Towards Graph Foundation Models: Training on Knowledge Graphs Enables Transferability to General Graphs

Kai Wang[†], **Siqiang Luo**[†,*], **Caihua Shan**[‡], **Yifei Shen**[‡],
[†]Nanyang Technological University [‡]Microsoft Research Asia
[†]kai_wang@ntu.edu.sg, siqiang.luo@ntu.edu.sg
[‡]caihuashan@microsoft.com, yifeishen@microsoft.com

Abstract

Inspired by the success of large language models, there is a trend toward developing graph foundation models to conduct diverse downstream tasks in various domains. However, current models often require extra fine-tuning to apply their learned structural and semantic representations to new graphs, which limits their versatility. Recent breakthroughs in zero-shot inductive reasoning on knowledge graphs (KGs), offer us a new perspective on extending KG reasoning to general graph applications. In this paper, we introduce SCR, a unified graph reasoning framework designed to train on knowledge graphs and effectively generalize across a wide range of graph tasks and domains. We begin by designing the task-specific KG structures to establish a unified topology for different task formats. Then we propose semantic-conditioned message passing, a novel mechanism addressing the inherent semantic isolation in traditional KG reasoning, by jointly modeling structural and semantic invariance patterns in graph representations. Evaluated on 38 diverse datasets spanning node-, link-, and graph-level tasks, SCR achieves substantial performance gains over existing foundation models and supervised baselines, demonstrating its remarkable efficacy and adaptability. Our source code is available on `https://github.com/KyneWang/SCR`.

1 Introduction

In pursuit of artificial general intelligence, graph foundation models (GFMs) are designed to pretrain on large-scale graph data, learn generalizable representations, and adapt them to a wide range of downstream tasks [39, 36]. However, most GFMs still face challenges, including format mismatches between pretraining objectives and downstream tasks, and semantic discrepancies between source and target datasets. As a result, extensive fine-tuning is often required.

In contrast to homogeneous or heterophilic graphs, which define a single relation among nodes, knowledge graphs capture complex, multi-relational connections among entities. The most common task is KG reasoning, which involves learning embeddings of entities and relations to infer missing components in triples (head entity, relation, tail entity) [1, 54]. Moving beyond traditional transductive knowledge graph (KG) reasoning, recent work [17, 8] explores zero-shot inductive reasoning. This paradigm learns relation and entity representations conditioned on the graph structure, enabling generalization to unseen KGs and the inference of entirely new entities and relations without any fine-tuning. [66].

Inspired by these breakthroughs, we improve the topological transferability of GFMs from a novel perspective: pre-training on knowledge graphs using inductive reasoning as the training objective,

* The Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

and then transferring to other graph domains, such as citation and molecular graphs, to perform downstream tasks like node and graph classification. Nevertheless, developing such a GFM faces two significant problems. (1) Cross-Task Transferability: It is challenging to generalize the KG reasoning format across diverse graph tasks and transfer learned representations to general graphs effectively. (2) Semantic Transferability: Semantic features in graphs, such as node features and textual attributes, capture domain-specific knowledge beyond the graph's topology. Integrating semantic features into the inductive reasoning process remains largely unexplored.

To tackle the cross-task transferability, we first design task-specific KG structures to transform general graphs and their tasks into KG formats. As shown in Figure 1, we introduce two new entities, "label □", and "super graph △", and define three new relations, "node ◯ is attributed with label □", "node ◯ belongs to super graph △", "super graph △ is attributed with label □". These definitions allow us to reframe classification tasks as KG reasoning, predicting the tail entity based on a given relation and head entity. For example, in a citation network, performing node classification is analogous to reasoning edges between papers and labels.

The semantic transferability problem in inductive KG reasoning manifests as a semantic isolation issue. Existing models prioritize graph structure at the expense of domain semantics, and attempts to integrate semantics (e.g., via feature initialization) often compromise topological generalization. To resolve this dilemma, we propose a novel Semantic Conditional Message Passing (SCMP) framework. SCMP moves beyond simple feature injection by explicitly conditioning the message-passing mechanism on both local semantic neighbors and global semantic contexts. By leveraging knowledge pretrained from diverse sources (e.g., textual descriptions and ontological axioms), our framework enhances reasoning performance while simultaneously preserving robust generalization across both semantic and topological dimensions.

In summary, we propose the Semantic Conditional Reasoner (SCR), a novel graph reasoning framework that leverages inductive KG reasoning to advance graph foundation models. To the best of our knowledge, this is the first work to employ KG reasoning as a pretraining objective for graph foundation models. We conducted extensive experiments on link prediction, node classification, and graph classification tasks across 38 datasets from diverse domains. The results demonstrate performance improvement over existing foundation models and supervised baselines, underscoring the transferability of our approach.

2 Related Works

Transferability is key to the success of graph foundation models. Here we describe current studies by clarifying how they address differences between source and target datasets and bridge gaps across task formats [29, 85, 37, 51]. OFA [33] and ZeroG [29] leveraged pre-trained language models to encode node/class features as text, creating a unified feature space across diverse datasets. Meanwhile, OpenGraph [74] adopted masked autoencoding, and AnyGraph [73] used link prediction loss during pretraining on multiple graphs, enabling direct application to conduct node classification and link prediction tasks on new graphs. GraphAny [85] designed a novel architecture to tune model parameters through an analytical solution, allowing it to fit unseen graphs for node classification tasks. All these methods claim to work on both in-domain and cross-domain datasets.

Beyond designing model architectures and pretraining objectives, graph prompt learning is also popular to employ lightweight prompts, aiming to align pre-training with downstream tasks [89]. Depending on different unsupervised pre-training and prompt learning strategies employed, current notable approaches include GPPT [52], All-in-one [53], GPrompt [22], and GPF-plus [15]. Typically, their source and target datasets are the same graph dataset.

The existing foundation models in KGs study the transferability across different KGs [8, 66, 84]. For example, ULTRA [17] learned transferable graph representations by conditioning on relational interactions, enabling generalization to unseen KGs.

We provide a detailed version of related works in Appendix N. Unlike all the prior studies, our method SCR explores a novel scenario, training solely on common-sense KG datasets while achieving transferability across a wide range of general graphs and tasks without the need for extra fine-tuning.

3 Background

Let a knowledge graph be represented as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$ is the set of entities and $\mathcal{R}$ is the set of relations. The factual triples in the KG are denoted by $\mathcal{T} = \{(e_h, r, e_t) \mid e_h, e_t \in \mathcal{E}, r \in \mathcal{R}\}$, where each triple consists of a head entity e_h, a relation r, and a tail entity e_t. Given a query (e_q, r_q), where $e_q \in \mathcal{E}$ is the query entity and $r_q \in \mathcal{R}$ is the query relation, the goal of ***KG Reasoning*** is to identify the correct entity $e_v \in \mathcal{E}$, such that either (e_q, r_q, e_v) or (e_v, r_q, e_q) forms a valid triple in $\mathcal{G}$. In addition, we define a feature matrix $\mathcal{X} \in \mathbb{R}^{|\mathcal{E}| \times d_0}$, where each row represents a feature vector of dimension d_0 for the corresponding entity in the set $\mathcal{E}$. Now, consider a model trained on a knowledge graph $\mathcal{G}_{tr} = \{\mathcal{E}_{tr}, \mathcal{R}_{tr}, \mathcal{T}_{tr}\}$. The task of zero-shot inductive reasoning on knowledge graphs is to test the model on a new inference graph $\mathcal{G}_{inf} = \{\mathcal{E}_{inf}, \mathcal{R}_{inf}, \mathcal{T}_{inf}\}$, where both entities and relations are completely unseen during training. The whole notation used are listed in Appendix A.

CMP-based Backbone Model: For inductive KG reasoning, recent studies utilize graph neural networks based on *Conditional Message Passing* (CMP) to represent KG triples [87, 82, 83, 88, 67]. Traditional message passing neural networks, such as GCN [26], GAT [61], and GraphSAGE [23], compute *unary* node representations and lack the ability to model interactions in a node set (such as edges) [81]. Differently, for a KG $\mathcal{G}$ and trainable relation embeddings $\mathbf{R}$, a CMP-based model $\mathcal{M}_\theta = CMP(\mathbf{q}, \mathcal{G}, \mathbf{R})$ calculates the triple representations $\boldsymbol{h}_v$ for each entity e_v conditioned on the query $\mathbf{q} = (e_q, r_q)$:

$$\boldsymbol{h}_v^{(0)} = \textsc{Init}(e_q, e_v, \boldsymbol{r}_q) = \mathbb{1}_{e_q = e_v} * \boldsymbol{r}_q, \tag{1}$$

$$\tilde{\boldsymbol{h}}_v^{(l)} = \textsc{Agg}(\{\!\{\textsc{Msg}(\boldsymbol{h}_w^{(l)}, \boldsymbol{r}) | \, e_w \in \mathcal{N}_r(e_v), r \in \mathcal{R}\}\!\}), \tag{2}$$

$$\boldsymbol{h}_v^{(l+1)} = \textsc{Upd}(\boldsymbol{h}_v^{(l)}, \tilde{\boldsymbol{h}}_v^{(l)}). \tag{3}$$

At initialization of $\textsc{Init}()$ function, only the query entity e_q carries information: its hidden state $\boldsymbol{h}_q^{(0)}$ is set to a non-zero vector determined by $\boldsymbol{r}_q = \mathbf{R}[r_q]$, while all other entities are zeroed out. During message passing, this signal propagates outward, and each target entity e_v ultimately learns an embedding $\boldsymbol{h}_v$ that reflects how it is viewed from the perspective of (e_q, r_q). $\textsc{Msg}()$ is a differentiable message function that integrates two types of information: the aggregated paths between e_q and e_w as $\boldsymbol{h}_w^{(l)}$, and the edge connecting e_w to e_v as $\boldsymbol{r} = \mathbf{R}[r]$. The representations are iteratively updated over L layers through $\textsc{Agg}()$ and $\textsc{Upd}()$ functions. The final representation $\boldsymbol{h}_v^{(L)}$ is then used to predict the plausibility of triples (e_q, r_q, e_v) in KG reasoning.

These conditional representations are theoretically expressive [25] and practically effective [17]. Using a specific initialization function $\textsc{Init}()$, CMP-based models rely solely on KG structures and relation embeddings, enabling inductive reasoning on new KGs. Moreover, CMP supports parallel learning of $\boldsymbol{h}_v$ for all $e_v \in \mathcal{E}$, reducing computational costs.

4 Methodology

We propose SCR, a novel graph learning framework, to achieve zero-shot reasoning across general graph tasks (node/link/graph-level) and diverse domains beyond Knowledge Graphs. The framework is structured around three core contributions:

- **Unified Reasoning Format (Section 4.1):** We define a format that reformulates standard node classification and graph classification into inductive KG reasoning tasks, enabling cross-task transferability.
- **Semantic Conditional Message Passing (SCMP) (Section 4.2):** We introduce SCMP to enhance the utilization of semantic features while preserving topological expressive power.
- **Inductive Training and Reasoning (Section 4.3):** This section details the complete workflow for handling unseen KGs with arbitrary types. While we focus on node features in this work, the framework is readily extendable to explicit edge features using SCMP.

As present in Figure 1, the training workflow operates via a query-conditional reasoning process. After preprocessing graph features and constructing the relation graph, SCR uses a base CMP module to derive relation representations. Our proposed SCMP module then leverages this information to compute triple representations, which are ultimately scored for training. In the inference phase,

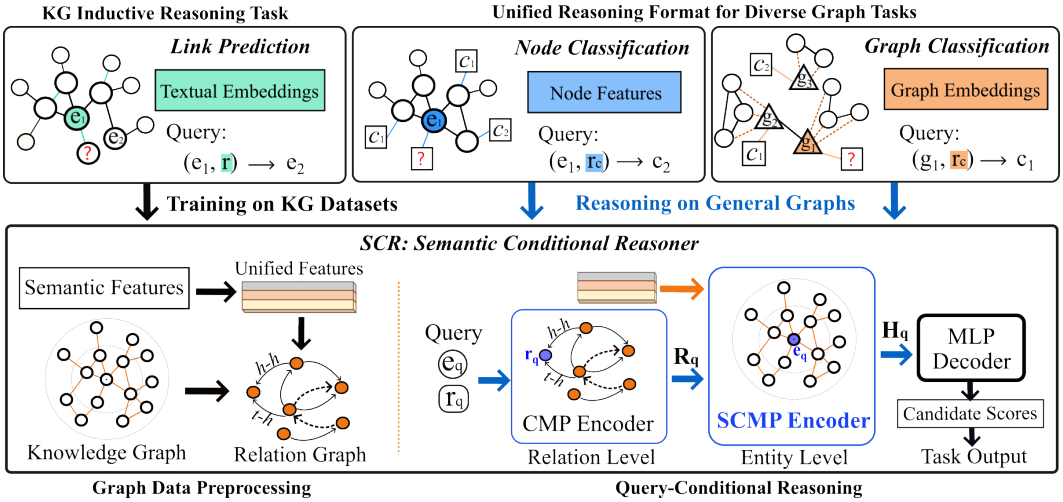


Figure 1: The proposed framework SCR transforms diverse graph tasks into inductive reasoning on knowledge graphs with semantic features.

SCR adapts to general classification tasks by transforming the graph into a KG format. This unified approach enables the learned reasoning patterns to be directly transferred for entity labeling without fine-tuning, demonstrating strong inductive generalization.

4.1 Unified Graph Reasoning Format

Here we aim to develop a unified framework that addresses node-level, edge-level, and graph-level tasks simultaneously. First, KG reasoning can be considered a specialized form of link prediction focused on a specific relation, making it straightforward to apply. Because node and graph classification tasks draw labels from a finite set, we reformulate them using the following task-specific KG structure (Definition 4.1), thereby transforming labeling tasks into KG reasoning.

Definition 4.1. (Task-specific KG Structure) For a given graph task on a dataset $\mathcal{D} = (X, Y)$, the task-specific knowledge graph $\widetilde{G}$ is constructed as follows:

$$\widetilde{G} = \{(x_i, \text{is\_attributed\_with}, y_i) | (x_i, y_i) \in \mathcal{D}\} \cup \mathcal{T}_X,$$

where $\mathcal{T}_X$ includes all original edges present within X.

