Large Language Model-driven Meta-structure Discovery in Heterogeneous Information Network

Lin Chen
Hong Kong University of Science and Technology, Hong Kong, China
lchencu@connect.ust.hk

Fengli Xu*
Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, China
fenglixu@tsinghua.edu.cn

Nian Li
Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
linian21@mails.tsinghua.edu.cn

Zhenyu Han
Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, China
hanzy19@mails.tsinghua.edu.cn

Meng Wang
Hefei University of Technology, Hefei, China
wangmeng@hfut.edu.cn

Yong Li*
Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, China
liyong07@tsinghua.edu.cn

Pan Hui*
Hong Kong University of Science and Technology (Guangzhou), China
Hong Kong University of Science and Technology, Hong Kong, China
panhui@ust.hk

Abstract

Heterogeneous information networks (HIN) have gained increasing popularity in recent years for capturing complex relations between diverse types of nodes. Meta-structures are proposed as a useful tool to identify the important patterns in HINs, but hand-crafted meta-structures pose significant challenges for scaling up, drawing wide research attention towards developing automatic search algorithms. Previous efforts primarily focused on searching for meta-structures with good empirical performance, overlooking the importance of human comprehensibility and generalizability. To address this challenge, we draw inspiration from the emergent reasoning abilities of large language models (LLMs). We propose *ReStruct*, a meta-structure search framework that integrates LLM reasoning into the evolutionary procedure. *ReStruct* uses a *grammar translator* to encode the meta-structures into natural language sentences, and leverages the reasoning power of LLMs to evaluate their semantic feasibility. Besides, *ReStruct* also employs performance-oriented evolutionary operations. These two competing forces allow *ReStruct* to jointly optimize the semantic explainability and empirical performance of meta-structures. Furthermore, *ReStruct* contains a *differential LLM explainer* to generate and refine natural language explanations for the discovered meta-structures by reasoning through the search history. Experiments on eight representative HIN datasets demonstrate that *ReStruct* achieves state-of-the-art performance in both recommendation and node classification tasks. Moreover, a survey study involving 73 graduate students shows that the discovered meta-structures and generated explanations by *ReStruct* are substantially more comprehensible. Our code and questionnaire are available at https://github.com/LinChen-65/ReStruct.

CCS Concepts

• **Information systems** → **Social networks**; **Data mining**; • **Computing methodologies** → **Knowledge representation and reasoning**.

Keywords

Heterogeneous Information Networks, Large Language Models, Graph Neural Networks.

ACM Reference Format:
Lin Chen, Fengli Xu, Nian Li, Zhenyu Han, Meng Wang, Yong Li, and Pan Hui. 2024. Large Language Model-driven Meta-structure Discovery in Heterogeneous Information Network. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671965

1 Introduction

Heterogeneous information networks (HINs) are effective in jointly modeling network topology and multi-typed relations [27], leading to their widespread adoption across various applications, such as social media [39], information retrieval [35], and recommender systems [2, 11]. To fully exploit the rich semantic information encoded in HINs, researchers have proposed to use *meta-path*s, which are

*Fengli Xu, Yong Li, and Pan Hui are the corresponding authors.

KDD'24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671965

templates of relation sequences to model the complex proximity on HINs [31]. They were later extended to *meta-structures* to capture more general interaction patterns beyond linear paths [13]. These meta-structures have been successfully utilized in heterogeneous graph neural networks (GNNs) to learn expressive representations for HINs [35, 39] for completing downstream tasks. However, the reliance on hand-crafted meta-structures, which depend on domain experts' knowledge, makes it challenging to scale up to larger and more complex HINs that are commonly encountered for real-world applications.

Driven by the importance of domain adaptation, recent research efforts have been dedicated to developing algorithms for automatic meta-structure search. Researchers propose to use genetic algorithm [11], deep reinforcement learning [23] and differentiable neural architectural search models [3] to automatically identify meta-structures that can enhance the performance of heterogeneous GNNs. However, these previous attempts primarily focus on the prediction performance of meta-structures, often resulting in highly complex structures that are challenging to interpret and prone to overfitting. Such "meta-structures" deviate from the original inspiration of meta-structure research that aims to extract semantically clear features from HINs [13].

The recent breakthrough in large language models (LLMs) [6] offers a unique opportunity to tackle the challenges of meta-structure discovery. The scaled-up versions of LLMs have exhibited emergent abilities for a wide range of complex tasks that go beyond auto-regression token generation [36]. For example, researchers have found that chain-of-thoughts prompting can effectively unlock LLMs' reasoning capability for commonsense, mathematical, and logical problems [37]. Such a general-purpose reasoning capability holds huge potential for comprehending the rich semantic information and produce human understandable knowledge from given HINs, which could be path-breaking to current performance-oriented meta-structure search algorithms.

In this paper, we propose a novel framework named *ReStruct* (short for **Re**asoning meta-**Struct**ure search) that integrates LLM reasoning into an evolutionary procedure for meta-structure search. In this framework, we design a *grammar translator* to encode meta-structures into natural language sentences with nested clauses (see Figure 2), ensuring that their semantic meanings can be readily comprehended by LLMs. Besides, we define a set of basic operations to modify a given meta-structure, allowing *ReStruct* to explore its adjacent possibilities in a valid space. Unlike pure performance-oriented search, we anticipate *ReStruct* to evaluate both semantic feasibility and empirical performance to identify promising candidates. To this end, we first design a *few-shot LLM predictor* to estimate the performance of meta-structure candidates with access to previously evaluated meta-structures from a history pool, followed by a *similarity-oriented LLM selector* to identify the most promising candidates based on the semantic similarities. After empirically evaluating the chosen candidates with heterogeneous GNNs, we design an *evolutionary updater* adopting the classic *elimination-reproduction* procedure to refine meta-structure candidates based on their performances. Finally, we design a *differential LLM explainer* that generates natural language explanations for the discovered meta-structure. It employs a chain-of-thought prompting technique to perform step-by-step *structural comprehension* and *performance attribution*. This reasoning process generates high-quality explanations by explicitly comparing the chosen meta-structures and the adjacent yet unchosen ones.

We evaluate *ReStruct* on eight representative HIN datasets. Experiments show that *ReStruct* achieves state-of-the-art performance on both recommendation and node classification tasks, and generates meaningful explanations as it searches through the solution space. To effectively assess the explainability of the discovered meta-structures, we conduct a user survey on 73 graduate students with domain knowledge in HIN research. According to the survey results, 46.6% of the participants consider the meta-structure discovered by *ReStruct* as the most comprehensible compared with three strong baselines, outperforming the second best baseline by 61.8%. Moreover, the natural language explanations generated by *ReStruct* are significantly preferred by the majority (77.6% on average) in a head-to-head comparison with baseline methods.

We summarize our main contributions below:

- We propose a novel *ReStruct* framework that integrates LLM reasoning into an evolutionary meta-structure search procedure. *ReStruct* jointly optimizes the empirical prediction performance and semantic explainability of meta-structures, by coordinating the competing forces of an *evolutionary updater* and a *semantic similarity-oriented LLM selector*. This represents a significant advancement in meta-structure search algorithms, enabling the generation of meta-structures that represent human digestible knowledge on HINs and are less prone to overfitting.
- We design a *grammar translator* to encode meta-structures as natural language sentences, which unleashes the reasoning power of LLMs to make sense of the rich semantic information on HINs. On top of this, we design a *differential LLM explainer* that can generate human-comprehensible natural language explanations for discovered meta-structures.
- We conduct extensive experiments to reveal *ReStruct*'s state-of-the-art performance on eight representative datasets. Furthermore, we carry out a user survey to validate that *ReStruct* substantially outperforms baseline methods in terms of the comprehensibility of discovered meta-structures and usefulness of generated explanations.

