HP-GMN: Graph Memory Networks for Heterophilous Graphs

Junjie Xu, Enyan Dai, Xiang Zhang, Suhang Wang
The Pennsylvania State University, USA
{junjiexu, emd5759, xzz89, szw494}@psu.edu

2022 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-5099-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDM54844.2022.00165

***Abstract*—Graph neural networks (GNNs) have achieved great success in various graph problems. However, most GNNs are Message Passing Neural Networks (MPNNs) based on the homophily assumption, where nodes with the same label are connected in graphs. Real-world problems bring us heterophily problems, where nodes with different labels are connected in graphs. MPNNs fail to address the heterophily problem because they mix information from different distributions and are not good at capturing global patterns. Therefore, we investigate a novel Graph Memory Networks model on Heterophilous Graphs (HP-GMN) to the heterophily problem in this paper. In HP-GMN, local information and global patterns are learned by local statistics and the memory to facilitate the prediction. We further propose regularization terms to help the memory learn global information. We conduct extensive experiments to show that our method achieves state-of-the-art performance on both homophilous and heterophilous graphs. The code of this paper can be found at https://github.com/junjie-xu/HP-GMN.**

***Index Terms*—Data Mining, Graph Neural Networks, Memory Networks, Heterophily, Node Classification**

I. Introduction

Graph Neural Networks have shown great ability in various graph tasks such as node classification [1], [2], link prediction [3], [4], and graph classification [5]. Generally, the success of GNNs relies on the message-passing mechanism, where a node representation will be updated by aggregating the representations of its neighbors. Thus, the learned node representation captures both node attributes and local neighborhood information, which facilitates downstream tasks. The aggregation process of most current GNNs is explicitly or implicitly designed based on the homophily assumption [6], i.e., two nodes of similar features or the same label are more likely to be linked. For example, in GCN, the representations are smoothed over connected neighbors with the assumption that the neighbors of a node are likely to have the same class and similar feature distribution. However, there are many heterophilous graphs in the real world that do not follow the homophily assumption. In heterophilous graphs, a node is also likely to connect to another node with dissimilar features or different labels. For example, fraudsters tend to contact normal users instead of other fraudsters in a trading network [7]; different amino acids are connected to form functional proteins [6]; interdisciplinary researchers collaborate more with people from different areas in a citation network.

Message Passing Neural Networks (MPNNs) with homophily assumption are challenged by heterophilous graphs. This is mainly because: (i) Since the majority of the neighborhood nodes lie in the same class as the center node in homophilous graphs, directly mixing the neighbor representations by averaging operation can preserve the context pattern, benefiting the downstream task. However, neighbors in heterophilous graphs come from different classes and have different distributions. The aggregation process in current MPNNs generally ignore the differences between nodes in different classes and simply mix them together. Therefore, the pattern of the local context in graphs with heterophily would not be well-preserved with the current traditional MPNNs; and (ii) only local context information of nodes is utilized in the prediction of MPNNs. The MPNNs generally fail to explicitly capture and utilize the global patterns of the nodes' local heterophilous context to give more accurate results.

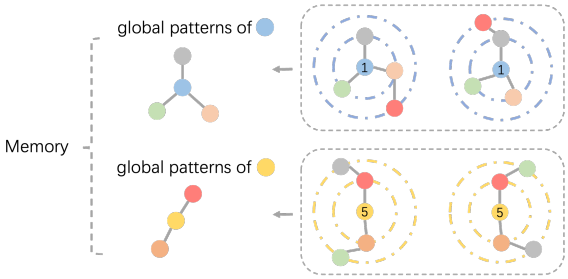


Fig. 1: Learning local statistics and global patterns on heterophilous graph. Color is the node label, while the number represents the attribute.

To overcome the shortcomings of MPNNs on heterophilous graphs, we propose the following two strategies. First, in heterophilous graphs, though the neighbors of a node can have dissimilar node attributes and labels with the center node, we observe that nodes of the same class tend to have similar neighborhood distributions while nodes of different classes tend to have dissimilar neighborhoods. For example, as shown in the toy example in Fig. 1, nodes of "blue" class have similar node attributes and subgraph structures, while nodes in the "blue" class and the "yellow" class have dissimilar attributes and subgraph structures. In addition, as Fig. 1 shows, neighbors' labels of blue and yellow nodes are different and blue nodes are connected to more neighbors. This indicates that the neighbors' distributions of "blue" and "yellow" are different. Thus, instead of simply aggregating the heterophilous neighbors' attributes which could result in noisy representations, we can summarize the local statistics from various aspects, such as features, the structure, and distributions of neighbors, to capture more comprehensive