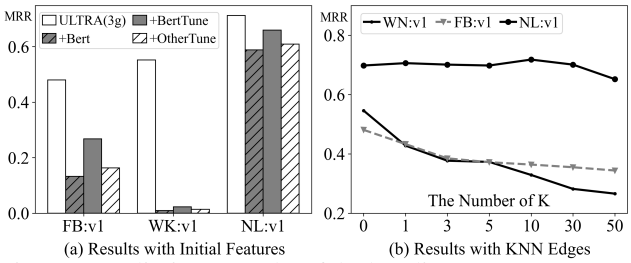
The examples are illustrated in Figure 1. In node classification tasks, a unique "label □" entity is introduced for each label type, with a defined relation "node ◯ is attributed with label □" connecting nodes to their corresponding labels. The original node connections are preserved, and the input node features are also retained as semantic features for entities. For graph classification, we integrate individual graphs into a KG structure by adding "super graph △" entities linked to their nodes via the relation "node ◯ belongs to super graph △". We then aggregate semantic features for each graph entity and add "semantically-nearest" edges between super graph nodes. The detailed procedures for task-specific KGs are given in Appendix E.

Therefore, an ideal, fully trained CMP-based graph model that generalizes across various knowledge graphs can simultaneously handle node and graph classification tasks. Knowledge graphs often contain numerous "n-to-1" relations, such as "person-to-gender" or "movie-to-genre," which are closely related to labeling tasks [9]. This relationship enables KG reasoning models to achieve strong performance in the new relation "is_attributed_with". Furthermore, this unified KG reasoning format eliminates the need to learn separate parameters for each label class, enabling support for unseen labels during inference.

4.2 Semantic Conditional Message Passing (SCMP)

Semantic Isolation Issue: Due to the specific design for topological generalization, CMP-based models cannot effectively utilize node semantic features in general graph tasks, named as the semantic isolation issue.

Some simple attempts even worsen performance. In the first attempt, we directly apply node semantic features into the INIT() function (Eq. 1) to initialize node representation $\boldsymbol{h}_v^{(0)}$, similar to standard GNNs. However, as shown in Figure 2(a), the performance degrades after injecting features ("+Bert") and even after fine-tuning with such initialization ("+BertTune"). This is because the core *target node distinguishability* assumption of CMP is violated [25], which requires, for all $r_q \in \mathcal{R}$ and $e_v \neq e_q \in \mathcal{E}$, the condition $\text{INIT}(e_q, e_q, \boldsymbol{r}_q) \neq \text{INIT}(e_q, e_v, \boldsymbol{r}_q)$ must hold [81]. Similar declines are observed when embeddings from other language models are used ("+OtherTune").



(a) Results with Initial Features (b) Results with KNN Edges

Figure 2: Preliminary results of the baseline ULTRA. "+Bert" denotes using BERT-encoded features as initialization. We also fine-tune the BERT-encoded ULTRA ("+BertTune"), followed by reasoning with features from the other source ("+OtherTune"). Higher MRR is better.

The second attempt is to introduce semantic feature similarity into the graph structure. We construct a k-nearest neighbor (KNN) graph based on the similarity of node features, and add new triples with the relation "is_semantic_similar_to" into the knowledge graph. As illustrated in Figure 2(b), increasing k reduces the performance of ULTRA. This is due to the added edges diluting local information and causing distant nodes to lose their distinctiveness. The resulting dense topology either amplifies or compresses embeddings unevenly, leading to over-smoothing and a decline in link prediction performance.

Although these attempts fail, they offer insights for improving CMP from three aspects. First, we adapt a semantic unifier to preprocess node features. We then design the new semantic-injected INIT function satisfying the assumption of target node distinguishability. Finally, we use the parameter-frozen CMP to embed the semantic node features directly.

Semantic Feature Unifier: To handle semantic diversity across domains, the semantic unifier is employed to preprocess node features without additional training. Given the feature matrix $\mathcal{X} \in \mathbb{R}^{|\mathcal{E}| \times d_0}$, we utilize singular value decomposition (SVD) in extracting important latent features:

$$\tilde{\mathcal{X}} = \text{LayerNorm}(\boldsymbol{U}\sqrt{\Lambda}),\ (\boldsymbol{U}, \Lambda, \boldsymbol{V}) = \text{SVD}(\mathcal{X}, d) \tag{4}$$

where LayerNorm$(\cdot)$ represents the layer normalization function, ensuring numerical stability. If $min(d_0, |\mathcal{E}|)$ is smaller than d, SVD will use a reduced rank to decompose $\mathcal{X}$, with the remaining dimensions zero-padded to reach d. Such that, the unified features $\tilde{\mathcal{X}} \in \mathbb{R}^{|\mathcal{E}| \times d}$ maintain consistent dimensionality d across different graph data. Besides, the relative spatial distances between nodes are preserved in the unified features due to the nature of SVD. For scalability, we employ randomized truncated SVD to ensure linear complexity.

Semantic-injected INIT Function: Given a query $\mathbf{q} = (e_q, \boldsymbol{r}_q)$, we first recall the initialization function in CMP: $\text{INIT}(e_q, e_v, \boldsymbol{r}_q) = \mathbb{1}_{e_q = e_v} * \boldsymbol{r}_q$. Instead of using the original semantic features, we inject the semantic neighbor labels into the entity initialization. The improved initialization function is defined as follows:

$$\text{INIT}^2(e_q, e_v, \boldsymbol{r}_q) = \mathbb{1}_{e_q = e_v} * \boldsymbol{r}_q + \mathbb{1}_{e_v \in \mathcal{S}_{e_q}} * \boldsymbol{v}_a, \tag{5}$$

where $\mathcal{S}_{e_q}$ represents the semantic neighbors of e_q. These neighbors are determined by selecting the top k spatially nearest entities in the semantic space based on pairwise similarities, while excluding direct topological neighbors. In addition, $\boldsymbol{v_a}$ denotes a trainable vector, randomly initialized and shared for all semantic neighbors in $\mathcal{S}_{e_q}$. In this schema, the initial representations of these neighbor entities are not all-zero vectors, enabling them to propagate effective high-order messages at the beginning of the CMP process. Note that, according to the theoretical study of CMP [25], if we assume $\boldsymbol{r}_q \neq \boldsymbol{v}_a$ and neither of them contains zero entries (Appendix H shows the assumption generally holds.), INIT^2 function satisfies the *target node distinguishability* assumption. Specifically, for all $r_q \in \mathcal{R}$ and for any $e_v \neq e_q \in \mathcal{E}$, it holds that $\text{INIT}^2(e_q, e_q, \boldsymbol{r}_q) \neq \text{INIT}^2(e_q, e_v, \boldsymbol{r}_q)$.

Non-parametric Semantic Representation: Although the new initialization function captures high-level semantic relationships among entities, the original semantic features remain excluded

from the computation process. To address this, we still use semantic features to initialize all the entities, but keep CMP parameters frozen. This setup is similar to SGC [72], a non-parametric adaptation of traditional GNNs that relies on repeated graph propagation for representation learning. Finally, an MLP is employed to merge the semantic representation $\mathbf{H}_g$ with the original CMP-based representations based on the specific query.

$$\mathbf{H}_g = CMP_\theta(\emptyset, \mathcal{G}, \mathbf{R}_g) \quad \text{where} \quad \mathbf{H}_g^{(0)} = \tilde{\mathcal{X}} \tag{6}$$

$$\mathbf{H} = CMP_\theta(\mathbf{q}, \mathcal{G}, \mathbf{R}_q) + \text{MLP}(\mathbf{H}_g), \tag{7}$$

where the parameters of both CMP instances are shared. The empty set used in Eq. 6 means that this CMP is query-independent, and we do not need to input the query. Besides, $\mathbf{R}_q$ and $\mathbf{R}_g$ are two parts of relation representations, which will be described in Sec. 4.3. Notably, $\mathbf{H}_g$ can be precomputed and seamlessly integrated into the computation process of query-specific CMP, enabling SCMP to preserve time and space complexities compared to CMP.

4.3 The Whole Process of SCR

Here we describe the entire process of training SCR on multiple KG datasets. As illustrated in the lower part of Figure 1, the first step is to preprocess the graph data. For each KG $\mathcal{G}$, we apply the semantic feature unifier to process node features, followed by the construction of the relation graph $\mathcal{G}_r$. The second step is query-conditional reasoning. Given a query $\mathbf{q} = (e_q, r_q)$ on $\mathcal{G}$, we first apply CMP to learn the relation representation $\mathbf{R_q}$ via the relation graph $\mathcal{G}_r$. Based on the relation representations $\mathbf{R_q}$, we further use our proposed SCMP to learn the triple representation $\boldsymbol{h}_v$ for (e_q, r_q, e_v). These representations are passed through an MLP to compute the scores for the existence of triples. We use cross-entropy loss to train the model for classifying positive and negative triples.

As for the inference phase, SCR unifies classification tasks as KG inductive reasoning by transforming a general graph into a KG. Thereby, the learned reasoning patterns in SCR can be adapted to label the entity without fine-tuning corresponding samples.

Build the Relation Graph $\mathcal{G}_r$: Given a KG $\mathcal{G}$, a relation graph $\mathcal{G}_r$ is constructed following ULTRA [17], to connect unseen relation types in $\mathcal{G}$ with four types of relation-level interactions (i.e., "head-to-head", "tail-to-tail", "head-to-tail", and "tail-to-head"). Please refer to Appendix C for further details.

Learn the Relation Representation $\mathbf{R_q}$: We then learn the relation embeddings via $\mathcal{G}_r$. Specifically, given a query $\mathbf{q} = (e_q, r_q)$, the calculation process is as follows:

$$\mathbf{R}_g = CMP_\phi(\emptyset, \mathcal{G}_r, \mathbf{P}) \quad \text{where} \quad \mathbf{R}_g^{(0)} = \mathbf{1} \tag{8}$$

$$\mathbf{R}_q = CMP_\phi(\mathbf{q}, \mathcal{G}_r, \mathbf{P}) \tag{9}$$

where $\mathbf{P}$ denotes the learnable embeddings corresponding to four types of interactions in the relation graph $\mathcal{G}_r$. For the query $\mathbf{q}$, the query-conditional relation representations $\mathbf{R}_q$ are generated using CMP on $\mathcal{G}_r$. Alternatively, when no query is provided and the initialized embedding $\mathbf{R}_g^{(0)}$ is set as an all-ones vector, the query-independent representations $\mathbf{R}_g$ are computed and utilized in Eq. 6.

Query Conditional Reasoning: Based on $\mathbf{R_q}$ and $\mathbf{R}_g$, we utilize our proposed SCMP model to learn the entity representation given the query $\mathbf{q} = (e_q, r_q)$:

$$\mathbf{H} = SCMP_\theta(\mathbf{q}, \mathcal{G}, \tilde{\mathcal{X}}, \mathbf{R}_q, \mathbf{R}_g), \tag{10}$$

$$p(\boldsymbol{q}, e_v) = \text{MLP}(\boldsymbol{h}_v) \tag{11}$$

where $\boldsymbol{h}_v \in \boldsymbol{H}$ denotes the final entity representation of the entity e_v. To evaluate the plausibility of the triple (e_q, r_q, e_v), an MLP is employed to compute a score, where a higher value indicates a greater likelihood of the triple being valid in $\mathcal{G}$.

Model Training: KG inductive reasoning models are typically trained by minimizing the binary cross-entropy loss over positive and negative triples. To handle semantic features across domains, we train one SCR model with multiple types of semantic features on diverse KG datasets. Specifically, we employ the BERT [10] sentence encoder to generate semantic features. We also incorporate ontology features and explore a non-feature scenario during model training, as presented in Appendix D. Note that, we focus on node semantics in this work, edge semantics can be supported with trivial

Table 1: Performance on KG inductive reasoning datasets. "(3g)" means training with three KGs, and "SCR-X" refers to results obtained using different types of semantic features (e.g., "One" means all-ones features). The best results are in bold.

Methods	IndE(FB)		IndE(WN)		IndE(NL)		IndER(FB)		IndER(WK)		IndER(NL)		**Total AVG**	
	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10
Supervised SOTA	0.477	0.636	0.640	0.734	0.464	0.654	0.166	0.296	0.152	0.244	0.296	0.481	0.366	0.507
ULTRA(3g)	0.486	0.667	0.517	0.678	0.561	0.742	0.386	0.599	**0.254**	0.403	0.393	0.561	0.433	0.608
ULTRA(4g)	0.491	0.670	0.567	0.689	**0.616**	**0.803**	0.387	0.598	0.251	**0.415**	0.398	0.588	**0.451**	0.627
ULTRA(50g)	0.493	0.664	0.558	0.664	0.590	0.777	0.382	0.585	0.251	0.406	0.397	0.582	0.445	0.613
ProLINK(3g)	0.494	0.684	0.553	0.690	0.546	0.759	0.372	0.591	0.234	0.393	0.400	0.590	0.433	0.618
SCR (3g)	**0.495**	**0.688**	**0.576**	**0.703**	0.592	0.791	**0.392**	**0.611**	0.251	0.407	**0.403**	**0.599**	**0.451**	**0.633**
SCR-One	0.491	0.678	0.569	0.688	0.581	0.773	0.390	0.604	0.250	0.399	0.388	0.578	0.445	0.620
SCR-MPNet	0.495	0.688	0.578	0.704	0.589	0.788	0.392	0.611	0.250	0.406	0.403	0.601	0.451	0.633
SCR-MiniLM	0.496	0.687	0.576	0.702	0.585	0.788	0.392	0.611	0.250	0.406	0.405	0.604	0.451	0.633
SCR-DistilBert	0.495	0.688	0.576	0.706	0.584	0.788	0.392	0.610	0.250	0.407	0.401	0.601	0.450	0.633
SCR-Ontology	0.489	0.684	0.570	0.679	0.575	0.772	0.387	0.605	0.230	0.395	0.391	0.584	0.440	0.620

modifications to our framework. At predefined mini-batch intervals, the feature type is reselected to help the model adapt to diverse input features and improve its generalization ability. The total pretraining loss is computed as follows:

$$\mathcal{L} = \sum_{\mathcal{X} \in \mathcal{F}} \left(-\log p(\boldsymbol{q}, e_a | \mathcal{X}) - \frac{1}{n} \sum_{i=1}^{n} \log(1 - p(\boldsymbol{q}, e_i | \mathcal{X})) \right)$$

Here, $p(\boldsymbol{q}, e_a|\mathcal{X})$ is the score for a positive triple in KG $\mathcal{G}$ with the node features $\mathcal{X}$, while $\{(\boldsymbol{q}, e_i)|\mathcal{X}\}_{i=1}^{n}$ contains negative samples created by corrupting the target entity.

