EXTRACT and REFINE: Finding a Support Subgraph Set for Graph Representation

Kuo Yang
University of Science and Technology of China
Hefei, China
yangkuo@mail.ustc.edu.cn

Zhengyang Zhou*
Suzhou Institute for Advanced Research, University of Science and Technology of China
Suzhou, China
zzy0929@mail.ustc.edu.cn

Wei Sun
University of Science and Technology of China
Hefei, China
sunwei3@mail.ustc.edu.cn

Pengkun Wang
Suzhou Institute for Advanced Research, University of Science and Technology of China
Suzhou, China
pengkun@mail.ustc.edu.cn

Xu Wang
University of Science and Technology of China
Hefei, China
wx309@mail.ustc.edu.cn

Yang Wang*
University of Science and Technology of China
Hefei, China
angyan@ustc.edu.cn

ABSTRACT

Subgraph learning has received considerable attention in its capacity of interpreting important structural information for predictions. Existing subgraph learning usually exploits statistics on predefined structures e.g., node degrees, occurrence frequency, to extract subgraphs, or refine the contents via only capturing label-relevant information with node-level sampling. Given diverse subgraph patterns, and mutual independence with local correlations on graphs, current solutions on subgraph learning still have two limitations in extraction and refinement stages. 1) The universality of extracting substructure patterns across domains is still lacking, 2) node-level sampling in refinement will distort the original local topology and none explicit guidance eliminating redundant information contribute to inefficiency issue. In this paper, we propose a unified subgraph learning scheme, Poly-Pivot Graph Neural Network (P2GNN) where we designate the centric node of each subgraph as the pivot. In the extraction stage, we present a general subgraph extraction principle, i.e., *Local Asymmetry* between the centric and affiliated nodes. To this end, we asymmetrically model the similarity between each pair of nodes with random walk and quantify mutual affiliations in Affinity Propagation architecture, to extract subgraph structures. In the refinement, we devise a subgraph-level exclusion regularization to squash the target-independent information by considering mutual relations across subgraphs, cooperatively preserving a support set of subgraphs and facilitating the refinement process for graph representation. Empirical experiments on diverse web and biological graphs reveal 1.1%~7.3% improvements against best baselines, and visualized case studies prove the universality and interpretability of our P2GNN.

CCS CONCEPTS

• **Mathematics of computing → Graph algorithms**; • **Computing methodologies → Learning latent representations**.

KEYWORDS

Graph Neural Network; Subgraph Extraction; Local Asymmetry; Subgraph Refinement

ACM Reference Format:
Kuo Yang, Zhengyang Zhou, Wei Sun, Pengkun Wang, Xu Wang, and Yang Wang. 2023. EXTRACT and REFINE: Finding a Support Subgraph Set for Graph Representation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599339

1 INTRODUCTION

Graph-structured data is ubiquitous across various real-world scenarios, ranging from social networks [30, 46], citation networks [12, 50], to molecular graphs [17, 42]. Recently, Graph Neural Networks (GNNs) have achieved great success to materialize diverse downstream tasks through a sparse message-passing process. However, there is a growing recognition that the message-passing paradigm has inherent limitations [22], i.e., the expressiveness of message passing in traditional GNNs is upper bounded by the first order Weisfeiler-Leman (1-WL) isomorphism test [39]. The limitation of GNNs motivates researchers to explore more expressive architecture. Most notably, subgraph-level explanations are more intuitive and useful, as subgraphs are simple building blocks of complex graphs and concerned with the functionalities of graphs. As a result, there is a growing trend to extract the subgraph patterns from original data for more efficient and accurate predictions [15].

Even flourishing, there remains two open issues in subgraph-based methods. 1) How to clearly extract major substructure patterns in a graph which mostly explain the predictions [32, 47], 2)

[*] Yang Wang and Zhengyang Zhou are corresponding authors.

KDD '23, August 6–10, 2023, Long Beach, CA, USA.

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599339

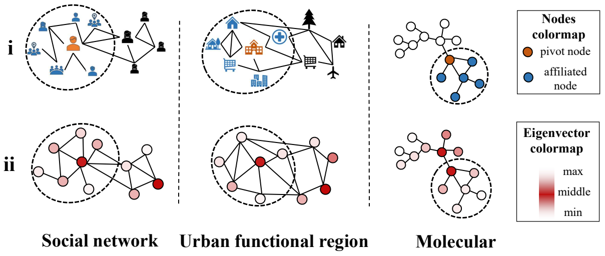


Figure 1: Examples of Local asymmetry in three different scenarios by contrast observation and spectral theory. (i) The asymmetric status between the pivot nodes and the affiliated nodes forms a stable substructure. (ii) The low-frequency eigenvectors often resonate with central nodes in local structures.

how to remove non-useful information, and obtain subgraphs that maximally benefit final prediction for better generalization [18, 26]. Therefore, finding a local structure descriptor to accommodate majority of subgraphs and reducing the target-detrimental information to achieve the minimal but sufficient subgraphs, are two main targets of subgraph-based methods.

Regarding substructure extractions, the mainstream solutions often predefined the subgraph structure with prior knowledge, such as EgoNets in social networks [41, 46], high-frequency functional groups in molecular graphs [19, 20]. However, these criterions of subgraph discovery require knowledge and experiences on specific domains, and consequently hampers the universality in extracting different substructural patterns across various domains [3, 4, 19, 36, 48, 49]. Concerning information refinement, the pioneering Graph Information Bottleneck (GIB) explores the information theory to squash and refine information with edge-wise sampling or node-level dropping [11, 37, 43, 44]. Unfortunately, these existing refinement solutions suffer two limitations. First, refining the whole graph on node levels inevitably distorts the original local topology and corrupt the node-wise correlations, leading to the intervention on following refinements. Then, without an explicit guidance for redundant information elimination, existing refinement solutions only exploit the labels to preserve the minimal information, which is inefficient for their convergence processes. To this end, we can summarize two issues that hinder the existing subgraph learning from achieving universal and robust graph representations, i.e., 1) the predefined substructure derived from domain knowledge usually **lacks universality in subgraph extraction**, 2) sampling on node levels and lacking guidance of redundance elimination introduce **the distorted and inefficient refinement process**. Therefore, how to find a support subgraph set that is minimal but sufficient for learning tasks, is still an open challenge. Fortunately, the following two observations can potentially facilitate to tackle above challenges.

Firstly, to address the universality limitation of previous methods, we discover that there exists a special common pattern of *Local Asymmetry*, shared across the majority subgraphs. As shown in Figure 1(i), subgraphs in web social networks are usually forged by an EgoNet centering with Internet celebrity, which also obeys asymmetric influencer-follower relations, urban functional regions are dominant by the density of major functional POIs but also with several affiliated sites, and the functional groups in molecules are usually identified with the statistical frequencies where it also can be interpreted by anisotropical inter-atomic attractions. Despite the diversity of subgraph formation principles, we can still summarize a common property that contributes to subgraphs, i.e., the *Local Asymmetry* between the centric and affiliated nodes. Quantitatively, consider the well-known property of *Spectral Theorem* discriminating between different graph structures and substructures [22], we exploit eigenvalues to prove our observations. In practices, the k-smaller eigenvectors depict smooth encoding coordinates and reveal a comprehensive electric potential field of neighboring nodes [13, 35]. Such principle motivates us to obtain Figure 1(ii), which theoretically supports our *Local Asymmetry*. More details are provided in Appendix C.

Secondly, the content refinement is usually realized by Information Bottleneck [37], which is prone to be fragile and inefficient as it performs on node levels and lacks guidance for actively eliminating redundance. Therefore, we have two observations about the refinement of subgraphs. 1) nodes in a subgraph exhibit clustering effects and a large real-world graph usually consists of multiple subgraphs with each subgraph accounting for specific properties [2, 5, 48]. 2) on inter-subgraph relationships, each subgraph should inherently reveal remarkable independence where nodes in a specific subgraph can be deemed as a collectivity and such individual collectivity is informative to represent all its members. These two observations, which are considered as the local dependence within subgraphs and inter-independence among subgraphs [37], can advance the refinement towards a more robust and efficient manner. Motivated by the local dependence, simultaneously discovering a bag of subgraphs and improving sampling from node to subgraph levels can promisingly avoid the corruptions of local topology with redundant subgraphs removed integrally. Considering the inefficiency induced by none guidance of information elimination, the inter-independence potentially provides a gathering and dispersion principle for excluding redundant information and thus facilitating the sampling process.

Present work. In this paper, to address the challenges of both general subgraph pattern extraction and subgraph-level refinements, we propose a novel and general subgraph learning scheme, Poly-Pivot Graph Neural Network (P2GNN) where we designate the centric node of each subgraph as the pivots in graphs. Our P2GNN composes of an extractor and a refiner, which extract substructure and refine subgraph information, respectively. For the extractor, we propose a substructure extraction strategy *Hitpath* based on the principle of *Local Asymmetry*. Firstly, we design an improved random walk, to measure the node-wise asymmetry by the differences between pairwise random walk distances, preserving both local topology and feature correlation. Secondly, to adaptively discover diverse subgraphs centered with different pivots, we take Affinity Propagation (AP) clustering as a basic framework and receives node-wise asymmetric similarity from random walk, where the AP clustering enjoys the nice property of modeling interactive relationships between pivot nodes and underlying affiliated points without pre-defining the number of subgraphs. In refiner, we refine information on the subgraph level to maximally avoid the corruption of local topology, and devise a novel Exclusion-based regularization to actively obtain support subgraphs. We also design *Information Shift Method (ISM)* to verify whether the subgraphs learned from P2GNN are with the nice support property for prediction. Specifically, we propose to rank all the subgraphs based on sampling confidence and realize the information shift with probability reassignment. With our proposed *ISM*, we can easily exploit the prediction performance trends to explore the support property of discovered subgraph set.

