2022 IEEE International Conference on Data Mining (ICDM)

Multi-Relational Graph Neural Architecture Search with Fine-grained Message Passing

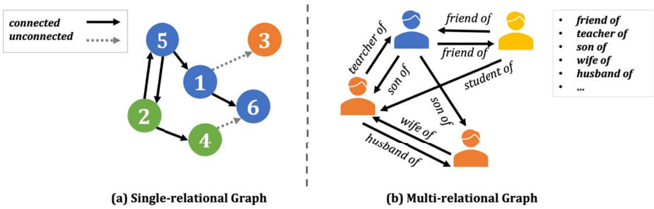
Xin Zheng[†], Miao Zhang[‡], Chunyang Chen[†], Chaojie Li[§], Chuan Zhou[††], Shirui Pan[¶*]
[†]Department of Software Systems and Cybersecurity, Monash University, Australia
[‡]Department of Computer Science, Aalborg University, Denmark
[§]School of Electrical Engineering and Telecommunications, University of New South Wales, Australia
[††]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
[¶]School of Information and Communication Technology, Griffith University, Australia
xin.zheng@monash.edu, miaoz@cs.aau.dk, chunyang.chen@monash.edu,
cjlee.cqu@163.com, zhouchuan@amss.ac.cn, s.pan@griffith.edu.au

***Abstract*—Graph neural architecture search (NAS) has gained great popularity in automatically designing powerful graph neural networks (GNNs) with superior learning abilities, significantly relieving human effort and expertise reliance. Despite the advanced performance of automated learning, existing graph NAS models mainly work on single-relational graphs, while the widespread multi-relational graphs in real-world applications, are not well addressed. Moreover, current search spaces of automated GNNs are generally coarse-grained by simply integrating typical GNN layers and hyper-parameters, resulting in severe limitations on search capacities and scopes for creating innovative GNN architectures. To tackle the limitations of single-relational setting and coarse-grained search space design in existing graph NAS, in this paper, we propose a novel framework of multi-relational graph neural architecture search, dubbed MR-GNAS, to automatically develop innovative and excellent multi-relational GNN architectures. Specifically, to enlarge search capacities and improve search flexibility, MR-GNAS contains a fine-grained search space that embraces the full-pipe multi-relational message passing schema, enabling expressive architecture search scopes. With the well-designed fine-grained search space, MR-GNAS constructs a relation-aware supernet with a tree topology, to jointly learn discriminative node and relation representations. By searching with a gradient-based strategy in the supernet, the proposed MR-GNAS could derive excellent multi-relational GNN architectures in multi-relational graph analysis. Extensive experiments on entity classification and link prediction tasks over multi-relational graphs illustrate the effectiveness and superiority of the proposed method.**

***Index Terms*—graph neural networks, automated graph learning, graph neural architecture search, multi-relational graphs, fine-grained message passing**

I. Introduction

Graphs are pervasive structured data and have been widely used in many real-world scenarios, such as social networks [1], knowledge bases [2], and recommendation systems [3]. Recently, graph neural networks (GNNs) have become prevalent models with excellent learning abilities for analyzing various graph-structure data [4]–[13]. Despite the remarkable success, existing GNN models are usually designed manually by experts for various network architectures on different graphs. Under the variety and complexity of tasks and structures on graphs, such manual designs would cost much human effort with heavy reliance on expert knowledge. Moreover, the design scopes of model architectures would be limited by human understanding, resulting in limited performance on graph learning. Hence, it is necessary to build tailored GNN architectures driven by specific graph tasks and data to relieve human effort.



(a) Single-relational Graph

(b) Multi-relational Graph

Fig. 1. Difference between single-relational graphs and multi-relational graphs.

In light of this, a line of automated machine learning research, *i.e.*, neural architecture search (NAS) [14]–[19], is introduced into GNN development for automatically discovering and creating excellent network architectures. With well-designed search spaces and well-customized search strategies, graph NAS methods have achieved promising progress in automated graph learning [20]–[25], yielding superior GNN architectures for specific tasks and data than hand-crafted models. However, there are two critical challenges remaining with existing graph NAS methods, from the perspectives of graph data modelling and search space design, respectively.

Concretely, current automated GNNs mainly serve on the single-relational graph setting [20], [21], where there exists at most one edge between arbitrary nodes and each edge indicates a binary relation, *i.e.*, connected and unconnected, as shown in Fig. 1(a). Such a setting significantly limits the applications of existing automated GNNs, while real-world graphs generally contain multiple relation types, *e.g.*, knowledge graphs [26], the typical multi-relational graphs with different relation types and edge directions, as shown in Fig. 1(b). Furthermore, existing graph NAS models merely focus on the node representation learning, while the relation representation learning is not well explored. On multi-relational graphs, node and

[*]Corresponding author.

2374-8486/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00089

relation representations are required to be learned jointly. From the perspective of search space design, current search spaces of automated GNNs generally integrate typical GNN layers and hyper-parameters as candidate operations. To some extent, such a coarse-grained search space design could be regarded as a modular ensemble of existing GNN layers, and the principle GNN architectures are not changed and innovated. This would significantly limit the capacity and scope of new GNN architecture development, leading to the damaged flexibility and reasoning ability of automated GNNs.

To tackle the two challenges mentioned above, we propose a novel framework of multi-relational graph neural architecture search, named MR-GNAS, to automatically develop powerful multi-relational GNN models with outstanding abilities in analyzing multi-relational graphs. Specifically, to enlarge the search capacities and improve the search flexibility, we propose a fine-grained GNN search space with functional candidate operations to integrate the critical components of the multi-relational message passing schema. In this way, the proposed MR-GNAS could build innovative multi-relational GNN architectures with excellent learning abilities. To jointly learn with the diverse entity and relation types, MR-GNAS constructs a relation-aware supernet with a tree topology, in which the informative messages would be effectively learned, resulting in discriminative node and relation representations for better analyzing multi-relational graphs.

Concretely, based on the fine-grained search space, the relation-aware supernet is composed of four sequentially stacked cells, embracing the overall pipeline of multi-relational message passing. Given a multi-relational graph, it first inputs (1) an entity-relation integration cell, so that the semantics and contexts of nodes and edges would be well captured and integrated according to their interactions, leading to informative messages of diverse entity and relation types. Then, the obtained informative messages flow to (2) a relation-aware message filtering cell to further learn effective messages with sparse and dense gating mechanisms, so that the most relevant information of relation-aware message passing could be preserved. Further, the filtered messages would be aggregated with (3) a neighbor aggregation cell according to the structure information on multi-relational graphs, leading to informative node representations. Finally, (4) an entity-aware embedding filtering cell outputs the ultimate node representations with entity-level sparse and dense feature selection operations. For relation representation learning, it is not only incorporated into the first entity-relation integration cell, but also implements a linear combination and mapping for yielding the ultimate relation embeddings.

Benefiting from the relation-aware supernet with full-pipe multi-relational message passing, the proposed MR-GNAS could incorporate different relation types and edge directions with joint node and relation representation learning. And the functional candidate operations in the fine-grained search space further encourage adequate exploration of multi-relational graph structures and contexts. Through a gradient-based search strategy, MR-GNAS can automatically derive expressive multi-relational GNN models driven by various multi-relational graph data and tasks, leading to wide architecture design scopes, flexible model architectures, and outstanding multi-relational graph analysis abilities.

In summary, the contributions of this work are as follows:

- To overcome the limitation of single-relational setting in graph NAS, we propose a novel framework of multi-relational graph neural architecture search, dubbed MR-GNAS, enabling innovative multi-relational GNN architecture design with excellent learning abilities.
- To enlarge the search capacities and improve the search flexibility, we design a fine-grained search space with functional candidate operations to embrace the multi-relational message passing schema, leading to outstanding architecture search abilities.
- To jointly learn with diverse entity and relation types, we construct a relation-aware supernet with four sequentially stacked cells in the tree topology, resulting in discriminative node and relation representations for benefiting multi-relational graph analysis.
- Through a gradient-based search strategy, the proposed MR-GNAS could derive expressive multi-relational GNN architectures and extensive experiments on entity classification and link prediction tasks illustrate its superiority.

