Differentiable Meta Multigraph Search with Partial Message Propagation on Heterogeneous Information Networks

Chao Li[1,2,*], **Hao Xu**[1,2,*], **Kun He**[1,2,†]

[1]School of Computer Science, Huazhong University of Science and Technology, Wuhan 430074, China.
[2]Hopcroft Center on Computing Science, Huazhong University of Science and Technology, Wuhan 430074, China.
{D201880880, M202173589, brooklet60}@hust.edu.cn

Abstract

Heterogeneous information networks (HINs) are widely employed for describing real-world data with intricate entities and relationships. To automatically utilize their semantic information, graph neural architecture search has recently been developed on various tasks of HINs. Existing works, on the other hand, show weaknesses in instability and inflexibility. To address these issues, we propose a novel method called *Partial Message Meta Multigraph search* (PMMM) to automatically optimize the neural architecture design on HINs. Specifically, to learn how graph neural networks (GNNs) propagate messages along various types of edges, PMMM adopts an efficient differentiable framework to search for a meaningful meta multigraph, which can capture more flexible and complex semantic relations than a meta graph. The differentiable search typically suffers from performance instability, so we further propose a stable algorithm called partial message search to ensure that the searched meta multigraph consistently surpasses the manually designed meta-structures, *i.e.*, meta-paths. Extensive experiments on six benchmark datasets over two representative tasks, including node classification and recommendation, demonstrate the effectiveness of the proposed method. Our approach outperforms the state-of-the-art heterogeneous GNNs, finds out meaningful meta multigraphs, and is significantly more stable. Our code is available at https://github.com/JHL-HUST/PMMM.

Introduction

Heterogeneous information networks (HINs) are widespread in the real world for abstracting and modeling complex systems for the rich semantic information provided by heterogeneous relations consisting of multiple node types, edge types, and node features. For instance, bibliographic graphs (such as DBLP) with three types of nodes, *i.e.*, author, paper, and venue, include multiple edge types, such as co-authorships, co-citations, publishing in the same venue.

To model the heterogeneous structure, there have been numerous heterogeneous GNNs combining meta-paths to learn with HINs (Schlichtkrull et al. 2018; Zhang et al. 2019a; Wang et al. 2019a; Yun et al. 2019a). More recently, in light of the achievement of neural architecture search (NAS) in

---

*These authors contributed equally.

†Corresponding author.

convolutional neural networks (CNNs), several works have extended NAS to heterogeneous GNNs, based on the ability to automate the customization of neural architecture for specific datasets and tasks.

However, there are some weaknesses in existing NAS methods for heterogeneous GNNs. For instance, GEMS (Han et al. 2020) uses an evolutionary algorithm as the search strategy, making its search cost dozens of times as training a single GNN. HGNAS (Gao et al. 2021), employing reinforcement learning as the search strategy, has the same issue of inefficiency. Inspired by the simplicity and computational efficiency of differentiable architecture search (Liu, Simonyan, and Yang 2019) in CNNs, DiffMG (Ding et al. 2021) proposes to search a meta graph in a differentiable fashion, making the search cost on a par with training a single GNN once. Yet, it suffers from performance instability and often finds architectures worse than hand-designed models.

Besides, existing works have a common weakness of inflexibility. For example, GEMS (Han et al. 2020) and DiffMG (Ding et al. 2021) search for meta graphs to guide how GNNs propagate messages. However, meta-paths and meta graphs are initially defined for hand-designed heterogeneous GNNs with fixed patterns, so they are insufficient to encode various rich semantic information on diverse HINs and will restrict the searched architecture to inflexible topology. HGNAS (Gao et al. 2021) searches for message encoding and aggregation functions but propagates messages using predefined meta-paths or meta graphs. It motivates us to propose a more expressive meta-structure for NAS on HINs.

To this end, we present a new method called PMMM (Partial Message Meta Multigraph search), which is composed of a novel differentiable search algorithm for stable search, namely partial message search, as well as an innovative meta-structure for flexible topology, namely meta multigraph. Specifically, to stabilize the differentiable search, PMMM randomly selects partial candidate message passing steps to update per iteration to ensure that all candidate paths are equally and fully trained, as well as to decouple the joint optimization of paths. To derive flexible topology, PMMM searches for a novel meta multigraph by selecting the top-k most promising candidate message passing types for aggregation. Searching for a meta multigraph is a free performance-enhancing strategy as it can en-
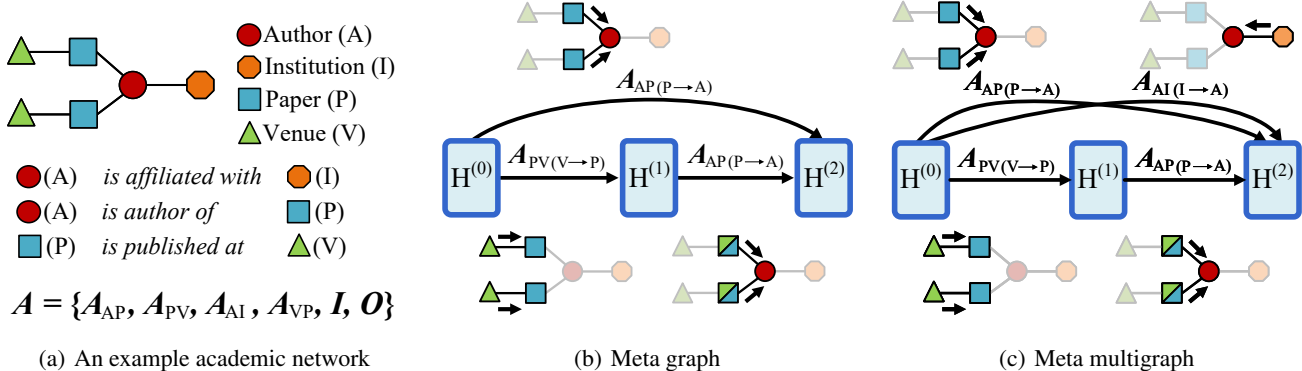
Figure 1: Illustration of (a) an example academic network, (b) meta graph, and (c) meta multigraph with depth $N = 2$. A is the set of candidate message passing types. I and O stand for the identity matrix and empty matrix, respectively. A meta multigraph allows propagating multiple types of message between nodes, which can not be realized by a meta graph.