Although the training objective is link prediction, it learns the neighborhood connectivity and therefore captures local neighborhoods and community structures. Global semantics are retained via our non-parametric semantic representation, which aggregates all original node features in the CMP process. Together, these components enable representations learned during KG training to transfer to downstream node- and graph-level tasks that rely on local structures and semantic cues.

We analyze the expressive power of our proposed SCR in Appendix F, and discuss the computational complexity and scalability in Appendix I.

5 Experiments

We evaluate our method on 38 diverse datasets across three-level tasks. In particular, we wish to answer the following research questions: **RQ1**: How effective is SCR in inductive reasoning across distinct knowledge graphs? **RQ2**: To what extent does SCR generalize across diverse feature spaces on the same KG? **RQ3**: How well does SCR generalize across a variety of graph-related tasks? **RQ4**: What is the impact of the main components on the performance of SCR? **RQ5**: How does the reasoning performance change when adjusting the key hyperparameters? Due to the space limitation, discussions about RQ5 are detailed in Appendix G.

5.1 Experimental Setup

We pre-train SCR on three commonly-used KG datasets, WN18RR [2], FB15k237 [59], and CodexM [46].The CMP follows NBFNet with a non-parametric DistMult [78] message function and a simplified PNA aggregation function [67]. For semantic features, we employ the BERT [10] sentence encoder to generate pre-training features. Hyperparameters are selected through grid search based on the metrics from the validation set without fine-tuning for each dataset. Implementation details and hyperparameter configurations are provided in Appendix D. Three graph learning tasks are used to evaluate: link-level KG inductive reasoning and node-/graph-level classification on general graphs, across 38 real-world datasets. The details of tasks and datasets are described in Appendix B.

Table 2: The accuracy results on node classification datasets. GraphAny(X) or AnyGraph(X) indicates pertaining on the X dataset. SCR-20% uses 20% of the "node-label" edges from the training set, while SCR-5 includes five edges per class. The best results are bolded.

Learning Paradigm	Methods	Cora	Citeseer	Pubmed	Wisconsin	Texas	Actor	**Avg.Rank**
Full-Shot Training	MLP	48.42±0.63	44.40±0.44	69.50±1.79	66.67±3.51	48.65±4.01	33.95±0.80	8.83
	GCN	81.40±0.70	63.40±0.63	76.60±0.32	37.25±1.64	51.35±2.71	28.55±0.68	7.33
	GAT	81.70±1.43	69.10±1.59	77.30±0.60	52.94±3.10	54.05±2.41	27.30±0.22	5.67
Graph Pre-Training Full-Shot Analytical Tuning	GraphAny(Products)	79.36±0.23	67.94±0.29	76.54±0.34	65.89±2.23	**73.52±2.96**	28.99±0.61	5.00
	GraphAny(Arxiv)	79.38±0.16	68.34±0.23	76.36±0.17	65.10±3.22	72.97±2.71	28.60±0.21	5.50
	GraphAny(Wisconsin)	77.82±1.15	67.50±0.44	77.46±0.30	**71.77±5.98**	73.51±1.21	**29.51±0.55**	4.33
	GraphAny(Cora)	80.18±0.13	68.90±0.07	76.60±0.31	61.18±5.08	71.89±1.48	27.91±0.16	5.17
Graph Pre-Training No Tuning	OpenGraph	80.65±0.69	69.99±0.83	80.15±1.28	24.42±5.64	21.78±6.32	16.74±5.68	7.33
	AnyGraph(Link1)	58.57±7.82	51.93±6.04	62.75±2.55	1.51±0.37	0.57±0.19	5.49±0.31	12.83
	AnyGraph(Link2)	69.05±4.71	45.52±4.26	78.02±1.46	1.29±0.24	0.81±0.60	5.56±0.21	11.00
KG Pre-Training No Tuning	ULTRA(3g)	79.40±0.00	67.40±0.00	77.90±0.00	49.02±0.00	56.76±0.00	22.61±0.00	7.50
	SCR (3g)	**81.80±1.02**	**71.33±0.27**	**82.93±0.55**	54.91±1.51	67.64±0.44	23.26±0.56	**3.67**
Few-Shot Labeling	SCR-20%	73.62±2.78	56.50±3.53	71.94±0.23	50.59±3.14	64.32±3.59	22.78±0.49	9.00
	SCR-5	54.48±1.96	32.38±3.58	50.38±6.17	28.63±8.82	58.38±4.39	19.00±1.22	11.67

Table 3: The accuracy results on graph classification datasets. SCR-20% and SCR-5 are two few-shot labeling variants of SCR. The best results are bolded.

Learning Paradigm	Methods	IMDB-B	COLLAB	PROTEINS	MUTAG	ENZYMES	COX2	BZR	DD	**Avg.Rank**
Full-Shot Training	GIN	**67.75±2.50**	58.20±10.22	64.72±0.84	75.50±5.74	21.88±0.55	77.90±1.57	**81.79±2.94**	**70.59±0.81**	3.13
One-Shot Training	GCN	57.30±0.98	47.23±0.61	56.36±7.97	65.20±6.70	20.58±2.00	27.08±1.95	25.80±6.53	55.33±6.22	10.25
	Pretrain&Finetune	57.75±1.22	48.10±0.23	63.44±3.64	65.47±5.89	22.21±2.79	76.19±5.41	34.69±8.50	57.15±4.32	7.25
Graph Pre-Training One-Shot Tuning	GPPT	50.15±0.75	47.18±5.93	60.92±2.47	60.40±15.43	21.29±3.79	**78.23±1.38**	59.32±11.22	57.69±6.89	8.50
	All-in-one	60.07±4.81	51.66±0.26	66.49±6.26	79.87±5.34	**23.96±1.45**	76.14±5.51	79.20±1.65	59.72±1.52	3.75
	GPrompt	54.75±12.43	48.25±13.64	59.17±11.26	73.60±4.76	22.29±3.50	54.64±9.94	55.43±13.69	57.81±2.68	7.50
	GPF	59.65±5.06	47.42±11.22	63.91±3.26	68.40±5.09	22.00±1.25	65.79±17.72	71.67±14.71	59.36±1.18	6.25
	GPF-plus	57.93±1.62	47.24±0.29	62.92±2.78	65.20±6.04	22.92±1.64	33.78±1.52	71.17±14.92	57.62±2.42	7.25
KG Pre-Training No Tuning	ULTRA(3g)	49.25±0.00	51.80±0.00	58.09±0.00	63.33±0.00	15.21±0.00	77.75±0.00	79.32±0.00	43.52±0.00	8.50
	SCR (3g)	61.83±1.60	**65.45±1.05**	**68.54±1.47**	**85.33±2.11**	22.92±2.03	78.08±1.33	79.32±0.06	69.96±0.74	**1.75**
Few-Shot Labeling	SCR-20%	53.45±3.46	60.72±1.07	66.13±4.08	52.93±14.37	17.25±1.29	75.5±5.06	79.51±0.37	69.75±3.19	6.13
	SCR-5	53.37±2.83	46.25±8.93	61.69±8.57	80.27±5.82	22.58±1.15	58.12±2.11	46.05±12.11	62.14±4.5	7.38

5.2 Main Experimental Results (RQ1)

All 24 inductive KG reasoning datasets are used in RQ1: the first 12 datasets from GraIL [58] with test graphs containing only unseen entities (termed as "IndE"), and the remaining 12 datasets from InGram [28] featuring both unseen entities and relations (termed as "IndER"). Notably, eight datasets in IndE/IndER(NL) come from NELL-995 (excluded from training), introducing new semantic features for each method. This setting prevents data leakage by dynamically generating entity representations based on the unique structure of each KG during training and inference. Even if a triple appears in both the pre-training and test datasets, the different structures around it ensure distinct representations, thereby mitigating memorization.

We compare SCR with two KG reasoning baselines (ULTRA and ProLINK pre-trained on different sizes of KGs) and one supervised SOTA. Two evaluation metrics are used: Mean Reciprocal Rank (MRR) and Hits@N, where higher scores indicate better performance [1, 58]. For semantic features, we generate entity-level embeddings from available textual descriptions using BERT sentence encoders.

We report the average performance for each benchmark and the results are summarized in Table 1. A comprehensive evaluation on more than 50 transductive and inductive KG datasets is provided in Appendix M. Overall, SCR outperforms all existing zero-shot models as well as the supervised model in the total average metrics, demonstrating its effectiveness. For individual benchmarks, we observe that SCR surpasses ULTRA(3g), ProLINK, and supervised results in most metrics. Although ULTRA(4g) and ULTRA(50g) achieve better performance on some metrics, they are pre-trained on more diverse KGs (including NELL-995) and still show poorer performance on the IndER(FB) and IndE(WN) benchmarks. Furthermore, compared to IndER(X) benchmarks, SCR shows substantial performance gains on IndE(X) benchmarks beneficial from node semantic features.

5.3 Generalizing Across Semantic Spaces (RQ2)

To address the cross-domain challenge, we explore the generalizability of SCR across different semantic feature spaces, and verify that SCR is not restricted to graphs with textual features. Specifically, we select five types of semantic features as input to the pre-trained SCR (within BERT-encoded semantic space). In Table 1, "MPNet", "MiniLM", and "DistilBERT" refer to three popular sentence encoders based on language models, while "One" and "Ontology" utilize all-ones features and ontology features derived from relation type counting, respectively. The results show that SCR variants using different sentence encoders achieve performance nearly identical to the original SCR, despite the encoders producing embeddings with varying dimensions. This consistency underscores the effectiveness of our unified semantic space. Although SCR (One) and SCR (Ontology) perform slightly worse than SCR on some metrics, they still outperform baselines such as ULTRA(3g). It indicates that SCR is not constrained by access to textual features and exhibits generalization capabilities across diverse feature sources, even all-ones features. The semantic content of IndE(NL) and IndER(NL) was not included in the pre-training KGs. Despite this, the performance improvement still demonstrates the robustness of SCR to unseen domains or semantic inputs. The key lies in the pre-training design of SCR: it systematically trains the model to handle diverse scenarios—from KGs with rich semantics (text/ontology) to those with no input features. SCR learns to reason over KGs with diverse semantics rather than depending on auxiliary textual data.

5.4 Generalizing Across Graph Tasks (RQ3)

In this section, we verify the performance of SCR on classification tasks across diverse general graphs. Following prior studies [89], we employ six node classification datasets, including homophilic graphs (Cora, Citeseer, PubMed) [49, 41] and heterophilic graphs (Wisconsin, Texas, Actor) [42, 56]. We utilize eight graph classification datasets from various domains, covering social networks (IMDB-B, COLLAB) [77], biological (ENZYMES, PROTEINS, DD) [11, 3, 69], and small molecule datasets (MUTAG, COX2, BZR) [27, 44]. The experimental results of prediction accuracy are shown in Table 2 and Table 3. Comprehensive results can be found in Appendix O.

For node-level tasks, SCR significantly outperforms existing foundation models on three homophilic graphs, and shows superior performance over ULTRA, OpenGraph, and AnyGraph on three heterophilic graphs. However, SCR lags behind GraphAny on heterophilic graphs, as GraphAny utilizes training labels to tune the model parameters through an analytical solution. Such task-specific tuning in GraphAny and graph prompt methods is powerful but limits their versatility as graph foundation models. To improve robustness on heterophilic graphs, several promising strategies can be explored, such as pre-training or fine-tuning on heterophilic structures, or replacing standard aggregation functions with operators more suitable for heterophilic settings. Enhancing SCR on heterophilic graphs will be our future work.

Table 3 shows the experimental results for graph-level tasks. Following the experimental setup from ProG [89], 80% of graph samples are divided into the test set and only a few labeled graphs are transformed into the task-specific KG. We observe that SCR outperforms existing graph models using one-shot training or prompt tuning. Further, SCR is competitive with the fully-trained GIN using the same training set, while GIN requires extra training time and SCR directly infers on the task-specific KG. Existing zero-shot foundation models such as OpenGraph, AnyGraph, and GraphAny are not suitable for graph classification tasks. This limitation underscores the value of SCR across diverse graph tasks. In addition, the fact that performance gains are achieved on IMDB-B and COLLAB, which lack node features, demonstrates that the improvement beyond ULTRA is not exclusively dependent on semantic features.

5.5 Ablation Studies (RQ4)

To assess the impact of the key techniques, we conduct ablation experiments for multiple pre-trained variants. The results are illustrated in Figure 3.

(1) Semantic Neighbors: The two variants, excluding Semantic-Augmented Relation Graph ("w/o SARG") and Semantic-Injected Entity Initialization ('w/o INIT2"), utilize the original relation graph and initialization function of ULTRA(3g) to omit semantic neighbor information. Across three tasks, the observed performance declines underscore the effectiveness of our semantic-injection techniques. The new INIT2 function has a relatively larger contribution, especially in graph-level tasks.

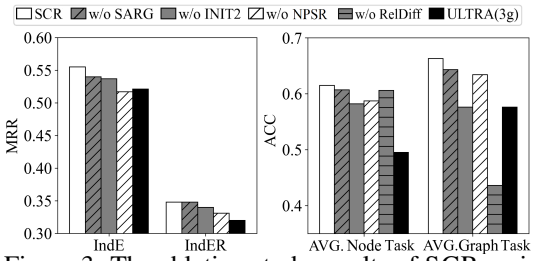


Figure 3: The ablation study results of SCR variants. "w/o SARG" denotes using the relation graph without semantic neighbors, the latter two have no semantic-injected initialization and global semantic encoding, respectively. "w/o RelDiff" denotes using identical relation embeddings in SCMP.

(2) Non-parametric Semantic Representation: The non-parametric semantic encoding in SCMP directly handles semantic features, with the reduced performance of "w/o NPSR" emphasizing its essential role in leveraging semantic diversity. However, except for IndE datasets, the observation that "w/o NPSR" outperforms the original ULTRA in most tasks suggests that the post-merging approach effectively preserves the functionality of CMP.

(3) Relational Condition: The variant "w/o RelDiff" utilizes the average vector of $\mathbf{R}_q$ as a substitute for each individual vector in $\mathbf{R}_q$, effectively removing the influence of relation type differences on entity-level inference. Despite the presence of only a few relation types in node/graph-level graph structures, the observed performance decline highlights the essential role of relational information. Notably, in graph-level tasks, relation-specific embeddings play a vital role in distinguishing rare graph-label relations from semantic and topological edges.

