Multimodal Knowledge Graph Completion via Relation-Aware Negative Sampling with Diffusion-Based Interpolation

Qian Ma
Dalian Maritime University
maqian@dlmu.edu.cn

Linfei Dai
Dalian Maritime University
dailinfei@dlmu.edu.cn

Zhongming Yao
Northeastern University
yaozming@stumail.neu.edu.cn

Yu Gu
Northeastern University
guyu@mail.neu.edu.cn

Tianyi Li
Aalborg University
tianyi@cs.aau.dk

Christian S. Jensen
Aalborg University
csj@cs.aau.dk

Ge Yu
Northeastern University
yuge@mail.neu.edu.cn

ABSTRACT

Multimodal Knowledge Graphs (MMKGs) enable structured reasoning across heterogeneous modalities and are essential infrastructure for data management and analytics. As MMKGs are inherently incomplete and generally contain noisy data, MMKG Completion (MMKGC) is a central task for improving data quality and semantic inference. Specifically, a crucial aspect of MMKGC is *negative sampling*, which impacts model discriminability and completion accuracy. However, existing negative sampling proposals often ignore the semantic properties of relation types and lack mechanisms for adaptive control of negative sample hardness, leading to suboptimal MMKGC performance. To address issues such as these, we propose RelDINS that improves semantic consistency and robustness by performing relation-type-aware negative sampling through diffusion-based interpolation. RelDINS incorporates two modules: (i) a Relation-type-aware Multimodal Embedding Learning (RMEL) module that adaptively injects relational semantics into entity representations based on cardinality constraints; (ii) and a Diffusion-based Interpolation Negative Sampling (DINS) module that dynamically generates hardness-tunable negative samples via spherical linear interpolation in diffusion noise space. Extensive experiments on three public benchmarks show that RelDINS achieves state-of-the-art performance, with average improvements of 3.5% in MRR, 5.0% in Hit@1, 2.6% in Hit@3, and 1.4% in Hit@10 over leading baselines. Supported by a complexity analysis and an empirical study, RelDINS is a principled and scalable solution to enhancing semantic consistency and data reliability in MMKGC.

PVLDB Reference Format:
Qian Ma, Linfei Dai, Zhongming Yao, Yu Gu, Tianyi Li, Christian S. Jensen, and Ge Yu. Multimodal Knowledge Graph Completion via Relation-Aware Negative Sampling with Diffusion-Based Interpolation. PVLDB, 19(9): 1949 - 1962, 2026.
doi:10.14778/3819518.3819526
PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/dailinfei/RelDINS.

Proceedings of the VLDB Endowment, Vol. 19, No. 9 ISSN 2150-8097.
doi:10.14778/3819518.3819526

1 INTRODUCTION

Knowledge Graphs (KGs) [54] represent real-world facts as triples, each consisting of a *head entity*, a *relation*, and a *tail entity*, denoted as (h, r, t) [58]. KGs serve as the foundation for tasks like semantic search [45], recommendation [14], and question answering [27], enabling machines to reason over knowledge. However, real-world KGs are often incomplete, motivating research on Knowledge Graph Completion (KGC) [10] to infer missing facts. A key task in KGC, link prediction [6], focuses on inferring missing entities.

Recent studies [21, 43, 63] integrate multimodal information (e.g., images and text) with KGs, creating Multimodal Knowledge Graphs (MMKGs). In MMKGC [48], negative sampling (NS) [55] is crucial, as negative samples (synthetic triples) are scarce in multimodal settings. By contrasting negative and positive samples, NS helps models capture semantic boundaries and relational dependencies. Many state-of-the-art MMKGC methods [3, 51, 58] focus on refining NS through visual–textual discrimination. However, most still rely on random [3] or global generative sampling [51], failing to address two key challenges:

Challenge I: Distinct relation types impose heterogeneous semantic constraints. Knowledge graphs have four relation types: (i) one-to-one (1–1), which enforces bidirectional uniqueness (e.g., country–capital); (ii) one-to-many (1–N), where the head entity serves as a semantic hub (e.g., author–works); (iii) many-to-one (N–1), where multiple heads semantically converge toward a single tail (e.g., students–school); (iv) many-to-many (N–N), which involves highly overlapping associations requiring strong representational discriminability. These relation types impose different cardinality constraints, shaping semantic distribution in the embedding space. Ignoring these constraints during negative sampling introduces semantic bias, degrades embedding quality, and may misidentify valid entities as negatives, especially for asymmetric relations like 1–N and N–1. For N–N relations, relation-type-agnostic sampling generates false negatives that resemble positives, causing gradient inconsistency, unstable optimization, and slow convergence.

