NumCoKE: Ordinal-Aware Numerical Reasoning over Knowledge Graphs with Mixture-of-Experts and Contrastive Learning

Ming Yin[1,2,3], **Zongsheng Cao**[4], **Qiqing Xia**[1,2,3], **Chenyang Tu**[1,2], **Neng Gao**[1,2,†]
[1]Institute of Information Engineering, Chinese Academy of Sciences.
[2]State Key Laboratory of Cyberspace Security Defense.
[3]School of Cyber Security, University of Chinese Academy of Sciences.
[4]Department of Information Science, Tsinghua University.
agiczsr@gmail.com, {yinming, xiaqiqing, tuchenyang, gaoneng}@iie.ac.cn

Abstract

Knowledge graphs (KGs) serve as a vital backbone for a wide range of AI applications, including natural language understanding and recommendation. A promising yet underexplored direction is *numerical reasoning* over KGs, which involves inferring new facts by leveraging not only symbolic triples but also numerical attribute values (e.g., *length*, *weight*). However, existing methods fall short in two key aspects: (1) **Incomplete semantic integration**: Most models struggle to jointly encode entities, relations, and numerical attributes in a unified representation space, limiting their ability to extract relation-aware semantics from numeric information. (2) **Ordinal indistinguishability**: Due to subtle differences between close values and sampling imbalance, models often fail to capture fine-grained ordinal relationships (e.g., longer, heavier), especially in the presence of hard negatives. To address these challenges, we propose **NumCoKE**—a numerical reasoning framework for KGs based on Mixture-of-Experts and Ordinal Contrastive Embedding. To overcome (C1), we introduce a Mixture-of-Experts Knowledge-Aware (MoEKA) encoder that jointly aligns symbolic and numeric components into a shared semantic space, while dynamically routing attribute features to relation-specific experts. To handle (C2), we propose Ordinal Knowledge Contrastive Learning (OKCL), which constructs ordinal-aware positive and negative samples using prior knowledge, enabling the model to better discriminate subtle semantic shifts. Extensive experiments on three public KG benchmarks demonstrate that **NumCoKE** consistently outperforms competitive baselines across diverse attribute distributions, validating its superiority in both semantic integration and ordinal reasoning.

Introduction

Knowledge graphs (KGs) represent structured factual knowledge as triples of entities and relations, and have become a key foundation for various AI applications, including recommendation systems (Cao et al. 2022a; Li et al. 2020), natural language processing (Guo et al. 2023; Lu et al. 2023; Sun et al. 2023), and multimodal tasks (Wang et al. 2023; Li et al. 2023). A crucial yet underexplored capability of KGs is *numerical reasoning*, the ability to infer facts involving quantitative comparisons or numerical ordering (e.g., *"the Nile is longer than the Amazon"*). This ability is particularly valuable for information services that require precise numerical judgments, such as fine-grained product recommendations or numerical question answering.

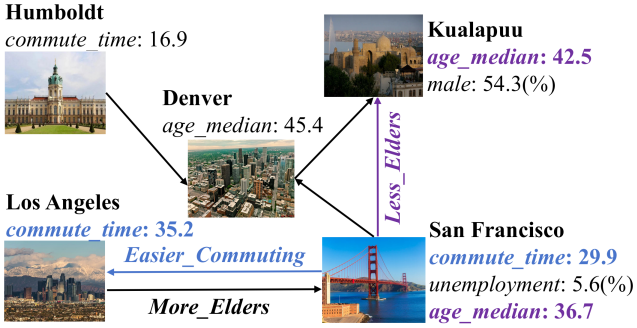


Figure 1: Example of KG-based numerical reasoning. Same colored text indicates strong related relation and attribute.

To enhance numerical reasoning, prior studies have explored integrating attribute values into knowledge graph embedding (KGE) frameworks. Some directly embed numerical entities into continuous spaces (Bai et al. 2023), while others employ graph neural networks (Vashishth et al. 2020) or augment traditional KGE with attribute-aware components (Kim et al. 2023). Despite these advances, existing methods face two persistent and intertwined challenges:

C1 Incomplete semantic integration. Current models often fail to effectively capture the joint semantics of entities, relations, and numerical attributes. In practice, the relevance of numerical attributes is highly dependent on the relational context. For instance, given the query (*San Francisco*, *Less_Elders*, *x*), the attribute *median_age* becomes critical; whereas in (*San Francisco*, *Easier_Commuting*, *x*), the key attribute is *commute_time*. Models lacking context-aware alignment treat numerical features uniformly, leading to semantic incompleteness and degraded reasoning performance. (see Figure 1).

C2 Ordinal indistinguishability. Numerical reasoning often requires fine-grained distinction between close values. However, existing models typically learn coarse semantic representations of numerical attributes, making it difficult to capture such subtle ordinal relations. This

† Corresponding author.

leads to confusion in tasks that rely on comparative inference, and the ambiguity becomes more pronounced when handling hard negative samples with near-equal values.

To address these challenges, we propose **NumCoKE**, a novel framework for numerical reasoning on knowledge graphs via Mixture-of-Experts and Ordinal Contrastive Embedding. To tackle (**C1**), we design a *Mixture-of-Experts Knowledge-Aware* (MoEKA) encoder that dynamically captures the contextual importance of numeric attributes under different relational settings. Specifically, we construct relation-aware expert modules that encode entities and relations into a shared vector space. Through adaptive routing, numerical attributes are directed to the most relevant experts based on their semantic context. This mechanism enables NumCoKE to selectively amplify semantically important attributes for each relation-entity pair, thereby producing fine-grained and context-sensitive representations. To tackle (**C2**), we propose an *Ordinal Knowledge Contrastive Learning* (OKCL) strategy. Instead of relying solely on binary positive-negative sampling, OKCL synthesizes ordinal sample triplets by computing cosine-based similarity among value distributions. We then select top-k ordinal neighbors to serve as fine-grained supervision signals. This strategy preserves the continuity of numerical features and explicitly encourages the model to recognize nuanced ordinal relations between close values (e.g., distinguishing *183cm* from *184cm*). We further provide a theoretical analysis to justify the consistency and effectiveness of OKCL.

We validate the effectiveness of NumCoKE through extensive experiments on three real-world benchmark datasets covering diverse types of numerical attribute distributions (discrete, continuous, and skewed). The results show that our method consistently outperforms strong baselines across all settings. In summary, our contributions are three-folds:

- We propose **NumCoKE**, a novel and efficient model for numerical reasoning over KGs, which for the first time incorporates relation-aware mixture-of-experts encoding to dynamically integrate entity, relation, and attribute semantics.
- We introduce ordinal knowledge contrastive learning, a new contrastive paradigm that generates ordinally structured samples to better capture fine-grained numerical distinctions, improving the model's sensitivity to subtle ordinal relationships.
- We conduct comprehensive experiments on three public datasets. NumCoKE achieves state-of-the-art performance, demonstrating superior accuracy of our model.

