Efficient and Exact Global Attention on Latent Summaries for Knowledge Graph Reasoning

Chenxiao Lin[1], **Lei Wang**[2], **Yin Zhang**[2], **Wei Liu**[2], **Ye Luo**[3*] and **Qingqiang Wu**[1,3,4,5,6*]
[1]School of Film, Xiamen University, Xiamen, China
[2]Academy of Military Medical Sciences, Academy of Military Sciences
[3]School of Informatics, Xiamen University, Xiamen, China
[4]Institute of Artificial Intelligence, Xiamen University, Xiamen, China
[5]Xiamen Key Laboratory of Intelligent Storage and Computing, Xiamen University, Xiamen, China
[6]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan, Ministry of Culture and Tourism, Xiamen University, Xiamen, China
lcx064@gmail.com, {liuweilcf, aring2010, wangleienjoy}@163.com, {luoye, wuqq}@xmu.edu.cn

Abstract

Capturing global context through attention is essential for reasoning over knowledge graphs, especially when relevant entities are distant or disconnected. To scale attention to large graphs, recent methods replace Softmax with kernel feature mappings, reducing computational complexity to linear in the number of nodes. While efficient, these approximations tend to produce overly smooth attention scores, which can reduce discrimination between correct triplets and hard negative samples. Moreover, they exhibit scale-dependent shifts in attention entropy, making them sensitive to changes in graph size during inductive inference. In this paper, we introduce LaGR, a novel approach for integrating global information in knowledge graph reasoning. Rather than approximating interactions among all nodes, LaGR compresses the graph into a fixed, compact set of latent summaries and applies exact self-attention within this latent space. This change yields scale-invariant attention and stable performance across diverse data settings. In addition, we propose a node-adaptive residual fusion mechanism that dynamically balances local and global information at the node level, leading to more expressive representations. Extensive experiments on both transductive and inductive benchmarks show that LaGR substantially outperforms state-of-the-art baselines, demonstrating that exact attention over latent summaries is an efficient and effective way to capture global context in knowledge graph reasoning. Our implementation is available at https://github.com/XMU-KG/LaGR.

1 Introduction

Knowledge graphs (KGs) underpin many real-world applications, including recommender systems [Chicaiza and Díaz, 2021] and question answering [Wang *et al.*, 2024]. Despite their large scale, most KGs are incomplete, which motivates the need for effective reasoning methods to infer missing links. Early approaches, such as TransE [Bordes *et al.*, 2013] and RotatE [Sun *et al.*, 2019], represent entities as independent vectors and model relations through simple geometric transformations. More recently, research has shifted toward graph neural networks (GNNs), including R-GCN [Schlichtkrull *et al.*, 2018] and CompGCN [Vashishth *et al.*, 2020]. By explicitly modeling graph structure, GNNs allow entities to aggregate information from their local neighborhoods, enabling the capture of multi-hop relational patterns that are crucial for accurate link prediction.

Despite these advantages, standard GNNs face two well-known challenges: their message-passing mechanism limits information exchange between disconnected nodes [Liu *et al.*, 2024], and over-squashing, where information from distant nodes becomes distorted [Alon and Yahav, 2021]. To overcome these locality constraints, recent work has introduced global attention mechanisms that allow each node to attend to all others. However, full self-attention scales quadratically with the number of nodes, making it impractical for large graphs. To address this, state-of-the-art methods [Liu *et al.*, 2024; Li *et al.*, 2025a] commonly adopt linear attention mechanisms [Katharopoulos *et al.*, 2020], reducing the computational cost to linear complexity. This efficiency comes at a cost: linear attention approximates the softmax operation and discards query magnitude information, often leading to overly smooth attention distributions [Fan *et al.*, 2025]. Such smoothing weakens the model's ability to focus on highly relevant entities and makes it harder to distinguish correct triplets from hard negative samples in knowledge graph reasoning tasks. Furthermore, the behavior of linear attention varies with graph size, making it sensitive to changes in the node set during inductive inference.

In this work, we propose a different paradigm: instead of approximating attention over the full graph, we compress the graph context to enable exact attention. We introduce LaGR (Latent Graph Reasoning), a framework that resolves the trade-off between efficiency and expressivity. Our key

*Corresponding authors.

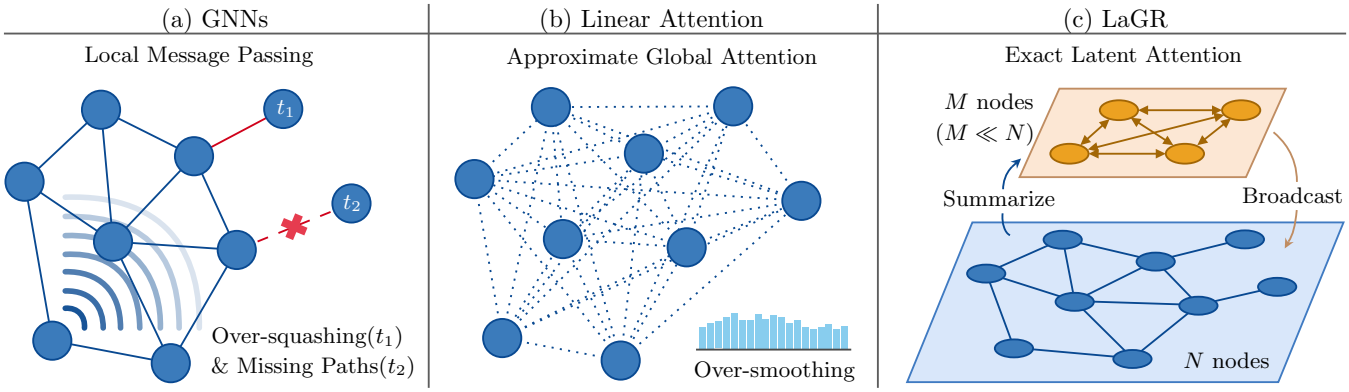

Figure 1: Comparison of local and global reasoning paradigms. (a) Standard GNNs suffer from over-squashing. (b) Linear attention methods mitigate this but result in over-smoothed attention distributions. (c) LaGR (Ours) compresses features into latent summaries for exact self-attention, enabling efficient and robust modeling of global context.

observation is that although the number of nodes N can be large, the global semantic context of a KG often lies in a much lower-dimensional space. Building on this insight, LaGR projects node representations into a small set of M latent summaries through weighted aggregation, where $M \ll N$. This latent space is sufficiently compact to support exact Softmax attention for modeling global correlations. After the attention and feed-forward computations, the resulting global context is broadcast back to the original nodes. As a result, LaGR preserves linear computational complexity with respect to N while maintaining robustness to variations in graph size, which linear attention methods often lack. For clarity, Figure 1 compares the architectures of standard GNNs, linear attention models, and LaGR.

