

# Beyond Completion: A Foundation Model for General Knowledge Graph Reasoning

Yin Hua<sup>♠</sup>, Zhiqiang Liu<sup>♠</sup>, Mingyang Chen<sup>♠</sup>, Zheng Fang<sup>♠</sup>, Chi Man Wong<sup>♠</sup>◇,  
Lingxiao Li<sup>♠</sup>, Chi Man VONG<sup>◇</sup>, Huajun Chen<sup>♠</sup>♡, Wen Zhang<sup>♠</sup>†

♠ Zhejiang University

♣ Shopee Pte.Ltd., ◇ University of Macau

♡ Zhejiang Key Laboratory of Big Data Intelligent Computing

{22351088, zhang.wen}@zju.edu.cn

## Abstract

In natural language processing (NLP) and computer vision (CV), the successful application of foundation models across diverse tasks has demonstrated their remarkable potential. However, despite the rich structural and textual information embedded in knowledge graphs (KGs), existing research of foundation model for KG has primarily focused on their structural aspects, with most efforts restricted to **in-KG** tasks (e.g., knowledge graph completion, KGC). This limitation has hindered progress in addressing more challenging **out-of-KG** tasks. In this paper, we introduce MERRY, a foundation model for general knowledge graph reasoning, and investigate its performance across two task categories: **in-KG** reasoning tasks (e.g., KGC) and **out-of-KG** tasks (e.g., KG question answering, KGQA). We not only utilize the structural information, but also the textual information in KGs. Specifically, we propose a multi-perspective Conditional Message Passing (CMP) encoding architecture to bridge the gap between textual and structural modalities, enabling their seamless integration. Additionally, we introduce a dynamic residual fusion module to selectively retain relevant textual information and a flexible edge scoring mechanism to adapt to diverse downstream tasks. Comprehensive evaluations on 28 datasets demonstrate that MERRY outperforms existing baselines in most scenarios, showcasing strong reasoning capabilities within KGs and excellent generalization to out-of-KG tasks such as KGQA.

recommendation (Guo et al., 2020), knowledge retrieval (Xu et al., 2024), and QA systems (Ji et al., 2022), as well as knowledge-grounded LLM alignment (Liu et al., 2025).

Recently, foundation models in NLP and CV Raffel et al. (2023); ChatGPT and Barnes (2023); Li et al. (2024); Ravi et al. (2024) have demonstrated significant advancements in transfer learning, enabling improved performance across datasets and tasks. Inspired by these successes, researchers have developed foundational models for KGs that aim to generalize across datasets and adapt to diverse reasoning tasks. KGs naturally encompass both structural and textual information, yet existing research has predominantly focused on leveraging their structural aspects, with relatively limited attention to the textual modality (Galkin et al., 2024; Zhu et al., 2021; Teru et al., 2020; Geng et al., 2022; Chen et al., 2022; Liu et al., 2024). However, fully utilizing both modalities is crucial, as textual information provides contextual knowledge that complements structural representations. This integration is particularly important for downstream applications such as commonsense reasoning and KGQA, where the combination of relational and contextual knowledge significantly enhances task performance (Yasunaga et al., 2021; Zhang et al., 2021; Markowitz et al., 2022). In addition, prior work has largely been restricted to in-KG reasoning tasks, such as KG Completion (KGC), and has not adequately addressed the challenges posed by out-of-KG reasoning tasks, such as KGQA. out-of-KG tasks require models to generalize beyond the explicit structure of KGs, incorporating both modalities to handle more complex reasoning scenarios.

Overcoming these limitations involves addressing three key challenges in model design: (1) mitigating the semantic disparity between textual and structural information to facilitate effective integration; (2) balancing the contributions of textual and

## 1 Introduction

Knowledge graphs (KGs) are structured knowledge bases that represent entities and their relationships, providing a foundation for reasoning and information retrieval in various real-world domains. With their rich entity representations and rigorous logical connections, KGs have become integral to applications such as classification (Liu et al., 2023),

† Corresponding author

structural modalities to suit diverse task requirements, particularly for reasoning beyond KGs; and (3) maintaining an unbiased training procedure to enable robust generalization across datasets without favoring specific entities or relations (Wang et al., 2022; Markowitz et al., 2022).

To address these challenges, we propose the Multi-perspective Reasoning sYstem, MERRY, a universal knowledge graph reasoning framework. MERRY integrates textual and structural information through a global structural semantic encoding module (GCMP), designed to reconcile their semantic differences. To enhance adaptability, we introduce a dynamic text-adaptive fusion module (DTAF) that selectively preserves essential textual information, facilitating effective application across a range of tasks. Furthermore, we develop a flexible edge scoring mechanism that adjusts adaptively to meet the specific requirements of downstream tasks, thereby enhancing the model’s transferability across diverse reasoning scenarios.

Both in-KG (zero-shot KGC) and out-of-KG (KGQA) tasks are evaluated in our MERRY. Results across 28 datasets demonstrate that MERRY consistently outperforms multiple benchmark models in both tasks, highlighting its robust generalization and adaptability. Our codes are released to the GitHub<sup>1</sup>. The main contributions of this paper are as follows:

- We propose a novel framework for addressing in-KG and out-of-KG reasoning tasks, integrating textual and structural modalities.
- We propose MERRY as a foundation model for general KG reasoning. By harmonizing structural and textual information, the framework achieves effective integration and ensures smooth transferability across reasoning tasks with varying modality demands.
- We validate MERRY’s performance on 28 datasets, demonstrating its effectiveness in zero-shot KGC and KGQA, with consistent improvements over multiple benchmarks.

## 2 Related Work

**Inductive Knowledge Graph Completion** KG Completion (KGC) is a fundamental task for reasoning over knowledge graphs. Its evolution can be categorized into three stages. Early work focused

on the transductive setting, where KGs are static, and entity and relation representations are precomputed and stored (Bordes et al., 2013; Sun et al., 2019; Vashishth et al., 2020).

Real-world KGs, however, are dynamic (Cui et al., 2022), requiring inductive methods to handle unseen entities and relations (Teru et al., 2020; Geng et al., 2022). These approaches rely on supervised training, limiting their generalization to unseen datasets and diverse KGC tasks.

Recent efforts leverage pre-training paradigms from NLP and CV. For example, ULTRA (Galkin et al., 2024) identifies meta-topology types in KG structures, enabling zero-shot transfer through dataset-agnostic representations of entities and relations. Nevertheless, it remains limited to structural information and does not incorporate textual modalities, which are critical for contextual reasoning. Moreover, it focuses exclusively on in-KG reasoning tasks, neglecting out-of-KG tasks.

### Text-aware Knowledge Graph Completion

While earlier studies emphasized KG structures, recent work explores textual information for improved reasoning. BLP and StAR enhance representation learning by initializing embedding tables with language models (LMs) (Daza et al., 2021; Wang et al., 2021). StATik (Markowitz et al., 2022) combines LMs and graph neural networks (GNNs) by encoding node text with LMs and capturing structural information via message passing.

Although these methods integrate textual and structural modalities effectively, their reliance on fine-tuning limits generalization to unseen datasets or tasks (Galkin et al., 2024). Additionally, they remain limited to in-KG reasoning tasks and lack the flexibility to address out-of-KG tasks, such as Knowledge Graph Question Answering (KGQA), which demands broader integration of textual and structural information.

### Knowledge Graph Question Answering

KGQA represents a key out-of-KG reasoning task. It links topic entities in queries to detailed KG, improving answer accuracy through relational and contextual reasoning (Wang et al., 2019).

Early methods used dual-tower architectures combining graph- and textual features with minimal interaction between modalities (Yang et al., 2019). Later approaches trained LMs on KG data to extract implicit knowledge and generate effective subgraphs for QA (Mihaylov and Frank, 2018; Lin et al., 2019; Feng et al., 2020; Lv et al., 2020).

<sup>1</sup><https://github.com/zjukg/MERRY>

Recent advancements include QA-GNN, which jointly updates LM and GNN layers through message passing (Yasunaga et al., 2021), and GreaseLM, which enhances LM-GNN integration by aligning GNN and Transformer layers for comprehensive information fusion (Zhang et al., 2021).