2 Preliminaries

Here, we provide the definitions of heterogeneous information networks, meta-paths, and meta-structures as in the literature.

Definition 2.1. **Heterogeneous Information Network (HIN)** [31]. An information network (IN) is mathematically a graph denoted as $G = \{V, E, T, R, \sigma, \phi\}$, with $V = \{v_1, v_2, ..., v_n\}$ being the set of nodes, $E = \{e_1, e_2, ..., e_m\}$ being the set of edges, $T = \{t_1, t_2, ..., t_k\}$ being the set of node types, and $R = \{r_1, r_2, ..., r_j\}$ being the set of edge types. $\sigma : V \rightarrow T$ is a function that maps each node to its associated type, and $\phi : E \rightarrow R$ is a function that maps each edge to its associated type. The network schema of G is then denoted as $S = \{T, R\}$. If $|T| > 1$ (multiple types of nodes) or $|R| > 1$ (multiple types of edges), G is a heterogeneous information network (HIN). Otherwise, it is a homogeneous information network.

Definition 2.2. **Meta-path** [31]. Given an HIN G, a meta-path $P = t_1 \xrightarrow{e_1} t_2 ... \xrightarrow{e_{p-1}} t_p$, is a sequence of node types and edge types

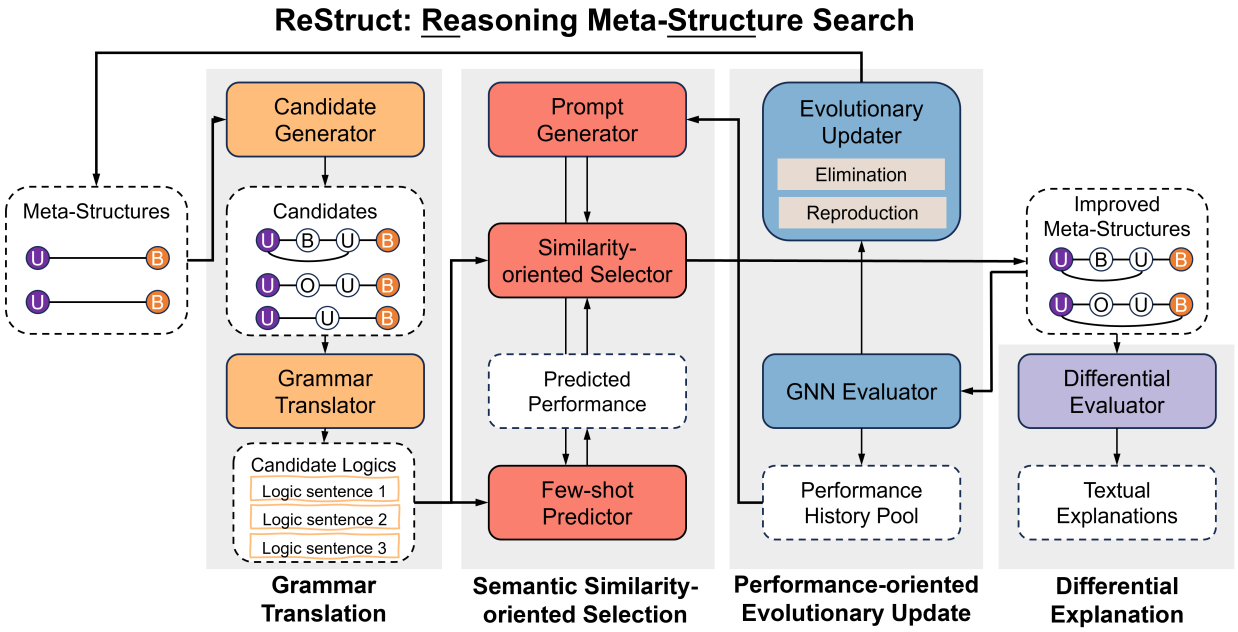

Figure 1: Overview of our proposed *ReStruct* framework.

defined on the network schema S, connecting a single source node type and a single target node type. One meta-path may correspond to many meta-path instances in G.

Definition 3.3. **Meta-structure** [13]. Given an HIN G, a meta-structure T is a generalization of the meta-path to allow for the existence of graph structures beyond linear connections between the source node type and the target node type.

3 Methods

In this section, we provide a detailed introduction to our proposed methods. In Section 3.1, we elaborate our novel design of natural language encoding of meta-structures to facilitate LLMs' understanding of its semantic meanings. In Section 3.2, we introduce our design of three basic operations for generating candidate meta-structures. In Section 3.3, we design two LLM agents to evaluate and select candidate meta-structures with semantic similarity orientation. In Section 3.4, we combine LLM-guided optimization with evolutionary processes to form an effective derivative-free optimization framework. The overview of our framework is shown in Figure 1.

3.1 Natural Language Encoding of Meta-Structures

Previous works represent meta-structures either as matrices or sets of numbers [3, 11], which can be challenging to interpret in terms of their semantic meanings. As a result, this poses obstacles for LLMs to effectively comprehend such representations. To address this limitation and enhance LLMs' comprehension of meta-structures, we design a *grammar translator* module to encode each meta-structure into a natural language sentence, as shown in Figure 2. For a given meta-structure, we begin by traversing its structure to find all possible simple paths connecting the source node to the target node. Each resulting path is equivalent to a meta-path decomposed from the original meta-structure. Next, we encode each path into a natural language sentence using nested clauses signified by a conjunction word *THAT*, which is a commonly-used grammar in English and thus expected to be well-comprehended by the LLM. In each clause, the central verb connecting two entities is the semantic meaning corresponding to the edge connecting two nodes. After obtaining the natural language encodings of the decomposed meta-paths, *i.e.*, *sub-logics*, we further combine them using another conjunction word *AND* to convey the logical summation effect.

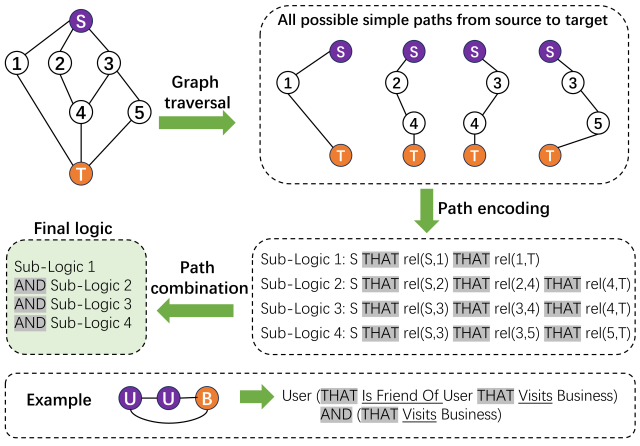

Figure 2: Natural language encoding of meta-structures.

3.2 Basic Operations for Candidate Meta-Structure Generation

To generate comprehensive candidates for LLM selection while ensuring validity, we define three basic operations for modifying any meta-structure, and design a set of components for these operations, analogous to playing with Lego blocks. Examples are shown in Figure 3.

- *INSERTION*. This operation replaces one edge of the original meta-structure with a component. It introduces new connections and expands the structure.
- *GRAFTING*. This operation takes a component, finds two nodes in the original meta-structure with the same type as the chosen component's first and last node, and merges them respectively. It creates branching structures to enhance expressiveness.
- *DELETION*. This operation removes a certain amount of nodes from the original meta-structure, and reconnects the remaining nodes to ensure the structure remains valid.