Our main contributions are summarized as:

• We emphasize the universality and efficiency of subgraph learning, and summarize a general principle of substructure extraction *Local Asymmetry*, which is further justified by spectral theory.

• We present a two-stage subgraph learning scheme P2GNN, which composes of an extractor and a refiner. The analysis of *Hitpath* provides theoretical guarantee on modeling the asymmetric relationship between nodes.

• Extensive experimental results reveal that our P2GNN excels best baselines, where our *Local Asymmetry* can exactly cover subgraph patterns across datasets. A novel evaluation method *ISM* is designed to verify the support property of refined subgraphs.

2 PRELIMINARIES AND DEFINITIONS

Graph. Let $G = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ denote a graph with $|\mathcal{V}|$ nodes. In particular, $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$ is the node set, $\mathcal{E} = \{e_{ij} : \langle v_i, v_j \rangle\}$ is the edge set with each pair of v_i and v_j connected, and $\boldsymbol{W} = \{w_{v_i v_j}\}$ denotes the weight of the edge $\langle v_i, v_j \rangle$. Besides, $d(v_i)$ denotes the degree of node v_i and $\mathcal{N}(v_i)$ is the set of v_i neighbor nodes.

Subgraph. Given a graph G, it can be decomposed into a set of M subgraphs $G_S = \{G_S^1, G_S^2, \dots, G_S^M\}$, where M is adaptively variable to different G. Each disentangled subgraph is denoted as $G_S^i = (\mathcal{V}_S^i, \mathcal{E}_S^i)$. All subgraphs in G_S must satisfy the following two conditions: (1) $\mathcal{V} = \mathcal{V}_S^1 \cup \mathcal{V}_S^2 \cup \dots \cup \mathcal{V}_S^M$, (2) $\mathcal{V}_S^i \cap \mathcal{V}_S^j = \emptyset$ $(\forall i, j \in [1, M],\ i \neq j)$.

Random Walk. It is a time-reversible Markov chain [24]. Given a graph G, and a starting node v_i, in the random walk, v_i will first jump to one of its neighbors with a transition probability, i.e., $p(v_i \to v_j) = \frac{1}{d(v_i)}(v_j \in \mathcal{N}(v_i))$, and then it will successively jump to high-order neighbors with corresponding probability. The sequence of nodes v_i walking through is defined as the walk path on the graph, and $\mathcal{H}(v_i, v_j)$ is denoted as walking distance from v_i to v_j, which is the expected distance that v_i first arrives at v_j.

Problem Definition. The goal of our work is to perform a systematical subgraph learning that finds a support subgraph set for graph representation. Given a graph G, first, we are expected to learn a general subgraph extraction model $P_\theta(G_S|G)$ to obtain a subgraph set $G_S = \{G_S^1, G_S^2, \dots, G_S^M\}$. Second, we further impose the subgraph-level refinement model $P_\phi(G_S^*|G_S)$ on G_S to derive a support subgraph set G_S^* that consists of minimal but sufficient subgraphs, simultaneously improving the performance and interpretation of graph-level classification.

3 POLY-PIVOT GRAPH NEURAL NETWORK

Our P2GNN is a two-stage subgraph learning scheme, consisting of a Local Asymmetry-based substructure extractor and a subgraph-level information refiner via mutual exclusions. The whole technical process is illustrated in Figure 2.

3.1 Substructure Extractor via Local Asymmetry

The goal of the first stage is to generalize the *Local Asymmetry* principle to extract subgraph structure, and to disentangle original graph into a subgraph set. Inspired by the analogy between node clustering and subgraph extraction, we propose a subgraph extractor based on AP algorithm, which employs HitPath as the asymmetric similarity measurement.

As shown in Figure 1, we discover that almost all subgraphs, even graphs across domains, share two rules, i.e., 1) members in a subgraph tend to be physically neighboring, and 2) there is the disparity of statuses among different members where some pivots are prone to attract affiliated nodes to formulate a subgraph community. We summarize such common properties as *Local* and *Asymmetry* and formalize *Local Asymmetry* as a novel principle for subgraph discovery. Given these observations, it is not proper to just mimic previous predefined substructure extraction. We then introduce an approach on how to implement the *Local Asymmetry* principle in graph.

A vital cognition about *Local Asymmetry* is that the cluster formed by the asymmetric attraction of the pivot nodes to the member nodes constitutes the subgraph, which is analogous to clustering of nodes. However, our task is significantly more complex, i.e. irregularity of graph structure leads to uncertain numbers of subgraphs, the determination of the pivot nodes is complicated, and asymmetric metrics are difficult to quantify. These factors hinder applying traditional symmetric clustering strategy to extract subgraphs. Fortunately, Affinity Propagation (AP) clustering algorithm based on message passing mechanism has many excellent properties, which lights up our idea. AP algorithm does not need to prespecify the number of clusters, can find the centers adaptively, and more importantly supports asymmetric similarity matrix. Note that these satisfactory properties all depend on appropriate similarity criteria. Given these evidences, the core challenge is how to design a similarity criteria according to *Local Asymmetry* principle.

Technically, *Local* can be easily implemented by calculating the hop distance $hop(v_i, v_j)$ between pairwise nodes $\langle v_i, v_j \rangle$, while *Asymmetry*, which is expected to quantify the status disparity among members, is too abstract to describe. Random Walk enjoys the nice property of modeling node-wise asymmetric statuses [28, 29]. Nodes in a random walk will stochastically jump to one of its neighbors with a transition probability inversely proportional to the node degree. In this way, the asymmetric statuses induced by the disparate local topology can be exactly captured. However, we argue that traditional random walk suffers two limitations when adopted in our subgraph discovery. First, since node features usually play significant roles in forming subgraphs, these solutions of Random Walk only consider the topology but fail to involve the semantic correlations induced by node features. Second, computing the expectation of walking distances introduces the inefficiency issue [33]. Therefore, we expand a weighted random walk, *HitPath*, by two modifications on the traditional one.

First, to accommodate the semantic similarity of node features, the walking path in our *HitPath* is not only determined by the node-wise connectivity, but also the disparity between node-specific representations. Specifically, consider a walking path $p = (v_{p_0}, v_{p_1}, \dots)$

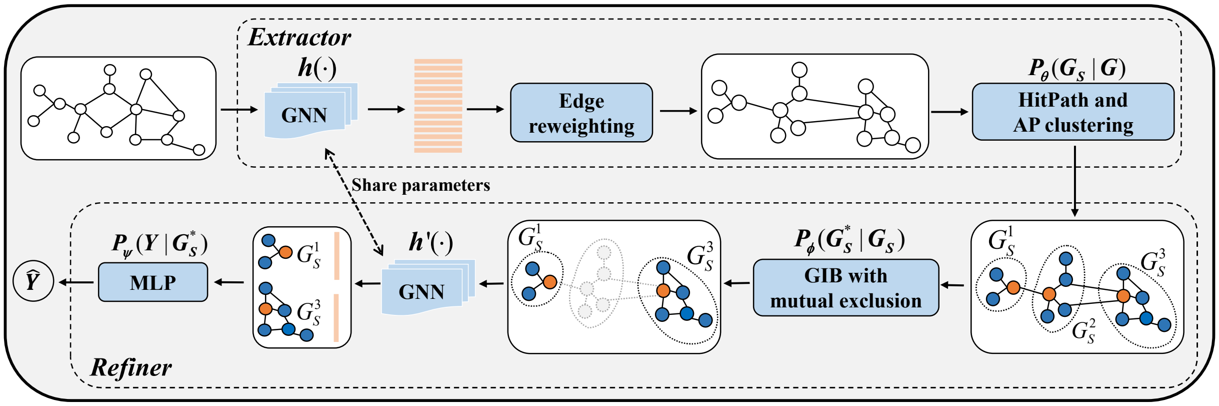


Figure 2: The architecture of our two-stage P2GNN. In the substructure extraction stage, the node-wise associations on both topology and features are explored by *HitPath* to obtain the re-weighted G. Then all nodes are disentangled into a subgraph set G_S. In the subgraph refinement stage, the subgraph-level exclusions are imposed to GIB for facilitating learning process. Finally, P2GNN outputs a set of support subgraphs G_S^* and performs graph classifications. Note that $h(\cdot)$ and $h'(\cdot)$ share the same GNN encoder. The orange nodes are pivots of subgraphs while the blue nodes are affiliated member points.

in graph G , each neighboring transition pair $\langle v_{p_i}, v_{p_{i+1}} \rangle \in \mathcal{E}$ consists of the sequential walking steps. Then we revise the degree-based transition probability into a weighted distance. For node v_i, the probability jumping to v_j in its neighborhood is,

$$P(v_i \rightarrow v_j) = \frac{w_{v_i v_j}}{\sum\limits_{v_k} w_{v_i v_k}} (v_k, v_j \in \mathcal{N}(v_i)) \tag{1}$$

where $w_{v_i v_j}$ is instantiated as the Euclidean distance between their representations learned by a GNN-based encoder $h(\cdot)$,

$$w_{v_i v_j} = ||h(v_i) - h(v_j)||^2 \tag{2}$$

For the transition probability $P(v_i \rightarrow v_j)$, the number of summed terms in the denominator implies the node degree of v_i and thus capturing the topological property, while $w_{v_i v_j}$ serves as a node-wise weighted distance, encapsulating the proximity of feature information. Due to the potentially disparate local neighborhood environments (both topology and feature distributions over neighbors) of different nodes, the node-wise asymmetric relationship can be well characterized by our modified transition probability.

