APKGC: Noise-enhanced Multi-Modal Knowledge Graph Completion with Attention Penalty

Yue Jian[1,2,3], **Xiangyu Luo**[4], **Zhifei Li**[1,2,3,*], **Miao Zhang**[1,2,3]
Yan Zhang[1,2,3], **Kui Xiao**[1,2,3], **Xiaoju Hou**[5]
[1]School of Computer Science, Hubei University, Wuhan 430062, China
[2]Hubei Key Laboratory of Big Data Intelligent Analysis and Application (Hubei University), Wuhan 430062, China
[3]Key Laboratory of Intelligent Sensing System and Security (Hubei University), Ministry of Education, Wuhan 430062, China
[4]School of Cyber Science and Technology, Hubei University, Wuhan 430062, China
[5]Institute of Vocational Education, Guangdong Industry Polytechnic University, Guangzhou 510300, China
{jianyue, xyluo}@stu.hubu.edu.cn, {zhifei1993, zhangmiao, zhangyan, xiaokui}@hubu.edu.cn

Abstract

Multimodal knowledge graphs (MMKG) store structured world knowledge enriched with multimodal descriptive information. However, MMKG often faces the challenge of incompleteness. The primary objective of multimodal knowledge graph completion (MMKGC) is to predict missing entities within MMKG. Current MMKGC methods struggle with addressing the issue of over-trust attention and how to enhance the robustness of the model. To overcome these problems, we introduce APKGC, a noise-enhanced multimodal method for knowledge graph completion with attention penalty. APKGC effectively adjusts the attention scores in the language model and alleviates over-trust attention through a specifically designed attention penalty module. Additionally, an adaptive noise sampling module is proposed to supplement the entity's multimodal information, thereby enhancing the model's robustness. Experimental evaluation demonstrates that APKGC excels in overcoming these challenges. Compared to the existing state-of-the-art MMKGC model, APKGC improves Hit@1 by 3.3% on the DB15K dataset and by 3.4% on the MKG-W dataset.

Code — https://github.com/HubuKG/APKGC

Introduction

Knowledge graphs (KGs), such as WordNet (Miller 1995), DB15K (Liu et al. 2019), and MKG-W (Xu et al. 2022a), have shown significant effectiveness in natural language processing tasks. Their broad application spans various fields, including intelligent conversational agents (Dinan et al. 2019), series prediction (Deng et al. 2019), and recommendation systems (Gao et al. 2023; Liu et al. 2021). Human-curated knowledge graphs typically suffer from incompleteness (Socher et al. 2013), which inevitably constrains their practical applications. To address this limitation, knowledge graph completion (KGC) aims to predict the missing triples in a knowledge graph. Various approaches have been proposed for KGC, employing different coding strategies. These strategies include graph convolutional network (GCN) models, matrix factorization-based models, translation-based models, and convolutional neural network (CNN) models (Ji et al. 2022). However, the aforementioned KGC methods concentrate solely on the structural information within the knowledge graph, neglecting the presence of multimodal information in real-life knowledge.

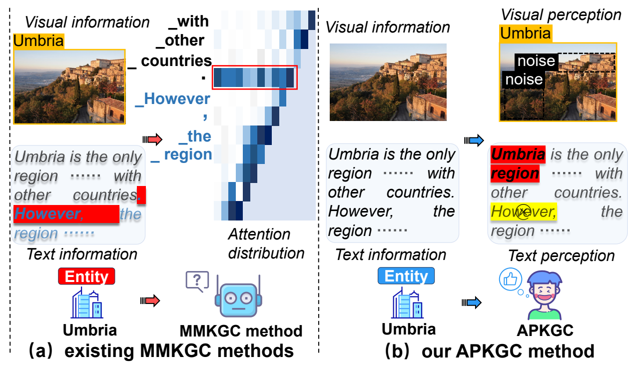

Figure 1: Intuitions of existing MMKGC methods and APKGC.

Recent research endeavors have introduced multimodal knowledge graphs (MMKGs) (Liu et al. 2019) to harness the diverse information present in the real world. MMKGs encapsulate diverse and complex world knowledge as structured triples (head entity, relation, tail entity) while integrating multimodal data, such as images and text, to provide additional context for entities. Multimodal knowledge graph completion (MMKGC) aims to automatically identify new knowledge from the provided MMKGs. Existing MMKGC methods (Cao et al. 2022; Li et al. 2024; Zhang et al. 2024a) represent modality information as single embeddings derived from pre-trained models (Devlin et al. 2019; Dosovitskiy et al. 2021), utilizing a fusion and prediction module to assess the plausibility of triples. These pre-trained language models have been widely employed in knowledge graph completion tasks, achieving favorable results.

Despite the advancements demonstrated by previous research on MMKGC over traditional methods, these approaches still encounter substantial limitations: **(1) Lack of robustness.** The real world contains a significant amount

[*]Corresponding Author.

of noise, and entities in the knowledge graph may encompass numerous extraneous data. However, most previous MMKGC methods treat noise as an interference factor, concentrating on rejecting and combating it within MMKGs. This approach may result in reduced model robustness. **(2) Attention partial over-trust.** Most current MMKGC methods employ pre-trained language models, which have been demonstrated to exhibit the attention partial over-trust problem due to multi-modal illusion when processing multi-modal data (Huang et al. 2023). As shown in Figure 1(a), for the entity text data in the knowledge graph, the language model assigns a high attention score to the token full stop[·], which lacks clear meaning during the training process. This attention to partial over-trust results in limitations in the learning of entity features, consequently affecting the performance of the MMKGC task.