However, KGQA methods focus solely on out-of-KG reasoning tasks, while most KGC methods are confined to in-KG reasoning. This task-specific specialization highlights a key limitation: the lack of a unified framework capable of addressing both in-KG and out-of-KG reasoning effectively.

### 3 Task Definition

A KG with textual information is defined as  $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{D}\}$ , where  $\mathcal{E}$  and  $\mathcal{R}$  are the set of entities and relations,  $\mathcal{D}$  is the set of textual descriptions for entities and relations. The set of factual triples in the KG is denoted as  $\mathcal{T} = \{(e_h, r, e_t) \mid e_h, e_t \in \mathcal{E}, r \in \mathcal{R}\}$ , where  $e_h, e_t \in \mathcal{E}$  and  $r \in \mathcal{R}$ .

**Inductive KGC.** KG Completion (KGC) task aims to predict the correct entity  $e$  from the given KG  $\mathcal{G}$  for query  $(h, r, ?)$  or  $(?, r, t)$ . In particular, inductive KGC tasks aim to train a score function based on the train KG  $\mathcal{G}_{tr} = \{\mathcal{E}_{tr}, \mathcal{R}_{tr}, \mathcal{T}_{tr}, \mathcal{D}_{tr}\}$ . Considering the different inductive settings of the test KG  $\mathcal{G}_{te} = \{\mathcal{E}_{te}, \mathcal{R}_{te}, \mathcal{T}_{te}, \mathcal{D}_{te}\}$ , we can categorize the evaluation into: **(1) KG containing only unseen entities**, which satisfies  $\mathcal{E}_{tr} \neq \mathcal{E}_{te}$  and  $\mathcal{R}_{tr} = \mathcal{R}_{te}$ ; **(2) KG containing both unseen entities and unseen relations**, which satisfies  $\mathcal{E}_{tr} \neq \mathcal{E}_{te}$  and  $\mathcal{R}_{tr} \neq \mathcal{R}_{te}$ .

**KGQA.** Given a query *question* and several answer options  $\mathcal{C}$ , the KGQA task aims to retrieve subgraph from the KG  $\mathcal{G}$  and predict the correct answer  $a \in \mathcal{C}$ . To maintain consistency with the KGC task format, we define query as  $q = (\text{question}, REL\_the\_answer\_is, ?)$ , where *REL\_the\_answer\_is* is an auxiliary relation specifically introduced to establish a connection between the query and its corresponding correct answer node. Additionally, a subgraph retrieved from the whole KG is represented as  $\mathcal{G}_{sub} = \{\mathcal{E}_{sub}, \mathcal{R}_{sub}, \mathcal{T}_{sub}, \mathcal{D}_{sub}\}$  with entities  $\mathcal{E}_{sub} = \{\mathcal{E}_{topic}, \mathcal{E}_{option}, \mathcal{E}_{other}\}$ , where  $\mathcal{E}_{topic}$  represents the entity mentioned in the question  $q$ ,  $\mathcal{E}_{option}$  represents the entity mentioned in the options, and  $\mathcal{E}_{other}$  encompasses entities within the subgraph that do not carry particular contextual significance. The goal is to identify the correct answer option such that the triple

$(\text{question}, REL\_the\_answer\_is, \text{answer})$  is logically valid.

## 4 Methodology

A detailed breakdown of MERRY’s components is presented in this section, as illustrated in Figure 1. MERRY adopts an encoder-decoder architecture, and its processing can be formalized as follows:

$$scores = \text{MERRY}(q, \mathcal{G}, \mathcal{C}) \quad (1)$$

where  $q$  is the query,  $\mathcal{G}$  is the graph containing relevant textual descriptions, and  $\mathcal{C}$  are the candidates to be predicted. For KGC,  $\mathcal{C}$  corresponds to candidate entities, while for KGQA, it includes all possible answer options. MERRY produces a probability distribution over the candidates, where higher scores reflect a higher likelihood of correctness.

In the encoding phase, MERRY encodes the graph structure to derive its structural representation (Section 4.2) and explores strategies to effectively integrate textual and structural information (Section 4.3). A multi-perspective fusion module further enhances this process, enabling robust feature integration while preserving key textual semantics (Section 4.4). Additionally, we employ a flexible edge scoring mechanism to adapt to different tasks (Section 4.5).

In the decoding phase, a flexible cross-attention decoder facilitates adaptation to diverse downstream tasks, including zero-shot KGC and KGQA.

### 4.1 Conditional Message Passing

MERRY adopts Conditional Message Passing (CMP) as the basic GNN unit. Compared to traditional message-passing neural networks (MPNNs) like GCN(Kipf and Welling, 2017), GAT(Veličković et al., 2018), and GraphSAGE(Hamilton et al., 2017), CMP explicitly conditions the representation of a target node  $v$  on both a source node  $u$  and a query relation  $r_q$ . For detailed architectural specifications of this conditioning mechanism, see Huang et al. (2023). This process generates pairwise contextualized representations that dynamically adapt to the structural and semantic constraints imposed by  $(u, r_q)$ , enabling direct modeling of triple-level interactions(Zhu et al., 2021; Zhang and Yao, 2022; Galkin et al., 2024). Formally, the CMP process can be defined as:

$$\mathbf{H}_{node} = \text{INIT}(q) \quad (2)$$

$$\mathbf{H}_{node} = \text{CMP}(\mathbf{H}_{node}, \mathbf{H}_{edge}, \mathcal{G}) \quad (3)$$

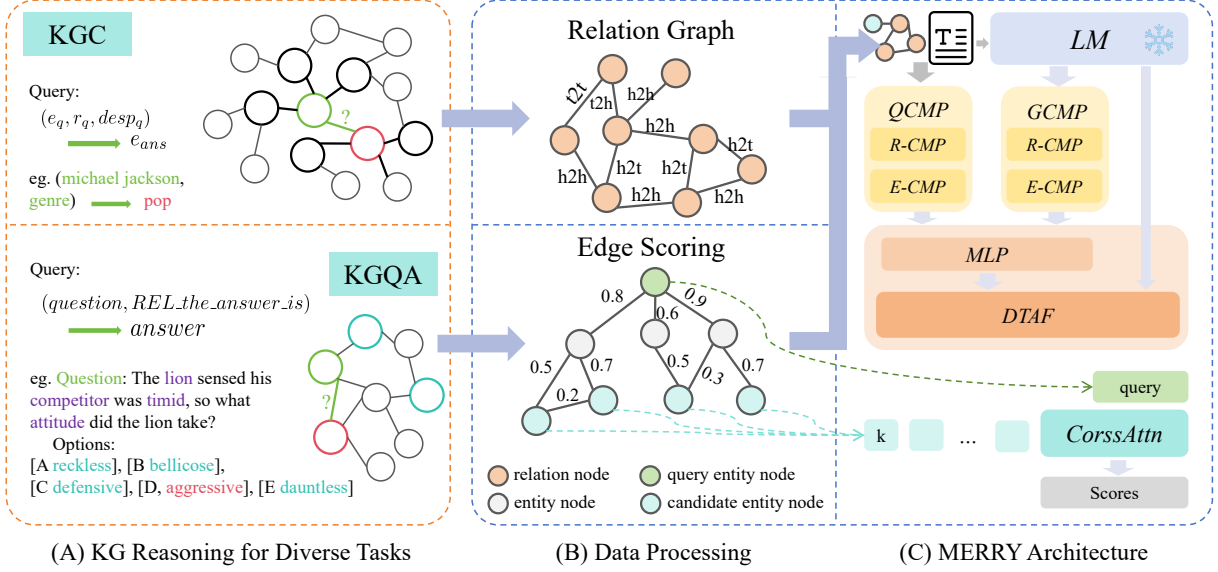


Figure 1: Overview of the MERRY Framework. (A) All tasks, including KGC and KGQA, are unified under a standardized query representation. (B) The data processing pipeline comprises two main components: (1) relation graph construction to model meta-relations, and (2) edge scoring to assign task-specific weights to edges. (C) The MERRY architecture processes these graphs through QCOMP, GCMP, and a multi-perspective dynamic fusion module. In the decoder, the query node is represented as the *Query* embedding, while candidate nodes serve as *Key* embeddings, outputting a probability distribution over all candidates.

where INIT is a conditional initialization function that initializes node representations conditioned on query  $q$ . It can be flexibly adapted for specific scenarios, as demonstrated in subsequent sections.  $\mathbf{H}_{node}$  represents the node representations,  $\mathbf{H}_{edge}$  is a learnable matrix for edge representations, and  $\mathcal{G}$  denotes the graph structure. Detailed descriptions of the CMP calculations are provided in Appendix A. In the following sections, we develop two core modules for structural and textual encoding based on CMP unit.