Example 1. *Fig. 1 shows the* ***director–movie*** *relation, a typical 1–N relation. Christopher Nolan is linked to multiple films, e.g.,* Interstellar, Inception, *and* Tenet. *However, the MMKG may include only a subset of these positive pairs (e.g., Nolan–Inception and Nolan–Tenet), while others (e.g., Nolan–Interstellar) are missing due to data incompleteness. For 1–N and N–1 relations, overlooking the asymmetric semantics may cause misclassifications of positive associations as*

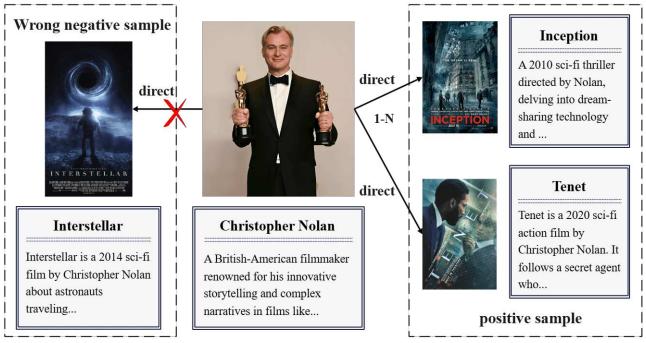


Figure 1: Director-movie relation illustrating 1-N asymmetric semantics and annotation gaps.

negative ones, causing semantic ambiguity and label noise that degrade the discriminative power of embeddings. In the example, the association between Nolan and Interstellar is incorrectly regarded as a negative sample.

Challenge II: Lack of dynamic tunability in negative sample strength and local semantic awareness. Existing negative sampling strategies [31, 51, 58] have two limitations. First, most methods [51, 58] lack a mechanism to dynamically adjust the hardness of negatives. Random sampling produces simple negatives that provide limited supervision, while adversarial methods like DHNS generate harder negatives but lack fine-grained control, hindering progressive optimization and reducing training efficiency.

Second, current methods often ignore local semantic awareness, treating negative sampling as a global operation. They fail to capture the fine-grained contextual semantics of positive triples (e.g., entity co-occurrence and relation type), leading to negatives that are either trivial or semantically irrelevant. As a result, models struggle to learn precise decision boundaries, leading to unstable convergence and limited generalization in MMKGC. Yet, such capabilities are important in real-world applications. For instance, in e-commerce product alignment, distinguishing visually similar products (e.g., shirts of the same style but different brands) requires fine-grained discrimination, which random negative sampling fails to provide. Similarly, in medical diagnosis, where entities (diseases) are associated with symptoms (text) and scans (images), avoiding false positives is critical. RelDINS contributes to improving robustness of such high-stakes decisions by enforcing strict semantic boundaries.

To address the two challenges, we propose **RelDINS**, a novel MMKGC framework that incorporates **Rel**ation-type-aware **N**egative **S**ampling via **D**iffusion-based **I**nterpolation[1]. RelDINS comprises two core modules: the *Relation-type-Aware Multimodal Embedding Learning* (RMEL) module and the *Diffusion-Based Interpolation Negative Sampling* (DINS) module, designed to address the lack of relation-type semantic constraints and the absence of dynamic tunability and local semantic awareness in negative sampling. Specifically, to tackle **Challenge I**, RMEL models the semantic constraints induced by different relation types by introducing a relation-type-adaptive strategy that adjusts multimodal fusion based on the relation cardinality (e.g., 1–N, N–1). This mechanism enables a model to capture asymmetric relational semantics and maintain consistent feature alignment across modalities. To tackle **Challenge II**, DINS introduces a diffusion-based interpolation mechanism that generates hard negatives adaptively through semantic interpolation between positives and randomly sampled negatives. By employing a dynamic interpolation coefficient, the module continuously tunes sample hardness throughout training while remaining sensitive to the local semantic context of each positive triple. This dual control over hardness and local semantic alignment enables RelDINS to generate challenging yet semantically relevant negatives, thereby enhancing discriminative learning, stability, and generalization in MMKGC. To the best of our knowledge, RelDINS is the first proposal to incorporate relation cardinality types into multimodal embedding learning and propose hardness-adaptive negative sampling via diffusion interpolation for MMKGC.