978-1-6654-5099-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00165

Authorized licensed use limited to: Penn State University. Downloaded on December 30,2023 at 03:43:38 UTC from IEEE Xplore. Restrictions apply.

and discriminative information of the local context, which facilitates better representations. Second, each class has some representative and frequent subgraphs. Taking the "blue" class as an example, we can observe that the blue nodes are generally connected with green, grey and orange nodes. The captured global patterns can benefit the prediction on the test instance by providing global information. However, current GNNs generally only rely on the representations of the local subgraph and fail to capture the global patterns of the nodes and their local context to facilitate the classification task. One promising method to address this issue is to learn global information by adopting Memory Networks [8], [9]. Specifically, in memory networks, multiple memory units are used to store the global patterns of instances in different classes. The predictions can be given by matching the local patterns with the learned global patterns, which effectively utilize both the local context information and the global information.

Therefore, this paper studies a novel problem of capturing both local and global information for heterophilous graphs. The main challenges are: **(1)** How to select local statistics that are good at capturing characteristics of subgraphs? **(2)** How to guarantee the memory stores global information favorable for the prediction? The memory units are required to be representative and diverse. Being representative makes sure that memory units record the most frequent patterns of the nodes while being diverse means that memory units are different from each other so that they will not record duplicate information. How should the update process be regulated to guarantee the diversity and representativeness of memory units? To deal with these issues, we propose Graph Memory Networks for Heterophilous Graphs (**HP-GMN**). HP-GMN incorporates local statistics that can effectively capture the information of nodes in attributes, structures, and neighbor distributions on graphs with heterophily, and the memory that can learn global patterns of the graph. To ensure learning global patterns in high-quality, regularization methods are deployed to keep the memory units representative and diverse.

In summary, we study the node classification task on heterophilous graphs, and the main contributions are:

- We develop a novel framework called HP-GMN using local statistics and the memory to learn local and global representations for heterophilous graphs.
- We propose regularization methods to encourage the update process of the memory to capture global information while keeping it diverse and representative.
- We conduct extensive experiments on real-world datasets and reveal our memory network outperforms state-of-the-art GNNs on both homophilous and heterophilous graphs.

II. Preliminaries

A. Notations and Definition

We use $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{X})$ to denote an attributed graph, where $\mathcal{V}=\{v_1,...,v_N\}$ is the set of N nodes, and $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$ is the set of edges. $\mathbf{X}=\{\mathbf{x}_1,...,\mathbf{x}_N\}\in\mathbb{R}^{|\mathcal{V}|\times F}$ is the node feature matrix, where $\mathbf{x}_i$ is the node features of node v_i and F is the feature dimension. $\mathbf{A}\in\mathbb{R}^{N\times N}$ is the adjacency matrix, where $\mathbf{A}_{ij}$ = 1 if nodes pair $\{i,j\}\in\mathcal{E}$; otherwise $\mathbf{A}_{ij}$ = 0.

The graph neural networks aim to learn effective node representations $\mathbf{H}\in\mathbb{R}^{|\mathcal{V}|\times F'}$ used in downstream tasks, where F' is the dimension of representations. Typically, GNNs use message passing [10] to learn representations. Each node aggregates its neighbors' representations and then combines them with the ego-representation. Hence, there are two essential steps in MPNNs:

$$\mathbf{a}_v^{k+1}=\text{AGGREGATE}^{k+1}(\{\mathbf{h}_u^k:u\in\mathcal{N}(v)\}), \quad (1)$$

$$\mathbf{H}_v^{k+1}=\text{COMBINE}^{k+1}(\mathbf{h}_v^k,\mathbf{a}_v^k), \quad (2)$$

where $\mathbf{h}_v^k$ is the representation of node v at layer k and $\mathcal{N}(v)$ is the neighbors of node v. AGGREGATE and COMBINE are two functions specified by GNNs.

Generally, the homophily degree of a graph can be measured by node homophily ratio [11].

Definition1 (Node homophily ratio) [11] *It is the average ratio of same-class neighbor nodes to the total neighbor nodes in a graph.*

$$\mathcal{H}_{node}=\frac{1}{|V|}\sum_{v\in\mathcal{V}}\frac{|\{u\in\mathcal{N}(v):y_v=y_u\}|}{|\mathcal{N}(v)|}\in[0,1]\ , \quad (3)$$

where y is the node label. Graphs with higher homophily are close to 1, and graphs with higher heterophily are close to 0.