## 4.2 Query Conditional Structural Encoding

To handle the scenario of unseen relationships in arbitrary KGs, we follow previous works (Galkin et al., 2024; Chen et al., 2021), using the raw entity graph  $\mathcal{G}$  and four fixed meta-relations  $\mathcal{R}_{meta} = \{h2h, h2t, t2h, t2t\}$  to construct the corresponding relation graph. The relation graph is denoted as  $\mathcal{G}_r = \{\mathcal{R}, \mathcal{R}_{meta}, \mathcal{T}_r\}$ , where the nodes are relations derived from the entity graph  $\mathcal{G}$ , and the edges correspond to four types of meta-relations  $\mathcal{R}_{meta}$ . Details on the construction of triple sets  $\mathcal{T}_r$  can be found in the Appendix B.

The introduction of the relation graph enables us to encode arbitrary structures. To achieve this, we propose the **QCOMP** module, which applies CMP updates sequentially on the relation graph and the entity graph. This process yields query

conditioned representations for both relations and entities. Given a query  $q = (e_q, r_q, ?)$  and a KG  $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{D}\}$ , we first extract its relation graph  $\mathcal{G}_r$  and then encode it as follows:

$$\mathbf{r}_r = \begin{cases} \mathbf{1}^d, & \text{if } r = r_q \\ \mathbf{0}^d, & \text{otherwise} \end{cases}, \text{ for } r \in \mathcal{R} \quad (4)$$

$$\mathbf{R}_q = \text{CMP}(\parallel_{r=1}^{|\mathcal{R}|} \mathbf{r}_r, \mathbf{R}_{meta}, \mathcal{G}_r) \quad (5)$$

where  $\parallel$  is the concatenation operation,  $\mathbf{R}_{meta} \in \mathbb{R}^{4 \times d}$  is a learnable matrix corresponding to the four types of meta-relations, and  $\mathcal{G}_r$  is the relation graph constructed from  $\mathcal{G}$ . The conditional initialization function assigns an all-ones embedding  $\mathbf{1}^d$  to the query relation  $r_q$ , while all other relations are initialized with an all-zeros embedding  $\mathbf{0}^d$ , where  $d$  is the dimension of embeddings. The final output  $\mathbf{R}_q$  represents the query conditioned relation embeddings. Subsequently, we update the entity graph with CMP module:

$$\mathbf{h}_e = \begin{cases} \mathbf{R}_q[r_q], & \text{if } e = e_q \\ \mathbf{0}^d, & \text{otherwise} \end{cases}, \text{ for } e \in \mathcal{E} \quad (6)$$

$$\mathbf{H}_q = \text{CMP}(\parallel_{e=1}^{|\mathcal{E}|} \mathbf{h}_e, \mathbf{R}_q, \mathcal{G}) \quad (7)$$

where the embedding of  $r_q$  is used as the initialization for  $e_q$ , while all other entities are initialized to all-zero embeddings. The final output  $\mathbf{H}_q$  represents the query conditioned entity embeddings.



### 4.3 Global Structural Semantic Encoding

Textual information, as intrinsic node information, can be considered global information for the nodes. However, directly merging it with the structural modality information output by QCMP can lead to ineffective fusion due to the significant difference in their semantic spaces. Therefore, we propose the **GCMP** module to eliminate the semantic gap and achieve a more comprehensive modality fusion.

Specifically, we employ a Large Language Model (LLM) to encode textual information. However, since CMP requires features for all nodes in the graph as input, the substantial size of LLM weights can lead to an out-of-memory (OOM) risk. Therefore, we adopt a parameter-free strategy that extracts the representation of the last token from the LLM output to derive textual features for all nodes. The process of GCMP can be formalized as follows:

$$\mathbf{R}_g = \text{CMP}(\mathbf{1}^{|\mathcal{R}| \times d}, \hat{\mathbf{R}}_{meta}, \mathcal{G}_r) \quad (8)$$

$$\mathbf{H}_g = \text{CMP}(\mathcal{X}_e, \mathbf{R}_g, \mathcal{G}) \quad (9)$$

where  $\hat{\mathbf{R}}_{meta} \in \mathbb{R}^{4 \times d}$  represents a learnable matrix for meta-relations from textual perspective,  $\mathcal{X}_e$  represents the textual embeddings of all entities obtained via the parameter-free strategy. Specifically, each relation is initialized as an all-ones embedding, while the entity graph uses the textual embeddings  $\mathcal{X}_e$  as the initial representations. By applying this sequential CMP update process, we generate the global semantic embeddings for relations  $\mathbf{R}_g$  and entities  $\mathbf{H}_g$ .

### 4.4 Multi-Perspective Dynamic Fusion

**Multi-Channel CMP Fusion** As discussed earlier, MERRY encodes entities and relations from both query-specific and global perspectives through QCMP and GCMP, respectively. To integrate the outputs of these two CMP channels, we employ a multi-layer perceptron (MLP) for fusion:

$$\mathbf{R}_{CMP} = \text{MLP}([\mathbf{R}_q || \mathbf{R}_g]) \quad (10)$$

$$\mathbf{H}_{CMP} = \text{MLP}([\mathbf{H}_q || \mathbf{H}_g]) \quad (11)$$

**Dynamic Text-Adaptive Fusion** Although multi-channel CMP fusion bridges structural and textual information, empirical observations indicate that tasks such as KGC and KGQA place differing levels of emphasis on textual features. To accommodate this variability and dynamically preserve task-specific textual information, we further

propose a **Dynamic Text-Adaptive Fusion (DTAF)** module. Specifically, we adopt a parameterized cross-attention mechanism to encode input textual descriptions  $d \in \mathcal{D}$  into fixed-length embeddings:

$$\mathcal{X} = \text{Attn}(\mathbf{Q}_{token}, \text{LM}(d), \text{LM}(d)) \quad (12)$$

where  $\mathbf{Q}_{token} \in \mathbb{R}^{k \times d}$  represents trainable query parameters,  $k$  is a tunable hyperparameter, and  $\text{LM}(d)$  serves as both the Key and Value in the cross-attention mechanism. DTAF aggregates token-level information into meaningful representations  $\mathcal{X}$  while avoiding information loss.

Building on the textual embeddings, DTAF adaptively fuses textual and structural features using learnable weights  $\alpha$  and  $\beta$ , balancing their contributions based on task requirements:

$$\mathcal{X}_r = \text{Attn}(\mathbf{Q}_{token}, \text{LM}(\mathcal{D}_r), \text{LM}(\mathcal{D}_r)) \quad (13)$$

$$\mathcal{X}_e = \text{Attn}(\mathbf{Q}_{token}, \text{LM}(\mathcal{D}_e), \text{LM}(\mathcal{D}_e)) \quad (14)$$

$$\mathbf{R}_f = \alpha * \mathcal{X}_r + (1 - \alpha) * \mathbf{R}_{CMP}, \quad (15)$$

$$\mathbf{H}_f = \beta * \mathcal{X}_e + (1 - \beta) * \mathbf{H}_{CMP}, \quad (16)$$

where  $\mathcal{D}_r$  and  $\mathcal{D}_e$  are the textual descriptions of relations and entities, respectively. The outputs  $\mathcal{X}_r$  and  $\mathcal{X}_e$  represent the textual features of relations and entities, respectively. The fused embeddings  $\mathbf{R}_f$  and  $\mathbf{H}_f$  are unified representations that integrate three different perspectives.

### 4.5 Query Conditional Edge Scoring

Edge scores in MPNNs are crucial for model performance and vary significantly across tasks. To adapt to these differences, we design a flexible module tailored to task-specific requirements.