To address the aforementioned challenges, this paper proposes a method called Noise-Enhanced Multi-Modal **K**nowledge **G**raph **C**ompletion with **A**ttention **P**enalty (APKGC). APKGC utilizes a pre-trained language model to encode multi-modal data and proposes the following two strategies: (1) **Adaptive noise sampling.** As illustrated in Figure 1(b), during the training process, we use the noise from image and text information to sample the multi-modal information of knowledge graph entities. To simulate real-world conditions, the noise sampling is set adaptively, dynamically adjusting the amount and proportion of sampled data throughout the training process. This approach leverages noisy data to enhance the robustness of the model. (2) **Attention over-trust penalty.** As shown in Figure 1(b), during the training process, we penalize the attention scores of certain tokens that generate over-trust to correct the attention errors produced by the language model, allowing the model to focus more on tokens with greater semantic importance. Through the carefully designed modules, APKGC effectively utilizes entity noise data and alleviates the problem of attention over trust, thereby achieving better results on MMKGC tasks. The primary contributions of our research are outlined as follows:

- We introduce a novel hybrid model, APKGC, that leverages pre-trained language models and integrates supplementary noise data to process multimodal information, thereby improving the model's robustness.
- We present a novel attention penalty strategy designed to penalize tokens with disproportionately high attention scores in the language model input, addressing the issue of attention over trust in multimodal language models.
- We perform extensive experiments on two separate public datasets. For the Hits@1 metric, APKGC improves the baseline model by 3.3% on the DB15K dataset and by 3.4% on the MKG-W, achieving SOTA performance.

Related Work

This section offers a succinct summary of unimodal knowledge graph completion models and current multimodal knowledge graph completion models.

Unimodal Knowledge Graph Completion

The conventional unimodal knowledge graph completion approach primarily utilizes graph embedding techniques. This method involves mapping entities and relations within the knowledge graph to a low-dimensional vector space. In translation-based methods, such as TransE, TransR, and RotatE, the plausibility of a triplet is determined by applying a translation function to the embeddings of the entity and the relation. In translation-based methods, such as TransE (Bordes et al. 2013), RotatE (Sun et al. 2019), and PairRE (Chao et al. 2021), the plausibility of a triplet is determined by applying a translation function to the embeddings of the entity and the relation. Matrix factorization-based models, such as DistMult (Yang et al. 2014), ComplEx (Trouillon et al. 2016), and TuckER (Balazevic, Allen, and Hospedales 2019), utilize tensor factorization techniques to learn knowledge graph representations. CNN-based models utilize convolutional neural networks to generate more powerful embeddings, and Examples include ConvE (Dettmers et al. 2018), ConvKB (Nguyen et al. 2018), and M-DCN (Zhang et al. 2022). In recent years, several studies have utilized graph neural networks to exploit the inherent structure of graphs (Wang et al. 2022; Li et al. 2022).

Most unimodal KGC methods primarily emphasize the structural and relational information within the knowledge graph, often overlooking the abundant multimodal information associated with real-world entities. The integration of multimodal information, such as images and text, into knowledge graphs, has become a prominent research focus.

Multimodal Knowledge Graph Completion

In recent years, multimodal knowledge graphs have garnered increasing attention. The multimodal knowledge graph completion method leverages information from various modalities within MMKG to enable the MMKGC method to achieve more comprehensive entity representations, thereby enhancing prediction accuracy. Some existing MMKGC methods focus on multimodal fusion and interaction, aiming to efficiently integrate information from different modalities (Xie et al. 2017; Cao et al. 2022; Sergieh et al. 2018; Zhang et al. 2024a). These methods seek to reduce model heterogeneity and optimize the utilization of multimodal information. The negative sampling methods (Xu et al. 2022b; Zhang, Chen, and Zhang 2023; Zhang et al. 2024b) aim to enhance the negative sampling process by incorporating multimodal information of entities to generate high-quality negative samples. The integrated decision approach (Li et al. 2023) typically involves learning a discriminative model for each modality and then ensemble them to make joint decisions. Some studies (Chen et al. 2024) design a unified framework and incorporate noise to enhance entity feature learning.

Unlike the above methods, APKGC adaptively integrates modal noise data and uses the attention penalty strategy to alleviate the language model's attention overtrust. This strategy can extract entity features more accurately to complete the prediction.

