# Beyond Homophily: Structure-aware Path Aggregation Graph Neural Network

**Yifei Sun**[1] , **Haoran Deng**[1] , **Yang Yang**[1*] , **Chunping Wang**[2] , **Jiarong Xu**[3] ,
**Renhong Huang**[1] , **Linfeng Cao**[4] , **Yang Wang**[2] and **Lei Chen**[2]

[1]Zhejiang University, [2]Finvolution, [3]Fudan University, [4]Shanghai Jiao Tong University

{yifeisun, denghaoran, yangya, renh2}@zju.edu.cn, jiarongxu@fudan.edu.cn,
{wangchunping02, wangyang09, chenlei04}@xinye.com, linfengcao1996@gmail.com

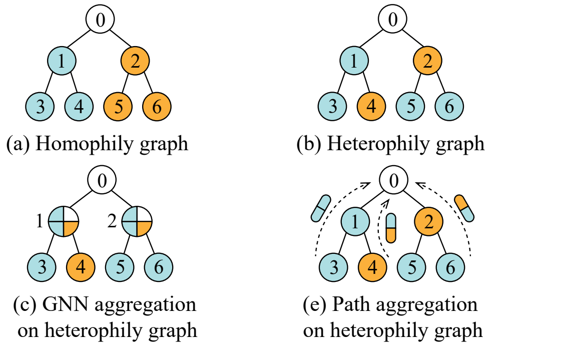
## Abstract

Graph neural networks (GNNs) have been intensively studied in various real-world tasks. However, the homophily assumption of GNNs' aggregation function limits their representation learning ability in heterophily graphs. In this paper, we shed light on the path level patterns in graphs that can explicitly reflect rich semantic and structural information. We therefore propose a novel Structure-aware Path Aggregation Graph Neural Network (PathNet)[1] aiming to generalize GNNs for both homophily and heterophily graphs. Specifically, we first introduce a maximal entropy path sampler, which helps us sample a number of paths containing structural context. Then, we introduce a structure-aware recurrent cell consisting of order-preserving and distance-aware components to learn the semantic information of neighborhoods. Finally, we model the preference of different paths to target node after path encoding. Experimental results demonstrate that our model obtains significant improvements in node classification on both heterophily and homophily graphs.

## 1 Introduction

Graph neural networks (GNNs) have attracted considerable research interest recently [Wu *et al.*, 2020] due to their superior performance in various applications, such as bioinformatics [Borgwardt *et al.*, 2005], finance [Yang *et al.*, 2019b], chemistry [Xiong *et al.*, 2019], social network analysis [Yang *et al.*, 2019a], etc. The typical framework of GNNs can be formulated as message passing, which means node representations are learned by recursively aggregating the neighborhood.

Despite the promising results achieved by GNNs, most of them inevitably assume *homophily*, that is, the connected nodes tend to have similar attributes or belong to the same class ("birds of a feather flock together") [McPherson *et al.*, 2001] as the example shown in Fig. 1 (a). However, numerous graphs exhibit the "opposites attract" phenomenon, which conflicts with the homophily assumption of GNNs. As the example in Fig. 1 (b) shown, the connected nodes tend to have dissimilar attributes or belong to different classes which leads to the formation of *heterophily* graphs [Newman, 2003]. For instance, in a heterosexual dating network, a candidate tends to be attracted by one of the opposite gender.



(a) Homophily graph

(b) Heterophily graph

(c) GNN aggregation on heterophily graph

(e) Path aggregation on heterophily graph

Figure 1: (a) & (b) show patterns of homophily and heterophily graphs. (c) demonstrates the GNN aggregation process for node 0 of (b). (d) our proposed path aggregation compared with (c).

Although GNNs perform well in homophily graphs, they cannot achieve satisfactory performance in heterophily graphs. Fig. 1 (c) gives us a potential explanation: to aggregate the neighborhood of node 0, it is required to aggregate node 1's neighbors and node 2's neighbors first. Since the receptive field of node 1 is only limited to the direct neighbor 0, 3, 4, and the receptive field of node 2 is limited to node 0, 5, 6, the aggregated features of node 1 and 2 becomes (⊕) and (⊕), which are indistinguishable to node 0. However, the original node features of node 1 and 2 are different, indicating the flawed aggregation process of GNNs.

In this paper, we shed light onto the *path level patterns* in a graph, which can model both homophily and heterophily graphs. First, due to the flexibility of the paths, both direct adjacent and higher-order neighborhood information can be retained rather than only depending on the first-order neighboring nodes for aggregation. As depicted in Fig. 1(d), the path on the left contains node 1 and 3, which cannot be aggregated to node 0 simultaneously by GNN. Second, due to the order information brought by the paths, contextual semantic information can be captured to construct distinguishable em-

[*]Corresponding author.

[1]Codes are available at https://github.com/zjunet/PathNet

bedding. Different from Fig. 1(c), our method gathers three types of paths: two of them go through node 1, forming (🟦🟦) by node 1 and 3, (🟦🟧) by node 1 and 4; the other one goes through node 2, forming (🟧🟦) by node 2 and 6, which are quite different for node 0. Although the paths of (🟦🟧) and (🟧🟦) contains two similar node features, the path embedding is still distinguishable because the order is different.

Based on aforementioned points, one natural implication is to learn each node's embedding by aggregating all involved paths instead of its neighborhoods. However, the question of how to obtain and encode the proper paths for extracting sufficient information is quite challenging.

Firstly, to obtain diverse path information, one straightforward way is to enumerate all paths for each node. However, the number of paths grows exponentially in the receptive field, especially when hub nodes are traversed. Thus, the first challenge is how to define an appropriate sampler that can avoid the over-expansion issue while retaining meaningful structural information.