In our experiments, we take all meta-paths with no more than 2 nodes as components for *INSERTION*, and all meta-paths with no more than 3 nodes as components for *GRAFTING*. *DELETION* does not require components as input, as it regards all existing nodes on the original meta-structure as operation candidates. While using components with more nodes expands the exploration space, it may also introduce complexity and confusion for the LLM. We leave it as future work to investigate the optimal component settings.

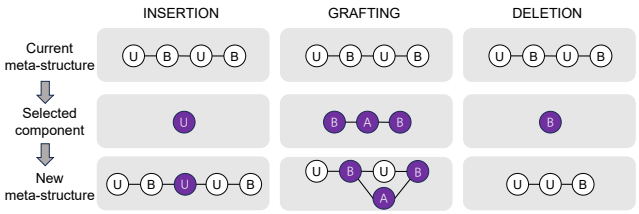

Figure 3: Basic operations of exploring adjacent meta-structures.

3.3 Semantic Similarity-Oriented LLM Agents for Candidate Selection

To regulate the explainability of discovered meta-structures, we design two LLM agents that evaluate and select candidate meta-structures in a semantic similarity-oriented manner. We illustrate the interaction processes between the main program and the LLM agents in Figure 4.

3.3.1 Few-Shot LLM Predictor. After applying the basic operations on each meta-structure, we derive a set of one-step neighbors as potential candidates. These neighbors can include meta-structures that have been encountered and evaluated in earlier generations, as well as entirely new ones. To leverage insights from previous evaluations and guide the decision-making process, we design an LLM agent as a *few-shot LLM predictor* (abbreviated as "predictor" below). This predictor estimates the performance $\hat{p}$ of each candidate through instruction tuning on a small set of structure-performance pairs sampled from a *performance pool* that records meta-structures in all previous rounds. Additionally, the predictor is asked to provide a self-estimated confidence value $\hat{c}$ for each prediction, resulting in a $(\hat{p}, \hat{c})$ pair associated with each candidate. Intuitively, if the predictor considers a candidate to be highly similar to a counterpart in the *performance pool*, it is likely to predict a similar performance and assign higher confidence to this prediction. This is grounded in the understanding that structural similarity often implies functional similarity. An illustrative prompt-response round is exemplified in Step 1 of Figure 4.

3.3.2 Similarity-Oriented LLM Selector. Upon receiving a set of candidates and their corresponding predicted performances from the *few-shot LLM predictor*, we design another LLM agent, i.e., a *similarity-oriented LLM selector* (abbreviated as "selector" below), to make the final decision of selecting one single candidate to proceed to the next generation. During this process, the selector is expected to consider multiple factors simultaneously, and potentially trade-off between them in order to make the optimal decision (see Appendix B,C). These factors include: (1) Semantic meanings, which reflect the relevance and alignment of the meta-structure with the desired objectives and requirements. (2) Structural complexities, which indicates the potential risks of overfitting. (3) Expected outcomes provided by the *few-shot LLM predictor*, which indicates the potential benefits from selecting a particular meta-structure in terms of performance improvement. (4) Credibility of outcome expectation also provided by the *few-shot LLM predictor*, which reflects the reliability and trustworthiness of the predictions. An illustrative prompt-response round is exemplified in Step 2 of Figure 4.

3.4 Performance-Oriented Evolutionary Updater

With closed-source LLM modules in the loop, it is not feasible for us to obtain the gradient for optimizing meta-structure search. Therefore, we operationalize a derivative-free optimization framework with inspirations from the genetic algorithm. Specifically, we maintain a population of N individuals, each representing a distinct meta-structure. In every generation, we first evaluate the performance of each meta-structure by using it to train a GNN for the given downstream task. After evaluation, the underperforming meta-structures are *eliminated* from the population. The surviving meta-structures undergo a *reproduction* phase, where duplication occurs with probabilities proportional to their performances. In essence, this phase uses promising meta-structures to replenish the population to its original size. Both *elimination* and *reproduction* processes mirror natural selection mechanisms that enable species evolution in the wild. After getting the modified population, we feed it into the aforementioned LLM agents to for a new round of individual meta-structure improvement. This step can be seen as a way of targeted *mutation* within the evolutionary framework, as new nodes and/or edges can be generated and some of the existing nodes and/or edges may be removed. The modified population will be re-evaluated at the onset of the next generation, forming a loop of derivative-free optimization. In summary, by utilizing this evolutionary optimization framework, we can iteratively search for

and improve meta-structures without relying on gradient-based optimization methods.

3.5 Differential LLM Explainer Agent

One prominent advantage of the LLM lies in its unparalleled ability for natural language generation. To harness this ability, we design a *differential LLM explainer* agent (abbreviated as "explainer" below) to automatically generate human-comprehensible textual explanations that elucidate the reasons behind the superior performance of discovered meta-structures. To guide the explainer in discerning the critical structural properties that contribute to performance enhancement, we design a prompting process in the *chain-of-thought* flavor [37] . Specifically, for analyzing a given meta-structure T, it unfolds in the following two steps:

Step 1: Structural Comprehension. We begin by sampling a set of n one-step neighbors for the meta-structure T, and translate each of them into a natural language sentence according to Method 3.1. Then, we prompt the *differential LLM explainer* to conduct a comprehensive analysis of both T and all its sampled neighbors. This process involves breaking down each of them into meaningful sub-structures and identifying the functions os these sub-structures.

Step 2: Performance Attribution. We first perform a quick evaluation of T and each of the sampled neighbors separately by training a GNN with one structure at a time for downstream tasks. Then, we ask the *differential LLM explainer* to identify the presence/absence of beneficial/detrimental sub-structures in the meta-structure T. This attribution process involves a joint consideration of the evaluated performances and the structural analysis conducted in the previous step.

The combination of these two steps empowers the explainer to unravel the intricate connections between structural properties and empirical performance, providing a comprehensive understanding of the discovered meta-structures. The effectiveness of this module is further justified by a user study involving human evaluators, which is elaborated in Section 4.4.2.

4 Experiments

4.1 Experimental Settings

4.1.1 Datasets. We evaluate *ReStruct* on two important tasks in HIN learning: recommendation and node classification, each with four datasets covering different fields. Detailed statistics of all eight datasets can be found in Appendix A.

In the recommendation task, our goal is to predict the existence of links between source nodes (*e.g.*, users) and target nodes (*e.g.*, items or businesses). We conduct experiments on four widely-used real-world datasets: Amazon, Yelp, Douban Movie (abbreviated as "Douban"), and LastFM[1]. For datasets including user ratings of items, the ratings are converted to 0-1 binary labels according to a threshold of 2. A label of 1 indicates the presence of preference, while 0 indicates the absence. Among the user-item pairs with the label '1', we randomly select half of them as positive pairs, which are further randomly split into training-validation-testing sets with a ratio of 3:1:1. The other half is reserved for network construction so as to prevent label leakage. We take all user-item pairs with the label '0' as negative pairs, and also randomly split them into train-validation-test sets to pair each positive pair. If the number of negative pairs is insufficient, we randomly sample unconnected items until reaching the desired number. The evaluation metric used in these experiments is AUC (Area Under the ROC Curve), which measures the model's ability to rank positive instances higher than negative instances.

In the node classification task, our goal is to predict the labels of nodes belonging to a specific type, such as determining the genre of a movie. To evaluate the effectiveness of our approach, we perform experiments on four widely-adopted real-world datasets: ACM, IMDB, DBLP, and OAG-NN. In these datasets, the classification targets correspond to the subjects of papers in ACM, the genres of movies in IMDB, the research areas of authors in DBLP, and the published venues of papers in OAG-NN, respectively. For ACM, IMDB, and DBLP, we follow the data splits used in previous works [3, 15, 43]. For OAG-NN [12], we filter the published venues with more than 100 recorded papers, and randomly split the dataset into training-validation-testing sets by 3:1:1. The evaluation metric used in these experiments is the Macro-F1 score, which measures the performance of the classification model in terms of precision and recall.