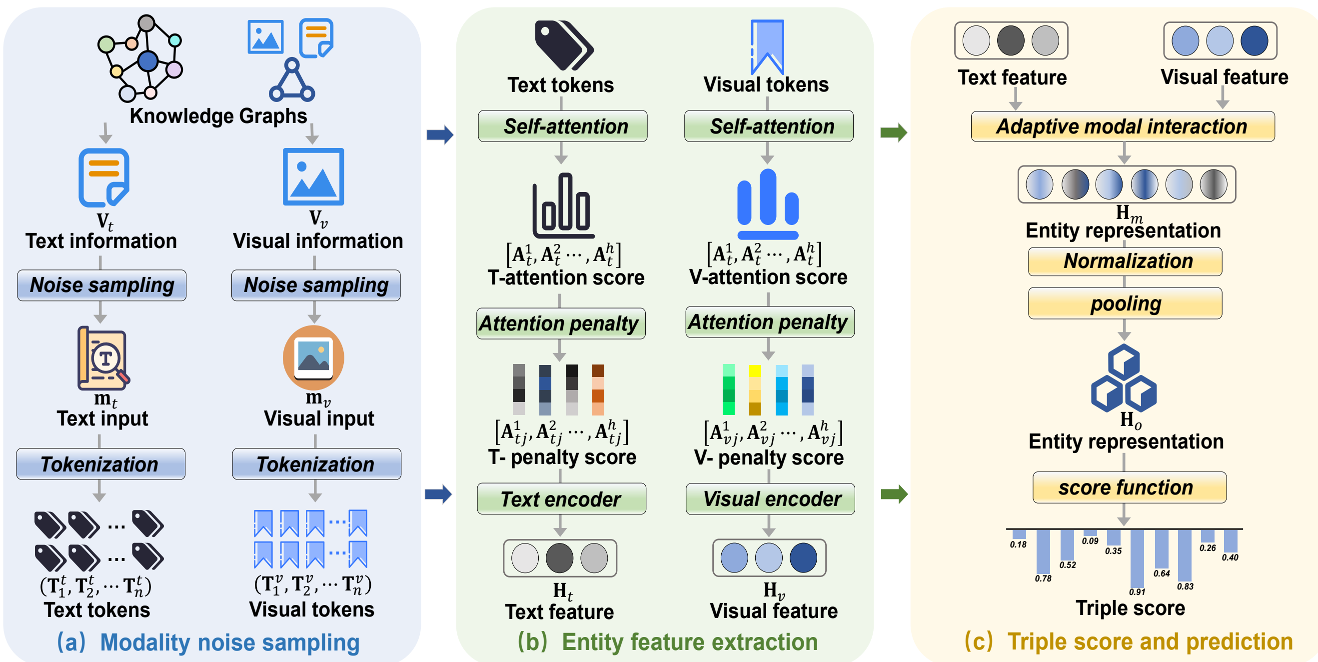

Figure 2: The overall framework of APKGC. (a) Modal noise sampling: The multimodal information in the knowledge graph is processed, and the noise data is adaptively sampled, generating the input token. (b) Entity feature extraction: Utilize a pre-trained model to extract entity features from the input multimodal data for subsequent predictions. (c) Triple score and prediction: Perform the prediction of missing entities through modal interaction and obtain the score function after the prediction vector is derived through vector processing.

Methodology

This section provides a detailed exposition of APKGC. As depicted in Figure 2, APKGC features a comprehensive architecture comprising three primary modules: (1) Modality Noise Sampling: Obtaining multimodal information containing noise. (2) Entity Feature Extraction: Extracting entity features that learn from the entity information. (3) Triple Score and Prediction: Processing the prediction vectors and obtaining the triple score.

Preliminaries

We define the knowledge graph (KG), denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, as a directed structure where $\mathcal{E}$ represents the set of entities, $\mathcal{R}$ denotes the set of relations, and $\mathcal{T}$ signifies the set of triplets. Each triplet is expressed as (head entity, relation, tail entity), i.e., (h, r, t). In multimodal knowledge graphs (MMKGs), each entity is characterized by multiple features derived from different modalities. The set of modalities is defined as $\mathcal{M} = \{t, v, m\}$, where t, v, and m represent the textual, visual, and modality, respectively.

Modality Noise Sampling

The purpose of Modality Noise Sampling is to process the multimodal information in the knowledge graph and incorporate adaptive sampling noise data to obtain the input vector required by the language model. We choose the pre-trained BERT as the APKGC text encoder to encode the text information.

For a given triplet (h, r, t) in the knowledge graph $\mathcal{G}$, we extract the text vectors $\mathbf{h}_t$ and $\mathbf{r}_t$ of the head entity and relation and obtain the sentence-level text description $\mathbf{d}_h$ of the head entity. The text input vector is obtained as follows:

$$\mathbf{V}_t = \mathbf{h}_t + \mathbf{r}_t + \mathbf{d}_h, \tag{1}$$

here, $\mathbf{V}_t$ is the preliminary text vector we obtain. For the visual information of the head entity $\mathbf{h}_v$, the Beit (Bao et al. 2022) model is utilized as the visual encoder in the APKGC framework. We divide the input image into patches at specific size intervals, as follows:

$$\mathbf{V}_v = (\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n) = \mathrm{patch}(\mathbf{h}_v), \tag{2}$$

where $\mathbf{V}_v$ is the preliminary visual vector we obtain, $(\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n)$ represents the patches acquired following the segmentation operation $\mathrm{patch}(\cdot)$.

Adaptive Noise Sampling. Recent studies in MMKG (Chen et al. 2023; Zhang et al. 2024b; Chen et al. 2024) have shown that models can tolerate a certain level of noise, which enhances robustness without significantly reducing the expressive power of multimodal entity representations. Inspired by these studies, we design an adaptive sampling strategy that introduces noisy data to address the lack of robustness in models for multimodal knowledge graphs, thereby enhancing the quality of the acquired information.

Specifically, we adopt two different sampling strategies in the model for different datasets: Gaussian noise sampling and average noise sampling. Gaussian noise sampling introduces random perturbations to enhance generalization, whereas average noise sampling generates smoother noise, mitigating the influence of outliers. The Gaussian noise sampling is calculated as follows:

$$\mathbf{N}_g = \theta_r \left(\varphi_m \odot g + \mu_m\right), g \sim K(0, Z), \tag{3}$$

where θ_r represents a learnable noise scaling factor, which adaptively changes the proportion of noise in the embedding during training. φ_m represents the standard deviation of the original noise-embedded data, and μ_m represents the mean value of the original noise-embedded data. $K(0, Z)$ is a Gaussian distribution with a mean vector of 0 and a unit covariance matrix Z. g is the sample value drawn from the Gaussian distribution K. $\mathbf{N}_g$ represents the noisy input vector obtained after Gaussian noise sampling. The formula for the average noise sampling is as follows:

$$\mathbf{N}_a = \theta_r \left[\beta_{min} + (\beta_{max} - \beta_{min}) \odot U\right], \tag{4}$$

where θ_r represents the learnable noise scale factor, β_{max} represents the maximum value of the random variable interval set, β_{min} represents the minimum value of the random variable interval set, and U is a random variable that obeys the uniform distribution on the interval $[\beta_{min}, \beta_{max}]$, $\mathbf{N}_a$ denotes the noisy input vector obtained after average noise sampling. To determine the proportion of vectors sampled with or without noise, we employ an adaptive random adjustment strategy as follows:

$$\mathbf{V}_n = \begin{cases} (1-\theta_r)\,\mathbf{V}_o + \mathbf{N}_m & if\ \rho\ > \theta_n \\ \mathbf{V}_o & otherwise \end{cases}, \tag{5}$$

here, to implement the adaptive noise sampling strategy, the random factor ρ is established and compared with the sampling factor θ_n of the embedding vector. The random factor ρ is uniformly assigned within the range [0, 1] for each entity, while the sampling factor θ_n is adaptively tuned within the interval [0, 1] during the training process. This comparison determines whether to add noise sampling $\mathbf{N}_m$ to the initial input vector, generating a judgment result. Based on this result, noise is added to the sample $\mathbf{N}_m$.

The adaptive noise sampling strategy we designed introduces modal noise into the multimodal information of the original knowledge graph. It employs a dynamically adjusted random factor ρ and a noise scaling factor θ_r to dynamically control the size and proportion of noise during the training process. This adaptive noise sampling method enhances the robustness of the model.