Secondly, as mentioned above, the order of node appearance in a path is critical for capturing the heterophily in neighborhood. However, most of existing GNNs aggregators are not order sensitive and can not explicitly distinguish the information from different hops. Thus, how to design an order-preservable aggregator is the second challenge.

Thirdly, nodes give preference to specific paths. For example, paths near the target node are preferred by homophily neighborhoods, whereas paths exploring deep receptive fields are preferred by heterophily ones. Nonetheless, one actually can hardly judge whether a graph is homophily or not with insufficient label information. Thus, the third challenge is how to capture the path preference of different nodes.

In this work, we introduce a structure-aware path aggregation graph neural network (abbreviated as PathNet) to address these challenges. To alleviate the over-expansion issue and to preserve structural patterns, we introduce a maximal entropy path sampler. Moreover, we theoretically and empirically prove that sampling an increasing amount of paths approaches the infinite paths scenario at an exponential rate. To encode paths while retaining the context of higher-order neighborhoods, we introduce a structure-aware path encoder which possesses two advantages: preserving the order through recurrent mechanism and capturing the contextual structure by leveraging distance to the target node. After the path encoding, we propose a path attention mechanism to model the preference of paths for nodes with different homophily level of neighborhoods.

The main contributions of our work are as follows:

(1) We propose a novel path aggregation paradigm to capture the structure context information in both homophily and heterophily graphs.

(2) We design an end-to-end model PathNet which leverages order and distance information of path to encode complex semantic information in graphs.

(3) Extensive experimental results demonstrate that our model achieves superior performance by up to 10.27% in the node classification task on heterophily graphs and competitive results on homophily graphs.

## 2 Related work

GNN methods have attracted considerable attention. GAT [Veličković *et al.*, 2018] introduces the attention mechanism to parameterize the aggregation function. GIN [Xu *et al.*, 2019] proposes a graph model with the same expressive power as WL graph isomorphism test. However, a majority of these works are based on the homophily assumption [Yang *et al.*, 2018; Wu *et al.*, 2017]. Thus, these approaches fail to achieve satisfactory performance on heterophily graphs.

Recently, several models have emerged to cope with the challenging and largely overlooked setting of heterophily. MixHop [Abu-El-Haija *et al.*, 2019] utilizes higher-order graph convolutional architectures to overcome the limitation of direct aggregation. Likewise, H2GCN [Zhu *et al.*, 2020] proposes to combine three components: ego- and neighbor embedding separation, higher-order neighborhoods and intermediate embedding. GPRGNN [Chien *et al.*, 2021] leverages the diffusion matrix for long-distance propagation and proposes a pagerank-based model under heterophily settings. FAGCN [Bo *et al.*, 2021] puts forward high-pass and low-pass filters to deal with both homophily and heterophily graphs. Although some of them propose to utilize the higher-order neighborhoods in aggregation process, they fail to capture the intrinsic semantic information since the order information are omitted. Furthermore, most of them treat the nodes in higher-order neighborhoods equally since they directly aggregate these nodes without distance information.

To the best of our knowledge, there are two existing works that claim to employ the path to model graphs. GeniePath [Liu *et al.*, 2019] proposes an adaptive path layer that can guide the breadth and depth exploration of the receptive fields. Nonetheless, instead of utilizing the concrete paths, they name the receptive fields as receptive paths. SPAGAN [Yang *et al.*, 2019c] conducts the operation of path-based higher-order attention to explore the the topological information. However, each node influences the target node only through the shortest distance and omit the structure of higher-order neighborhoods. In general, feature propagation in these models fails to capture the connections between direct neighbors of target nodes and fails to encode different structure into distinguishable embedding, which results in the unsatisfactory performance.

## 3 PathNet

We begin with some necessary definitions.

**Problem definition.** Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V}$ is the node-set and $\mathcal{E}$ is the edge-set. The adjacency matrix and the node feature matrix of $\mathcal{G}$ is denoted by $\mathbf{A} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$ and $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times F}$ respectively, where $F$ is the dimension of feature for each node. In this paper, we focus on the semi-supervised node classification task. Specifically, given nodes in training set $\mathcal{T}_{\mathcal{V}}$ with known labels $y_v$ and feature vectors $\mathbf{x}_v$ for $v \in \mathcal{V}$, we aim to infer the unknown $y_u$ for all $u \in (\mathcal{V} - \mathcal{T}_{\mathcal{V}})$.

**Homophily and heterophily.** The concept of homophily is derived from the tendency of a node to have the same class as its neighbor. The homophily level can be quanti-

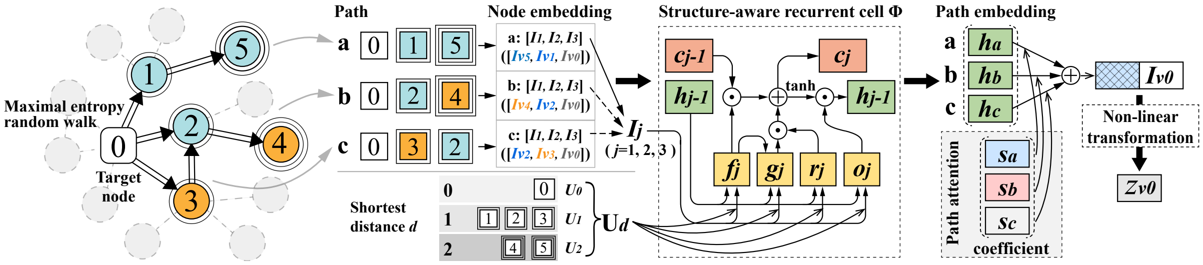

Path
Node embedding
Structure-aware recurrent cell Φ
Path embedding
Maximal entropy random walk
Target node
a: [I1, I2, I3] ([Iv5, Iv1, Iv0])
b: [I1, I2, I3] ([Iv4, Iv2, Iv0])
c: [I1, I2, I3] ([Iv2, Iv3, Iv0])
Ij (j=1, 2, 3)
Shortest distance d
U0
U1
U2
Ud
cj-1
hj-1
cj
tanh
fj
gj
rj
oj
ha
hb
hc
Path attention coefficient
Sa
Sb
Sc
Iv0
Non-linear transformation
Zv0