4.1.2 Baselines. We compare *ReStruct* with a set of state-of-the-art baselines. These baselines can be classified into three categories:

- Hand-crafted meta-paths: (1) metapath2vec [4], which trains a skip-gram model with meta-path guided random walks; (2) HIN2Vec [7], which learns latent vectors by jointly training for multiple prediction tasks; (3) HAN [35], which is a heterogeneous GNN that learns graph representation with multiple hand-crafted meta-paths and fuses them with a multi-head attention mechanism; (4) HERec [26], which combines random walks with an extended matrix factorization model.
- Automatically-searched meta-paths: RMSHRec [22], which adopts a reinforcement learning framework to search for meta-paths.
- Automatically-searched meta-structures: (1) GEMS [11], which employs a genetic algorithm; (2) DiffMG [3], which adopts a neural architecture search manner and searches for meta-structures in a differentiable manner; (3) PMMM [15], which further generalizes DiffMG with multi-graph search.

4.1.3 Hyperparameter Settings. For our model, we run the algorithm for 30 generations with a population size of 5 and an elimination rate of 0.2. When modifying each meta-structure, we randomly sample a set of 20 candidates if there are too many of them from the one-step neighbors. When predicting meta-structure performances with the *few-shot LLM predictor*, we randomly sample 30 records from the *performance pool* to fuel the few-shot learning paradigm. To implement the LLM agents, we use the GPT-4 model by calling the OpenAI API[2], while robustness analysis with other LLM models are also carried out (see Section 4.5). We employ the DGL implementation of HAN[3]. For all the other baseline models, we follow the implementation released by the authors. We fix the

[1] https://github.com/librahu/HIN-Datasets-for-Recommendation-and-Network-Embedding

[2] https://platform.openai.com/docs/models/

[3] https://github.com/dmlc/dgl/tree/master/examples/pytorch/han

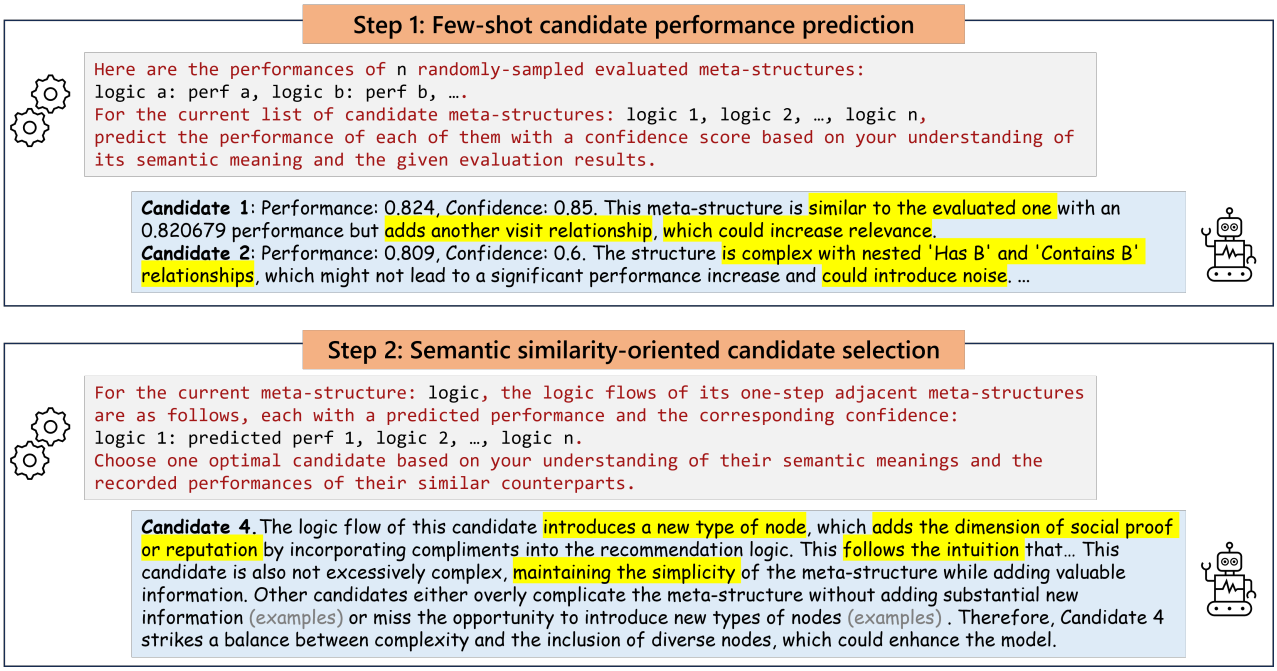

Figure 4: Example of LLM prompts and feedbacks.

Table 1: AUC (%) of recommendation on four datasets.

	metapath2vec	HIN2Vec	HAN	RMSHRec	HERec	GEMS	DiffMG	PMMM	***ReStruct***
Amazon	56.88±0.27	58.66±0.12	60.03±0.48	61.80±0.15	70.55±0.05	63.76±0.47	73.27±0.20	73.78±0.04	**75.27±0.19**
Yelp	52.29±0.59	68.06±2.31	63.88±2.32	59.39±0.81	68.37±0.34	75.46±0.28	77.63±0.40	76.76±0.17	**84.04±0.24**
Douban	52.84±0.00	85.95±0.06	63.27±0.38	79.72±0.32	92.22±0.00	90.84±0.10	93.94±0.04	94.31±0.02	**94.49±0.03**
LastFM	68.57±0.10	69.62±2.00	72.93±2.42	81.94±0.22	79.64±0.06	78.23±0.08	82.53±0.11	82.88±0.07	**85.21±0.09**

hidden dimensions to 64 for all evaluated models, and tune other hyperparameters including learning rate, weight decay, and dropout by referring to the performances on the validation set. To reduce the noise brought by randomness during program execution, for each combination of (model, task, dataset), we run experiments with 10 different random seeds and report the average performance with standard deviation.

4.2 Comparison on Recommendation

In Table 1, we report the experimental results of *ReStruct* on the recommendation task compared to baselines. First, we observe that models using meta-paths generally exhibit substantially lower performances than those using meta-structures, confirming that the stronger expression capability of meta-structures is desired for heterogeneous graph learning. Second, *ReStruct* consistently achieves the best performance across four datasets, showcasing the effectiveness of our framework in identifying meaningful and useful structures in various HINs. In particular, the performance gain over GEMS confirms that the LLM-guided "targeted mutation" converges to better solutions than pure random mutation in a classic genetic framework.

4.3 Comparison on Node Classification

In Table 2, we report the experimental results of *ReStruct* on the node classification task compared to baselines. First, metapath2vec, HIN2Vec, and HAN do not achieve desirable performances, mainly due to their heavy reliance on hand-crafted meta-structures. Second, DiffMG and PMMM demonstrate improved performances, showcasing the advantage brought by meta-structures over meta-paths, as well as the NAS searching framework. Finally, *ReStruct* achieves the best performance on ACM, IMDB, and OAG datasets. It closely aligns with state-of-the-art results on the DBLP dataset, where Macro F1 scores already exceed 94% – a level challenging to significantly surpass.