B. Intuition and Problem Definition

There are two main reasons leading to the incompatibility between MPNNs and heterophily. Firstly, AGGREGATE is usually implemented by a mean or weighted mean function and COMBINE is to mix a node's ego- and neighbor-representations. Because neighbors are from different classes and have different distributions in the heterophily assumption, such process can mix representations from different distributions and make the result of aggregation less discriminative. This mixture gives us poor node representations, which further degrades the quality of global patterns learned from each node. In this case, even MLP can have better performance than GCN because the structure information is harmful according to some empirical results [6], [12]. Secondly, in MPNNs, a node only gathers information from the local neighbors and fails to get more global information. As a result, MPNNs cannot learn global patterns from nodes' local contexts. Therefore, MPNNs are not good at addressing the heterophily problem.

In a graph, each node has a k-hop subgraph centered on itself. The characteristics of the subgraph can provide much information for the node's local context; therefore, local statistics can be designed to capture this local information for downstream classification tasks. Furthermore, for all the nodes of each class, we can learn the global patterns of the class by extracting representative information of the nodes' local representations. Then each node can learn global information from similar global patterns. We need to measure the similarity here because we do not need information from different distributions. Based on the high-level ideas, we develop HP-GMN to learn from both local and global information. To

Authorized licensed use limited to: Penn State University. Downloaded on December 30,2023 at 03:43:38 UTC from IEEE Xplore. Restrictions apply.

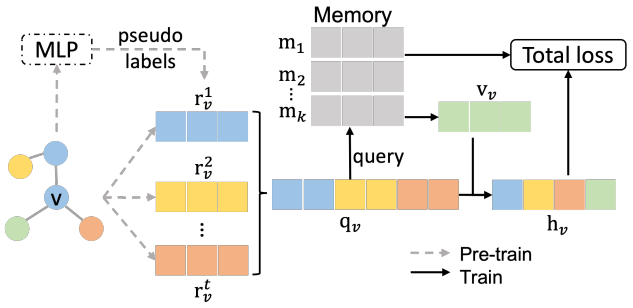


Fig. 2: The framework of HP-GMN. Informative local statistics $\{\mathbf{r}^1, \cdots, \mathbf{r}^t\}$ are summarized and transformed to obtain local pattern vectors $\mathbf{q}$. As for the memory network module, it deploys a memory matrix $\mathbf{M}$ consisting of memory units to store global patterns of the nodes in different classes. $\mathbf{q}$ will be used to query the memory matrix to give the value vector $\mathbf{v}$.

measure the performance of our framework, we mainly have semi-supervised node classification as the downstream task.

Problem1 *Given an attributed graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ *and a training set* $\mathcal{V}_\mathcal{L} \subseteq \mathcal{V}$*, with known class labels* $\mathcal{Y}_L = \{y_v, \forall v \subseteq \mathcal{V}_\mathcal{L}\}$*, the heterophily ratio of the graph* $\mathcal{G}$ *is low, we aim to learn a GNN that can infer the unknown class labels* $\hat{\mathcal{Y}}_U = \{\hat{y}_u, \forall u \subseteq \mathcal{V}_\mathcal{U} = (\mathcal{V} - \mathcal{V}_\mathcal{L})\}$.

III. Methodology

In this section, we describe our proposed HP-GMN thoroughly. The overall idea is to utilize local statistics to capture useful local information and simultaneously deploy the memory network to extract global patterns. Then, accurate predictions can be given by combining both their local context information and the captured global patterns. There are several challenges in the process: (1) How to design local statistics that can effectively describe the local context of nodes in heterophilous graphs? (2) How to obtain representative and diverse memory units to capture informative global patterns? In an attempt to address these challenges, node attributes and diffusion matrix are incorporated into the local statistics to preserve the local pattern in attribute and structure information. In addition, label-wise statistics [13] are also used to describe neighbors' distributions of center nodes. To ensure the quality of the memory units, Kpattern regularization and Entropy regularization are adopted to guarantee diversity and representativeness. The overall framework of the proposed HP-GNN is shown in Fig. 2.

A. Local Representation with Local Statistics

It has been observed that the local context of nodes in heterophilous graphs can differ a lot for nodes are in different classes, while two nodes in the same class often exhibit similar node features and local subgraphs [13]. Thus, we propose the local statistics including node attributes, neighborhood distributions, and local topology.

1) Node Attributes: As node attributes contains important information about node labels and similar nodes tend to have similar attributes, the first local statistic is given by $\mathbf{r}_v^1 = \mathbf{x}_v$, where $\mathbf{x}_v$ denotes the attributes of node v.