code more flexible and sophisticated semantic information in HINs compared with a meta-path or meta graph. Experiments show that PMMM consistently outperforms state-of-the-art baselines and significantly improves the stability compared with differential meta graph search on both node classification and recommendation tasks. Our main contributions are threefold:

- To our knowledge, PMMM is the first NAS method to search multigraph as the architecture. We propose a new concept of meta multigraph to guide GNNs to propagate messages for NAS on HINs.

- We propose the first stable differentiable architecture search algorithm on HINs, called partial message search, which can consistently discover promising architectures that outperform hand-designed models.

- Thorough experiments are conducted to demonstrate the effectiveness, stability, and flexibility of our method.

Related Works

Heterogeneous GNNs

Heterogeneous GNNs are proposed to handle HINs as rich and diverse information could be better utilized. One category of heterogeneous GNNs uses hand-designed meta-paths to define neighbors, including HAN (Wang et al. 2019b), MAGNN (Fu et al. 2020), NIRec (Jin et al. 2020), *et al*. Different from these methods, our PMMM does not require specific prior knowledge. Another category aims to solve the meta-path selection conundrum via fusing various edge types based on the attention mechanisms, including GTN (Yun et al. 2019b), HetGNN (Zhang et al. 2019b), HGT (Hu et al. 2020), *et al*. Compared with these methods, PMMM can find efficient architectures by filtering out unrelated edge types.

Graph Neural Architecture Search

Neural architecture search (NAS) has shown promising results in searching for convolutional neural networks since it has brought up the prospect of automating the customization of neural architectures for specific tasks (Zoph and

Le 2017; Xie and Yuille 2017; Liu, Simonyan, and Yang 2019). Recently, numerous NAS-based works have been proposed to obtain data-specific homogeneous GNN topology (Zhou et al. 2019; Qin et al. 2021; Zhao, Yao, and Tu 2021; Wei, Zhao, and He 2022). Meanwhile, only a few works attempt to employ NAS in HINs due to the complex semantic relationships. GEMS (Han et al. 2020) is the first NAS method on HINs that uses an evolutionary algorithm to search for meta graphs for recommendation. HG-NAS (Gao et al. 2021) searches for message encoding and aggregation functions by using reinforcement learning. Considering the inefficiency of evolutionary algorithm and reinforcement learning, DiffMG (Ding et al. 2021) employs an differentiable algorithm to search for meta graphs. However, its performance is unstable. Compared with them, PMMM can perform an efficient and stable search for different tasks on HINs. Furthermore, our model searches for a meta multigraph, which shows better diversity and a stronger capacity to capture complex semantic information.

Definitions

We first provide several definitions used in the literature.

Definition 2.1 Heterogeneous Information Network (HIN) (Sun et al. 2011a). An HIN is defined as a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{R}, f_{\mathcal{T}}, f_{\mathcal{R}}\}$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of edges, $\mathcal{T}$ is the node-type set and $\mathcal{R}$ is the edge-type set. Each node $v \in \mathcal{V}$ and each edge $e \in \mathcal{E}$ are associated with their type mapping functions $f_{\mathcal{T}}(v) \in \mathcal{T}$ and $f_{\mathcal{R}}(e) \in \mathcal{R}$, respectively. We define its network schema as $\mathcal{S} = \{\mathcal{T}, \mathcal{R}\}$, with $|\mathcal{T}| > 1$ or $|\mathcal{R}| > 1$.

Definition 2.2 Meta-path (Sun et al. 2011b). A meta-path P is a path with length l defined on the schema $\mathcal{S} = \{\mathcal{T}, \mathcal{R}\}$ of $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{R}, f_{\mathcal{T}}, f_{\mathcal{R}}\}$, and is denoted in the form of $t_1 \xrightarrow{r_1} t_2 \xrightarrow{r_2} ... \xrightarrow{r_l} t_{l+1}$, where $t_1, ..., t_{l+1} \in \mathcal{T}$ and $r_1, ..., r_l \in \mathcal{R}$. One meta-path can correspond to multiple meta-path instances in the underlying HIN.

Definition 2.3 Meta Graph. A meta graph is a directed acyclic graph on the network schema $\mathcal{S}$ with a single source node $t_s \in \mathcal{T}$ (with zero in-degree) and a single sink (target)

node $t_e \in \mathcal{T}$ (with zero out-degree).

As shown in Fig. 1 (b), a meta graph can only propagate one message passing type between two nodes, which is insufficient to encode rich semantic information on HINs. One extreme example is when all candidate message passing types are necessary, a meta graph will have extremely bad performance due to the restriction of only retaining one message passing type. Another example is shown in Fig. 1 (c). The meta multigraph in (c) allows the author to aggregate the initial information of both paper and institution, while the meta graph in (b) can only aggregate the initial information of either paper or institution. Based on the above concerns, we define the meta multigraph to facilitate the description of our method.

