Heterophily-Aware Personalized PageRank for Node Classification

Giuseppe Pirrò
Department of Mathematics and Computer Science, University of Calabria
87046, Rende (CS), Italy
giuseppe.pirro@unical.it

Abstract

Node classification in heterophilous graphs, where connected nodes often have different characteristics, presents a significant challenge. We introduce HAPPY, which combines heterophily-aware random walks with targeted subgraph extraction. Our approach enhances Personalized PageRank by incorporating both label and feature diversity into the random walk process. Through theoretical analysis, we demonstrate that HAPPYeffectively captures both homophilous and heterophilous relationships. Comprehensive experiments validate our method's state-of-the-art performance across challenging heterophilous benchmarks.

1 Introduction

Node classification in heterophilous graphs—where linked nodes often exhibit significant dissimilarity in their attributes or class labels—poses unique challenges [Luan *et al.*, 2024]. For example, in a citation network, a groundbreaking paper might be most relevant to works that challenge or diverge from its approach rather than similar papers. The inherent structure of heterophilous graphs presents a key challenge for graph-based learning. A fundamental question emerges:

> *Can we identify truly relevant neighbors for a node when traditional proximity-based assumptions break down?*

Related Work. Research in heterophilous graphs has received significant attention [Luan *et al.*, 2024; Zheng *et al.*, 2022; Platonov *et al.*, 2023b]. Below, we categorize existing approaches most related to our work following Table 1.
(1) Multi-hop view: These methods leverage multi-hop neighborhood information. H_2GCN [Zhu *et al.*, 2020] separates ego and neighbor embeddings, while FSGNN [Maurya *et al.*, 2022] guides aggregation through feature similarity.
(2) Global Homophily: These approaches exploit global graph properties. GloGNN [Li *et al.*, 2022] integrates global-local information, CPGNN [Zhu *et al.*, 2021] uses compatibility matrices, Geom-GCN [Liu *et al.*, 2021] leverages geometric relationships, and GPNN [Yang *et al.*, 2022] learns neighbors via attention.

Category	**Method**	**Key Characteristics**
Multi-hop view	H_2GCN [Zhu *et al.*, 2020]	Ego/neighbor-embedding separation
	FSGNN [Maurya *et al.*, 2022]	Feature-similarity guided aggregation
Global Homophily	GloGNN [Li *et al.*, 2022]	Global-local information integration
	CPGNN [Zhu *et al.*, 2021]	Prior-based compatibility matrix
	Geom-GCN [Liu *et al.*, 2021]	Geometric relationships
	GPNN [Yang *et al.*, 2022]	Learned neighbour via attention
Discriminative Message Passing	FAGCN [Bo *et al.*, 2021]	Adaptive edge-aware techniques
	GBK-GNN [Du *et al.*, 2022]	Bi-kernel transformation
New Neighbours	NLGNN [Liu *et al.*, 2021]	Non-local Neighbours
	OGNN [Pirrò, 2023]	Semantic Overlay Network
Spectral	GPR-GNN [Chien *et al.*, 2021]	Adaptive PageRank weights
	BernNet [He *et al.*, 2021]	Bernstein polynomial filtering
	JacobiConv [Wang and Zhang, 2022]	Polynomial-based filtering
	ChebNetII [He *et al.*, 2022]	Chebyshev polynomial-based GNN
Adaptive	Restruct-GCN [Li *et al.*, 2023]	Learned topology restructuring
	ACM-GCN [Luan *et al.*, 2022]	Adaptive frequency band usage
	ASCG [Chanpuriya and Musco, 2022]	Heterophily-aware coefficients
	CDE [Zhao *et al.*, 2023]	Cross-feature diffusion networks
	ACMP [Wang *et al.*, 2023]	Interactive particle system
	HeterGCL [Wang *et al.*, 2024]	Graph Contrastive learning
Subgraph-based	HAPPY (**Ours**)	Node-centric subgraph extraction using heterophily-aware PPR scores.

Table 1: Characteristics of most related approaches to our work.

(3) Discriminative Message Passing: Methods like FAGCN [Bo *et al.*, 2021] and GBK-GNN [Du *et al.*, 2022] focus on adaptive edge-aware techniques and bi-kernel transformations respectively.
(4) New Neighbours: Instead of using only original connections, NLGNN [Liu *et al.*, 2021] and OGNN [Pirrò, 2023] identify non-local neighbours via learnable node orderings.
(5) Spectral Approaches: These methods employ various spectral techniques: GPR-GNN [Chien *et al.*, 2021] uses adaptive PageRank weights, BernNet [He *et al.*, 2021] , JacobiConv [Wang and Zhang, 2022], and ChebNetII [He *et al.*, 2022] leverage different types of polynomial filtering.
Adaptive Approaches: These methods dynamically adjust to graph properties. Restruct-GCN [Li *et al.*, 2023] learns topology restructuring, ACM-GCN [Luan *et al.*, 2022] adapts frequency bands, ASCG [Chanpuriya and Musco, 2022] uses heterophily-aware coefficients, CDE [Zhao *et al.*, 2023] employs cross-feature diffusion, ACMP [Wang *et al.*, 2023] uses interactive particle systems, and HeterGCL [Wang *et al.*, 2024] applies graph contrastive learning.

Limitations of Existing Approaches. Despite considerable progress, each category of heterophilous graph methods faces specific challenges. *Multi-hop view* approaches often struggle with computational complexity as the hop count increases.

Global Homophily methods can overlook local structural patterns while introducing significant memory overhead. *Discriminative Message Passing* techniques require careful feature engineering and often lack theoretical guarantees. *New Neighbours* strategies introduce computational cost in neighbor discovery and rely heavily on domain-specific heuristics. *Spectral* methods demand non-trivial parameter tuning for filter orders and can miss localized graph properties. *Adaptive* approaches typically require complex architectures.

Our Proposal. We introduce HAPPY (Heterophily-Aware Personalized PageRank), which builds upon a key insight: *in heterophilous graphs, influential nodes might not be immediate neighbors, but rather nodes that share similar patterns across multiple hops*. This observation makes Personalized PageRank (PPR) [Haveliwala, 2002] an attractive foundation, as it naturally measures node importance through random walks while allowing for personalization based on node-specific characteristics. However, traditional PPR falls short in heterophilous settings due to its inherent homophily assumption in the random walk process. Our approach makes three key contributions to address these challenges:

(1) *Heterophily-Aware Random Walks:* We modify PPR's random walk process to integrate feature diversity directly, enabling the discovery of relevant nodes even when they differ from their neighbors. This adaptation balances structural connectivity with feature diversity.

(2) *Context-Sensitive Subgraphs:* Using our enhanced PPR scores, we develop efficient node-centric subgraph extraction methods that preserve both feature dissimilarity and structural patterns, creating robust local neighborhoods that capture heterophilous relationships.

(3) *Theoretical Guarantees:* We provide a theoretical analysis of how HAPPY preserves both homophilous and heterophilous relationships, making it particularly suitable for graphs with varying structural patterns.

(4) *Effective and Scalable:* By combining our subgraph reasoning with simple feature transformation approaches like SGC [Wu *et al.*, 2019], HAPPY achieves state-of-the-art performance while maintaining high scalability.