Related Work

Knowledge Graph Embedding (KGE). KGE amis to capture the latent representations of entities and relations in KGs. TransE (Bordes et al. 2013) and its extended models (Zhang et al. 2021; Li et al. 2024), focus on treating relation as a *translation* from the head entity to the tail entity. DaBR (Wang et al. 2025) utilize distance-adaptive translations to learn geometric distance between entities, KGDM (Long et al. 2024) estimates the probabilistic distribution of target entities in prediction through diffusion models. RuIE (Liao et al. 2025) leverage logical rules to enhance reasoning. ConvE (Dettmers et al. 2018) extracts deep features of head entity and relation based on 2D convolution. ConE (Bai et al. 2021), MuRP (Balazevic, Allen, and Hospedales 2019a), GIE (Cao et al. 2022b), AttH (Chami et al. 2020) and LorentzKG (Fan et al. 2024) embed KGs into hyperbolic spaces to model hierarchical relations. Further, models (Schlichtkrull et al. 2018; Shang et al. 2019; Vashishth et al. 2020) of GNNs have been proposed to model higher-order connectivity in knowledge graphs. However, these models do not consider numerical values, making it difficult to accomplish numerical reasoning tasks.

KGE with Numerical Attributes. To improve the performance of KGE, several KGE models have been proposed to incorporate auxiliary information about the entity such as literals and numerical attributes. For example, KBLRN (García-Durán and Niepert 2018) accomplish the KGE task based on numerical values, considering numerical differences between different entities. MT-KGNN (Tay et al. 2017) is a multitask model to predict both numeric attribute values and entity/relation embeddings. LiteralE (Kristiadi et al. 2019) combines the entity embedding with numeric attributes based on a learnable gating function. Deterministic stand-alone value representation methods including NEKG (Duan, Yang, and Tam 2021) and NRN (Bai et al. 2023) are also used to predict numerical attributes in KGs. However, these models do not exploit joint interactions among entities, relations and numerical attributes to mine the useful semantic information.

Contrastive Learning (CL). CL is a self-supervised learning method by pulling semantically close neighbors together while pushing non-neighbors away, which can improving the representations of entities/graphs. SimGCL (Yu et al. 2022) adds uniform noise to the entity representation, which is an expansion-free CL method. HeCO (Wang et al. 2021b) employs contrast learning in network schema and meta-path, capturing both local and global features of entities. SLiCE (Wang et al. 2021a) makes use of mutual attraction between closed nodes to learn subgraph representations. RAKGE (Kim et al. 2023) interacts with training samples with head entities to generate samples. RRNE (Jeong et al. 2025) selects hard samples from subgraph. However, none of the above methods can guarantee effective samples for numerical reasoning tasks.

Methodology

In this section, we present a new numerical reasoning framework for KG termed NumCoKE. Specifically, there are two main components: Mixture-of-Experts-Knowledge-Aware Encoder (MoEKA-Encoder) and ordinal knowledge contrastive learning (OKCL) strategy. The overall framework of NumCoKE is illustrated in Figure 2.

Problem Definition. Given a knowledge graph as a collection of real triples $\mathcal{G} = \{(h, r, t)\}$, where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$ represent the set of entities and relations, respectively. Denote an entity-numeric value matrix for entities as $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{M}|}$ and the m-th numeric value belongs to the entity i as X_{im}, where $\mathcal{M}$ is the set of numeric attribute

fields (e.g., *height*, *weight*, and *born_year*). Due to the incompleteness of the KG, KGE emerge as an essential approach to conduct the knowledge graph completion, which is to map each triple (*h, r, t*) to a reasonableness score based on the KGE model, where a low score means that the triple is not reasonable, and vice versa. In this paper, our purpose is to construct a new KGE model to conduct numerical reasoning with the assistance of numerical attributions.

Knowledge-Aware Learning

Numeric Value Embedding Learning. To learn the numeric values, for observable scalars, we use a learnable embedding matrix to map them to a vector space. For missing scalars, we use learnable special missing value embeddings to preserve the magnitude information and prevent the task from being threatened, instead of using fixed values such as zero: $\boldsymbol{o}_i^m = (\boldsymbol{W}_x^m + \boldsymbol{w}_m X_{im}) \odot \boldsymbol{v}_m$, where X_{im} is the corresponding numeric value of the entity *i*. $\boldsymbol{v}_m \in \mathbb{R}^{d_{att}}$ represents the embedding vector of the numeric attribute field *m*, d_{att} stands for the embedding dimension of attribute. $\boldsymbol{w}_m \in \mathbb{R}^{d_{att}}$ and $\boldsymbol{W}_x^m \in \mathbb{R}^{d_{att} \times d_{att}}$ are linear transformations that learn the context information between each numeric value and attribute. The operation symbol $\odot$ represents point-wise multiplication. We denote $\boldsymbol{o}_i^m \in \mathbb{R}^{d_{att}}$ as the *m*-th field attribute embedding of the entity *i*. In this way, we can learn a more robust representation of both existing/missing numerical attributes.

Mixture of Knowledge Experts (MOE). To better learn entity embeddings from different perspectives with relation contexts and numerical attributes, we first employ a module called Mixture of Knowledge Experts to build expert networks, where each perspective corresponds to an expert network. First, the entity $i \in \mathcal{E}$ has a raw feature $\tilde{e}_i$. We then learn the multi-perspective embeddings $\mathcal{H}_1^{\tilde{e}_i}, \mathcal{H}_2^{\tilde{e}_i}, \ldots, \mathcal{H}_K^{\tilde{e}_i}$ for the entity i by establishing K knowledge experts denoted as $\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_K$. This process can be represented as $\mathcal{H}_k^{\tilde{e}_i} = \mathcal{W}_k(\tilde{e}_i)$. Next, we design a semantic-guided gated fusion network (SGN) to facilitate the fusion of inter-perspective entity embeddings with relation guidance.

$$\boldsymbol{e}_i = \sum\nolimits_{k=1}^{K} \mathcal{F}_k(\mathcal{H}_k^{\tilde{e}_i}, r)\mathcal{H}_k^{\tilde{e}_i} \tag{1}$$

where $\boldsymbol{e_i} \in \mathbb{R}^{d_{emb}}$ is the output entity embedding of entity *i* for relation r, $\mathcal{F}_k$ is the weight for each expert:

$$\mathcal{F}_k(\mathcal{H}_k^{\tilde{e}_i}, r) = \frac{exp((\mathcal{U}(\mathcal{H}_k^{\tilde{e}_i}) + \psi_k)/\rho(\epsilon_r))}{\sum_{j=1}^{K} exp((\mathcal{U}(\mathcal{H}_j^{\tilde{e}_i}) + \psi_j)/\rho(\epsilon_r))} \tag{2}$$

where $\psi_k \sim \mathcal{N}(0, \mathcal{U}'(\mathcal{H}_k^{\tilde{e}_i}))$, $\mathcal{U}$ and $\mathcal{U}'$ are two projection layers, and ψ_k is tunable Gaussian noise (Bian et al. 2023) used to balance the weights for each expert and enhance the model's robustness. Additionally, we add a relation-aware temperature ϵ_r with a sigmoid function ρ to limit the temperature within the range $(0, 1)$. Our aim is to obtain an entity embedding within the relational context of the current prediction before making the final decision.