Definition 2.4 Meta Multigraph. A meta multigraph defined on network schema $\mathcal{S}$ is a directed acyclic multigraph consisting of multi-edges and hyper-nodes. Each multi-edge $\mathcal{R}_i \subseteq \mathcal{R}$ consists of multiple edges. Each hyper-node $\mathcal{T}_j \subseteq \mathcal{T}$ is the set of the heads of its incoming multi-edges and the tails of its outgoing multi-edges. A meta multigraph contains a single source hyper-node and a single sink (target) hyper-node.

A meta multigraph allows propagating multiple message passing types between two different hyper-nodes, offering a natural generalization of a meta graph.

Methodology

In this section, we first briefly show the framework of the proposed differentiable meta multigraph search on HINs, then we develop a partial message search algorithm and meta multigraph deriving strategy, showing how they improve the stability and generate flexible and effective meta multigraphs, respectively.

Differentiable Meta Multigraph Search

Differentiable meta multigraph search is developed on differentiable meta graph search (DiffMG) (Ding et al. 2021). The training of differentiable meta multigraph search consists of two stages. At the search stage, we train a super-net, from which sub-networks can be sampled exponentially. The super-net is constructed by directed acyclic multigraphs, whose multi-edges consist of multiple paths corresponding to the candidate message passing types. Each candidate message passing type is weighted by architecture parameters, which are jointly optimized with the super-net weights in an end-to-end manner. The goal of the search stage is to determine the architecture parameters. At the evaluation stage, the strongest sub-network is preserved as the target-net by pruning redundant paths based on the searched architecture parameters. The target-net is then retrained from scratch to get the final results.

We define the search space of differentiable meta multigraph search as a directed acyclic multigraph, in which the ordered nodes $\boldsymbol{H} = \{\boldsymbol{H}^{(0)}, \boldsymbol{H}^{(1)}, \cdots, \boldsymbol{H}^{(n)}, \cdots, \boldsymbol{H}^{(N)}\}$ denote the hyper-nodes in the message passing process and the multi-edges $E = \{\mathcal{R}^{(i,j)} | 0 \leq i < j \leq N\}$ signify the message passing types between hyper-nodes. Each hyper-node $\boldsymbol{H}^{(n)} \subseteq \mathcal{T}$. $\boldsymbol{H}^{(0)}$ and $\boldsymbol{H}^{(N)}$ is the input and out-

put of the meta multigraph, respectively. Each multi-edge $\mathcal{R}^{(i,j)} \subseteq \mathcal{R}$ contains multiple paths, corresponding to candidate message passing types. For $r \in \mathcal{R}^{(i,j)}$, we use $\mathcal{A}_r^{(i,j)}$ to denote the adjacency matrix formed by the edges of type r in $\mathcal{G}$. The core idea of differentiable meta multigraph search is to formulate the information propagated from $\boldsymbol{H}^{(i)}$ to $\boldsymbol{H}^{(j)}$ as a weighted sum over all candidate message passing steps, namely:

$$\boldsymbol{H}^{(j)} = \sum_{i<j} \sum_{r \in \mathcal{R}^{(i,j)}} p_r^{(i,j)} f\left(\mathcal{A}_r^{(i,j)}, \boldsymbol{H}^{(i)}\right), \quad (1)$$

$$p_r^{(i,j)} = \exp(\alpha_r^{(i,j)}) / \sum_{r \in \mathcal{R}^{(i,j)}} \exp(\alpha_r^{(i,j)}). \quad (2)$$

Here $f(\mathcal{A}_r^{(i,j)}, \boldsymbol{H}^{(i)})$ denotes one message passing step that aggregates $\boldsymbol{H}^{(i)}$ along $\mathcal{A}_r^{(i,j)}$, $\alpha_r^{(i,j)}$ indicates the architecture parameters of $\mathcal{A}_r^{(i,j)}$, and $p_r^{(i,j)} \in (0, 1]$ denotes the corresponding path strength calculated by a softmax over $\alpha_r^{(i,j)}$. $f(\cdot)$ can be any aggregation function. Following DiffMG (Ding et al. 2021), we employs the aggregation function in graph convolutional network (GCN).

The parameter update in Eq. 1 involves a bilevel optimization problem (Anandalingam and Friesz 1992; Colson, Marcotte, and Savard 2007; Xue et al. 2021):

$$\min_{\boldsymbol{\alpha}} \quad \mathcal{L}_{val}(\boldsymbol{\omega}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \quad (3)$$

$$\text{s.t.} \quad \boldsymbol{\omega}^*(\boldsymbol{\alpha}) = \text{argmin}_{\boldsymbol{\omega}} \ \mathcal{L}_{train}(\boldsymbol{\omega}, \boldsymbol{\alpha}), \quad (4)$$

where $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ denote the training and validation loss, respectively. The goal of the search stage is to find $\boldsymbol{\alpha}^*$ that minimizes $\mathcal{L}_{val}(\boldsymbol{\omega}^*, \boldsymbol{\alpha}^*)$.

At the evaluation stage, we derive a compact meta multigraph by pruning redundant paths based on path strengths $p_r^{(i,j)}$ determined by the architecture parameters. The meta multigraph is then retrained from scratch to generate node representations for different downstream tasks, *i.e.*, node classification and recommendation.