Second, to alleviate the inefficiency issue, a restart strategy along with truncated walking steps is devised to approximate the walking distance $\mathcal{H}$. Given the path p starting at v_{p_0}, the hitting distance from v_{p_0} to any other node $v_{p_k} \in p$ is truncated by a fixed step of l, i.e., walking distance makes sense only when v_{p_0} is accessible to v_{p_k} within l steps, and thus p is modified as $p = (v_{p_0}, v_{p_1}, \ldots, v_{p_l})$. We then also impose a fixed restart times K, to accommodate the tradeoff between performance and efficiency in *HitPath*.Therefore, we have the one-time hitting distance $H\,(v_0, v_k)$ from v_{p_0} to any other node v_{p_k} by summing step-wise feature-based distances along the path p,

$$H(v_0, v_k) = \begin{cases} \sum\limits_{v_i \in (v_0, \ldots, v_{k-1})} w_{v_i v_{i+1}}, v_k \in p \\ \sigma, v_k \notin p \end{cases} \tag{3}$$

where a large upper limit σ will be set as the hitting distance if v_k is out of range of path p. By repeating walking for K times, we can obtain the modified random walking distance in our *HitPath* (directional walking distance) $\tilde{\mathcal{H}}$ between nodes v_i and v_j,

$$\tilde{\mathcal{H}}(v_i, v_j) = HitPath(v_i, v_j) = \frac{1}{K} \sum_{k=1}^{K} H_k(v_i, v_j) \tag{4}$$

Such step truncation in *HitPath* can naturally eliminate the influences from nodes within high-order neighborhoods, and preserve the localization constraint to satisfy *Local Asymmetry* principle.

Furthermore, in the process of subgraph discovery, what we actually focus on is not distinguishing the exact nodes with higher status, but to identify a clustering of nodes with locally asymmetric relationships. In this way, we can leverage the difference between the asymmetric $HitPath(v_i, v_j)$ and $HitPath(v_j, v_i)$ to characterize the disparity of statuses between v_i and v_j.

Finally, considering the *Local* property with hop distances, we derive the quantitative node-wise relationship between v_i and v_j under the *Local Asymmetry* principle,

$$s(v_i, v_j) = \gamma \cdot \mathrm{e}^{-\mathrm{hop}(v_i, v_j)} + (1-\gamma) \cdot ||HitPath(v_i, v_j) - HitPath(v_j, v_i)||^2 \tag{5}$$

This $s(v_i, v_j)$ can also be viewed as a node-wise similarity measurement, where γ balances the importance between the former term for localization and the latter one for asymmetry. Noted that there are two major hyper-parameters in *HitPath*. We empirically let the truncated step l be $\frac{|\mathcal{V}|}{2}$, while the balance coefficient γ be 0.8, and then fine-tune these settings with experiments. More settings can be found in Appendix B.

3.2 Mutual Exclusion-based Subgraph Refiner

In this subsection, we present our subgraph refinement model $P_\phi(G_S^*|G_S)$ as well as the classifier $P_\psi(Y|G_S^*)$ for downstream tasks, where G_S^* is the set of support subgraphs refined from G_S. Recall that we have pointed out that conventional node-level refinement will potentially distort the original local topology and lead to intervention on following refinements [32]. What's more, existing refinement solutions only exploit the label information to preserve the minimal information but neglect any other guidance for redundant information elimination, leading to their inefficient convergence.

To break up above two dilemmas, we devise a subgraph-level refiner. From the perspective of statistics, we are expected to obtain a series of independent and label-relevant subgraphs, which shares similar goals of GIB [37]. However, GIB-based refinements usually seek for one subgraph in a given graph by node-level sampling [43, 44], which are trivially dropped into the topology corruption. In this work, we make modifications to GIB on two aspects, 1) expand the sampling unit of nodes into subgraphs and 2) impose a mutual exclusion regularization to decentralize each subgraph and facilitate the learning process.

Given a graph G, a discovered subgraph set G_S and the label Y of G, our subgraph-level GIB is expected to find a support subgraph set G_S^*, which not only squashes the subgraph information towards labels but also constrains the least number of reserved subgraphs. It is formulated as,

$$\min_{G_S^*} -I(Y, G_S^*) + \beta I(G_S, G_S^*) \tag{6}$$

where β is set as 1 according to common practice [37, 43, 44]. We then elaborate the implementation of above GIB. For the first term $I(Y, G_S^*)$, we exploit the tractable lower bound obtained by [44], and realize it with the standard cross-entropy loss. In particular, this cross-entropy loss is also minimized to materialize the classifier $P_\psi(Y|G_S^*)$ for downstream tasks. For the second term $I(G_S, G_S^*)$, we introduce a variational estimator $q(G_S)$ for the marginal distribution $p(G_S)$, and derive the variational upper bound with KL-divergence, i.e., $I(G_S, G_S^*) \leq KL\left(P_\phi(G_S^*|G_S)||q\left(G_S\right)\right)$ [1]. Actually, since there is no premise or prior knowledge and further and an inaccurate $q(G_S)$ will terribly mislead the refinement and deteriorate the efficiency, finding an accurate variational approximation of $p(G_S)$ is intractable. To explicitly guide eliminating redundant information on subgraph-level sampling, besides a KL-divergence objective, we further introduce another principle of mutual exclusion on subgraphs. This principle interprets a generally recognized but less exploited knowledge on subgraph refinement, i.e., the node representations within the same subgraph should be as close as possible while the representation disparity among different subgraphs should be large. Then we consider this mutual exclusion as a regularization term in our objective to jointly optimize a decentralized and effective set of subgraphs. Technically, let $p_{sub}(\{v_*\} \in \widetilde{G}_S^t)$ be the probability that all the nodes v_* in a tentative $\widetilde{G}_S^t$ forming an exact subgraph, and $p_{sub}(\widetilde{G}_S^p, \widetilde{G}_S^q)$ be the joint occurrence probability of both subgraphs $\widetilde{G}_S^p$ and $\widetilde{G}_S^q$. Therefore, the support subgraph set G_S^* considering the mutual exclusion are expected to obtain the maximal $\sum\limits_{G_S^i \in G_S^*} p_{sub}(G_S^i)$ and the minimal $\sum\limits_{G_S^i \in G_S^*} \sum\limits_{G_S^-} p_{sub}(G_S^i, G_S^-)$, where G_S^- denotes any other subgraph except G_S^i. We can formally derive the regularization objective of mutual exclusion as,

$$\begin{aligned} ME(G_S^*) &= \mathbb{E}_{p_{sub}(G_S^i)}\mathbb{E}_{p_{sub}(G_S^-)}[p_{sub}(G_S^i, G_S^-) - p_{sub}(G_S^i)] \\ &= \mathbb{E}_{p_{sub}(G_S^i)}\mathbb{E}_{p_{sub}(G_S^-)}[\log \sum_{G_S^i \subset G_S^*} \exp(h(G_S^i)^T h(G_S^-)) \\ &\quad - \log \sum_{(v_k, v_m) \sim G_S^i} \exp(h(v_k)^T h(v_m))] \end{aligned} \tag{7}$$

In this way, the second term $I(G_S, G_S^*)$ can be realized by integrating both variational approximation of KL-divergency and mutual exclusion regularization,

$$I(G_S, G_S^*) = KL\left(P_\phi(G_S^*|G_S)||q(G_S)\right) + ME(G_S^*) \tag{8}$$

Optimization objective. Considering the mutual exclusion regularization, we jointly optimize the integrated objectives of P2GNN,

$$\min_{\theta,\phi,\psi} -\mathbb{E}(P_\psi(Y|G_S^*)) + \beta\mathbb{E}\left[KL\left(P_\phi(G_S^*|G_S)||q\left(G_S\right)\right) + ME(G_S^*)\right] \tag{9}$$

So far, we can finally obtain the refined subgraph set G_S^* from the discovered subgraph set G_S.