The main contributions are summarized as follows:

- We propose RelDINS, a novel framework for MMKGC that incorporates relation-type-aware negative sampling via diffusion-based interpolation, linking relation-type semantics and adaptive negative sample generation.
- We propose RMEL, a relation-type-aware multimodal embedding module that dynamically integrates multimodal information according to relation cardinality.
- We propose DINS, a diffusion-based interpolation sampler that enables dynamic hardness adjustment and local semantic alignment, generating high-quality negatives near decision boundaries for progressive model training.
- We conduct extensive experiments on three MMKG benchmarks and compare with twenty baselines. Experimental results show that RelDINS achieves state-of-the-art performance, with average improvements across the three benchmarks of 3.5% in MRR, 5.0% in Hit@1, 2.6% in Hit@3, and 1.4% in Hit@10 over the leading baselines.

2 RELATED WORK

Knowledge graph completion (KGC) is a fundamental task in data management that has attracted extensive attention in the database community [17, 28, 32, 37, 56, 59, 61]. Existing KGC models can generally be categorized according to their scoring mechanisms: (i) Translation-based models such as TransE [2] represent relations as translations between entity embeddings, while TransH [47] and TransR [22] further introduce relation-specific hyperplanes or embedding spaces to improve modeling flexibility; (ii) Semantic matching models, including RESCAL [30], DistMult [53], and ComplEx [40], score triples via tensor or matrix factorization, where ComplEx further captures symmetric and antisymmetric relations using complex-valued embeddings; and (iii) Neural and geometric embedding models such as ConvE [7] and RotatE [38], which respectively employ convolutional operations and complex-space rotations for triple scoring. Despite their effectiveness, these methods rely solely on structural triples and ignore multimodal information (e.g., textual and visual cues), limiting their ability to capture richer semantic context in real-world scenarios.

[1]To make the framework name more word-like and easier to pronounce, we reversed the order of "NS" and "DI" in the acronym.

2.1 MMKGC

Recent studies [4, 43, 46, 60, 63] integrate multimodal information for MMKGC. TransAE [46] uses an autoencoder to reconstruct multimodal embeddings, while RSME [43] employs multi-gating to filter visual cues. VBKGC [63] integrates VisualBERT [20] for deep feature fusion, OTKGE [4] applies optimal transport to align representations, and AdaMF [60] uses adaptive gating for modality selection. While these methods enhance representation richness, they focus on generic multimodal fusion and neglect the structural complexity of relations, particularly the role of relation types in shaping semantics. This limits their ability to model complex relational semantics and perform well in real-world multimodal reasoning tasks. Beyond representation learning, large-scale MMKG training also requires efficient graph-centric computation, as explored in distributed GNN training [34, 35] and distributed graph processing [13].

2.2 NS-Enhanced KGC

Some studies [1, 3, 23, 31, 44, 51, 58, 62] employ negative sampling to generate false triples for optimizing embeddings by distinguishing true from false statements. Random negative sampling [23] replaces the head or tail entity at random but often yields easy negatives. To improve learning, earlier methods like KBGAN [3] and IGAN [44] use adversarial frameworks where a generator creates negatives to "fool" the discriminator. NSCaching [62] caches high-scoring negatives for reuse, while SANS [1] samples from neighboring entities to increase sample difficulty. With MMKGs, negative sampling has been extended to incorporate multimodal semantics. MMRNS [51] integrates relational context, and MANS [58] adjusts both structural and visual embeddings. Recently, generative methods like DHNS [31] use diffusion models to synthesize hierarchical hard negatives through a denoising process.

However, existing generative approaches such as DHNS follow a reconstruction-based paradigm that generates negatives around the positive distribution without explicit guidance toward the decision boundary. Consequently, they cannot adaptively control negative hardness during training. In contrast, RelDINS introduces an interpolation-based directional generation mechanism that synthesizes boundary-critical negatives via spherical interpolation between positive and negative entity pairs, enabling controllable and phase-aware hard negative generation.