Partial Message Based Search Algorithm

DiffMG is most related to our differentiable multigraph search. In each iteration, DiffMG samples one candidate message passing type on each edge for the forward propagation and backpropagation, and the strongest message passing type is most likely to be sampled, resulting in random and insufficient training. DiffMG is efficient and outperforms existing baselines. However, one limitation of DiffMG lies in its instability. As illustrated in Figure 4 of the experiments, DiffMG is only effective in a few random seeds and the performance dramatically declines in most random seeds.

To address the instability issue in DiffMG, intuitively we wish to ensure that all message passing types are equally and fully searched in the search stage. An alternative solution is to employ Eq. 1, which trains all possible message passing steps together and formulates the information propagated as
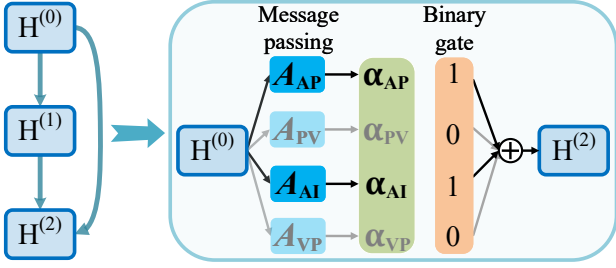
Figure 2: Illustration of the proposed partial message search algorithm.

a weighted sum over all the paths. However, the architecture parameters of various paths in Eq. 1 are deeply coupled and jointly optimized. The greedy nature of the differentiable methods inevitably misleads the architecture search due to the deep coupling (Guo et al. 2020), especially when the number of candidate paths is large. It motivates us to propose a new search method for the search stage to achieve stable meta multigraph search as well as overcome the coupling in optimization.

To overcome the coupling optimization, we define a binary gate $M_r^{(i,j)}$ for each message passing type, which assigns 1 to the selected message passing types and 0 to the masked ones. Specifically, we let paths in each multi-edge be sampled equally and independently, and we set the proportion of $M_r^{(i,j)} = 1$ to $1/p$ by regarding p as a hyperparameter. Then we can get the set of all active paths between hyper-node $\boldsymbol{H}^{(i)}$ and $\boldsymbol{H}^{(j)}$:

$$S^{(i,j)} = \{r | M_r^{(i,j)} = 1, \forall r \in \mathcal{R}^{(i,j)}\}. \quad (5)$$

As illustrated in Figure 2, by introducing the binary gates, only $1/p$ paths of message passing steps are active. Then we formulate the information propagated from $\boldsymbol{H}^{(i)}$ to $\boldsymbol{H}^{(j)}$ as a weighted sum over active candidate message passing steps:

$$\boldsymbol{H}^{(j)} = \sum_{i<j} \sum_{r \in S^{(i,j)}} p_r^{(i,j)} f\left(\boldsymbol{\mathcal{A}}_r^{(i,j)}, \boldsymbol{H}^{(i)}\right), \quad (6)$$

where path strength $p_r^{(i,j)}$ is calculated by Eq. 2. Since message sampling masks $M^{(i,j)}$ are involved in the computation graph, parameters updated in Eq. 6 can be calculated through backpropagation. The overall algorithm is given in Algorithm 1.

Meta Multigraph-Based Architecture Derivation

Once the training of architecture parameters has been completed, we can then derive the compact architecture by pruning redundant paths. As far as we know, all existing differentiable architecture search algorithms for CNNs choose one path with the highest path strength on each edge. As retaining multiple paths means employing multiple types of operations between two node representations, which will damage the performance in most cases.

DiffMG inherits the derivation strategy of these methods to generate meta graphs. However, instead of operations,

---

Algorithm 1: Search algorithm

Require:
  Network weights $\boldsymbol{\omega}$; Architecture parameters $\boldsymbol{\alpha}$; Number of iterations T; Sampling proportion $1/p$.
Ensure:
  Searched architecture parameters $\boldsymbol{\alpha}$.
 1: Initialize network weights $\boldsymbol{\omega}$ and architecture parameters $\boldsymbol{\alpha}$
 2: **for** each iteration $t \in [1, T]$ **do**
 3:   Randomly sample $1/p$ candidate message passing steps in each edge. The collection of network weights and architecture parameters of sampled paths is denoted as $\bar{\boldsymbol{\omega}}$ and $\bar{\boldsymbol{\alpha}}$, respectively
 4:   Update weights $\bar{\boldsymbol{\omega}}$ by $\nabla_{\bar{\boldsymbol{\omega}}} \mathcal{L}_{train}(\bar{\boldsymbol{\omega}}, \bar{\boldsymbol{\alpha}})$
 5:   Execute step 3 again
 6:   Update parameters $\bar{\boldsymbol{\alpha}}$ by $\nabla_{\bar{\boldsymbol{\alpha}}} \mathcal{L}_{val}(\bar{\boldsymbol{\omega}}, \bar{\boldsymbol{\alpha}})$
 7: **end for**
 8: **return** Searched architecture parameters $\boldsymbol{\alpha}$.

DiffMG searches the message passing types that determine which messages are propagated between different types of node representations in a meta graph. Considering that an HIN consists of multiple node types and edge types, deriving a single message passing type between two different node representations is insufficient and inflexible to encode rich semantic information. As shown in Figure 1 (c), there are multiple paths between $H^{(0)}$ and $H^{(2)}$, which can not be learned by the traditional derivation strategy. Another issue caused by deriving a single path is that some effective message passing types with similar but weaker path strengths will be dropped. So simply selecting the message passing type with the highest path strength may reject potentially good architectures.

