LINKX	fusion layer: 1, edge layer: 1, node layer: 1, lr: 0.001, hidden: 512, L_2: 0.01, dropout: 0.0, activation: GeLU
	MLP	layer: 1, lr: 0.001, hidden: 512, L_2: 10^{-5}, dropout: 0.6, activation: ReLU
	MLP+adj	layer: 1, lr: 0.001, hidden: 512, L_2: 10^{-4}, dropout: 0.4, activation: ReLU
	GP-GNN	layer: 3, lr: 10^{-4}, hidden: 512, L_2: 0.001, dropout: 0.4, activation: GeLU
	GT	layer: 3, lr: 0.001, hidden: 64, L_2: 0.01, dropout: 0.4, activation: ReLU, heads: 8
CiteSeer	FSGNN	hops: 1, lr: 0.01, hidden: 512, L_2: 0.001, dropout: 0.6, activation: GeLU
	GAT	layer: 1, lr: 0.01, hidden: 64, L_2: 10^{-4}, dropout: 0.4, activation: ReLU, heads: 2
	GCN	layer: 1, lr: 0.01, hidden: 64, L_2: 0.001, dropout: 0.4, activation: ReLU
	GPRGNN	steps: 4, lr: 10^{-4}, hidden: 128, L_2: 10^{-4}, dropout: 0.4, activation: ReLU
	GraphSAGE	layer: 1, lr: 0.001, hidden: 128, L_2: 0.001, dropout: 0.6, activation: ReLU
	H_2GCN	hops: 2, lr: 0.01, hidden: 64, L_2: 10^{-5}, dropout: 0.2, activation: ReLU
	LINKX	fusion layer: 1, edge layer: 1, node layer: 1, lr: 0.001, hidden: 512, L_2: 0.001, dropout: 0.4, activation: GeLU
	MLP	layer: 2, lr: 10^{-4}, hidden: 512, L_2: 10^{-5}, dropout: 0.6, activation: ReLU
	MLP+adj	layer: 2, lr: 0.001, hidden: 128, L_2: 0.001, dropout: 0.4, activation: ReLU
	GP-GNN	layer: 1, lr: 10^{-4}, hidden: 512, L_2: 10^{-4}, dropout: 0.2, activation: ReLU
	GT	layer: 3, lr: 0.001, hidden: 256, L_2: 0.001, dropout: 0.6, activation: ReLU, heads: 8
PubMed	FSGNN	hops: 3, lr: 0.01, hidden: 512, L_2: 10^{-4}, dropout: 0.6, activation: ReLU
	GAT	layer: 2, lr: 0.01, hidden: 64, L_2: 0.001, dropout: 0.6, activation: ReLU, heads: 2
	GCN	layer: 2, lr: 0.01, hidden: 512, L_2: 0.01, dropout: 0.6, activation: ReLU
	GPRGNN	steps: 9, lr: 0.01, hidden: 512, L_2: 0.01, dropout: 0.6, activation: ReLU
	GraphSAGE	layer: 3, lr: 0.001, hidden: 512, L_2: 10^{-4}, dropout: 0.6, activation: ReLU
	H_2GCN	hops: 2, lr: 0.01, hidden: 128, L_2: 0.001, dropout: 0.4, activation: ReLU
	LINKX	fusion layer: 2, edge layer: 1, node layer: 4, lr: 0.001, hidden: 512, L_2: 10^{-5}, dropout: 0.0, activation: GeLU
	MLP	layer: 1, lr: 0.01, hidden: 512, L_2: 0.001, dropout: 0.6, activation: ReLU
	MLP+adj	layer: 4, lr: 0.01, hidden: 512, L_2: 10^{-5}, dropout: 0.6, activation: ReLU
	GP-GNN	layer: 3, lr: 0.01, hidden: 256, L_2: 0.01, dropout: 0.6, activation: ReLU
	GT	layer: 4, lr: 0.01, hidden: 512, L_2: 10^{-4}, dropout: 0.6, activation: ReLU, heads: 2
Amz. Computers	FSGNN	hops: 2, lr: 0.01, hidden: 128, L_2: 10^{-5}, dropout: 0.6, activation: ReLU
	GAT	layer: 3, lr: 0.001, hidden: 256, L_2: 10^{-4}, dropout: 0.6, activation: ReLU, heads: 2
	GCN	layer: 3, lr: 0.001, hidden: 512, L_2: 10^{-4}, dropout: 0.6, activation: ReLU
	GPRGNN	steps: 9, lr: 0.01, hidden: 256, L_2: 0.001, dropout: 0.6, activation: ReLU
	GraphSAGE	layer: 3, lr: 0.01, hidden: 512, L_2: 0.001, dropout: 0.6, activation: ReLU
	H_2GCN	hops: 3, lr: 0.01, hidden: 64, L_2: 0.01, dropout: 0.6, activation: GeLU
	LINKX	fusion layer: 2, edge layer: 1, node layer: 1, lr: 0.001, hidden: 512, L_2: 0.001, dropout: 0.4, activation: ReLU
	MLP	layer: 1, lr: 0.01, hidden: 512, L_2: 10^{-5}, dropout: 0.4, activation: ReLU
	MLP+adj	layer: 2, lr: 0.01, hidden: 256, L_2: 10^{-5}, dropout: 0.2, activation: ReLU