(4) Few-Shot Labeling: We evaluate the performance of SCR in scenarios where the 'node/graph-label' edges are sparse within the graph. "SCR-20%" uses 20% of the "node-label" edges from the training set, while "SCR-5" includes five edges per class. Both variants have no fine-tuning process, but utilize different scopes of "label information" when inference on the pre-trained model. As shown in Table 2 and Table 3, reducing the number of labels in the task-specific KG leads to a decline in reasoning performance. However, "SCR-20%" and "SCR-5" still obtain better accuracy than some baselines, which highlights the effectiveness of our method. In a few small-scale datasets, such as MUTAG and ENZYMES, the performance of "SCR-20%" is lower than "SCR-5" due to the former having fewer accessible label edges.

6 Conclusion

In this paper, we take the first successful step toward generalizing inductive KG reasoning in graph foundation models. The proposed method, SCR, conducts semantic conditional message passing on multi-relational graphs, effectively integrating semantic features with structural information while addressing node-level, edge-level, and graph-level tasks simultaneously. Extensive experiments demonstrate that SCR achieves strong generalizability across diverse graph domains and tasks. We further discuss the limitations of our work in Appendix L. Our future research agenda will prioritize two key objectives: (1) extending the applicability of this approach to broader graph-based tasks across diverse domains, and (2) systematically validating the scaling principles that govern large-scale model architectures.

Acknowledgments

This research is supported by the National Research Foundation, Singapore, under its Frontier CRP Grant (NRF-F-CRP-2024-0005), and under its AI Singapore Programme (AISG Award No: AISG3-RP-2024-034). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore.

References

[1] Bordes, A., García-Durán, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems, December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 2787–2795, 2013.

[2] Bordes, A., Glorot, X., Weston, J., and Bengio, Y. A Semantic Matching Energy Function for Learning with Multi-relational Data. *Machine Learning*, 94:233–259, 2014.

[3] Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S., Smola, A. J., and Kriegel, H.-P. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005.

[4] Chami, I., Wolf, A., Juan, D.-C., Sala, F., Ravi, S., and Ré, C. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 6901–6914, 2020.

[5] Chen, J., He, H., Wu, F., and Wang, J. Topology-aware correlations between relations for inductive link prediction in knowledge graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 6271–6278. AAAI Press, 2021. URL https://doi.org/10.1609/aaai.v35i7.16779.

[6] Chen, M., Liu, Z., Liu, C., Li, J., Mao, Q., and Sun, J. ULTRA-DP: Unifying Graph Pre-training with Multi-task Graph Dual Prompt. *arXiv preprint*, 2023.

[7] Corso, G., Cavalleri, L., Beaini, D., Liò, P., and Velickovic, P. Principal neighbourhood aggregation for graph nets. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[8] Cui, Y., Sun, Z., and Hu, W. A prompt-based knowledge graph foundation model for universal in-context reasoning. *CoRR*, abs/2410.12288, 2024. doi: 10.48550/ARXIV.2410.12288. URL https://doi.org/10.48550/arXiv.2410.12288.

[9] Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 1811–1818, 2018.

[10] Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

[11] Dobson, P. D. and Doig, A. J. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003.

[12] Du, E., Liu, S., and Zhang, Y. Graphoracle: A foundation model for knowledge graph reasoning, 2025. URL https://arxiv.org/abs/2505.11125.

[13] Dwivedi, V. P., Rampášek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D. Long range graph benchmark, 2023. URL https://arxiv.org/abs/2206.08164.

[14] Fang, P., Luo, S., Wang, F., Zheng, B., Jiang, H., Feng, D., Pan, H., and Wan, X. Omega: Boosting large-scale graph embeddings with heterogeneous memory processing. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 3369–3383. IEEE Computer Society, 2025.

[15] Fang, T., Zhang, Y., YANG, Y., Wang, C., and Chen, L. Universal prompt tuning for graph neural networks. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 52464–52489. Curran Associates, Inc., 2023.

[16] Galkin, M., Denis, E. G., Wu, J., and Hamilton, W. L. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[17] Galkin, M., Yuan, X., Mostafa, H., Tang, J., and Zhu, Z. Towards foundation models for knowledge graph reasoning. *CoRR*, abs/2310.04562, 2023. URL https://doi.org/10.48550/arXiv.2310.04562.

[18] Gao, J., Zhou, Y., and Ribeiro, B. Double permutation equivariance for knowledge graph completion. *arXiv preprint arXiv:2302.01313*, 2023. URL https://doi.org/10.48550/arXiv.2302.01313.

[19] Ge, Q., Zhao, Z., Liu, Y., Cheng, A., Li, X., Wang, S., and Yin, D. Enhancing graph neural networks with structure-based prompt. *CoRR*, abs/2310.17394, 2023.

[20] Geng, Y., Chen, J., Pan, J. Z., Chen, M., Jiang, S., Zhang, W., and Chen, H. Relational message passing for fully inductive knowledge graph completion. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1221–1233. IEEE, 2023. URL https://doi.org/10.1109/ICDE55515.2023.00098.

[21] Gesese, G. A., Sack, H., and Alam, M. RAILD: towards leveraging relation features for inductive link prediction in knowledge graphs. In Artale, A., Calvanese, D., Wang, H., and Zhang, X. (eds.), *Proceedings of the 11th International Joint Conference on Knowledge Graphs, IJCKG 2022, Hangzhou, China, October 27-28, 2022*, pp. 82–90. ACM, 2022. URL https://doi.org/10.1145/3579051.3579066.

[22] Gong, C., Li, X., Yu, J., Yao, C., Tan, J., Yu, C., and Yin, D. Prompt Tuning for Multi-View Graph Contrastive Learning. *arXiv preprint*, 2023.

[23] Hamilton, W. L., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1024–1034, 2017.

[24] Huang, Q., Ren, H., Chen, P., Kržmanc, G., Zeng, D., Liang, P., and Leskovec, J. PRODIGY: Enabling In-context Learning Over Graphs. In *NeurIPS*, 2023.

[25] Huang, X., Romero, M., Ceylan, İ. İ., and Barceló, P. A theory of link prediction via relational weisfeiler-leman on knowledge graphs. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[26] Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[27] Kriege, N. and Mutzel, P. Subgraph matching kernels for attributed graphs. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pp. 291–298, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

[28] Lee, J., Chung, C., and Whang, J. J. InGram: Inductive knowledge graph embedding via relation graphs. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 18796–18809. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/lee23c.html.

[29] Li, Y., Wang, P., Li, Z., Yu, J. X., and Li, J. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1725–1735, 2024.

[30] Liao, N., Liu, H., Zhu, Z., Luo, S., and Lakshmanan, L. V. A comprehensive benchmark on spectral gnns: The impact on efficiency, memory, and effectiveness. *Proceedings of the ACM on Management of Data*, 3(4):1–29, 2025.

[31] Liao, N., Yu, Z., Zeng, R., and Luo, S. Unifews: You need fewer operations for efficient graph neural networks, 2025. URL https://arxiv.org/abs/2403.13268.

[32] Liu, B., Peng, M., Xu, W., and Peng, M. Neighboring relation enhanced inductive knowledge graph link prediction via meta-learning. *World Wide Web (WWW)*, 26(5):2909–2930, 2023.

[33] Liu, H., Feng, J., Kong, L., Liang, N., Tao, D., Chen, Y., and Zhang, M. One for all: Towards training one graph model for all classification tasks. *ICLR*, 2024.

[34] Liu, H., Liao, N., and Luo, S. Sigma: An efficient heterophilous graph neural network with fast global aggregation. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 1924–1937. IEEE, 2025.

[35] Liu, J. Combining structure and text: Learning representations for reasoning on graphs, 2025. URL https://openreview.net/forum?id=hJ8OQAiTrl.

[36] Liu, J., Yang, C., Lu, Z., Chen, J., Li, Y., Zhang, M., Bai, T., Fang, Y., Sun, L., Yu, P. S., and Shi, C. Towards graph foundation models: A survey and beyond. *CoRR*, abs/2310.11829, 2023. doi: 10.48550/arXiv.2310.11829.

[37] Liu, J., Mao, H., Chen, Z., Fan, W., Ju, M., Zhao, T., Shah, N., and Tang, J. One model for one graph: A new perspective for pretraining with cross-domain graphs, 2024. URL https://arxiv.org/abs/2412.00315.

[38] Liu, Z., Yu, X., Fang, Y., and Zhang, X. Graphprompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. In *The Web Conference*, pp. 417–428, 2023.

[39] Mao, H., Chen, Z., Tang, W., Zhao, J., Ma, Y., Zhao, T., Shah, N., Galkin, M., and Tang, J. Graph foundation models. *CoRR*, abs/2402.02216, 2024. doi: 10.48550/arXiv.2402.02216.

[40] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[41] Namata, G., London, B., Getoor, L., Huang, B., and Edu, U. Query-driven active surveying for collective classification. In *10th international workshop on mining and learning with graphs*, volume 8, pp. 1, 2012.

[42] Pei, H., Wei, B., Chang, K. C.-C., Lei, Y., and Yang, B. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2019.

[43] Rosenstein, M. T., Marx, Z., Kaelbling, L. P., and Dietterich, T. G. To transfer or not to transfer. In *NeurIPS*, volume 898, 2005.

[44] Rossi, R. and Ahmed, N. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

[45] Sadeghian, A., Armandpour, M., Ding, P., and Wang, D. Z. DRUM: end-to-end differentiable rule mining on knowledge graphs. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 15321–15331, 2019.

[46] Safavi, T. and Koutra, D. Codex: A comprehensive knowledge graph completion benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 8328–8350, 2020.

[47] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

[48] Schlichtkrull, M. S., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pp. 593–607. Springer, 2018.

[49] Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[50] Song, K., Tan, X., Qin, T., Lu, J., and Liu, T. Mpnet: Masked and permuted pre-training for language understanding. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[51] Sun, L., Huang, Z., Zhou, S., Wan, Q., Peng, H., and Yu, P. Riemanngfm: Learning a graph foundation model from riemannian geometry, 2025. URL https://arxiv.org/abs/2502.03251.

[52] Sun, M., Zhou, K., He, X., Wang, Y., and Wang, X. GPPT: Graph Pre-Training and Prompt Tuning to Generalize Graph Neural Networks. In *KDD*, pp. 1717–1727, 2022.

[53] Sun, X., Cheng, H., Li, J., Liu, B., and Guan, J. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2120–2131, 2023.

[54] Sun, Z., Deng, Z., Nie, J., and Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[55] Tan, Z., Guo, R., Ding, K., and Liu, H. Virtual Node Tuning for Few-shot Node Classification. In *KDD*, pp. 2177–2188, 2023.

[56] Tang, J., Sun, J., Wang, C., and Yang, Z. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 807–816, 2009.

[57] Tang, J., Yang, Y., Wei, W., Shi, L., Xia, L., Yin, D., and Huang, C. Higpt: Heterogeneous graph language model, 2024. URL `https://arxiv.org/abs/2402.16024`.

[58] Teru, K. K., Denis, E. G., and Hamilton, W. L. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9448–9457. PMLR, 2020.

[59] Toutanova, K. and Chen, D. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pp. 57–66, 2015.

[60] Vashishth, S., Sanyal, S., Nitin, V., and Talukdar, P. P. Composition-based multi-relational graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[61] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. 2018.

[62] Wang, B., Wang, G., Huang, J., You, J., Leskovec, J., and Kuo, C. J. Inductive learning on commonsense knowledge graph completion. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pp. 1–8. IEEE, 2021. URL `https://doi.org/10.1109/IJCNN52387.2021.9534355`.

[63] Wang, K., Liu, Y., Ma, Q., and Sheng, Q. Z. Mulde: Multi-teacher knowledge distillation for low-dimensional knowledge graph embeddings. In *Proceedings of the WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pp. 1716–1726, 2021.

[64] Wang, K., Liu, Y., and Sheng, Q. Z. Swift and sure: Hardness-aware contrastive learning for low-dimensional knowledge graph embeddings. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pp. 838–849. ACM, 2022.

[65] Wang, K., Xu, Y., and Luo, S. TIGER: training inductive graph neural network for large-scale knowledge graph reasoning. *Proc. VLDB Endow.*, 17(10):2459–2472, 2024.

[66] Wang, K., Xu, Y., Wu, Z., and Luo, S. LLM as prompter: Low-resource inductive reasoning on arbitrary knowledge graphs. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 3742–3759, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. URL `https://aclanthology.org/2024.findings-acl.224`.

[67] Wang, K., Lin, D., and Luo, S. Graph percolation embeddings for efficient knowledge graph inductive reasoning. *IEEE Transactions on Knowledge and Data Engineering*, 37(3):1198–1212, 2025. doi: 10.1109/TKDE.2024.3508064.

[68] Wang, Q., Mao, Z., Wang, B., and Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12): 2724–2743, 2017.

[69] Wang, S., Dong, Y., Huang, X., Chen, C., and Li, J. Faith: Few-shot graph classification with hierarchical task graphs. *arXiv preprint arXiv:2205.02435*, 2022.

[70] Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[71] Wang, Y., Fan, W., Wang, S., and Ma, Y. Towards graph foundation models: A transferability perspective. *CoRR*, abs/2503.09363, 2025. doi: 10.48550/ARXIV.2503.09363. URL https://doi.org/10.48550/arXiv.2503.09363.

[72] Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. Simplifying graph convolutional networks. In *International conference on machine learning*, pp. 6861–6871. PMLR, 2019.

[73] Xia, L. and Huang, C. Anygraph: Graph foundation model in the wild, 2024. URL https://arxiv.org/abs/2408.10700.

[74] Xia, L., Kao, B., and Huang, C. Opengraph: Towards open graph foundation models, 2024. URL https://arxiv.org/abs/2403.01121.

[75] Xiong, W., Hoang, T., and Wang, W. Y. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 564–573. Association for Computational Linguistics, 2017.

[76] Yan, Z., Ma, T., Gao, L., Tang, Z., and Chen, C. Cycle representation learning for inductive relation prediction. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24895–24910. PMLR, 2022. URL https://proceedings.mlr.press/v162/yan22a.html.

[77] Yanardag, P. and Vishwanathan, S. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1365–1374, 2015.

[78] Yang, B., Yih, W., He, X., Gao, J., and Deng, L. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*, 2015.

[79] Ye, R., Zhang, C., Wang, R., Xu, S., and Zhang, Y. Language is all a graph needs, 2024. URL https://arxiv.org/abs/2308.07134.

[80] Yu, Z., Liao, N., and Luo, S. Genti: Gpu-powered walk-based subgraph extraction for scalable representation learning on dynamic graphs. *Proceedings of the VLDB Endowment*, 17(9): 2269–2278, 2024.

[81] Zhang, M., Li, P., Xia, Y., Wang, K., and Jin, L. Labeling trick: A theory of using graph neural networks for multi-node representation learning. *CoRR*, abs/2010.16103, 2020. URL https://arxiv.org/abs/2010.16103.