II. Preliminary

General Message Passing Schema. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected, unweighted graph where $\mathcal{V} = \{v_1, \cdots, v_{|\mathcal{V}|}\}$ is the node set and $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the edge set. The neighbor set of node v is $\mathcal{N}(v) = \{u : (v, u) \in \mathcal{E}\}$, and initial node features are represented by $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d_0}$ with d_0-dimension features. For the uniform message passing scheme of general GNNs, node representations are learned by first aggregating the messages from local neighbors, and then combining the aggregated messages with ego-node representations [27] for update. This process is denoted as:

$$\begin{aligned} \mathbf{m}_v^{(l)} &= AGG^{(l)}(\{\mathbf{h}_u^{(l-1)} : u \in \mathcal{N}(v)\}), \\ \mathbf{h}_v^{(l)} &= UPDATE^{(l)}(\mathbf{h}_v^{(l-1)}, \mathbf{m}_v^{(l)}), \end{aligned} \tag{1}$$

where $\mathbf{m}_v^{(l)}$ and $\mathbf{h}_v^{(l)}$ are the message vector and the representation vector of node v at the l-th layer, respectively. $AGG(\cdot)$ and $UPDATE(\cdot)$ are the aggregation function and the update function, respectively.

Based on the general message passing schema, existing automated GNNs on single-relational graphs usually design their search spaces on $AGG(\cdot)$ and $UPDATE(\cdot)$ as $\mathcal{O}_{AGG}$ and $\mathcal{O}_{UPD}$, respectively. Concretely, $\mathcal{O}_{AGG}$ usually contains typical GNN layers as candidate operations, *i.e.*, GCN [28], GAT [29], and GraphSAGE [1]. And $\mathcal{O}_{UPD}$ generally contains combination and update functions to integrate neighbor information and central node information, *i.e.*, Multi-Layer Perceptron (MLP) and concatenation operations [21], [22]. Moreover, existing search spaces might contain intra-layer operations, *i.e.*, skip connection, to benefit multi-layer message passing. Furthermore, some of current automated GNNs also involve certain hyper-parameters in their search spaces, *i.e.*, activation functions and the number of multi-head attentions [20], [30].

Multi-relational Message Passing Schema. Given a multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$ with the set of nodes $\mathcal{V}$, relations $\mathcal{R}$, and edges $\mathcal{E}$, each edge (u, r, v) represents that the relation r exists from source node u to target node v for $\forall u, v \in \mathcal{V}$ and $r \in \mathcal{R}$. Typically, RGCN [7] first proposed multi-relational message passing schema by incorporating edge-specific weights into $AGG(\cdot)$ as:

$$\mathbf{m}_v^{(l)} = AGG^{(l)}(\{\mathbf{W}_r^{(l)}\mathbf{h}_u^{(l-1)} : u \in \mathcal{N}_r(v)\}). \tag{2}$$

To further address relation representation learning, CompGCN [6] integrated the relation embedding $\mathbf{z}_r \in \mathbb{R}^d$ into the multi-relational message passing with $\phi(\cdot)$ operator for combining the entity-relation representations as:

$$\mathbf{m}_v^{(l)} = AGG^{(l)}(\{\mathbf{W}_{\lambda(r)}^{(l)}\phi(\mathbf{h}_u^{(l-1)}, \mathbf{z}_r^{(l-1)}) : u \in \mathcal{N}_{\lambda(r)}(v)\}), \tag{3}$$

Note that $\mathcal{N}_{\lambda(r)}(v)$ indicates the neighbor set with different relation types and edge directions as $\lambda(r) = \left\{r \in \mathcal{R} \cup r^{-1} \in \mathcal{R}^{-1} \cup r_l = \top\right\}$, where $r_l = \top$ and $r^{-1} \in \mathcal{R}^{-1}$ denote the self-loop connection with $(u, \top, u)$ and the inverse relation with (v, r^{-1}, u), respectively. To enlarge the search capacities for flexible automated GNN construction, in this work, we build upon the multi-relational message passing schema by designing a fine-grained search space, with $\phi(\cdot)$ operator, $AGG(\mathbf{W}_{\lambda(r)})$, and $UPDATE^{(l)}(\cdot)$ in different component cells, composing of a relation-aware supernet with a tree topology.

III. The Proposed Method

Through the lens of multi-relational message passing schema, we propose a novel framework of multi-relational graph neural architecture search, named MR-GNAS. Specifically, MR-GNAS contains a fine-grained search space with functional candidate operations. Based on the well-designed search space, MR-GNAS constructs a relation-aware supernet with four sequentially stacked cells in the tree topology, and the overall pipeline is illustrated in Fig. 2. Importantly, such multi-relational message passing schema ensures the proposed MR-GNAS to better incorporate multi-relational graph structures into the learning process of tree-topology supernet. Based on this, the delicately tailored search space with adequate fine-grained candidates make MR-GNAS enjoy the advantages of different functional operations, and then, the expressive architectures would be selected to automatically build powerful and innovative MR-GNN models.

Specifically, given a multi-relational graph, MR-GNAS first takes the d-dimensional entity embeddings $\mathbf{h}_u, \mathbf{h}_v \in \mathbb{R}^d$ and relation embedding $\mathbf{z}_r \in \mathbb{R}^d$ as the initial inputs. Then, the entity and relation inputs would experience four-component cells sequentially : C$_1$: entity-relation integration cell, C$_2$: relation-aware message filtering cell, C$_3$: neighbor aggregation, and C$_4$: entity-aware embedding filtering cell.

- C$_1$: Entity-relation Integration Cell: capturing and integrating the semantics and contexts of nodes and edges according to their interactions, leading to informative messages of diverse entity and relation types.
- C$_2$: Relation-aware Message Filtering Cell: learning effective messages with sparse and dense gating mechanisms, preserving the most relevant information of relation-aware message passing.
- C$_3$: Neighbor Aggregation Cell: merging filtered messages according to multi-relational graph structures, leading to informative node representations.
- C$_4$: Entity-aware Embedding Filtering Cell: conducting entity-level sparse and dense feature selections for obtaining the ultimate node representations.

In the meantime, the relation embeddings are further updated with linear combination and mapping to generate representative relation features. Hence, the proposed MR-GNAS encases an overall pipe of multi-relational message passing with fine-grained functional operations, enabling it to jointly learn with diverse entity and relation types. In this way, discriminative node and relation representations could be generated for benefiting multi-relational graph analysis. More details of the proposed search space for each cell are as follows.

A. Search Space

C$_1$: Entity-relation Integration Cell. To capture the interactions between entity nodes and relation-type specific edges, we introduce three types of entity-relation integration operations to capture and compose the semantics and contexts of entities and relations, leading to informative and beneficial messages.

Specifically, we extend the $\phi(\cdot)$ operator in Eq. (3) to the set of integration operator with following candidate operations: (1) $\phi_+(\mathbf{h}_u, \mathbf{z}_r) = \mathbf{h}_u + \mathbf{z}_r$; (2) $\phi_-(\mathbf{h}_u, \mathbf{z}_r) = \mathbf{h}_u - \mathbf{z}_r$; (3) $\phi_*(\mathbf{h}_u, \mathbf{z}_r) = \mathbf{h}_u * \mathbf{z}_r$. Then, we can obtain the integrated entity-relation message $\mathbf{m}_e^{c_1} \in \mathbb{R}^d$ for each edge $e \in \mathcal{E}$ as:

$$\mathbf{m}_e^{c_1} = \Phi_{op}(\mathbf{h}_u, \mathbf{z}_r), \tag{4}$$

where $\Phi_{op} = \{\phi_+, \phi_-, \phi_*\}$ is the candidate set of the composition operators.

C$_2$: Relation-aware Message Filtering Cell. The key challenge of message passing on multi-relational graphs is to preserve the most relevant messages and filter the redundant information for specific tasks. In light of this, we develop a set of message gating mechanisms inspired by [31] in the message filtering cell. Different from [31] merely considering the node representation filtering with node-level messages, we introduce relation-aware message filtering, which incorporates diverse relation types and edge directions. Hence, the proposed MR-GNAS could learn informative entity and relation interaction messages, leading to better adaption and learning ability for multi-relational graph analysis.