2) Label-wise Statistics: To effectively preserve the neighborhood distribution, we adopt the label-wise aggregation, which has been proved to be an effective method for heterophilous graphs [13] because it aggregates neighbors from different classes separately and avoids obscuring the boundaries between classes. In this paper, we use two label-wise characteristics as local statistics, i.e., label-wise neighbor class distribution and label-wise neighbor feature distribution.

First, as suggested by [13], we estimate pseudo labels of the unlabelled nodes, which are used to guide the label-wise aggregation. Specifically, a label estimator f_E is utilized to obtain the pseudo label of node v: $\hat{y}_v^{pseudo} = f_E(x_v)$. Many classifiers can be used as the estimator f_E, such as GCN or MLP. We adopt MLP in our framework for simplicity. f_E is trained with labelled nodes.

We assume similar nodes to have similar local neighbor class distributions. Thus, for each node v, label-wise neighbor class distribution counts the numbers of neighbors of each class using the pseudo labels:

$$\mathbf{r}_v^2 = \left[\ |\mathcal{N}_1(v)|\ ,\ |\mathcal{N}_2(v)|, \cdots, |\mathcal{N}_{|\mathcal{Y}|}(v)|\ \right], \tag{4}$$

where $\mathcal{N}_i(v) = \{u : (v,u) \in \mathcal{E}\ and\ \hat{y}_u = i\}$ denote the set of neighbors of v with pseudo label i and $|\cdot|$ is the size of a set. Hence, $|\mathcal{N}_i(v)|$ gives the number of neighbors of node v with label i.

Similarly, we expect similar nodes in heterophilous graphs to have similar local neighborhood feature distributions. Thus, for each node, label-wise neighbor feature distribution calculates the average features of neighbors from each label as:

$$\mathbf{r}_v^3 = \left[\ \sum_{u \in \mathcal{N}_1(v)} \frac{\mathbf{x}_u}{|\mathcal{N}_1(v)|}\ \middle\|\ \cdots\ \middle\|\ \sum_{u \in \mathcal{N}_{|\mathcal{Y}|}(v)} \frac{\mathbf{x}_u}{|\mathcal{N}_{|\mathcal{Y}|}(v)|}\ \right], \tag{5}$$

where $\|$ denotes the concatenation operation.

3) Diffusion Matrix: Structure information is essential and similar nodes often have similar structures in their subgraphs. We exploit higher-order structural information by utilizing the diffusion matrix. The diffusion matrix can remove the restriction of using only the direct 1-hop neighbors through different powers of the adjacency matrix [14]. It is as:

$$\mathbf{S} = \sum_{k=0}^{\infty} \theta_k \mathbf{T}^k, \tag{6}$$

where $\mathbf{s}_v$ is the v-th row of matrix $\mathbf{S}$ and $\mathbf{T}$ is the transition matrix. We require that $\sum_{k=0}^{\infty} \theta_k = 1, \theta_k \in [0,1]$, and the eigenvalues of $\mathbf{T} \in [0,1]$ to guarantee the convergence. The transition matrix $\mathbf{T}$ and weighting coefficients θ_k are defined in various ways. In this paper, we use the PPR diffusion matrix [15] due to its good performance and flexibility:

$$\mathbf{T} = \mathbf{A}\mathbf{D}^{-1},\ \theta_k = \alpha(1-\alpha)^k, \tag{7}$$

where teleport probability $\alpha \in (0,1)$.

4) Local Representation: With the above four types of features capturing different perspectives of the local statistics, we transform them by MLPs since some local statistics might be high dimensional, and transformation to the latent space can reduce the dimension and better reflect their patterns. Finally, we concatenate them to form the local representation

Authorized licensed use limited to: Penn State University. Downloaded on December 30,2023 at 03:43:38 UTC from IEEE Xplore. Restrictions apply.

$\mathbf{q}_v$ of node v as:

$$\mathbf{q}_v = [\mathrm{MLP}_1(\mathbf{r}_v^1)||\mathrm{MLP}_2(\mathbf{r}_v^2)||\mathrm{MLP}_3(\mathbf{r}_v^3)||\mathrm{MLP}_4(\mathbf{r}_v^4)]. \quad (8)$$

In summary, the advantages of $\mathbf{q}_v$ are: (i) it avoids aggregating neighbors of dissimilar features and captures node attributes and heterophilous local graph information. (ii) it can be used to query the memory to get better global information.