We further observe that prior work typically fuses global and local information using a single trainable scalar shared across all nodes (e.g., $\alpha \cdot \text{global} + (1-\alpha) \cdot \text{local}$) [Li *et al.*, 2025a]. This uniform weighting overlooks the fact that different nodes may require different levels of global context. To address this limitation, we introduce a node-adaptive residual fusion strategy that dynamically adjusts the contribution of global context for each node through a learned projection over local and global features. This design allows LaGR to preserve local structural patterns captured by GNNs while selectively incorporating global signals to support long-range reasoning. Our contributions are summarized as follows:

- We propose LaGR, which offers a superior efficiency-effectiveness trade-off compared to linear attention approaches, mitigating over-smoothing and scale-dependent effects while maintaining linear computational complexity.
- We introduce a node-adaptive residual fusion strategy that replaces uniform scalar interpolation with a dynamic, node-specific weighting mechanism, improving feature expressiveness.
- We conduct extensive experiments on knowledge graph reasoning benchmarks in both transductive and inductive settings. Our results demonstrate that LaGR consistently outperforms existing GNN and efficient transformer baselines.

2 Related Work

Knowledge graph reasoning methods can be categorized by how they model interactions between entities and relations. Embedding-based approaches, such as TransE [Bordes *et al.*, 2013], DistMult [Yang *et al.*, 2015], and RotatE [Sun *et al.*, 2019], represent relations as translation operations in a vector space. Path-based methods propagate entity representations through local neighborhoods using message passing, as in CompGCN [Vashishth *et al.*, 2020], NBFNet [Zhu *et al.*, 2021], RED-GNN [Zhang and Yao, 2022], AdaProp [Zhang *et al.*, 2023], and ULTRA [Galkin *et al.*, 2024]. However, these approaches inherit well-known limitations of message-passing networks, including missing connected paths and over-squashing [Alon and Yahav, 2021]. Transformer-based methods, including HittER [Chen *et al.*, 2021], N-Former [Liu *et al.*, 2022], and SAttLE [Baghershahi *et al.*, 2023], typically convert graph structures into sequential representations during encoding, which may lead to the loss of important structural information.

More recently, hybrid architectures that combine GNNs and Transformers, such as KnowFormer [Liu *et al.*, 2024] and DuetGraph [Li *et al.*, 2025a], have been proposed to integrate local structural information with global graph context. To enable scalable global modeling, these methods often replace Softmax attention with linear attention mechanisms [Katharopoulos *et al.*, 2020]. While efficient, this substitution can result in over-smoothed feature representations and scale-dependent shifts in attention entropy. In contrast, we propose LaGR, which addresses these limitations while preserving scalability.

Notably, several existing methods also adopt a pooling–processing–unpooling paradigm for modeling KGs, including DiffPool [Ying *et al.*, 2018], Graph U-Nets [Gao and Ji, 2019], and KGPCL [Li *et al.*, 2025b]. These approaches reduce graph size by pruning paths, selecting representative nodes, or merging concepts, thereby transforming the original graph into a compact subgraph. However, such transformations remain constrained by the underlying graph structure and continue to inherit the limitations of message-passing networks. In contrast, our approach learns the pooling projec-

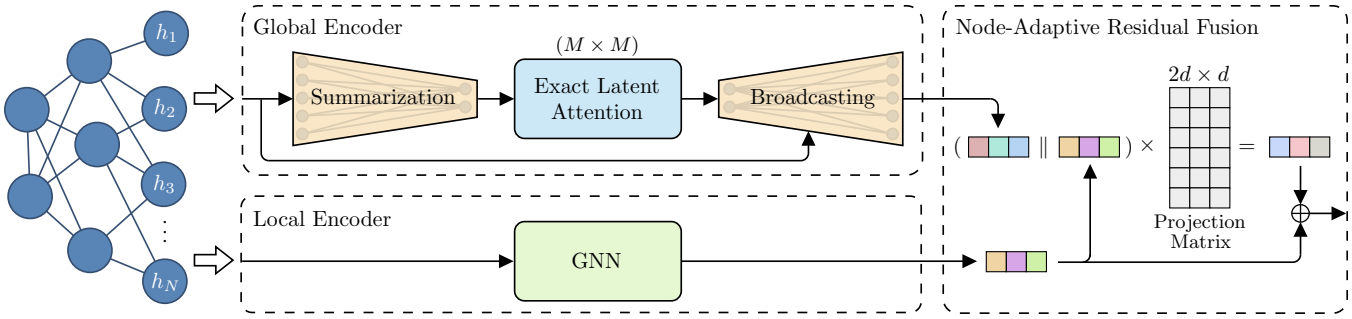

Figure 2: Overview of the LaGR framework. Each layer comprises three components. First, a local encoder based on a GNN captures neighborhood structure. Second, a global encoder compresses node features into latent summaries, applies exact attention in this compact space, and broadcasts the refined global context back to nodes. Third, a node-adaptive residual fusion module dynamically combines local and global features through a learnable projection. The symbols $\|$ and $\oplus$ denote feature concatenation and element-wise addition, respectively.

tion without relying on adjacency information as prior knowledge, enabling more effective modeling of global context.

3 Preliminary

3.1 Knowledge Graph

A KG is formally represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{V}$ denotes the set of entities, $\mathcal{R}$ the set of relations, and $\mathcal{E}$ the set of factual triplets. Each triplet is defined as $(h, r, t) \in \mathcal{E}$, where $h, t \in \mathcal{V}$ and $r \in \mathcal{R}$.