Taking the relation-aware message matrix $\mathbf{M}^{c_1} = [\mathbf{m}_1^{c_1}, \cdots, \mathbf{m}_e^{c_1}] \in \mathbb{R}^{|\mathcal{E}| \times d}, e \in \mathcal{E}$ from C$_1$ as the input, C$_2$ considers three types of filters with five candidate operations, by extending $AGG(\mathbf{W}_{\lambda(r)})$ in Eq. (3) to Γ_{op} searching set, encouraging effective multi-relational message passing via

$$\mathbf{M}^{c_2} = \Gamma_{op}(\mathbf{M}^{c_1}), \tag{5}$$

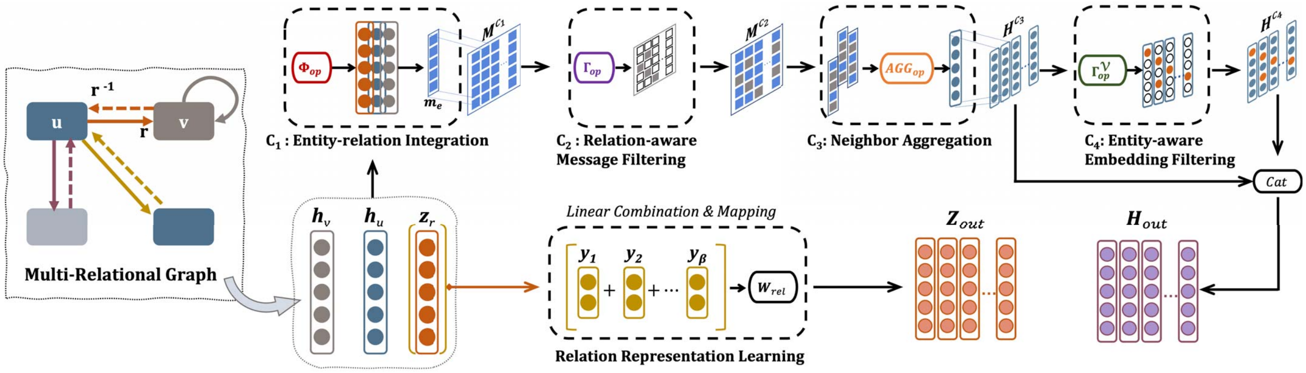

r -1
u
v
r
Multi-Relational Graph
Φ_{op}
m_e
M^{C_1}
C₁ : Entity-relation Integration
Γ_{op}
C₂ : Relation-aware Message Filtering
M^{C_2}
AGG_{op}
C₃: Neighbor Aggregation
H^{C_3}
$\Gamma_{op}^{\mathcal{V}}$
C₄: Entity-aware Embedding Filtering
H^{C_4}
Cat
h_v
h_u
z_r
Linear Combination & Mapping
y_1
y_2
y_β
W_{rel}
Relation Representation Learning
Z_{out}
H_{out}

Fig. 2. The overall framework of multi-relational graph neural architecture search with fine-grained message passing.

where $\Gamma_{op} = \left\{\mathcal{F}_\lambda,\, \mathcal{F}^s,\, \mathcal{F}^s_\lambda,\, \mathcal{F}^d,\, \mathcal{F}^d_\lambda\right\}$, *i.e.*, vanilla filter $\mathcal{F}_\lambda$, sparse filters $\mathcal{F}^s$ and $\mathcal{F}^s_\lambda$, as well as dense filters $\mathcal{F}^d$ and $\mathcal{F}^d_\lambda$. Note that $\mathcal{F}_\lambda$ could be taken as the basic $AGG(\mathbf{W}_{\lambda(r)})$ in Eq. (3). Moreover, we include the identity operation $\mathcal{I}(\mathbf{M}^{c_1}) = \mathbf{M}^{c_1}$ in this cell, which can be taken as the skip-connection to preserve the original beneficial messages. Different types of message filters learn and select specific messages at different scales of relation types and edge directions, leading to the full exploration of data-driven and task-driven messages of entity-relation interactions.

We summarize the main differences of different relation-aware message filters in Γ_{op} in Table I, from the perspectives of three essential components, *i.e.*, mapping functions, weight matrices, and operators. For convenience, we simplify the relation-type set $\lambda(r)$ as λ and more details of each filter can be found as follows.

(1) Vanilla Filter $\mathcal{F}_\lambda$ typically weights the beneficial messages according to the directions of edges and the types of relations. In detail, it learns the re-scaling weight parameter matrix $\mathbf{W}_\lambda = [\mathbf{W}_r; \mathbf{W}_{r^{-1}}; \mathbf{W}_\top] \in \mathbb{R}^{3\times d\times d'}$, where each of $\mathbf{W}_r$, $\mathbf{W}_{r^{-1}}$, and $\mathbf{W}_\top$ learns a $\mathbb{R}^d \to \mathbb{R}^{d'}$ mapping with the original relations r, inverse relations r^{-1}, and self-loop relations $\top$, respectively. Hence, the vanilla filter is calculated by

$$\begin{aligned}\mathcal{F}_\lambda\left(\mathbf{M}^{c_1}\right) &= \mathbf{W}_{\lambda(r)} \cdot \mathbf{M}^{c_1} \\ &= [\mathbf{M}^{c_1}_r\mathbf{W}_r;\, \mathbf{M}^{c_1}_{r^{-1}}\mathbf{W}_{r^{-1}};\, \mathbf{M}^{c_1}_\top\mathbf{W}_\top],\end{aligned} \tag{6}$$

where $\mathbf{M}^{c_1}_\lambda$ denotes corresponding messages with specific relation types, *e.g.*, $\mathbf{M}^{c_1}_r \in \mathbb{R}^{|\mathcal{E}_r|\times d}$, and $|\mathcal{E}_r|$ denotes the number of edges with the original relations r in $\mathcal{G}$.

(2) Sparse Filters $\mathcal{F}^s$ and $\mathcal{F}^s_\lambda$ select and re-scale the beneficial messages through the sparse gating mechanism by fully understanding the importance of relation-aware information. Specifically, this mechanism is implemented by the sparse mapping function $\varphi^s(\cdot)$ and its relation-type specific variant $\varphi^s_\lambda(\cdot)$. We first calculate the weight matrix $\mathbf{W}^s \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ with the sparse mapping function $\varphi^s : \mathbb{R}^d \to \mathbb{R}^1$, then the messages can be re-scaled as:

$$\mathbf{W}^s = \operatorname{diag}\left[\varphi^s(\mathbf{M}^{c_1})\right], \quad \mathcal{F}^s(\mathbf{M}^{c_1}) = \mathbf{W}^s \cdot \mathbf{M}^{c_1}, \tag{7}$$

where $\operatorname{diag}[\cdot]$ is adopted for converting the vector to diagonal matrix, and $\varphi^s(\cdot)$ is implemented by a two-layer MLP with sigmoid activation function.