3.3 Detailed Implementation of P2GNN

P2GNN includes an extractor $P_\theta(G_S|G)$, a refiner $P_\phi(G_S^*|G_S)$ and a classifier $P_\psi(Y|G_S^*)$ for prediction. Besides, a support property evaluation *ISM* is also designed to explore whether the discovered subgraph set G_S^* satisfies the expected support property. We introduce the detailed implementation of P2GNN as follows:

Substructure extractor $P_\theta(G_S|G)$. Our substructure extractor is instantiated as the combination of *HitPath* and AP clustering. The AP clustering exploits two metrics of "Responsibility" and "Availability" [16] to jointly measure the interactive relationships between pivot nodes and potentially affiliated points by considering influences from other nodes, which opportunely match our *Local Asymmetry* principle. Concretely, consider the potential pivot node as v_k and other affiliated point as v_i. The "Responsibility" $r\left(v_i, v_k\right)$ reflects the accumulated evidence for how well-suited the node v_k is to serve as the pivot for v_i, while "Availability" $a\left(v_i, v_k\right)$ reflects the accumulated evidence for how appropriate v_i chooses v_k as its pivot. In this way, by exploiting the *Local Asymmetry*-based node-level relationship measurement $s(\cdot,\cdot)$, the responsibility $r\left(v_i, v_k\right)$ and availability $a\left(v_i, v_k\right)$ in AP clustering can be respectively formalized by,

$$r(v_i, v_k) = \begin{cases} s(v_i, v_k) - \max\limits_{k' \neq k}\{a(v_i, v_{k'}) + s(v_i, v_{k'})\}, v_i \neq v_k \\ s(v_i, v_k) - \max\limits_{k' \neq k}\{s(v_i, v_{k'})\}, v_i = v_k \end{cases} \tag{10}$$

$$a(v_i, v_k) = \begin{cases} \min\{0, r(v_k, v_k) + \sum\limits_{v_j \neq v_i} \max\{r(v_j, v_k), 0\}\}, v_i \neq v_k \\ \sum\limits_{v_j \neq v_k} \max\{r(v_j, v_k), 0\}, v_i = v_k \end{cases} \tag{11}$$

Based on the responsibility and availability iteratively updated by Equation 12, we can obtain a set of pivot nodes $\{v_{c_1}, v_{c_2}, ..., v_{c_M}\}$, and subsequently construct the corresponding subgraphs $G_S = \{G_S^1, G_S^2, \ldots, G_S^M\}$. The node set $\mathcal{V}_S^i$ consists of the node membership in i-th subgraph.

$$\max_{v_k} \quad r(v_i, v_k) + a(v_i, v_k) \tag{12}$$

Subgraph refiner $P_\phi(G_S^*|G_S)$. Subgraph refiner consists of two submodules, a subgraph-level GIB and a subgraph-level mutual exclusion for complementing the lacking of prior knowledge on subgraph clustering. Concretely, the first term in Equation 6 is implemented by the cross-entropy loss for squashing the subgraph information towards labels, and the second term constrained by the KL-divergence and mutual exclusion is devised to preserve the minimal but sufficient information. Given a graph $G = \{G_S^1, ..., G_S^M\} \sim P_G$, we learn the sampling probability p_i at the subgraph level, where G_S^i

will be removed if $p_i = 1$. Inspired by [26], we impose $q_i \sim Bern(t)$ [1] to introduce the stochasticity and further define the variational distribution as

$$\begin{aligned} q(G_S) &= \sum_G P(G_S|G)P_G(G) \\ &= P(G_S^1, G_S^2, \cdots, G_S^M|G) \cdot P_G(G) \\ &= P(G_S^1|G) \cdot P(G_S^2|G) \cdots P(G_S^M|G) \cdot P_G(G) \\ &= P(q_1) \cdot P(q_2) \cdots P(q_M) \cdot P_G(G) \\ &= P_G(G) \prod_{i=1}^{M} P(q_i) \end{aligned} \tag{13}$$

Considering $P_G(G)$ is a constant that can be ignored in optimization, the KL-divergence for the variational approximation is formalized as

$$KL(P_\phi(G_S^*|G_S)||q(G_S)) = \sum_{G_S^i \in G_S} p_i \log \frac{p_i}{t} + (1 - p_i) \log \frac{1 - p_i}{1 - t} \tag{14}$$

Therefore, Equation 14 and Equation 7 jointly constitute of the optimization objective for our g_ϕ.

Classifier $P_\psi(Y|G_S^*)$. The classifier receives the support subgraph set G_S^*, which includes a GNN block with an MLP layer. The GNN block separately encodes each subgraph into corresponding subgraph-level representation where each block shares the same parameters. We then exploit a sum-pooling strategy on the refined set of subgraphs by element-wise addition [39]. The MLP layer imposes linear transformations on compressed subgraph-level representation and finally outputs the categorical predictions for the downstream classification task.

Support property evaluation. To explore whether the discovered subgraph set G_S^* satisfies the expected support property, we devise an *Information Shift Method (ISM)* to reveal the performances of P2GNN when G_S^* is under diverse shifts. In practice, the encapsulated information in our model can be quantified as the number of subgraphs. Specifically, the core idea of *ISM* to impose interventions on G_S^* by decreasing or increasing subgraphs that input into classifier P_ψ, thus we can further verify the refined subgraph set G_S^* perfectly supports the prediction. This information quantification can lead to one issue, i.e., our learning system will be considered as experiencing the same information variation if only the number of subgraph shifts is the same. To this end, a pricinple guiding the ordering of information variation is required. As the learned p_i from P2GNN indicates the removal probability, then it is less confident with its removal decision when it becomes closer to 0.5, thus subgraphs with 0.5 removal probability should be first analyzed. With this insight, we can exploit p_i to rank the priority of each subgraph G_S^i and take 0.5 as the threshold to determine the direction of information shift. Therefore, we divide the G_S^* into two categories and rank them as,

$$S^+ =< G_1^+, G_2^+, ..., G_l^+ > \tag{15}$$

$$S^- =< G_1^-, G_2^-, ..., G_k^- > \tag{16}$$

where $0.5 \leq p(G_1^+) \leq p(G_2^+) \leq, ..., \leq p(G_m^+), p(G_k^-) \leq, ..., \leq p(G_2^-) \leq p(G_1^-) < 0.5$. When extending information, we employ the order of S^+ for subgraph selection to progressively increase information for shift. Otherwise, the order of S^- will be utilized to decrease the information. To perturb the input information to the classifier P_ψ, *ISM* devises a simplified stratege, which modifies the removal probability p_i of these selected subgraphs into equal ones, i.e., 0.5. Specifically, we can formulate this operations by,

$$\text{ISM}(G_S^*, t) = \begin{cases} p(G_1^+) = p(G_2^+) = p(G_t^+) = 0.5, \textit{shift} =' +' \\ p(G_1^-) = p(G_2^-) = p(G_t^-) = 0.5, \textit{shift} =' -' \end{cases} \tag{17}$$

where t denotes the number of selected new subgraphs (information), $'+'$ indicates information expanding while $'-'$ is information decreasing. With our proposed *ISM*, we can easily select the subgraphs for information shifts and obtain the prediction performance trends to explore the support property of G_S^*. The detailed algorithm of P2GNN is provided in Algorithm 1.

Time complexity analysis. For each graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, P2GNN has two parts of calculation: subgraphs extraction and refinement. The execution time of the first stage main comes from calculating the walk distance $HitPath(v_i, v_j)$ and AP clustering. Their corresponding complexities are $O(\frac{|\mathcal{V}|}{2}|\mathcal{V}|)$ and $O(T|\mathcal{V}|^2)$, respectively, where $\frac{|\mathcal{V}|}{2}$ is truncated step, T is the number of iterations before convergence or max iterations. The reason for not including the hop distance calculation is that it has been calculated before training. In the subgraphs refinement stage, exploring the attention of each subgraph will take $O(M)$ time, where M is the number of subgraphs from subgraphs extraction. Therefore, the time complexity of P2GNN is $O((T + \frac{1}{2})|\mathcal{V}|^2 + M)$.

3.4 Theoretical Analysis

In our work, the asymmetric relationship between nodes is not directly measured by the ordering of status, but is measured with the relative status disparity. In this section, we will discuss two manners of status disparity measurement and justify the rationality of our solution with theoretical analysis.

Firstly, according to literature [24], we have two principles to describe the relationships among the walking distances across different nodes, i.e., Lemma 1 and Lemma 2.

Lemma 1. Given any three nodes v_i, v_j and v_k in graph G, the pairwise walking distances can be described as

$$\mathcal{H}(v_i, v_j) + \mathcal{H}(v_j, v_k) \geq \mathcal{H}(v_i, v_k) \tag{18}$$

Lemma 2. (Symmetric walk distances) Given two looped walk paths, the summations of walking distances on these two paths preserve invariant,

$$\mathcal{H}(v_i, v_j) + \mathcal{H}(v_j, v_k) + \mathcal{H}(v_k, v_i) = \mathcal{H}(v_i, v_k) + \mathcal{H}(v_k, v_j) + \mathcal{H}(v_j, v_i) \tag{19}$$

Proposition 1. Given $\mathcal{H}(v_i, v_j) > \mathcal{H}(v_j, v_i)$ that the statues v_i precedes v_j, this ordering is ambiguous in the whole graph, i.e., the ordering *Asymmetry* in graph is not transitive, while the relative difference between the walking distance $\mathcal{H}(v_k, v_i) - \mathcal{H}(v_i, v_k)$ can exactly capture the asymmetry of node-wise statuses, and quantify the status disparity between two nodes.