4.4 Explainability Analysis

4.4.1 Visualization of Discovered Structures and LLM-generated Explanation. Figure 5 showcases 3 discovered meta-structures that are among the top-performing ones on the Yelp dataset for recommendation, each with a summary text explaining the structural attributes underlying their outstanding performances. These summaries are generated by our *differential LLM explainer* and further

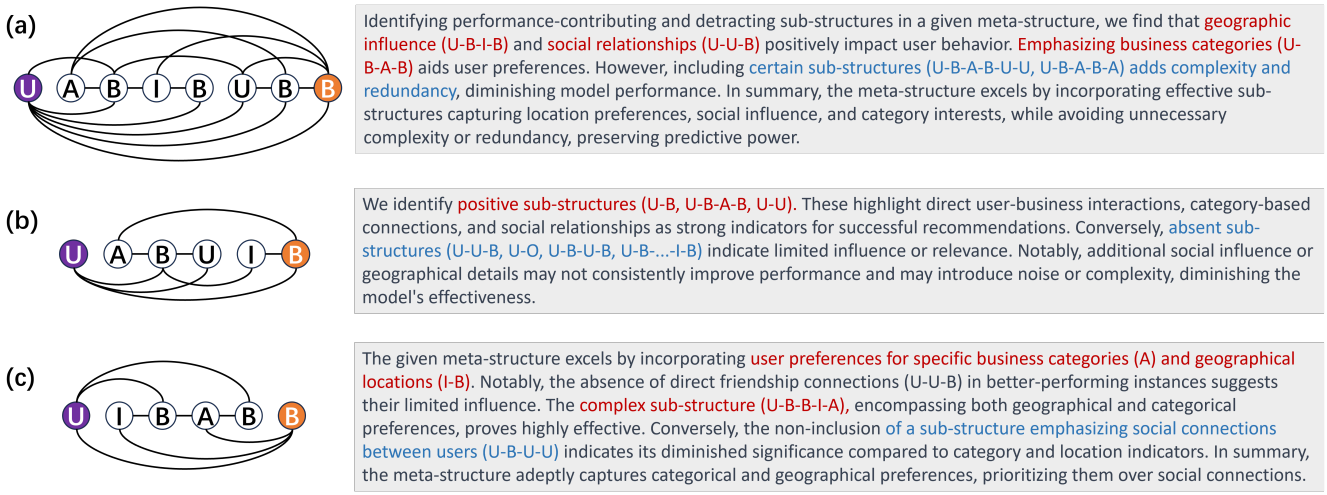

Figure 5: Discovered meta-structures on Yelp and the corresponding generated natural language explanations.

Table 2: Macro F1 scores (%) of node classification on four datasets.

	metapath2vec	HIN2Vec	HAN	DiffMG	PMMM	***ReStruct***
ACM	67.13±0.50	80.75±0.77	91.20±0.25	92.65±0.15	92.76±0.14	**92.82±0.23**
IMDB	40.82±1.48	48.16±0.44	55.09±0.67	61.04±0.56	61.69±0.40	**63.32±0.62**
DBLP	89.93±0.45	90.58±0.62	92.13±0.26	94.45±0.15	94.69±0.10	**94.09±0.36**
OAG-NN	27.61±1.65	47.13±1.14	45.80±5.84	37.67±3.13	30.18±3.87	**47.52±1.56**

condensed for clarity. We highlight the LLM-identified good (relevant) sub-structures in red, and the bad (distracting) sub-structures in blue. This visualization facilitates a comprehensive understanding of each structure's composition. For example, the meta-structure in Figure 5 (a) contains critical yet simple sub-structures describing the geographical, social, and business category contexts of user behavior. While more nodes and edges can create complex relationships, they are not always desirable for HIN learning, as showcased by the identified distracting sub-structures. These structures, though challenging to handcraft, carry semantically meaningful explanations that remain accessible with textual assistance.

4.4.2 User Study for Human Evaluation. We conduct a user study to evaluate the explainability provided by our framework compared to baselines from a human perspective. As our framework targets HIN researchers and engineers as potential users, we recruit 73 graduate students with domain knowledge of HIN research as our participants. To further ensure participants' solid understanding of the survey's processes and questions, we provide clear explanations of key concepts such as *HIN*, *meta-path* and *meta-structure*, supported by illustrative examples at the beginning of the survey. We structure the study around two sets of questions. The first set of questions is designed to assess the inherent comprehensibility of generated meta-structures without textual explanations. To this end, we present the visualizations of the best meta-structures discovered by our model alongside those from three baseline models (GEMS, DiffMG, and PMMM) [3, 11, 15], and ask the participants to select the most comprehensible one from their point of view. The second set of questions is designed to assess the comprehension gain brought by the textual explanation generated by our *differential LLM explainer* (see Method 3.5), coined the *Differential Explanation*. As a baseline, we include a *Non-Differential Explanation*, generated by directly prompting an LLM to explain the reasons behind a meta-structure's strong performance without undergoing the two-step prompting process. For three meta-structures discovered by our model on the Yelp HIN, participants are presented with both types of explanations. They are then asked to determine which one is more helpful in enhancing their understanding of how the meta-structure is constructed and gaining insights on how to design a better one. By engaging participants in head-to-head comparisons, we aim to gather valuable feedback on the relative helpfulness of each explanation type. Before starting the survey, we also carried out a pilot study [24] with 5 participants to ensure the clarity of the questions and visualizations, and none of them expressed confusion or difficulty in understanding these materials. The complete questionnaire utilized in our study is available in our GitHub repository.

Figure 6 shows the result of our first question set. Among all four models, the meta-structure discovered by our model is regarded by most people (46.6%) as the most informative or comprehensible one, outperforming the second baseline, DiffMG (28.8%), by 61.8%. GEMS and PMMM follow with the same level of recognition,

12.3%. This outcome aligns with our initial objective of discovering meta-structures that not only excel in downstream tasks but are also accessible and easily understood by human users, presumably because our framework seeks to find meta-structures that are semantically meaningful while keeping as simple as possible.

Figure 7 shows the result of our second question set. When comparing two types of generated textual explanations, the *Differential Explanation* is consistently and significantly more preferred by human participants (77.6 ± 2.8%) across various meta-structures. This outcome justifies the effective design of our *differential LLM explainer* that unleashes the LLM's reasoning ability on the intricate connections between sub-structures, sub-functions, and how they interact to determine the ultimate performances.

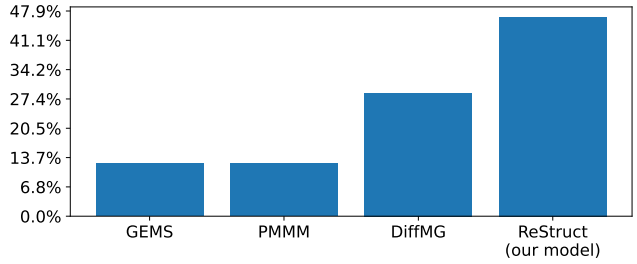

Figure 6: Human evaluated explainability of the best meta-structures discovered by different models.

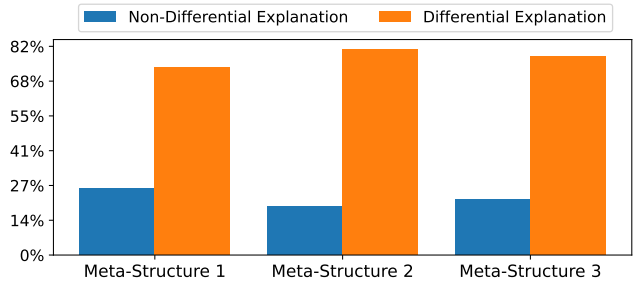

Figure 7: Explainability gain brought by LLM-generated differential meta-structure explanation compared to non-differential explanation.