[82] Zhang, Y. and Yao, Q. Knowledge graph reasoning with relational digraph. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pp. 912–924. ACM, 2022.

[83] Zhang, Y., Zhou, Z., Yao, Q., Chu, X., and Han, B. Adaprop: Learning adaptive propagation for graph neural network based knowledge graph reasoning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pp. 3446–3457. ACM, 2023. doi: 10.1145/3580305.3599404. URL https://doi.org/10.1145/3580305.3599404.

[84] Zhang, Y., Bevilacqua, B., Galkin, M., and Ribeiro, B. TRIX: A more expressive model for zero-shot domain transfer in knowledge graphs. *CoRR*, abs/2502.19512, 2025. doi: 10.48550/ARXIV.2502.19512. URL https://doi.org/10.48550/arXiv.2502.19512.

[85] Zhao, J., Mostafa, H., Galkin, M., Bronstein, M., Zhu, Z., and Tang, J. Graphany: A foundation model for node classification on any graph, 2024. URL https://arxiv.org/abs/2405.20445.

[86] Zhou, J., Bevilacqua, B., and Ribeiro, B. An ood multi-task perspective for link prediction with new relation types and nodes. *arXiv preprint arXiv:2307.06046*, 2023. URL https://doi.org/10.48550/arXiv.2307.06046.

[87] Zhu, Z., Zhang, Z., Xhonneux, L. A. C., and Tang, J. Neural bellman-ford networks: A general graph neural network framework for link prediction. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 29476–29490, 2021.

[88] Zhu, Z., Yuan, X., Galkin, M., Xhonneux, L. A. C., Zhang, M., Gazeau, M., and Tang, J. A*net: A scalable path-based reasoning approach for knowledge graphs. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[89] Zi, C., Zhao, H., Sun, X., Lin, Y., Cheng, H., and Li, J. Prog: A graph prompt learning benchmark. *arXiv preprint arXiv:2406.05346*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: To the best of our knowledge, the abstract and introduction clearly state the claims made.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations and point out the assumption in Appendix H and L.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We propose a novel KG reasoning foundation model in Section 5. We provide more details of the model architecture and implementation in Appendix B-E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The source code is accessible via the link provided in the Abstract. All datasets used in this study are publicly available, and their corresponding sources are appropriately cited within the manuscript.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided experimental settings and implementation details in the main paper and appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We reported error bars of the main experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided the information on compute resources in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, we have reviewed the NeurIPS Code of Ethics and ensured full compliance throughout our research process.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, we discuss the potential positive societal impacts and negative societal impacts in Section L.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All third-party assets are properly credited with original sources, and their licenses/terms are explicitly stated and adhered to in the paper and code.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, we communicate the details of the code as part of our submission. Our source code is anonymous.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We did not utilize crowdsourcing experiments and research with human subjects in this work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe the usage of LLMs in Section L.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A Notations and Definitions

The notations used in this paper and their descriptions are summarized in Table 4.

Table 4: Summary of the major notations in this paper.

Symbol	Description
$\mathcal{G}$	A knowledge graph (KG)
$\mathcal{T}$	The set of triples in a KG
$\mathcal{E}, \mathcal{R}$	The entity set and relation set in a KG
$\lvert\mathcal{T}\rvert, \lvert\mathcal{E}\rvert, \lvert\mathcal{R}\rvert$	The item number in a specific set
e, r	An entity (e) or a relation (r) in a KG
$\boldsymbol{q} = (e_q, r_q)$	A query with an entity e_q and a relation r_q
e_a	The ground-truth entity of a query
$\mathcal{X}$	node/entity feature matrix
d^0, d	Dimension of features and embeddings
$\mathcal{G}_{tr}, \mathcal{G}_{inf}$	The training KG and inference KG
$CMP()$	A conditional message passing module
$\boldsymbol{h}_v$	The representation of e_v conditioned on $\boldsymbol{q}$
$\mathbf{P}$	Relation-level interaction representations
$\mathbf{R}_g$	Query-independent relation representations
$\mathbf{R}_q$	Relation representations conditioned on $\boldsymbol{q}$
$\mathbf{H}_g$	Query-independent entity representations
$\mathbf{H}_q$	Entity representations conditioned on $\boldsymbol{q}$
$\textsc{Init}(\boldsymbol{q}, e_v)$	The initialization function of CMP
$\textsc{Msg}()$	The message passing function of CMP
$\textsc{Agg}(), \textsc{Upd}()$	The aggretation and update functions
$\mathcal{N}_r(e_v)$	The direct neighbors of e_v connecting via r
$p(\boldsymbol{q}, e_v)$	The plausibility score for the triple
$\tilde{\mathcal{X}}$	The unified feature matrix
$\mathcal{G}_r$	The relation graph corresponding to $\mathcal{G}$
$\mathcal{R}_{fund}$	The interaction types in the relation graph
$\mathcal{T}_r$	The triples in the relation graph
$\mathcal{S}_{e_v}$	The semantic neighbors of e_v
$\mathbb{1}(q = v)$	The indicator function
$\mathcal{L}$	The loss function of SCR
$\mathcal{F}$	The set of diverse features for training

B Tasks and Datasets

B.1 Task-Specific Datasets

Link-level KG Reasoning: We conduct inductive KG reasoning experiments on 24 datasets. Half of them are derived from the GraIL work [58], which are constructed from commonly-used KG benchmarks, including WN18RR [2], FB15k237 [59] and NELL-995 [75]. In these datasets, the train graphs and the test graphs share the same relation types. To evaluate performance in the full context of inductive reasoning, we also employ 12 datasets used in the InGram work [28]. The InGram datasets were derived from three real-world KG benchmarks: FB15k237 [59], Wikidata68K [21], and NELL-995 [75]. There are four datasets in each series with different proportions of triplets with new relations as 100%, 75%, 50%, and 25%. Note that, there are other KG datasets where

textual descriptions are not easily accessible; we leave the evaluation of such datasets for future work. Structural statistics for these datasets can be found in Table 10.

Node/Graph-level Classification: To evaluate the adaptability of our method across various types/domains of graph tasks, we conduct experiments on 14 datasets involving both node-level and graph-level classification tasks. Following prior studies [89], we employ six commonly-used node classification datasets, including homophilic graph datasets (Cora, Citeseer, PubMed) [49, 41], and heterophilic graph datasets (Wisconsin, Texas, Actor) [42, 56]. Furthermore, we considered eight graph classification datasets from various domains, such as social networks (IMDB-B, COLLAB) [77], biological datasets (ENZYMES, PROTEINS, DD) [11, 3, 69], and small molecule datasets (MUTAG, COX2, BZR) [27, 44]. More detailed information on these datasets can be found in Table 11.

Table 5: Statistics of pre-training KG datasets.

Dataset	$\lvert\mathcal{E}_{tr}\rvert$	$\lvert\mathcal{R}_{tr}\rvert$	$\lvert\mathcal{T}_{tr}\rvert$		
			#Train	#Validation	#Test
WN18RR	40.9k	11	86.8k	3.0k	3.1k
FB15k-237	14.5k	237	272.1k	17.5k	20.4k
CodexMedium	17.0k	51	185.5k	10.3k	10.3k

Table 6: Statistics of Bert-based text encoder.

Method	Dim	Model Name
Bert	768	bert-base-nli-mean-tokens
MPNet	768	all-mpnet-base-v2
MiniLM	384	paraphrase-MiniLM-L6-v2
DistilBert	512	distiluse-base-multilingual-cased-v1

B.2 Task-Specific Data Preparation

To evaluate the generalizability of SCR as a foundational graph reasoning engine, we reformulate various task data into a unified KG reasoning format, which includes a multi-relational graph structure and semantic node features. We illustrate the task-specific data forms in Figure 1.

Link-level KG Reasoning: For graph structure, KG triples can be directly represented as a multi-relational graph structure. For semantic features, we generate entity-level embeddings from available textual descriptions using BERT-based sentence encoders. To simulate different feature spaces, we employ four classical language models as sentence encoders, including BERT [10], MPNet [50], MiniLM [70], and DistilBERT [47]. MiniLM and DistilBERT have different embedding dimensions compared to BERT and MPNet. Details of textual encoders are provided in Table 6 in the Appendix.

Node-level Classification: For semantic features, we directly utilize the provided input node features. In terms of graph structure, we augment the original homogeneous graph by introducing "label classes" as distinct nodes. Edges with a "labeling" relation type are added to connect nodes with training labels to their corresponding class nodes. Consequently, the augmented graph contains two relation types and two entity types (nodes and labels). This approach eliminates the need to learn specific parameters for each class, enabling support for new nodes or labels through a zero-shot classification paradigm. Prior work [73, 74] adopted a similar format but in a homogeneous graph, lacking the relation distinguishability between original edges and labeling edges.

Graph-level Classification: Unlike node-level classification, graph classification tasks aim to predict the category of an entire graph. The graphs in the training set have no direct connections to the test graphs, which presents a challenge to the CMP reasoning process. To address task requirements, both the graph structure and the semantic feature format are specifically designed. First, we integrate all individual graphs into a single disconnected large graph, adding a "graph" node representing each graph, which connects to its corresponding nodes via a new relation type. Next, we perform global semantic encoding of SCR on this large graph to obtain the global representations of each graph node, and then capture semantically similar edges among graph nodes. This allows us to convert the task into a node-level task, where the global representations serve as semantic features and reasoning occurs on an augmented graph encompassing all graph nodes and labels.

C Semantic-Augmented Relation Graph

CMP-based models eliminate the necessity for learning unique embeddings for each entity. Instead, they depend on trainable relational representations to facilitate relation-specific message functions. To accommodate varied relational vocabularies in new KGs, recent research [20, 86, 18] emphasizes the importance of identifying the "invariance" present in the KG relational structure, thereby enabling any new relation type to be represented using a predefined set of parameters.

Drawing from insights in prior research [17], we construct a relation graph $\mathcal{G}_r = \{\mathcal{R}, \mathcal{R}_{fund}, \mathcal{T}_r\}$, where the nodes represent the relations in $\mathcal{G}$, and the edges $\mathcal{R}_{fund}$ capture four types of interactions between relations: "head-to-head", "tail-to-tail", "head-to-tail", and "tail-to-head". For instance, if two triples (e_1, r_1, e_2) and (e_2, r_2, e_3) are linked tail-to-head, an edge $(r_1, \text{“t-h”}, r_2)$ would be added to $\mathcal{T}_r$. Since these four interaction types are inherently derived from the triple structure in knowledge graphs, the pre-trained embeddings of these interaction types can be universally shared across KGs, allowing for the parameterization of any unseen relations.

In our SCR framework, we refine the relation graph by supplementing the original triple data $\mathcal{T}$ with additional edges obtained through semantic augmentation. Specifically, we derive semantic interactions among entities from the unified features $\tilde{\boldsymbol{X}}$. For each entity e_v, we identify the top k spatially nearest entities in the unified feature space via pairwise similarities, while excluding its direct topological neighbors. The set of semantic neighbors $\mathcal{S}_{e_v}$ is defined as follows:

$$\mathcal{S}_{e_v} = \{e_i \in \mathcal{E} \mid e_i \in f_s(\tilde{\mathcal{X}}, e_v, k, \delta) \wedge\ e_i \notin \mathcal{N}_{e_v}\}, \tag{12}$$

Here, $f_s(\cdot)$ represents the similarity function, $\mathcal{N}_{e_v} = \{e_i \in \mathcal{E} \mid (e_v, r, e_i) \in \mathcal{T}, r \in \mathcal{R}\}$ refers to the topological neighbor set of e_v. The hyperparameters k and δ refer to the number of neighbors and the similarity threshold, respectively. The semantic interaction between e_v and each element in $\mathcal{S}_{e_v}$ is regraded as an additional relation type r_s. Finally, the construction rules for the relation graph $\mathcal{G}_r$ can be formalized as follows:

$$\begin{aligned}
&\exists e_v \in \mathcal{E}, r_1, r_2 \in \mathcal{R}: \\
&\quad \exists\, e \in \tilde{\mathcal{N}}^{l_1}_{r_1} \cap \tilde{\mathcal{N}}^{l_2}_{r_2} \ \Rightarrow\ (r_1, l_1\text{-}l_2, r_2), (r_2, l_2\text{-}l_1, r_1) \in \mathcal{T}_r, \\
&\quad \exists\, e \in \mathcal{S}_{e_v} \cap \tilde{\mathcal{N}}^{l_1}_{r_1} \ \Rightarrow\ (r_1, l_1\text{-`t'}, r_s), (r_s, \text{`t'-}l_1, r_1) \in \mathcal{T}_r, \\
&\mathcal{S}_{e_v} \neq \emptyset \wedge e_v \in \tilde{\mathcal{N}}^{l_1}_{r_1} \ \Rightarrow\ (r_1, l_1\text{-`h'}, r_s), (r_s, \text{`h'-}l_1, r_1) \in \mathcal{T}_r,
\end{aligned}$$

where $l_i \in \{\text{`h'}, \text{`t'}\}$ denotes the side of the relation (head or tail), $\tilde{\mathcal{N}}^{l_i}_{r_i} \subset \mathcal{E}$ represents the set of entities connected to relation r_i on the l_i side. r_s is a newly introduced relation in the semantic space, and the final node set of $\mathcal{G}_r$ is equal to $(\mathcal{R} \cup \{r_s\})$.

In classification tasks, although there are a few relation types in the relation graph, it still follows strict topological rules. For example, label edges only link from nodes to their labels, while original edges connect among nodes. This creates meaningful asymmetric constraints, i.e. there are no "t-h" and "t-t" interactions from the label edge type to the original edge type. By learning how different relation interactions interact topologically across the graph, our model derives transferable reasoning capabilities that generalize beyond explicit relation semantics.

D Implementation Details

We introduce the statistics of pre-training KGs in Table 5. Following previous work [82, 87], we augment the triples in each $\mathcal{G}$ with reverse and identity relations. The augmented triple set $\mathcal{T}^+$ is defined as: $\mathcal{T}^+ = \mathcal{T} \cup \{(e_t, r^{-1}, e_h) | (e_h, r, e_t) \in \mathcal{T}\} \cup \{(e, r^i, e) | e \in \mathcal{E}\}$, where the relation r^{-1} is the reverse relation of a relation r, the relation r^i refers to the identity relation, and the number of augmented triples is $|\mathcal{T}^+| = 2|\mathcal{T}| + |\mathcal{E}|$.