Knowledge Perceptual Attention. Based on the representations above, we then learn the knowledge-aware representations via the joint semantic interactions among entities, relations, and attributes. In this way, we propose a *knowledge perceptual attention mechanism* for entity-relation attributes. Specifically, we leverage a single-layer perception f_{e-att} and f_{r-att} to project the entity and relation embeddings onto the attribute embedding subspace, respectively. Thereafter, we mixing the information of entity *e* and relation *r*, so as to get the joint projected embedding $\boldsymbol{p}_{joint}^{att} \in \mathbb{R}^{d_{att}}$:

$$\boldsymbol{p}_{joint}^{att} = \sum_{*\in\{e,r\}} \delta_* f_{*-att}(\boldsymbol{p}_*) \tag{3}$$

where $\boldsymbol{p}_e$ corresponds to $\boldsymbol{e}_i$, $\boldsymbol{p}_r$ corresponds to the relation embedding, $\delta_* \in [0, 1]$ are hyperparameters and $\delta_e + \delta_r = 1$.

Then, we apply the *knowledge perceptual attention mechanism* to establish the joint interactions between entities, relations, and attributes. Specifically, We apply this mechanism for all attribute fields m = 1,2,$\cdots$, $|\mathcal{M}|$ and we repeat this step for all attention heads l = 1,2,$\cdots$,L, where L is the number of multi-head attentions. The result of the joint interaction of head *l* is shown as follows:

$$\boldsymbol{o}_{joint,i}^{(l)} = \sum\nolimits_{m=1}^{|\mathcal{M}|} a_{joint,i,m}^{(l)} (\boldsymbol{W}_{agg}^{(l)} \boldsymbol{o}_i^m) \tag{4}$$

where the linear transformation matrix $\boldsymbol{W}_{agg}^{(l)} \in \mathbb{R}^{d_{sub} \times d_{att}}$ aims to project attribute embedding $\boldsymbol{o}_i^m$ into low-dimensional subspaces to capture the importance of each numeric attribute from the given entity with relation accurately. Under a specific attention head *l*, we map $\boldsymbol{e}_{joint}^{att}$ and $\boldsymbol{o}_i^1, \cdots \boldsymbol{o}_i^{|\mathcal{M}|} \in \mathbb{R}^{d_{att}}$ onto smaller spaces and capture the attention score. The normalized attention weight $a_{joint,i,m}^{(l)}$ of the attribute embedding $\boldsymbol{o}_i^m$ is formulated as follows:

$$a_{joint,i,m}^{(l)} = \frac{exp(s^{(l)}(\boldsymbol{p}_{joint}^{att}, \boldsymbol{o}_i^m))}{\sum_{n=1}^{|\mathcal{M}|} exp(s^{(l)}(\boldsymbol{p}_{joint}^{att}, \boldsymbol{o}_i^n))} \tag{5}$$

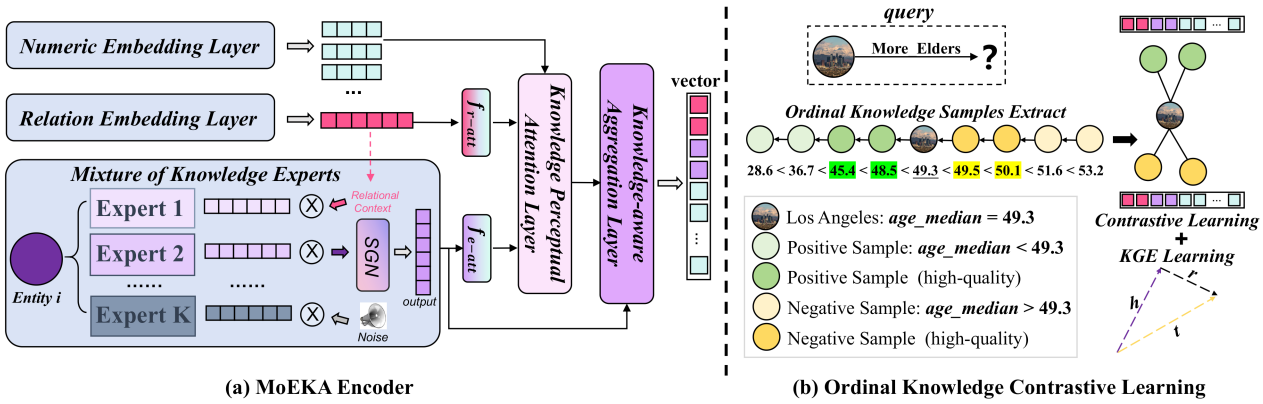
$$s^{(l)}(\boldsymbol{p}_{joint}^{att}, \boldsymbol{o}_i^m) = \sigma\left(\|_{j=1}^{H} \boldsymbol{W}_j^{(l)} \phi_j(\boldsymbol{p}_{joint}^{att}, \boldsymbol{o}_i^m)\right) \tag{6}$$

where σ denotes the LeakyReLU activation function, $\boldsymbol{W}_j^{(l)}$ denotes the weighted matrix corresponding to the i-th operator in the l-th layer, $\phi_j((\boldsymbol{p}_{joint}^{att}, \boldsymbol{o}_i^m))$ denotes the fusion operation, which can be model by MLP. $\|$ here denotes the multi-head mechanism.

Knowledge-aware Aggregation. The final attribute embedding for entity *i* related to the joint perceptual vector is as follows: $\boldsymbol{o}_{joint,i} = \boldsymbol{o}_{joint,i}^{(1)} \| \boldsymbol{o}_{joint,i}^{(2)} \| \cdots \| \boldsymbol{o}_{joint,i}^{(L)}$, where $\boldsymbol{o}_{joint,i} \in \mathbb{R}^{d_{att}}$ as the attribute vector of entity i, which is aware of the entity and the relation. Symbol $\|$ represents a concatenation operator. To balance the joint knowledge-aware attribute vectors and the embedded entities, we have:

$$\boldsymbol{e}_{joint,i}^{att} = \sigma(\boldsymbol{W}_{ge} \boldsymbol{e}_i + \boldsymbol{W}_{ga} \boldsymbol{o}_{joint,i}) + \boldsymbol{b} \tag{7}$$

where $\boldsymbol{W}_{ge} \in \mathbb{R}^{d_{emb} \times d_{emb}}$ and $\boldsymbol{W}_{ga} \in \mathbb{R}^{d_{emb} \times d_{att}}$ are different linear transformations to assign different weights to each entity representation and fused numerical representation by adaptively learning to obtain high quality final embeddings and to focus on the more important information. $\boldsymbol{b}$ is a bias. σ is a sigmoid function. We name $\boldsymbol{e}_{joint,i}^{att} \in \mathbb{R}^{d_{emb}}$ as the attribute-enriched vector of entity *i*.