Outline. We start with some background (§2). Then, we present H-PPR (§3), the subgraph extraction strategies (§4), the classification pipeline (§5), and the theoretical analysis (§6). We evaluate HAPPY (§7) and then conclude (§8).

2 Background and Problem Formulation

Let $G = (\mathcal{V}, \mathcal{E})$ denote a graph with node features $\mathbf{X} \in \mathbf{R}^{N \times N_F}$. We denote by $\mathcal{N}(u)$ the neighbors of a node u. For nodes u and v, their feature dissimilarity is: $h(u, v) = 1 - \text{sim}(\mathbf{x}_u, \mathbf{x}_v)$, where $\text{sim}(\cdot,\cdot)$ is a bounded similarity function in $[0, 1]$. The edge homophily ratio and its adjusted version [Platonov *et al.*, 2023a] are defined as:

$$\mathcal{H}_{\text{edge}} = \frac{|\{(u,v) : (u,v) \in \mathcal{E} \wedge y_u = y_v\}|}{|\mathcal{E}|} \tag{1}$$

$$\mathcal{H}_{\text{adjusted}} = \frac{\mathcal{H}_{\text{edge}} - \sum_{k=1}^{C} D_k^2/(2|\mathcal{E}|)^2}{1 - \sum_{k=1}^{C} D_k^2/(2|\mathcal{E}|)^2} \tag{2}$$

where $D_k := \sum_{i:y_i=k} d(i)$ for class label k.

> **Problem**: Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with labeled nodes $\mathbf{Y}_L$, we aim to learn $f_p(\mathcal{G}, \mathbf{Y}_L) \rightarrow \hat{\mathbf{Y}}_U$, where $\hat{\mathbf{Y}}_U$ are the predicted labels for unlabeled nodes.

HAPPY is motivated by recent advances in graph learning that emphasize the importance of effectively combining structural and feature information while adapting to graph properties [Zhu *et al.*, 2020; Bo *et al.*, 2021; Yang *et al.*, 2021; Li *et al.*, 2022]. This insight stems from a key observation: real-world graphs exhibit varying levels of heterophily, where connected nodes may have either different or similar features. To address this challenge, HAPPY employs a decoupled architecture that leverages both *similarity-driven* and *dissimilarity-driven* context. The approach operates in three steps: first computing Heterophily-Aware Personalized PageRank (H-PPR) scores (§ 3), then extracting *node-specific* contextual subgraphs (§ 4), and finally learning, transforming and feeding node representations to a classifier (§ 5). Through this node-centric approach, HAPPY tailors both structural and feature information to each node's local neighborhood, enabling better pattern identification by combining structural connectivity with relevant feature patterns.

3 Heterophily-Aware Personalized PageRank

H-PPR extends PPR [Haveliwala, 2002] by enabling effective navigation of graphs where connected nodes can be either similar or dissimilar thus proving more robust node classification across diverse graph structures.

Definition 3.1 (H-PPR)**.** We define a transition probability matrix $\mathbf{P}$ that adapts to each node's local heterophily level. Specifically, for nodes $u \in \mathcal{V}$ and neighbors $v \in \mathcal{N}(u)$, let

$$h_{\text{adaptive}}(u,v) \;=\; \lambda_u \cdot \text{dissim}(u,v) + \big(1 - \lambda_u\big) \cdot \text{sim}(u,v) \tag{3}$$

where $\lambda_u \in [0, 1]$ is the *local heterophily coefficient* of node u (§3.1). Then each entry P_{uv} of $\mathbf{P}$ is:

$$P_{uv} = \begin{cases} \dfrac{h_{\text{adaptive}}(u,v)}{\sum_{w \in N(u)} h_{\text{adaptive}}(u,w)} & \text{if } v \in N(u), \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The H-PPR score vector $\boldsymbol{\pi}_t$ satisfies:

$$\boldsymbol{\pi}_t \;=\; \alpha\, \mathbf{P}\, \boldsymbol{\pi}_t \;+\; (1-\alpha) \Big[\beta\, \mathbf{e}_t \;+\; (1-\beta)\, \mathbf{r}_{\text{adaptive}} \Big] \tag{5}$$

where $\alpha \in (0, 1)$ is a damping factor, $\beta \in [0, 1]$ a local–global restart parameter, $\mathbf{e}_t$ a vector that is one-hot at target node t, and $\mathbf{r}_{\text{adaptive}}$ an *adaptive restart* distribution (§3.2).

3.1 Local Heterophily and Adaptive Transition

HAPPY aims at leveraging both graph structure and features while effectively handling both homophilous and heterophilous relationships. However, with only limited labeled nodes available in real-world graphs, we face two challenges: (1) estimating local heterophily patterns to guide random walks, and (2) measuring neighborhood diversity to enable adaptive transitions. To address these challenges, *we leverage pseudo-labels that propagate train labels across the network through both structural connections and feature similarities*. We derive pseudo-labels $\mathbf{Y}$ via label propagation:

$\mathbf{Y}^{(k+1)} = \mathbf{D}^{-1}\mathbf{A}\mathbf{S}\mathbf{Y}^{(k)}$ where $\mathbf{D}$ is the degree matrix, $\mathbf{A}$ is the adjacency matrix, and $\mathbf{S}$ is a diagonal feature similarity matrix with $S_{ij} = \text{sim}(X_i, X_j)$ (e.g., based on cosine) for neighboring nodes. Pseudo-labels enable us to compute essential metrics like *local heterophily* coefficients and *label diversity*, which are crucial for guiding adaptive transitions and balancing exploration between diverse and similar regions.

Local Heterophily Computation: for each node $v \in \mathcal{V}$, we define λ_v to capture whether its neighbors are more similar or more dissimilar. We combine two heterophily indicators:

$$\lambda_v \;=\; \gamma \cdot h_{\text{label}}(v) \;+\; \bigl(1-\gamma\bigr) \cdot h_{\text{feat}}(v) \tag{6}$$

where $\gamma \in [0,1]$ is a balance parameter. The label-based term $h_{\text{label}}(v)$ is the fraction of neighbors whose pseudo-label in $\mathbf{Y}$ differs from v's own label:

$$h_{\text{label}}(v) = \begin{cases} \dfrac{|\{\, u \in \mathcal{N}(v) : y_u \neq y_v \}|}{|\mathcal{N}(v)|} & \text{if } \mathcal{N}(v) \neq \emptyset \\ 0.5 & \text{otherwise} \end{cases}$$

While the feature-based term $h_{\text{feat}}(v)$ is the *average dissimilarity* w.r.t. features $\mathbf{X}$:

$$h_{\text{feat}}(v) = \begin{cases} \dfrac{1}{|\mathcal{N}(v)|} \displaystyle\sum_{u \in \mathcal{N}(v)} \bigl[\, 1 - \cos(\mathbf{x}_v, \mathbf{x}_u) \bigr] & \text{if } \mathcal{N}(v) \neq \emptyset \\ 0.5 & \text{otherwise} \end{cases}$$

Hence, λ_v approaches 1 if v's neighborhood is highly *heterophilous* (different labels or features), and it approaches 0 if neighbors are mostly *similar*.