Therefore, following noise sampling, the preliminary vector of multimodal information obtains a more robust information vector as described below:

$$\begin{cases} \mathbf{m}_t = \mathrm{noise}\left(\mathbf{V}_t\right) \\ \mathbf{m}_v = \mathrm{noise}\left(\mathbf{V}_v\right) \end{cases}, \tag{6}$$

here, $\mathrm{noise}(\cdot)$ represents the adaptive noise sampling operation we designed. while $\mathbf{m}_t$ and $\mathbf{m}_v$ denote the sampled vectors of text information and image information, respectively. Then, we use the pre-trained model tokenizer $\mathrm{tokenizer}(\cdot)$ to segment it, obtaining the input tokens$(\mathbf{T}_1^t, \mathbf{T}_2^t, \cdots \mathbf{T}_n^t)$ and $(\mathbf{T}_1^v, \mathbf{T}_2^v, \cdots \mathbf{T}_n^v)$ for the text and image, respectively.

$$\begin{cases} (\mathbf{T}_1^t, \mathbf{T}_2^t, \cdots \mathbf{T}_n^t) = \mathrm{tokenizer}\left(\mathbf{m}_t\right) \\ (\mathbf{T}_1^v, \mathbf{T}_2^v, \cdots \mathbf{T}_n^v) = \mathrm{tokenizer}\left(\mathbf{m}_v\right) \end{cases}. \tag{7}$$

Entity Feature Extraction

The objective of Entity Feature Extraction is to utilize a pre-trained model to extract entity features from the input multimodal data for subsequent prediction. For the input token derived in the preceding subsection, the self-attention mechanism is employed to compute the attention score as follows:

$$\begin{cases} \left[\mathbf{A}_t^1, \mathbf{A}_t^2 \cdots, \mathbf{A}_t^h\right] = \mathrm{attention}\left(\mathbf{T}_1^t, \mathbf{T}_2^t, \cdots \mathbf{T}_n^t\right) \\ \left[\mathbf{A}_v^1, \mathbf{A}_v^2 \cdots, \mathbf{A}_v^h\right] = \mathrm{attention}\left(\mathbf{T}_1^v, \mathbf{T}_2^v, \cdots \mathbf{T}_n^v\right) \end{cases}, \tag{8}$$

here, $\mathrm{attention}(\cdot)$ denotes the self-attention computation we executed, while $[\mathbf{A}_t^1, \mathbf{A}_t^2 \cdots, \mathbf{A}_t^h]$ and $[\mathbf{A}_t^1, \mathbf{A}_t^2 \cdots, \mathbf{A}_t^h]$ represent the preliminary attention scores calculated.

Attention Over-Trust Penalty. Previous research has established that the multi-modal large language model can generate multi-modal illusions when processing multi-modal data, leading to the issue of attention partial over-trust (Huang et al. 2023). This problem also arises when employing the language model in the MMKGC model. To address the aforementioned issues, we propose an over-trust attention penalty strategy within the attention calculation process. Specifically, for tokens exhibiting over-trust with excessive attention, we implement the following operations:

$$\mathbf{A}_j^p = \mathbf{A}_i^p - \theta_p\left(\mathbf{A}_i^p\right), if\ \mathbf{A}_i^p\ > \phi, \tag{9}$$

here, $\mathbf{A}_i^p$ denotes the original attention score before penalization, θ_p represents the attention penalty ratio value, ϕ signifies the set attention punishment threshold, and $\mathbf{A}_j^p$ represents the attention score obtained after the penalization operation. After penalizing the excessive attention value, we address the dilution of attention value caused by an over-trusted token by redistributing the attention values of its surrounding tokens as follows:

$$\mathbf{A}_j^n = \mathbf{A}_i^n + \theta_j^n\left(\mathbf{A}_i^p - \phi\right), n \sim (p-k, p+k), \tag{10}$$

where $\mathbf{A}_i^n$ represents the original attention value before reassignment, k denotes the set attention reassignment window value, θ_j^n signifies the attention reassignment weight value, and $\mathbf{A}_j^n$ represents the attention value obtained after reassignment.

By incorporating the attention over-trust penalty module, we can dynamically adjust the attention value of tokens with excessive attention during the language model's attention calculation process. This redistribution alleviates over-trust in attention, leading to more accurate entity features and influencing the prediction choices.

With the aforementioned attention penalty module, we can derive the penalized attention score vector as follows:

$$\begin{cases} \mathbf{V}_t^p = \left[\mathbf{A}_{tj}^1, \mathbf{A}_{tj}^2 \cdots, \mathbf{A}_{tj}^h\right] = \mathrm{penalty}\left[\mathbf{A}_t^1, \mathbf{A}_t^2 \cdots, \mathbf{A}_t^h\right] \\ \mathbf{V}_v^p = \left[\mathbf{A}_{vj}^1, \mathbf{A}_{vj}^2 \cdots, \mathbf{A}_{vj}^h\right] = \mathrm{penalty}\left[\mathbf{A}_v^1, \mathbf{A}_v^2 \cdots, \mathbf{A}_v^h\right] \end{cases}, \tag{11}$$

where $\mathrm{penalty}(\cdot)$ represents the attention punishment operation we perform, $\mathbf{V}_t^p$ and $\mathbf{V}_v^p$ represent the obtained attention vector after the punishment. Subsequently, we derive the multi-modal entity feature vectors $\mathbf{H}_t$ and $\mathbf{H}_v$ through encoding as follows:

$$\begin{cases} \mathbf{H}_t = \mathrm{encoder}\left(\mathbf{V}_t^p\right) \\ \mathbf{H}_v = \mathrm{encoder}\left(\mathbf{V}_v^p\right) \end{cases}, \tag{12}$$

here, $\mathrm{encoder}(\cdot)$ represents the sequence of encoding operations we perform with the language model.

Triple Score and Prediction

In this subsection, we predict missing entities by modal interaction and obtain the score function after the prediction vector is obtained by vector processing. A loss function was established to train the model iteratively.