(a) MoEKA Encoder (b) Ordinal Knowledge Contrastive Learning

Figure 2: The Overview of our model. (a) The MoEKA Encoder encodes each entity with the relation and attributes to a unified, elaborate semantic representation. (b) To capture the fine-grained semantic information, we utilize a new knowledge contrastive learning method to generate high-quality ordinal samples to learn the nuances in attributes and distinguish similar semantics.

Ordinal Knowledge Contrastive Learning

Learning ordinal relations is essential for numerical reasoning tasks. Unlike the previous work (Kim et al. 2023) mainly focusing on generating valid, positive, and negative samples, we turn to a more difficult and significant task that generates high-quality ordinal samples.

Ordinal Relation Learning. Given N objects and the ordinal relationship triples $\mathcal{S} = \{(a, b, c) \mid a, b, c \in \mathcal{E}, a \neq b \neq c\}$, where $[N] = \{1, \ldots, N\}$ and object a is more similar to object b than it is to object c. Given some distance function $d(\cdot, \cdot)$, we aim to learn the representations of objects, denoted as $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, such that the following objectives hold as much as possible:

$$d(\mathbf{x}_a, \mathbf{x}_b) < d(\mathbf{x}_a, \mathbf{x}_c), \quad \forall (a, b, c) \in \mathcal{S} \tag{8}$$

Knowledge Samples Generator. On the analysis above, to distinguish precise semantics in numerical reasoning, we generate the preliminary positive/negative ($\mathcal{E}^+$ and $\mathcal{E}^-$) samples based on the available head entities and relations, the set of positive/negative samples is as follows:

$$\mathcal{E}^+ = \{\boldsymbol{e}^{att}_{joint,i} | i \in \mathcal{P}[h, r]\},\ \mathcal{E}^- = \{\boldsymbol{e}^{att}_{joint,j} | j \in \mathcal{N}[h, r]\} \tag{9}$$

where $\mathcal{P}[h, r]$ represents the set of positive tail entities, $\mathcal{N}[h, r]$ represents the set of positive tail entities.

Ordinal Samples Extractor. Since each attribute field (e.g., *year*, *age*, *weight*) has its unique distribution, we select k number of samples with the highest cosine-similarity to $\boldsymbol{e}^{att}_{joint,i}$ from all the generated positive/negative samples with the assistance of Eq. (8). Note that this step is essential and it can generate the samples obeying ordinal embedding. Then we generate high-quality ordinal samples as follows:

$$\boldsymbol{e}^{t+}_{mix} = \alpha \cdot \sum_{p \in Topk(\mathcal{E}^+)} \boldsymbol{e}^{att}_{joint,p} + (1 - \alpha) \cdot \boldsymbol{e}^{att}_{joint,h} \tag{10}$$

$$\boldsymbol{e}^{t-}_{mix} = \beta \cdot \sum_{n \in Topk(\mathcal{E}^-)} \boldsymbol{e}^{att}_{joint,n} + (1 - \beta) \cdot \boldsymbol{e}^{att}_{joint,h} \tag{11}$$

where α, β are blending coefficients that are sampled from the uniform distribution $[0, 1]$. $Topk(\mathcal{E}^+)$ and $Topk(\mathcal{E}^-)$ are sampling sets with the highest cosine similarity to $\boldsymbol{e}^{att}_{joint,i}$ in $\mathcal{E}^+$ and $\mathcal{E}^-$, respectively. In this sense, to satisfy ordinal relation, we have:

$$d(\boldsymbol{e}^{att}_{joint,h}, p) < d(\boldsymbol{e}^{att}_{joint,h}, n) \tag{12}$$

where $p \in Topk(\mathcal{E}^+)$ and $n \in Topk(\mathcal{E}^-)$. In this way, our contrastive learning loss is defined as follows:

$$\mathcal{L}_{CL} = -\frac{1}{|\mathcal{G}|} \sum\nolimits_{l \in \mathcal{G}} \log \frac{\varpi(\mathbf{p}^\top \mathbf{u}/\tau)}{\varpi(\mathbf{p}^\top \mathbf{u}/\tau) + \sum_{v_- \in \boldsymbol{e}^{t-}_{mix}} \varpi(\mathbf{v}^\top \mathbf{v}_-/\tau)} \tag{13}$$

where $\boldsymbol{p} = \boldsymbol{e}^{att}_{joint,h} + \boldsymbol{e}_r$, $\boldsymbol{u} = \boldsymbol{e}^{t+}_{mix}$, τ is the temperature hyperparameter. ϖ represents exponential function $exp(\cdot)$. By implementing the strategy of randomly mixing the positive tails and employing head blending techniques (as described in Eqs. (10)-(11)), we notably increase the diversity of our samples and interpret the nuances of semantic information.

Training

Score Function. A classical assumption of typical KGE methods is to compute the distances between head entities, relations, and tail entities. However, in numerical reasoning tasks, these methods suffer from serious drawbacks because they do not effectively address the order embedding problem posed by relational patterns in complex tasks. For example, given the head entity (*weight=80kg*) and its relation (*is_heavier_than*), and using the TransE score function, the two tail entities e_1 (*weight=70kg*) and e_2 (*weight=60kg*) are mapped to the same location, failing to express a numerical ordering relation such as 80>70>60, which impedes numerical reasoning. To address this issue, we design a rationality score function specifically for the numerical reasoning task as follows:

$$\begin{aligned} &score(\boldsymbol{e}^{att}_{joint,h}, \boldsymbol{e}_r, \boldsymbol{e}^{att}_{joint,i}) \\ =&\varepsilon - \|\boldsymbol{e}^{att}_{joint,h} + \boldsymbol{e}_r - \boldsymbol{e}^{att}_{joint,i}\|_{1/2} + \\ &\Delta \|max(0, \boldsymbol{W}_r \boldsymbol{e}_{joint,h} - \boldsymbol{W}_r \boldsymbol{e}^{att}_{joint,i})\|^2 \end{aligned} \tag{14}$$

where Δ represents the weight in the numerical reasoning score function. The first term is derived from TransE (Bor-

des et al. 2013), while the second term modifies the Order-Embedding (Vendrov et al. 2016). $\boldsymbol{W}_r \in \mathbb{R}^{d_{emb} \times d_{emb}}$ is a projection matrix critical for mapping entities to a specific relational space, indicating that depending on the type of relation, each entity may assume a different order.