Heterophily-Aware Transitions: using each node's local heterophily λ_u in (3), we define the base transition probability:

$$P(u \to v) \;\propto\; \lambda_u \, \text{dissim}(u, v) \;+\; (1 - \lambda_u) \, \text{sim}(u, v) \tag{7}$$

If u has $\lambda_u \approx 1$, it prefers neighbors more dissimilar in features/labels; if $\lambda_u \approx 0$, it prefers those more similar.

3.2 Adaptive Restart Distribution

In addition to local node-level coefficients, we also estimate a *global* heterophily level λ_{global} (e.g., the average of all $\{\lambda_v\}$). This value governs how the random walk restarts from diverse vs. homogeneous nodes. For each node v we define:

1. Local Diversity: we measure *label diversity* $r_{\text{div}}(v)$ in v's neighborhood computing pseudo-label entropy as:

   $$r_{\text{div}}(v) = \frac{\text{Entropy}\bigl(\mathbf{Y}[\mathcal{N}(v)]\bigr) + \epsilon}{\sum_{u \in \mathcal{V}} \bigl[\text{Entropy}\bigl(\mathbf{Y}[\mathcal{N}(u)]\bigr) + \epsilon\bigr]} \tag{8}$$

   where ϵ is a small constant (e.g. 10^{-10}) to avoid zero denominators. A higher $r_{\text{div}}(v)$ indicates that v's neighborhood has a broader distribution of pseudo-labels.

2. Local Similarity: it measures *feature homogeneity* $r_{\text{sim}}(v)$ in v's neighborhood via average similarity:

   $$r_{\text{sim}}(v) = \frac{\text{Homogeneity}(\mathcal{N}(v)) + \epsilon}{\sum_{u \in \mathcal{V}} \bigl[\text{Homogeneity}(\mathcal{N}(u)) + \epsilon\bigr]} \tag{9}$$

   where $\text{Homogeneity}(\mathcal{N}(v))$ can be computed, for instance, as $\frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} \cos(\mathbf{x}_v, \mathbf{x}_u)$.

We then form the final *adaptive restart* distribution:

$$\mathbf{r}_{\text{adaptive}}(v) \;=\; \lambda_{\text{global}} \, r_{\text{div}}(v) \;+\; \bigl(1 - \lambda_{\text{global}}\bigr) \, r_{\text{sim}}(v) \tag{10}$$

Hence, when λ_{global} is high (i.e., the graph is largely diverse), the walk restarts more frequently in neighborhoods with high label entropy, and vice versa when λ_{global} is low. This ensures that the random walk *globally* balances exploring dissimilar regions against reinforcing neighborhoods of similarity.

Balanced Feedback to Avoid Local Traps

Despite the above mechanisms, a random walk may still converge too quickly into exclusively similar or dissimilar clusters. To mitigate such "traps," we introduce a *balanced feedback* adjustment to the transition probabilities:

$$P'(u \to v) \;=\; P(u \to v) \;\times\; \Bigl[\, 1 \;-\; \beta \, \texttt{bal}\bigl(v, \mathcal{V}_v, \lambda_u\bigr) \Bigr] \tag{11}$$

where $\mathcal{V}_v$ is the set of nodes visited so far, $\beta \in [0,1]$ is a *feedback strength*, and $\texttt{bal}(v, \mathcal{V}_v, \lambda_u)$ is a function that evaluates whether v has been *over*- or *under*-explored under a similar local heterophily context λ_u. For example, if v has been visited frequently when λ_u is high, we reduce $P'(u \to v)$ to encourage more exploration of homophilous regions and vice versa. This can be implemented by tracking the visit counts during the walk and then adjusting $P(\cdot)$ at each iteration.

3.3 Algorithm and Implementation Details

Algorithm 1 illustrates a reference implementation of H-PPR, unifying local and global adaptations, as well as optional *balanced feedback* if desired.

Algorithm 1 Heterophily-Aware Personalized PageRank

```
Input:
  Graph $G = (\mathcal{V}, \mathcal{E})$ with node features $\mathbf{X}$ and pseudo-labels $\mathbf{Y}$
  Parameters: damping $\alpha \in (0,1)$, local–global restart $\beta \in [0,1]$, balance $\gamma \in [0,1]$
  $max\_iter$, convergence tolerance $\epsilon$
Output:
  H-PPR score dictionary $\{\boldsymbol{\pi}_u : u \in \mathcal{V}\}$
function H-PPR($G, \mathbf{X}, \mathbf{Y}, \alpha, \beta, \gamma, max\_iter, \epsilon$)
    $\lambda_{\text{global}} \leftarrow$ EstimateGlobalHeterophily($G, \mathbf{Y}$)
    $\boldsymbol{\lambda} \leftarrow$ ComputeLocalHeterophily($G, \mathbf{X}, \mathbf{Y}, \gamma$)
    $\mathbf{r}_{\text{adaptive}} \leftarrow$ ComputeAdaptiveRestart($G, \mathbf{X}, \mathbf{Y}, \lambda_{\text{global}}$)
    $\mathbf{S} \leftarrow \text{cosine\_similarity}(\mathbf{X})$
    $\mathbf{D} \leftarrow \mathbf{1} - \mathbf{S}$
    scores $\leftarrow \{\}$
    for each source node $u \in \mathcal{V}$ do
        // Build transition matrix $\mathbf{P}$
        $\mathbf{P} \leftarrow \mathbf{0}^{|\mathcal{V}| \times |\mathcal{V}|}$
        for each node $v \in \mathcal{V}$ do
            $\mathcal{N}_v \leftarrow \text{neighbors}(v)$
            if $\mathcal{N}_v \neq \emptyset$ then
                $\mathbf{w}_v \leftarrow \lambda_v \, \mathbf{D}[v, \mathcal{N}_v] \; + \; (1 - \lambda_v) \, \mathbf{S}[v, \mathcal{N}_v]$
                $\mathbf{P}[v, \mathcal{N}_v] \leftarrow \mathbf{w}_v \; / \; \sum(\mathbf{w}_v)$
        // Power iteration for node $u$
        $\mathbf{p} \leftarrow \mathbf{e}_u$     // One-hot vector
        for iter = 1 to $max\_iter$ do
            $\mathbf{p}_{\text{new}} \leftarrow \alpha \, \mathbf{P}^{\top} \mathbf{p} \; + \; (1 - \alpha) \bigl[\beta \, \mathbf{e}_u + (1 - \beta) \, \mathbf{r}_{\text{adaptive}}\bigr]$
            if $\|\mathbf{p}_{\text{new}} - \mathbf{p}\|_1 < \epsilon$ then
                break
            $\mathbf{p} \leftarrow \mathbf{p}_{\text{new}}$
        scores[$u$] $\leftarrow \mathbf{p}$
    return scores
```

Interpretation. By incorporating both *feature-based* and *label-based* information, H-PPR can rank nodes that are similar in one modality but complementary in another. The adaptive restart further ensures that the random walk explores diverse neighborhoods if the overall graph is heterophilous, or focuses on more homogeneous regions if the graph is highly homophilous. Finally, *balanced feedback* (§3.2) ensures that the walk does not get "stuck" in only one type of region. Hence, H-PPR provides a robust and flexible ranking mechanism that adapts to graphs spanning the entire homophily–heterophily spectrum.