Furthermore, as shown in Table I, $\varphi^s_\lambda(\cdot)$ extends the above mapping with different relation types and edge directions by assembling them together as $\varphi^s_\lambda = [\varphi^s_r;\, \varphi^s_{r^{-1}};\, \varphi^s_\top]$. Hence, the relation-specific sparse filter $\mathcal{F}^s_\lambda$ is natural to be derived following the same logic in Eq. (6) and Eq. (7) as $\mathcal{F}^s_\lambda(\mathbf{M}^{c_1}) = \mathbf{W}^s_\lambda\mathbf{M}^{c_1}$, where

$$\begin{aligned}\mathbf{W}^s_\lambda &= \operatorname{diag}\left[\varphi^s_\lambda(\mathbf{M}^{c_1})\right] \\ &= \operatorname{diag}\left[\varphi^s_r(\mathbf{M}^{c_1}_r);\, \varphi^s_{r^{-1}}(\mathbf{M}^{c_1}_{r^{-1}});\, \varphi^s_\top(\mathbf{M}^{c_1}_\top)\right].\end{aligned} \tag{8}$$

(3) Dense Filters $\mathcal{F}^d$ and $\mathcal{F}^d_\lambda$ consider a more fine-grained message gating scheme to control information flows and exploit the importance of relation-aware messages in a dense manner. Specifically, the dense mapping function $\varphi^d(\cdot)$ and its relation-type specific variant $\varphi^d_\lambda(\cdot)$ learn the $\mathbb{R}^d \to \mathbb{R}^{d'}$ mapping, yielding dense message re-scaling weights $\mathbf{W}^d$ and $\mathbf{W}^d_\lambda$ with the dimension of $\mathbb{R}^{|\mathcal{E}|\times d'}$. Then, we have:

$$\mathbf{W}^d = \varphi^d\left(\mathbf{M}^{c_1}\right), \quad \mathcal{F}^d(\mathbf{M}^{c_1}) = \mathbf{W}^d \odot \mathbf{M}^{c_1}, \tag{9}$$

where $\odot$ denotes the Hardmard product and $\varphi^d(\cdot)$ is a single-layer MLP with sigmoid activation function. Similarly, the relation-type specific dense filter can be obtained by $\mathcal{F}^d_\lambda(\mathbf{M}^{c_1}) = \mathbf{W}^d_\lambda \odot \mathbf{M}^{c_1}$, where

$$\mathbf{W}^d_\lambda = \varphi^d_\lambda\left(\mathbf{M}^{c_1}\right) = \left[\varphi^d_r(\mathbf{M}^{c_1}_r);\, \varphi^d_{r^{-1}}(\mathbf{M}^{c_1}_{r^{-1}});\, \varphi^d_\top(\mathbf{M}^{c_1}_\top)\right]. \tag{10}$$

In summary, sparse filters calculate the weights among edges with different relation types, while dense filters go for weighting each attribute of each edge. The ensemble of dense and sparse filters would ensure comprehensive message learning at different weighting scales, leading to effective information passing.

TABLE I
SUMMARY OF DIFFERENT TYPES OF RELATION-AWARE MESSAGE FILTERS. '·' DENOTES THE MATRIX PRODUCT AND '⊙' IS THE HARDMARD PRODUCT.

Types	Candidates	Mappings	Weights	Operators
Vanilla	$\mathcal{F}_\lambda$	-	$\mathbf{W}_\lambda \in \mathbb{R}^{3\times d\times d'}$	·
Sparse	$\mathcal{F}^s$	φ^s	$\mathbf{W}^s \in \mathbb{R}^{\lvert\mathcal{E}\rvert\times\lvert\mathcal{E}\rvert}$	·
	$\mathcal{F}^s_\lambda$	$\varphi^s_\lambda = [\varphi^s_r; \varphi^s_{r^{-1}}; \varphi^s_\top]$	$\mathbf{W}^s_\lambda \in \mathbb{R}^{\lvert\mathcal{E}\rvert\times\lvert\mathcal{E}\rvert}$	·
Dense	$\mathcal{F}^d$	φ^d	$\mathbf{W}^d \in \mathbb{R}^{\lvert\mathcal{E}\rvert\times d'}$	⊙
	$\mathcal{F}^d_\lambda$	$\varphi^d_\lambda = [\varphi^d_r; \varphi^d_{r^{-1}}; \varphi^d_\top]$	$\mathbf{W}^d_\lambda \in \mathbb{R}^{\lvert\mathcal{E}\rvert\times d'}$	⊙

TABLE II

ARCHITECTURE COMPARISON OF THE PROPOSED MR-GNAS *vs.*EXISTING MULTI-RELATIONAL GNNS.

Models	C_1 Entity-relation Integration	C_2 Relation-aware Message Filtering	C_3 Neighbor Aggregation	C_4 Entity-aware Embedding Filtering	#Leaf Nodes of DAG
RGCN [7]	$\times$	$\approx \mathcal{F}_\lambda$	AGG_{mean}	$\times$	$[0, 1, 1, 0]$
SACN [32]	$\times$	$\approx \mathcal{F}_\lambda$	AGG_{sum}	$\times$	$[0, 1, 1, 0]$
CompGCN [6]	$\approx \Phi_{op}$	$\mathcal{F}_\lambda$	AGG_{sum}	$\times$	$[1, 1, 1, 0]$
MR-GNAS (ours)	$\Phi_{op} = \{\phi_+, \phi_-, \phi_*\}$	$\Gamma_{op} = \left\{\mathcal{F}_\lambda,\, \mathcal{F}^s,\, \mathcal{F}^s_\lambda,\, \mathcal{F}^d,\, \mathcal{F}^d_\lambda\right\}$	$AGG_{op} = \{sum, mean, max\}$	$\Gamma^{\mathcal{V}}_{op} = \left\{\mathcal{F}^s, \mathcal{F}^d, \mathcal{I}\right\}$	$[N_{c_1}, N_{c_2}, N_{c_3}, N_{c_4}]$

C$_3$: Neighbor Aggregation Cell. Based on the multi-relational graph structures, we consider three types of aggregation functions to generate node embeddings via merging filtered relation-aware messages $\mathbf{M}^{c_2}$ from their neighbors, *i.e.*, $AGG_{op} = \{sum, mean, max\}$. As demonstrated by [27], AGG_{sum} works well for capturing comprehensive structure information, AGG_{mean} considers the statistics of input messages in the aggregating process, while AGG_{max} is robust to noise and performs better on identifying the typical and critical information. Hence, neighbor aggregation cell C$_3$ would yield node embeddings $\mathbf{H}^{c_3} \in \mathbb{R}^{|\mathcal{V}| \times d'}$ as:

$$\mathbf{H}^{c_3} = AGG_{op}(\mathbf{M}^{c_2}). \tag{11}$$

With the ensemble of these complementary aggregation operations, the proposed MR-GNAS could benefit from the advantages of each operation so that the searched models would be more expressive compared with manually designed multi-relational GNN architectures.

C$_4$: Entity-aware Embedding Filtering Cell. We impose sparse and dense filters $\mathcal{F}^s$ and $\mathcal{F}^d$ on node embeddings to further capture discriminative entity representations, corresponding to coarse-grained and fine-grained feature selections, respectively. Note that different from C$_2$ that considers the relation-aware message gating, this cell C$_4$ conducts the entity-aware embedding gating. That means the former focuses on the message level with diverse relation types and edge directions, and the latter concerns the embedding level with various entity types. Thus, the candidate operation set of the entity-aware embedding filtering cell is $\Gamma^{\mathcal{V}}_{op} = \left\{\mathcal{F}^s, \mathcal{F}^d, \mathcal{I}\right\}$ and the filtered node features can be obtained by:

$$\mathbf{H}^{c_4} = \Gamma^{\mathcal{V}}_{op}(\mathbf{H}^{c_3}), \tag{12}$$

where the computations of sparse and dense weights in $\mathcal{F}^s$ and $\mathcal{F}^d$ following the same paradigm in Eq. (7) and Eq. (9) but working on the node embeddings. We introduce the superscript $\mathcal{V}$ to make a clear distinction. At last, we concatenate the outputs of C$_3$ and C$_4$ and impose an MLP for obtaining the ultimate node representations:

$$\mathbf{H}_{out} = MLP([\mathbf{H}^{c_3}; \mathbf{H}^{c_4}]). \tag{13}$$

Relation Representation Learning. Considering the scalability with large-scale relation types and edges, we adopt the linear combination of a set of basis relation vectors following [6]. Then, the relation embeddings can be updated with a single fully-connected mapping with parameters $\mathbf{W}_{rel}$ as follows:

$$\mathbf{z}_r = \sum_{b=1}^{\mathcal{B}} \beta_r \mathbf{y}_b, \quad \mathbf{Z}_{out} = \mathbf{W}_{rel}[\mathbf{z}_r]_{r=1}^{\mathcal{R}}, \tag{14}$$

where $\mathcal{B}$ denotes the number of bases, $\{\mathbf{y}_b\}_{b=1}^{\mathcal{B}}$ is the set of learnable basis vector and $\beta_r \in \mathbb{R}$ is the learnable relation specific weight scalar of a certain basis. Moreover, the relation embeddings are also incorporated into the first entity-relation integration cell, leading to the joint learning of discriminative node and relation representations.