B. Global Representation Learning with Memory Unit

The global patterns are some representative local graph patterns that appear most frequently for nodes of each class. We want to learn them so that nodes can aggregate from similar patterns instead of dissimilar neighbors. However, directly learning representative subgraphs is difficult as we need to consider both graph structure and node attributes. Thus, we learn representative global vector representations using a memory module instead. Specifically, the memory is a matrix $\mathbf{M} \in \mathbb{R}^{K \times hidden} = [\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_K]^{\mathrm{T}}$, where K is the number of memory units and $hidden$ is the feature dimension of a memory unit. Each row $\mathbf{m_i}$ is a memory unit used to learn a global pattern.

To get the global representations, we use the attention scores to measure the similarity between each node's local representation and every global pattern. Then a node gathers information from each pattern according to the attention scores. Intuitively, the node get global information by the combinations of the patterns that can describe the distribution it belongs to. We follow [16] to query the memory and get attention matrix:

$$\mathbf{S} = \mathrm{Softmax}(\mathbf{M}\mathbf{Q}^{\mathrm{T}}), \quad (9)$$

where $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_n]^{\mathrm{T}}$ is the query matrix. s_{ij} indicates the importance of the i-th memory to j-th node. Then we calculate the value matrix $\mathbf{V}$, which is global representation by weighted average of the memories as:

$$\mathbf{V} = \mathbf{S}^{\mathrm{T}}\mathbf{M}. \quad (10)$$

Eventually, we concatenate the local and global representation of a node followed by an MLP transformation to get the final representation $\mathbf{h}_v$. Then we can predict the label distribution of $\hat{\mathbf{y}}_v$.

$$\mathbf{h}_v = \mathrm{MLP}_5([\mathbf{q}_v||\mathbf{v}_v]), \quad (11)$$

$$\hat{\mathbf{y}}_v = \mathrm{Softmax}(\mathbf{h}_v), \quad (12)$$

where $\hat{\mathbf{y}}_i$ is the predicted label probabilities of node v_i.

For the training, we minimize the cross-entropy loss as

$$\mathcal{L}_{class} = \frac{1}{|\mathcal{V}_{train}|} \sum_{v \in \mathcal{V}_{train}} l(\hat{\mathbf{y}}_v, \mathbf{y}_v), \quad (13)$$

where $\mathcal{V}_{train}$ is the set of labeled nodes, $\mathbf{y}_v$ is the one-hot-encoding of v's label and $l(\cdot,\cdot)$ is the cross entropy loss.

C. Memory Regularization

In order to capture global information effectively, we desire two properties of the memory units, i.e., representativeness and diversity. Representativeness is desired because the memory is expected to capture the most representative patterns of the distributions of a class. Diversity is also required to encourage richer information and avoid much-duplicated information recorded in memory units. However, during the training process, memory units are updated by gradient descent. The learned memory units are likely to lack the properties we desire. Therefore, we propose Kpattern and Entropy regularizations to encourage the properties.

TABLE I: Statistics of heterophilous and homophilous datasets.

Datasets	#Nodes	#Edges	#Attributes	#Classes	$\mathcal{H}_{node}$
Texas	183	309	1,703	5	0.06
Wisconsin	251	499	1,703	5	0.16
Cornell	183	295	1,703	5	0.11
Chameleon	2,277	36,101	2,325	5	0.25
Squirrel	5,201	217,073	2,089	5	0.22
Crocodile	11,631	360,040	128	5	0.30
Cora	2708	5429	1433	7	0.83
Citeseer	3327	4732	3703	6	0.71
Pubmed	19717	44338	500	3	0.79

Kpattern Regularization. To make the memory representative, we want each query $\mathbf{q}_v$ to be at least close to one of the memory units so that each query can retrieve important global information. In other words, the memory should be useful for each node. Thus, we calculate the distances between each node's local representation and every memory unit. Then we choose the memory unit with the smallest distance. We sum up all the smallest distances over all nodes, and we aim to learn memory that can minimize the total loss as:

$$\min_{\mathbf{M}} \mathcal{L}_{kpattern} = \min_{\mathbf{M}} \sum_{v \in \mathcal{V}} \min_{\mathbf{m}_i \in \mathbf{M}} dist(\mathbf{m}_i, \mathbf{q}_v), \quad (14)$$

where $dist$ can be any function measuring distance between two vectors. We adopt the Euclidean distance in this paper. Intuitively, we prefer representative $\mathbf{M}$ that makes the total distance between the memory and the query minimized, which means the K patterns become representative for the query.