Complexity Analysis. Algorithm 1 primarily spends its time in three stages for each source node $u \in \mathcal{V}$: (1) *Feature Similarity and Transition Matrix Construction* (e.g., line 9 and 13–19). Computing cosine similarities of dimension d and then forming neighbor-based weights is $O\big(|\mathcal{N}_u|\, d\big)$ for node u; (2) *Power Iteration* (e.g., lines 21–26). Each update of the score vector $\mathbf{p}$ takes $O(|\mathcal{E}|)$ when the transition matrix is stored in a sparse format. If k is the number of iterations until convergence, this part costs $O(k\,|\mathcal{E}|)$; (3) *Local/Global Heterophily Computations*. Estimating local heterophily λ_v and the global level λ_{global} each require summing over a node's neighborhood, i.e. $O(|\mathcal{N}_u|)$ per node. Summed over all nodes, this is $O\big(\sum_{v\in\mathcal{V}}|\mathcal{N}_v|\big) = O(|\mathcal{E}|)$ in an undirected graph. Hence, for a single source node, the total cost is $O\Big(\sum_{u\in\mathcal{V}}\big(|\mathcal{N}_u|\,(d+1)\big) \;+\; k\,|\mathcal{E}|\Big)$. Because $\sum_{u\in\mathcal{V}}|\mathcal{N}_u| = 2|\mathcal{E}|$, this simplifies to $O\big(|\mathcal{E}|\,(d+1)\big) \;+\; O\big(k\,|\mathcal{E}|\big) \;=\; O\big(k\,|\mathcal{E}| + d\,|\mathcal{E}|\big)$. If *every* node serves as a source, we multiply by $|\mathcal{V}|$, unless certain computations (e.g., similarity) are reused across different source nodes. In practice, many of these steps (e.g., $\mathbf{S}$ and λ_v) need to be computed only once, amortizing the cost over multiple runs.

Parallelization. Since each node's H-PPR can be computed independently, we can partition the node set $\mathcal{V}$ across p processors: $\mathcal{V}$=$\bigcup_{i=1}^{p}\mathcal{V}_i$, achieving up to $\frac{1}{p}$ speedup (neglecting communication overhead C_{comm}). Both neighbor-weight construction and power iteration steps can be parallelized with sparse matrix operations.

4 Heterophily-Aware Subgraph Extraction

We present four approaches for extracting node-centric subgraphs reflecting meaningful relationships specific to a target node. Balancing structural similarity and diversity based on local heterophily, these strategies produce representations of both homophilous and heterophilous interactions:

1. Adaptive Top-k H-PPR. Given the H-PPR scores $\{\pi_v[u]\}$ for a node v, we extract nodes with *significant* scores:

$$\mathcal{N}(v) \;=\; top_{k_v}\big(\{\,u \in V : u \neq v\},\; \pi_v[u]\big) \tag{12}$$

where k_v adapts to each node's influence pattern:

$$k_v = \lceil p \cdot |\{u : \pi_v[u] > \epsilon\}|\rceil \tag{13}$$

Here, ϵ is a significance threshold (10^{-4}) filtering negligible relationships, and p (typically 0.4-0.6) controls the proportion of *influential* nodes to include. This adaptive approach ensures the subgraph size naturally scales with each node's effective influence range.

2. Adaptive Diversity-Aware. This strategy iteratively builds *a diverse subgraph* by selecting nodes that balance both structural importance and feature diversity. Starting from a seed node v, each new node u^* is chosen to maximize the sum of its PPR score and its minimum feature distance $d(u,w)$ from currently selected nodes, weighted by the local heterophily coefficient λ_v:

$$u^* = \arg\max_{u\in V\setminus S}\Big(\pi_v[u] + \lambda_v \cdot \min_{w\in S} d(u,w)\Big) \tag{14}$$

where S is the set of selected nodes (initialized with v), $d(u,w)$ measures feature distance between nodes, and λ_v controls the diversity-similarity trade-off.

3. Adaptive Threshold. Here, the subgraph size depends on the H-PPR score distribution and local properties:

$$\mathcal{N}_{\text{adaptive}}(v) = \{\,u : \pi_v[u] > \mu_v \;+\; \alpha_v\,\sigma_v\,\}, \tag{15}$$

where μ_v and σ_v are the mean and standard deviation of $\{\pi_v[u]\}$, and $\alpha_v \;=\; \delta(1-\lambda_v) + \lambda_v$ with $\delta \;>\; 1$ being a hyper-parameter controlling the amplification of the homophilous component. When $\lambda_v = 1$ (high heterophily) we have $\alpha_v = 1$, while for $\lambda_v = 0$ (high homophily) we get maximum amplification $\alpha_v = \delta$.

4. Structure-Preserving. We balance H-PPR scores with structural connectivity while adapting to local properties as: (1) Initialize S=$\{\arg\max_{u\neq v}\pi_v[u]\}$; (2) Iteratively select nodes maximizing:

$$\text{score}(u) = (1-\beta_v)\,\pi_v[u] \;+\; \beta_v \;\cdot\; \frac{|\{\,w \in S : (u,w) \in \mathcal{E}\}|}{|S|}$$

where $\beta_v \;=\; \gamma\,\big(1-\lambda_v\big) \;+\; (1-\gamma)\,\lambda_v$ with $\gamma \in [0,1]$ being a hyperparameter to control how much weight is given to homophilous $(1-\lambda_v)$ versus heterophilous (λ_v) relationships in the score computation. This produces subgraphs that remain well-connected while balancing homophilous and heterophilous relationships.

5 Feature Transformation and Classifier

We process nodes in batches for efficient feature transformation and classification. For each batch $\mathcal{B} \;=\; \{v_1,\ldots,v_b\}$, we extract node-centric subgraphs $\{\mathcal{G}_1,\ldots,\mathcal{G}_b\}$ using one of the above strategies (§4). Each subgraph $\mathcal{G}_i = (\mathcal{V}_i,\mathcal{E}_i)$ contains up to m nodes most relevant to v_i. Although we can use any architecture for feature transformation (§ 7.2), we focus on SGC [Wu *et al.*, 2019] for its effectiveness and efficiency. For subgraph $\mathcal{G}_i$, we define the normalized adjacency $\hat{\mathbf{A}}_i = \mathbf{D}_i^{-1/2}\mathbf{A}_i\,\mathbf{D}_i^{-1/2}$. The batch transformation is:

$$\mathbf{H}_{\mathcal{B}} = \begin{bmatrix}\hat{\mathbf{A}}_1^{K}\,\mathbf{X}_1\,\mathbf{W}\\ \vdots\\ \hat{\mathbf{A}}_b^{K}\,\mathbf{X}_b\,\mathbf{W}\end{bmatrix}, \tag{16}$$

where $\mathbf{X}_i$ is the subgraph's feature matrix and $\mathbf{W}$ is a learnable weight matrix.