3.2 Knowledge Graph Reasoning

The goal of knowledge graph reasoning is to infer missing elements in query triplets. In this work, we focus on tail prediction, where queries take the form $(h, r, ?)$. Following common practice, head prediction queries $(?, r, t)$ are converted into tail prediction by introducing an inverse relation r^{-1}, yielding the equivalent form $(t, r^{-1}, ?)$.

3.3 Linear Attention

Linear attention is based on the assumption that the exponential function in Softmax attention can be approximated by a product of kernel functions [Fan *et al.*, 2025]. This reformulation decomposes the attention computation and reduces its complexity to linear time. As a representative graph embedding method, SGFormer [Wu *et al.*, 2023] implements linear attention as follows:

$$\boldsymbol{Z} = \mathrm{diag}\left(\mathbf{1} + \frac{1}{N}\boldsymbol{Q}(\boldsymbol{K}^\top \mathbf{1})\right)^{-1}\left(\boldsymbol{V} + \frac{1}{N}\boldsymbol{Q}(\boldsymbol{K}^\top \boldsymbol{V})\right). \quad (1)$$

Here, N denotes the number of nodes, $\mathrm{diag}(\cdot)$ constructs a diagonal matrix from its input, and $\cdot^\top$ denotes matrix transposition. $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ represent the query, key, and value matrices, respectively, with $\boldsymbol{Q}$ and $\boldsymbol{K}$ normalized by their Frobenius norms.

4 Methodology

As illustrated in Figure 2, LaGR processes the KG using both a local encoder and a global encoder in parallel, and then integrates their outputs through a node-adaptive residual fusion module. This architecture can be stacked across multiple layers. Section 4.1 describes the global encoder and analyzes its computational efficiency. Section 4.2 explains how the global context is incorporated into local representations via node-adaptive residual fusion. Section 4.3 presents the training objective of LaGR.

4.1 Global Encoder

To model long-range dependencies and the global semantic context that local message passing often fails to capture, we introduce a global encoder. As shown in Figure 2, this module follows a compress–attend–expand design that decouples the cost of global interaction from the number of graph nodes. It consists of three sequential steps: summarization, exact latent attention, and broadcasting.

Summarization. The summarization step projects N node features into M latent summaries, where $M \ll N$. Let $\boldsymbol{H} \in \mathbb{R}^{N\times d}$ denote the original node representations, where d is the feature dimension. Using projection matrices $\boldsymbol{W}_1 \in \mathbb{R}^{d\times d}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{d\times M}$, we compute a weight matrix:

$$\widetilde{\boldsymbol{W}}_{\text{sum}} = \sigma(\boldsymbol{H}\boldsymbol{W}_1)\boldsymbol{W}_2 \in \mathbb{R}^{N\times M}, \quad (2)$$

where $\sigma(\cdot)$ denotes a non-linear activation, and bias terms are omitted for simplicity. A *column-wise* softmax is then applied to normalize the weights, and the node features are aggregated into M latent summaries:

$$\widetilde{\boldsymbol{H}} = \widetilde{\boldsymbol{W}}_{\text{sum}}^\top \boldsymbol{H} \in \mathbb{R}^{M\times d}. \quad (3)$$

Exact latent attention. We next apply standard attention within this compact latent space, consisting of a softmax-based attention mechanism followed by a feedforward network [Vaswani *et al.*, 2017]. Operating in the reduced space allows exact attention while keeping computation efficient.

Broadcasting. Finally, the broadcasting step distributes the refined global information back to individual nodes using a mirrored projection. With projection matrices $\boldsymbol{W}_3 \in \mathbb{R}^{d\times d}$ and $\boldsymbol{W}_4 \in \mathbb{R}^{d\times M}$, we compute:

$$\widetilde{\boldsymbol{W}}_{\text{bro}} = \sigma(\boldsymbol{H}\boldsymbol{W}_3)\boldsymbol{W}_4 \in \mathbb{R}^{N\times M}. \quad (4)$$

In contrast to summarization, a *row-wise* softmax is applied to obtain node-specific mixing weights. The final global features for each node are then given by:

$$\boldsymbol{H}_{\text{glob}} = \widetilde{\boldsymbol{W}}_{\text{bro}}\hat{\boldsymbol{H}} \in \mathbb{R}^{N\times d}, \quad (5)$$

where $\hat{\boldsymbol{H}} \in \mathbb{R}^{M\times d}$ denotes the output of the latent attention module.

Theoretical Analysis of Latent Size

The Johnson-Lindenstrauss Lemma [Johnson *et al.*, 1984] states that N points can be projected into a space of dimension $k = \Omega(\frac{\log N}{\epsilon^2})$ while preserving pairwise distances within a factor of $(1 \pm \epsilon)$. This result suggests that the geometric structure of a KG can be preserved in a latent space whose dimension M grows logarithmically, rather than linearly, with graph size. Under this paradigm, the sensitivity of model performance to changes in graph scale is bounded by $O(\log N)$, yielding improved robustness in inductive reasoning.

From an efficiency perspective, M should be as small as possible. However, recent theoretical studies on deep sequence models highlight the risk of *rank collapse* [Joseph *et al.*, 2025], where representations degenerate toward a rank-one subspace, severely limiting expressiveness. When $M < d$, latent summarization becomes a lossy compression that discards orthogonal feature components, making distinct entities increasingly indistinguishable. This observation suggests a theoretical lower bound of $M \geq d$ to preserve sufficient representational capacity.