4.5 Robustness Analysis

To test the robustness of our method, we first analyze its performance variation under prompt perturbation. Following previous studies [21, 29, 44], we ask ChatGPT to paraphrase our prompts while maintaining the key module designs in the framework such as grammar translation and LLM-guided performance prediction, and use them to replace the original prompts in the *ReStruct* framework. As shown in Table 3, we find that prompt paraphrasing does not have a clear impact on model performance, as long as the key designs are kept, the importance of which has been validated in the previous section.

Second, we analyze *ReStruct*'s robustness by changing the underlying LLM. Specifically, we conduct experiments with nine different LLMs from three popular series (GPT, Mistral, and GLM), with different training data, training methods, and parameter sizes. As shown in Table 4, the model performances across these LLMs consistently and significantly outperform all baselines, demonstrating the robustness of our framework against LLM versions.

5 Related Works

5.1 Identifying Meta-structures on HINs

Over the past decade, HINs have gained popularity for their ability to capture the complex relations between multi-typed nodes, which play important roles in various research areas such as information retrieval and social network modelling [27]. Meta-path, a pre-defined path template of relation sequences, was proposed to measure the similarities between nodes on HIN [31]. It allows search algorithms like *PathSim* [31] to find peer nodes that are connected by paths with different semantic meanings. The concept of meta-path was later extended beyond the linear relationship to a more general form of meta-structure [13], where the relation patterns between connected nodes can be characterized as a directed acyclic graph. Previous works have found meta-structures useful for boosting machine learning performance on HINs [14]. However, early works were based on carefully hand-crafted meta-structures, which heavily relied on experts' domain knowledge. To address this challenge, several recent works proposed to automate meta-structure design with heuristic algorithms [20], reinforcement learning [40], evolutionary search [11] and differentiable structure learning [3]. Different from previous efforts of automatic meta-structure design, we are first to leverage the emergent reasoning ability of LLMs [36] for this task. We design novel LLM agents for the automatic generation, evaluation, and explanation of novel meta-structures, which are proven effective in eight representative datasets.

5.2 Deep Learning on HINs

The success of graph neural networks introduces revolutionary deep learning techniques into HIN modeling [38]. Metapath2Vec [4] proposed to learn deep representations for nodes via meta-path-guided random walks. Attention mechanism was later introduced to learn more expressive representations for HINs, proving effective for link prediction [39] and node classification [35]. Subsequent research endeavors focused on designing more effective heterogeneous GNN frameworks [8, 16]. Besides, considerable research efforts were drawn to replace handcrafted meta-structures with automatic search. GEMS [11] proposed to combine heterogeneous GNN with evolutionary algorithms to identify meta-structures and learn deep neural networks simultaneously. Several deep reinforcement learning models and neural architecture search models are also proposed to jointly optimize the meta-structures with heterogeneous GNN [3, 15, 17, 22, 23, 33]. However, previous automatic meta-structure design method solely focused on prediction performance, often yielding complex and difficult to explain meta-structures. To the best of our knowledge, our study is the first to harness the semantic reasoning capability of LLMs for automatic meta-structure design. Our model can discover meta-structures

Table 3: Model robustness to prompt perturbation with ChatGPT (taking recommendation on Yelp as an example).

Index	Prompt for Paraphrasing	AUC (%)
0 (Original)	-	84.04±0.24
1	You are given an instruction. Now, paraphrase it into a new instruction with equivalent meaning. Instruction: {original prompt}	84.02±0.10
2	You are provided with the utterance of a specific task and I need you to paraphrase it. The actual input, question, and examples in the task should not be changed. You should only paraphrase the instructions. Task: {original prompt} The paraphrased utterance:	84.01±0.26

Table 4: Model performance with different LLM versions (taking recommendation on Yelp as an example).

Model	AUC (%)
gpt-4-1106-preview (original)	84.04±0.24
gpt-3.5-turbo-1106	84.03±0.11
mistral-tiny-2312	83.65±0.17
mistral-small-2312	83.89±0.09
mistral-small-2402	83.96±0.12
mistral-medium-2312	84.04±0.16
mistral-large-2402	83.87±0.13
glm-3-turbo	84.12±0.06
glm-4	83.76±0.07

that not only show high prediction performance, but also can be adequately explained with natural languages.

5.3 LLM for Graph Learning

LLMs have demonstrated general capabilities beyond natural language tasks, attracting graph learning researchers who are particularly interested in leveraging their ability for graph reasoning tasks [9]. Several attempts have been made to enhance node features using LLMs or employ them as standalone graph predictors [1]. The research community is also actively discussing the perspective of developing large graph models [18, 45]. Recent research has found that LLMs exhibit certain ability for graph tasks such as detecting connectivity and cycles, performing topological sort, and emulating GNNs [34]. Besides, LLMs can effectively perform reasoning on knowledge graphs [30]. To better align graph problems with LLMs, recent works propose various methods to encode the geometric structure and node features of graph problems [5, 46]. With these encodings, researchers have explored the possibility of replacing GNNs with LLM reasoning [42] and performing instruction tuning for graph tasks [32]. In this paper, we fundamentally extend LLM reasoning to HIN meta-structure discovery. Specifically, we propose a novel meta-structure encoding method, which effectively boosts LLMs' reasoning capability on HINs.

5.4 LLM for Pattern Discoveries

The scaled-up language models have emerged reasoning capability for general tasks [37], including commonsense reasoning, logical reasoning, and mathematics reasoning. With the help of optimized prompting routines such as *chain-of-thought* (CoT) [37] prompting and *tree-of-thoughts* (ToT) [41] prompting, the scaling curve of LLMs' reasoning capability can be further effectively improved. As a result, recent research has shown that LLMs can be leveraged to identify novel patterns and feasible solutions in large problem spaces. For example, *FunSearch* was proposed to discover algorithm programs for solving mathematical problems [25]. Previous works also designed LLM-driven algorithms for evolutionary search [10], reinforcement learning [28], and hyper-parameter optimization [19]. In this paper, we propose a novel framework to harness the reasoning power of LLM for meta-structure discovery. Our framework equips LLMs with enhanced capability to understand the semantic meaning of meta-structures and search for promising candidates.

6 Conclusion

In this work, we propose a novel framework, *ReStruct*, that fuses the power of LLMs with evolutionary algorithms to facilitate automatic meta-structure discovery across diverse HINs. On both recommendation and node classification tasks, extensive experiments demonstrate that *ReStruct* excels in uncovering previously undiscovered meta-structures, thereby significantly enhancing downstream model performance compared to a set of state-of-the-art baselines. Notably, a user study involving human participants confirms that *ReStruct* substantially outperforms baseline methods in terms of the comprehensibility of discovered meta-structures and usefulness of generated explanations. For future work, we will explore the feasibility of finetuning local models to mitigate network communication costs associated with API calls.

7 Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant 23IAA02114, the Guangzhou Municipal Nansha District Science and Technology Bureau under Contract No.2022ZD012, and the National Natural Science Foundation of China under Grant U23B2030 and Grant U22B2057.

References

[1] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2023. Exploring the potential of large language models (llms) in learning on graphs. *arXiv preprint arXiv:2307.03393* (2023).

[2] Jingtao Ding, Chang Liu, Yu Zheng, Yunke Zhang, Zihan Yu, Ruikun Li, Hongyi Chen, Jinghua Piao, Huandong Wang, Jiazhen Liu, et al. 2024. Artificial Intelligence for Complex Network: Potential, Methodology and Application. *arXiv preprint arXiv:2402.16887* (2024).

[3] Yuhui Ding, Quanming Yao, Huan Zhao, and Tong Zhang. 2021. Diffmg: Differentiable meta graph search for heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, 279–288.

[4] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 135–144.

[5] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2023. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560* (2023).