	GP-GNN	layer: 3, lr: 0.01, hidden: 256, L_2: 0.01, dropout: 0.6, activation: GeLU
	GT	layer: 3, lr: 0.01, hidden: 64, L_2: 0.001, dropout: 0.4, activation: GeLU, heads: 8
Amz. Photo	FSGNN	hops: 1, lr: 0.01, hidden: 128, L_2: 10^{-5}, dropout: 0.6, activation: ReLU
	GAT	layer: 3, lr: 0.001, hidden: 128, L_2: 10^{-5}, dropout: 0.4, activation: GeLU, heads: 2
	GCN	layer: 4, lr: 0.001, hidden: 512, L_2: 10^{-4}, dropout: 0.6, activation: ReLU
	GPRGNN	steps: 2, lr: 0.001, hidden: 512, L_2: 10^{-4}, dropout: 0.6, activation: ReLU
	GraphSAGE	layer: 5, lr: 0.001, hidden: 128, L_2: 10^{-5}, dropout: 0.6, activation: ReLU
	H_2GCN	hops: 2, lr: 0.001, hidden: 128, L_2: 0.001, dropout: 0.6, activation: ReLU
	LINKX	fusion layer: 2, edge layer: 1, node layer: 1, lr: 0.001, hidden: 512, L_2: 0.01, dropout: 0.4, activation: ReLU
	MLP	layer: 1, lr: 0.01, hidden: 256, L_2: 10^{-5}, dropout: 0.6, activation: ReLU
	MLP+adj	layer: 1, lr: 0.01, hidden: 256, L_2: 10^{-4}, dropout: 0.6, activation: ReLU
	GP-GNN	layer: 4, lr: 0.001, hidden: 256, L_2: 0.01, dropout: 0.6, activation: ReLU
	GT	layer: 3, lr: 0.01, hidden: 32, L_2: 0.001, dropout: 0.4, activation: ReLU, heads: 8

Table 7: Node classification Accuracy (%). Results are reported as mean ± standard deviation over 50 random splits. Nominal ranks: first, second, third. Asterisks (*) denote a statistical tie with GP-GNN ($\alpha = 0.05$). Note that all models were tuned using the validation Macro-F1 score; these results demonstrate that GP-GNN's performance gains are consistent across diverse evaluation metrics.

Dataset	MLP	MLP+adj	GCN	GAT	SAGE	GT	LINKX	H_2GCN	FSGNN	GPR-GNN	GP-GNN
Heterophilic Datasets											
Roman Empire	66.27 ± 0.46	77.10 ± 0.58	84.95 ± 0.46	85.19 ± 0.57	88.38 ± 0.42	89.45 ± 0.47	65.20 ± 0.48	82.93 ± 0.45	79.21 ± 0.52	73.94 ± 0.49	89.93 ± 0.48
Amz. Ratings	49.44 ± 0.65	52.60 ± 0.61	51.51 ± 0.62	51.75 ± 0.62	53.33 ± 0.52	51.96 ± 0.61	52.81 ± 0.58	51.69 ± 0.57	52.31 ± 0.62	53.29 ± 0.54	53.63 ± 0.64
Minesweeper	79.99 ± 0.04	84.05 ± 0.51	86.59 ± 0.53	88.89 ± 0.46	89.55 ± 0.44*	88.04 ± 0.72	80.00 ± 0.00	85.90 ± 0.51	85.59 ± 0.69	84.75 ± 0.46	89.46 ± 0.46
Tolokers	78.13 ± 0.25	79.06 ± 0.67	82.00 ± 0.65	82.25 ± 0.59	81.93 ± 0.70	81.74 ± 0.56	74.56 ± 3.88	81.77 ± 0.60	81.98 ± 0.60	78.53 ± 0.44	82.83 ± 0.53
Questions	97.24 ± 0.08*	97.17 ± 0.07	97.12 ± 0.05	97.16 ± 0.06	97.17 ± 0.06	97.13 ± 0.05	97.05 ± 0.34	97.16 ± 0.06	97.11 ± 0.06	97.09 ± 0.08	97.22 ± 0.08
Homophilic Datasets											
Cora	75.81 ± 1.57	86.09 ± 1.16	86.98 ± 1.34*	87.67 ± 1.20	86.81 ± 1.22*	86.71 ± 1.22*	83.69 ± 1.22	87.35 ± 1.18	87.17 ± 1.01*	87.42 ± 1.12	86.87 ± 1.32
CiteSeer	72.35 ± 1.38	75.27 ± 1.28	75.60 ± 1.42	75.83 ± 1.50	76.26 ± 1.19*	73.87 ± 1.32	73.09 ± 1.37	75.41 ± 1.42	75.44 ± 1.51	75.55 ± 1.44	76.35 ± 1.33
PubMed	87.51 ± 0.47	87.90 ± 0.46	88.91 ± 0.46	89.35 ± 0.43	89.06 ± 0.43	88.78 ± 0.55	87.11 ± 0.48	89.11 ± 0.40	88.83 ± 0.46	88.97 ± 0.43	89.24 ± 0.43
Amz. Computers	86.08 ± 0.55	90.21 ± 0.38	92.59 ± 0.40	92.58 ± 0.46	91.94 ± 0.37	92.08 ± 0.44	91.19 ± 0.49	91.99 ± 0.44	92.24 ± 0.48*	91.75 ± 0.43	92.30 ± 0.41
Amz. Photo	92.13 ± 0.60	95.09 ± 0.48	95.83 ± 0.41*	95.79 ± 0.46*	95.75 ± 0.42*	95.58 ± 0.45	95.24 ± 0.41	95.84 ± 0.40*	95.65 ± 0.38	95.57 ± 0.42	95.80 ± 0.35