Figure 2: The workflow of PathNet for node classification of node 0 with the walk length $k = 3$ . The color of node stands for label and the number stands for node index.

fied by homophily edge ratio [Zhu *et al.*, 2020], denoted as $h(\mathcal{G}) = \frac{|\{(u,v):(u,v)\in\mathcal{E}\wedge y_u=y_v\}|}{|\mathcal{E}|}$, which calculates the proportion of the edges connect two nodes with the same label. The lower $h(\mathcal{G})$ implies more edges connecting nodes in different classes, that is, stronger heterophily.

The paths have great potential to represent the complex semantic information of graphs. Thus we propose PathNet (Fig. 2) to utilizes the path aggregation paradigm aiming to generalize GNNs for both homophily and heterophily graphs. The paths are firstly sampled under the guidance of maximal entropy random walk, which effectively explores the diversity of the graph structure. Then the paths are encoded by sending the node embedding sequences into the *structure-aware recurrent cell* $\Phi$ with distance-aware component, which captures structural context and extracts semantic information. To model the preference of each path embedding to the target node, a path attention module is employed for path aggregation. Finally the node prediction $z_{v0}$ can be obtained after a non-linear transformation.

## 3.1 Maximal Entropy Path Sampler

To obtain the paths with a consideration of the efficiency, a sampling strategy like random walk is required. However, the conventional random walk (CRW) suffers from the identical treatment to different nodes and ignores the centrality of nodes of a graph. To address these problems, we propose to sample paths under the guidance of the maximal entropy random walk (MERW), shown in Fig. 2, left. The MERW seeks the paths in the orientation of entropy rate increase for each step and incorporates the eigenvector centrality which is widely applied to measure the importance of the nodes.

Consider a path sampled with length $k$ from $v_i$ to $v_j$, the probability of the path is $\mathcal{P}^k_{ij} = p_{ii_1}p_{i_1i_2}\cdots p_{i_{k-2}i_{k-1}}p_{i_{k-1}j}$, where $p_{ij}$ is a element in transition matrix. As declared in [Cover, 1999], the maximal entropy rate $\eta$ of random walk on a graph can be computed from the transition matrix $\mathcal{P}$ and the stationary distribution $\pi$, which can be described as follows:

$$\eta = -\sum_i \pi_i \sum_j p_{ij} \ln p_{ij}. \tag{1}$$

Moreover, the maximal entropy rate of random walk is bounded by $\ln \lambda$ [Parry, 1964], where $\lambda$ is the largest eigenvalue of $\mathbf{A}$. In order to maximize the entropy rate of a walk, MERW constructs the transition probabilities as $p_{ij} = \frac{A_{ij}u_j}{\lambda u_i}$ [Burda *et al.*, 2009], where $u = (u_1, u_2, \cdots, u_n)$ is the normalized eigenvector. Note that the transition probabilities are proportional to the eigenvector centrality guaranteeing the MERW the ability to capture the structural context of nodes in a graph. It can be reformulated into the maximal entropy transition (MET) matrix, which is defined as

$$\mathbf{P}_u = \frac{\mathbf{D}_u^{-1}\mathbf{A}\mathbf{D}_u}{\lambda}, \tag{2}$$

where $\mathbf{D}_u = \mathrm{diag}(u_1, u_2, \cdots, u_n)$. In short, maximal entropy path sampler not only maximizes the entropy rate but also obtains the complex structual information of a graph.

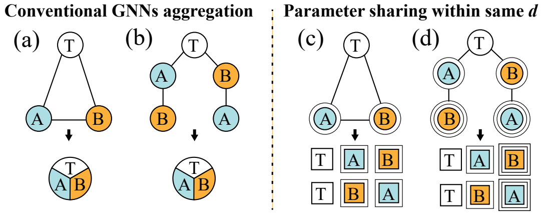

Conventional GNNs aggregation
Parameter sharing within same d
(a)
(b)
(c)
(d)
T
A
B

Figure 3: Intuitive case where GNNs are not able to distinguish the target nodes in (a) & (b), while our path-based aggregation with parameter sharing mechanism can capture the topological information and make the embedding distinguishable (c) & (d). (Nodes with similar characteristic represented as the same notation.)

## 3.2 Structure-aware Path Aggregator

To aggregate the path information, we propose a structure-aware path aggregator. Specifically, we design a structure-aware recurrent cell to encode the path embedding, which is capable to incorporate the *order* and *distance* information of each node in the path, so as to capture the semantic information. Furthermore, we model the path preference to distinguish the importance of different paths and realize self-adaptive path embedding aggregation.