We employ the ULTRA(3g) [17] model as the major baseline, utilizing the released checkpoint pre-trained on three knoglwedge graphs. We evaluate OpenGraph [74] and AnyGraph [73] across diverse node-level datasets using their publicly released pre-trained model weights. The metric results of GraphAny are referred to its official paper [85], and some node-level and graph-level results for graph prompt learning models are from the ProG work [89]. Although there are some previous KG reasoning baselines [16, 45, 58] with no semantic features involving in, we ignore them in our more challenging generalized reasoning tasks.

We train SCR with three commonly-used KG datasets, WN18RR [2], FB15k237 [59], and CodexM [46], following the hyperparameter settings of ULTRA(3g). Concerns about potential relation leakage during pre-training can be ignored because neither our method nor ULTRA learns relation-specific parameters. Specifically, the CMP module follows NBFNet [87] with a non-parametric DistMult [78] message function and a simplified PNA aggregation funcition [7], which leverages only two sub-aggregations: MEAN and STD. The number of layers L for both CMP and SCMP is set to 6, with the hidden dimension configured at 64. The relation encoder utilizes randomly initialized edge embeddings for $\mathcal{R}_{fund}$. In contrast, $SCMP(\cdot)$ initializes the embeddings of edge types using the relative relation embeddings $\mathbf{R}_q$. We suggest consulting the ULTRA paper [17] for further details.

We introduce two common types of node semantic features in knowledge graphs for model training.

- **Textual Embeddings** are vector representations of textual information associated with entities in a KG, typically generated using models like BERT [10] or Word2Vec [40]. Textual embeddings are broadly applicable across different KGs, as most KGs contain some form of textual metadata. However, the richness and variety of text data across KGs—such as short labels or multilingual content—can introduce diversity in how these embeddings are utilized, requiring models to generalize across various linguistic and domain-specific contexts.
- **Ontology Features** refer to structured representations of entities within a formalized schema, such as a $|R|$-length vector that counts the relation types associated with each entity. These features offer a global understanding of an entity's role in the graph by capturing its relational context. Common across various domains, they provide a simplified view of an entity's interactions. However, the diversity of relation types and their distribution can vary significantly across KGs, which affects how well these features generalize.

Hyperparameters are selected through grid search based on the metrics from the validation set. The similarity threshold δ and the number of neighbors k were not fine-tuned for individual datasets. Specifically, we set δ=0.9 for all node/graph-level datasets. For the neighbor number k, all six node-level datasets share k=2, while graph-level datasets use either k=1 or k=3, depending on their domain. SCR maintains robust performance across datasets with a single task-level configuration, preserving its zero-shot capability. All experiments are performed on Intel Xeon Gold 6238R CPU @ 2.20GHz and NVIDIA RTX A30 GPUs (four for pretraining and one for evaluation), and are implemented in Python using the PyTorch framework. Our source code is implemented based on ULTRA[2], which is available under the MIT License. All employed KG datasets are open and commonly used.

E Task-specific KG Structure

Node Classification Task: We first define the unified reasoning format from the view of node classification. Suppose $G = (V, E)$ is a graph in $\mathcal{D}$ whose nodes in V must be classified by labels in a finite label set $\mathcal{L}$. We construct the new, heterogeneous graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ as follows.

Let $\widetilde{V} = V \cup \mathcal{L}$ be the set of all nodes in the new graph, where each $\ell \in \mathcal{L}$ is viewed as a distinct "label node". Retain every original edge from E in $\widetilde{G}$, so that if $(v, u) \in E$ in the original graph G, the same (possibly typed) edge is preserved in $\widetilde{G}$. Furthermore, introduce a designated relation type r_{label} connecting nodes $v \in V$ to label nodes $\ell \in \mathcal{L}$, i.e.:

$$E_{\text{label}} = \big\{ (v,\, r_{\text{label}},\, \ell) \;\mid\; v \in V,\, \ell \in \mathcal{L}, \\ \text{and } v \text{ is assigned training label } \ell \text{ in } \mathcal{D} \big\}. \tag{13}$$

Then let the edge set $\widetilde{E} = E \cup E_{\text{label}}$. This completes the construction of the heterogeneous graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$.

Node classification in G bijectively map to link prediction in $\widetilde{G}$, observe that assigning a label ℓ to a node v in the original problem becomes the presence of an edge (v, r_{label}, ℓ) in $\widetilde{G}$. First, assume a labeling function $f : V \to 2^{\mathcal{L}}$ is given. Each instance $v \mapsto \ell$ that appears in f corresponds to including $(v,\, r_{\text{label}},\, \ell) \in E_{\text{label}}$. Hence the labeling of G completely specifies the set of label-links in $\widetilde{G}$. Conversely, given a link $(v,\, r_{\text{label}},\, \ell) \in \widetilde{E}$, one uniquely recovers the statement that v is labeled

[2]https://github.com/DeepGraphLearning/ULTRA

by ℓ in the original classification problem. This one-to-one correspondence implies that any function assigning labels to nodes in G bijectively maps to a set of label-links in $\widetilde{G}$.

Graph Classification Task: Let $\mathcal{D} = \{G_1, G_2, \ldots, G_M\}$ be a collection of graphs, where each G_i is to be assigned one or more labels from a finite label set $\mathcal{L} = \{\ell_1, \ldots, \ell_K\}$. Suppose each graph G_i in $\mathcal{D}$ has node set V_i and edge set E_i. We aim to label each G_i with one or more labels from $\mathcal{L}$. An extended graph $\widetilde{G}$ is constructed as follows.

Step 1: Introduce a new *graph node* s_i for each graph G_i, which will represent the entire graph G_i as a single entity in $\widetilde{G}$.

Step 2: Include in $\widetilde{G}$ all original nodes from each G_i. That is, take $V_{\text{all}} = \bigcup_{i=1}^{M} V_i$, and add these re-indexed nodes to $\widetilde{G}$ along with their internal edges $E_{\text{all}} = \bigcup_{i=1}^{M} E_i$.

Step 3: For each $v \in V_i$, add an edge $(s_i,\, r_{\text{node-graph}},\, v)$ to indicate that v is a member of the graph G_i. The relation $r_{\text{node-graph}}$ is a designated edge type (e.g., "belongsToGraph").

Step 4: For labeling, add a node for each label $\ell \in \mathcal{L}$. Let these form the set of *label nodes* in $\widetilde{G}$. To encode the classification of G_i with label ℓ, add an edge $(s_i,\, r_{\text{label}},\, \ell)$ whenever G_i is assigned label ℓ. Let E_{label} be the set of all such edges.

Step 5: For edges between graph nodes, add an edge $\big(s_i,\, r_{\text{similar}},\, s_j\big)$ whenever the graph embedding vector of G_i is close to that of G_j. Let E_{similar} be the set of all such edges.

Hence, the heterogeneous graph is merged as follows:

$$\begin{aligned}\widetilde{G} \;=\; \big(\, & V_{\text{all}} \cup \{s_1, \ldots, s_M\} \cup \mathcal{L}, \\ & E_{\text{all}} \cup E_{\text{node-graph}} \cup E_{\text{label}} \cup E_{\text{similar}}\big).\end{aligned} \tag{14}$$

Similar to the claims about node classification, labeling G_i with ℓ in the original problem is exactly equivalent to the statement that $\big(s_i,\, r_{\text{label}},\, \ell\big)$ is an edge in $\widetilde{G}$. Because every valid assignment $G_i \mapsto \ell$ corresponds bijectively to an edge $\big(s_i,\, r_{\text{label}},\, \ell\big)$.

F Expressive Power

We formally analyze the expressive power of SCR by comparing it with ULTRA [17]. Following the theory of the Weisfeiler-Leman test, we measure expressivity via a method's ability to distinguish non-isomorphic subgraphs in knowledge graphs.

Firstly, we show that SCR is at least as expressive as ULTRA. For any non-isomorphic graphs distinguishable by ULTRA, there exists a parameter configuration of SCR that achieves identical distinguishability.

We establish this through architectural reduction. Let $\theta = (k, W_{\text{MLP}})$ denote SCR's key hyperparameters where k controls semantic neighborhood size and W_{MLP} the MLP weights from Eq. (7). When $\theta_0 = (0, \mathbf{0})$, SCR reduces to ULTRA through three key simplifications:

1. The augmented relation graph $\mathcal{G}_r$ collapses to ULTRA's original structure by removing the semantic relation type r_s;
2. The INIT function (Eq. (5)) reduces to ULTRA's initialization by eliminating the v_a term;
3. The representation fusion becomes identity because $\text{MLP}(\mathbf{H}_g) = 0$;

Under θ_0, the message passing dynamics of both architectures become isomorphic. Therefore, ULTRA constitutes a proper subspace of SCR's parameter space.

Secondly, we indicate that there exists a class of non-isomorphic triples distinguishable by SCR but not by ULTRA, provided that semantic features contain discriminative information beyond graph topology.

Consider two candidate entities $e_1, e_2 \in \mathcal{E}$ with identical topological signatures relative to query entity e_q:

$$\forall p \in \Pi_{\text{path}} : f_{\text{ULTRA}}(e_q, p, e_1) = f_{\text{ULTRA}}(e_q, p, e_2) \tag{15}$$

where Π_{path} denotes relational paths and f the path encoding function.

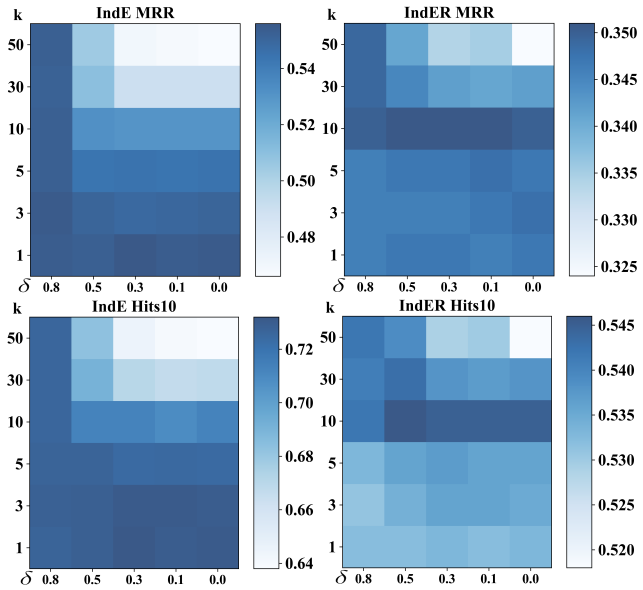


Figure 4: Comparison of performance metrics under different hyperparameter settings (k and δ) for semantic neighbor selection. (a) Average MRR results. (b) Hits@10 results. Darker colors indicate higher values.

ULTRA cannot distinguish (e_q, r_q, e_1) from (e_q, r_q, e_2) since their topological embeddings coincide. However, if the semantic features satisfy:

$$\min_{v \in \mathcal{N}_{r_q}(e_q)} \|\tilde{x}_{e_1} - \tilde{x}_v\|_2 \ll \min_{v \in \mathcal{N}_{r_q}(e_q)} \|\tilde{x}_{e_2} - \tilde{x}_v\|_2 \tag{16}$$

where $\mathcal{N}_{r_q}(e_q)$ are r_q-neighbors of e_q, then SCR's semantic proximity measure induces divergent embeddings:

$$\|h_{e_1}^{(L)} - h_{e_2}^{(L)}\|_2 \geq \gamma > 0 \tag{17}$$

The separation constant γ persists through MLP fusion (Eq. (7)) by the Lipschitz continuity of neural networks. Thus, SCR distinguishes the triplets while ULTRA cannot.

This analysis reveals SCR's enhanced expressiveness stems from its *semantic-topological fusion* mechanism. By jointly optimizing structural and semantic proximity measures during pre-training, the model learns disentangled yet complementary representations that strictly subsume purely topological approaches like ULTRA.

G Hyperparameter Sensitivity (RQ5)

Furthermore, we evaluate the impact of hyperparameters k and δ on the selection of semantic neighbors. Here, k specifies the number of neighbors, while δ determines the minimum similarity threshold. As illustrated in Figure 4, variations in these hyperparameters slightly affect prediction performance. Different choices of k show similar performance across the IndE benchmarks, but for IndER datasets, $k = 10$ clearly outperforms other values. When $k > 10$, the influence of δ becomes more pronounced, as a lower δ includes more dissimilar neighbors, negatively affecting model performance.

H Assumption in SCR

In Section 4.2, we claim that the target node distinguishability assumption holds for the $INIT^2()$ function, if $\mathbf{r_q} \neq \mathbf{v_a}$ and neither of them contains zero entries. This assumption usually holds because of the distinction between $\mathbf{r_q}$ (relation representations outputted by CMP_ϕ) and $\mathbf{v_a}$ (trainable vector parameters) is inherently preserved through their distinct initialization and optimization mechanisms.

Empirically, we conducted parameter analysis on our pre-trained SCR model and observed that: All $\mathbf{v_a}$ parameters maintained non-zero magnitudes (> 0.5511); $\mathbf{r_q}$ outputs on six datasets showed no exact zero entries or equality to $\mathbf{v_a}$. While theoretical equality is possible, it would require exact parameter convergence to zero or identical gradient updates—statistically implausible in practice.

I Complexity and Scalability Analysis

Given a knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, we have that $|\mathcal{E}|, |\mathcal{R}|, |\mathcal{T}|$ represent the size of entities, relation types, and triples, respectively. $\mathcal{G}_r = (\mathcal{R}, \mathcal{R}_{fund}, \mathcal{T}_r)$ denotes the relation graph of $\mathcal{G}$. $\mathcal{X} \in \mathbb{R}^{|\mathcal{E}| \times d_0}$ is the input feature matrix, d is the hidden dimension of the model, and L is the number of layers in the model. In the preprocessing stage, unifying semantic features requires a time complexity of $\mathcal{O}(|\mathcal{E}|dd_0)$ for the SVD low-rank approximation. Our adoption of "torch.svd_lowrank(X, q=d, niter=2)" leverages randomized truncated SVD, avoiding the $O(|E|^3)$ complexity of the full SVD. Constructing relation graphs involves extracting the top K most similar neighbors for each entity, with a time complexity of $\mathcal{O}(|\mathcal{E}|^2(d + \log K))$, which simplifies to $\mathcal{O}(|\mathcal{E}|^2 d)$ as $d \gg \log K$. Therefore, the overall complexity is $\mathcal{O}((d_0 + |\mathcal{E}|)|\mathcal{E}|d)$.