In KGC tasks, most methods focus on message passing and aggregation, often setting all edge scores to 1 (Veličković et al., 2018). But in KGQA tasks, noisy paths in the retrieved subgraph necessitate more refined edge scoring. Compared to node relevance scores, edge scores capture richer interactions among the head entity, relation, and tail entity, offering a more accurate relevance measure for the query (Yasunaga et al., 2021). For each edge  $(h, r, t)$  in the subgraph, its query relevance is calculated using a bilinear layer:

$$\eta = \text{Norm}([\mathbf{x}_h || \mathbf{x}_r || \mathbf{x}_t]^\top \mathbf{W} \mathbf{x}_q), \quad (17)$$

where  $\mathbf{W} \in \mathbb{R}^{3d \times d}$  is the bilinear coefficient,  $\mathbf{x}_h, \mathbf{x}_r, \mathbf{x}_t, \mathbf{x}_q \in \mathcal{X}$  represent the textual features of  $(h, r, t)$  and the query  $q$ , obtained using the parameter-free method introduced in Section 4.3.

The output  $\eta \in \mathbb{R}^{2 \times 1}$  includes relevance and irrelevance scores, normalized with a Softmax function. The relevance score is then used in the update function of CMP.

#### 4.6 Training Mechanism

**Self-Supervised Pre-Training** The encoding process of MERRY is both entity-agnostic and relation-agnostic, making it suitable for inductive scenarios and allowing pre-training on arbitrary or hybrid KGs. The pre-training task employs self-supervised link prediction, with binary cross-entropy loss for positive and negative samples (Sun et al., 2019; Zhu et al., 2021):

$$\mathcal{L} = -\log p(q, ans) - \sum_{i=1}^n \frac{1}{n} \log(1 - p(q, neg\_ans)), \quad (18)$$

where  $q$  is the query prefix of the triple  $(h, r, ?)$ , and  $ans$  is the tail entity  $t$  that makes  $(h, r, t)$  valid in the knowledge graph. Negative samples are generated by randomly selecting tail entities. MERRY is pre-trained on multiple hybrid KG datasets, which equips it with generalizable transferability across diverse knowledge graphs.

**Task Adaptation** For the KGC task, the model is evaluated in a zero-shot setting without fine-tuning, using the same process as pre-training.

For the KGQA task, input questions are summarized as a combination of the query and the retrieved subgraph. The query is formalized as  $q = (question, REL\_the\_answer\_is)$ , where the candidates are the possible options. The goal is to select the correct answer such that  $(question, REL\_the\_answer\_is, answer)$  forms a valid triple, with  $REL\_the\_answer\_is$  is a newly introduced relation.

We adapt the data in three steps. First, a question-node is introduced to represent the input question, connected to all topic entities via a new relation. Its text description is the question itself. Additionally, each candidate option is represented by an answer-node, connected to the entities in the option via a special relation. Its text description is the original text of the option. Finally, we introduce a new relation,  $REL\_the\_answer\_is$ , which connects the question-node to the correct answer-node.

Since  $REL\_the\_answer\_is$  lacks neighboring nodes in the relation graph, we adopt a few-shot approach. Using Sentence-BERT (Reimers and Gurevych, 2019) we compute sentence embeddings

for each question and retrieve the top-K most similar questions based on cosine similarity. These few-shot examples are used to enrich the instances of  $REL\_the\_answer\_is$ .

With these modifications, MERRY can seamlessly transfer to perform the KGQA task.

## 5 Experiments

We evaluate MERRY on 28 datasets across two tasks: Inductive Knowledge Graph Completion (KGC) and Knowledge Graph Question Answering (KGQA). Our evaluation focuses on the following research questions: **RQ1**: How effective is MERRY in reasoning for **in-KG** tasks under a zero-shot setting? **RQ2**: Can MERRY effectively transfer and generalize to **out-of-KG** tasks? **RQ3**: What is the impact of key components on the performance of MERRY? **RQ4**: How do key hyperparameters affect the performance of MERRY?

### 5.1 Datasets and Metrics

**Inductive KGC** We perform zero-shot inductive KGC experiments on 27 datasets, categorized by entity and relation visibility: (1) **Inductive Entity (e) Datasets (IndE)**: These datasets feature unseen entities in the test set, with fixed relations. This category includes 12 datasets from (Teru et al., 2020): WN18RR (WN), FB15k-237 (FB), and NELL-995 (NL), each with four different versions. (2) **Inductive Entity and Relation (e, r) Datasets (IndER)**: These datasets include unseen entities and relations in the test set. This category comprises 13 graphs from (Lee et al., 2023): FB15k-237 (FB) and Wikidata68K (WK), each with four versions, and NELL-995 (NL), which has five versions. We report Mean Reciprocal Rank (MRR) and Hits@10 results.

**KGQA** We use CommonsenseQA (CSQA) dataset (Talmor et al., 2019), which focuses on commonsense reasoning. It consists of 12,102 multiple-choice questions. We follow the in-house split method from (Lin et al., 2019) for experiments and compare our results with several baseline models. We report Accuracy (Acc) on the CSQA dataset.

For detailed information on datasets and metric computation formulas, refer to Appendix C and Appendix D, respectively.

### 5.2 Baselines

**Inductive KGC** We compare MERRY against state-of-the-art supervised methods and recent KG

Methods	IndE(WN)		IndE(FB)		IndE(NL)		IndER(FB)		IndER(WK)		IndER(NL)		Total AVG		SOTA Num
	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	
Supervised SOTA	0.640	0.734	0.477	0.636	0.464	0.654	0.166	0.296	0.152	0.244	0.296	0.481	0.366	0.507	-
ULTRA(3g)	0.517	0.678	0.486	<u>0.667</u>	<u>0.561</u>	0.742	<b>0.386</b>	<b>0.599</b>	<u>0.254</u>	0.403	0.393	0.561	0.433	0.608	4 / 24
ProLINK	0.553	0.690	<b>0.494</b>	<b>0.684</b>	0.546	<u>0.759</u>	0.372	0.591	0.234	0.393	<b>0.400</b>	<b>0.590</b>	0.433	0.618	<u>8 / 24</u>
MERRY	<b>0.563</b>	<b>0.709</b>	0.486	0.662	<b>0.567</b>	<b>0.767</b>	<u>0.378</u>	<u>0.592</u>	<b>0.282</b>	<b>0.443</b>	<u>0.397</u>	<u>0.586</u>	<b>0.445</b>	<b>0.626</b>	<b>12 / 24</b>
MERRY <sub>PNA</sub>	<u>0.559</u>	<u>0.694</u>	0.484	0.660	0.560	0.754	0.359	0.584	0.261	<u>0.426</u>	0.384	0.569	0.435	<u>0.615</u>	-

Table 1: Zero-shot and supervised SOTA performance on 24 KG inductive reasoning datasets. The best results across baselines, supervised methods, and MERRY are **bolded**. The second-best results are underlined. The **SOTA Num** column indicates the number of datasets where each method achieves SOTA performance.

foundation models, including ULTRA and ProLINK (Galkin et al., 2024; Wang et al., 2024), for zero-shot learning. Here, ULTRA(3g) refers to pre-training on three graphs.

**KGQA** For KGQA, we use a fine-tuned standard LM as the baseline for models without external knowledge. Additionally, we evaluate several LM+KG-based methods, including RN (Santoro et al., 2017), RGCN (Schlichtkrull et al., 2017), GconAttn (Wang et al., 2018), KagNet (Lin et al., 2019), MHGRN (Feng et al., 2020), QA-GNN (Yasunaga et al., 2021), and GreaseLM (Zhang et al., 2021). Among these, the best-performing models synchronize updates between the LM and GNN, enabling mutual interaction between textual and structural modalities.