Loss Function. In this paper, we use the binary cross-entropy loss (Dettmers et al. 2018; Kristiadi et al. 2019) to optimize the training process. Let $\mathcal{T} = \mathcal{G} \cup \mathcal{G}^-$ denote the training dataset,where $\mathcal{G}$ denotes the set of positive knowledge triples, $\mathcal{G}^-$ denotes the set of negative knowledge triples $\{(h, r, t^{'})|h, t^{'} \in \mathcal{E}, r \in \mathcal{R}, (h, r, t^{'}) \notin \mathcal{G}\}$. The binary-cross entropy loss is defined as follows:

$$\mathcal{L}_{BCE} = -\frac{1}{|\mathcal{T}|} \sum_{l \in \mathcal{T}} (y_l \log(p_l) + (1 - y_l) \log(1 - p_l)) \quad (15)$$

where $y_l \in \{0, 1\}$ is the truth label, and $p_l \in$ [0,1] is the probability of each triple $(h, r, t) = l \in \mathcal{T}$, which is formulated as Eq. (14). Combining Eq. (13) and Eq. (15), the final loss can be summarized as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{BCE} + \lambda \mathcal{L}_{CL} \quad (16)$$

where λ is a hyperparameter, stands for the coefficient of contrastive learning loss.

Theoretical Analysis

To clarify how NumCoKE differs from other models, we compare it with popular models in the Appendix.

Time Complexity Analysis. NumCoKE consists of three components, the MoEKA encoder, a process of contrastive learning, and the score function. In terms of MoEKA Encoder, the time complexity of processing every triple is $\mathcal{O}(|\mathcal{M}|)$. The time complexity of the numerical value embedding layer is $\mathcal{O}(|\mathcal{M}|)$ because we transform each attribute field into an embedding vector. In the joint perceptual attention layer, the query is the target relation and the number of keys and values corresponds to the number of attribute fields, so the time complexity remains $\mathcal{O}(|\mathcal{M}|)$. The gating layer of the MoEKA Encoder is not involved in the calculation of time complexity. During the process of contrastive learning, for every given triple*(h,r,t)*, the time complexity is proportional to the number of positive samples $|\mathcal{P}[h, r]|$ and the number of negative samples $|\mathcal{N}[h, r]|$. The time complexity is $\mathcal{O}(|\mathcal{M}| \cdot rel\_num)$, where *rel_num* denotes the number of relations per entity. The TransE score function involves only addition and subtraction operations, so it does not change the time complexity. Therefore, the overall time complexity of NumCoKE is $\mathcal{O}(\mathcal{T} \cdot (|\mathcal{M}| + |\mathcal{M}| \cdot rel\_num)) \approx \mathcal{O}(|\mathcal{T}| \cdot rel\_num)$, where $\mathcal{T}$ denotes the number of positive and negative triples used for training.

The Expressiveness of Our Model. We conduct the expressiveness of numerical reasoning from the perspective of a one-hop reasoning task. Specifically, we aim to theoretically analyze the process of contrastive learning in NumCoKE and establish theoretical guarantees for the downstream performance of the learned representations. In the numerical reasoning, we denote the set of node representations as $\mathcal{V}$, and define the mean representations from one-hop neighborhoods of node v as $\boldsymbol{z} = \frac{1}{\mathcal{N}(v)} \sum_{\boldsymbol{u} \in \mathcal{N}(v)} \boldsymbol{u}$, where $\boldsymbol{u}$ represents the projected node representations of positive samples of node v, $\boldsymbol{z}$ describes the one-hop neighborhood pattern of node v. In the world, nodes belonging to the same semantic class tend to have similar neighborhood patterns, so $\boldsymbol{z}$ can be viewed sampled from $Z|Y \sim \mathcal{N}(\mathbf{z}_Y, I)$ where Y is the latent semantic class indicating the one-hop pattern of node v. We demonstrate that minimizing NumCoKE's objective in Equation (13) with an exponential moving average is equivalent to maximizing mutual information between representation V and the one-hop pattern Y, which explains the rationality of NumCoKE in capturing one-hop patterns as follows:

$$\mathcal{L}_{CL} \geq H(V|Y) - H(V) = -I(V; Y) \quad (17)$$

Eq. (17) indicates minimizing NumCoKE loss in Eq. (13) promotes maximizing the mutual information $I(V; Y)$ between representations and one-hop neighborhood context. In this way, our NumCoKE can exploit more latent semantics among KGs, hence facilitating numerical reasoning. All the detailed proofs are in Appendix B.

Experiment

Experimental Settings

Datasets. We select three real-world KG datasets. Specifically, US-Cities[1] contains basic information about cities in the United States. Spotify is a dataset of songs for developers[2]. Credit (Yeh and Lien 2009) is a knowledge graph constructed from credit events in Taiwan. Additionally, all the three datasets are modified to simulate numerical reasoning tasks under real conditions. Close to 20% of the numerical values are masked to zero (missing values). More details and the statistics of datasets can be found in Appendix C.

Evaluation Metrics. We used the improved learning methods based on dropout, batch normalization, and linear transformation proposed alongside LTE (Zhang et al. 2022) for evaluation. Mean Reciprocal Rank (MRR), Mean Rank (MR) and Hits@{1, 3, 10} are reported as metrics. Higher Hit@n and MRR values indicate a better performance, whereas lower MR values imply a better performance.

Baseline. We consider six groups of 23 baseline methods: Euclidean KGE including TransE (Bordes et al. 2013), ConvE (Dettmers et al. 2018), TuckER (Balazevic, Allen, and Hospedales 2019b), HAKE (Zhang et al. 2020), DaBR (Wang et al. 2025), KGDM (Long et al. 2024), and RuIE (Liao et al. 2025). Hyperbolic KGE including MuRP (Balazevic, Allen, and Hospedales 2019a), ConE (Bai et al. 2021), AttH (Chami et al. 2020), GIE (Cao et al. 2022b) and LorentzKG (Fan et al. 2024); GNNs-KGE including R-GCN (Schlichtkrull et al. 2018) and WGCN (Shang et al. 2019); Attributed KGE including KBLRN (García-Durán and Niepert 2018), MT-KGNN (Tay et al. 2017) and LiteralE (Kristiadi et al. 2019); Self-supervised model for graph including BiGI (Cao et al. 2021), SLiCE (Wang et al. 2021a), SimGCL (Yu et al. 2022), and RAKGE (Kim et al. 2023); Deterministic value representation methods including NEKG (Duan, Yang, and Tam 2021) and NRN (Bai et al.