**Structure-aware recurrent cell.** Although conventional GNNs are capable of gathering the information of higher-order neighborhoods, they inevitably lead to indistinguishable embedding for treating them in the same way and overlook the global structure context of neighborhoods in graphs. Fig. 3 (a, b) gives an intuitive case that a GNN within two layers generates indistinguishable embeddings of the target node T in two graphs. Because from the GNNs' point of view, only the direct neighbors are visible for each node, and all nodes in (a) and top three nodes in (b) share the identical neighbors. Hence, the two graphs would generate the identical node embedding of T after two-layer aggregation. On the contrary, if we embed the distance to the target information into the directed path sequence (shown in Fig. 3 c, d), the structural information (i.e., closed triangle $vs.$ open paths) is well preserved and the path embedding is distinguishable. Inspired by this, we claim that the order and distance of each node are the key information for capturing the structural context through path.

Instead of aggregating each layer of nodes independently, we propose a recurrent cell to encode path sequence, so as to retain the order information of each node. We next introduce the details of our structure-aware recurrent cell. Note that we encode all the node feature $\mathbf{X}$ as node embedding $\mathbf{I}$ before engaging the structural information, which is described as

$$\mathbf{I} = \sigma\left(\mathbf{W_{in}}\mathbf{X} + \mathbf{b_{in}}\right), \tag{3}$$

where $\mathbf{W_{in}}$ and $\mathbf{b_{in}}$ are learnable parameters of the sigmoid function $\sigma$. In order to capture the feature-based diffusion structure, we construct structure-aware recurrent cell $\Phi$ as

$$\begin{aligned}
\mathbf{r}_j &= \sigma\left(\mathbf{W_r} \cdot h_{j-1} + \mathbf{U_d} \cdot \mathbf{I}_j\right), \\
\mathbf{f}_j &= \sigma\left(\mathbf{W_f} \cdot h_{j-1} + \mathbf{U_d} \cdot \mathbf{I}_j\right), \\
\mathbf{o}_j &= \sigma\left(\mathbf{W_o} \cdot h_{j-1} + \mathbf{U_d} \cdot \mathbf{I}_j\right), \\
\mathbf{g}_j &= \tanh\left(\mathbf{W_g} \cdot h_{j-1} + \mathbf{U_d} \cdot \mathbf{I}_j\right), \\
\mathbf{c}_j &= \mathbf{f}_j \odot \mathbf{c}_{j-1} + \mathbf{r}_j \odot \mathbf{g}_j, \\
\mathbf{h}_j &= \mathbf{o}_t \odot \tanh\left(\mathbf{c}_j\right),
\end{aligned} \tag{4}$$

where $h_j$ denotes the latent embedding of node $v$ at the $j$-th step on the path, $\odot$ is the Hadamard product and $\mathbf{r}_j, \mathbf{f}_j, \mathbf{o}_j$ represent the input, forget and output gates, respectively. As a part of the forget gate, $\mathbf{g}_j$ contributes to the long term memory cell $\mathbf{c}_j$ which filter the node feature along the paths. Considering the target node as the most basic part contributes the most, the input sequence is reversed from the collection order of the path (Fig. 2, Node embedding). Finally, we obtain the final path embedding $\mathbf{h}_p$ for the $p$-th path.

Considering that the path embedding is distance sensitive, we employ a parameter sharing mechanism based on the node distance to the path embedding computation. Specifically, for each node in a path, we use the pre-calculated shortest distance $d$ from the target node as side information, and the nodes with the same $d$ share the same $\mathbf{U_d}$ during path encoding (Eq. 4), so as to incorporate the distance information into path embeddings and make them distance-aware.

**Modeling path preference.** Naturally, neighborhoods with different homophily level have different path preferences. For homophily neighborhoods, the path nearby the target node may contribute more to the classification, while for heterophily ones, the paths to explore broader context of graph structure will be preferred. However, the homophily level of each neighborhood is hard to know. Thus, we model the path preference for each node, that is, take different paths into consideration and learn their importance to facilitate the self-adaptive embedding aggregation.

More specifically, for a particular path $p$, we obtain its embedding $\mathbf{h}_p$ after Eq. 4, which is then concatenated with the target node embedding $\mathbf{I}_v$ as the input for computing the preference coefficient $\boldsymbol{s}_{v,p}$ by a trainable weight $\boldsymbol{a}$. We calculation the preference coefficient $\boldsymbol{s}_{v,p}$ as

$$\boldsymbol{s}_{v,p} = \mathrm{SOFTMAX}\left(\delta\left(\boldsymbol{a}\left(\mathbf{I}_v \| \mathbf{h}_p\right)\right)\right), \tag{5}$$

where $\delta = \mathrm{LeakyReLU}$, and $\boldsymbol{a}$ is the trainable weights. In the last step, we use the preference coefficient $\boldsymbol{s}_{v,p}$ to weight the path and make the final prediction $\mathbf{z}_v$ by $\mathbf{W_{out}}$ and $\mathbf{b_{out}}$:

$$\mathbf{z}_v = \sigma\left(\mathbf{W_{out}}\left(\mathbf{I}_v \| \sum_{j \in p} \boldsymbol{s}_{v,p}\mathbf{h}_p\right) + \mathbf{b_{out}}\right). \tag{6}$$

The computational complexity of PathNet is discussed in the Appendix A.

## 4 Theoretical Analysis

### 4.1 Convergence of Path Sampler

As illustrated in Sec. 3.1, we propose to sample paths under the guidance of MERW that retains the structural information of the neighborhoods. Moreover, sampling various paths composed of different node features helps to capture more contextual patterns of the target node. In this section, we prove that sampling finite paths is indeed enough to cover the overall path patterns. Assigning the same tag $l$ to nodes with similar features, we can define an attributed path as follows: For different paths $\{v_1, v_2, ..., v_k\}$ and $\{v_a, v_2, ..., v_k\}$ of length $k$, if node features of $v_1$ and $v_a$ are similar, they both count as the same attributed path sequence of $\{l_1, l_2, ..., l_k\}$. Hence, an attributed path can be generated only if nodes with dissimilar features are introduced. The more attributed paths we sample, the more semantic information we can capture from the neighborhoods.