[6] Luciano Floridi and Massimo Chiriatti. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines* 30 (2020), 681–694.

[7] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1797–1806.

[8] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*. ACM, 2331–2341.

[9] Jiayan Guo, Lun Du, and Hengyu Liu. 2023. GPT4Graph: Can Large Language Models Understand Graph Structured Data? An Empirical Evaluation and Benchmarking. *arXiv preprint arXiv:2305.15066* (2023).

[10] Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2023. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532* (2023).

[11] Zhenyu Han, Fengli Xu, Jinghan Shi, Yu Shang, Haorui Ma, Pan Hui, and Yong Li. 2020. Genetic meta-structure search for recommendation on heterogeneous information network. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. ACM, 455–464.

[12] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*. 2704–2710.

[13] Zhipeng Huang, Yudian Zheng, Reynold Cheng, Yizhou Sun, Nikos Mamoulis, and Xiang Li. 2016. Meta structure: Computing relevance in large heterogeneous information networks. In *Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining*. ACM, 1595–1604.

[14] He Jiang, Yangqiu Song, Chenguang Wang, Ming Zhang, and Yizhou Sun. 2017. Semi-supervised Learning over Heterogeneous Information Networks by Ensemble of Meta-graph Guided Random Walks. IJCAI, 1944–1950.

[15] Chao Li, Hao Xu, and Kun He. 2023. Differentiable meta multigraph search with partial message propagation on heterogeneous information networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. AAAI, 8518–8526.

[16] Xiang Li, Danhao Ding, Ben Kao, Yizhou Sun, and Nikos Mamoulis. 2021. Leveraging meta-path contexts for classification in heterogeneous information networks. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 912–923.

[17] Yi Li, Yilun Jin, Guojie Song, Zihao Zhu, Chuan Shi, and Yiming Wang. 2021. GraphMSE: efficient meta-path selection in semantically aligned feature space for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. AAAI, 4206–4214.

[18] Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. 2023. Towards graph foundation models: A survey and beyond. *arXiv preprint arXiv:2310.11829* (2023).

[19] Siyi Liu, Chen Gao, and Yong Li. 2024. Large Language Model Agent for Hyper-Parameter Optimization. arXiv:2402.01881 [cs.LG]

[20] Changping Meng, Reynold Cheng, Silviu Maniu, Pierre Senellart, and Wangda Zhang. 2015. Discovering meta-paths in large heterogeneous information networks. In *Proceedings of the 24th international conference on world wide web*. ACM, 754–764.

[21] Raha Moraffah, Shubh Khandelwal, Amrita Bhattacharjee, and Huan Liu. 2024. Adversarial text purification: A large language model approach for defense. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 65–77.

[22] Wentao Ning, Reynold Cheng, Jiajun Shen, Nur Al Hasan Haldar, Ben Kao, Xiao Yan, Nan Huo, Wai Kit Lam, Tian Li, and Bo Tang. 2022. Automatic meta-path discovery for effective graph-based recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. ACM, 1563–1572.

[23] Hao Peng, Ruitong Zhang, Yingtong Dou, Renyu Yang, Jingyi Zhang, and Philip S Yu. 2021. Reinforced neighborhood selection guided multi-relational graph neural networks. *ACM Transactions on Information Systems (TOIS)* 40, 4 (2021), 1–46.

[24] Janice Rattray and Martyn C Jones. 2007. Essential elements of questionnaire design and development. *Journal of clinical nursing* 16, 2 (2007), 234–243.

[25] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. 2023. Mathematical discoveries from program search with large language models. *Nature* (2023), 1–3.

[26] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. 2018. Heterogeneous information network embedding for recommendation. *IEEE transactions on knowledge and data engineering* 31, 2 (2018), 357–370.

[27] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2016. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering* 29, 1 (2016), 17–37.

[28] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*. NeurIPS.

[29] Jiuding Sun, Chantal Shaib, and Byron C Wallace. 2023. Evaluating the zero-shot robustness of instruction-tuned language models. *arXiv preprint arXiv:2306.11270* (2023).

[30] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. *arXiv preprint arXiv:2307.07697* (2023).

[31] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment* 4, 11 (2011), 992–1003.

[32] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2023. Graphgpt: Graph instruction tuning for large language models. *arXiv preprint arXiv:2310.13023* (2023).

[33] Guojia Wan, Bo Du, Shirui Pan, and Gholameza Haffari. 2020. Reinforcement learning based meta-path discovery in large-scale heterogeneous information networks. In *Proceedings of the aaai conference on artificial intelligence*, Vol. 34. AAAI, 6094–6101.

[34] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. Can Language Models Solve Graph Problems in Natural Language? *arXiv preprint arXiv:2305.10037* (2023).

[35] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*. ACM, 2022–2032.

[36] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).

[37] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

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

[39] Fengli Xu, Jianxun Lian, Zhenyu Han, Yong Li, Yujian Xu, and Xing Xie. 2019. Relation-aware graph convolutional networks for agent-initiated social e-commerce recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management*. ACM, 529–538.

[40] Carl Yang, Mengxiong Liu, Frank He, Xikun Zhang, Jian Peng, and Jiawei Han. 2019. Similarity modeling on heterogeneous networks via automatic path discovery. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, Dublin, Ireland, 2018, Part II 18*. Springer, 37–54.

[41] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601* (2023).

[42] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2023. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134* (2023).

[43] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. *Advances in neural information processing systems* 32 (2019).

[44] Zekai Zhang, Yiduo Guo, Yaobo Liang, Dongyan Zhao, and Nan Duan. 2024. PPTC-R benchmark: Towards Evaluating the Robustness of Large Language Models for PowerPoint Task Completion. *arXiv preprint arXiv:2403.03788* (2024).

[45] Ziwei Zhang, Haoyang Li, Zeyang Zhang, Yijian Qin, Xin Wang, and Wenwu Zhu. 2023. Graph Meets LLMs: Towards Large Graph Models. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*. NeurIPS.

[46] Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. 2023. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089* (2023).

A Statistics of Datasets

Table 5: Statistics of datasets for node classification.

Dataset	ACM	IMDB	DBLP	OAG-NN
Node types	Author (A) Paper (P) Subject (S)	Movie (M) Actor (A) Director (D)	Author (A) Paper (P) Conference (C)	Paper (P) Author (A) Affiliation (I) Field (F)
Edge types	A-P, P-A, P-S, S-P	M-D, D-M, M-A, A-M	A-P, P-A, P-C, C-P	P-P, P-A, A-P, P-F, F-P, A-I, I-A
# Nodes	8,994	12,624	18,405	64,203
# Edges	25,922	37,288	67,946	403,974
# Classes	3	3	4	8
# Training	600	300	800	2,334
# Validation	300	300	400	778
# Testing	2,125	2,339	2,857	778

Table 6: Statistics of datasets for recommendation.