Proof. First, consider the three nodes v_i, v_j and v_k where v_i precedes v_j and v_j precedes v_k i.e. $\mathcal{H}(v_i, v_j) > \mathcal{H}(v_j, v_i)$ and $\mathcal{H}(v_j, v_k) > \mathcal{H}(v_k, v_j)$. We dissect the order of v_i and v_k through progressively deriving the following inequalities,

$$\mathcal{H}(v_i, v_j) + \mathcal{H}(v_j, v_k) > \mathcal{H}(v_j, v_i) + \mathcal{H}(v_k, v_j) \tag{20}$$

$$\mathcal{H}(v_i, v_j) + \mathcal{H}(v_j, v_k) > \mathcal{H}(v_k, v_i) \tag{21}$$

[1] Parameter t controls the sampling probability in Bernoulli distribution.

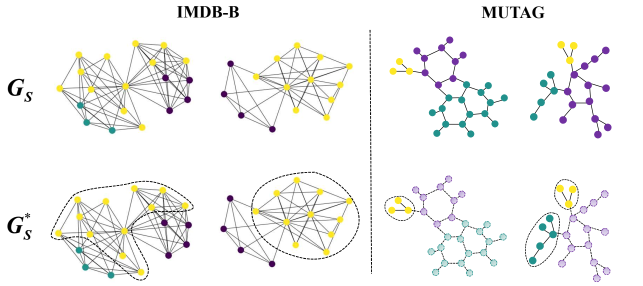


Figure 3: The universality and effectiveness of P2GNN on social networks and bioinformatics. Nodes with the same color belong to the same subgraph, and the captured support subgraphs are circled by the dashed line.

$$\mathcal{H}(v_i, v_k) + \mathcal{H}(v_k, v_j) + \mathcal{H}(v_j, v_i) > 2\mathcal{H}(v_k, v_i) \tag{22}$$

$$\mathcal{H}(v_i, v_k) - \mathcal{H}(v_k, v_i) > \mathcal{H}(v_k, v_i) - \mathcal{H}(v_k, v_j) - \mathcal{H}(v_j, v_i) \tag{23}$$

Since $\mathcal{H}(v_k, v_i) \le \mathcal{H}(v_k, v_j) + \mathcal{H}(v_j, v_i)$ (Lemma 1), we have that $\mathcal{H}(v_i, v_k) > \mathcal{H}(v_k, v_i)$ is not always hold on, demonstrating the status relationship between v_i and v_k is ambiguous. Therefore, we can conclude that *Asymmetry* in graphs does not have the property of transitivity.

Second, considering v_k as a fixed intermediate node and assuming that v_i precedes v_j where these two nodes are both anchored on v_k, we can get the following derivations,

$$\mathcal{H}(v_i, v_k) - \mathcal{H}(v_k, v_i) \ge \mathcal{H}(v_j, v_k) - \mathcal{H}(v_k, v_j) \tag{24}$$

$$\mathcal{H}(v_i, v_k) + \mathcal{H}(v_k, v_j) \ge \mathcal{H}(v_j, v_k) + \mathcal{H}(v_k, v_i) \tag{25}$$

$$\mathcal{H}(v_i, v_j) \ge \mathcal{H}(v_j, v_i) \tag{26}$$

To this end, the relative difference can lead to a unique asymmetric ordering between v_i and v_j. Therefore, $\mathcal{H}(v_k, v_i) - \mathcal{H}(v_i, v_k)$ can definitely represent the *Asymmetry* in graphs.

Remark. Since the ordering of status relationships is not transmissible, only the relative walking distance can characterize the status disparity among nodes. In our implementation, we instantiate $||\mathcal{H}(v_i, v_j) - \mathcal{H}(v_j, v_i)||^2$ to measure the status disparity, where $\mathcal{H}$ is alternatively approximated by *HitPath* proposed in Sec 3.1.

4 EXPERIMENT

4.1 Experimental Settings

4.1.1 *Datasets*.

The experimnents are conducted on eight public datasets regarding bioinformatics and social networks. More details of these datasets are summarized in Table 2 of Appendix A.

- **Bioinformatics datasets:** MUTAG [10], PROTEINS [6], and MOLHIV [38].
- **Social network datasets:** IMDB-B [40] and IMDB-M [40], REDDIT-B [40] and REDDIT-M [40], COLLAB [40].

4.1.2 *Baselines*.

Our baselines are three-fold, including GNN backbones, subgraph learning methods, and interpretable models. • **Backbone baselines:** GCN [21], GraphSAGE [17], GIN [39] and PNA [9].

- **Subgraph methods:** GNN-AK [48], GIB [44] and SUGAR [32].
- **Interpretable models:** GSAT [26] and CAL [31].

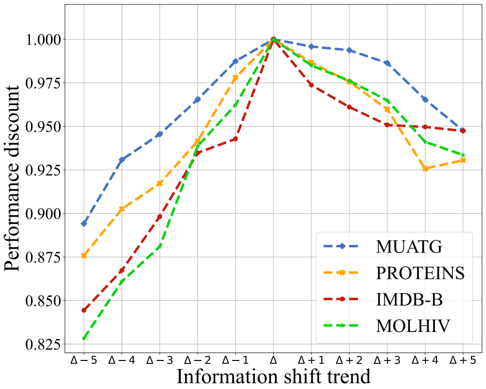


Figure 4: The performance discount with shifting the information of support subgraphs. Δ represents the information of G_S^*.

4.1.3 *Our Setups*.

We provide some important training hyper-parameters and metrics for different datasets.

- **Hyper-parameters.** We take GIN as the backbone of P2GNN. The balance coefficient γ is set to 0.8, and the walking step l is set to $\frac{|\mathcal{V}|}{2}$. The parameter sensitivity analysis is performed in Sec 4.4, with more detailed implementations presented in the Appendix B.
- **Metrics.** Following the common practice [39, 48], we report ROC-AUC on MOLHIV while present classification accuracy for all other datasets.

4.2 Result Analysis

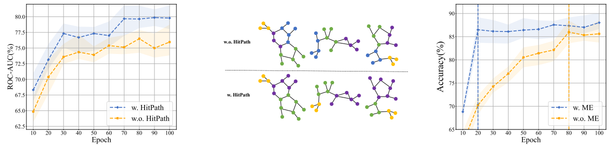
Table 1 shows the performance comparisons across different methods, and we can obtain the following three **Obs**ervations.

Obs 1: The methods based on subgraph discovery consistently outperform the traditional backbone models on all datasets. In backbone methods, GIN and PNA have achieved competitive performance, GIN (89.9%) on Reddit-B and PNA (79.1%) on MOLHIV could even obtain the expression ability of subgraph learning methods. In fact, both of them are often used as the backbone in subgraph learning methods. This observation demonstrates that subgraph learning can indeed improve the graph representation ability. Almost all subgraph methods greatly improve the prediction accuracy, however, we obtain the counter-examples on two complex social network datasets, such as Reddit-M and COLLAB. The maximal performance drop among subgraph learning baselines is 6.9% on Reddit-M, and PNA and GIB achieved comparable prediction accuracy on COLLAB. This phenomenon reveals the common shortcomings of subgraph learning methods on some social network datasets.

Obs 2: Compared with other subgraph learning models, our P2GNN achieves the most competitive results where we achieve the SOTA on four datasets. Specifically, our P2GNN outperforms best baselines by 6.3% and 1.3% respectively on IMDB-B and MUTAG, and encouragingly, P2GNN has significantly improved over other methods on most scenarios except three complex social network datasets. Such performance superiority can be explicitly attributed to the coupling effects of both two objectives, i.e., asymmetry-based subgraph extraction and robust refinement.

We believe that the two-stage subgraph learning can exactly better extract label-relevant subgraphs. Further, it is worth noting that our P2GNN does not have the best performance on the REDDIT-B, REDDIT-M and COLLAB. The underlying reasons are that 1) unlike the edges of friends, the interactions of users in REDDIT-B and REDDIT-M are more random thus it is less discriminative on QA community and discussion-based community, and due to the superior performance of CAL on them, we consider that the causal perspective may be able to further solve the common problem of subgraph discovery learning. 2) subgraph-level refinement may lead to information loss if the local pattern is not well captured. Therefore, given the large number of nodes in graphs, extraction and refinement on large-scale heterogeneous graphs with noisy or non-robust edges are still under explored.

Obs 3: The support subgraph set has the most sufficient and minimal information for prediction. We verify the support property of our discovered subgraphs using visualization method and *ISM* . Figure 3 visualizes the discovered subgraph set G_S and the support subgraph set G_S^* with our P2GNN. On IMDB-B, different patterns of the ego-network centering with various actors/actresses are bond to determine the genre of the movie [40]. Encouragingly, our support subgraphs captured by P2GNN can reveal prominent consistency with such ego-network. For one graph of MUTAG, P2GNN tends to take these two functional groups $-NO_2$ and $-NH_2$, or some substructures containing these functional groups, as support subgraphs. By looking into the chemical explanations in literature [25], we find that the ground-truth interpretation corresponds to our empirical results on MUTAG. Based on information theory, we also quantitatively analyze the support property of our discovered subgraphs via designing *ISM*. Centered on the support subgraph set G_S^*, we respectively explore the performance trends of set when it expands and decreases information. Figure 4 visualizes the results of *ISM* on three datasets. Reducing the information content of the support subgraph set G_S^* will obviously lead to a decrease in the prediction power. Also, we are more surprised to find that continuously expanding G_S^* doesn't improve the prediction ability and even hurt the accuracy of the prediction.