Attributed path sampling deviation is defined as the difference between the proportion of attributed paths from finite sampling and the real attributed path proportion. The exponential convergence relationship between the finite sampling number of paths and the probability of attributed path sampling deviation being smaller than a certain constant is established below:

**Theorem 1.** *Given a graph with adjacency matrix* $\mathbf{A}$*, for any* $\epsilon > 0$*, the probability of the attributed path sampling deviation being larger than* $\epsilon$ *decreases exponentially as the number of attributed paths* $M$ *increases.*

The proof can be found in Appendix B. The theory indicates that although our method samples a finite number of paths, the ability to capture the semantic information grows tremendously when the number of sampled paths increases, revealing the feasibility of gathering enough representative information of neighbors by sampling an achievable number of paths.

### 4.2 Expressive Power of PathNet

PathNet provides a more intact receptive field for target node and generates more distinguishable embedding. Neither stochastic GNNs like GraphSAGE nor deterministic GNNs like GCN can capture the connections between direct neighbors of a node. Because they use the permutation invariant operation such as mean or max pooling for neighboring nodes ignoring the connections between them. Even for the attentive GNNs like GAT, the attention coefficient of neighbors still cannot indicate whether they are connected or not. Instead, our method captures rich structural information by sampling paths in entire neighborhood that contain the edges omitted by GNNs, and also our aggregation method maintains to be permutation invariant because both directions of the edges between direct neighboring nodes can be sampled. Furthermore, as shown in Fig. 3, the distance sensitive mechanism generates the distinguishable embedding for different structure which differs from the conventional aggregation.

## 5 Experiments

In this section, we aim to practically validate our proposed model by evaluating its performance on semi-supervised node classification task. To be specific, we want to answer the following questions:

**Q1** Does PathNet achieve satisfactory performance on both homophily and heterophily graphs?

**Q2** How does each component contribute to the performance of our model?

**Q3** Is the maximal entropy path sampler able to capture enough structural information to achieve a satisfying performance?

**Q4** Can PathNet actually improve the expressive power by capturing the structural context of neighborhoods?

### 5.1 Experiment Setup

To answer **Q1**, we evaluate our model on the semi-supervised node classification with seven real world graph datasets: (1) three widely adopted citation network benchmark: Cora, Pubmed and Citeseer[Kipf and Welling, 2017] with strong homophily; (2) four representative heterophily datasets that span across various domains: Cornell[Rozemberczki *et al.*, 2021], BGP[Luckie *et al.*, 2013], NBA[Dai and Wang, 2021], Electronics[McAuley *et al.*, 2015] collected from web pages, BGP network, NBA players and Amazon product catalog separately. The experiment setting and hyperparameter can be found in Appendix C.

**Baselines.** We compare PathNet with the following states of the art baselines: (1) Traditional methods: multi-layer perceptron (MLP, graph-agnostic), several classical GNN models GraphSAGE [Hamilton *et al.*, 2017], GAT [Veličković *et al.*, 2018] and GIN [Xu *et al.*, 2019] under the assumption of homophily are used for fundamental evaluation. (2) Models for heterophily graphs: MixHop [Abu-El-Haija *et al.*, 2019], H2GCN [Zhu *et al.*, 2020], GPRGNN [Chien *et al.*, 2021], FAGCN [Bo *et al.*, 2021]. (3) Models using path or node position: GeniePath [Liu *et al.*, 2019] adaptively explores the receptive field which is termed as receptive paths. SPAGAN [Yang *et al.*, 2019c] conducts path-based higher-order attention. P-GNN [You *et al.*, 2019] emphasizes the importance of encoding the position of each node.

### 5.2 Comparison with State-of-the-art Models

The results are reported in Tab. 1 (upper half). For H2GCN, we use the best result between the H2GCN-1 and H2GCN-2. "OOM" means out of memory.

Our approach PathNet outperforms all the baselines on all four heterophily graphs, demonstrating the necessity of introducing path aggregation paradigm. Specifically, our model achieves an improvement of 10.65%, 10.98% and 18.16% on average against the traditional methods, the models for heterophily graphs and the models using path or node position respectively. (The calculation of average improvement is to average the result of different methods within the same dataset, and then average their results cross all datasets.) The performance proves the effectiveness of our model to capture the complex structure information on heterophily graphs.

For homophily graphs, our model achieves best performance on Pubmed and Citeseer and presents highly competitive results on Cora (0.05% decline compared to the best). Specifically, our model outperforms the three types of models mentioned in baselines by 5.76%, 2.05% and 5.42% on average. Compared with the performance on heterophily datasets, the improvement of our model is not that notable on homophily graphs, which might result from the similarity of the node features in the nearest neighborhood caused by the high homophily level. In summary, our method achieves the best performance on heterophily graphs and competitive performance on homophily graphs.