We next formalize LaGR as a learnable low-rank approximation of the full-graph Softmax attention matrix to characterize constraints on the projection matrix. Let $\mathcal{A} \in \mathbb{R}^{N\times N}$ denote the ideal all-pairs attention matrix, where $\mathcal{A}_{ij}$ measures the relevance of entity j to entity i. LaGR approximates this interaction through a latent bottleneck:

$$\mathcal{A} \approx \widetilde{\mathcal{A}} = \widetilde{\boldsymbol{W}}_{\text{bro}}\text{Attention}(\widetilde{\boldsymbol{W}}_{\text{sum}}^{\top}\boldsymbol{H}). \tag{6}$$

Because attention is computed in an $\mathbb{R}^{M\times M}$ latent space, the rank of $\widetilde{\mathcal{A}}$ is bounded by M. According to the Eckart-Young-Mirsky Theorem [Golub *et al.*, 1987], the optimal rank-M approximation of $\mathcal{A}$ with respect to the Frobenius norm $||\mathcal{A}-\widetilde{\mathcal{A}}||_F$ is achieved when the columns of $\widetilde{\boldsymbol{W}}_{\text{sum}}$ align with the top-M singular vectors of $\mathcal{A}$. Since these singular vectors are unavailable during training, we instead encourage the projection to implicitly capture distinct principal components across latent summaries. In this view, latent summaries are not treated as simple clusters but as basis vectors spanning the principal semantic manifold of the graph.

To support this objective, we introduce an auxiliary orthogonality regularization that promotes diversity among latent summaries by penalizing correlations in the projection weights:

$$\mathcal{L}_{\text{orth}} = \left\| \widetilde{\boldsymbol{W}}_{\text{sum}}^{\top}\widetilde{\boldsymbol{W}}_{\text{sum}} - \boldsymbol{I} \right\|_F^2. \tag{7}$$

Complexity Analysis

This paragraph analyzes the computational efficiency of different approaches for modeling global graph context. For clarity, we omit the feedforward network, which is shared by all methods. Standard softmax attention computes pairwise interactions between all nodes, leading to a quadratic complexity of $O(N^2d)$. To scale to large KGs, linear attention methods approximate softmax attention using kernel-based formulations and the associativity of matrix multiplication, reducing the complexity to $O(Nd^2)$. In LaGR, both the summarization and broadcasting modules have a complexity of $O(Nd^2 + NMd)$, while exact attention in the latent space costs $O(M^2d)$. Since $M \ll N$, the overall complexity of the global encoder is dominated by the projection operations, resulting in a total complexity of $O(Nd^2 + NMd)$.

4.2 Node-Adaptive Residual Fusion

The need for global context varies across the graph. Some entities can be accurately represented using only their immediate neighbors, while others are affected by over-squashing or missing paths and, therefore, require broader contextual information. To account for this heterogeneity, we dynamically adjust the contribution of global features for each node through a mutual attention mechanism implemented with a fully connected layer and a residual connection. Let $\boldsymbol{H}_{\text{glob}}$ and $\boldsymbol{H}_{\text{loc}}$ be the global and local feature matrices, respectively. The fused representations are computed as:

$$\boldsymbol{H}_{\text{proj}} = (\boldsymbol{H}_{\text{glob}} \| \boldsymbol{H}_{\text{loc}})\boldsymbol{W}_{\text{proj}} + \boldsymbol{H}_{\text{loc}}, \tag{8}$$

where $\|$ denotes feature concatenation along the column dimension, and $\boldsymbol{W}_{\text{proj}} \in \mathbb{R}^{2d\times d}$ is a learnable projection matrix.

Expressivity Analysis

Here, we compare our node-adaptive residual fusion strategy with scalar interpolation [Li *et al.*, 2025a], which combines global and local features as

$$\boldsymbol{H}_{\text{scal}} = \alpha\boldsymbol{H}_{\text{loc}} + (1-\alpha)\boldsymbol{H}_{\text{glob}}, \tag{9}$$

where $\alpha \in [0, 1]$ is a learnable scalar obtained via a sigmoid function.

For a direct comparison, we rewrite our fusion operation in Equation 8 as

$$\boldsymbol{H}_{\text{proj}} = \boldsymbol{H}_{\text{loc}}(\boldsymbol{W}_{\text{b}} + \boldsymbol{I}) + \boldsymbol{H}_{\text{glob}}\boldsymbol{W}_{\text{a}}, \tag{10}$$

where $\boldsymbol{W}_{\text{a}} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{W}_{\text{b}} \in \mathbb{R}^{d\times d}$ are column partitions of the projection matrix $\boldsymbol{W}_{\text{proj}}$, such that $(\boldsymbol{W}_{\text{a}} \| \boldsymbol{W}_{\text{b}}) = \boldsymbol{W}_{\text{proj}}$.

Let $\mathcal{H}_{\text{scal}}$ and $\mathcal{H}_{\text{proj}}$ denote the hypothesis spaces induced by $\boldsymbol{H}_{\text{scal}}$ and $\boldsymbol{H}_{\text{proj}}$, respectively. Since scalar interpolation is a special case of the projection-based formulation, it follows that $\mathcal{H}_{\text{scal}} \subset \mathcal{H}_{\text{proj}}$. This shows that our method strictly generalizes the expressiveness of scalar interpolation.

4.3 Training Objective

At the output layer, the final node representation is mapped to a scalar score using a multi-layer perceptron (MLP) $f : \mathbb{R}^d \to \mathbb{R}$, which represents the probability that a candidate entity is a valid answer. Given a query triplet $(h, r, ?)$ and a candidate tail entity e, we denote the normalized score as $s(e|h, r) \in [0, 1]$. Model parameters are trained by minimizing a negative sampling loss for each fact (h, r, t):

$$\mathcal{L}_{\text{neg}} = -\log(s(t|h, r)) - \sum_{e\in\mathcal{N}_{hr}} \log(1 - s(e|h, r)), \tag{11}$$

where $\mathcal{N}_{hr} = \{e \in \mathcal{V} | (h, r, e) \notin \mathcal{E}\}$ is the set of negative candidates for the query $(h, r, ?)$.