Classifier. After obtaining node representations, we perform node classification. Although our framework supports any classifier, we demonstrate its effectiveness using simpler models. This choice highlights that the benefit of our approach stems from the high-quality subgraphs and transformations produced by H-PPR, rather than from a complex classifier. As shown by our experiments (§7), even simple classifiers can yield strong performance.

6 Theoretical Analysis

We now highlight some key theoretical properties of our proposal. Each result shows how local (λ_v) and global (λ_{global}) heterophily measures govern the random-walk transition dynamics, ensuring that both similar and dissimilar relationships are preserved. This dual emphasis makes H-PPR particularly well-suited for graphs where homophily is not the dominant pattern or where the structure varies regionally. A more comprehensive discussion is available online[1].

Theorem 6.1 (Local Heterophily Bounds)**.** *For any node v, its local heterophily coefficient λ_v is bounded by:*

$$\lambda_v \leq \max\{h_{\textit{label}}(v), h_{\textit{feat}}(v)\}$$

where $h_{\textit{label}}(v)$ is the label-based heterophily and $h_{\textit{feat}}(v)$ is the feature-based heterophily.

This theorem provides an upper bound on the local heterophily coefficient λ_v for any node v. The intuition is that since λ_v is computed as a weighted average of label-based and feature-based heterophily: $\lambda_v = \gamma \cdot h_{\text{label}}(v) + (1-\gamma) \cdot h_{\text{feat}}(v)$ it cannot exceed either component. This shows that our local heterophily measure is well-behaved and bounded by the structural properties of the node's neighborhood.

Lemma 6.2 (Adaptive Restart Property)**.** *For a graph with global heterophily level λ_{global}, the probability of restarting at a structurally diverse node is proportional to λ_{global}:*

$$P(\textit{diverse restart}) \propto \lambda_{global} \cdot \frac{\textit{entropy}(\mathcal{N}(v))}{\sum_{u \in \mathcal{V}} \textit{entropy}(\mathcal{N}(u))}$$

This lemma says how the global heterophily level influences the restart distribution. The key insight is that the probability of restarting at diverse nodes (those with high label entropy in their neighborhood) is directly proportional to λ_{global}.

Theorem 6.3 (Homophily-Heterophily Duality)**.** *For any node v with local heterophily coefficient λ_v and neighborhood $\mathcal{N}(v)$, H-PPR's transition probabilities satisfy:*

$$\sum_{u \in \mathcal{N}(v)} \widetilde{P}_{vu} \cdot \textit{sim}(v,u) = 1 - \lambda_v$$

$$\sum_{u \in \mathcal{N}(v)} \widetilde{P}_{vu} \cdot \textit{dissim}(v,u) = \lambda_v$$

where $\textit{sim}(v,u) + \textit{dissim}(v,u) = 1$ for all pairs (v,u).

This theorem establishes a fundamental relationship between similarity and dissimilarity in the transition probabilities. It shows that for any node, the expected similarity and dissimilarity of its transitions sum to 1, weighted by its local heterophily coefficient.

Lemma 6.4 (Heterophily Probability)**.** *The probability of selecting heterophilous neighbors in H-PPR is locally proportional to λ_v and globally proportional to λ_{global}.*

This lemma connects local and global heterophily measures to transition probabilities. This is crucial because it shows how H-PPR adapts at different scales.

Theorem 6.5 (Preservation of Structure)**.** *For a graph with heterophily ratio $\mathcal{H}(G)$, H-PPR adaptively preserves both heterophilous and homophilous relationships through:*

$$P(\textit{heterophilous step}) \propto \max\{\lambda_v, \lambda_{global}, \mathcal{H}(G)\}$$

$$P(\textit{homophilous step}) \propto \max\{1-\lambda_v, 1-\lambda_{global}, 1-\mathcal{H}(G)\}$$

This theorem shows how H-PPR maintains both heterophilous and homophilous patterns. This demonstrates the algorithm's ability to adapt to different graph structures.

7 Experimental Evaluation

We describe the datasets and the experimental setting (§7.1). We compare HAPPY with the state-of-the-art (§7.2) providing also an ablation analysis. Then, we evaluate subgraph extraction strategies (§7.3) and PageRank alternatives (§7.4).

7.1 Datasets and experimental setting

We considered state-of-the-art hetherophilous datasets [Platonov *et al.*, 2023b], which addressed issues in standard benchmarks for heterophily graphs, such as duplicates and imbalances, in datasets like Squirrel, Chameleon, Texas, Wisconsin, and Cornell [Zheng *et al.*, 2022]. These enhanced datasets are larger and cover a broader range of domains, as summarized in Table 2.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	C	F	$\mathcal{H}_{\text{edge}}$	$\mathcal{H}_{\text{adjusted}}$
Roman-empire	22662	32927	18	300	0.05	-0.05
Minesweeper	10000	39402	2	7	0.68	0.01
Wiki-cooc	10000	2243042	5	100	0.34	-0.03
Questions	48921	153540	2	301	0.84	0.02
Tolokers	11758	519000	2	10	0.59	0.09
Amazon-ratings	24492	93050	5	300	0.38	0.14
arXiv-year	169343	1166243	5	128	0.22	0.01
Squirrel-filtered	2223	46998	5	2086	0.20	0.12
Chameleon-filtered	183	280	5	2325	0.23	0.14

Table 2: Dataset statistics. We used the filtered version of squirrel and chameleon where duplicate nodes are removed.

Dataset splits. We used the dataset splits provided by [Platonov *et al.*, 2023b] and available online[2] The authors fix 10 random 50%/25%/25% train/validation/test splits. All models' performances are obtained by computing the averaged results and the standard deviation over the splits.

Implementation and Hyperparameters. We implemented[1] in PyTorch[3] and MLX[4] and integrated it into the evaluation pipeline provided by Platonov et al. [Platonov *et al.*, 2023b], which includes many baselines and facilitates the overall evaluation process. The pipeline also allows tuning HAPPY's hyperparameters based on the validation performance. We tuned random walk controls (α, $\beta \in [0.1, 0.9]$), computational settings (max_iter $\in [100, 1000]$, $\epsilon \in [10^{-6}, 10^{-8}]$), and SGC iterations K (2-4). A detailed ablation analysis is discussed below. We ran experiments on a Mac Studio M2 Ultra with a 24-core CPU, 60-core GPU, and 32-core Neural Engine with 192GB of unified memory.