### 5.3 Ablation Study

To answer **Q2**, we conduct an ablation study by changing parts of the entire framework. The results are shown in Tab. 1 (lower half). When we replace the MERW with CRW while keeping the number and length of paths consistent (RW-PathNet), the performance drops by 1.28% on average across all datasets. It shows that MERW indeed preserves the complex structure context of neighborhoods. Furthermore, since the structure-aware recurrent mechanism preserves the semantic information contained in the maximal entropy paths, when we use MLP as a substitute for the recurrent cell $\phi$ (PathNet-MLP), the performance is reduced by an average of 2.45%, revealing that the recurrent mechanism which retains the node order information is helpful to capture context in neighborhoods. Moreover, the mean accuracy drops by 1.87% and 1.75% on average when replacing the parameter sharing recurrent cell with conventional GRU [Cho *et al.*, 2014] and LSTM [Hochreiter and Schmidhuber, 1997] (PathNet-LSTM/PathNet-GRU). We can tell from the performance that the LSTM version and the GRU version are slightly different on different datasets though they are consistently lower than the proposed parameter sharing recurrent cell $\phi$, which indicates that parameter sharing mechanism for path distance-aware contributes to perceive the contextual structure of neighbors. In addition, path preference modeling contributes to our model by an average of 1.66% and 2.85% compared with replacing the path attention into mean/sum pooling respectively (PathNet-Mean/PathNet-Sum), indicating the effectiveness of choosing different paths in different neighborhoods. In summary, even some components are replaced, our model that explicitly exploits paths still gains satisfactory results.

### 5.4 Model Variants

To answer **Q3**, we conduct experiments by varying the number and length of maximal entropy paths. Take NBA dataset with path length of 3 as an example (Fig. 4(a)), we change the number of sampled paths for every epochs at increasing

|  |  | Cora | Pubmed | Citeseer | Cornell | NBA | BGP | Electronics |
|---|---|---|---|---|---|---|---|---|
| #Hom. ratio |  | 0.81 | 0.80 | 0.74 | 0.30 | 0.39 | 0.37 | 0.25 |
| **Baselines** | MLP | 67.42± 0.44 | 83.90±0.11 | 70.77±1.22 | 80.80±0.81 | 59.21±6.92 | 63.39±0.34 | 75.03±0.08 |
|  | GIN | 84.97±1.51 | 86.97±0.53 | 72.19±1.74 | 58.10±5.70 | 65.47±6.85 | OOM | OOM |
|  | GAT | **86.72±1.58** | 86.56±0.54 | 74.69±1.71 | 63.50±1.81 | 67.19±1.04 | 62.25±0.90 | 64.64±0.27 |
|  | GraphSage | 86.44±1.41 | 88.12±0.24 | 74.76±1.53 | 72.40±3.97 | 61.70±2.40 | 61.71±0.85 | 74.92±0.19 |
|  | MixHop | 85.41±1.61 | 87.83±0.26 | 75.43±1.89 | 61.89±6.43 | 68.89±5.95 | 64.80±0.83 | 67.84±0.50 |
|  | H2GCN | 86.21±0.98 | 87.86±0.19 | 76.73±1.48 | 82.16±4.41 | 66.67±7.02 | 65.13±1.01 | 73.92±0.52 |
|  | GPRGNN | 86.00±2.46 | 86.56±0.29 | 78.45±0.27 | 50.82±3.28 | 48.25±4.97 | 61.49±0.40 | 75.79±0.16 |
|  | FAGCN | 86.30±1.74 | 88.50±0.27 | 76.20±1.45 | 72.70±4.50 | 63.49±3.89 | 64.48±0.55 | 71.10±2.02 |
|  | P-GNN | 68.05±1.30 | 84.97±0.38 | 64.81±1.29 | 58.65±3.21 | 58.41±7.40 | 54.04±3.81 | 57.25±2.78 |
|  | GeniePath | 85.15±0.65 | 86.50±0.34 | 76.46±1.42 | 59.19±4.43 | 68.73±5.41 | 63.15±2.94 | 73.39±0.35 |
|  | SPAGAN | 86.12±0.54 | 85.10±0.19 | 77.41±0.82 | 55.41±2.18 | 53.65±7.23 | 52.59±0.67 | 53.93±5.08 |
| **Ablation** | PathNet-MLP | 82.68±0.39 | 87.86±0.07 | 75.78±1.50 | 90.54±1.35 | 69.05±6.08 | 64.36±0.54 | 75.81±0.58 |
|  | PathNet-GRU | 84.76±1.52 | 87.89±0.12 | 76.57±1.08 | 90.74±1.81 | 69.52±7.16 | 64.46±0.76 | 76.16±0.52 |
|  | PathNet-LSTM | 84.60±0.59 | 87.89±0.14 | 76.44±2.86 | 91.35±1.62 | 69.37±6.27 | 65.19±0.79 | 76.12±0.39 |
|  | RW-PathNet | 85.08±1.17 | 87.84±0.34 | 78.54±2.13 | 90.27±2.16 | 71.27±5.65 | 64.92±0.61 | 76.31±0.45 |
|  | PathNet-Mean | 83.96±0.67 | 88.18±3.94 | 76.59±1.61 | 91.08±2.43 | 70.16±6.18 | 64.81±0.77 | 76.85±0.55 |
|  | PathNet-Sum | 84.23±0.54 | 86.93±0.27 | 74.80±2.44 | 89.19±2.70 | 67.70±6.44 | 65.39±0.83 | 75.06±0.44 |
|  | PathNet | 86.67±1.83 | **88.92±0.21** | **78.86±1.02** | **92.43±1.62** | **72.79±4.93** | **65.72±0.66** | **77.83±0.42** |

Table 1: Mean accuracy and standard deviation of PathNet on node classification compared with baselines and ablation study.