Modality Interaction. We achieve adaptive multimodal interaction through the designed multi-head attention mechanism as follows:

$$\mathbf{Q}_a = \mathbf{H}^a \mathbf{W}_q^a; \mathbf{K}_a = \mathbf{H}^a \mathbf{W}_k^a; \mathbf{V}_a = \mathbf{H}^a \mathbf{W}_v^a, \tag{13}$$

here, $\mathbf{H}^a$ represents the attention input, while $\mathbf{Q}_a$, $\mathbf{K}_a$, and $\mathbf{V}_a$ are the query matrix, key matrix, and value matrix, respectively, obtained by transforming the shared matrices $\mathbf{W}_q^a$, $\mathbf{W}_k^a$, and $\mathbf{W}_v^a$. Unlike the traditional multi-head attention calculation, we introduce the modal correlation weighting factor θ_{ab}, which is calculated as follows:

$$\theta_{ab} = \frac{\exp\left(\mathbf{Q}_a^\top \mathbf{K}_b / \sqrt{d}\right)}{\exp\left(\frac{\mathbf{Q}_a^\top \mathbf{K}_a}{\sqrt{d}}\right) + \exp\left(\frac{\mathbf{Q}_a^\top \mathbf{K}_b}{\sqrt{d}}\right)}, \tag{14}$$

where θ_{ab} represents the correlation weight between modalities a and b, and d denotes the hidden state dimension. We then incorporate the modal correlation into the multi-head attention computation as follows:

$$\mathbf{V}_a^i = \theta_{ab} \mathbf{V}_a, \tag{15}$$

here $\mathbf{V}_a^i$ represents the acquired value matrix with relevance information. So the attention value $\mathbf{A}_a$ is calculated as follows:

$$\mathbf{A}_a = \mathrm{softmax}\left(\frac{\mathbf{Q}_a \left(\mathbf{K}_a\right)^\top}{\sqrt{d}}\right) \mathbf{V}_a^i. \tag{16}$$

MMKGC Prediction. Through the aforementioned modality interaction, we derive the language prediction head $\mathbf{H}_m$ as follows:

$$\mathbf{H}_m = \mathrm{interaction}\left(\mathbf{H}_t, \mathbf{H}_v\right), \tag{17}$$

where $\mathrm{interaction}(\cdot)$ denotes the modality interaction operation conducted. Subsequently, we normalize the vectors to obtain the final prediction vector $\mathbf{H}_o$.

$$\mathbf{H}_o = \mathrm{normalization}(\mathbf{H}_m), \tag{18}$$

in this process, we utilize L2 normalization. We utilize the RotatE (Sun et al. 2019) model as our scoring function to evaluate the plausibility score of triples as follows:

$$\mathcal{S}(h, r, t) = ||\bar{h} \circ r - \bar{t}||, \tag{19}$$

where $\circ$ denotes a rotation operation in complex space. $\mathcal{S}(h, r, t)$ represents the obtained triple score. We utilize a sigmoid-based loss function (Sun et al. 2019) to optimize the embedding as follows:

$$\begin{aligned}\mathcal{L} = \frac{1}{|\mathcal{T}|} \sum_{(h,r,t)\in\mathcal{T}} &(-\log\left[\sigma(\lambda - S(h, r, t)\right)\\ &- \sum_{i=1}^{E} v_i \sigma(\mathcal{S}(h', r', t') - \lambda)^{\phi}]),\end{aligned} \tag{20}$$

where λ denotes the margin, E represents the number of negative samples for each positive sample, v_i is the self-adversarial weight, and σ is the sigmoid function.

Datasets	DB15K	MKG-W
# Entities	12,842	15,000
# Relations	279	169
# Train	79,222	34,196
# Valid	9,902	4,276
# Test	9,904	4,274
# Total	99,028	42,746

Table 1: Statistics of the experimental datasets.

Experiments

In this section, we conducted an extensive series of experiments using two publicly available datasets to evaluate the effectiveness of the proposed APKGC model. The empirical findings address the following four research questions:

- RQ1: How does the performance of APKGC compare to other MMKGC models?
- RQ2: How does the inclusion of modal information influence the performance of APKGC?
- RQ3: How do the primary modules impact the performance of APKGC?
- RQ4: How do different hyperparameter configurations affect the performance of APKGC?
- RQ5: How does APKGC perform when tested on real-world datasets?

Experimental Settings

Datasets: In this study, we utilize two publicly available MMKGC benchmarks, DB15K (Liu et al. 2019) and MKG-W (Xu et al. 2022a), to evaluate model performance. The relevant statistical details of these datasets are summarized in Table 1. DB15K is derived from DBPedia(Lehmann et al. 2015) and enhanced with images obtained through a search engine. MKG-W comprises subsets of Wikidata (Vrandecic and Krötzsch 2014) and YAGO (Suchanek, Kasneci, and Weikum 2007). The raw data for each modality are sourced from their official releases.

Evaluation Metrics: The evaluation methodology incorporates four automated metrics: Mean Reciprocal Rank (MRR) and Hits@N (N=1, 3, 10). These metrics are evaluated in a filtered setting as described in previous research (Wang et al. 2014), where scores for known valid triplets from the training, validation, and test sets are excluded from consideration.

Baseline Models: To demonstrate the effectiveness of our proposed model, we conducted a comparative analysis against a range of established and prominent KGC models. These baseline models are categorized into two groups: (1) Unimodal KGC model: **TransE** (Bordes et al. 2013), **DistMult** (Yang et al. 2014), **ComplEx** (Trouillon et al. 2016), **RotatE** (Sun et al. 2019), and **pairRE** (Chao et al. 2021); (2) Multimodal KGC model: **IKRL** (Xie et al. 2017), **TransAE** (Wang et al. 2019), **RSME** (Wang et al. 2021), **VBKGC** (Zhang and Zhang 2022), **OTKGE** (Cao et al. 2022), **IMF** (Li et al. 2023), **QEB** (Wang et al. 2023), **VISTA** (Lee