### 5.3 Implementation & Training details

We pre-train MERRY on three hybrid knowledge graph datasets: WN18RR, CoDEx-Medium, and FB15k237, to capture diverse relational structures and sparsity patterns (Dettmers et al., 2018; Toutanova and Chen, 2015; Safavi and Koutra, 2020). Based on ULTRA, we set QCOMP to a 6-layer CMP and GCMP to a 3-layer CMP, with each hidden layer having a dimension 64. To enhance convergence, we employ a two-stage training strategy: (1) QCOMP weights from ULTRA are frozen, and other modules, particularly GCMP, are trained. (2) All components are unfrozen, allowing QCOMP and other modules to converge jointly. During training, the LM backbone remains frozen.

For Inductive KGC, we evaluate the zero-shot capability of the pre-trained model directly on downstream datasets, using the Llama3 8B LM backbone (Grattafiori et al., 2024).

For KGQA, due to the substantial gap between pre-training and the downstream task, we fine-tune the model with three few-shot examples before testing. Considering commonsense reasoning requires alignment with human cognitive preferences, we use the Llama3 8B Instruct backbone.

Methods	IHdev-Acc.(%)	IHtest-Acc.(%)
RoBERTa-Large	73.1	68.7
LLaMA-3-8b-instruct	72.9	71.9
RGCN	72.7	68.4
GconAttn	72.6	68.6
KagNet	73.5	69.0
RN	74.6	69.1
MHGRN	74.5	71.1
QA-GNN	76.5	73.4
GreaseLM	<u>78.5</u>	<u>74.2</u>
MERRY	<b>78.6</b>	<b>74.9</b>

Table 2: Performance comparison on CommonsenseQA in-house split (controlled experiments).

### 5.4 Main Results (RQ1)

We compare MERRY with baselines on 27 inductive link prediction KG datasets, categorized into 7 benchmarks based on data sources. For a fair comparison, datasets IndE (ILPC-small), IndE (ILPC-large), and IndER (NL-0) are excluded. Table 1 presents the average results across 6 benchmarks, 24 datasets. A full comparison of results across 27 datasets is provided in Appendix E.

Four benchmarks, IndE(X) from (Teru et al., 2020), contain unseen entities in the test graph. In contrast, the IndER (X) benchmark from (Lee et al., 2023) includes unseen entities and relations, making it significantly more challenging. Among all dataset benchmarks, IndER (WK), IndE (NL), and IndER (NL) contain entities and relations unseen during pre-training, providing a strong evaluation of the model’s zero-shot generalization capability. Table 1 shows that MERRY outperforms baselines.

Additionally, we compare MERRY with a parameter-free PNA method (Corso et al., 2020), used for encoding textual descriptions of entities and relations (4.3). From the average results, while the MERRY<sub>PNA</sub> variant shows a slight decline in performance, it demonstrates that our design retains a certain level of robustness.

Overall, MERRY surpasses state-of-the-art supervised models and existing zero-shot transfer methods in total average metrics. While ULTRA and ProLINK excel on specific datasets, their performance is largely limited to datasets they were trained on.

### 5.5 Generalization to KGQA (RQ2)

Table 2 compares MERRY with previous state-of-the-art methods on the CSQA dataset. MERRY achieves superior performance, surpassing all baselines and delivering the best overall results. Notably, compared to GreaseLM, which integrates GNN and LM layers through bidirectional interactions, MERRY performs comparably on the validation set but exceeds it on the test set. This demonstrates the effectiveness of our approach in integrating textual and structural modalities.

These results highlight the robustness of our multimodal fusion strategy and strong generalization capabilities. Additionally, in zero-shot inference using Llama3 8b Instruct, MERRY shows significant improvement, further validating its ability to incorporate structural information without compromising textual understanding.

### 5.6 Ablation Studies (RQ3)

We conducted ablation experiments on multiple datasets, including IndE(X) and IndER(X), to evaluate the impact of two key components in our method for KGC. As shown in Figure 2, "w/o GCMP" indicates the removal of the GCMP module, where node text and structural features are instead concatenated and fused via an MLP. "w/o DTAF" refers to the model where DTAF is ignored, relying solely on CMP-based fusion for downstream predictions.

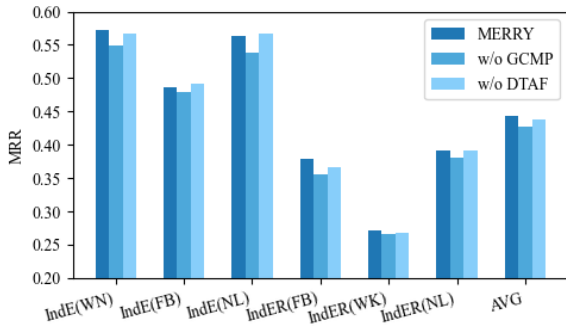


Figure 2: Ablation study results.

The results demonstrate a significant performance drop in the "w/o GCMP" variant, highlighting its critical role in bridging the gap between textual and structural modalities for better integration. In contrast, the "w/o DTAF" variant shows a slight performance decline, indicating that while original text features aid KGC, DTAF primarily enhances the understanding of structural information.

Similarly, we conducted ablation experiments on the CSQA dataset, as shown in Table 3. An additional variant, "w/o Edge Scoring", sets all edge

Edge Scoring	DTAF	IHdev-Acc.(%)	IHtest-Acc.(%)
✓	✓	78.6	74.9
	✓	77.7	75.0
		71.4	70.7

Table 3: Ablation results of the edge scoring mechanism and DTAF module on the CSQA dataset.

scores to 1, similar to the KGC tasks. The results indicate that DTAF significantly impacts KGQA performance, highlighting the importance of text feature understanding in these tasks and its role in preserving the LM’s text processing capability. Moreover, ignoring edge scores results in a performance decline, underscoring the importance of edge weights in KGQA.

### 5.7 Hyperparameter Sensitivity (RQ4)

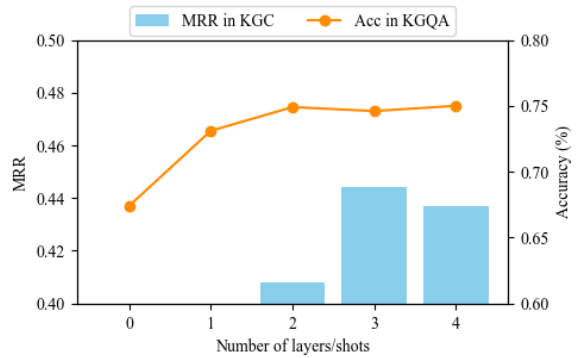


Figure 3: Performance of different GCMP layers in KGC and different numbers of shots in KGQA.

We investigated the impact of GCMP layers on zero-shot KGC tasks and assessed the role of few-shot learning in KGQA. As illustrated in Figure 3, using too few GCMP layers results in poor convergence, while excessive layers lead to feature smoothing. Aggregating information from up to three hops strikes an optimal balance, enabling effective performance.

For KGQA, the introduction of few-shot learning proves essential. As expected, zero-shot performance is initially poor. However, as the number of shots increases, performance stabilizes, demonstrating the model’s capacity to rapidly adapt and learn new relationships with minimal data.

### 5.8 Computational Complexity and Scalability Analysis

To ensure practical applicability, we theoretically analyze MERRY’s computational efficiency under two decoupled phases:

- **Phase 1: LLM Text Encoding Complexity** scales as  $O(|V| \cdot T_{LLM})$ , where  $|V|$  is the



node count and  $T_{LLM}$  is the per-node encoding time. Our parameter-free feature extraction (Section 4.3) enables *one-time offline pre-processing*, converting  $T_{LLM}$  into a fixed cost during model deployment.

- **Phase 2: CMP Graph Updates** Each iteration requires  $O(|E|d + |V|d^2)$  operations, where  $|E|$  denotes the number of edges and  $d$  is the feature dimension. This complexity aligns with state-of-the-art GNNs like ULTRA (Galkin et al., 2024) and NBFNet (Zhu et al., 2021), while demonstrating significant advantages over classic inductive KGC approaches. Specifically, compared to GraIL’s  $O(|E|d^2 + |V|d^2)$  complexity for *closed sub-graph encoding* (Teru et al., 2020), MERRY achieves a  **$d$ -fold reduction** in edge-related computation, making it particularly advantageous for graphs with large edge sets or high-dimensional features.