To address the above issues, we propose to derive a meta multigraph as a heterogeneous message passing layer, making the message passing types more diverse than simply retaining the one with the highest path strength. An alternative solution to derive a meta multigraph is to set a threshold τ, message passing types with path strengths above which are retained. As path strengths are keeping changing during the search stage, τ needs to be changed with them. Here, we set the threshold $\tau^{(i,j)}$ as a value between the largest and the smallest path strengths in each multi-edge $\mathcal{R}^{(i,j)}$:

$$\tau^{(i,j)} = \lambda \cdot \max_{r \in \mathcal{R}^{(i,j)}} \{p_r^{(i,j)}\} + (1-\lambda) \cdot \min_{r \in \mathcal{R}^{(i,j)}} \{p_r^{(i,j)}\}, \quad (7)$$

where $p_r^{(i,j)}$ is inherited from the search stage, $\lambda \in [0,1]$ is a hyper-parameter controlling the number of retrained paths in each multi-edge.

Then, we formulate the information propagated from $\boldsymbol{H}^{(i)}$ to $\boldsymbol{H}^{(j)}$ in the derived meta multigraph as an unweighted sum over candidate paths with path strengths above $\tau^{(i,j)}$:

$$\boldsymbol{H}^{(j)} = \sum_{i<j} \sum_{r \in \hat{S}^{(i,j)}} f\left(\boldsymbol{\mathcal{A}}_r^{(i,j)}, \boldsymbol{H}^{(i)}\right), \quad (8)$$

$$\hat{S}^{(i,j)} = \{r | p_r^{(i,j)} \geq \tau^{(i,j)}, \forall r \in \mathcal{R}^{(i,j)}\}, \quad (9)$$

| Method | Search | | Derivation |
	Coupling	Probability	
DARTS	✓	high	single path
PC-DARTS	✓	high	single path
ProxylessNAS	✗	low	single path
SPOS	✗	low	single path
DiffMG	✗	low	single path
PMMM	✗	middle	multiple paths

Table 1: Comparison of PMMM with related NAS algorithms. *Coupling* denotes whether all architecture parameters are coupling optimized. *Probability* denotes the updating probability of each path in one iteration.

where $\hat{S}^{(i,j)}$ is the set of all retained paths between hypernode $\boldsymbol{H}^{(i)}$ and $\boldsymbol{H}^{(j)}$. Then, the derived meta multigraph can be used as the target-net for retraining from scratch.

Differences to Prior Works

Table 1 details the differences between our approach and related differentiable NAS algorithms, including DARTS (Liu, Simonyan, and Yang 2019), PC-DARTS (Xu et al. 2020), ProxylessNAS (Cai, Zhu, and Han 2019), SPOS (Guo et al. 2020), and DiffMG (Yao et al. 2020). The first four algorithms search convolution or pooling operations in CNNs instead of meta-structures in GNNs. We ignore these differences and focus on the algorithms. In contrast to DARTS and PC-DARTS, our search algorithm does not jointly optimize all the architecture parameters, which reduces the inconsistency between the search and evaluation phase. Compared to ProxylessNAS, SPOS, and DiffMG, our search algorithm updates each message passing step with a higher probability, avoiding unfairness caused by insufficient training. Regarding the derivation strategy, our approach is distinct from all the above methods.

Experiments

For experiments, we first compare PMMM with baselines on two representative tasks to evaluate its performance. We then evaluate the efficiency of PMMM and visualize our searched architectures to analyze how meta multigraph improves the performance. We also compare PMMM with differentiable meta graph search to show its stability under various settings. In the end, we conduct parameter analysis on hyperparameter λ to show the flexibility of PMMM.

Experimental Setup

Datasets We evaluate our method on two popular tasks (Yang et al. 2020): node classification and recommendation. The goal of the node classification task is to predict the correct labels for nodes based on network structure and node features. We use three widely-used real-world datasets : DBLP, ACM, and IMDB. Regarding the recommendation task, we aim to predict links between source nodes (*e.g.*, users) and target nodes (*e.g.*, items). We adopt three widely used heterogeneous recommendation datasets: Amazon, Yelp, and Douban Movie (abbreviated as Douban). The

details of all datasets are shown in Appendix[1].

Baselines We compare our method with eleven methods, including: 1) a random walk based network embedding method, metapath2vec (Dong, Chawla, and Swami 2017); 2) two homogeneous GNNs, *i.e.*, GCN (Kipf and Welling 2017) and GAT (Velickovic et al. 2018); 3) five heterogeneous GNNs, *i.e.*, HAN (Wang et al. 2019b), MAGNN (Fu et al. 2020), GTN (Yun et al. 2019b), HGT (Hu et al. 2020), and GraphMSE (Li et al. 2021); 4) three AutoML methods, *i.e.*, GEMS (Han et al. 2020) for recommendation, HGNAS (Gao et al. 2021) for node classification, and DiffMG (Ding et al. 2021). Specially, HGNAS employs a different data division and does not provide the source code, so we do a separate comparison with HGNAS using its data division in Appendix. More details of these methods and the differences with our model can be found in Appendix.

Parameter Settings Following DiffMG, we run the search algorithm three times with different random search seeds to derive the meta multigraph from the run that achieves the best validation performance. For a fair comparison, the candidate message passing types are the same with DiffMG and the searching epochs are set to 30. We set $p = 2$ for most datasets, *i.e.*, only $1/2$ paths are randomly sampled on each edge, except for DBLP and ACM with small K, we set $p = 1$. To coordinate with baselines, we set the steps $N = 4$, which is the same as the length of meta-paths learned by GTN and meta graph searched by DiffMG. Besides, we set $\lambda = 0.9$. To demonstrate the effectiveness of our meta multigraph strategy, we combine the search algorithm of DiffMG and our meta multigraph derivation strategy to obtain new architectures, which we call Multigraph, and compare them with DiffMG. We put the settings for baselines in Appendix. Experiments are conducted on a single RTX 2080 Ti GPU with 11GB memory.