Methods		DB15K				MKG-W			
		MRR	Hits			MRR	Hits		
			@1	@3	@10		@1	@3	@10
Unimodal	TransE (NeurIPS'13)	0.249	0.128	0.315	0.471	0.292	0.211	0.332	0.442
	DistMult (ICLR'15)	0.230	0.148	0.263	0.396	0.210	0.159	0.223	0.309
	ComplEx (ICML'16)	0.275	0.184	0.316	0.454	0.249	0.191	0.267	0.367
	RotatE (ICLR'19)	0.293	0.179	0.361	0.497	0.337	0.268	0.367	0.460
	PairRE (ACL'21)	0.311	0.216	0.359	0.493	0.344	0.282	0.367	0.460
Multimodal	IKRL (IJCAI '17)	0.268	0.141	0.349	0.491	0.324	0.261	0.348	0.441
	TransAE (IJCNN '19)	0.281	0.213	0.312	0.412	0.300	0.212	0.349	0.447
	RSME (ACM MM '21)	0.298	0.242	0.321	0.403	0.292	0.234	0.320	0.404
	VBKGC (KDD '22)	0.306	0.198	0.372	0.494	0.306	0.249	0.330	0.409
	OTKGE (NeurIPS '22)	0.239	0.185	0.259	0.342	0.344	0.289	0.363	0.449
	IMF (WWW '23)	0.323	0.242	0.360	0.482	0.345	0.288	0.366	0.454
	QEB (ACM MM '23)	0.282	0.148	0.367	0.516	0.324	0.255	0.351	0.453
	VISTA (EMNLP '23)	0.304	0.225	0.336	0.459	0.329	0.261	0.354	0.456
	MANS (IJCNN '23)	0.288	0.169	0.366	0.493	0.309	0.249	0.336	0.418
	MMRNS (ACM MM '22)	0.297	0.179	0.367	0.510	0.341	0.274	0.375	0.468
	AdaMF (COLING '24)	0.325	0.213	0.397	0.517	0.343	0.272	0.379	0.472
	SnAg (Arixv '24)	0.342	0.250	0.391	0.515	0.359	0.284	0.394	0.492
	NativE (SIGIR '24)	<u>0.362</u>	<u>0.273</u>	<u>0.411</u>	**0.529**	<u>0.366</u>	<u>0.296</u>	<u>0.395</u>	<u>0.494</u>
	APKGC (Ours)	**0.364**	**0.282**	**0.413**	<u>0.527</u>	**0.374**	**0.306**	**0.404**	**0.501**

Table 2: Performance comparison of different KGC models. The optimal results are highlighted in bold, while the second-best results are underlined.

Modal	MRR	H@1	H@3	H@10
Structure	0.211	0.125	0.247	0.345
Visual	0.336	0.261	0.365	0.469
Textual	0.348	0.273	0.392	0.505
Structure & Visual	0.340	0.260	0.376	0.480
Textual & Structure	0.351	<u>0.279</u>	0.396	0.509
Visual & Textual	<u>0.357</u>	0.274	<u>0.405</u>	<u>0.521</u>
APKGC	**0.364**	**0.282**	**0.411**	**0.527**

Table 3: Ablation study on the different modalities of APKGC.

et al. 2023), **MANS** (Zhang, Chen, and Zhang 2023), **MMRNS** (Xu et al. 2022b), **AdaMF** (Zhang et al. 2024b), **SnAg** (Chen et al. 2024), **NativE** (Zhang et al. 2024a).

Parameter Settings: The APKGC model utilizes PyTorch [47] and operates on an Nvidia RTX 4090 GPU. For the training, the embedding dimension is set to 128, the batch size is set to 2048, the number of negative samples is set to 64, and the learning rate is set to 1e-4. For the DB15K dataset, Gaussian noise sampling is employed, while for the MKG-W dataset, average noise sampling is utilized.

Performance Comparison (RQ1)

As presented in Table 2, the results of the Knowledge Graph Completion (KGC) task demonstrate that APKGC surpasses the current state-of-the-art (SOTA) model across most evaluation metrics when compared to the baseline model. Notably, it outperforms recent approaches such as AdaMF (Zhang et al. 2024b) and SnAg (Chen et al. 2024), both of which utilize noisy data. Specifically, the Hits@1 shows an increase of 3.3% (from 0.273 to 0.282) on the MKG-W dataset and 3.4% (from 0.296 to 0.306) on the DB15K dataset. Additionally, the other metrics also demonstrate improvements compared to the SOTA model.

APKGC employs a pre-trained language model to mitigate the issue of attention overtrust caused by multimodal illusion through a specifically designed attention penalty module. Additionally, it utilizes an adaptive noise sampling module to enhance the model's robustness. APKGC's superior performance compared to other multimodal models suggests promising research directions for addressing the multimodal illusion problem in the MMKGC task.

Effects of Modality (RQ2)

A study was conducted on the APKGC model to examine the influence of different modalities on MMKGC outcomes. This study compared the complete APKGC with six variants, and the experimental results on the DB15K dataset are presented in Table 3. The results indicate that the APKGC model achieves optimal performance when all modal information is considered. Specifically, the exclusion of visual information results in a 3.4% decrease in Hits@10, while the exclusion of textual information leads to an 8.9% decrease in Hits@10.

Overall, the results confirm APKGC's effectiveness in utilizing multimodal information and underscore the critical importance of each modality for MMKGC. By leveraging textual and visual inputs, APKGC illustrates integrating multimodal data can enhance entity feature extraction.

Methods	MRR	H@1	H@3	H@10
w/o noise	0.358	0.289	0.393	0.486
w/o penalty	0.360	0.286	0.397	0.492
w/o increase	0.365	0.297	0.399	0.491
w/o decrease	<u>0.369</u>	<u>0.302</u>	<u>0.398</u>	<u>0.495</u>
APKGC	**0.374**	**0.306**	**0.404**	**0.501**

Table 4: Ablation study on key components of APKGC.

Key Components (RQ3)

To evaluate the significance of specific components in the APKGC framework, we analyzed four variants: (1) excluding the noise sampling layer, (2) without the attention penalty module, (3) excluding the attention increase layer, and (4) without the attention decrease layer. These modifications aimed to isolate and assess the contributions of key elements to the model's performance. The impacts on the performance of these variants are documented in Table 4.