The final training objective combines the negative sampling loss with the orthogonality regularization:

$$\mathcal{L} = \mathcal{L}_{\text{neg}} + \mathcal{L}_{\text{orth}}. \tag{12}$$

Methods	v1			v2			v3			v4		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
FB15k-237												
DRUM [Sadeghian *et al.*, 2019]	0.333	24.7	47.4	0.395	28.4	59.5	0.402	30.8	57.1	0.410	30.9	59.3
NBFNet [Zhu *et al.*, 2021]	0.442	33.5	57.4	0.514	42.1	68.5	0.476	38.4	63.7	0.453	36.0	62.7
RED-GNN [Zhang and Yao, 2022]	0.369	30.2	48.3	0.469	38.1	62.9	0.445	35.1	50.3	0.442	34.0	62.1
A*Net [Zhu *et al.*, 2023]	0.457	38.1	58.9	0.510	41.9	67.2	0.476	38.9	62.9	0.466	36.5	64.5
AdaProp [Zhang *et al.*, 2023]	0.310	19.1	55.1	0.471	37.2	65.9	0.471	37.7	63.7	0.454	35.3	63.8
Ingram [Lee *et al.*, 2023]	0.293	16.7	49.3	0.274	16.3	48.2	0.233	14.0	40.8	0.214	11.4	39.7
KnowFormer [Liu *et al.*, 2024]	0.466	37.8	60.6	0.532	43.3	70.3	0.494	40.0	65.9	0.480	38.3	65.3
DuetGraph [Li *et al.*, 2025a]	0.507	42.7	63.2	0.549	44.8	72.9	0.518	42.3	69.9	0.501	39.8	67.0
LaGR	**0.513**	**43.4**	**63.9**	**0.566**	**46.7**	**73.6**	**0.536**	**44.7**	**70.4**	**0.517**	**41.7**	**70.3**
WN18RR												
DRUM [Sadeghian *et al.*, 2019]	0.666	61.3	77.7	0.646	59.5	74.7	0.380	33.0	47.7	0.627	58.6	70.2
NBFNet [Zhu *et al.*, 2021]	0.741	69.5	**82.6**	0.704	65.1	79.8	0.452	39.2	56.8	0.641	60.8	69.4
RED-GNN [Zhang and Yao, 2022]	0.701	65.3	79.9	0.690	63.3	78.0	0.427	36.8	52.4	0.651	60.6	72.1
A*Net [Zhu *et al.*, 2023]	0.727	68.2	81.0	0.704	64.9	80.3	0.441	38.6	54.4	0.661	61.6	74.3
AdaProp [Zhang *et al.*, 2023]	0.733	66.8	80.6	0.715	64.2	**82.6**	0.474	39.6	58.8	0.662	61.1	75.5
Ingram [Lee *et al.*, 2023]	0.277	13.0	60.6	0.236	11.2	48.0	0.230	11.6	46.6	0.118	4.1	25.9
KnowFormer [Liu *et al.*, 2024]	0.752	71.5	81.9	0.709	65.6	81.7	0.467	40.6	57.1	0.646	60.9	72.7
DuetGraph [Li *et al.*, 2025a]	0.758	72.1	81.7	0.719	66.7	81.1	0.501	44.3	62.2	0.662	62.1	73.1
LaGR	**0.766**	**73.1**	**82.6**	**0.724**	**67.6**	82.0	**0.517**	**45.2**	**62.7**	**0.671**	**62.8**	**76.1**
NELL-995												
NBFNet [Zhu *et al.*, 2021]	0.584	50.0	79.5	0.410	27.1	63.5	0.425	26.2	60.6	0.287	25.3	59.1
RED-GNN [Zhang and Yao, 2022]	0.637	52.2	86.6	0.419	31.9	60.1	0.436	34.5	59.4	0.363	25.9	60.7
AdaProp [Zhang *et al.*, 2023]	0.644	52.2	88.6	0.452	34.4	65.2	0.435	33.7	61.8	0.366	24.7	60.7
Ingram [Lee *et al.*, 2023]	0.697	57.5	86.5	0.358	25.3	59.6	0.308	19.9	50.9	0.221	12.4	44.0
KnowFormer [Liu *et al.*, 2024]	0.827	77.0	93.0	0.465	35.7	65.7	0.478	37.8	65.7	0.378	26.7	59.8
DuetGraph [Li *et al.*, 2025a]	0.850	78.5	96.5	0.543	44.4	69.1	0.535	43.2	72.6	0.464	35.4	**68.4**
LaGR	**0.855**	**79.0**	**96.9**	**0.558**	**44.6**	**76.1**	**0.550**	**45.6**	**73.2**	**0.472**	**36.9**	63.1

Table 1: Inductive knowledge graph reasoning performance across 12 subsets from FB15k-237, WN18RR, and NELL-995. Best-performing results are shown in **bold**, and second-best results are underlined.

5 Experiments

This section first describes the experimental setup. We then present a comprehensive evaluation that (1) assesses LaGR's performance on both inductive and transductive reasoning tasks, (2) analyzes the contribution of its key components, (3) evaluates training and inference efficiency, (4) examines score distributions under different attention mechanisms, and (5) studies the model's sensitivity to the hyperparameter M, which determines the latent summary size.

5.1 Experimental Setup

Datasets. We evaluate LaGR on both transductive and inductive benchmarks. For the transductive setting, we use four KGs of varying sizes: FB15k-237 [Toutanova and Chen, 2015], WN18RR [Dettmers *et al.*, 2018], NELL-995 [Xiong *et al.*, 2017], and YAGO3-10 [Mahdisoltani *et al.*, 2013]. In the inductive setting, the training and test graphs contain disjoint entity sets while sharing the same relations. We adopt inductive benchmarks derived from FB15k-237, WN18RR, and NELL-995, each split into four variants, yielding a total of 12 subsets. For all datasets, we use the standard data splits established in prior work [Liu *et al.*, 2024].

Baselines. We compare LaGR against a diverse set of representative methods, including embedding-based models, path-based approaches, transformer-based models, hybrid methods, and other implementations such as meta-learning, geometric representation learning, and regularization-based optimization. The complete list of baselines is reported in Tables 1 and 2.

Evaluation metrics. We measure performance using standard metrics for knowledge graph reasoning: Mean Reciprocal Rank (MRR) and Hit Rate at k (H@k with $k = 1, 10$), where higher values indicate better performance. Following common practice, we report results under the filtered setting [Bordes *et al.*, 2013], which removes known true triples from the ranking candidates.

5.2 Performance

Following DuetGraph [Li *et al.*, 2025a], we adopt a coarse-to-fine optimization strategy. Specifically, a pre-trained coarse model partitions entities into high- and low-score subsets, after which the proposed model refines the final rankings within each subset. For all baseline methods, we report results as published in their original papers.