Entropy Regularization. To make the memory diverse, we want each memory to have similar chances of being selected globally. In other words, the averaged importance score over all the nodes for each memory should be similar to each other. Since s_{ij} indicates the importance of i-th memory to j-th node, the overall importance score of $\mathbf{m}_i$ can be written as $\mathbf{s}'_i = \sum_{j=1}^{N} s_{ij}$. Denote the memory importance vector as $\mathbf{s}' \in \mathbb{R}^K$ with $\mathbf{s}'_i$ as the i-th element of $\mathbf{s}'$ indicating the total importance of i-th memory unit.

We need the elements in $\mathbf{s}'$ to be equal or similar to others, i.e., $\mathbf{s}'$ is uniformly distributed, which means all the memory units are equally important and are used with the same frequency. Otherwise, if some units have low overall importance scores than others, there are two possible reasons. (1) Some memory units are unimportant because they contain duplicated information from others. As a result, the information contained in these units can also be expressed by others. (2) Some memory units are useless, i.e., not representative. They do not contain any useful information demanded by downstream tasks, so they are rarely used. Therefore, we can achieve uniformly distributed $\mathbf{s}'$ by maximizing the entropy of it, namely, i.e., minimizing the negative entropy as:

$$\min_{\mathbf{M}} \mathcal{L}_{entropy} = -\mathrm{H}(\mathbf{s}') \quad (15)$$

Authorized licensed use limited to: Penn State University. Downloaded on December 30,2023 at 03:43:38 UTC from IEEE Xplore. Restrictions apply.

TABLE II: Node classification performance on heterophily. (Accuracy(%) ± Std.)

Methods	Texas	Wisc.	Cornell	Cham.	Squi.	Croco.
MLP	80.8±7.0	85.1±5.6	83.2±6.4	48.3±2.2	33.4±1.1	65.7±1.0
GCN	59.5±7.1	51.4±3.1	56.8±4.5	67.5±2.6	56.6±1.4	71.9±0.9
MixHop	78.7±6.3	83.7±4.6	79.5±6.3	49.3±1.2	49.3±1.2	73.9±1.2
GCN+JK	61.6±7.2	58.4±4.1	59.2±7.0	52.6±1.3	52.6±1.3	71.9±0.8
APPNP	80.5±3.8	84.7±3.9	83.0±7.2	40.4±1.6	40.4±1.6	66.3±1.1
GPRGNN	81.4±5.2	82.8±3.8	77.6±7.1	69.3±1.4	49.6±1.7	68.2±0.8
BMGCN	80.0±5.4	75.3±6.1	74.6±5.0	69.0±1.6	52.7±1.1	64.3±1.1
MMP	77.6±6.0	84.1±4.0	79.5±8.5	66.1±2.2	53.1±10.9	66.2±0.4
HPGMN	**85.1±4.2**	**86.5±3.1**	**84.1±5.3**	**79.6±0.8**	**72.3±2.3**	**80.8±0.3**

D. Final Objective Function of HP-GMN

With $\mathbf{h}_i$ in Eq.(11) capturing both local and global information, $\mathcal{L}_{kpattern}$ in Eq.(14) to make the memory representative and $\mathcal{L}_{entropy}$ in Eq.(15) to make the memory diverse, the final objective function of HP-GMN is:

$$\min_{\theta,\mathbf{M}} \mathcal{L}_{total} = \mathcal{L}_{class} + \alpha \mathcal{L}_{kpattern} + \beta \mathcal{L}_{entropy}, \qquad (16)$$

where α and β are scalars to control the contributions of $\mathcal{L}_{kpattern}$ and $\mathcal{L}_{entropy}$. θ is the set of parameters of MLPs.

IV. Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of our proposed framework. In particular, we aim to answer the following research questions: **RQ1** How effective is the Memory Network in the node classification on heterophilous graphs? **RQ2** Can HP-GMN be extended to homophilous graphs? **RQ3** What are the contributions of each component to the final results?

A. Experimental Settings

1) Baselines: We compare our method with representative GNNs (MLP, GCN), GNNs incorporating high-order neighborhood information (MixHop [17], GCN+Jump Knowledge [18], APPNP [19]), as well as GNNs for heterophily (BMGCN [20], GPRGNN [21], MMP [22]).

2) Datasets: We utilize six publicly available heterophilous datasets, including three WebKB datasets (Cornell, Texas, Wisconsin), and three Wikipedia network datasets (Chameleon, Squirrel, Crocodile) [11]. We use three homophilous datasets from citation networks [23] (Cora, Citeseer, PubMed). The key statistics of the datasets are summarized in Table I.