(a) Performance comparison on MOLHIV.

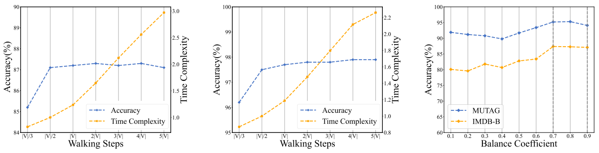
(b) Comparison of the discovered subgraphs on MOLHIV.

(c) Comparison of convergence speed on IMDB-B.

Figure 5: Ablation studies on *HitPath* and mutual exclusion in P2GNN.

4.3 Ablation Study

Ablation studies consist of two aspects. 1) We replace the *HitPath* with a symmetrical subgraph discovery measure with hop distance to verify the effectiveness of asymmetry measures. 2) We ablate the mutual exclusion (*ME*) principle in GIB to demonstrate whether our solution can exactly facilitate the learning convergence. Due to the limited space, we only illustrate studies on two datasets, more comprehensive evaluations can be found in Appendix D.



(a) Variations of walking steps l on IMDB-B.

(b) Variations of walking steps l on MUTAG.

(c) Variations of balance coefficient γ.

Figure 6: Parameter sensitivity analysis.

Figure 5(a) and 5(b) respectively illustrate the performances and visualized discovered subgraphs (with and without *HitPath*) on MOLHIV. Subgraph discovery based on asymmetry measures obviously reveals the better performances than hop-based measures, where solution w.o. *HitPath* only focuses on the topology and neglects the feature information. And most subgraphs discovered by solution w.o. *HitPath* are prone to be in a small size, probably due to the lacking connections of feature correlations. We can conclude that the asymmetric topology and feature correlations modeled by *HitPath* is superior to original random walk for subgraph discovery.

Figure 5(c) shows the comparison of convergence speeds between our P2GNN and variant without *ME*. Intuitively, we have a faster convergence speed with the mutual exclusion, verifying that the mutual exclusion principle can be viewed as a prior knowledge, to provide effective guidance for subgraph sampling and makes it easier to obtain label-relevant representation. It should not be ignored that our performance is also better than w.o. *ME*, which suggests that the mutual exclusion also plays an important role in the downstream tasks for better representation.

4.4 Parameter Sensitivity Analysis

The main hyper-parameters in P2GNN are two-fold. 1) The walking step l in *HitPath*, and 2) the topology-feature balance coefficient γ in Equation 5.

Figure 6 shows the performance and training time with different l on various datasets. With increasing l, especially when $l > \frac{|\mathcal{V}|}{2}$, the increased training time haven't led to better performance. Thus, to simultaneously balance efficiency and accuracy, we set $l = \frac{|\mathcal{V}|}{2}$. We analyze that the walking distance l should be neither set too small nor a fixed value per graph. When the value of walking distance l is too small, the randomness of walking makes the process may be limited to a few nodes, resulting in local information cannot be captured. Fixed l is often unreasonable for graphs of different sizes. Therefore, although the purpose of *HitPath* is to encode local information, we still walking more than half of the number of nodes $\frac{|\mathcal{V}|}{2}$.

Figure 6(c) shows the fluctuation of performances under different γ. Obviously, we get stable results at $\gamma \in [0.7, 0.9]$. Then we set γ as 0.8 in our experiments. More hyper-parameter settings are carefully described in Table 3 in Appendix B.

Table 1: Performance comparisons. The best result is in bold across all methods and the second best is underlined.

	MUTAG	PROTEINS	IMDB-B	IMDB-M	Reddit-B	Reddit-M	COLLAB	MOLHIV
GCN	74.3±11.0	74.2±3.1	70.0±0.9	51.5±3.2	85.5±2.1	48.6±2.3	69.6±2.1	75.5±1.6
Graph-SAGE	74.3±7.7	73.0±4.5	70.9±4.1	47.6±3.5	84.3±1.9	50.0±1.3	71.6±1.5	74.8±3.4
GIN	89.4±5.6	77.0±4.3	75.6±3.7	48.5±3.3	89.9±1.9	56.1±1.7	73.9±1.7	75.6±1.4
PNA	89.6±5.3	76.7±4.2	79.8±4.5	48.0±2.0	81.7±6.1	54.2±1.2	74.2±2.1	79.1±1.3
GNN-AK	92.3±6.8	77.1±5.1	75.2±3.1	53.2±1.2	**94.6±1.0**	54.8±2.1	78.5±1.8	79.2±1.1
GIB	83.9±6.4	77.2±3.4	73.7±7.0	51.4±2.1	90.3±1.9	49.2±2.0	74.5±2.0	76.4±2.7
SURGAR	92.4±2.1	81.0±2.4	80.1±2.8	51.1±1.9	89.4±2.1	50.1±2.4	77.4±1.7	77.0±3.1
GSAT	94.1±2.1	76.2±1.4	72.6±4.4	54.5±3.2	85.4±2.0	56.2±1.7	81.8±1.4	78.1±2.0
CAL	89.9±8.3	76.9±3.3	74.1±5.2	52.6±2.4	91.2±2.3	**57.1±1.9**	**82.7±1.3**	78.1±2.0
P2GNN	**95.4±2.9**	**82.1±3.1**	**87.4±3.9**	**56.2±2.1**	91.4±2.4	56.8±2.0	81.6±1.9	**79.8±1.4**

5 RELATED WORK

Graph representation model. There is a growing interest in exploring more expressive graph learning model. Most previous works designed more powerful node-level learning methods, such as GCN [21], GraphSAGE [17], GAT [34] and GIN [39], which rely on stronger Aggregate, Combine and Readout functions. Along another line of research, subgraph-based representation strategy has been proposed to provide more intuitive understanding and better interpretability. These methods expect to obtain subgraphs or network motifs which are simple building blocks of complex networks and determine the functionalities of graphs. More details, we categorize these strategies into two classes: extracting substructure patterns to capture certain topological structure of subgraphs, and removing non-useful information to capture subgraphs that affect the model predictions the most. The former focuses on extracting the structure patterns of subgraph from the perspective of topology, and the latter aims to refine the most valuable subgraphs for learning tasks from the perspective of information.

Substructure extraction. Most previous works on substructure extraction tend to determine substructure patterns based on domain knowledge or directly adopt ego-net structure. [14, 27] propose density-based subgraph discovery methods for some specific areas, including network science, biological analysis, and graph databases. GNN-AK [48] directly applies star-pattern subgraph and convolves all subgraphs with a base GNN as kernel, which produce multiple rich subgraph-node embeddings. SubGNN [2] decouples the graph topology to three property-aware channels, and designs three structure patterns subgraph which capture position, neighborhood, and structure. XGNN [45] trains a graph generator to interpret GNNs by edge-wise sampling, which provides an understanding of which parts contributing to final predictions. However, these methods reveal great limitations. First, scenario-based subgraph discovery models greatly lack generality. Then, the substructure discovery methods of ego-net still follow MPNNs' local neighbor aggregation, which don't take advantage of subgraph representation learning.

Information refinement. Recent works borrow the information theory to develop the Graph Information Bottleneck (GIB). These GIB-based solutions squash the original graph into one minimal sufficient subgraph by removing the redundant nodes [43, 44]. Unfortunately, given the personalized structures of subgraphs across domains, and the entangled correlations within subgraphs, existing subgraph learning scheme reveals two limitations. First, there still lack a general principle to accommodate various subgraph structures across multiple domains. Second, refinement performing on node levels without explicit guidance for eliminating redundant information tend to distort the local topology and lead to an inefficiency convergence process.

6 CONCLUSION

In this paper, we present a two-stage subgraph learning architecture P2GNN, which performs general subgraph extract and efficient refinement. In the extraction stage, *Local Asymmetry* is proposed to accommodate diverse domain-specific subgraphs discovery tasks. We design a novel node-wise asymmetry measurement *HitPath*, and achieve subgraph extraction via *AP* clustering. Theoretical analysis of *Hitpath* is also provided to verify its asymmetry measuring capacity. In the refinement stage, we propose the principle of mutual exclusion regularization to explicitly guide eliminating redundant information and thus boost the efficiency of refinement. Further, we empirically verify the superiority on universality and effectiveness via experiments and propose *ISM* to prove the support property of discovered subgraph set.

Limitations: In the subgraphs extraction stage, personalizing the number of subgraphs according to domain knowledge may result in better predictive power, which remains unexplored. And in the subgraphs refinement stage, the analysis of our refinement method from the perspective of causality can be further studied.

ACKNOWLEDGMENTS

This paper is partially supported by the National Natural Science Foundation of China (No.62072427, No.12227901), the Project of Stable Support for Youth Team in Basic Research Field, CAS (No.YSBR-005), Academic Leaders Cultivation Program, USTC.

REFERENCES

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410* (2016).