In terms of the CMP module, as shown in Zhu et al. [87], the time complexity for a forward pass on $\mathcal{G}$ is $\mathcal{O}(L(|\mathcal{T}|d + |\mathcal{E}|d^2))$ to compute one query reasoning. The runtimes of CMP and SCMP are comparable because SCMP's global encoding is shared across all queries, resulting in only a linear overhead. Combining the CMP calculations on $\mathcal{G}_r$, the total complexity is $\mathcal{O}((|\mathcal{T}| + |\mathcal{T}_r|)Ld + (|\mathcal{E}| + |\mathcal{R}|)Ld^2)$. Because one CMP-based reasoning calculates $|\mathcal{E}|$ candidate triples at the same time, resulting in an amortized complexity is better than traditional relational message-passing models, such as RGCN [48], CompGCN [60], and GraIL [58].

Regarding scalability, SCR exhibits comparable scalability and running time to ULTRA. To ease the concern, we conducted additional experiments in Appendix M on all transductive KG datasets mentioned in ULTRA, including large-scale datasets with over 100k triples, such as YAGO310. We acknowledge that both methods face practical limitations when applied to KGs with millions or billions of triples. Specifically, the time cost of subgraph extraction and message passing becomes non-trivial compared to traditional embedding-based models. This limitation is inherent to subgraph-based inductive frameworks but does not preclude SCR 's applicability to typical large-scale KGs. We will prioritize this in future work. For large-scale KGs, recent acceleration techniques like TIGER [65] (enabling efficient subgraph extraction for inductive reasoning on Freebase) are critical. While SCR 's current implementation does not yet integrate these optimizations, its framework is compatible with such methods.

J Impact of Feature Dimension

In terms of the impact of the dimension gap between semantic features and node features, we conduct additional experiments with 50-dimensional Glove Features (with zero-padding), Clipped 64-dimensional Bert features, random features, and all-ones features. The results in Table 7 indicate that using node features with closer dimensions still invokes the performance drop. Additionally, we test the variant using the semantic features of the query node in $INIT^3()$ in Huang et al. [25]. There still exists a performance drop compared with the original ULTRA on most datasets. These results indicate that the semantic isolation issue still holds when setting the dimension of the node features closer to the dimension of the semantic features and using $INIT^3$ fusion.

K Impact of Long-range Dependency

It is worth noting that unlike conventional GNNs, Graph Foundation Models are still in their early stages, and many associated challenges remain open. Over-smoothing and over-squashing are known issues in message-passing-based models, especially in long-range dependency tasks.

Over-smoothing typically occurs in deep architectures, where repeated message passing leads to uniform node embeddings. In our model, however, message propagation occurs over relatively shallow subgraphs (3-6 layers), which mitigates this risk. Furthermore, our message passing is source-conditioned: only the source node e_q is initialized with a non-zero embedding, and information

Table 7: Performance of ULTRA with different semantic features.

	FB_v1	WN_v1	NE_v1
ULTRA	0.486	0.593	0.716
+Bert	0.163	0.014	0.580
+All One(64d)	0.227	0.024	0.684
+Random(64d)	0.218	0.015	0.658
+Bert(Clip64)	0.200	0.013	0.593
+Glove(50d)	0.160	0.007	0.609
+Bert+INIT3	0.483	0.549	0.648
+Glove+INIT3	0.483	0.524	0.720

Table 8: Performance on the long-range PascalVOC-SP dataset.

Method	Macro F1
ULTRA (full labeling)	0.039
SCR (full labeling)	0.053
GCN (full training)	0.101
GraphTransformer(full training)	0.121
SCR (20% labeling)	0.051
GCN (20% training)	0.046
GraphTransformer(20% training)	0.052

flows outward. Thus, each node's embedding is a view from e_q, not a globally shared representation, preserving diversity. Over-squashing, caused by bottlenecks in aggregating long-range messages into fixed-size vectors, may arise due to dense multi-hop paths. To address this, we employ the expressive PNA aggregator and set the hidden dimension to 64, balancing representational capacity with computational efficiency, while also accommodating both BERT-based textual features and common graph-domain features.

To further validate SCR's performance on deep or extreme-range tasks, we employed a subset of PascalVOC-SP (685K nodes, 5M edges) in the LRGB benchmark [13]. While LRGB assumes fully-supervised training on very large graphs, our proposed SCR was designed for zero-shot generalization. SCR only saw the label information during inference but no any fine-tuning on them. As shown in Table 8, SCR delivers a substantial gain over ULTRA, but under full supervision, it still trails GCN and GraphTransformer. We believe that large-scale training enhances the performance of GCN and GraphTransformer. To mitigate the effect of training, we use a few-shot setting. GCN and GraphTransformer are trained on 20% samples, while SCR remains strictly zero-shot, merely seeing those labels at inference. Under this setting, SCR matches GraphTransformer and outperforms GCN, demonstrating that it retains long-range ability even without task-specific training.

While these are well-known challenges, our current focus is to investigate the semantic and structural transferability of KG pre-training to diverse graph tasks—a core goal in the emerging Graph Foundation Model paradigm. Our work takes a step forward by focusing on cross-task and cross-graph transferability, which we believe is largely orthogonal to the over-smoothing, over-squashing, and long-range issues. We consider these important directions for future work.

L Limitations

Here, we discuss two limitations of this work. First, the training data for SCR is confined to three popular KGs. There is room for improvement by training with more diverse graphs, particularly on challenging tasks like those involving heterophilic graphs. Second, the scalability of SCR on larger-scale graphs remains to be verified in future work. The expectation is positive, given the recent efforts focused on accelerating KG reasoning through system and algorithmic optimizations [83, 88, 65]. Third, this paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here. In this work, the LLM is used only for writing and editing.

Table 9: Per-dataset and average performance of ULTRA and SCR on 54 KG datasets.

KG Datasets	ULTRA(3g)		SCR(3g)	
	MRR	Hits@10	MRR	Hits@10
WikiTopicsMT1:tax	0.242	0.305	0.182	0.312
WikiTopicsMT1:health	0.279	0.332	0.265	0.410
WikiTopicsMT2:org	0.083	0.145	0.078	0.139
WikiTopicsMT2:sci	0.258	0.348	0.245	0.354
WikiTopicsMT3:art	0.251	0.414	0.244	0.407
WikiTopicsMT3:infra	0.622	0.779	0.635	0.781
WikiTopicsMT4:sci	0.293	0.455	0.256	0.461
WikiTopicsMT4:health	0.557	0.707	0.615	0.753
Metafam	0.330	0.821	0.246	0.560
FBNELL	0.473	0.653	0.480	0.676
ILPC2022:small	0.296	0.441	0.285	0.443
ILPC2022:large	0.297	0.423	0.285	0.419
HM:1k	0.079	0.150	0.055	0.097
HM:3k	0.063	0.120	0.047	0.083
HM:5k	0.055	0.101	0.041	0.072
HM:indigo	0.436	0.649	0.425	0.631
YAGO310	0.480	0.658	0.488	0.666
NELL995	0.437	0.575	0.456	0.608
CoDExSmall	0.472	0.668	0.436	0.653
CoDExLarge	0.333	0.461	0.329	0.458
Hetionet	0.261	0.382	0.289	0.402
ConceptNet100k	0.061	0.117	0.115	0.218
DBpedia100k	0.397	0.565	0.401	0.573
AristoV4	0.183	0.262	0.227	0.349
WDsinger	0.370	0.488	0.371	0.498
NELL23k	0.241	0.406	0.234	0.402
FB15k237_10	0.159	0.272	0.155	0.265
FB15k237_20	0.183	0.309	0.179	0.305
FB15k237_50	0.230	0.396	0.222	0.389
FB15k237	0.369	0.562	0.344	0.533
WN18RR	0.369	0.533	0.444	0.571
CoDExMedium	0.374	0.527	0.350	0.498
Inductive e,r (23 graphs)	0.342	0.510	0.338	0.516
Inductive e (18 graphs)	0.416	0.568	0.433	0.582
Transductive (13 graphs)	0.293	0.428	0.300	0.445
Total AVG (54 graphs)	0.355	0.510	0.361	0.521
Pretraining (3 graphs)	0.371	0.541	0.379	0.534

M Evaluation on More KG Datasets

We rigorously test on all 54 datasets and the 3 pre-training graphs referenced in ULTRA's framework [17]. The results shown in Table 9 confirm that SCR outperforms ULTRA across the majority of metrics when evaluated under identical conditions (PyG-based implementation and pre-training data).

N Related Work

N.1 Knowledge Graph Reasoning

Traditional *transductive* KG reasoning models, such as TransE [1], DistMult [78], RotatE [54], RGCN [48], and CompGCN [60], represent entities and relations within a knowledge graph using continuous vector embeddings [68, 64]. These models, however, assume that all entities and relations in the KG are known beforehand, which limits their ability to generalize to unseen entities within the same graph or across different KGs [63, 4]. In contrast, *inductive* KG reasoning approaches [87] address this limitation by enabling generalization to KGs with previously unseen entities or relations. Most existing inductive methods [76, 62, 32, 5] employ query-conditional MPNNs to generate "relative" entity embeddings by extracting local structural features from a subgraph induced by the

Table 10: Inductive KG datasets used in the experiments. "Triples" refers to the number of edges in the graph used for training, validation, or testing. "Valid" and "Test" refer to the triples that need to be predicted in the validation and test sets, respectively, within the corresponding graphs.

Group	Dataset	Training Graph			Validation Graph				Test Graph			
		Entities	Rels	Triples	Entities	Rels	Triples	Valid	Entities	Rels	Triples	Test
IndE(FB)	FB:v1 [58]	1594	180	4245	1594	180	4245	489	1093	180	1993	411
	FB:v2 [58]	2608	200	9739	2608	200	9739	1166	1660	200	4145	947
	FB:v3 [58]	3668	215	17986	3668	215	17986	2194	2501	215	7406	1731
	FB:v4 [58]	4707	219	27203	4707	219	27203	3352	3051	219	11714	2840
IndE(WN)	WN:v1 [58]	2746	9	5410	2746	9	5410	630	922	9	1618	373
	WN:v2 [58]	6954	10	15262	6954	10	15262	1838	2757	10	4011	852
	WN:v3 [58]	12078	11	25901	12078	11	25901	3097	5084	11	6327	1143
	WN:v4 [58]	3861	9	7940	3861	9	7940	934	7084	9	12334	2823
IndE(NL)	NL:v1 [58]	3103	14	4687	3103	14	4687	414	225	14	833	201
	NL:v2 [58]	2564	88	8219	2564	88	8219	922	2086	88	4586	935
	NL:v3 [58]	4647	142	16393	4647	142	16393	1851	3566	142	8048	1620
	NL:v4 [58]	2092	76	7546	2092	76	7546	876	2795	76	7073	1447
IndER(FB)	FB-25 [28]	5190	163	91571	4097	216	17147	5716	4097	216	17147	5716
	FB-50 [28]	5190	153	85375	4445	205	11636	3879	4445	205	11636	3879
	FB-75 [28]	4659	134	62809	2792	186	9316	3106	2792	186	9316	3106
	FB-100 [28]	4659	134	62809	2624	77	6987	2329	2624	77	6987	2329
IndER(WK)	WK-25 [28]	12659	47	41873	3228	74	3391	1130	3228	74	3391	1131
	WK-50 [28]	12022	72	82481	9328	93	9672	3224	9328	93	9672	3225
	WK-75 [28]	6853	52	28741	2722	65	3430	1143	2722	65	3430	1144
	WK-100 [28]	9784	67	49875	12136	37	13487	4496	12136	37	13487	4496
IndER(NL)	NL-25 [28]	4396	106	17578	2146	120	2230	743	2146	120	2230	744
	NL-50 [28]	4396	106	17578	2335	119	2576	859	2335	119	2576	859
	NL-75 [28]	2607	96	11058	1578	116	1818	606	1578	116	1818	607
	NL-100 [28]	1258	55	7832	1709	53	2378	793	1709	53	2378	793

Table 11: Statistics of node/graph classification datasets.

Dataset	Graphs	Nodes	Edges	Feature Dims	Classes	Node-level Task
Cora	1	2,708	5,429	1,433	7	Homophilic Node Classification
CiteSeer	1	3,327	9,104	3,703	6	Homophilic Node Classification
Pubmed	1	19,717	88,648	500	3	Homophilic Node Classification
Actor	1	7600	30,019	932	5	Heterophilic Node Classification
Wisconsin	1	251	515	1,703	5	Heterophilic Node Classification
Texas	1	183	325	1703	5	Heterophilic Node Classification
Dataset	**Graphs**	**Avg.nodes**	**Avg.edges**	**Feature Dims**	**Classes**	**Graph-level Task**
IMDB-BINARY	1,000	19.8	96.53	0	2	Social Network Classification
COLLAB	5,000	74.5	2457.8	0	3	Social Network Classification
PROTEINS	1,113	39.1	72.8	3	2	Protein Graph Classification
ENZYMES	600	32.6	62.1	3	6	Protein Graph Classification
DD	1,178	284.1	715.7	89	2	Protein Graph Classification
MUTAG	188	17.9	19.8	7	2	Small Molecule Classification
COX2	467	41.2	43.5	3	2	Small Molecule Classification
BZR	405	35.8	38.4	3	2	Small Molecule Classification

query entity. GraIL [58], for example, extracts an enclosing subgraph between the query entity and each candidate entity, but this approach suffers from high computational costs. Other models, such as NBFNet [87] and RED-GNN [82], propagate query features through the L-hop neighborhood subgraph of the query entity. To improve computational efficiency, recent works have focused on optimizing algorithms, including path-pruning techniques during the GNN propagation process [83, 88, 67]. In the direction of building a foundation model for KG reasoning, ULTRA [17] utilizes four basic interaction types of the KG relational structure to perform inductive reasoning on entirely novel KGs. Building on ULTRA, ProLINK [66] harnesses the power of large language models (LLMs) to enhance reasoning performance for few-shot relation types on low-resource KGs. Nevertheless, these inductive methods face challenges in generalizing to a wide variety of graph tasks and feature spaces, as their reasoning capabilities remain primarily confined to topological structures.