3) Implementation Details: The performance on heterophilous datasets is evaluated on ten train/validation/test splits. We use five repeated experiments with different random seeds to evaluate the performance on homophilous datasets. We use the fixed splits provided by PyG [24] if available. Otherwise, we conduct experiments on ten random splits. For HP-GMN, all MLPs are implemented with two layers. We vary the number of memory unit K among {20, 50, 100, 200, 300, 500} and the hidden number of each memory unit $hidden$ among {100, 200, 300, 500}. We add the Frobenius norm of the memory $||\mathbf{M}||_{\mathrm{F}}^2$ into the loss to prevent overfitting. Coefficients α and β are set by grid search in {0.0001, 0.001, 0.01, 0.1, 1, 10, 100}.

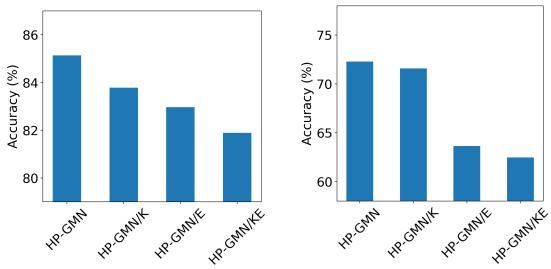
TABLE III: Node classification performance on homophily. (Accuracy(%) ± Std.)

Methods	Cora	Citeseer	Pubmed
MLP	53.5±1.5	51.9±1.5	69.9±0.6
GCN	78.5±1.4	67.8±1.0	76.6±1.1
MixHop	77.0±0.7	63.9±0.4	74.3±0.5
GCN+JK	76.7±2.4	64.8±1.4	75.0±0.8
APPNP	83.1±0.5	70.8±0.9	79.9±0.3
HP-GMN$_{\mathrm{GCN}}$	80.3±1.1	69.0±0.9	77.8±0.7
HP-GMN$_{\mathrm{APPNP}}$	**84.7±0.6**	**73.0±1.0**	**82.4±1.3**

B. Node Classification on Graphs with Hetreophily

To answer **RQ1**, we conduct node classification on heterophilous graphs and compare HP-GMN with the baselines. The average accuracy with standard deviation are shown as Table II and we have the following observations:

- GCN is a little better than MLP or even no better than MLP on heterophilous datasets because GCN fails to exploit the structure information of heterophilous graphs. Our HP-GMN considers heterophily properties, and the local statistics can capture heterophilous structures well; thus, it achieves much better performance than MLP and GCN.
- While methods employing higher-order neighborhood information improve the performance, the extent is limited because they aggregate all the higher-order neighbors regardless the homophily or heterophily. HP-GMN only aggregates from similar global patterns, which leads to further improvements in accuracy.
- Methods designed for heterophily have better performance than general graph methods even without global patterns. However, HP-GMN takes advantage of global patterns and outperforms all other baselines significantly.



(a) Texas (b) Squirrel

Fig. 3: Impact of regularization terms on accuracy.

TABLE IV: Impacts of local statistics.

Methods	Texas	Squirrel
All Local Statistics	**85.1±4.2**	**72.3±2.3**
W/O Node Attributes	70.3±5.5	71.7±1.9
W/O Label-wise Class	83.2±6.0	57.8±2.5
W/O Label-wise Feature	79.2±4.0	41.7±3.4
W/O Diffusion Matrix	83.2±5.9	68.4±6.6

C. Node Classification on Graphs with Homophily

To answer **RQ2**, we aim to demonstrate that HP-GMN can also work on homophilous datasets. However, the local statistics in III-A are designed for heterophily and do not fit the homophily problem. In order to show the effectiveness on the homophilous datasets, we use the representation learned by GCN or APPNP as the local statistic on graphs with homophily,

Authorized licensed use limited to: Penn State University. Downloaded on December 30,2023 at 03:43:38 UTC from IEEE Xplore. Restrictions apply.

denoted as HP-GMN$_{\text{GCN}}$ and HP-GMN$_{\text{APPNP}}$. We conduct experiments on three homophilous citation networks. The average accuracy with the standard deviation is shown in Table III. From Table III, we observe that the memory not only promotes the prediction on heterophily but also can facilitate learning on homophily. That is because homophilous graphs also contain some global information that is not captured by other methods.

D. Ablation Study

To answer **RQ3**, we compare the performance of HP-GMN, HP-GMN without Kpattern regularization (HP-GMN/K), HP-GMN without Entropy regularization (HP-GMN/E), and HP-GMN without regularizations (HP-GMN/KE). We report the performance on Texas and Squirrel in Fig. 3. Then we show the contribution of local statistics. We remove each local statistic to get different designs of the query. We observe the best performance when we use all four local statistics. It reveals that all of them can collect different aspects of local information in the graph, and everyone is essential for describing the local subgraph. Results are shown in Table IV, where "W/O Node Attributes" denotes that "Node Attributes" are removed from local statistics.