[1]https://simplemaps.com/data/us-cities

[2]https://www.kaggle.com/datasets/geomack

Model		US-Cities					Spotify					Credit				
		H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10	MR	MRR
Euclidean	**TransE**	0.189	0.248	0.324	367	0.239	0.259	0.355	0.462	115	0.332	0.421	0.520	0.630	39	0.493
	ConvE	0.158	0.202	0.274	385	0.198	0.231	0.307	0.414	110	0.295	0.171	0.281	0.430	58	0.261
	TuckER	0.156	0.212	0.308	321	0.207	0.211	0.296	0.411	98	0.278	0.405	0.512	0.638	36	0.485
	HAKE	0.003	0.025	0.064	981	0.024	0.008	0.085	0.120	196	0.075	0.051	0.151	0.273	128	0.134
	DaBR	0.202	0.256	0.337	352	0.241	0.266	0.367	0.477	104	0.334	0.425	0.526	0.621	36	0.498
	KGDM	0.232	0.271	0.353	329	0.246	0.274	0.374	0.489	99	0.345	0.419	0.532	0.626	39	0.482
	RuIE	0.084	0.126	0.185	587	0.120	0.124	0.188	0.284	173	0.182	0.291	0.388	0.513	65	0.366
Hyperbolic	**AttH**	0.051	0.069	0.108	1255	0.076	0.052	0.082	0.146	348	0.087	0.176	0.261	0.395	101	0.251
	ConE	0.009	0.042	0.114	239	0.048	0.001	0.006	0.064	227	0.028	0.006	0.241	0.395	82	0.156
	GIE	0.095	0.134	0.206	570	0.132	0.121	0.183	0.281	185	0.174	0.285	0.388	0.512	76	0.359
	MuRP	0.082	0.113	0.175	457	0.115	0.024	0.179	0.324	135	0.119	0.151	0.246	0.432	80	0.228
	LorentzKG	0.179	0.221	0.286	368	0.209	0.239	0.312	0.412	126	0.320	0.402	0.509	0.597	38	0.482
GNNs	**R-GCN**	0.212	0.271	0.355	314	0.263	0.288	0.382	0.504	89	0.364	0.480	0.570	0.675	34	0.546
	WGCN	0.029	0.056	0.104	968	0.057	0.095	0.162	0.261	331	0.153	0.170	0.256	0.369	100	0.241
Attributed	**KBLRN**	0.006	0.018	0.046	2164	0.021	0.017	0.039	0.086	347	0.044	0.007	0.020	0.079	271	0.061
	MT-KGNN	0.071	0.109	0.156	653	0.102	0.108	0.182	0.302	142	0.176	0.211	0.304	0.436	74	0.285
	LiteralE	0.246	0.308	0.402	228	0.299	0.264	0.371	0.498	76	0.345	0.475	0.562	0.676	36	0.549
Deterministic	**NEKG**	0.196	0.230	0.304	376	0.217	0.246	0.323	0.423	121	0.336	0.398	0.512	0.589	41	0.477
	NRN	0.298	0.326	0.432	155	0.323	0.342	0.442	0.546	102	0.414	0.560	0.639	0.740	21	0.620
Self-supervised	**BiGI**	0.185	0.249	0.331	359	0.236	0.260	0.354	0.468	118	0.331	0.418	0.507	0.622	39	0.487
	SliCE	0.185	0.250	0.331	359	0.237	0.261	0.354	0.469	117	0.332	0.420	0.510	0.622	38	0.490
	SimGCL	0.344	0.415	0.502	162	0.399	0.000	0.255	0.467	59	0.167	0.000	0.399	0.645	22	0.239
	RAKGE	0.395	0.455	0.529	199	0.442	0.502	0.608	0.674	37	0.573	0.647	0.733	0.823	12	0.708
NumCoKE (ours)		**0.439**	**0.536**	**0.640**	**119**	**0.508**	**0.636**	**0.694**	**0.758**	**30**	**0.680**	**0.745**	**0.818**	**0.888**	**7**	**0.794**
lmprovement		11.1%	17.8%	21.0%	26.5%	14.9%	26.7%	14.1%	12.5%	18.9%	18.7%	15.1%	11.6%	7.9%	41.7%	12.1%

Table 1: Results for numerical reasoning. Bold scores indicate the best results, while underlined scores represent the second-best results. The % of Improvement column shows the relative improvements of NumCoKE compared to the second-best scores.

2023). Among all the baselines, for those that do not take care of external numerical attributes, such as TransE, we follow the original approach and concat the embeddings of entity and attributes. The results of self-supervised KGE are from RAKGE. More details can refer to the Appendix C.

Experimental Results

The main results are detailed in Table1. We conduct 5 rounds of experiments and take the average as the final result, which reveals that NumCoKE makes significant progress in all the metrics and achieves new SOTA results. All the evaluated models are implemented on a server with one GPU (*NVIDIA A800, 80GB*). Notably, MRR on Credit increased from 0.493 for TransE and 0.498 for DaBR to 0.794, which highlights NumCoKE's robust capability in capturing the asymmetric nature of relations and the ordinal information. At the same time, NumCoKE's MRR significantly outperforms to hierarchy-aware models such as TuckER and hyperbolic KGE methods, affirming its superior utilization of the hierarchical structures inherent in numerical reasoning relations. RuIE performs weakly, which we infer is due to excessive focus on logical rules rather than comparative relationships between numerical semantics. Although utilizing the extra attributes of entities, attribute-based models like KBLRN and MT-KGNN experience performance declines due to the relevance of numerical attributes, whereas NumCoKE excels under these conditions. Deterministic approaches such as NEKG and NRN are unable to recognize semantic differences between numerical fields and only consider the distribution of attribute values in each field (e.g., *51kg* and *51 years old*), whereas NumCoKE sets up different representation for each field and utilizes the *knowledge perceptual attention mechanism* that allows it to focus on attributes that are more relevant to the relation. In addition, compared to LiteralE and RAKGE, NumCoKE excels by fully exploiting the potential of the relational contexts and the interactions among entities, relations and numerical attributes. More details can refer to the study for ordinal knowledge contrastive learning in Appendix D.

Experimental Analysis

Ablation Study. To assess the contribution of each component in **NumCoKE**, we conduct ablation studies by removing key modules individually. (1) Knowledge-aware attention (KA): We disable the KA module by setting $\delta_e = \delta_r = 0$ in Eq. (3), effectively removing the influence of relation- and entity-specific signals on attribute modeling. As shown in Table 2, this leads to a significant performance drop (↓8.1% on Credit), confirming the importance of incorporating both entity and relation semantics for numerical reasoning. (2) Mixture-of-experts (MoE): We remove the MoE mechanism and instead use static entity representations across all relation contexts. This results in a 5.0% accuracy drop, indicating that dynamically adapting entity embeddings based on relational context is crucial for capturing semantic nuances. (3) Ordinal sampling (OS): We further ablate the top-k ordinal sampling strategy used in Eq. (10)–(11). Without OS, the model relies solely on ran-

domly sampled contrastive pairs, resulting in degraded performance (↓6.4% on Credit). This highlights the effectiveness of OS in improving the model's ability to distinguish subtle ordinal differences among close numerical values. We also conduct a single-module-addition study, the results can be seen in the Appendix.