B. Relation-aware Supernet of MR-GNAS

Based on the proposed framework that embraces the overall pipe of multi-relational message passing schema in Fig. 2, we further build a multi-relational GNN supernet within a tree topology by injecting different numbers of nodes into each cell, leading to a relation-aware Directed Acyclic Graph (DAG) shown in the left of Fig. 3.

Specifically, in the supernet, we keep the sequential connection of four cells in the search space and expand the single-stream message passing to N leaf nodes with multiple information streams for each cell in DAG, leading to more comprehensive and adequate informative message capture and flow. Note that different from the entity nodes on multi-relational graphs, each leaf node in the supernet $x^{(i)}$ is the latent representation of the output of each cell. For example, in the first cell of entity-relation integration, $x^{(i)}_{c_1}$ denote the message matrix $\mathbf{M}^{c_1}$; While in the cell of entity-aware embedding filtering, $x^{(i)}_{c_4}$ denote the entity representation matrix $\mathbf{M}^{c_4}$ of all nodes on multi-relational graphs. In the left of Fig. 3, we take $N_{c_1}\{x^{(0)}\} = 1, N_{c_2}\{x^{(1,2,3)}\} = N_{c_3}\{x^{(4,5,6)}\} = N_{c_4}\{x^{(7,8,9)}\} = 3$ for illustration convenience. With the ensemble of complementary N functional candidates in each cell, the proposed relation-aware supernet of MR-GNAS could benefit from the advantages of different operations. This further enables MR-GNAS to incorporate multiple information streams in each cell and further evaluate the strengths of different candidate operations. At the end of the search stage, MR-GNAS selects appropriate candidates cell-by-cell and builds the ultimate multi-relational GNN architectures driven by specific data and tasks, as shown in the right of Fig. 3. In this way, the searched MR-GNNs would be more expressive and powerful due to the fine-grained candidate selection inside each cell.

To illustrate the effectiveness well-designed search space and the sufficiency of the proposed supernet, we make a detailed comparison between our MR-GNAS supernet and state-of-the-art multi-relational GNNs in Table II. As can be generally observed, existing prevalent human-designed GNN models on multi-relational graphs, *i.e.*, RGCN [7], SACN [32], CompGCN [6], can be approximated as the sub-architectures in the proposed supernet of MR-GNAS. In each cell, our MR-GNAS contains more adequate functional candidate operations to enlarge the search scope of architecture design, along with more flexible leaf node number settings for developing creative and expressive MR-GNN models.

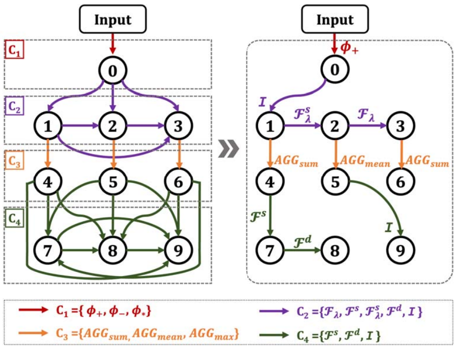

Input
C1
C2
C3
C4
0
1
2
3
4
5
6
7
8
9
Input
ϕ_+
I
$\mathcal{F}^s_\lambda$
$\mathcal{F}_\lambda$
AGG_{sum}
AGG_{mean}
AGG_{sum}
$\mathcal{F}^s$
$\mathcal{F}^d$
I
$C_1 = \{\phi_+, \phi_-, \phi_*\}$
$C_2 = \{\mathcal{F}_\lambda, \mathcal{F}^s, \mathcal{F}^s_\lambda, \mathcal{F}^d, I\}$
$C_3 = \{AGG_{sum}, AGG_{mean}, AGG_{max}\}$
$C_4 = \{\mathcal{F}^s, \mathcal{F}^d, I\}$

Fig. 3. Supernet of MR-GNAS (left). The right shows an example of model architecture after searching.

C. Search Algorithm

Based on the proposed relation-aware supernet of MR-GNAS, we introduce the gradient-based differential search strategy following [14] to relax the categorical choices of candidate operations, resulting in the continuous optimization of the supernet. Concretely, given MR-GNAS supernet $\mathcal{S}(\mathcal{X}, \mathcal{A})$ with leaf node set $\mathcal{X}$ and the edge set $\mathcal{A}$, as well as the overall search space $\mathcal{O} = \{\Phi_{op}, \Gamma_{op}, AGG_{op}, \Gamma^{\mathcal{V}}_{op}\}$ containing multi-relational candidate operations, we first define the mixed operation $\bar{o}^{(i,j)}$ relaxed by softmax function as:

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp\left(\alpha_o^{(i,j)}\right)}{\sum_{o' \in \mathcal{O}} \exp\left(\alpha_{o'}^{(i,j)}\right)} o(x), \tag{15}$$

where $\alpha_o^{(i,j)}$ denotes the operation specific weight vector, *i.e.*, certain weight of one edge in $\mathcal{S}$. The leaf node pair $(x^{(i)}, x^{(j)}) \in \mathcal{X}$ is denoted as (i, j) for brevity.

In this way, the task of architecture search is to learn the network architecture $\boldsymbol{\alpha} \in \mathcal{A}$, *i.e.*, a set of continuous edge variables $\boldsymbol{\alpha} = \left\{\alpha_o^{(i,j)} | o \in \mathcal{O}\right\}$. Then, we jointly learn the architecture $\boldsymbol{\alpha}$ and weights $\boldsymbol{w}$ in the proposed MR-GNAS supernet through the bi-level optimization as

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \mathcal{L}_{val}\left(\boldsymbol{w}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}\right), \\ s.t. \quad & \boldsymbol{w}^*(\boldsymbol{\alpha}) = \operatorname{argmin}_{\boldsymbol{w}} \mathcal{L}_{train}(\boldsymbol{w}, \boldsymbol{\alpha}). \end{aligned} \tag{16}$$

At the upper level, we find $\boldsymbol{\alpha}^*$ by minimizing the validation loss $\mathcal{L}_{val}$ for learning network architectures, while at the lower level, we learn weights $\boldsymbol{w}^*$ by minimizing the training loss $\mathcal{L}_{train}$ for learning network parameters. At the end of the gradient-based search, we construct the corresponding discrete network architecture by selecting the operation with the highest architecture weights as $o^* = \operatorname{argmax}_{o \in \mathcal{O}} \alpha_o^{(i,j)}$.

IV. Experiments

We conduct experiments within two stages following the general NAS pattern [14], *i.e.*, the search stage and the training stage. In the first search stage, we search in the relation-aware supernet of MR-GNAS for an optimal multi-relational GNN model with the bi-level optimization scheme. Then, in the second training stage, we train the searched models from scratch. Finally, we evaluate the proposed MR-GNAS on the tasks of entity classification and link prediction, respectively. The overall experiments are based on the open-source Deep Graph Library (DGL) [33] under the Pytorch framework and run with V100 and Quadro RTX 6000 GPUs.

TABLE III
The statistics of evaluation datasets.

(a) Entity Classification Datasets

Datasets	Entities	Relations	Edges	Labeled	Classes
AIFB	8,285	45	29,043	176	4
MUTAG	23,644	23	74,227	340	2
BGS	333,845	103	916,199	146	2
AM	1,666,764	133	5,988,321	1,000	11

(b) Link Prediction Datasets

Datasets	Entities	Relations	Train	Validate	Test
FB15K-237	15k	237	272k	18k	20k
WN18RR	41k	11	87k	3k	3k

A. Evaluation Tasks and Datasets

Entity Classification is the task of inferring the entity types on multi-relational graphs. We evaluate our model on four datasets: AIFB, MUTAG, BGS, and AM, whose statistics are listed in Table III (a). We adopt the standard test protocol in Resource Description Framework (RDF) format from [34] and take the average classification accuracy (ACC%) as the evaluation metric. As for the baselines, we compare the proposed MR-GNAS with following human-designed models: Feat [35], WL [5], RDF2Vec [36], RGCN [7], SACN [32], and CompGCN [6], as well as automated graph NAS model GNAS [31].