**Scalability Advantages:** Based on the above time-complexity analysis, MERRY demonstrates strong scalability on large-scale graphs. By decoupling the LLM encoding phase, all node textual features can be precomputed offline at a cost of  $O(|V| \cdot T_{LLM})$  and then stored and retrieved via a distributed system. Furthermore, the CMP graph-update complexity shows that, for a fixed hidden-layer dimension  $d$ , MERRY’s online computation  $O(|E|d + |V|d^2)$  is substantially lower than the  $O(|E|d^2 + |V|d^2)$  required by classical approaches. Together, these results demonstrate that our framework achieves a favorable trade-off between performance and efficiency.

## 6 Conclusion

In this paper, we introduced MERRY, a general knowledge graph reasoning framework that bridges textual and structural modalities through multi-channel CMP encoding and multi-perspective dynamic fusion mechanisms. Additionally, we proposed a flexible edge scoring mechanism to adapt to diverse downstream tasks. Experiments across 28 datasets demonstrate MERRY’s strong generalization capabilities in in-KG tasks, such as zero-shot KGC, and its adaptability to out-of-KG tasks, such as KGQA, highlighting its potential as a unified framework for reasoning across in-KG and out-of-KG tasks.

## Acknowledgment

This work is funded by National Natural Science Foundation of China (NSFC62306276/NSFCU23B2055/NSFCU19B2027), Zhejiang Provincial Natural Science Foundation of China (No. LQ23F020017), Yongjiang Talent Introduction Programme (2022A-238-G), and Fundamental Research Funds for the Central Universities (226-2023-00138).

## Limitations

Here, we discuss three limitations of this work. First, through hyperparameter tuning experiments, it is evident that the CMP module’s depth has limitations. A higher number of layers leads to feature smoothing, which is a challenge commonly faced by models incorporating GNN architectures. Second, we assumed that each entity and relation in the KG dataset has a corresponding textual description. However, our investigation discovered that some datasets need better maintenance, resulting in missing textual fields for certain entities. This issue of data completeness poses challenges for approaches that rely on language models. Finally, while LLM have demonstrated significant potential across various tasks, they face unique challenges in the in-KG task. Due to the size of the graph, encoding all nodes becomes particularly difficult, not only introducing substantial time and memory overhead during training but also consuming considerable storage space for offline feature storage. Efficiently leveraging LLMs in the in-KG tasks thus remains a crucial area for future exploration.

## References

- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.
- ChatGPT and Richard L. Barnes. 2023. [AI sarcasm detection: Insult your AI without offending it](#). *RFC*, 9405:1–5.
- Jiajun Chen, Huarui He, Feng Wu, and Jie Wang. 2021. [Topology-aware correlations between relations for inductive link prediction in knowledge graphs](#). *Preprint*, arXiv:2103.03642.
- Mingyang Chen, Wen Zhang, Zhen Yao, Xiangnan Chen, Mengxiao Ding, Fei Huang, and Huajun Chen. 2022. [Meta-learning based knowledge extrapolation](#)

- for knowledge graphs in the federated setting. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1966–1972. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. 2020. [Principal neighbourhood aggregation for graph nets](#). *Preprint*, arXiv:2004.05718.
- Yuaning Cui, Yuxin Wang, Zequn Sun, Wenqiang Liu, Yiqiao Jiang, Kexin Han, and Wei Hu. 2022. [Inductive knowledge graph reasoning for multi-batch emerging entities](#). *Preprint*, arXiv:2208.10378.
- Daniel Daza, Michael Cochez, and Paul Groth. 2021. [Inductive entity representations from text via link prediction](#). In *Proceedings of the Web Conference 2021*, WWW ’21, page 798–808. ACM.
- Tim Dettmers, Pasquale Minervini, Pontus Stenertorp, and Sebastian Riedel. 2018. [Convolutional 2d knowledge graph embeddings](#). *Preprint*, arXiv:1707.01476.
- Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. [Scalable multi-hop relational reasoning for knowledge-aware question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online. Association for Computational Linguistics.
- Mikhail Galkin, Max Berrendorf, and Charles Tapley Hoyt. 2022a. [An open challenge for inductive link prediction on knowledge graphs](#). *CoRR*, abs/2203.01520.
- Mikhail Galkin, Etienne Denis, Jiapeng Wu, and William L. Hamilton. 2022b. [Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs](#). *Preprint*, arXiv:2106.12144.
- Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2024. [Towards foundation models for knowledge graph reasoning](#). In *The Twelfth International Conference on Learning Representations*.
- Yuxia Geng, Jiaoyan Chen, Jeff Z. Pan, Mingyang Chen, Song Jiang, Wen Zhang, and Huajun Chen. 2022. [Relational message passing for fully inductive knowledge graph completion](#). *Preprint*, arXiv:2210.03994.
- Genet Asefa Gesese, Harald Sack, and Mehwish Alam. 2023. [Raild: Towards leveraging relation features for inductive link prediction in knowledge graphs](#). In *Proceedings of the 11th International Joint Conference on Knowledge Graphs, IJCKG ’22*, page 82–90, New York, NY, USA. Association for Computing Machinery.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yearly, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kam-badur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shao-liang Nie, Sharan Narang, Sharath Rapparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collet, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenxin Fu, Whit-

ney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit San-gani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchan-dani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yaz-dan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippas Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanaz-eri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jen-nifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khan-delwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Ki-ran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrst-edt, Madian Khabza, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Pat-el, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso,

Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pe-dro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lind-say, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wen-wen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.

Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. [A survey on knowledge graph-based recommender systems](#). *Preprint*, arXiv:2003.00911.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, page 1025–1035, Red Hook, NY, USA. Curran Associates Inc.

Xingyue Huang, Miguel Romero Orth, İsmail İlkan Ceylan, and Pablo Barceló. 2023. [A theory of link prediction via relational weisfeiler-leman on knowl-edge graphs](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Mart-tinen, and Philip S. Yu. 2022. [A survey on knowl-edge graphs: Representation, acquisition, and appli-](#)



- cations. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):494–514.
- Thomas N. Kipf and Max Welling. 2017. [Semi-supervised classification with graph convolutional networks](#). In *International Conference on Learning Representations*.
- Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. 2023. [Ingram: Inductive knowledge graph embedding via relation graphs](#). *Preprint*, arXiv:2305.19987.
- Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*.
- Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. [KagNet: Knowledge-aware graph networks for commonsense reasoning](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2829–2839, Hong Kong, China. Association for Computational Linguistics.
- Ye Liu, Kai Zhang, Zhenya Huang, Kehang Wang, Yang-hai Zhang, Qi Liu, and Enhong Chen. 2023. [Enhancing hierarchical text classification through knowledge graph integration](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5797–5810, Toronto, Canada. Association for Computational Linguistics.
- Zhiqiang Liu, Mingyang Chen, Yin Hua, Zhuo Chen, Ziqi Liu, Lei Liang, Huajun Chen, and Wen Zhang. 2024. Unih: Hierarchical representation learning for unified knowledge graph link prediction. *arXiv preprint arXiv:2411.07019*.
- Zhiqiang Liu, Chengtao Gan, Junjie Wang, Yichi Zhang, Zhongpu Bo, Mengshu Sun, Huajun Chen, and Wen Zhang. 2025. [Ontotune: Ontology-driven self-training for aligning large language models](#). In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pages 119–133. ACM.
- Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2020. [Graph-based reasoning over heterogeneous external knowledge for commonsense question answering](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8449–8456.
- Elan Markowitz, Keshav Balasubramanian, Mehrnoosh Mirtaheiri, Murali Annavaram, Aram Galstyan, and Greg Ver Steeg. 2022. [StATIK: Structure and text for inductive knowledge graph completion](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 604–615, Seattle, United States. Association for Computational Linguistics.
- Todor Mihaylov and Anette Frank. 2018. [Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832, Melbourne, Australia. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Preprint*, arXiv:1910.10683.
- Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. 2024. [Sam 2: Segment anything in images and videos](#). *arXiv preprint arXiv:2408.00714*.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). *Preprint*, arXiv:1908.10084.
- Tara Safavi and Danai Koutra. 2020. [CoDEX: A Comprehensive Knowledge Graph Completion Benchmark](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8328–8350, Online. Association for Computational Linguistics.
- Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. [A simple neural network module for relational reasoning](#). *Preprint*, arXiv:1706.01427.
- Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. [Modeling relational data with graph convolutional networks](#). *Preprint*, arXiv:1703.06103.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. [Rotate: Knowledge graph embedding by relational rotation in complex space](#). *Preprint*, arXiv:1902.10197.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. [Commonsenseqa: A question answering challenge targeting commonsense knowledge](#). *Preprint*, arXiv:1811.00937.
- Komal K. Teru, Etienne Denis, and William L. Hamilton. 2020. Inductive relation prediction by subgraph reasoning. *arXiv: Learning*.
- Kristina Toutanova and Danqi Chen. 2015. [Observed versus latent features for knowledge base and text inference](#). In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China. Association for Computational Linguistics.



- Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. [Composition-based multi-relational graph convolutional networks](#). *Preprint*, arXiv:1911.03082.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. [Graph attention networks](#). *Preprint*, arXiv:1710.10903.
- Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. [Structure-augmented text representation learning for efficient knowledge graph completion](#). In *Proceedings of the Web Conference 2021*, WWW '21, page 1737–1748. ACM.
- Kai Wang, Yuwei Xu, Zhiyong Wu, and Siqiang Luo. 2024. [LLM as prompter: Low-resource inductive reasoning on arbitrary knowledge graphs](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3742–3759, Bangkok, Thailand. Association for Computational Linguistics.
- Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. [SimKGC: Simple contrastive knowledge graph completion with pre-trained language models](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294, Dublin, Ireland. Association for Computational Linguistics.
- Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, and Michael Witbrock. 2018. [Improving natural language inference using external knowledge in the science questions domain](#). *Preprint*, arXiv:1809.05724.
- Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, and Michael Witbrock. 2019. [Improving natural language inference using external knowledge in the science questions domain](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7208–7215.
- Wenhan Xiong, Thien Hoang, and William Yang Wang. 2018. [Deeppath: A reinforcement learning method for knowledge graph reasoning](#). *Preprint*, arXiv:1707.06690.
- Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. 2024. [Retrieval-augmented generation with knowledge graphs for customer service question answering](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2024, page 2905–2909. ACM.
- An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaobao She, and Sujian Li. 2019. [Enhancing pre-trained language representations with rich knowledge for machine reading comprehension](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357, Florence, Italy. Association for Computational Linguistics.
- Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. [Qa-gnn: Reasoning with language models and knowledge graphs for question answering](#). In *North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2021. [GreaseLM: Graph reasoning enhanced language models](#). In *International Conference on Learning Representations*.
- Yongqi Zhang and Quanming Yao. 2022. [Knowledge graph reasoning with relational digraph](#). In *Proceedings of the ACM Web Conference 2022*, pages 912–924.
- Zhaocheng Zhu, Xinyu Yuan, Mikhail Galkin, Sophie Xhonneux, Ming Zhang, Maxime Gazeau, and Jian Tang. 2023. [A\\*net: A scalable path-based reasoning approach for knowledge graphs](#). *Preprint*, arXiv:2206.04798.
- Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. 2021. [Neural bellman-ford networks: A general graph neural network framework for link prediction](#). *Advances in Neural Information Processing Systems*, 34.

## A Details of CMP Updates

Given a graph  $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ , where the feature of any entity  $u$   $\mathbf{h}_u$  and the feature of any relation is denoted as  $\mathbf{r}$ , the update process for the  $(t + 1)$ -th layer of CMP (Conditional Message Passing) is formalized as follows:

$$\mathbf{m}_u^{t+1} = \text{MSG}(\mathbf{h}_w^t, \mathbf{r}), w \in \mathcal{N}_r(u), \quad (19)$$

$$\mathbf{h}_u^{t+1} = \text{UPDATE}(\mathbf{h}_u^t, \text{AGG}(\mathbf{m}_u^{t+1})) \quad (20)$$

where, we follow the settings of NBFNet, where the message function uses the parameter-free DistMult, the aggregation function employs summation, and UPDATE is implemented as a linear layer with LayerNorm.

When edge scores are introduced, the message function is adjusted to incorporate relevance scores. If the relevance score for any edge is denoted as  $s$ , the modified update equations become:

$$\mathbf{m}_u^{t+1} = s \cdot \text{MSG}(\mathbf{h}_w^t, \mathbf{r}), w \in \mathcal{N}_r(u), \quad (21)$$

$$\mathbf{h}_u^{t+1} = \text{UPDATE}(\mathbf{h}_u^t, \text{AGG}(\mathbf{m}_u^{t+1})) \quad (22)$$

where the edge score  $s$  weights the message contribution from each neighbor, enhancing the model’s ability to capture relevance-specific information in graph updates.

## B Relation Graph Construction

Given a graph  $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ , we apply the lifting function  $\mathcal{G}_r = \text{LIFT}(\mathcal{G})$  to build a graph of relations  $\mathcal{G}_r = (\mathcal{E}_r, \mathcal{R}_{meta}, \mathcal{T}_r)$  where each node is a distinct relation type in  $\mathcal{G}$ . Triples  $\mathcal{T}_r \in (\mathcal{R} \times \mathcal{R}_{meta} \times \mathcal{R})$  in the relation graph  $\mathcal{G}_r$  denote interactions between relations in the original graph  $\mathcal{G}$ , and we distinguish four such meta-relation interactions  $\mathcal{R}_{meta}$ : *tail-to-head* (*t2h*) edges, *head-to-head* (*h2h*) edges, *head-to-tail* (*h2t*) edges, and *tail-to-tail* (*t2t*) edges. Each of the four adjacency matrices can be efficiently obtained with one sparse matrix multiplication; for details, refer to Galkin et al. (2024).

## C Datasets

**Pre-Training** Considering MERRY’s effective generalization across datasets, we perform pre-training using a mix of the WN18RR, FB15k237, and CodexMedium datasets. Table 4 presents the statistics of these three datasets, highlighting their data diversity.

**Inductive KGC** Our zero-shot Inductive KG Completion (KGC) experiments are conducted on 27 datasets. Among these, 12 datasets are derived from the GraIL framework (Teru et al., 2020), which utilizes widely recognized KG benchmarks such as WN18RR (Dettmers et al., 2018), FB15k237 (Toutanova and Chen, 2015), and NELL-995 (Xiong et al., 2018), and 2 datasets are derived from the ILPC (Galkin et al., 2022a). These datasets are designed such that the training and testing graphs maintain consistent relation types.

Additionally, we incorporate 13 datasets from the InGram framework (Lee et al., 2023) to further assess inductive reasoning performance. These datasets are generated from three real-world knowledge graph benchmarks: FB15k237 (Toutanova and Chen, 2015), Wikidata68K (Gesese et al., 2023), and NELL-995 (Xiong et al., 2018). Each dataset is partitioned into subsets with varying proportions of novel relational triples, specifically 100%, 75%, 50%, and 25%, enabling evaluation under diverse inductive settings. Additionally, the NELL-995 also has a variant dataset with 0

While other KG datasets with textual descriptions exist, their limited accessibility precludes

their inclusion in this study. Future research may focus on evaluating these datasets. Comprehensive structural statistics for the datasets employed in this work are presented in Table 5.

**KGQA** In our KG question answering (KGQA) experiments, the CommonsenseQA dataset is used as a representative for this type of task (Talmor et al., 2019). CSQA is a multiple-choice question-answering benchmark with five answer options per question, aimed at assessing reasoning based on commonsense knowledge. It includes a total of 12,102 questions. As the test set for CSQA is not openly accessible, evaluation can only be conducted biweekly through submissions to the official leaderboard.

For our primary experiments, we rely on the in-house (IH) data splits introduced by (Lin et al., 2019) for training and validation purposes. The performance of our final system is also evaluated on the official test set to provide a direct comparison with existing methods.

Dataset	$ \mathcal{E}_{tr} $	$ \mathcal{R}_{tr} $	#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 4: Statistics of pre-training KG datasets.