Evaluation Metrics For evaluation, we use Macro-F1 score as well as Micro-F1 score for the node classification task and AUC (area under the ROC curve) for the recommendation task. For each method, we repeat the process for 10 runs with different random training seeds and report the average score and standard deviation.

Comparison on Node Classification

Table 2 summarizes the results of the proposed model and the baselines. First, Heterogeneous GNNs relying on manually designed meta-paths, such as HAN and MAGNN, do not achieve desirable performance. They can even perform worse than homogeneous GNNs like GAT, which suggests that hand-crafted rules may even have adverse implications. Second, DiffMG outperforms heterogeneous GNNs employing meta-path, demonstrating the power of NAS as well as the advantages of meta graphs over meta-paths. However, the performance of DiffMG is fragile as it relies on carefully chosen random seeds as shown in the subsequent Figure 4. Finally, PMMM outperforms all the state-of-the-art

---

[1]Appendix can be found at https://arxiv.org/pdf/2211.14752.pdf.

Architecture	DBLP		ACM		IMDB	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
metapath2vec	89.93±0.45	88.62±0.44	67.13±0.50	68.24±0.41	40.82±1.48	43.47±1.21
GCN	90.46±0.41	91.47±0.34	92.56±0.20	91.22±0.64	55.19±0.99	54.63±1.03
GAT	93.92±0.28	93.39±0.30	92.50±0.23	92.86±0.29	53.37±1.27	53.64±1.43
HAN	92.13±0.26	92.05±0.62	91.20±0.25	91.57±0.38	55.09±0.67	55.32±0.81
MAGNN	92.81±0.30	92.58±0.21	91.15±0.19	91.06±0.33	56.44±0.63	56.60±0.72
GTN	93.98±0.32	94.01±0.28	92.62±0.17	91.78±0.59	59.68±0.72	59.40±1.44
HGT	93.67±0.22	93.49±0.25	91.83±0.23	91.12±0.23	59.35±0.79	59.42±0.74
GraphMSE	94.08±0.14	94.44±0.13	92.58±0.50	92.54±0.49	57.60±2.13	62.37±1.03
DiffMG	94.45±0.15	94.85±0.27	92.65±0.15	92.21±0.31	61.04±0.56	61.81±1.59
Multigraph (ours)	**94.75±0.11**	95.38±0.13	**93.36±0.09**	**93.27±0.09**	61.46±0.77	63.44±1.01
PMMM(ours)	94.69±0.10	**95.40±0.10**	92.76±0.14	92.65± 0.12	**61.69±0.40**	**64.31±0.77**

Table 2: Macro-F1 (%) and Micro-F1 (%) on the node classification task (mean in percentage ± standard deviation). The best and second-best results are shown in bold and underlined, respectively.

Architecture	Amazon	Yelp	Douban
metapath2vec	58.17±0.14	51.98±0.14	51.60±0.07
GCN	66.64±1.00	58.98±0.52	77.95±0.05
GAT	55.70±1.13	56.55±0.05	77.58±0.33
HAN	67.35±0.11	64.28±0.20	82.65±0.08
MAGNN	68.26±0.09	64.73±0.24	82.44±0.17
GEMS	70.66±0.14	65.12±0.27	83.00±0.05
GTN	71.82±0.18	66.27±0.31	83.26±0.10
HGT	74.75±0.08	68.07±0.35	83.38±0.06
DiffMG	75.28±0.08	68.77±0.13	83.78±0.09
Multigraph (ours)	75.32±0.04	**69.48±0.13**	**83.90±0.02**
PMMM (ours)	**75.48±0.04**	69.27±0.13	83.88±0.02

Table 3: AUC (%) on the recommendation task (mean in percentage ± standard deviation).

Method	Amazon	Yelp	Douban
GEMS	800	1500	2000
DiffMG	0.7	1.1	1.6
PMMM	0.9	1.4	2.1

Table 4: Search cost compared with automated machine learning algorithms on HINs, measured in GPU minutes.
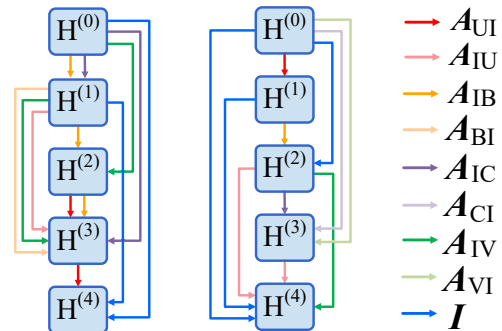

Figure 3: Architectures searched on Amazon.

models consistently, which further indicates the importance of automatically leveraging task-dependent semantic information in HINs. Moreover, the performance improvements of Multigraph over DiffMG validate that a meta multigraph surpasses a meta graph in the representation learning on HINs. Note that the meta multigraph strategy can not address the issue of instability. So the performance of Multigraph is worse than PMMM in most other search seeds.

Comparison on Recommendation

Table 3 reports the AUC results on the recommendation task of our methods and the baselines. Both our Multigraph and PMMM surpass the state-of-the-art models on all the three datasets, demonstrating that a meta multigraph surpasses a meta-path or meta graph in capturing the semantic information on HINs. PMMM also outperforms HGT (Hu et al. 2020), a far more complex architecture. The results indicate the superiority of the NAS method in HINs.