Inductive datasets. Table 1 reports performance comparisons on twelve inductive benchmark subsets. Across these settings, LaGR consistently achieves the best or near-best

Methods	FB15k-237			WN18RR			NELL-995			YAGO3-10		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
Embedding-based methods												
TransE [Bordes *et al.*, 2013]	0.330	23.2	52.6	0.222	1.4	52.8	0.507	42.4	64.8	0.510	41.3	68.1
DistMult [Yang *et al.*, 2015]	0.358	26.4	55.0	0.455	41.0	54.4	0.510	43.8	63.6	0.566	49.1	70.4
RotatE [Sun *et al.*, 2019]	0.337	24.1	53.0	0.477	42.8	57.1	0.508	44.8	60.8	0.495	40.2	67.0
HousE [Li *et al.*, 2022a]	0.361	26.6	55.1	0.511	46.5	60.2	0.519	45.8	61.8	0.571	49.1	71.4
Path-based methods												
DRUM [Sadeghian *et al.*, 2019]	0.343	25.5	51.6	0.486	42.5	58.6	0.532	46.0	66.2	0.531	45.3	67.6
CompGCN [Vashishth *et al.*, 2020]	0.355	26.4	53.5	0.479	44.3	54.6	0.463	38.3	59.6	0.421	39.2	57.7
RNNLogic [Qu *et al.*, 2021]	0.344	25.2	53.0	0.483	44.6	55.8	0.516	46.3	57.8	0.554	50.9	62.2
NBFNet [Zhu *et al.*, 2021]	0.415	32.1	59.9	0.551	49.7	66.6	0.525	45.1	63.9	0.563	48.0	70.8
RED-GNN [Zhang and Yao, 2022]	0.374	28.3	55.8	0.533	48.5	62.4	0.543	47.6	65.1	0.556	48.3	68.9
A*Net [Zhu *et al.*, 2023]	0.411	32.1	58.6	0.549	49.5	65.9	0.521	44.7	63.1	0.556	47.0	70.7
AdaProp [Zhang *et al.*, 2023]	0.417	33.1	58.5	0.562	49.9	67.1	0.554	49.3	65.5	0.573	51.0	68.5
ULTRA [Galkin *et al.*, 2024]	0.368	27.2	56.4	0.480	41.4	61.4	0.509	44.1	66.0	0.557	47.1	71.0
Transformer-based methods												
HittER [Chen *et al.*, 2021]	0.373	27.9	55.8	0.503	46.2	58.4	0.518	43.7	65.9	0.339	25.1	50.8
KGT5 [Saxena *et al.*, 2022]	0.276	21.0	41.4	0.508	48.7	54.4	-	-	-	0.426	36.8	52.8
N-Former [Liu *et al.*, 2022]	0.373	27.9	55.6	0.489	44.6	58.1	-	-	-	-	-	-
SAttLE [Baghershahi *et al.*, 2023]	0.360	26.8	54.5	0.491	45.4	55.8	0.512	42.2	66.0	0.475	36.7	68.2
Other methods												
MetaSD [Li *et al.*, 2022b]	0.391	30.0	57.1	0.491	44.7	57.0	0.516	45.5	61.5	OOM	OOM	OOM
TuckeER-IVR [Xiao and Cao, 2024]	0.368	27.4	55.5	0.501	46.0	57.9	0.505	42.8	63.7	0.581	50.8	71.2
HyperKGR [Liu, 2025]	0.417	33.1	58.5	0.565	51.2	66.6	0.547	48.2	64.8	-	-	-
Hybrid methods												
KnowFormer [Liu *et al.*, 2024]	0.430	34.3	60.8	0.579	52.8	68.7	0.566	50.2	67.5	0.615	54.7	73.4
DuetGraph [Li *et al.*, 2025a]	0.453	36.1	62.4	0.593	54.2	**69.9**	0.590	52.1	**71.2**	0.631	56.1	**74.8**
LaGR	**0.458**	**36.6**	**62.6**	**0.594**	**54.5**	68.7	**0.593**	**52.4**	69.4	**0.636**	**56.8**	**74.8**

Table 2: Transductive knowledge graph reasoning performance across four datasets. A dash ("-") indicates results that are unavailable due to insufficient information for reproduction. OOM stands for out-of-memory.

results on almost all evaluation metrics. In particular, on FB15k-237, LaGR outperforms existing methods across all versions, improving upon the strongest baseline, DuetGraph. Although some methods, such as AdaProp and DuetGraph, perform competitively on specific metrics (e.g., Hits@10 on WN18RR v2 and NELL-995 v4), LaGR demonstrates more stable performance across datasets and splits. These results indicate that modeling global context through a fixed, compact latent summary space is effective for inductive knowledge graph reasoning.

Transductive datasets. Table 2 summarizes results on four transductive benchmarks. LaGR achieves state-of-the-art performance on most metrics across all datasets. A clear pattern is that hybrid methods, including KnowFormer, DuetGraph, and LaGR, generally outperform single-paradigm methods. For example, on FB15k-237, LaGR attains an MRR of 0.458, exceeding both embedding-based models such as HousE (0.361) and path-based methods such as AdaProp (0.417). On the larger YAGO3-10 dataset, LaGR maintains strong performance, achieving the highest MRR (0.636) and Hits@1 (56.8). While DuetGraph achieves the best Hits@10 on WN18RR and NELL-995, LaGR remains the strongest overall in terms of ranking accuracy, as reflected by MRR. These results demonstrate the effectiveness and robustness of LaGR in transductive reasoning settings.

5.3 Ablation Study

Table 3 reports a component-wise ablation study on the FB15k-237 inductive subsets, examining the impact of key design choices in LaGR. Across all subsets and evaluation metrics, the original LaGR configuration (first row) achieves the strongest overall performance.

Modeling global context through latent summaries consistently outperforms linear attention approximation (Appr), indicating the benefit of projecting a dynamic graph into a fixed latent space for inductive reasoning. The role of orthogonality regularization is particularly evident: removing this constraint (denoted by •) leads to a noticeable performance degradation, especially on Hits@10. For example, on v3, Hits@10 drops from 70.4 to 67.9 without regularization, suggesting that promoting diverse and non-redundant latent summaries is important for effective global context modeling.