MoE	KA	OS	H@1	H@3	H@10	MR	MRR
×	√	√	0.712	0.764	0.851	9	0.765
√	×	√	0.688	0.763	0.846	10	0.740
√	√	×	0.698	0.776	0.843	10	0.750
√	√	√	**0.745**	**0.818**	**0.888**	**7**	**0.794**

Table 2: Ablation study on Credit dataset.

Analysis of MoEKA Encoder. We investigate the effect of the number of experts K in the MoE module, as depicted in Figure 3. It can be observed that the impact of the number of experts K on the final results generally follows a pattern of initial increase followed by a decrease, mainly affecting fine-grained metrics such as Hit@1 and MRR. Having either too many or too few experts is detrimental to the model's learning performance, this is because the same relational category corresponds to similar contexts, for example, *ranking_comp* and *disabled_comp* are both comparative relations, and too many experts can cause semantic confusion.

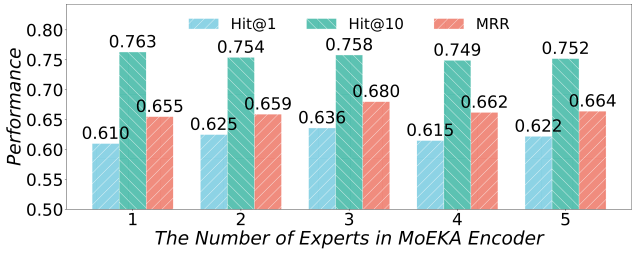


Figure 3: Research of Hyperparameter K on Spotify dataset.

Effect of Relation-Entity Perception Weights. To assess the relative contributions of entities and relations in numerical attribute modeling, we vary their attention weights in the joint perception mechanism (Eq. 3). Specifically, we adjust the proportion assigned to relations from 0% to 100% and evaluate the performance of the model on the CREDIT dataset. As shown in Figure 4, **NumCoKE achieves optimal performance when relations contribute 40%–90% of the total attention weight**, with peak results at 90%. Compared to the entity-only variant (0% relation weight), this configuration improves MRR by **46.1%** (from 0.186 to 0.272), demonstrating the significant value of relation semantics in attribute perception. However, performance deteriorates when relations are used exclusively (i.e., 100%), mirroring the trend observed in RAKGE (Kim et al. 2023), which neglects entity semantics. These results **highlight the necessity of jointly modeling entity and relation semantics** to effectively capture numerical attributes.

Case Study. Furthermore, for easier visualization, we select part of the samples from Credit to show the attention scores between entities and attributes using heatmaps. As shown

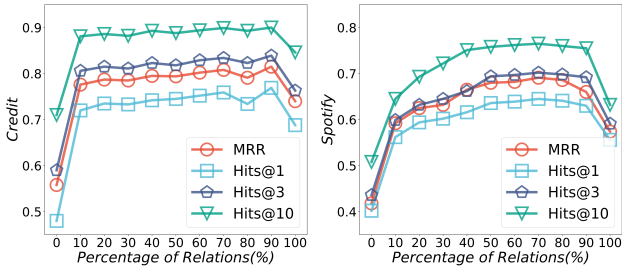


Figure 4: Proportional test on Credit and Spotify. A relation percentage of 100% means that we only consider relations, while 0% means that we only consider entities.

in Figure 5, the color blocks in each row of the first heat map are similar, indicating that RAKGE pays almost equal attention to each attribute in most samples, while the color blocks from NumCoKE are clearly differentiated, demonstrating that NumCoKE can better distinguish the importance of attributes in different relation contexts. More case studies can refer to Appendix E for more details.

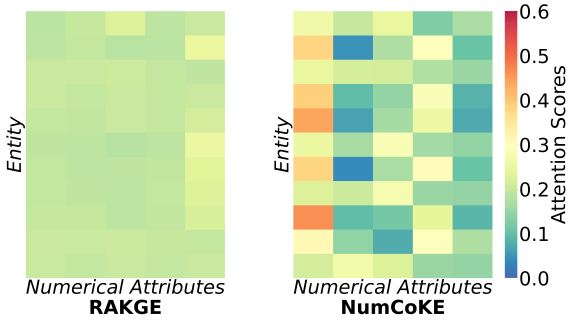


Figure 5: Visualization of relevance scores of each numeric attribute of RAKGE and NumCoKE on Credit dataset.

Supplementary Experiments

We performed other experiments, including hyperparameter analysis, evaluation on FB15k-237, etc. (see Appendix E).

Conclusion

In this paper, we introduce a novel KGE model called NumCoKE. Specifically, we incorporates a Mixture-of-Experts-Knowledge-Aware Encoder for elaborate semantic modeling, capturing the connections among entities, relations, and attributes within a unified framework. To distinguish similar semantics, we introduce a novel ordinal knowledge contrastive learning strategy that that generates high-quality ordinal samples from original data, capturing fine-grained semantic nuances over close numerical values. Extensive experiments on three standard benchmarks demonstrate NumCoKE's effectiveness for numerical reasoning on KGs.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2022YFB3903904, the Key R&D Pro-

gram of Jiangxi Province under Grant 20232BBGW001.

References

Bai, J.; Luo, C.; Li, Z.; Yin, Q.; Yin, B.; and Song, Y. 2023. Knowledge Graph Reasoning over Entities and Numerical Values. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023*, 57–68. ACM.

Bai, Y.; Ying, Z.; Ren, H.; and Leskovec, J. 2021. Modeling Heterogeneous Hierarchies with Relation-specific Hyperbolic Cones. In *Advances in Neural Information Processing Systems.*, 12316–12327.

Balazevic, I.; Allen, C.; and Hospedales, T. M. 2019a. Multi-relational Poincaré Graph Embeddings. In *Advances in Neural Information Processing Systems.*, 4465–4475.

Balazevic, I.; Allen, C.; and Hospedales, T. M. 2019b. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing.*, 5184–5193.

Bian, S.; Pan, X.; Zhao, W. X.; Wang, J.; Wang, C.; and Wen, J. 2023. Multi-modal Mixture of Experts Represetation Learning for Sequential Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM*, 110–119.

Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems.*, 2787–2795.