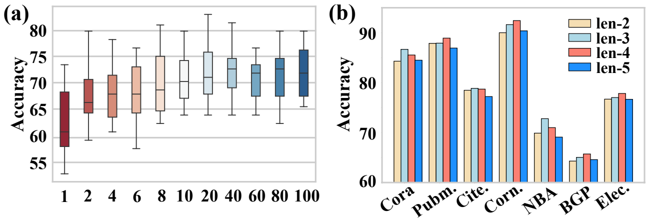


Figure 4: Model variants of different number (a) and length (b) of sampled paths.

intervals for the performance becomes stable when the number is around 40. Before performance convergences, the performance increases with the number of paths increasing, as we analysis in Sec. 4.1. Since we train our model for 1000 epochs, the total number of paths sampled for each node during the entire training process is 1000 times larger. Thus, even when the number is 1 per epoch, the performance is just slightly lower than the average of baselines.

Moreover, when we change the path length between 2 to 5 (Fig. 4(b)), the changes of performance varies with datasets. Generally, the performance of these datasets forms a peek shape. The performance of Cora, Citeseer and NBA peeks at the length of 3 and the other datasets peeks at the length of 4. In summery, the experiments reflect the stability and capability of PathNet on mining the structure information.

### 5.5 Synthetic Experiment

To answer **Q4**, we design a synthetic experiment inspired by [Chen *et al.*, 2021]. [Chen *et al.*, 2020] proves that the capability to count substructures indicates the strength of the expressive power. Thus, we aim to test the expressive power of our model by evaluating the ability of finding out the attributed path (mentioned in Sec. 3.1) with the largest number since attributed paths can be seen as a kind of substructure. The datasets are synthesized as follows: we use Cora and Citeseer as graph structure and change the node attributes into a 2-dimensional one-hot vector The vector is assigned as blue if the original node index is even, and as red if the original node index is odd. Since it has two kinds of node attributes, there are eight different attributed paths of length 3. We number different kinds of attributed paths as indexes and there might be a certain amount of each attributed path for one node. The node labels are defined as the indexes of the largest number of attributed path. We use 10 randomly generated splits with 60%, 20%,and 20% for training, validation and testing. The baselines include GIN which achieves the expressive power of the WL test [Xu *et al.*, 2019], GAT which utilizes the attention mechanism and GPRGNN designed for heterophily graphs.

As shown in Tab. 2, our model significantly outperforms other models on both datasets, which proves that PathNet indeed boosts the expressive power as analysised in Sec. 4.2.

|  | Syn-Cora | Syn-Citeseer |
|---|---|---|
| #Hom. ratio $\mathcal{H}_{\mathcal{G}}$ | 0.37 | 0.39 |
| GIN | 51.40±1.55 | 59.09±2.71 |
| GAT | 36.96±1.60 | 47.44±1.76 |
| GPRGNN | 43.62±1.69 | 53.95±1.87 |
| PathNet | **57.59±1.54** | **71.42±1.15** |

Table 2: Node classification on synthetic datasets.

## 6 Conclusion

We propose a novel path aggregation paradigm and design PathNet to capture the semantic and structural information in both homophily and heterophily graphs. Firstly, we sample a reasonable number of paths. Then, we proposes a structure-aware recurrent cell with the parameter sharing mechanism to aggregate each path. Furthermore, our path attention module can distinguish the preference of paths. As an end-to-end model, we achieve the SOTA performance on both heterophily and homophily graphs.

# A Computational Complexity

Here we discuss the computational complexity of PathNet. In general, our method is rather efficient. Concretely, the preprocessing includes the path sampling and shortest distance calculation, with the precise time complexity of $O(d^k n)$, where $n$ denotes number of nodes, $d$ is the average degree of the graph ($d = 3.89$ in Cora) and $k$ is the length of sampled paths ($k = 4$ in our experiments). Moreover, the overall time complexity of PathNet is $O(d^k n + mtnhh')$, where $m$ denotes the number of sampled path ($m = 40$ in our experiments), $t$ is the iteration (epoch) value, $h$ and $h^{'}$ represent the dimensions of input and output.

# B Proof of Theorem 1

*Proof.* A tag-limited transition matrix $\mathbf{A}^{(l)}$ is derived by setting the same elements as the original adjacency matrix $\mathbf{A}$ if the end node can be tagged as $l$, while other elements are tagged as zero:

$$\mathbf{A}^{(l)}_{i,j} = \begin{cases} \mathbf{A}_{i,j}, & \mathbf{Y}_j = l, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

During the path sampling process, we sample the neighbors with the probability matrix of maximal entropy calculated as the Eq. 2. Thus, the probability matrix $\mathbf{P}^{\{l_1,l_2,\ldots,l_k\}}$ of sampling a attributed path is as follows:

$$\mathbf{P}^{\{l_1,l_2,\ldots,l_k\}} = \prod_{i=1}^{k} (\mathbf{P}^{(l_i)}_{\mathbf{u}}) = \prod_{i=1}^{k} (\frac{\mathbf{D}_{\mathbf{u}}^{-1}\mathbf{A}^{(l_i)}\mathbf{D}_{\mathbf{u}}}{\lambda}). \tag{8}$$

The element in the $i$-th row and $j$-th column of $\mathbf{P}^{\{l_1,l_2,\ldots,l_k\}}$ is the probability of walking from $i$ to $j$ through an attributed path. Moreover, $P_{S,\{l_1,l_2,\ldots,l_k\}}$, the probability of sampling an attributed path starting from node $S$, is derived as follows:

$$P_{S,\{l_1,l_2,\ldots,l_k\}} = \sum_{i=1}^{|\mathcal{V}|} \mathbf{P}^{\{l_1,l_2,\ldots,l_k\}}_{S,i}. \tag{9}$$