3 PRELIMINARIES

3.1 MMKG Formulation

An MMKG extends the conventional knowledge graph by associating entities with multimodal au xiliary information such as textual descriptions and visual images. To formally describe this structure, an MMKG is defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{I}, \mathcal{F})$ [58, 60], where $\mathcal{E}$ denotes the set of entities, $\mathcal{R}$ the set of relations, and $\mathcal{F} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ the set of factual triples. $\mathcal{T}$ and $\mathcal{I}$ represent the sets of textual descriptions and images associated with entities, respectively. This formulation follows the standard MMKGC setting [24, 49, 51, 58], which focuses on relational triples with unstructured multimodal features, where literal attributes are implicitly incorporated into textual descriptions $\mathcal{T}$.

Each entity $e \in \mathcal{E}$ is represented by three embeddings: $\boldsymbol{e} = [\boldsymbol{e}_s; \boldsymbol{e}_t; \boldsymbol{e}_v]$, where $\boldsymbol{e}_s$, $\boldsymbol{e}_t$, and $\boldsymbol{e}_v$ are the structural, textual, and visual representations, respectively, with each embedding in $\mathbb{R}^{d_e}$. The structural embedding $\boldsymbol{e}_s$ is learned from the graph topology, while $\boldsymbol{e}_t$ and $\boldsymbol{e}_v$ are extracted from textual and visual modalities (e.g., via BERT [8] and VGG [33]). Similarly, each relation $r \in \mathcal{R}$ is encoded as $\boldsymbol{r} \in \mathbb{R}^{d_r}$, where d_r is the relation embedding dimension.

3.2 MMKGC Task Defnition

The goal of MMKGC is to infer missing facts in $\mathcal{G}$, i.e., to predict whether a candidate triple (h, r, t) is valid.

Given a query with an unknown entity, either $(h, r, ?)$ or $(?, r, t)$, the model estimates the plausibility of candidate triples through a scoring function [51, 58]:

$$\phi(h, r, t) : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \mathbb{R},$$

where a higher $\phi(h, r, t)$ value indicates a higher likelihood that the triple holds true in the knowledge graph. Accordingly, the learning objective is to optimize entity and relation embeddings such that valid triples receive higher scores than invalid ones:

$$\phi(h, r, t) > \phi(h', r, t'),\ \forall (h, r, t) \in \mathcal{F},\ (h', r, t') \notin \mathcal{F}.$$

3.3 Negative Sampling for Contrastive Learning

Since KG contain only factual triples, negative sampling is crucial to provide contrastive supervision during training [31, 51]. For each positive triple $(h, r, t) \in \mathcal{F}$, a set of negative samples is constructed by corrupting either the head or the tail entity: (h', r, t) *or* (h, r, t'), satisfying: $h', t' \in \mathcal{E} \setminus \{h, t\}$, $(h', r, t), (h, r, t') \notin \mathcal{F}$.

The model is then trained to maximize $\phi(h, r, t)$ for positive triples and minimize $\phi(h', r, t')$ for negative ones, thereby learning discriminative embeddings through contrastive optimization.

However, traditional random sampling produces easy negatives that are far from the decision boundary, yielding weak supervision signals and slow convergence. Recent studies show that introducing hard negatives, i.e., samples semantically close to positives in the embedding space, can enhance the model's discriminative power and generalization. In multimodal settings, this challenge becomes more complex, as semantic similarity must be preserved not only structurally but also across textual and visual modalities.

Final Objective: The goal of this study is to develop a **relation-type-aware and hardness-adaptive negative sampling framework** for MMKGC, which enables the generation of **semantically consistent** and **controllably hard** negative samples, thereby ensuring stable optimization, enhanced discriminative learning, and improved completion accuracy.

4 FRAMEWORK

4.1 Framework Overview

We first introduce an overview of RelDINS, as illustrated in Fig. 2. It consists of a relation-type-aware multimodal embedding learning (RMEL) module and a diffusion-based interpolation negative sampling (DINS) module.