The results show that without the attention penalty module, the model performance significantly degrades. Moreover, the lack of attention increase and attention decrease strategies in the penalty module will affect the model performance. These results verify the effectiveness of the attention punishment strategy developed in this study, which significantly alleviates the attention over-trust problem. However, the performance of the model decreases significantly when there is no noise sampling, which shows that the adaptive noise sampling of APKGC improves the performance of MMKGC.

Hyperparameter Analysis (RQ4)

The number of penalty windows k and the penalty ratio value θ_p: Figure 3 shows the impact of The number of penalty windows k and the penalty ratio value θ_p. The results indicate that a penalty ratio or penalty window that is too small results in insufficient attention punishment, leading to suboptimal model performance. Conversely, a penalty ratio or penalty window that is too large causes overcorrection in attention punishment, thereby reducing the accuracy of entity feature extraction. Therefore, we selectively determine the optimal number of penalty windows and the penalty ratio to conduct the MMKGC experiment.

Case Study (RQ5)

To examine the interpretability of APKGC, we conducted case studies on representative triples from the DB15K dataset, as illustrated in Figure 4. This figure displays the attention scores and the top three predicted entities in the model without the attention penalty strategy, as well as in the complete APKGC model. For the prediction triplet (`National Society of Film Critics Award for Best Supporting Actor`, `award`, ?), the model without the attention penalty received the original attention score, predicting the correct answer `Dennis Hopper` as the second predicted entity. In contrast, the APKGC model with the attention penalty successfully identified the correct entity as the first predicted entity.

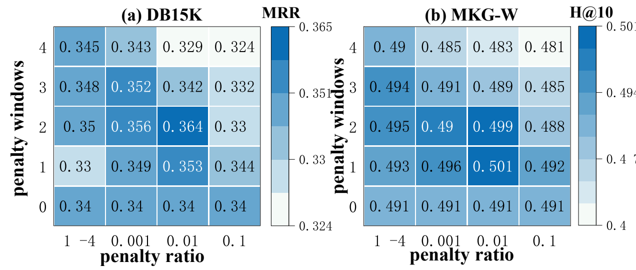

Figure 3: A hyper-parameter analysis is performed on The number of penalty windows k and the penalty ratio value θ_p for datasets (a) DB15K and (b) MKG-W.

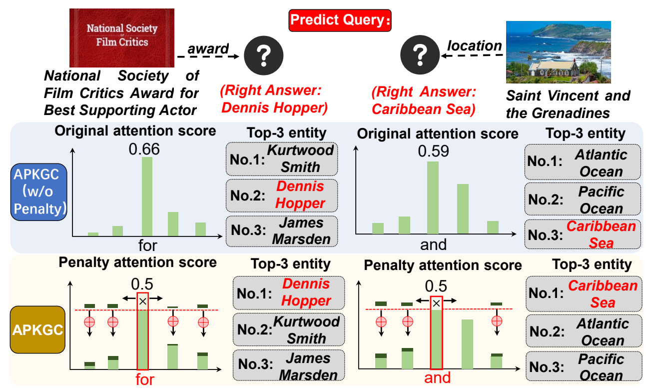

Figure 4: Case examples of APKGC. The top-3 predicted entities represent those ranked in the top three positions in the prediction outcomes.

Similarly, for predicting triplet (?, `location` , `Saint Vincent and the Grenadines`), the complete model with the attention penalty outperformed the model without it. These examples demonstrate the effectiveness of our attention penalty strategy.

Conclusion

This paper introduces the APKGC, a noise-enhanced multi-modal method for knowledge graph completion with an attention penalty. APKGC investigates the impact of multi-modal illusion on language models and introduces an attention penalty strategy to mitigate language model attention dependence, thereby achieving more accurate entity feature extraction. Additionally, an adaptive noise sampling strategy is proposed to enhance the robustness of the model by improving the quality of entity multi-modal data. Experimental evaluations on two widely used datasets demonstrate that APKGC outperforms current state-of-the-art methods in the MMKGC task.

For future work, we aim to delve deeper into the MMKGC task, building upon our APKGC research. We believe that incorporating additional modal information, such as video and audio data, into the MMKGC model presents a promising research direction. Therefore, incorporating more kinds of modal information into the APKGC model may be a future research direction for us.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 62207011, 62377009, 62407013), the Open Research Fund of Key Laboratory of Intelligent Sensing System and Security of Hubei University, Ministry of Education (No. KLISSS202410), the Ministry of Education of Humanities and Social Science project (NO. 22YJCZH224), the Key Research and Development Project of Hubei Province (NO. 2021BAA184), the Major Program (JD) of Hubei Province (NO. 2023BAA018).

References

Balazevic, I.; Allen, C.; and Hospedales, T. M. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*, 5188–5197.

Bao, H.; Dong, L.; Piao, S.; and Wei, F. 2022. BEiT: BERT Pre-Training of Image Transformers. In *Proceedings of the Tenth International Conference on Learning Representations*.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 2787–2795.

Cao, Z.; Xu, Q.; Yang, Z.; He, Y.; Cao, X.; and Huang, Q. 2022. OTKGE: Multi-modal Knowledge Graph Embeddings via Optimal Transport. In *Proceedings of The Annual Conference on Neural Information Processing Systems*.

Chao, L.; He, J.; Wang, T.; and Chu, W. 2021. PairRE: Knowledge Graph Embeddings via Paired Relation Vectors. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 4360–4369.

Chen, Z.; Fang, Y.; Zhang, Y.; Guo, L.; Chen, J.; Chen, H.; and Zhang, W. 2024. The Power of Noise: Toward a Unified Multi-modal Knowledge Graph Representation Framework. *CoRR*, abs/2403.06832.

Chen, Z.; Guo, L.; Fang, Y.; Zhang, Y.; Chen, J.; Pan, J. Z.; Li, Y.; Chen, H.; and Zhang, W. 2023. Rethinking Uncertainly Missing and Ambiguous Visual Modality in Multi-Modal Entity Alignment. In *Proceedings of the 22nd International Semantic Web Conference*, volume 14265, 121–139.