In addition, the node-adaptive projection (Proj) strategy shows strong generalization across settings. It consistently outperforms scalar interpolation (Scal), regardless of whether global information is captured through latent summaries or linear attention approximation. This result supports the use of a learnable, node-specific fusion mechanism over a uniform weighting scheme when integrating local and global features.

Overall, these findings validate the architectural design of LaGR and highlight the complementary roles of latent global

Global Context		Fusion Strategy		v1			v2			v3			v4		
Lat	**Appr**	**Proj**	**Scal**	**MRR**	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**	**MRR**	**H@1**	**H@10**
✓		✓		**0.513**	**43.4**	**63.9**	**0.566**	**46.7**	**73.6**	**0.536**	**44.7**	**70.4**	**0.517**	**41.7**	**70.3**
•		✓		0.506	42.4	62.0	0.560	46.1	71.2	0.528	43.3	67.9	0.511	41.2	69.0
✓			✓	0.495	41.5	62.6	0.562	46.5	71.9	0.531	43.5	68.7	0.510	40.7	69.5
•			✓	0.492	41.2	61.6	0.559	46.0	70.9	0.526	42.6	67.6	0.508	40.3	68.8
	✓	✓		0.492	41.1	62.4	0.556	45.6	71.7	0.525	42.6	68.2	0.515	41.3	69.5
	✓		✓	0.489	41.0	61.2	0.554	45.4	71.7	0.523	42.3	68.4	0.506	40.6	69.2

Table 3: Component-wise ablation study on four FB15k-237 subsets. **Lat** and **Appr** denote capturing global context via latent summaries and linear attention approximation, respectively. **Proj** and **Scal** refer to fusing local and global features using node-adaptive projection and scalar interpolation. A checkmark (✓) indicates that the corresponding component is included, while a dot (•) indicates the use of latent summaries without orthogonality regularization. Best results are shown in **bold**.

Methods	FB15k-237			WN18RR			NELL-995			YAGO3-10		
	Param	**Train**	**Infer**	**Param**	**Train**	**Infer**	**Param**	**Train**	**Infer**	**Param**	**Train**	**Infer**
KnowFormer [Liu *et al.*, 2024]	6.1 M	700	49	342 K	512	18	5.1 M	1665	34	1 M	44030	171
DuetGraph [Li *et al.*, 2025a]	6.1 M	709	59	**336 K**	581	26	5.1 M	1749	42	997 K	40530	185
LaGR	**4.7 M**	**263**	**27**	354 K	**396**	**13**	**4.0 M**	**1573**	**30**	**850 K**	**36728**	**130**

Table 4: Efficiency comparison of methods across transductive datasets of different scales. **Param** denotes the number of trainable parameters (K: thousands, M: millions). **Train** and **Infer** indicate per-epoch training time and inference time, measured in seconds.

modeling and adaptive projection in improving inductive reasoning performance.

5.4 Efficiency Analysis

Table 4 compares the computational efficiency of LaGR with two hybrid methods on transductive datasets of different scales. Since these methods reach their best performance within a similar number of epochs and do not employ early stopping, we report the training time per epoch.

Overall, LaGR is consistently more efficient in both model size and runtime. It uses the fewest trainable parameters on three of the four datasets, with about a 23% reduction on FB15k-237 and NELL-995 compared to the baselines. This compact design leads to clear runtime benefits: LaGR achieves the fastest training per epoch and the lowest inference time across all benchmarks. On FB15k-237, it cuts training time by over 60% and reduces inference time by nearly half relative to DuetGraph. Importantly, LaGR retains this advantage on the large-scale YAGO3-10 dataset, showing that its performance gains come from a more efficient and lightweight design rather than increased computational cost.

5.5 Analysis of Score Over-Smoothing

To assess the extent of over-smoothing, we analyze the variance of output score distributions on the four inductive subsets of FB15k-237. In this analysis, over-smoothing refers to the collapse of representations into a narrow range, which results in similar prediction scores for different candidate triples. We compute the variance of these scores as an indicator of the model's ability to produce a discriminative distribution in which the ground-truth entity is separated from negative candidates.

As illustrated in Figure 3, LaGR consistently produces higher score variance than all hybrid baseline methods across the evaluated subsets, whereas KnowFormer yields the lowest

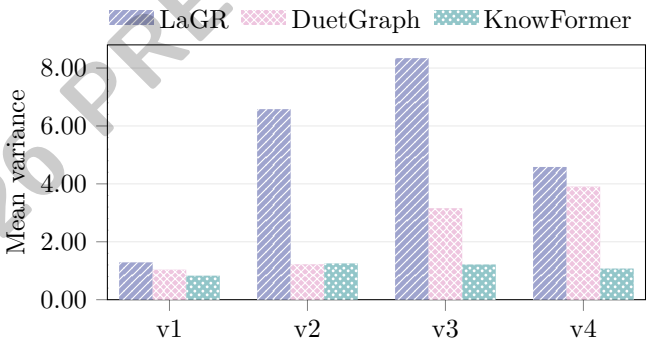

Figure 3: Comparison of the mean variance of output scores on inductive subsets of FB15k-237.

variance. Notably, on the v2 and v3 subsets, LaGR achieves variances of 6.6 and 8.3, exceeding those of the second-best model by 424.9% and 164.1%, respectively. These results suggest that LaGR better preserves feature distinctiveness, alleviating over-smoothing in deep graph reasoning models and contributing to more reliable ranking performance.

6 Conclusion

This paper proposes LaGR, a novel paradigm for capturing global interactions in knowledge graphs with linear computational complexity in the number of nodes. LaGR addresses the over-smoothing and scale-dependent shifts in attention entropy observed in prior linear attention approaches. In addition, we introduce a node-adaptive residual fusion strategy for integrating global and local representations, which improves the expressiveness of the fused features. Extensive experiments demonstrate that LaGR achieves state-of-the-art performance while remaining efficient and robust across diverse settings.