Link Prediction is the task of inferring missing facts based on the known facts in knowledge graphs, where the facts are denoted by the set of triples (subject, relation, object), *i.e.*, (u, r, v). Given a knowledge graph, link prediction aims to find the correct entities to complete the facts of $(u, r, ?)$ or the $(?, r, t)$, corresponding to the tail entity prediction and the head entity prediction, respectively. The proposed MR-GNAS serves as a graph encoder and we adopt 'ConvE' [41] as the score function for simple setting. We conduct experiments on FB15K-237 [44] and WN18RR [41] datasets in Table III (b) with two commonly used evaluation metrics, *i.e.*, Mean Reciprocal Rank (MRR) and Hits@k $(k = 1, 3, 10)$ under the filtered setting following [37]. Specifically, through conducting head and tail prediction on all test triples, the rank q of each target entity against the others would be calculated, leading to the overall test prediction ranks Q. Then, MRR evaluates the average of the inverse of the obtained ranks [45] as $MRR = 1/|Q| \sum_{q \in Q} 1/q$. And Hits@$k$ measures the

TABLE IV
Results of average accuracy (%) for the entity classification task. Best results are in bold, and the second best results are underlined.

Types	Models	AIFB	MUTAG	BGS	AM
Human-designed	Feat [35]	55.55	77.94	72.41	66.66
	WL [5]	80.55	80.88	86.20	87.37
	RDF2Vec [7]	88.88	67.20	87.24	88.33
	RGCN [7]	95.83	73.23	83.10	89.29
	SACN [32]	-	77.90	-	90.20
	CompGCN [6]	-	85.30	-	**90.60**
Graph NAS	GNAS [31]	88.90	85.86	75.90	85.35
	MR-GNAS (ours)	**100.00**	**89.70**	**89.70**	89.90

TABLE V

MRR AND HITS@k RESULTS FOR THE LINK PREDICTION TASK. BEST RESULTS ARE IN BOLD, AND THE SECOND BEST RESULTS ARE UNDERLINED.

Types	Models	FB15K-237				WN18RR			
		MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
Human-designed	TransE [37]	0.294	0.465	-	-	0.226	0.501	-	-
	TransH [38]	0.233	0.401	-	-	0.186	0.451	-	-
	DisMult [39]	0.241	0.419	0.263	0.155	0.430	0.490	0.440	0.390
	ComplEx [40]	0.247	0.428	0.275	0.158	0.440	0.510	0.460	0.410
	ConvE [41]	0.325	0.501	0.356	0.237	0.430	0.520	0.440	0.400
	RGCN [7]	0.248	0.417	-	0.151	-	-	-	-
	ConvKB [42]	0.243	0.421	0.371	0.155	0.249	0.524	0.417	0.057
	VR-GCN [43]	0.248	0.432	0.272	0.159	-	-	-	-
Graph NAS	GNAS [31]	0.273	0.422	0.299	0.197	0.175	0.273	0.199	0.123
	MR-GNAS (ours)	**0.348**	**0.530**	**0.380**	**0.258**	**0.456**	**0.541**	**0.470**	**0.414**

ratio of predictions ranked in top $k = \{1, 2, 3\}$ as $Hit@k = |\{q \in Q : q \leq k\}|/|Q|$. The higher MRR and Hit@k indicate the better link prediction performance. We compare the proposed MR-GNAS with following human-designed methods: TransE [37], TransH [38], DistMult [39], ComplEx [40], ConvE [41], RGCN [7], ConvKB [42], and VR-GCN [43], as well as the automated graph NAS method GNAS [31]. Note that considering GNAS is proposed to serve single-relational graph learning, we reproduce the experimental results of both entity classification and link prediction tasks by adapting it to the multi-relational setting without relation representation learning.

B. Experimental Results.

Entity Classification. The experiment results of the entity classification task are reported in Table IV. In general, our proposed MR-GNAS achieves the best and highly competitive performance of average classification accuracy on all datasets. Specifically, MR-GNAS significantly exceeds the existing models with the performance improvement of 4.4%, 5.2%, and 2.8% on AIFB, BGS, and MUTAG datasets, respectively. Due to large-scale and complex data statistics of AM dataset, automated search in the proposed supernet of MR-GNAS would become more complicated than other datasets. It is more challenging to derive optimal multi-relational GNN models driven by large-scale graphs with the computation memory limitation, causing limited entity classification performance improvement. This point could be further verified by the performance of GNAS, which is under-performed than most existing human-designed models. Even though, our MR-GNAS still achieves better classification results than automated GNAS (89.90% *vs.*85.35%) and most human-designed models. We attribute the superiority of the proposed MR-GNAS to the well-designed and wide-scope search spaces, as well as the multi-relational message passing based supernet construction, enabling its excellent performance to automatically develop task-specific and data-driven MR-GNN architectures.

Link Prediction. The experiment results of the link prediction task is presented in Table V. On the whole, our proposed MR-GNAS achieves outstanding performance on both FB15K-237 dataset and WN18RR dataset. MR-GNAS significantly surpasses the human-designed multi-relational GNN models, *i.e.*, RGCN and VR-GCN, on FB15K-237 dataset with 40.3% performance improvement for MRR from 0.248 to 0.348. Moreover, MR-GNAS still has better link prediction performance when compared with the automated GNAS model. Specifically, MR-GNAS improves MRR on FB15K-237 from 0.273 to 0.348, while on WN18RR dataset, the MRR performance improvement is greater from 0.175 to 0.456. We attribute this to the joint node and relation representation learning of the proposed MR-GNAS in the entity-relation integration cell, as well as the relation linear combination and mapping, since GNAS is unable to deal with the relation representation learning. This further reflects the effectiveness of the delicate search space design with relation-aware message passing of the proposed MR-GNAS. By enriching relation-aware candidate operations in the fine-grained manner, MR-GNAS could automatically derive expressive and powerful MR-GNNs for multi-relational graph learning.

C. Discussion and Analysis

a) Understanding of Searched Models: We present the derived models for each task and dataset by searching in the relation-aware supernet of MR-GNAS in Fig. 4. We set the number of leaf nodes in the supernet as $[1, 3, 3, 3]$ with two layers and $[1, 2, 2, 2]$ with a single layer for entity classification and link prediction, respectively. In general, we can observe that different tasks and datasets have different preferences for the functional candidate operations in each cell. Therefore, resort to the well-designed search space with adequate candidates, the proposed MR-GNAS could enjoy the advantages of different functional operations for multi-relational message passing and automatically select the optimal ones to build creative and expressive MR-GNN architectures.

b) Scalability with Number of Relations: We discuss the effects of different numbers of relation basis vectors on AM dataset for the entity classification task in Fig 5. As can be observed, the performance would not improve along with the increased numbers of bases, and there might be a proper basis number for better performance, *i.e.*, 80 in our searched model. Hence, the number of relation basis vectors could be taken as a hyper-parameter of the automated MR-GNAS tuned for better multi-relational graph learning.

c) Effectiveness of the Search Strategy: To verify the effectiveness of the search strategy on deriving optimal searched models, we randomly select five MR-GNN architectures (Rand_arch_01∼Rand_arch_05) from the proposed search space on FB15K-237 dataset for the link prediction

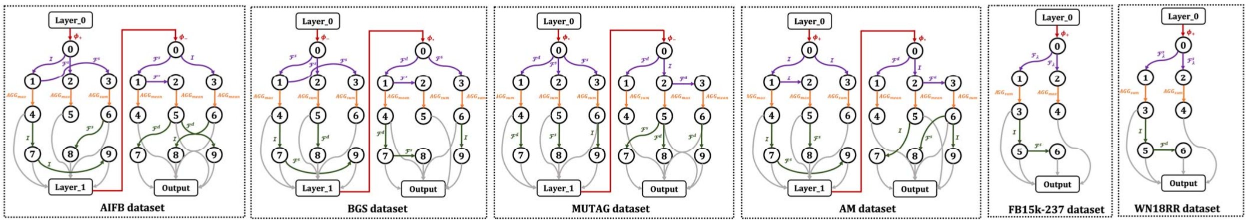


Fig. 4. The searched models from the supernet of MR-GNAS on different datasets for entity classification and link prediction tasks.