[2] Emily Alsentzer, Samuel Finlayson, Michelle Li, and Marinka Zitnik. 2020. Subgraph neural networks. *Advances in Neural Information Processing Systems* 33 (2020), 8017–8029.

[3] Yunsheng Bai, Derek Xu, Yizhou Sun, and Wei Wang. 2021. Glsearch: Maximum common subgraph detection via learning to search. In *International Conference on Machine Learning*. PMLR, 588–598.

[4] Pablo Barceló, Floris Geerts, Juan Reutter, and Maksimilian Ryschkov. 2021. Graph neural networks with local graph parameters. *Advances in Neural Information Processing Systems* 34 (2021), 25280–25293.

[5] Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M Bronstein, and Haggai Maron. 2022. Equivariant Subgraph Aggregation Networks. In *International Conference on Learning Representations*.

[6] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. 2005. Protein function prediction via graph kernels. *Bioinformatics* 21, suppl_1 (2005), i47–i56.

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

[8] Fan Chung and S-T Yau. 2000. Discrete Green's functions. *Journal of Combinatorial Theory, Series A* 91, 1-2 (2000), 191–214.

[9] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. 2020. Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems* 33 (2020), 13260–13271.

[10] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. 1991. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry* 34, 2 (1991), 786–797.

[11] Huiqi Deng, Qihan Ren, Hao Zhang, and Quanshi Zhang. 2021. DISCOVERING AND EXPLAINING THE REPRESENTATION BOTTLENECK OF DNNS. In *International Conference on Learning Representations*.

[12] Lun Du, Xiaozhou Shi, Qiang Fu, Xiaojun Ma, Hengyu Liu, Shi Han, and Dongmei Zhang. 2022. GBK-GNN: Gated Bi-Kernel Graph Neural Networks for Modeling Both Homophily and Heterophily. In *Proceedings of the ACM Web Conference 2022*. 1550–1558.

[13] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2020. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982* (2020).

[14] Yixiang Fang, Kaiqiang Yu, Reynold Cheng, Laks VS Lakshmanan, and Xuemin Lin. 2019. Efficient algorithms for densest subgraph discovery. *arXiv preprint arXiv:1906.00341* (2019).

[15] Fabrizio Frasca, Beatrice Bevilacqua, Michael Bronstein, and Haggai Maron. 2022. Understanding and extending subgraph gnns by rethinking their symmetries. *Advances in Neural Information Processing Systems* 35 (2022), 31376–31390.

[16] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science* 315, 5814 (2007), 972–976.

[17] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[18] Yifan Hou, Jian Zhang, James Cheng, Kaili Ma, Richard TB Ma, Hongzhi Chen, and Ming-Chang Yang. 2022. Measuring and improving the use of graph information in graph neural networks. *arXiv preprint arXiv:2206.13170* (2022).

[19] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2020. Hierarchical generation of molecular graphs using structural motifs. In *International conference on machine learning*. PMLR, 4839–4848.

[20] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2020. Multi-objective molecule generation using interpretable substructures. In *International conference on machine learning*. PMLR, 4849–4859.

[21] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[22] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. 2021. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems* 34 (2021), 21618–21629.

[23] A Lasota and James A Yorke. 1982. Exact dynamical systems and the Frobenius-Perron operator. *Transactions of the american mathematical society* 273, 1 (1982), 375–384.

[24] László Lovász. 1993. Random walks on graphs. *Combinatorics, Paul erdos is eighty* 2, 1-46 (1993), 4.

[25] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. *Advances in neural information processing systems* 33 (2020), 19620–19631.

[26] Siqi Miao, Mia Liu, and Pan Li. 2022. Interpretable and Generalizable Graph Learning via Stochastic Attention Mechanism. In *International Conference on Machine Learning*. PMLR, 15524–15543.

[27] Dung Nguyen and Anil Vullikanti. 2021. Differentially private densest subgraph detection. In *International Conference on Machine Learning*. PMLR, 8140–8151.

[28] Giannis Nikolentzos and Michalis Vazirgiannis. 2020. Random walk graph neural networks. *Advances in Neural Information Processing Systems* 33 (2020), 16211–16222.

[29] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

[30] Christoph Schweimer, Christine Gfrerer, Florian Lugstein, David Pape, Jan A Velimsky, Robert Elsässer, and Bernhard C Geiger. 2022. Generating Simple Directed Social Network Graphs for Information Spreading. In *Proceedings of the ACM Web Conference 2022*. 1475–1485.

[31] Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. 2022. Causal attention for interpretable and generalizable graph classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1696–1705.

[32] Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Yuanxing Ning, Philip S Yu, and Lifang He. 2021. Sugar: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. In *Proceedings of the Web Conference 2021*. 2081–2091.

[33] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*. IEEE, 613–622.

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[35] Haorui Wang, Haoteng Yin, Muhan Zhang, and Pan Li. 2022. Equivariant and stable positional encoding for more powerful graph neural networks. *arXiv preprint arXiv:2203.00199* (2022).

[36] Pengkun Wang, Chuancai Ge, Zhengyang Zhou, Xu Wang, Yuantao Li, and Yang Wang. 2021. Joint Gated Co-attention Based Multi-modal Networks for Subregion House Price Prediction. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[37] Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. 2020. Graph information bottleneck. *Advances in Neural Information Processing Systems* 33 (2020), 20437–20448.

[38] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical science* 9, 2 (2018), 513–530.

[39] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[40] Pinar Yanardag and SVN Vishwanathan. 2015. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1365–1374.

[41] Carl Yang, Mengxiong Liu, Vincent W Zheng, and Jiawei Han. 2018. Node, motif and subgraph: Leveraging network functional blocks through structural convolution. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 47–52.

[42] Nianzu Yang, Kaipeng Zeng, Qitian Wu, Xiaosong Jia, and Junchi Yan. 2022. Learning substructure invariance for out-of-distribution molecular representations. In *Advances in Neural Information Processing Systems*.

[43] Junchi Yu, Jie Cao, and Ran He. 2022. Improving subgraph recognition with variational graph information bottleneck. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19396–19405.

[44] Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. 2020. Graph information bottleneck for subgraph recognition. *arXiv preprint arXiv:2010.05563* (2020).

[45] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 430–438.

[46] Yanfu Zhang, Hongchang Gao, Jian Pei, and Heng Huang. 2022. Robust Self-Supervised Structural Graph Neural Network for Social Network Prediction. In *Proceedings of the ACM Web Conference 2022*. 1352–1361.

[47] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. 2022. Protgnn: Towards self-explaining graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 9127–9135.

[48] Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. 2021. From stars to subgraphs: Uplifting any GNN with local structure awareness. *arXiv preprint arXiv:2110.03753* (2021).

[49] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Hengchang Liu. 2020. RiskOracle: a minute-level citywide traffic accident forecasting framework. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 1258–1265.

[50] Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. 2021. Graph neural networks with heterophily. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11168–11176.

A DETAILS OF DATASETS

We provide the details and statistics of datasets in this section, where the statistics of these eight datasets are illustrated in Table 2.

• **MUTAG** [10] is a binary dataset of molecular property, where nodes are atoms and edges are chemical bonds. Each graph is associated with a binary label based on its mutagenic effect.

• **PROTEINS** [6] is a dataset of proteins that are classified as enzymes or non-enzymes. Nodes represent the amino acids and two nodes are connected if they are less than 6 Angstroms apart.

• **IMDB-B** and **IMDB-M** [40] are two movie collaboration datasets, where nodes represent actors/actress. There is an edge between nodes if they appear in the same movie.

• **REDDIT-B** and **REDDIT-M** [40] are two datasets of social networks, where nodes represent users. There is an edge between nodes if the comment interaction has appeared between them.

• **COLLAB** [40] is a scientific collaboration dataset, derived from 3 public collaboration datasets, namely, High Energy Physics, Condensed Matter Physics and Astro Physics.

• **MOLHIV** [38] is a molecular property dataset, where nodes are atoms and edges are chemical bonds. Each molecule has a binary label, which depends on whether the molecule can inhibit HIV virus replication or not.

Algorithm 1 P2GNN

```
Input: Graph $G$, the number of learning epochs $N$, the maximum iterations
    of AP clustering $T$.
Output: The prediction $\hat{Y}$ of graph $G$.
Initialization: $\theta, \phi, \psi$, GNN encoder $h(\cdot)$.
for $n = 0$ to $N$ do
    $w_{v_i v_j} \leftarrow ||h(v_i) - h(v_j)||^2$
    for $t = 0$ to $T$ do
        /*     $P_\theta(G_S|G)$     */
        AP clustering to disentangle $G$ and obtain $G_S$:
        if convergence then
            break
        end if
    end for
    /*     $P_\phi(G_S^*|G_S)$     */
    Sample a subgraph set $G_S^*$ from $G_S$ .
    /*     $P_\psi(Y|G_S^*)$     */
    Calculate $\mathcal{L}$ with Equation 9.
    $\theta, \phi, \psi \leftarrow$ Adam($\theta, \phi, \psi$).
end for
Predict by support subgraph set $\hat{Y} = P_\psi(Y|G_S^*)$.
return $\hat{Y}$
```

B DETAILS OF THE CONFIGURATION

To ensure fair comparisons on all datasets, we keep the same configurations on all of them and present Table 3 to summarize the detailed configurations.