N.2 Graph Foundation Models

Graph foundation models are increasingly recognized for their ability to manage diverse graph-structured data across various tasks. Traditionally, model fine-tuning offers a simple method to adapt

these models to downstream tasks [80, 31, 30]. However, significant discrepancies between tasks can lead to negative transfer and catastrophic forgetting [43]. An alternative to fine-tuning is graph prompt learning, which reformulates input graph data to better align with the pretext task [6, 55, 22]. In this context, GPF utilizes a prompt token by adding supplementary features to the base graph. Building on this, GPF-plus [15] trains multiple independent basis vectors and integrates them through attentive aggregation facilitated by several learnable linear projections. GPPT [52] introduces graph prompts as additional tokens comprising task-specific and structural elements, aiding in node tasks and link prediction pretext alignment. Gprompt [38] incorporates prompt vectors into graph pooling through element-wise multiplication. Other research considers graph prompts as additional graphs [19, 24]. The All-in-one model [53], for instance, integrates token graphs as prompts within the original graph, linking tokens directly with the original graph elements. Although these graph prompt learning methods achieve good performance in various graph tasks, they require an additional learning phase for task-specific parameters.

Recently, several foundational models for specific graph tasks have been proposed to adapt to diverse unseen data without model tuning. ULTRA [17] and KG-ICL [8] are pre-trained on multiple KGs to obtain the capability of reasoning on new KGs. ProLINK [66] and TRIX [84] further expand on ULTRA with prompt graphs from LLMs and iterative updates of relation/entity embeddings. GraphAny [85] addresses inductive node classification by formulating inference as an analytical solution to a linear GNN architecture, while an attention module fuses predictions from multiple models, ensuring scalability. Models like InstructGLM [79] and HiGPT [57] leverage large language models (LLMs), using natural language prompts to guide graph learning and handling heterogeneous graphs without downstream fine-tuning, broadening the applicability of foundation models to diverse graph tasks. Building a general graph foundation model is not trivial; the major challenges to overcome are related to structural and feature heterogeneity. OpenGraph [74] proposes a zero-shot graph learning framework with a unified graph tokenizer and a scalable graph transformer, allowing the model to handle unseen graph data, aided by LLM-based data augmentation. AnyGraph [73] extends this by addressing structural and feature heterogeneity through a Graph Mixture-of-Experts architecture, supporting fast adaptation and scaling efficiently. RiemannGFM [51] incorporates diverse vocabulary geometries via a novel product bundle and learns structural representations in Riemannian manifolds through stacked Riemannian layers, enabling cross-domain transferability. OMOG [37] trains dataset-specific expert models and dynamically integrates them via adaptive gating functions for unseen graphs, optimizing prior knowledge transfer while suppressing negative interference. Research on Graph Foundation Models is still in its early stages, and current methods often struggle to match the competitive performance of task-specific supervised methods [71, 36, 14, 34].

Several recent methods also incorporate semantic features into the graph reasoning framework. The initializer $INIT^3()$ in Huang et al. [25] incorporates the semantic feature of the query node u, but ignores the semantics of other nodes in the neighborhood. As a result, using $INIT^3()$ would not significantly drop the performance of ULTRA, because it still cannot exploit the full range of node semantics. This is also one of the motivations we propose the semantic isolation issue. Liu [35] employs PLM-based textual embeddings as the input node feature of GNN, and alternates the training of GNN and PLM within a single dataset. This method cannot handle our zero-shot GFM settings, especially for non-textual semantic features in general graphs. Additionally, the performance drop reported in their experiments when directly using pre-trained PLM embeddings indicates the necessity of re-training a PLM for enhanced text representation. A contemporaneous work [12] achieves semantic injection for KG foundational reasoning, but it focuses solely on text embeddings derived from a single semantic space (i.e., an LLM). Consequently, their pretraining approach cannot process zero-shot features originating from different semantic spaces. We note that their late fusion between query-conditioned structural encoding and global structural semantic encoding is similar to our proposed strategy, indirectly confirming the feasibility and rationale of our design choice. In summary, these recent methods cannot effectively integrate multi-source zero-shot features. Consequently, none of these approaches provides a viable alternative to our method within GFM.

O Additional Experimental Results

Table 12: Accuracy results on node classification datasets where 90% samples are divided into the test set.

Learning Paradigm	Methods	Cora*	Citeseer*	Pubmed*	Wisconsin*	Texas*	Actor*
One-Shot Training	GCN	26.56±5.55	21.78±7.32	39.37±16.34	41.60±3.10	37.97±5.80	20.57±4.47
	Pre-train & Fine-Tune	40.40±4.66	35.05±4.37	46.74±14.09	40.69±4.13	47.66±2.37	20.74±4.12
Graph Pre-Training One-Shot Tuning	GPPT	43.15±9.44	37.26±6.17	48.31±17.72	30.40±6.81	31.81±15.33	22.58±1.97
	All-in-one	52.39±10.17	40.41±2.80	45.17±6.45	78.24±16.68	65.49±7.06	24.61±2.80
	Gprompt	56.66±11.22	53.21±10.94	39.74±15.35	83.80±2.44	33.25±40.11	25.26±1.10
	GPF	38.57±5.41	31.16±8.05	49.99±8.86	88.67±5.78	87.40±3.40	28.70±3.35
	GPF-plus	55.77±10.30	59.67±11.87	46.64±18.97	**91.03±4.11**	**95.83±4.19**	29.32±8.56
KG Pre-Training No Tuning	**SCR (3g)**	76.18±0.09	50.40±0.08	72.76±0.14	46.67±0.28	54.15±0.24	23.52±0.14
	SCR-20%	58.81±2.74	36.62±1.62	67.94±0.56	45.42±1.47	52.68±0.83	21.81±0.96
	SCR-5	56.80±3.55	32.16±3.85	51.63±5.46	45.60±2.06	56.59±0.71	20.68±1.09

Table 13: F1 results on node classification datasets where 90% samples are divided into the test set.

Learning Paradigm	Methods	Cora*	Citeseer*	Pubmed*	Wisconsin*	Texas*	Actor*
One-Shot Training	GCN	16.60±2.54	10.81±4.90	37.23±15.48	26.34±4.01	24.05±5.12	11.56±3.08
	Pre-train & Fine-Tune	35.92±4.06	30.78±3.91	41.03±13.36	27.43±4.47	29.53±6.44	15.91±0.98
Graph Pre-Training One-Shot Tuning	GPPT	38.99±8.32	33.00±6.49	46.43±16.73	23.74±5.95	25.64±8.12	19.62±0.56
	All-in-one	46.58±8.42	30.20±4.44	38.05±6.24	67.68±10.36	43.37±16.01	16.05±3.88
	GPrompt	46.28±8.46	49.65±11.42	39.46±15.97	77.03±6.40	29.20±35.62	22.00±1.74
	GPF	23.79±5.49	18.63±7.34	45.36±15.88	82.97±6.10	78.43±9.49	31.69±5.47
	GPF-plus	53.28±11.46	56.22±13.99	42.38±19.01	85.24±5.45	86.22±10.29	24.56±8.79
KG Pre-Training No Tuning	**SCR (3g)**	73.06±0.12	50.02±0.05	70.79±0.18	17.82±0.05	17.51±0.49	19.64±0.15
	SCR-20%	55.81±3.87	36.27±3.43	64.92±0.90	17.14±4.06	16.05±2.91	19.14±0.52
	SCR-5	55.56±2.29	28.10±3.12	47.19±6.23	27.41±2.26	23.73±0.37	18.74±0.64

Table 14: F1 performance on node classification datasets.

	Methods	Cora	Citeseer	Pubmed	Wisconsin	Texas	Actor
One-shot Training	GCN	16.60±2.54	10.81±4.90	37.23±15.48	26.34±4.01	24.05±5.12	11.56±3.08
	Pre-train & Fine-tune	35.92±4.06	30.78±3.91	41.03±13.36	26.74±3.28	29.53±6.44	15.91±0.98
Graph Pre-Training No Tuning	OpenGraph	79.85±0.71	67.52±0.75	77.74±1.65	15.45±3.00	17.78±5.07	9.84±2.66
	AnyGraph (Link1)	60.5±5.28	49.81±5.18	58.44±4.28	1.33±0.34	0.40±0.19	4.98±0.27
	AnyGraph (Link2)	68.5±3.16	43.47±3.34	75.91±1.54	1.27±0.22	0.68±0.47	4.93±0.31
KG Pre-Training No Tuning	ULTRA(3g)	78.40±0.00	64.68±0.00	76.15±0.00	25.71±0.00	19.81±0.00	14.62±0.00
	SCR (3g)	80.92±0.61	69.24±1.10	77.91±1.31	29.03±3.66	28.73±1.59	20.29±0.41
Few-Shot Labeling	SCR-20%	71.98±2.95	53.48±2.34	69.09±0.34	23.62±1.30	26.74±7.10	19.45±0.57
	SCR-5	54.08±1.90	32.35±2.71	47.19±4.45	19.76±2.84	37.29±6.94	18.19±0.91

Table 15: F1 performance on graph classification datasets.

	Methods	IMDB-B	COLLAB	PROTEINS	MUTAG	ENZYMES	COX2	BZR	DD
One-Shot Training	GCN	54.62±1.12	41.10±0.39	46.69±10.82	63.47±6.36	15.25±3.96	22.78±10.69	23.71±8.23	44.74±4.23
	Pre-train & Fine-tune	55.24±1.07	41.71±0.17	59.73±1.34	63.70±5.32	19.17±3.42	45.06±1.93	33.12±7.45	48.68±6.42
Graph Pre-Training One-Shot Tuning	GPPT	44.16±6.70	42.87±7.70	47.07±11.95	53.15±16.82	19.87±2.99	44.68±1.17	49.40±8.41	51.50±6.54
	All-in-one	56.88±0.80	47.78±0.10	64.68±5.35	78.57±4.92	19.66±3.11	49.62±10.42	62.11±7.06	56.70±1.89
	GPrompt	52.10±13.61	43.35±10.75	58.30±10.88	71.38±3.64	19.52±3.36	46.26±5.14	44.81±6.73	52.80±3.60
	GPF	56.22±6.17	38.14±0.44	57.01±5.79	63.90±4.05	17.34±2.45	43.08±4.88	48.83±5.30	48.52±7.11
	GPF-plus	55.55±2.03	41.24±0.31	57.58±7.28	63.20±5.31	18.39±2.76	30.90±11.56	46.57±4.62	46.24±4.86
KG Pre-Training No Tuning	ULTRA(3g)	38.87±0.00	23.04±0.00	37.48±0.00	38.78±0.00	5.84±0.00	43.74±0.00	44.23±0.00	37.05±0.00
	SCR	60.91±2.18	57.71±1.82	65.23±1.37	84.23±1.90	21.77±2.17	49.24±3.55	51.09±8.61	69.85±0.51
Few-Shot Labeling	SCR-20%	49.04±7.20	46.35±4.28	57.48±11.12	34.01±6.45	9.38±1.49	45.80±3.41	45.39±2.31	68.85±2.62
	SCR-5	51.29±4.41	46.67±8.78	57.92±12.11	79.33±5.38	21.56±1.18	51.06±0.86	40.2±6.46	70.27±4.51

Table 16: Per-dataset results of performance on zero-shot KG inductive reasoning.

Datasets	Supervised SOTA		ULTRA(3g)		SCR		SCR (One)		SCR (MPNet)		SCR (Ontology)		SCR (4g)	
	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10
FB:v1	0.457	0.589	0.486	0.657	0.496	0.684	0.489	0.670	0.496	0.684	0.493	0.681	0.499	0.657
FB:v2	0.51	0.672	0.501	0.694	0.511	0.720	0.507	0.709	0.509	0.718	0.498	0.713	0.509	0.713
FB:v3	0.476	0.637	0.482	0.644	0.490	0.666	0.485	0.656	0.491	0.667	0.485	0.663	0.494	0.663
FB:v4	0.466	0.645	0.477	0.671	0.485	0.683	0.481	0.678	0.485	0.682	0.481	0.679	0.489	0.676
WN:v1	0.741	0.826	0.593	0.779	0.661	0.795	0.641	0.772	0.663	0.799	0.658	0.780	0.640	0.796
WN:v2	0.704	0.798	0.620	0.752	0.650	0.785	0.657	0.765	0.650	0.783	0.653	0.755	0.645	0.788
WN:v3	0.452	0.568	0.371	0.494	0.399	0.529	0.387	0.517	0.400	0.532	0.373	0.492	0.388	0.520
WN:v4	0.661	0.743	0.484	0.687	0.594	0.704	0.592	0.699	0.598	0.704	0.598	0.688	0.590	0.714
NL:v1	0.637	0.866	0.716	0.861	0.783	0.913	0.743	0.861	0.771	0.908	0.764	0.898	0.745	0.888
NL:v2	0.419	0.601	0.525	0.719	0.538	0.761	0.533	0.750	0.540	0.760	0.516	0.739	0.552	0.753
NL:v3	0.436	0.594	0.511	0.687	0.554	0.750	0.553	0.750	0.552	0.751	0.544	0.740	0.556	0.753
NL:v4	0.363	0.556	0.490	0.701	0.493	0.740	0.494	0.732	0.493	0.734	0.475	0.712	0.499	0.739
FB:25	0.133	0.271	0.383	0.633	0.389	0.645	0.388	0.641	0.389	0.645	0.386	0.640	0.387	0.640
FB:50	0.117	0.218	0.330	0.536	0.341	0.548	0.335	0.537	0.341	0.549	0.336	0.541	0.340	0.543
FB:75	0.189	0.325	0.391	0.594	0.400	0.611	0.399	0.603	0.400	0.610	0.395	0.603	0.397	0.603
FB:100	0.223	0.371	0.438	0.631	0.437	0.642	0.438	0.636	0.438	0.640	0.431	0.637	0.439	0.642
WK:25	0.186	0.309	0.307	0.507	0.292	0.497	0.297	0.495	0.290	0.491	0.289	0.483	0.301	0.518
WK:50	0.068	0.135	0.158	0.296	0.160	0.299	0.159	0.295	0.159	0.299	0.146	0.293	0.173	0.318
WK:75	0.247	0.362	0.373	0.519	0.365	0.532	0.368	0.522	0.365	0.531	0.342	0.514	0.375	0.536
WK:100	0.107	0.169	0.178	0.289	0.186	0.302	0.176	0.283	0.186	0.302	0.142	0.289	0.188	0.309
NL:25	0.334	0.501	0.387	0.538	0.392	0.601	0.359	0.562	0.394	0.604	0.376	0.566	0.404	0.612
NL:50	0.281	0.453	0.398	0.549	0.394	0.565	0.375	0.540	0.394	0.567	0.381	0.557	0.406	0.589
NL:75	0.261	0.464	0.348	0.527	0.349	0.535	0.350	0.519	0.350	0.540	0.341	0.535	0.360	0.562
NL:100	0.309	0.506	0.442	0.631	0.475	0.695	0.468	0.692	0.473	0.693	0.464	0.678	0.476	0.687