TABLE VI
SEARCH TIME OF 10 EPOCHS (CLOCK TIME IN SECONDS) COMPARISON.

Search time (10 epochs)		GNAS [31]	MR-GNAS (ours)
Entity Classification Dataset	AIFB	36.49	36.12
	MUTAG	69.97	69.11
	BGS	26.59	26.89
	AM	317.95	329.42
Link Prediction Dataset	FB15K-237	2.41	3.92
	WN18RR	4.23	5.91

task, and make comparison with the derived MR-GNAS based on the gradient-based differential search strategy. The experimental results are presented in Fig. 6. As can be observed, the proposed MR-GNAS achieves the best performance on link prediction for all metrics of MRR and Hit@k than all other randomly-built architectures, significantly verifying the effectiveness of the search strategy adopted by the proposed method. Moreover, the randomly-built MR-GNN architectures from the proposed search space are all with MRR metrics over 0.270, which still outperform existing multi-relational GNNs, *e.g.*, RGCN with 0.248 MRR in Table V, further illustrating the effectiveness of the proposed search space with multi-relational message passing schema.

d) Search Efficiency: To illustrate the search efficiency of the proposed method, we report the search time costs of 10 epochs between our MR-GNAS and graph NAS model GNAS presented in Table VI. As can be observed, even with enlarged search spaces containing more operations and relation representation learning progress, MR-GNAS does not take much extra search time compared with GNAS, illustrating the outstanding search efficiency of the proposed method.

V. Related Work

Multi-Relational Graph Neural Networks. Multi-relational graphs are widespread graph-structured data in the real world, reflecting complex and diverse relationships among various objects [37], [41], [43], [46]. As a typical category of multi-relational graphs, knowledge graphs (KGs) provide a formal understanding of the world based on the human knowledge [26], [32], [39], [47]. Specifically, KGs denote nodes as entities and edges as relations with labels, indicating the specific triplet facts connecting arbitrary two entities with a certain relation. Knowledge graph representation learning (KRL) is an important research line of KG analysis, where entities and relations would be embedded into low-dimensional latent spaces for capturing their semantic and context information. There are two critical components of KRL, *i.e.*, embedding models and score functions, where the former learns low-dimensional entity and relation representations and the latter measures the plausibility of triplet facts. As a prevalent type of KG embedding models, a multi-relational graph neural network (MR-GNN) utilizes the multi-relational structure connections to benefit GNN learning under message passing schema. MR-GNN models jointly learn node and relation representations of various entity and relation types with different edge directions. Typically, RGCN [7] constructed graph convolutional network (GCN) on relational graphs with relation-specific filters, where basis and block diagonal decomposition were introduced to deal with the scale issue of numerous relation types. SACN [32] developed a weighted GCN as the embedding model to capture node structures and attributes, as well as edge relation types, with adaptive relation-specific scalar weight learning. Despite the satisfying performance, these methods paid more attention to the node representation learning while the relation representation learning was not well addressed. In light of this, CompGCN [6] introduced the entity-relation embedding composition to incorporate relation learning, resulting in a more general relational GNN framework with superior performance.

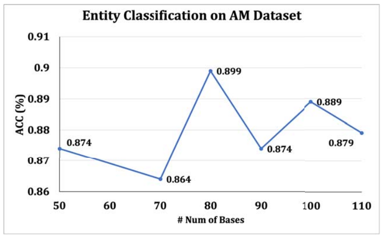


Fig. 5. Effects of different numbers of relation basis vectors on AM dataset for the entity classification task.

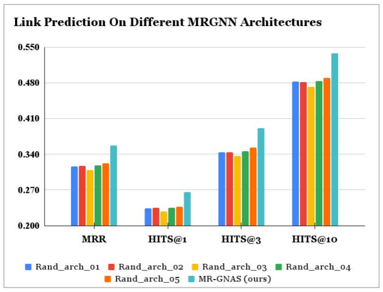


Fig. 6. Comparison between randomly-selected architectures *vs.*gradient-based differential searched architectures on FB15K-237 dataset for link prediction.

Nevertheless, all these relational GNN models are human-designed, leading to heavy reliance on expert knowledge with much human effort cost, since massive manual enumerations need to be conducted for obtaining expressive network architectures. To relieve human effort and enlarge the multi-relational GNN design prospect, we leverage the NAS technique for automated multi-relational GNN architecture design

TABLE VII
The comprehensive comparison between the proposed MR-GNAS and other existing graph NAS models.('Single-rel' and 'Multi-rel' denote the single-relational graphs and multi-relational graphs, respectively. 'Coarse' and 'Fine' indicate the coarse-grained and fine-grained search space characteristics, respectively. 'Dete. Differ.' and 'Stoc. Differ' denote the deterministic differential search strategy and stochastic differential search strategy, respectively.)

Methods	Graph Types	Search Spaces		Search Strategy
		Character	Models	
GraphNAS [20]	Single-rel	Coarse	GNN	RL
AGNN [30]	Single-rel	Coarse	GNN	EA+RL
SANE [21]	Single-rel	Coarse	GNN	Dete. Differ.
SANG [48]	Single-rel	Coarse	GNN	RL
Auto-SF [25]	Multi-rel	Fine	Bi-Linear	Greedy Search
GNAS [31]	Single-rel	Fine	GAP	Dete. Differ.
MR-GNAS (ours)	Multi-rel	Fine	MR-GNN	Dete. Differ.

to address the above challenges. With the fine-grained search space and relation-aware supernet, the proposed MR-GNAS could automatically develop powerful multi-relational GNNs for discriminative node and relation representation learning.

Graph Neural Architecture Search. As a critical research branch of automated machine learning, NAS techniques [14]–[17] have been introduced into automated GNN model design recently, leading to a promising research direction on graph NAS. Existing research on graph NAS has greatly enlarged the design picture of automated GNN development and relieved the human effort cost of discovering excellent and powerful GNN architectures [20]–[25], [30], [48]. Generally, graph NAS methods have two important components: search space defines functional candidate operations and search strategy explores optimal network architectures. Typically, GraphNAS [20] and AGNN [30] constructed the micro-level search space containing classical GNN layers and related hyper-parameters, along with architecture controllers based reinforcement learning (RL) search strategy. And SNAG [48] further simplified GraphNAS at the micro-level and introduced the macro-level inter-layer connections. Based on this search space, SANE [21] implemented DARTS [14], a gradient-based differential search strategy, on graphs for deriving optimal GNN architectures efficiently. Also taking the gradient-based DARTS as the search strategy, GNAS [31] developed a Graph Neural Architecture Paradigm (GAP) composed of two types of operations, *i.e.*, feature filtering and neighbor aggregation, to explore better architectures with the optimal depth of message passing on the graphs.

Despite excellent learning abilities, these graph NAS methods mainly work in the single-relational graph setting, significantly limiting their applications of learning on multi-relational graphs. In light of these, few researchers divert their attention to the multi-relational graph setting. For example, Auto-SF [25] worked on the score function (SF) component of KRL to automatically search in a bi-linear model based SF search space. Through a unified representation of existing prevalent SFs, Auto-SF could derive creative and KG-dependent SFs with superior KG analysis abilities. Nevertheless, the advance of automated GNN learning is not enjoyed by current KRL embedding models on multi-relational graphs. The main reason behind this lies in the complexity and difficulty of training GNNs when lacking domain-specific constraints on diverse multi-relational graph structures, as also verified by Auto-SF [25]. This further leads to instinctively less well performance of GNN embedding models than bilinear models. However, due to the powerful abilities of GNNs on complex graph structure and representation learning, it motivates us to overcome such challenges to explore the automated multi-relational GNN architecture development. Although Auto-GEL [22] attempted to involve the implicit link information of multi-relational graphs for automated multi-relational GNN design, it still falls into the coarse-grained design of search spaces by integrating existing typical GNN layer based operations straightforwardly. That means the principle of multi-relational GNN architectures is not changed, leading to significant limitations on the capacity and scope of novel multi-relational GNN architecture development.