Dataset	Relations (S-T)	# S	# T	# S-T
Yelp	**User-Business (U-B)**	**16,239**	**14,284**	**84,993**
	User-User (U-U)	16,239	16,239	158,590
	User-Compliment (U-O)	16,239	11	76,875
	Business-Category (B-A)	14,284	511	40,009
	Business-City (B-I)	14,284	47	14,267
Douban	**User-Movie (U-M)**	**13,367**	**12,677**	**500,515**
	User-User (U-U)	13,367	13,367	4,085
	User-Group (U-G)	13,367	2,753	570,047
	Movie-Actor (M-A)	12,677	6,311	33,587
	Movie-Director (M-D)	12,677	2,449	11,276
	Movie-Type (M-T)	1,2677	38	27,668
Amazon	**User-Item (U-I)**	**6,170**	**2,753**	**86,191**
	Item-View (I-V)	2,753	3,857	5,694
	Item-Category (I-C)	2,753	22	5,508
	Item-Brand (I-B)	2,753	334	2,753
LastFM	**User-Artist (U-A)**	**1,892**	**17,632**	**46,417**
	User-User (U-U)	1,892	1,892	25,434
	Artist-Tag (A-T)	17,632	9,718	108,437

B Emergence of Occam's Razor Phenomenon

In our user study, we demonstrate that *ReStruct* finds meta-structures that are more comprehensible to human researchers (Figure 6), implying a tendency to avoid overcomplicated meta-structures. To provide further evidence, we add an experiment on the Yelp recommendation task, asking the LLM to select between meta-structure pairs with near-equal performance on the validation set but varying structural complexity (#nodes and #edges), which are visualized in Figure 8. As shown in Table 7, LLM consistently prefers meta-structures with lower structural complexity, citing reasons such as "simplicity," "generalizability," and even invoking "Occam's razor". Therefore, LLMs possess human-like preferences for simpler and more comprehensible meta-structures that mitigates overfitting.

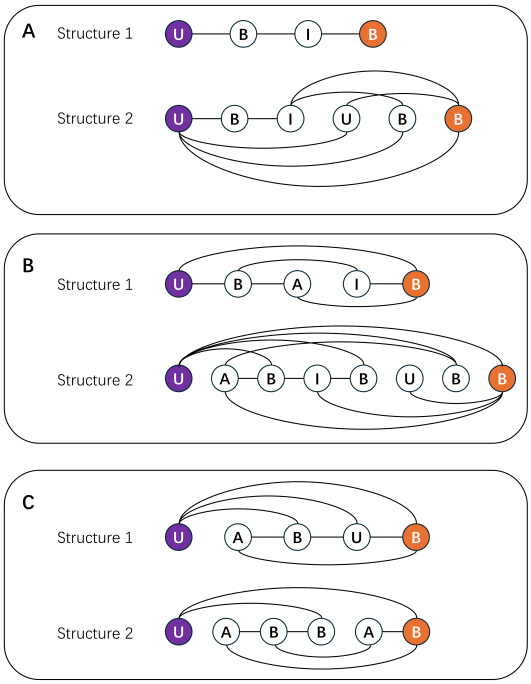

Figure 8: Visualization of meta-structure pairs for comparison.

C Evidence of opportunity-risk tradeoff during candidate selection

To check whether there is an overreliance on confidence values during candidate selection, we add an experiment to compare the confidence values between chosen and unchosen meta-structure candidates. As shown in Figure 9, *ReStruct* does not adhere strictly to the candidates with the highest confidence values. Take Experiment #14 as an example: although Candidate 0 and 5 have higher confidence values, *ReStruct* decides to choose Candidate 3, likely due to its higher predicted performance via effective semantic analysis. This highlights *ReStruct*'s attempts to balance between opportunities and risks, when predicted performances and confidence values become trade-offs.

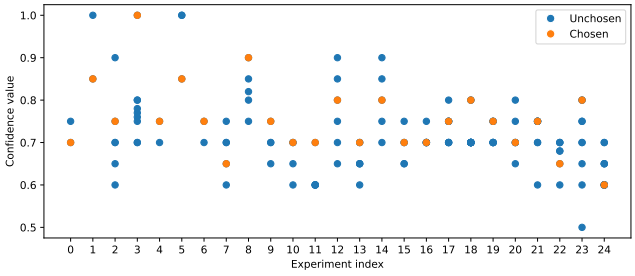

Figure 9: Comparison of confidence values between chosen and unchosen candidates.

Table 7: LLM selection between meta-structure pairs with near-equal validation performances: three examples.

Pair	Chosen Index	Reasons
1 (Fig. 8A)	1	"- Complexity vs. Performance: The second meta-structure has a more complex path, suggesting potentially richer semantics being captured. However, if both meta-structures show almost equal performance, the additional complexity might not translate into a significant performance gain. In such cases, it is generally better to opt for the simpler model (Occam's razor), which is why Meta-structure 1 is preferred. - Overfitting Risks: The more complex second meta-structure might be more prone to overfitting, as it captures a more specific set of interactions. The simpler Meta-structure 1 might generalize better to unseen data. - Interpretability: Meta-structure 1 is easier to interpret because it has a shorter path and fewer node interactions. This can be beneficial for understanding why certain recommendations are being made, which is important for transparency and trust in the recommendation system. - Computational Efficiency: With shorter paths and fewer relations, Meta-structure 1 is likely to be more computationally efficient for both training and inference, which can be important in practical, large-scale applications. - Meta-path Relevance: The paths in Meta-structure 1 adequately capture the relevant user-business interactions via a business's location. Adding more nodes and relations in the second meta-structure might not contribute much to capturing the essential features for the recommendation task. In conclusion, the first meta-structure is selected as it achieves a good balance between simplicity, performance, and interpretable results while potentially being more computationally efficient and generalizable."
2 (Fig. 8B)	1	"- Path Simplicity: Meta-structure 1 offers a more straightforward path from 'U' (User) to 'B' (Local Business) through 'A' (Business Category) and 'I' (Business City), with minimal backtracking or redundant connections. The path follows a logical progression from user preference to business through the context of category and location. - Computational Efficiency: The simpler meta-structure 1 should, in theory, be computationally less intensive because it involves fewer steps and connections. This could be advantageous when scaling the recommendation system and processing a large dataset. - Interpretability: Meta-structure 1 is easier to interpret. In the context of a recommendation system, a clear logical flow from user to business via category and city is likely to align with a user's thought process when seeking recommendations, enhancing the user experience. - Relevance of Categories and Location: By going through 'A' (Business Category) and 'I' (Business City), Meta-structure 1 reflects important aspects of businesses that users consider while making choices—what kind of service or product a business offers (category) and where it is located (city). While Meta-structure 2 includes a potentially rich and complex connection pattern that might capture more nuanced relationships, the balance between complexity and utility must be considered, particularly if both structures have shown similar performance on the validation set. Given these considerations, the simpler, more interpretable, and likely more efficient Meta-structure 1 would likely be the optimal choice for the recommendation model."
3 (Fig. 8C)	1	"- **Simplicity and Interpretability**: The first meta-structure is less complex as it contains a shorter sequence of nodes, potentially making it easier to interpret and explain how the recommendation is generated. A simpler model is often preferred when performances are almost equal because it can be more transparent, and it can generalize better, avoiding overfitting the data. - **Computational Efficiency**: Fewer node hops suggest lower computational costs in terms of executing the meta-path. This can be important for scaling the recommendation model to a large number of users or in an environment where computational resources are constrained. - **Redundancy Avoidance**: The second meta-structure involves a repeated sequence of 'Business Category' to 'Local Business' ('A' to 'B') hops, which may introduce redundancy that does not necessarily contribute to predictive performance. Such over-specified meta-structures can sometimes lead to overfitting, where the performance on the validation set is good, but the generalization to unseen data may suffer. - **Logical Coherence**: The first meta-structure provides a more direct flow of information from 'User' to 'Local Business' through their categories and mutual connections to other users. This logical flow is more coherent compared to the second meta-structure, which seems to loop back through 'Business Category' without a clear additional benefit. - **Real-World Significance**: From a real-world perspective, the first meta-structure captures a user's preference through the categories of businesses they have visited and the influence of their friends' preferences for the same business categories. This could be a robust basis for recommendations without the need for the extra category-business-category loop in the second meta-structure. Given these reasons, meta-structure 1 seems to be the optimal choice for improving the recommendation model on this Yelp HIN."