[1] https://github.com/giuseppepirro/happy

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

[3] https://pytorch.org

[4] https://github.com/ml-explore/mlx

Group	**Approach**	**SquirrelF**	**ChameleonF**	**Wiki-cooc**	**RomanEmp**	**AmazonR**	**Minesweeper**	**Tolokers**	**Questions**	**arXiv-year**
Multi-hop view	H_2GCN	32.98 ± 1.15	25.95 ± 3.45	94.98 ± 0.35	56.28 ± 0.48	34.18 ± 0.25	88.95 ± 0.33	69.32 ± 0.34	63.21 ± 1.38	49.46 ± 0.16
	FSGNN	32.66 ± 1.28	39.27 ± 2.92	87.89 ± 0.45	74.72 ± 0.56	51.94 ± 0.82	83.24 ± 0.72	78.13 ± 0.65	70.62 ± 0.88	54.62 ± 0.72
Global Homophily	GloGNN	35.01 ± 1.18	23.65 ± 3.35	89.60 ± 0.42	59.88 ± 0.72	34.95 ± 0.16	49.23 ± 1.18	72.17 ± 1.22	58.97 ± 1.12	49.94 ± 3.28
	CPGNN	30.31 ± 2.12	33.27 ± 3.22	85.16 ± 1.08	60.32 ± 0.55	39.28 ± 0.71	50.19 ± 5.48	71.33 ± 5.52	62.13 ± 1.92	45.51 ± 2.28
Discriminative Message passing	FAGCN	36.87 ± 2.22	39.19 ± 2.52	92.05 ± 0.38	60.90 ± 0.58	43.95 ± 0.32	84.76 ± 0.78	69.69 ± 0.68	71.47 ± 1.28	54.46 ± 2.18
	GBK-GNN	35.07 ± 1.58	36.79 ± 2.62	96.33 ± 0.39	66.12 ± 0.42	44.45 ± 0.68	89.26 ± 0.62	75.74 ± 0.62	66.04 ± 0.78	50.43 ± 1.22
New Neighbours	NLGNN	37.93 ± 2.08	35.99 ± 1.28	91.47 ± 0.32	72.24 ± 0.38	50.49 ± 0.18	85.24 ± 0.28	78.89 ± 0.25	71.61 ± 0.98	45.73 ± 1.28
	OGNN	38.24 ± 1.22	<u>41.54</u> ± 1.52	92.78 ± 0.28	81.42 ± 0.15	<u>52.70</u> ± 0.15	87.82 ± 0.19	83.27 ± 0.19	75.50 ± 0.92	49.64 ± 0.92
Spectral	GPR-GNN	36.56 ± 1.89	38.55 ± 3.32	90.49 ± 0.42	60.68 ± 0.29	42.96 ± 0.36	78.00 ± 0.58	68.18 ± 0.65	54.58 ± 0.82	45.47 ± 0.28
	BernNet	31.76 ± 0.15	31.79 ± 0.14	78.73 ± 0.82	65.68 ± 1.15	33.11 ± 0.14	88.75 ± 0.15	72.86 ± 0.52	72.23 ± 1.28	49.75 ± 2.18
	JacobiConv	27.96 ± 1.72	37.30 ± 4.12	95.16 ± 0.38	66.53 ± 0.48	42.09 ± 0.52	80.31 ± 0.38	63.55 ± 0.42	68.32 ± 1.18	53.42 ± 2.28
	Restruct-GCN	34.48 ± 0.48	39.94 ± 0.55	89.42 ± 0.95	84.47 ± 0.52	50.08 ± 1.15	82.64 ± 0.68	<u>87.46</u> ± 0.82	<u>76.57</u> ± 0.38	47.42 ± 1.15
	ACM-GCN	37.06 ± 0.28	38.77 ± 0.28	86.95 ± 0.98	67.43 ± 1.28	37.78 ± 1.18	90.68 ± 0.32	78.28 ± 1.08	74.48 ± 1.28	53.81 ± 1.18
Adaptive	CDE	<u>39.42</u> ± 0.68	39.03 ± 0.62	<u>97.08</u> ± 0.35	<u>90.48</u> ± 0.25	46.62 ± 0.35	<u>95.76</u> ± 1.18	83.00 ± 0.58	74.08 ± 0.58	<u>57.42</u> ± 0.92
	ACMP	34.23 ± 0.52	39.36 ± 0.78	92.56 ± 0.56	71.43 ± 0.35	44.34 ± 0.57	74.05 ± 1.44	74.15 ± 0.72	71.82 ± 0.64	45.23 ± 0.66
	HeterGCL	35.36 ± 0.26	38.64 ± 0.45	87.64 ± 0.65	70.35 ± 0.54	46.16 ± 0.49	73.97 ± 1.36	75.07 ± 0.65	70.27 ± 1.13	44.62 ± 0.58
Ours	HAPPY$_{best}$	**43.48** ± 0.56	**43.66** ± 0.43	**97.58** ± 0.23	**90.89** ± 0.11	55.14 ± 0.04	**95.82** ± 0.22	**88.97** ± 0.11	**78.24** ± 0.33	**59.34** ± 0.72
Ablation Study 1: Feature Transformation Analysis (fixed two-layer feed-forward-network used as classifier).										
Feature transformation approach	SGC (k=2)	41.00 ± 0.31	40.50 ± 0.45	96.50 ± 0.25	89.90 ± 0.12	53.90 ± 0.05	95.50 ± 0.21	88.50 ± 0.11	77.00 ± 0.30	58.90 ± 0.11
	SGC (k=4)	40.95 ± 0.29	40.45 ± 0.42	96.40 ± 0.28	89.80 ± 0.15	53.80 ± 0.06	95.48 ± 0.20	88.45 ± 0.12	76.85 ± 0.31	58.75 ± 0.10
	GCN (l=2)	40.85 ± 0.33	40.00 ± 0.48	96.00 ± 0.22	89.50 ± 0.14	53.50 ± 0.07	95.30 ± 0.22	88.30 ± 0.13	76.70 ± 0.32	58.60 ± 0.12
	GCN (l=3)	40.80 ± 0.34	39.95 ± 0.50	95.90 ± 0.24	89.40 ± 0.16	53.40 ± 0.08	95.25 ± 0.24	88.25 ± 0.14	76.55 ± 0.33	58.45 ± 0.11
	GAT (l=2)	40.75 ± 0.36	39.80 ± 0.52	95.70 ± 0.26	89.20 ± 0.18	53.20 ± 0.09	95.15 ± 0.23	88.15 ± 0.15	76.40 ± 0.34	58.30 ± 0.10
	GAT (l=3)	40.70 ± 0.37	39.70 ± 0.54	95.60 ± 0.28	89.10 ± 0.19	53.10 ± 0.10	95.10 ± 0.25	88.10 ± 0.16	76.30 ± 0.35	58.20 ± 0.11
Ablation Study 2: Classifier Analysis (fixed feature transformation via SGC with k=3).										
Classifier	FFW$_3$	41.80 ± 0.72	43.20 ± 0.47	97.00 ± 0.26	90.75 ± 0.13	54.10 ± 0.09	95.70 ± 0.22	88.80 ± 0.14	77.20 ± 0.65	58.20 ± 0.13
	Logistic	42.13 ± 0.70	43.34 ± 0.45	97.18 ± 0.25	90.61 ± 0.12	54.27 ± 0.08	95.78 ± 0.21	88.92 ± 0.12	77.34 ± 0.64	58.34 ± 0.12
	SVM	41.95 ± 0.72	43.12 ± 0.50	96.98 ± 0.28	90.80 ± 0.14	54.05 ± 0.10	95.65 ± 0.23	88.75 ± 0.15	77.10 ± 0.68	58.10 ± 0.14

Table 3: Node classification results; accuracy (Wiki-cooc, Roman-empire, Amazon-ratings, Squirrel-F, Chameleon-F) and ROC-AUC scores (Minesweeper, Tolokers, Questions, arXiv-year). Bold and underlined indicate best and second-best results. The two ablation studies compare different feature transformations (with fixed logistic regression classifier) and classifiers (with fixed feature transformation SGC with k=3).