## D Metrics

**Mean Reciprocal Rank (MRR)** The **Mean Reciprocal Rank (MRR)** evaluates the quality of the ranking in Knowledge Graph Completion (KGC) tasks. For a given query  $q$ , let the rank of the correct candidate be  $r_q$ . The reciprocal rank is defined as  $\frac{1}{r_q}$ . Averaging over all queries, MRR is calculated as:

$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{r_q} \quad (23)$$

where  $Q$  represents the set of all queries. A higher MRR indicates better model performance in ranking the correct candidate higher in the prediction list.

**Hits@10** The **Hits@10** metric measures the proportion of queries for which the correct candidate is ranked within the top 10 predictions. For a given query  $q$ , let the rank of the correct candidate be  $r_q$ . Hits@10 is defined as:

$$\text{Hits@10} = \frac{1}{|Q|} \sum_{q \in Q} \mathbf{1}[r_q \leq 10], \quad (24)$$

where  $\mathbf{1}[\cdot]$  is an indicator function that equals 1 if the condition inside is true and 0 otherwise. A higher Hits@10 value reflects the model’s ability to include the correct candidate within the top 10 ranked predictions.

**Accuracy (Acc)** The **Accuracy (Acc)** metric is used to evaluate performance on Knowledge Graph Question Answering (KGQA) tasks. For a dataset of queries, let  $\mathbf{1}[q]$  indicate whether the predicted answer for question  $q$  matches the ground truth. Accuracy is computed as:

$$\text{Acc} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbf{1}[q], \quad (25)$$

where  $\mathcal{Q}$  represents the set of all questions. A higher Accuracy score indicates the model’s effectiveness in selecting the correct answer from the set of options.

## E Full Results

The full, per-dataset results of MRR and Hits@10 of the zero-shot inference of the pre-trained MERRY model, the pre-trained ULTRA model, and best reported supervised SOTA baselines are presented in Table 6.

The detailed results from Table 1 are presented in Table 6, which also includes the outcomes for two ILPC datasets and IndER(NL-0) that are not covered in (Wang et al., 2024).

Group	Dataset	Training Graph			Validation Graph			Test Graph			SOTA
		Entities	Rels	Triples	Entities	Rels	Triples	Entities	Rels	Triples	
IndE(WN)	WN:v1	2746	9	5410	2746	9	5410	922	9	1618	<a href="#">Zhu et al. (2021)</a>
	WN:v2	6954	10	15262	6954	10	15262	2757	10	4011	<a href="#">Zhu et al. (2021)</a>
	WN:v3	12078	11	25901	12078	11	25901	5084	11	6327	<a href="#">Zhu et al. (2021)</a>
	WN:v4	3861	9	7940	3861	9	7940	12334	9	7084	<a href="#">Zhu et al. (2023)</a>
IndE(FB)	FB:v1	1594	180	4245	1594	180	4245	1093	180	1993	<a href="#">Zhu et al. (2023)</a>
	FB:v2	2608	200	9739	2608	200	9739	1660	200	4145	<a href="#">Zhu et al. (2021)</a>
	FB:v3	3668	215	17986	3668	215	17986	2501	215	7406	<a href="#">Zhu et al. (2021)</a>
	FB:v4	4707	219	27203	4707	219	27203	3352	219	11714	<a href="#">Zhu et al. (2023)</a>
IndE(NL)	NL:v1	3103	14	4687	3103	14	4687	833	14	833	<a href="#">Zhang and Yao (2022)</a>
	NL:v2	2564	88	8219	2564	88	8219	2086	88	4586	<a href="#">Zhang and Yao (2022)</a>
	NL:v3	4647	142	16393	4647	142	16393	3566	142	8048	<a href="#">Zhang and Yao (2022)</a>
	NL:v4	2092	76	7546	2092	76	7546	2795	76	7073	<a href="#">Zhang and Yao (2022)</a>
IndE(ILPC)	ILPC:small	10230	48	78616	6653	48	2908	6653	48	2902	<a href="#">Galkin et al. (2022b)</a>
	ILPC:large	46626	65	202446	29246	65	10179	29246	65	10184	<a href="#">Galkin et al. (2022b)</a>
IndER(FB)	FB-25	5190	163	91571	4097	216	17147	5716	4097	17147	<a href="#">Lee et al. (2023)</a>
	FB-50	5190	153	85375	4445	205	11636	3879	4445	11636	<a href="#">Lee et al. (2023)</a>
	FB-75	4659	134	62809	2792	186	9316	3106	2792	9316	<a href="#">Lee et al. (2023)</a>
	FB-100	4659	134	62809	2624	77	6987	2329	2624	6987	<a href="#">Lee et al. (2023)</a>
IndER(WK)	WK-25	12659	47	41873	3228	74	3391	1310	3228	3391	<a href="#">Lee et al. (2023)</a>
	WK-50	12022	72	82481	9328	93	9672	3224	9328	9672	<a href="#">Lee et al. (2023)</a>
	WK-75	6853	52	28741	2722	65	3430	1143	2722	3430	<a href="#">Lee et al. (2023)</a>
	WK-100	9784	67	49875	12136	97	13487	4496	12136	13487	<a href="#">Lee et al. (2023)</a>
IndER(NL)	NL-0	1814	134	7796	2026	112	2287	2026	112	2287	<a href="#">Lee et al. (2023)</a>
	NL-25	4396	106	17578	2230	146	2230	743	2230	2230	<a href="#">Lee et al. (2023)</a>
	NL-50	4396	106	17578	2335	119	2576	859	2335	2576	<a href="#">Lee et al. (2023)</a>
	NL-75	2607	96	11058	1578	116	1818	607	1606	1818	<a href="#">Lee et al. (2023)</a>
	NL-100	1258	55	7832	1709	53	2378	793	1709	2378	<a href="#">Lee et al. (2023)</a>

Table 5: 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	Supervised SOTA		ULTRA(3g)		MERRY	
		MRR	Hits@10	MRR	Hits@10	MRR	Hits@10
IndE(WN)	WN:v1	0.741	0.826	0.593	0.779	0.635	0.795
	WN:v2	0.704	0.798	0.620	0.752	0.654	0.783
	WN:v3	0.452	0.568	0.371	0.494	0.397	0.526
	WN:v4	0.661	0.743	0.484	0.687	0.562	0.710
IndE(FB)	FB:v1	0.457	0.589	0.486	0.657	0.478	0.628
	FB:v2	0.51	0.672	0.501	0.694	0.503	0.694
	FB:v3	0.476	0.637	0.482	0.644	0.478	0.636
	FB:v4	0.466	0.645	0.477	0.671	0.484	0.688
IndE(NL)	NL:v1	0.637	0.866	0.716	0.861	0.643	0.892
	NL:v2	0.419	0.601	0.525	0.719	0.558	0.753
	NL:v3	0.436	0.594	0.511	0.687	0.564	0.730
	NL:v4	0.363	0.556	0.490	0.701	0.498	0.691
IndE(ILPC)	ILPC:small	0.130	0.251	0.302	0.443	0.335	0.472
	ILPC:large	0.070	0.146	0.290	0.424	0.302	0.437
IndER(FB)	FB-25	0.133	0.271	0.383	0.633	0.363	0.616
	FB-50	0.117	0.218	0.330	0.536	0.330	0.540
	FB-75	0.189	0.325	0.391	0.594	0.377	0.574
	FB-100	0.223	0.371	0.438	0.631	0.443	0.638
IndER(WK)	WK-25	0.186	0.309	0.307	0.507	0.293	0.487
	WK-50	0.068	0.135	0.158	0.296	0.216	0.402
	WK-75	0.247	0.362	0.373	0.519	0.401	0.531
	WK-100	0.107	0.169	0.178	0.289	0.220	0.360
IndER(NL)	NL-0	0.269	0.431	0.342	0.523	0.351	0.536
	NL-25	0.334	0.501	0.387	0.538	0.406	0.601
	NL-50	0.281	0.453	0.398	0.549	0.376	0.530
	NL-75	0.261	0.464	0.348	0.527	0.344	0.550
	NL-100	0.309	0.506	0.442	0.631	0.462	0.666

Table 6: The full results (MRR and Hits@10) of MERRY, ULTRA, and the best-reported Supervised SOTA are presented across 27 datasets, highlighting their performance under both zero-shot inference and fine-tuning scenarios.