Cao, J.; Lin, X.; Guo, S.; Liu, L.; Liu, T.; and Wang, B. 2021. Bipartite Graph Embedding via Mutual Information Maximization. In *Proceedings of the International Conference on Web Search and Data Mining.*, 635–643.

Cao, X.; Shi, Y.; Wang, J.; Yu, H.; Wang, X.; and Yan, Z. 2022a. Cross-modal Knowledge Graph Contrastive Learning for Machine Learning Method Recommendation. In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, 3694–3702.

Cao, Z.; Xu, Q.; Yang, Z.; Cao, X.; and Huang, Q. 2022b. Geometry Interaction Knowledge Graph Embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence.*, 5521–5529.

Chami, I.; Wolf, A.; Juan, D.; Sala, F.; Ravi, S.; and Ré, C. 2020. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.*, 6901–6914.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence.*, 1811–1818.

Duan, H.; Yang, Y.; and Tam, K. Y. 2021. Learning Numeracy: A Simple Yet Effective Number Embedding Approach Using Knowledge Graph. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, 2597–2602.

Fan, X.; Xu, M.; Chen, H.; Chen, Y.; Das, M.; and Yang, H. 2024. Enhancing Hyperbolic Knowledge Graph Embeddings via Lorentz Transformations. In *Findings of the Association for Computational Linguistics, ACL 2024*, 4575–4589.

García-Durán, A.; and Niepert, M. 2018. KBlrn: End-to-End Learning of Knowledge Base Representations with Latent, Relational, and Numerical Features. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence.*, 372–381.

Guo, H.; Dai, T.; Zhu, M.; Meng, G.; Chen, B.; Wang, Z.; and Xia, S. 2023. One-stage Low-resolution Text Recognition with High-resolution Knowledge Transfer. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, 2189–2198.

Jeong, H.; Jung, H.; Kim, G.; Kim, J.; Kim, K. K.; and Park, H. 2025. Enhancing Inductive Numerical Reasoning in Knowledge Graphs with Relation-Aware Relative Numeric Encoding. In *Advances in Knowledge Discovery and Data Mining - 29th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2025*, 173–186.

Kim, G.; Kim, S.; Kim, K. K.; Park, S.; Jung, H.; and Park, H. 2023. Exploiting Relation-aware Attribute Representation Learning in Knowledge Graph Embedding for Numerical Reasoning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.*, 1086–1096.

Kristiadi, A.; Khan, M. A.; Lukovnikov, D.; Lehmann, J.; and Fischer, A. 2019. Incorporating Literals into Knowledge Graph Embeddings. In *Proceedings of the International Semantic Web Conference.*, 347–363.

Li, J.; Qi, G.; Zhang, C.; Chen, Y.; Tan, Y.; Xia, C.; and Tian, Y. 2023. Incorporating Domain Knowledge Graph into Multimodal Movie Genre Classification with Self-Supervised Attention and Contrastive Learning. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023*, 3337–3345.

Li, J.; Su, X.; Zhang, F.; and Gao, G. 2024. TransERR: Translation-based Knowledge Graph Embedding via Efficient Relation Rotation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, 16727–16737.

Li, Z.; Xu, Q.; Jiang, Y.; Cao, X.; and Huang, Q. 2020. Quaternion-Based Knowledge Graph Network for Recommendation. In *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, 880–888.

Liao, X.; Duan, J.; Huang, Y.; and Wang, J. 2025. RUIE: Retrieval-based Unified Information Extraction using Large Language Model. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025*, 9640–9655.

Long, X.; Zhuang, L.; Li, A.; Wei, J.; Li, H.; and Wang, S. 2024. KGDM: A Diffusion Model to Capture Multiple Relation Semantics for Knowledge Graph Embedding. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024*, 8850–8858.

Lu, Y.; Deng, B.; Yu, W.; and Yang, D. 2023. HELIOS: Hyper-Relational Schema Modeling from Knowledge Graphs. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, 4053–4064.

Schlichtkrull, M. S.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling Relational Data with Graph Convolutional Networks. In *Proceedings of the European Semantic WebConference.*, 593–607.

Shang, C.; Tang, Y.; Huang, J.; Bi, J.; He, X.; and Zhou, B. 2019. End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion. In *Proceedings of the AAAI Conference on Artificial Intelligence.*, 3060–3067.

Sun, J.; Yu, F.; Liu, S.; Luo, Y.; Liang, R.; and Shen, X. 2023. Adversarial Bootstrapped Question Representation Learning for Knowledge Tracing. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, 8016–8025.

Tay, Y.; Tuan, L. A.; Phan, M. C.; and Hui, S. C. 2017. Multi-Task Neural Network for Non-discrete Attribute Prediction in Knowledge Graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.*, 1029–1038.

Vashishth, S.; Sanyal, S.; Nitin, V.; and Talukdar, P. P. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations.*

Vendrov, I.; Kiros, R.; Fidler, S.; and Urtasun, R. 2016. Order-Embeddings of Images and Language. In *Proceedings of the International Conference on Learning Representations.*

Wang, P.; Agarwal, K.; Ham, C.; Choudhury, S.; and Reddy, C. K. 2021a. Self-Supervised Learning of Contextual Embeddings for Link Prediction in Heterogeneous Networks. In *Proceedings of the ACM Web Conference.*, 2946–2957.

Wang, W.; Liang, Q.; Bao, F.; and Gao, G. 2025. Distance-Adaptive Quaternion Knowledge Graph Embedding with Bidirectional Rotation. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, 4219–4231.

Wang, X.; Liu, N.; Han, H.; and Shi, C. 2021b. Self-supervised Heterogeneous Graph Neural Network with Co-contrastive Learning. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining.*, 1726–1736.

Wang, X.; Meng, B.; Chen, H.; Meng, Y.; Lv, K.; and Zhu, W. 2023. TIVA-KG: A Multimodal Knowledge Graph with Text, Image, Video and Audio. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, 2391–2399. ACM.

Yeh, I.; and Lien, C. 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Syst. Appl.*, 36(2): 2473–2480.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; and Nguyen, Q. V. H. 2022. Are Graph Augmentations Necessary?: Simple Graph Contrastive Learning for Recommendation. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval.*, 1294–1303.

Zhang, Z.; Cai, J.; Zhang, Y.; and Wang, J. 2020. Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence.*, 3065–3072.

Zhang, Z.; Jia, J.; Wan, Y.; Zhou, Y.; Kong, Y.; Qian, Y.; and Long, J. 2021. TransR*: Representation learning model by flexible translation and relation matrix projection. *J. Intell. Fuzzy Syst.*, 40(5): 10251–10259.

Zhang, Z.; Wang, J.; Ye, J.; and Wu, F. 2022. Rethinking Graph Convolutional Networks in Knowledge Graph Completion. In *Proceedings of the ACM Web Conference.*, 798–807.