C ANALYSIS OF SPECTRAL THEOREM

In this paper, we employ *Spectral Theorem* to verify the observation of Local Asymmetry, and our asymmetric metric criterion (*HitPath*) using random walk is also inspired by spectral decomposition. Therefore, it is necessary to construct a comprehensive understanding of *Spectral Theorem*, and we provide a detailed analysis in this section.

Table 2: Statistics of datasets.

Dataset	Graphs	Classes	Avg. Nodes	Avg. Edges
MUTAG	188	2	17.93	19.79
PROTEINS	1,113	2	39.06	72.82
IMDB-B	1,000	2	19.77	96.53
IMDB-M	1,500	3	13.00	65.94
REDDIT-B	2,000	2	429.63	497.75
REDDIT-M	4,999	5	508.52	594.87
COLLAB	5,000	3	74.49	2457.78
MOLHIV	41,127	2	25.50	27.50

The idea of *Spectral Theorem* is to study graph structure via its Laplacian operator of graph. According to different interpretation strategies of eigenvectors, many works have achieved great success.

C.1 Position encoding with Spectral Theorem

In this subsection, we provide an analysis of the application of *Spectral Theorem* to the field of Position encoding (PE), and demonstrate the observation of Local Asymmetry from our understanding perspective as shown in Figure 1(ii).

Given a graph G, $d(v_i)$ denotes the degree of the node v_i in G, and let $\boldsymbol{W}$ denote adjacency matrix. We perform the following analysis.

In electromagnetic theory, the Green's function of the Laplacian [8] shows the electrostatic potential of a given charge. This understanding inspired the PE of nodes on the graph. Consider the Laplacian G and can be computed by its eigenfunctions,

$$G(j_1, j_2) = d_{j_1}^{-\frac{1}{2}} d_{j_2}^{\frac{1}{2}} \sum_{i>0} \frac{(\alpha_{i,j_1}\alpha_{i,j_2})^2}{\hat{\lambda}_i} \tag{27}$$

Further, researchers [7] use the interaction between two heat kernels to define in Equation 27 the diffusion distance d_D between nodes j_1, j_2,

$$d_D^2(j_1, j_2) = \sum_{k>0} e^{-2t\lambda_i}(\alpha_{i,j_1} - \alpha_{i,j_2})^2 \tag{28}$$

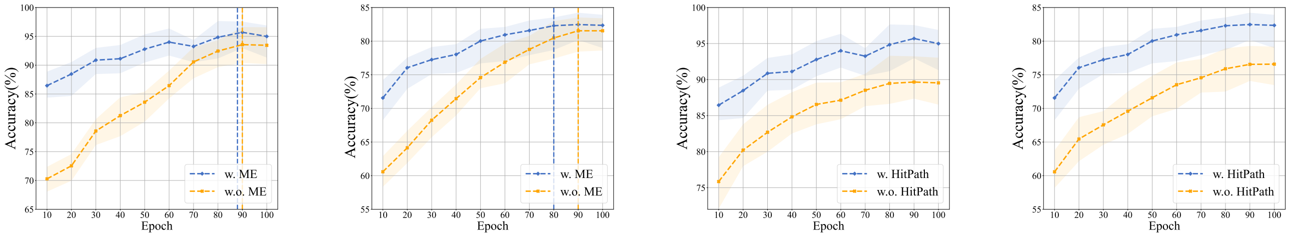
Inspired by it, the biharmonic distance d_B was proposed as a better measure of distances [22],

$$d_B^2(j_1, j_2) = \sum_{i>0} \frac{(\alpha_{i,j_1} - \alpha_{i,j_2})^2}{\lambda_i^2} \tag{29}$$

Equation29 shows that smaller frequencies/eigenvalues are more heavily weighted when determining distances between nodes. Thus, we study the 5-th lowest eigenvectors and draw heat maps Figure 1(ii). We are excited to find that there is a perfect match for our proposed Local Asymmetry. Therefore, we successfully prove Local Asymmetry via exploiting *Spectral Theorem.*

C.2 Random walk with Spectral Theorem

In this subsection, we analyze random walks from the perspective of spectral theory, and subsequently supplement the preliminary knowledge of **Lemma 1** and **Lemma 2**.



(a) Performance comparison on MUTAG. (b) Performance comparison on PROTEINS. (c) Performance comparison on MUTAG. (d) Performance comparison on PROTEINS.

Figure 7: Ablation study on *ME* and *HitPath*.

Table 3: Configurations in P2GNN.

Config	Description	Value
Backbone	The backbone of P2GNN	GIN
Smoothing coefficient α_r	α_r in Algorithm 1	0.5
Smoothing coefficient α_a	α_a in Algorithm 1	0.5
Balance coefficient β	β in Equation 9	1
Balance coefficient γ	γ in Equation 5	0.8
Walking steps l	Walking steps of *HitPath*	$\frac{\lvert\mathcal{V}\rvert}{2}$
K	Restart times of *HitPath*	100
σ	Distance between inaccessible nodes	$4\lvert\mathcal{V}\rvert$
Batchsize	Number of graphs in a batch	128
Split	Train set/Validation set/Test set	8/1/1
Hidden dimension	Hidden dimension of backbone	64
Base learning rate	Initial learning rate	1e-3
Dropout	Dropout rate of MLP	0.3

Actually, the cornerstone of our theoretical analysis is *Spectral Theorem*, which draws the topological characteristics of graphs from spectral perspective [22]. Here, we will briefly describe the connection between *Spectral Theorem* and random walks.

In the traditional strategy, the walking distance is calculated by the transition matrix $\boldsymbol{M} = \boldsymbol{DW}$, where M_{ij} represents the probability that node v_i jumps to v_j in one step. For a high-order formation, we let $\boldsymbol{M}_{ij}^t$ stand for the transition probability from v_i to v_j at the step t. Thus, the expected walking distance $\mathcal{H}$ can be calculated by,

$$\mathcal{H} = \sum_{t=1}^{\infty} t \cdot M^t \tag{30}$$

In contrast, exploiting *Spectral Theorem* to calculate the walking distance has more elegant properties where we can arrive a closed form of our distance. Consider the matrix $\boldsymbol{N} = \boldsymbol{D}^{1/2}\boldsymbol{W}\boldsymbol{D}^{1/2} = \boldsymbol{D}^{-1/2}\boldsymbol{M}\boldsymbol{D}^{1/2}$. Due to the symmetry of $\boldsymbol{N}$, it can be written in the spectral form,

$$\boldsymbol{N} = \sum_{k=1}^{n} \lambda_k \alpha_k \alpha_k^T \tag{31}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ are the eigenvalues of $\boldsymbol{N}$ and $\alpha_1, \cdots, \alpha_n$ are corresponding eigenvectors of unit length. By Frobenius-Perron Theorem [23], we have $\lambda_1 = 1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -1$, and

$$\boldsymbol{M}^t = \boldsymbol{D}^{1/2}\boldsymbol{N}^t\boldsymbol{D}^{-1/2} = \sum_{k=1}^{n} \lambda_k^t \boldsymbol{D}^{1/2}\alpha_k\alpha_k^T\boldsymbol{D}^{-1/2} \tag{32}$$

Further, we can obtain walking distance from v_s to v_t,

$$\mathcal{H}(v_s, v_t) = 2m \sum_{k=2}^{n} \frac{1}{1-\lambda_k}\left(\frac{\alpha_{kt}^2}{d(t)} - \frac{\alpha_{ks}\alpha_{kt}}{\sqrt{d(s)d(t)}}\right) \tag{33}$$

Therefore, Equation 33 can exactly provide a closed-form solution to obtain the walking distance, which is superior to traditional strategy. By this exact solution, the principles in **Lemma 1** and **Lemma 2** can be easily derived.

D DETAILED ABLATION STUDY

We provide the detailed ablation studies on all datasets, which are shown in Figure 7. For ablations on *HitPath*, it is worth noting that the performance of solution with *HitPath* is still going to increase even achieving 100 epochs, manifesting that *HitPath* can better capture the correlation between nodes with ever-increasing epochs. For ablations on *ME*, we observe that the convergence speeds of these two solutions are similar on MUTAG and MOLHIV. We speculate that the subgraphs G_S discovered by P2GNN in the bioinformatic datasets are naturally mutually exclusive, so the regularization of *ME* does not significantly improve the convergence speeds. Even so, the performance of our method is still better than that w.o. *ME*.

E BROADER IMPACTS

As shown in Figure 4, our P2GNN automatically captures label-relevant functional groups on the chemical datasets. However, we also mentioned that P2GNN's performance on the social network datasets drops slightly.

We consider the main reason of such suboptimal performance on social networks is the edge diversity property, which is crucial for subgraph extraction. The node-wise relations in social networks can be classified into friendship, collaboration, and common interests, etc. Even, the edges of certain collaboration also tend to contain diverse information. Unfortunately, such unavailable edges features in our datasets lead to much difficulty in subgraph extraction. But in contrast, the edges in the molecular graph described by interatomic force are homogeneous. Then such edge type homogeneity can contribute to successful subgraphs extraction. Therefore, we will explore the homogenous relationship between the nodes of the graph to facilitate the in-depth study of asymmetry.