7.2 Comparison with the state-of-the-art

For sake of space, we directly consider models specifically crafted for node classification in heterophilous contexts and do not report, for instance, methods such as ResNet [He *et al.*, 2016] and GNN-variants [Kipf and Welling, 2017]) that have been shown to be non-competitive in heterophilous settings as shown in [Platonov *et al.*, 2023b]. We compared HAPPY against 16 state-of-the-art methods using their publicly accessible implementations. Results are in Table 3.

HAPPY$_{best}$, our best performing variant, uses SGC (k=3) as feature transformation, adaptive diversity as subgraph extraction mechanism (§4), and a FFW with two layers as classifier. HAPPY$_{best}$ consistently outperforms state-of-the-art methods across all benchmarks, with particularly notable improvements on challenging heterophilous datasets. On small-scale heterophilous graphs like Squirrel-filtered (2,223 nodes) and Chameleon-filtered (183 nodes), our method achieves significant gains of 4.28% and 4.02% respectively over the best baseline (OGNN). The superiority of HAPPY is even more evident on large-scale datasets: it surpasses CDE by 0.8% on Wiki-cooc (10,000 nodes, $\mathcal{H}_{\text{edge}} = 0.34$) and shows remarkable improvement on highly heterophilous graphs like Questions (48,921 nodes, $\mathcal{H}_{\text{edge}} = 0.84$), outperforming ACM-GCN by 4.06%. These results demonstrate that our heterophily-aware PPR effectively captures both local and global patterns regardless of graph size or heterophily levels ($\mathcal{H}_{\text{edge}}$ ranging from 0.05 to 0.84). Notably, HAPPY maintains its strong performance even on challenging datasets, such as Roman-empire ($\mathcal{H}_{\text{adjusted}} = -0.05$), demonstrating its versatility across different graph structures. Our ablation studies provide key insights into the components of HAPPY. In Study 1, we analyze different feature transformation approaches while keeping the classifier fixed. SGC with k=2 generally performs best, achieving superior results across most datasets (e.g., 96.50% on Wiki-cooc). Notably, performance gradually decreases with higher k SGC values or when using more complex architectures like GCN/GAT, suggesting that simpler transformations better preserve heterophilous patterns. Study 2 examines classifier choices while fixing the feature transformation to SGC (k=3). The results show minimal variation across classifiers (e.g., on Wiki-cooc: Logistic 97.18%, SVM 96.98%), with Logistic Regression performing marginally better. This indicates that adaptive diversity-based subgraph extraction captures the graph's structural properties, making the choice of classifier less critical.

7.3 Evaluating subgraph extraction strategies

We now evaluate the effectiveness of each subgraph extraction strategy (§ 4). Fig. 1 provides a comprehensive analysis of our four subgraph extraction strategies across diverse datasets. The Adaptive Diversity-aware approach consistently demonstrates superior performance. This validates our intuition that explicitly balancing structural importance with feature diversity through λ_v effectively captures heterophilous patterns. The Adaptive Top-k H-PPR strategy shows competitive performance on homophilous datasets,

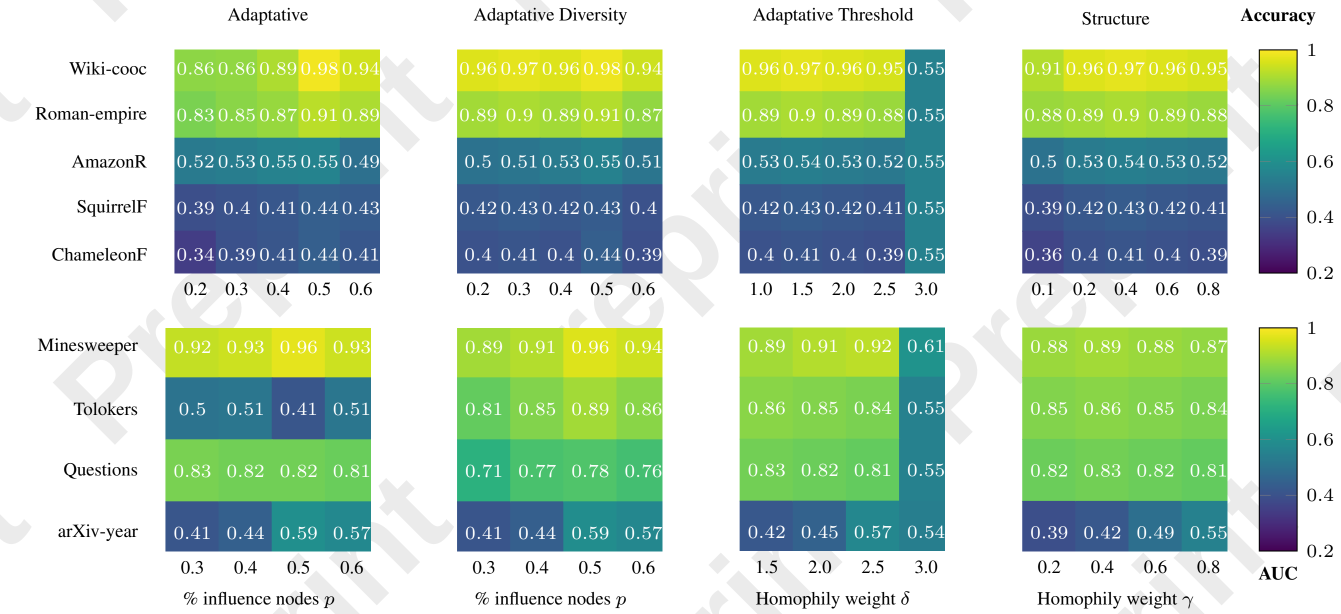

Figure 1: Subgraph extraction strategies using HAPPY$_{best}$. For adaptative and adaptative diversity we vary the % of nodes that have a HPPR score withing $\epsilon = 10^-4$. For adaptive threshold, $\alpha_v = \delta(1 - \lambda_v) + \lambda_v$. For structure preserving we have $\beta_v = \gamma(1 - \lambda_v) + (1 - \gamma)\lambda_v$.

achieving 0.96 accuracy on Minesweeper with $p = 0.4$, but slightly underperforms on more heterophilous graphs. The Adaptive Threshold approach, controlled by $\delta(1 - \lambda_v) + \lambda_v$, maintains stable performance across different homophily levels but shows sensitivity to the parameter, visible in the sharp performance drop at δ=3.0. The Structure-Preserving strategy, while ensuring well-connected subgraphs through β_v, demonstrates more consistent but generally lower performance across all datasets, suggesting that strictly maintaining structural connectivity might limit the capture of important heterophilous relationships. Notably, all strategies show improved performance compared to traditional homophily-based approaches, with the Adaptive Diversity-aware method providing the best balance between homophilous and heterophilous pattern recognition. These results highlight that subgraph extraction benefits significantly from *both* node-centric relevance (*via* H-PPR) *and* local heterophily-driven diversification, ensuring that each neighborhood reflects the most informative aspects of the target node's surroundings.