Efficiency

Table 4 compares the search cost of our method against existing works employing NAS for heterogeneous GNNs on three recommendation datasets. HGNAS is omitted as it

does not provide the search cost and source code. GEMS is time-consuming as it evaluates each candidate model individually. For DiffMG and PMMM, we present the overall search cost of three runs. Both methods combine candidate models into a super-net. So the search cost can be reduced by about 1000 times, enabling them to be applied to large-scale HINs. Compared with DiffMG, our method has a similar search efficiency but exhibits better performance and much-improved stability. The evaluation efficiency is illustrated in Appendix, which has a similar performance to search efficiency.

Visualization

Figure 3 visualizes the architectures searched on Amazon by PMMM. We show both the meta multigraph (left) whose target node type is user (U) and the meta multigraph (right)
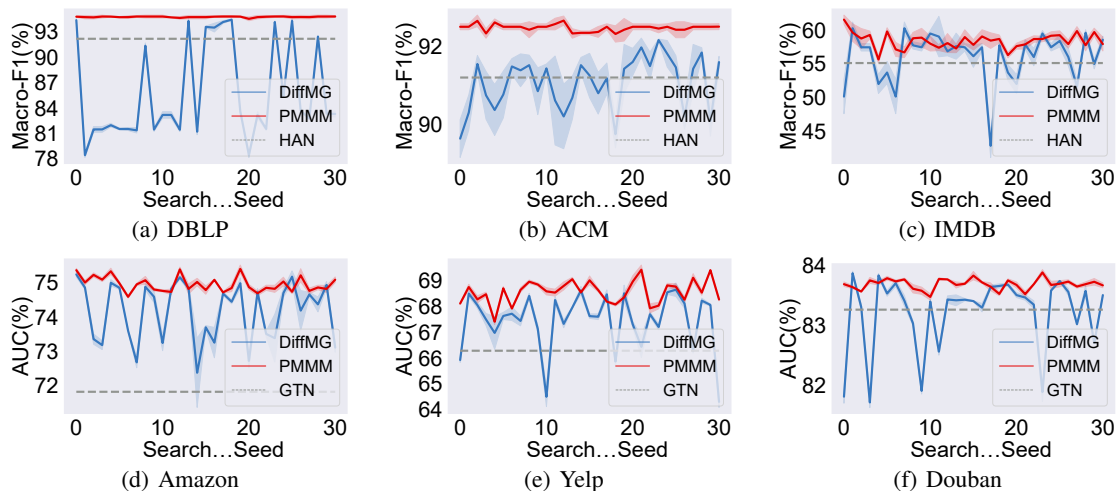
Figure 4: Performance on different search seeds (best viewed in color). The first three figures are results for node classification task, followed by three figures for recommendation task.

whose target node type is item (I). The searched architectures by PMMM are more complex than those of DiffMG. However, more message passing steps have little negative impact on efficiency due to parallelization in the training of neural architecture. In the left meta multigraph, $H^{(2)}$ has two incoming edge types. We found that both A_{IV} and A_{IB} are important. If we search for a meta graph like DiffMG, A_{IV} has to be discarded as multiple paths are not allowed, which will seriously impact the performance.

Stability Study

To evaluate the stability of our method, we compare PMMM with differentiable meta graph search by using different random search seeds. We run the two algorithms on random search seeds from 0 to 30, and plot the Macro-F1 and AUC scores averaged from 3 dependent retraining of the searched architecture under different random training seeds. The results are illustrated in Figure 4. The gray dotted line shows the results of hand-designed heterogeneous GNNs, HAN on node classification and GTN on recommendation as the baselines. Although DiffMG shows excellent performance in a few search seeds, the performance dramatically declines in most other seeds. In most cases, its performance is even worse than HAN and GTN. In contrast, PMMM can overcome the instability issue in DiffMG. Besides, PMMM significantly outperforms DiffMG in most search seeds and consistently surpasses the manual designed networks. Specifically, considering the average results on search seeds from 0 to 30, PMMM has 8.97%, 8.33%, 1.53%, 1.78%, 2.67% and 3.11% better performance compared to DiffMG.

Parameter Study

We carry out analysis on hyper-parameters λ controlling derived meta multigraphs of our model. When $\lambda = 1$, only the strongest path is retained in each edge, which is the deriving strategy of most differentiable NAS, including DiffMG.
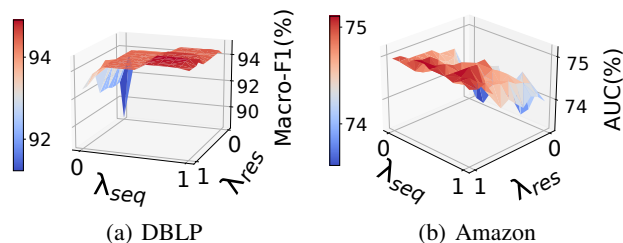

Figure 5: Performances of PMMM with varying λ.

When $\lambda = 0$, all paths are retained. We speculate that λ should be close to 1 to ensure that all effective paths are retained and weak paths are dropped. We divide λ into two parts, λ_{seq} and λ_{res}, which controls the number of retrained paths in sequential multi-edges and in residual multi-edges, respectively. We vary λ_{seq} and λ_{res} from 0 to 1 and plot the results of DBLP for node classification and Amazon for recommendation in Figure 5. The performance reaches the peak when λ_{seq} and λ_{res} are around 0.9, which demonstrates the effectiveness of meta multigraph strategy and verifies our speculation. Notably, PMMM can derive flexible meta multigraphs by varying λ_{seq} and λ_{res}.