Deng, S.; Zhang, N.; Zhang, W.; Chen, J.; Pan, J. Z.; and Chen, H. 2019. Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network. In *Proceedings of The 2019 World Wide Web Conference*, 678–685.

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

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Dinan, E.; Roller, S.; Shuster, K.; Fan, A.; Auli, M.; and Weston, J. 2019. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. In *Proceedings of The 7th International Conference on Learning Representations*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the9th International Conference on Learning Representations*.

Gao, J.; Peng, P.; Lu, F.; Claramunt, C.; and Xu, Y. 2023. Towards travel recommendation interpretability: Disentangling tourist decision-making process via knowledge graph. *Information Processing & Management*, 60(4): 103369.

Huang, Q.; Dong, X.; Zhang, P.; Wang, B.; He, C.; Wang, J.; Lin, D.; Zhang, W.; and Yu, N. 2023. OPERA: Alleviating Hallucination in Multi-Modal Large Language Models via Over-Trust Penalty and Retrospection-Allocation. *CoRR*, abs/2311.17911.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Yu, P. S. 2022. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2): 494–514.

Lee, J.; Chung, C.; Lee, H.; Jo, S.; and Whang, J. J. 2023. VISTA: Visual-Textual Knowledge Graph Representation Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 7314–7328.

Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; and Bizer, C. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2): 167–195.

Li, R.; Cao, Y.; Zhu, Q.; Bi, G.; Fang, F.; Liu, Y.; and Li, Q. 2022. How Does Knowledge Graph Embedding Extrapolate to Unseen Data: A Semantic Evidence View. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, 5781–5791.

Li, X.; Zhao, X.; Xu, J.; and and, Y. Z. 2023. IMF: Interactive Multimodal Fusion Model for Link Prediction. In *Proceedings of the ACM Web Conference 2023*, 2572–2580.

Li, Z.; Jian, Y.; Xue, Z.; Zheng, Y.; Zhang, M.; Zhang, Y.; Hou, X.; and Wang, X. 2024. Text-enhanced knowledge graph representation learning with local structure. *Information Processing & Management*, 61(5): 103797.

Liu, D.; Lian, J.; Liu, Z.; Wang, X.; Sun, G.; and Xie, X. 2021. Reinforced Anchor Knowledge Graph Generation for News Recommendation Reasoning. In *Proceedings of The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1055–1065.

Liu, Y.; Li, H.; García-Durán, A.; Niepert, M.; Oñoro-Rubio, D.; and Rosenblum, D. S. 2019. MMKG: Multi-modal

Knowledge Graphs. In *Proceedings of 16th International Conference on Semantic Web*, volume 11503, 459–474.

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the Acm*, 38(11): 39–41.

Nguyen, D. Q.; Nguyen, T. D.; Nguyen, D. Q.; and Phung, D. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, 327–333.

Sergieh, H. M.; Botschen, T.; Gurevych, I.; and Roth, S. 2018. A Multimodal Translation-Based Approach for Knowledge Graph Representation Learning. In Nissim, M.; Berant, J.; and Lenci, A., eds., *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, 225–234.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. Y. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In *Proceedings of The 27th Annual Conference on Neural Information Processing Systems*, 926–934.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, 697–706. ACM.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *Proceedings of the 7th International Conference on Learning Representations*, 1–18.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning*, 2071–2080.

Vrandecic, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10): 78–85.

Wang, H.; Dai, S.; Su, W.; Zhong, H.; Fang, Z.; Huang, Z.; Feng, S.; Chen, Z.; Sun, Y.; and Yu, D. 2022. Simple and Effective Relation-based Embedding Propagation for Knowledge Representation Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 2755–2761.

Wang, M.; Wang, S.; Yang, H.; Zhang, Z.; Chen, X.; and Qi, G. 2021. Is Visual Context Really Helpful for Knowledge Graph? A Representation Learning Perspective. In *Proceedings of the 29st ACM International Conference on Multimedia*, 2735–2743.

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*, 2391–2399.

Wang, Z.; Li, L.; Li, Q.; and Zeng, D. 2019. Multimodal Data Enhanced Representation Learning for Knowledge Graphs. In *Proceedings of the International Joint Conference on Neural Networks*, 1–8.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 1112–1119.

Xie, R.; Liu, Z.; Luan, H.; and Sun, M. 2017. Image-embodied Knowledge Representation Learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 3140–3146.

Xu, D.; Xu, T.; Wu, S.; Zhou, J.; and Chen, E. 2022a. Relation-enhanced Negative Sampling for Multimodal Knowledge Graph Completion. In *Proceedings of The 30th ACM International Conference on Multimedia*, 3857–3866.

Xu, D.; Xu, T.; Wu, S.; Zhou, J.; and Chen, E. 2022b. Relation-enhanced Negative Sampling for Multimodal Knowledge Graph Completion. In *Proceedings of the 30th ACM International Conference on Multimedia*, 3857–3866.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2014. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the 3rd International Conference on Learning Representations*, 1–12.

Zhang, Y.; Chen, M.; and Zhang, W. 2023. Modality-Aware Negative Sampling for Multi-modal Knowledge Graph Embedding. In *Proceedings of the International Joint Conference on Neural Networks*, 1–8.

Zhang, Y.; Chen, Z.; Guo, L.; Xu, Y.; Hu, B.; Liu, Z.; Zhang, W.; and Chen, H. 2024a. NativE: Multi-modal Knowledge Graph Completion in the Wild. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 91–101.

Zhang, Y.; Chen, Z.; Liang, L.; Chen, H.; and Zhang, W. 2024b. Unleashing the Power of Imbalanced Modality Information for Multi-modal Knowledge Graph Completion. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics*, 17120–17130.

Zhang, Y.; and Zhang, W. 2022. Knowledge Graph Completion with Pre-trained Multimodal Transformer and Twins Negative Sampling. *CoRR*, abs/2209.07084.

Zhang, Z.; Li, Z.; Liu, H.; and Xiong, N. N. 2022. Multi-Scale Dynamic Convolutional Network for Knowledge Graph Embedding. *IEEE Transactions on Knowledge and Data Engineering*, 34(5): 2335–2347.