7.4 Comparison with Page Rank variants

We compare HAPPY$_{best}$ based on our Heterophily-aware PPR (H-PPR) against two baselines: Traditional PPR (T-PPR), which uses fixed transition probabilities based solely on graph structure, and Personalized PPR (P-PPR), which introduces node-specific bias terms but still assumes homophily in transitions. As shown in Table 4, H-PPR consistently outperforms both variants across all datasets, with particularly notable improvements on challenging heterophilous graphs. On SquirrelF and ChameleonF, H-PPR achieves gains of +2.17% and +1.77% over P-PPR respectively, demonstrating its effectiveness in capturing heterophilous relationships. The improvement is even more pronounced on large-scale datasets with mixed homophily patterns: H-PPR surpasses P-PPR by +2.69% on AmazonR ($\mathcal{H}_{\text{edge}} = 0.38$) and +2.35% on Questions ($\mathcal{H}_{\text{edge}} = 0.84$). Notably, H-PPR maintains its advantages on homophilous graphs like Wiki-cooc (+1.69%) and Minesweeper (+2.15%), indicating that our heterophily-aware modifications do not compromise performance when homophily dominates. The lower variance in H-PPR's results (e.g., ±0.04 on AmazonR compared to ±0.12 for P-PPR) suggests more stable performance, likely due to its adaptive balancing of structural and feature-based relationships.

Method	Wiki-cooc	RomanEmp	AmazonR	SquirrelF	ChameleonF	Minesweeper	Tolokers	Questions	arXiv-year
T-PPR	92.45	85.67	48.32	35.78	34.91	91.23	82.45	72.34	52.67
P-PPR	95.89	88.92	52.45	41.23	41.89	95.67	86.78	75.89	56.78
H-PPR	**97.58**	**90.89**	**55.14**	**43.40**	**43.66**	**97.82**	**88.97**	**78.24**	**59.34**
Improvement	+1.69	+1.97	+2.69	+2.17	+1.77	+2.15	+2.19	+2.35	+2.56

Table 4: Comparison of PPR variants across datasets.

These results validate our insights about integrating feature diversity into the random walk process, showing that H-PPR effectively captures both homophilous and heterophilous patterns while maintaining computational efficiency.

8 Concluding Remarks

We presented HAPPY, a framework for node classification in heterophilous graphs that adapts Personalized PageRank to balance structural and feature relationships. By integrating feature diversity into random walks and using adaptive subgraph extraction, HAPPY captures both homophilous and heterophilous patterns efficiently. Experiments show that HAPPY outperforms state-of-the-art methods. Future work could explore self-supervised learning applications and extensions to temporal graphs with evolving structures and features.

Acknowledgments

This work was partially supported by the European Union – Next Generation EU through the MUR Project HypeKG within PRIN 2022 Program (CUP: H53D23003710006), and the MUR PRIN 2022-PNRR project DISTORT (CUP: H53D23008170001) under the Italian PNRR Mission 4 Component 1.

References

[Bo *et al.*, 2021] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3950–3957, 2021.

[Chanpuriya and Musco, 2022] Sudhanshu Chanpuriya and Cameron Musco. Simplified graph convolution with heterophily. *Advances in Neural Information Processing Systems*, 35:27184–27197, 2022.

[Chien *et al.*, 2021] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[Du *et al.*, 2022] 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 Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, editors, *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1550–1558. ACM, 2022.

[Hamilton *et al.*, 2017] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034, 2017.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[He *et al.*, 2021] Mingguo He, Zhewei Wei, Hongteng Xu, et al. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34:14239–14251, 2021.

[He *et al.*, 2022] Mingguo He, Zhewei Wei, and Ji-Rong Wen. Convolutional neural networks on graphs with chebyshev approximation, revisited. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[Li *et al.*, 2022] 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*, pages 13242–13256. PMLR, 2022.

[Li *et al.*, 2023] Shouheng Li, Dongwoo Kim, and Qing Wang. Restructuring graph for higher homophily via adaptive spectral clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8622–8630, 2023.

[Liu *et al.*, 2021] Meng Liu, Zhengyang Wang, and Shuiwang Ji. Non-local graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[Luan *et al.*, 2022] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. *Advances in neural information processing systems*, 35:1362–1375, 2022.

[Luan *et al.*, 2024] Sitao Luan, Chenqing Hua, Qincheng Lu, Liheng Ma, Lirong Wu, Xinyu Wang, Minkai Xu, Xiao-Wen Chang, Doina Precup, Rex Ying, et al. The heterophilic graph learning handbook: Benchmarks, models, theoretical analysis, applications and challenges. *arXiv preprint arXiv:2407.09618*, 2024.

[Maurya *et al.*, 2022] Sunil Kumar Maurya, Xin Liu, and Tsuyoshi Murata. Simplifying approach to node classification in graph neural networks. *Journal of Computational Science*, 62:101695, 2022.

[Pirrò, 2023] Giuseppe Pirrò. Overlay neural networks for heterophilous graphs. In *ECAI 2023 - 26th European Conference on Artificial Intelligence*, pages 1890–1897. IOS Press, 2023.

[Platonov *et al.*, 2023a] Oleg Platonov, Denis Kuznedelev, Artem Babenko, and Liudmila Prokhorenkova. Characterizing graph datasets for node classification: Homophily-heterophily dichotomy and beyond. In *The Second Learning on Graphs Conference*, 2023.

[Platonov *et al.*, 2023b] 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 *11th International Conference on Learning Representations, ICLR 2023, To appear*, 2023.

[Wang and Zhang, 2022] Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23341–23362. PMLR, 2022.

[Wang *et al.*, 2023] Yuelin Wang, Kai Yi, Xinliang Liu, Yu Guang Wang, and Shi Jin. ACMP: Allen-cahn message passing with attractive and repulsive forces for graph neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[Wang *et al.*, 2024] Chenhao Wang, Yong Liu, Yan Yang, and Wei Li. Hetergcl: Graph contrastive learning framework on heterophilic graph. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 2397–2405. ijcai.org, 2024.

[Wu *et al.*, 2019] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

[Yang *et al.*, 2021] Liang Yang, Mengzhe Li, Liyang Liu, Chuan Wang, Xiaochun Cao, Yuanfang Guo, et al. Diverse message passing for attribute with heterophily. *Advances in Neural Information Processing Systems*, 34:4751–4763, 2021.

[Yang *et al.*, 2022] Tianmeng Yang, Yujing Wang, Zhihan Yue, Yaming Yang, Yunhai Tong, and Jing Bai. Graph pointer neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8832–8839, 2022.

[Zhao *et al.*, 2023] K. Zhao, Q. Kang, Y. Song, R. She, S. Wang, and W. P. Tay. Graph neural convection-diffusion with heterophily. In *Proc. International Joint Conference on Artificial Intelligence*, Macao, China, Aug 2023.

[Zheng *et al.*, 2022] Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S Yu. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*, 2022.

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

[Zhu *et al.*, 2021] Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. Graph neural networks with heterophily. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11168–11176, 2021.

[Haveliwala, 2002] Taher H. Haveliwala. Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. In *Proceedings of the 11th International World Wide Web Conference (WWW)*, pp. 517–526, Honolulu, Hawaii, USA, 2002.