Conclusion

In this work, we propose a new concept of meta multigraph and present critical contributions to both the first stable search algorithm and flexible deriving strategy for differentiable neural architecture search on HINs. Extensive experiments demonstrate that our method is stable, flexible, and consistently surpassing state-of-the-art heterogeneous GNNs on six datasets of two representative tasks on HINs. In future work, we will try to explore the interpretability of meta multigraphs theoretically.

Acknowledgments

References

Anandalingam, G.; and Friesz, T. L. 1992. Hierarchical optimization: An introduction. *Annals of Operations Research*, 34(1): 1–11.

Cai, H.; Zhu, L.; and Han, S. 2019. ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Colson, B.; Marcotte, P.; and Savard, G. 2007. An overview of bilevel optimization. *Annals of operations research*, 153(1): 235–256.

Ding, Y.; Yao, Q.; Zhao, H.; and Zhang, T. 2021. DiffMG: Differentiable Meta Graph Search for Heterogeneous Graph Neural Networks. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18*, 279–288.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *KDD '17: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, 135–144.

Fu, X.; Zhang, J.; Meng, Z.; and King, I. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, 2331–2341.

Gao, Y.; Zhang, P.; Li, Z.; Zhou, C.; Liu, Y.; and Hu, Y. 2021. Heterogeneous Graph Neural Architecture Search. In *2021 IEEE International Conference on Data Mining, ICDM*, 1066–1071.

Guo, Z.; Zhang, X.; Mu, H.; Heng, W.; Liu, Z.; Wei, Y.; and Sun, J. 2020. Single path one-shot neural architecture search with uniform sampling. In *European Conference on Computer Vision, ECCV*, 544–560.

Han, Z.; Xu, F.; Shi, J.; Shang, Y.; Ma, H.; Hui, P.; and Li, Y. 2020. Genetic Meta-Structure Search for Recommendation on Heterogeneous Information Network. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, 455–464.

Hu, Z.; Dong, Y.; Wang, K.; and Sun, Y. 2020. Heterogeneous Graph Transformer. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, 2704–2710.

Jin, J.; Qin, J.; Fang, Y.; Du, K.; Zhang, W.; Yu, Y.; Zhang, Z.; and Smola, A. J. 2020. An Efficient Neighborhood-based Interaction Model for Recommendation on Heterogeneous Graph. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, 75–84.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Li, Y.; Jin, Y.; Song, G.; Zhu, Z.; Shi, C.; and Wang, Y. 2021. GraphMSE: Efficient Meta-path Selection in Semantically Aligned Feature Space for Graph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 5, 4206–4214.

Liu, H.; Simonyan, K.; and Yang, Y. 2019. DARTS: Architecture Search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Qin, Y.; Wang, X.; Zhang, Z.; and Zhu, W. 2021. Graph differentiable architecture search with structure learning. *Advances in Neural Information Processing Systems*, 34: 16860–16872.

Schlichtkrull, M. S.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC'2018*.

Sun, Y.; Han, J.; Yan, X.; Yu, P. S.; and Wu, T. 2011a. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*, 4(11): 992–1003.

Sun, Y.; Han, J.; Yan, X.; Yu, P. S.; and Wu, T. 2011b. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *Proceedings of the Vldb Endowment*, 4(11): 992–1003.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019a. Heterogeneous Graph Attention Network. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 2022–2032.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019b. Heterogeneous Graph Attention Network. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 2022–2032.

Wei, L.; Zhao, H.; and He, Z. 2022. Designing the Topology of Graph Neural Networks: A Novel Feature Fusion Perspective. In *Proceedings of the ACM Web Conference 2022*, 1381–1391.

Xie, L.; and Yuille, A. L. 2017. Genetic CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 1388–1397.

Xu, Y.; Xie, L.; Zhang, X.; Chen, X.; Qi, G.; Tian, Q.; and Xiong, H. 2020. PC-DARTS: Partial Channel Connections for Memory-Efficient Architecture Search. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Xue, C.; Wang, X.; Yan, J.; Hu, Y.; Yang, X.; and Sun, K. 2021. Rethinking Bi-Level Optimization in Neural Architecture Search: A Gibbs Sampling Perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 12, 10551–10559.

Yang, C.; Xiao, Y.; Zhang, Y.; Sun, Y.; and Han, J. 2020. Heterogeneous Network Representation Learning: A Unified Framework with Survey and Benchmark. *IEEE Transactions on Knowledge and Data Engineering*.

Yao, Q.; Xu, J.; Tu, W.-W.; and Zhu, Z. 2020. Efficient Neural Architecture Search via Proximal Iterations. In *AAAI*.

Yun, S.; Jeong, M.; Kim, R.; Kang, J.; and Kim, H. J. 2019a. Graph Transformer Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 11960–11970.

Yun, S.; Jeong, M.; Kim, R.; Kang, J.; and Kim, H. J. 2019b. Graph Transformer Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 11960–11970.

Zhang, C.; Song, D.; Huang, C.; Swami, A.; and Chawla, N. V. 2019a. Heterogeneous Graph Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 793–803.

Zhang, C.; Song, D.; Huang, C.; Swami, A.; and Chawla, N. V. 2019b. Heterogeneous Graph Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 793–803.

Zhao, H.; Yao, Q.; and Tu, W. 2021. Search to aggregate neighborhood for graph neural network. In *ICDE*.

Zhou, K.; Song, Q.; Huang, X.; and Hu, X. 2019. Auto-GNN: Neural architecture search of graph neural networks. *arXiv preprint arXiv:1909.03184*.

Zoph, B.; and Le, Q. V. 2017. Neural Architecture Search with Reinforcement Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.