Acknowledgments

This work is supported by the Solfeggio ear training intelligent robot and cloud platform research and development project for music education (No.2024CXY0102), the 3D visualization digital twin integrated control system (No.2023CXY0111), the public technology service platform project of Xiamen City (No.3502Z20231043) and Fujian Provincial Science and Technology Major Project (No.2024HZ022003)

References

[Alon and Yahav, 2021] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[Baghershahi *et al.*, 2023] Peyman Baghershahi, Reshad Hosseini, and Hadi Moradi. Self-attention presents low-dimensional knowledge graph embeddings for link prediction. *Knowl. Based Syst.*, 260:110124, 2023.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[Chen *et al.*, 2021] Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. Hitter: Hierarchical transformers for knowledge graph embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 10395–10407, 2021.

[Chicaiza and Díaz, 2021] Janneth Chicaiza and Priscila Valdiviezo Díaz. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Inf.*, 12(6):232, 2021.

[Dettmers *et al.*, 2018] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[Fan *et al.*, 2025] Qihang Fan, Huaibo Huang, Yuang Ai, and Ran He. Rectifying magnitude neglect in linear attention. In *ICCV*, 2025.

[Galkin *et al.*, 2024] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[Gao and Ji, 2019] Hongyang Gao and Shuiwang Ji. Graph u-nets. In *international conference on machine learning*, pages 2083–2092. PMLR, 2019.

[Golub *et al.*, 1987] Gene H Golub, Alan Hoffman, and Gilbert W Stewart. A generalization of the eckart-young-mirsky matrix approximation theorem. *Linear Algebra and its applications*, 88:317–327, 1987.

[Johnson *et al.*, 1984] William B Johnson, Joram Lindenstrauss, et al. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[Joseph *et al.*, 2025] Federico Arangath Joseph, Jerome Sieber, Melanie Nicole Zeilinger, and Carmen Amo Alonso. Lambda-skip connections: the architectural component that prevents rank collapse. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[Katharopoulos *et al.*, 2020] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[Lee *et al.*, 2023] Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. Ingram: Inductive knowledge graph embedding via relation graphs. In *International conference on machine learning*, pages 18796–18809. PMLR, 2023.

[Li *et al.*, 2022a] Rui Li, Jianan Zhao, Chaozhuo Li, Di He, Yiqi Wang, Yuming Liu, Hao Sun, Senzhang Wang, Weiwei Deng, Yanming Shen, et al. House: Knowledge graph embedding with householder parameterization. In *International conference on machine learning*, pages 13209–13224. PMLR, 2022.

[Li *et al.*, 2022b] Yunshui Li, Junhao Liu, Min Yang, and Chengming Li. Self-distillation with meta learning for knowledge graph completion. In *EMNLP*, 2022. 2048–2054.

[Li *et al.*, 2025a] Jin Li, Zezhong Ding, and Xike Xie. Duetgraph: Coarse-to-fine knowledge graph reasoning with dual-pathway global-local fusion. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[Li *et al.*, 2025b] Juan Li, Wen Zhang, Zhiqiang Liu, Mingchen Tu, Mingyang Chen, Ningyu Zhang, and Shijian Li. Knowledge graph pooling and unpooling for concept abstraction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5364–5374, 2025.

[Liu *et al.*, 2022] Yang Liu, Zequn Sun, Guangyao Li, and Wei Hu. I know what you do not know: Knowledge graph embedding via co-distillation learning. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 1329–1338, 2022.

[Liu *et al.*, 2024] Junnan Liu, Qianren Mao, Weifeng Jiang, and Jianxin Li. Knowformer: Revisiting transformers for knowledge graph reasoning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[Liu, 2025] Lihui Liu. Hyperkgr: Knowledge graph reasoning in hyperbolic space with graph neural network encoding symbolic path. In *Proceedings of the 2025 Conference*

on Empirical Methods in Natural Language Processing, pages 25188–25199, 2025.

[Mahdisoltani *et al.*, 2013] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*, 2013.

[Qu *et al.*, 2021] Meng Qu, Jun-Kun Chen, Louis-Pascal A. C. Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[Sadeghian *et al.*, 2019] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in neural information processing systems*, 32, 2019.

[Saxena *et al.*, 2022] Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. Sequence-to-sequence knowledge graph completion and question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2814–2828. Association for Computational Linguistics, 2022.

[Schlichtkrull *et al.*, 2018] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer, 2018.

[Sun *et al.*, 2019] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[Toutanova and Chen, 2015] Kristina Toutanova and Danqi Chen. 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, 2015.

[Vashishth *et al.*, 2020] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 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.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2024] Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 19206–19214, 2024.

[Wu *et al.*, 2023] Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. Sgformer: Simplifying and empowering transformers for large-graph representations. *Advances in Neural Information Processing Systems*, 36:64753–64773, 2023.

[Xiao and Cao, 2024] Changyi Xiao and Yixin Cao. Knowledge graph completion by intermediate variables regularization. In *NeurIPS*, 2024.

[Xiong *et al.*, 2017] Wenhan Xiong, Thien Hoang, and William Yang Wang. Deeppath: A reinforcement learning method for knowledge graph reasoning. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 564–573. Association for Computational Linguistics, 2017.

[Yang *et al.*, 2015] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[Ying *et al.*, 2018] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018.

[Zhang and Yao, 2022] Yongqi Zhang and Quanming Yao. Knowledge graph reasoning with relational digraph. In *Proceedings of the ACM web conference 2022*, pages 912–924, 2022.

[Zhang *et al.*, 2023] Yongqi Zhang, Zhanke Zhou, Quanming Yao, Xiaowen Chu, and Bo Han. 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*, pages 3446–3457, 2023.

[Zhu *et al.*, 2021] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in neural information processing systems*, 34:29476–29490, 2021.

[Zhu *et al.*, 2023] Zhaocheng Zhu, Xinyu Yuan, Michael Galkin, Louis-Pascal Xhonneux, Ming Zhang, Maxime Gazeau, and Jian Tang. A* net: A scalable path-based reasoning approach for knowledge graphs. *Advances in Neural Information Processing Systems*, 36:59323–59336, 2023.