Therefore, instead of the simple ensemble of existing GNN layer components, we propose a fine-grained search space with functional candidate operations that embrace the multi-relational message passing schema, leading to a novel relation-aware supernet followed by a gradient-based search strategy. The comprehensive comparison between the proposed MR-GNAS and other existing graph NAS models are listed in Table VII. As can be observed, our MR-GNAS is the only method that works on automated multi-relational GNN architecture development with the fine-grained search space.

VI. Conclusion

This paper proposes a novel framework for multi-relational graph neural architecture search, dubbed MR-GNAS, to automatically develop innovative and excellent multi-relational GNN architectures. Specifically, to tackle the challenges of the single-relational setting and coarse-grained search spaces in existing graph NAS, MR-GNAS first designs a fine-grained search space that embraces the full-pipe multi-relational message passing schema, and then constructs a relation-aware supernet composed of four sequentially stacked cells within the tree topology. In this way, MR-GNAS greatly enlarges the search capacity and improves the search flexibility for enabling expressive architecture search scopes. Through searching with a gradient-based strategy in the supernet, the proposed MR-GNAS could automatically derive creative multi-relational GNN architectures and jointly learn discriminative node and relation representations, enabling excellent multi-relational graph learning abilities. Experimental results demonstrate that the proposed MR-GNAS not only achieves outstanding performance on entity classification and link prediction, but also generally provides fresh ideas for future task-specific and data-driven multi-relational GNN design.

Acknowledgment

This work was partially supported by an Australian Research Council (ARC) Future Fellowship (FT210100097), Independent Research Fund Denmark under agreements 8022-00246B and 8048-00038B, the VILLUM FONDEN under agreement 34328, and the Innovation Fund Denmark centre, DIREC.

References

[1] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017, pp. 1025–1035.

[2] Y. Zhang, Q. Yao, and L. Chen, "Interstellar: Searching recurrent architecture for knowledge graph embedding," in *NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.

[3] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *ACM SIGKDD*, 2018, pp. 974–983.

[4] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE TNNLS*, vol. 32, no. 1, pp. 4–24, 2020.

[5] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels." *JMLR*, vol. 12, no. 9, 2011.

[6] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in *ICLR*, 2019.

[7] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *ESWC*. Springer, 2018, pp. 593–607.

[8] Y. Liu, Y. Zheng, D. Zhang, H. Chen, H. Peng, and S. Pan, "Towards unsupervised deep graph structure learning," in *WWW Conference*, 2022.

[9] H. Zhang, B. Wu, X. Yuan, S. Pan, H. Tong, and J. Pei, "Trustworthy graph neural networks: Aspects, methods and trends," *arXiv preprint arXiv:2205.07424*, 2022.

[10] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *IEEE TNNLS*, vol. 33, no. 6, pp. 2378–2392, 2021.

[11] Y. Zheng, S. Pan, V. C. Lee, Y. Zheng, and P. S. Yu, "Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination," *NeurIPS*, 2022.

[12] M. Jin, Y.-F. Li, and S. Pan, "Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs," in *Advances in Neural Information Processing Systems*, 2022.

[13] H. Zhang, B. Wu, X. Yang, C. Zhou, S. Wang, X. Yuan, and S. Pan, "Projective ranking: A transferable evasion attack method on graph neural networks," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3617–3621.

[14] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *ICLR*, 2018.

[15] X. Chen, L. Xie, J. Wu, and Q. Tian, "Progressive differentiable architecture search: Bridging the depth gap between search and evaluation," in *ICCV*, 2019, pp. 1294–1303.

[16] X. Li, C. Lin, C. Li, M. Sun, W. Wu, J. Yan, and W. Ouyang, "Improving one-shot nas by suppressing the posterior fading," in *CVPR*, 2020, pp. 13 836–13 845.

[17] X. Xia, X. Xiao, X. Wang, and M. Zheng, "Progressive automatic design of search space for one-shot neural architecture search," in *WCACV*, 2022, pp. 2455–2464.

[18] M. Zhang, S. Pan, X. Chang, S. Su, J. Hu, G. R. Haffari, and B. Yang, "Balenas: Differentiable architecture search via the bayesian learning rule," in *CVPR*, 2022, pp. 11 871–11 880.

[19] M. Zhang, S. W. Su, S. Pan, X. Chang, E. M. Abbasnejad, and R. Haffari, "idarts: Differentiable architecture search with stochastic implicit gradients," in *ICML*. PMLR, 2021, pp. 12 557–12 566.

[20] Y. Gao, H. Yang, P. Zhang, C. Zhou, and Y. Hu, "Graph neural architecture search." in *IJCAI*, vol. 20, 2020, pp. 1403–1409.

[21] Z. Huan, Y. Quanming, and T. Weiwei, "Search to aggregate neighborhood for graph neural network," in *ICDE*. IEEE, 2021, pp. 552–563.

[22] Z. Wang, S. Di, and L. Chen, "Autogel: An automated graph neural network with explicit link information," *NeurIPS*, vol. 34, 2021.

[23] Z. Pan, S. Ke, X. Yang, Y. Liang, Y. Yu, J. Zhang, and Y. Zheng, "Autostg: Neural architecture search for predictions of spatio-temporal graph," in *WWW Conference*, 2021, pp. 1846–1855.

[24] Y. Ding, Q. Yao, H. Zhao, and T. Zhang, "Diffmg: Differentiable meta graph search for heterogeneous graph neural networks," in *ACM SIGKDD*, 2021, pp. 279–288.

[25] Y. Zhang, Q. Yao, W. Dai, and L. Chen, "Autosf: Searching scoring functions for knowledge graph embedding," in *ICDE*. IEEE, 2020, pp. 433–444.

[26] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE TNNLS*, 2021.

[27] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019.

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

[29] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[30] K. Zhou, Q. Song, X. Huang, and X. Hu, "Auto-gnn: Neural architecture search of graph neural networks," *arXiv preprint arXiv:1909.03184*, 2019.

[31] S. Cai, L. Li, J. Deng, B. Zhang, Z.-J. Zha, L. Su, and Q. Huang, "Rethinking graph neural architecture search from message-passing," in *CVPR*, 2021, pp. 6657–6666.

[32] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion," in *AAAI*, vol. 33, 2019, pp. 3060–3067.

[33] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, "Deep graph library: A graph-centric, highly-performant package for graph neural networks," *arXiv preprint arXiv:1909.01315*, 2019.

[34] P. Ristoski, G. K. D. De Vries, and H. Paulheim, "A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web," in *ISWC*. Springer, 2016, pp. 186–194.

[35] H. Paulheim and J. Fümkranz, "Unsupervised generation of data mining features from linked open data," in *WIMS*, 2012, pp. 1–12.

[36] P. Ristoski and H. Paulheim, "Rdf2vec: Rdf graph embeddings for data mining," in *ISWC*. Springer, 2016, pp. 498–514.

[37] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *NeurIPS*, vol. 26, 2013.

[38] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *AAAI*, vol. 28, no. 1, 2014.

[39] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.

[40] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *ICML*. PMLR, 2016, pp. 2071–2080.

[41] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *AAAI*, 2018.

[42] T. D. N. Dai Quoc Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proceedings of NAACL-HLT*, 2018, pp. 327–333.

[43] R. Ye, X. Li, Y. Fang, H. Zang, and M. Wang, "A vectorized relational graph convolutional network for multi-relational network alignment." in *IJCAI*, 2019, pp. 4135–4141.

[44] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *CVSCWorkshop*, 2015, pp. 57–66.

[45] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, "Knowledge graph embedding for link prediction: A comparative analysis," *ACM TKDD*, vol. 15, no. 2, pp. 1–49, 2021.

[46] X. Jiang, Q. Wang, and B. Wang, "Adaptive convolution for multi-relational learning," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 978–987.

[47] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," in *ICLR*, 2018.

[48] H. Zhao, L. Wei, and Q. Yao, "Simplifying architecture search for graph neural network," in *CIKM Workshops*, S. Conrad and I. Tiddi, Eds., vol. 2699, 2020.