V. Related Work

Some methods for the heterophily modify the current GNN framework. H_2GCN [6] uses ego and neighbor separation to encode the ego and neighbor representations separately instead of mingling them together. CPGNN [25] uses the compatibility matrix to initialize and guide the propagation of the GNN. GPRGNN [21] combines intermediate representations and learn adaptive weights for them using Generalized PageRank. Some methods try to exploit global information by using higher-order neighborhoods. MixHop [17] learns higher-order information by utilizing multiple powers of the adjacency matrix. GCN-Cheby [26] uses k-order Chebyshev Polynomials to replace the first order Chebyshev Polynomials in GCN so that it can learn from up to k-order neighbors. Recently, [27] also studied the problem of conducting self-supervised learning for node representation on heterophilous graphs when task-specific labels are scarce.

VI. Conclusion

In this paper, we develop a novel graph memory network to address the heterophily problem. Local statistics and memory are utilized to capture local and global information in a graph. Kpattern and Entropy regularizations are proposed to encourage the memory to learn enough global information. Extensive experiments show that our method can achieve state-of-the-art performance on heterophilous and homophilous graphs.

Acknowledgment

This project was partially supported by NSF projects IIS-1707548, CBET-1638320, IIS-1909702 and IIS-1955851, the Army Research Office under grant W911NF21-1-0198, and DHS CINA under grant E205949D.

References

[1] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[2] E. Dai, T. Zhao, H. Zhu, J. Xu, Z. Guo, H. Liu, J. Tang, and S. Wang, "A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability," *arXiv preprint arXiv:2204.08570*, 2022.

[3] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," *NeurIPS*, vol. 31, 2018.

[4] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[5] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *AAAI*, 2018.

[6] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," *NeurIPS*, vol. 33, 2020.

[7] S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos, "Netprobe: a fast and scalable system for fraud detection in online auction networks," in *WWW*, 2007, pp. 201–210.

[8] J. Weston, S. Chopra, and A. Bordes, "Memory networks," *CoRR*, vol. abs/1410.3916, 2015.

[9] X. Tang, H. Yao, Y. Sun, C. Aggarwal, P. Mitra, and S. Wang, "Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values," in *AAAI*, vol. 34, no. 04, 2020, pp. 5956–5963.

[10] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*. PMLR, 2017, pp. 1263–1272.

[11] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-gcn: Geometric graph convolutional networks," in *ICLR*, 2020.

[12] Y. Ma, X. Liu, N. Shah, and J. Tang, "Is homophily a necessity for graph neural networks?" in *ICLR*, 2022.

[13] E. Dai, S. Zhou, Z. Guo, and S. Wang, "Label-wise message passing graph neural network on heterophilic graphs," 2021.

[14] J. Klicpera, S. Weißenberger, and S. Günnemann, "Diffusion improves graph learning," *NeurIPS*, vol. 32, 2019.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[16] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," *NeurIPS*, vol. 28, 2015.

[17] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, and A. Galstyan, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *ICML*. PMLR, 2019, pp. 21–29.

[18] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *ICML*. PMLR, 2018, pp. 5453–5462.

[19] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," *arXiv preprint arXiv:1810.05997*, 2018.

[20] D. He, C. Liang, H. Liu, M. Wen, P. Jiao, and Z. Feng, "Block modeling-guided graph convolutional neural networks," *arXiv preprint arXiv:2112.13507*, 2021.

[21] E. Chien, J. Peng, P. Li, and O. Milenkovic, "Adaptive universal generalized pagerank graph neural network," in *ICLR*, 2021.

[22] J. Chen, W. Liu, and J. Pu, "Memory-based message passing: Decoupling the message for propagation from discrimination," in *ICASSP*. IEEE, 2022, pp. 4033–4037.

[23] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[24] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[25] J. Zhu, R. A. Rossi, A. Rao, T. Mai, N. Lipka, N. K. Ahmed, and D. Koutra, "Graph neural networks with heterophily," in *AAAI*, vol. 35, no. 12, 2021, pp. 11 168–11 176.

[26] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *NeurIPS*, vol. 29, 2016.

[27] T. Xiao, Z. Chen, Z. Guo, Z. Zhuang, and S. Wang, "Decoupled self-supervised learning for non-homophilous graphs," in *NeurIPS*, 2022.

Authorized licensed use limited to: Penn State University. Downloaded on December 30,2023 at 03:43:38 UTC from IEEE Xplore. Restrictions apply.