Here, $N_{S,\{l_1,l_2,\ldots,l_k\}}$ denotes the number of attributed paths in $m$ times of sampling from graph. Since we use the same sampling strategy for every epoch, the probability of attributed paths being sampled is $P_{S,\{l_1,l_2,\ldots,l_k\}}$, while the probability of not being sampled is $1 - P_{S,\{l_1,l_2,\ldots,l_k\}}$. Whether an attributed path would be sampled in two samples is independent and identically distributed. Therefore, $N_{S,\{l_1,l_2,\ldots,l_k\}}$ follows a binomial distribution:

$$N_{S,\{l_1,l_2,\ldots,l_k\}} \sim \mathrm{B}(m, P_{S,\{l_1,l_2,\ldots,l_k\}}). \tag{10}$$

According to Hoeffding's inequality[Hoeffding, 1963], we have

$$\Pr(|\frac{N_{S,\{l_1,l_2,\ldots,l_k\}}}{M} - P_{S,\{l_1,l_2,\ldots,l_k\}}| > \epsilon) \leq 2\exp(-2\epsilon^2 m). \tag{11}$$

Thus, as the number of sampled paths $m$ increases, the probability on the left of inequality decreases exponentially. □

# C Experiment Settings and Hyperparameters

As important parts of preprocess, the path sampling and shortest distance between nodes are finished before model training with the time complexity shown in supplemental material. The final hyper-parameter settings for PathNet are as follows: 1000 epochs, learning rate as 0.005 and weight decay as 0.0005. For other baselines, we choose the hyperparameter settings which can make them perform best. For simplicity, we set the number of paths as 40 for both heterophily and homophily graphs. Moreover, we set the length of paths as 3 for Cora, Citeseer and NBA while 4 for other datasets. We use 10 random splits with 48%, 32%, and 20% for training, validation and testing. For Cora, Pubmed, Citeseer and Cornell, we use the the same splits provided by [Zhu *et al.*, 2020]. For NBA, BGP, and Electronics, we randomly generate the 10 splits.

# Acknowledgments

This work was partially supported by Zhejiang NSF (LR22F020005), the National Key Research and Development Project of China (2018AAA0101900), and the Fundamental Research Funds for the Central Universities.

# References

[Abu-El-Haija *et al.*, 2019] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *ICML*, pages 21–29, 2019.

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

[Borgwardt *et al.*, 2005] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):47–56, 2005.

[Burda *et al.*, 2009] Zdzislaw Burda, Jarek Duda, Jean-Marc Luck, and Bartek Waclaw. Localization of the maximal entropy random walk. *Physical review letters*, 102(16):160602, 2009.

[Chen *et al.*, 2020] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? In *NIPS*, volume 33, pages 10383–10395, 2020.

[Chen *et al.*, 2021] Lei Chen, Zhengdao Chen, and Joan Bruna. On graph neural networks versus graph-augmented mlps. In *ICLR*, 2021.

[Chien *et al.*, 2021] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *ICLR*, 2021.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014.

[Cover, 1999] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[Dai and Wang, 2021] Enyan Dai and Suhang Wang. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *WSDM*, pages 680–688, 2021.

[Hamilton *et al.*, 2017] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, volume 30, pages 1025–1035, 2017.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hoeffding, 1963] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Liu *et al.*, 2019] Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, Le Song, and Yuan Qi. Geniepath: Graph neural networks with adaptive receptive paths. In *AAAI*, pages 4424–4431, 2019.

[Luckie *et al.*, 2013] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, et al. As relationships, customer cones, and validation. In *IMC*, pages 243–256, 2013.

[McAuley *et al.*, 2015] Julian McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *SIGKDD*, pages 785–794, 2015.

[McPherson *et al.*, 2001] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[Newman, 2003] Mark EJ Newman. Mixing patterns in networks. *Physical review E*, 67(2):026126, 2003.

[Parry, 1964] William Parry. Intrinsic markov chains. *Transactions of the American Mathematical Society*, 112(1):55–66, 1964.

[Rozemberczki *et al.*, 2021] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-Scale Attributed Node Embedding. *Journal of Complex Networks*, 9(2), 2021.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Wu *et al.*, 2017] Boya Wu, Jia Jia, Yang Yang, Peijun Zhao, Jie Tang, and Qi Tian. Inferring emotional tags from social images with user demographics. *IEEE Transactions on Multimedia*, 19(7):1670–1684, 2017.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[Xiong *et al.*, 2019] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of medicinal chemistry*, 63(16):8749–8760, 2019.

[Xu *et al.*, 2019] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[Yang *et al.*, 2018] Yang Yang, Zongtao Liu, Chenhao Tan, Fei Wu, Yueting Zhuang, and Yafeng Li. To stay or to leave: Churn prediction for urban migrants in the initial period. In *WWW*, pages 967–976, 2018.

[Yang *et al.*, 2019a] Yang Yang, Yuhong Xu, Yizhou Sun, Yuxiao Dong, Fei Wu, and Yueting Zhuang. Mining fraudsters and fraudulent strategies in large-scale mobile social networks. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):169–179, 2019.

[Yang *et al.*, 2019b] Yang Yang, Yuhong Xu, Chunping Wang, Yizhou Sun, Fei Wu, Yueting Zhuang, and Ming Gu. Understanding default behavior in online lending. In *CIKM*, pages 2043–2052, 2019.

[Yang *et al.*, 2019c] Yiding Yang, Xinchao Wang, Mingli Song, Junsong Yuan, and Dacheng Tao. Spagan: shortest path graph attention network. In *IJCAI*, pages 4099–4105, 2019.

[You *et al.*, 2019] Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *ICML*, pages 7134–7143. PMLR, 2019.

[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. In *NIPS*, volume 33, pages 7793–7804, 2020.