- **RMEL Module:** learns discriminative entity embeddings by fusing multimodal information (structural, textual, and image) based on relation cardinality, ensuring that embeddings remain

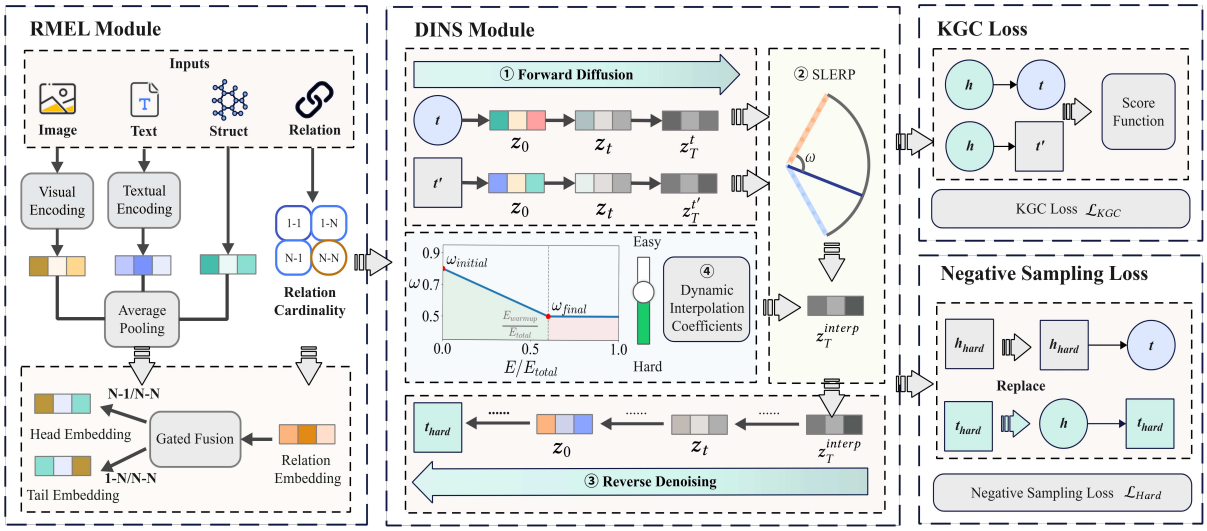


Figure 2: Overall framework of RelDINS.

consistent with the semantic patterns of different relation types (e.g., 1–N, N–1, N–N).

- **DINS Module:** introduces a hardness-adaptive diffusion interpolation mechanism to generate negatives with controllable difficulty and high semantic relevance. By performing spherical linear interpolation (SLERP) between positives and randomly sampled negatives, DINS creates a continuum of negatives with adjustable hardness through a dynamic interpolation coefficient.

4.2 RMEL Module

4.2.1 Modality-Specific Encoders. We adopt a standardized multimodal encoding framework using pre-trained models to extract modality-specific features and a unified projection network to align cross-modal dimensions and semantics.

Given an entity e with associated text description set $\mathcal{T}(e)$ and image set $\mathcal{I}(e)$, the modality-specific encoding is defined as follows:

$$\boldsymbol{e}_t^{raw} = \frac{1}{|\mathcal{T}(e)|} \sum_{text \in \mathcal{T}(e)} \text{BERT}(text) \tag{1}$$

$$\boldsymbol{e}_v^{raw} = \frac{1}{|\mathcal{I}(e)|} \sum_{img \in \mathcal{I}(e)} \text{VGG-16}(img), \tag{2}$$

where BERT [8] and VGG-16 [33] serve as base encoders to extract high-level semantic and image features, respectively. For entities with multiple textual descriptions or images, an average pooling strategy [60] is applied to aggregate instance-level features, ensuring balanced modality information. The resulting representations $\boldsymbol{e}_t^{raw}$ and $\boldsymbol{e}_v^{raw}$ denote the raw text and image embeddings.

Note that although advanced joint vision–language models such as CLIP provide superior cross-modal alignment, we adopt standard BERT and VGG-16 encoders to ensure a fair comparison with existing MMKGC baselines [31, 51, 60, 63]. Exploring stronger encoders like CLIP is left as a promising direction for future work.

Furthermore, to unify the representation space across modalities and enhance semantic consistency, we introduce a two-layer projection network, denoted as $P_m(\cdot)$, where $m \in t, v$ corresponds to the text or image modality. This network maps modality-specific features into a shared semantic space of target dimension d_e. The resulting semantically aligned representations are $\boldsymbol{e}_t = P_t(\boldsymbol{e}_t^{raw}) \in \mathbb{R}^{d_e}$ for text and $\boldsymbol{e}_v = P_v(\boldsymbol{e}_v^{raw}) \in \mathbb{R}^{d_e}$ for image.

Moreover, to establish the foundational structural representations of the KG, the entity structure embedding $\boldsymbol{e}_s \in \mathbb{R}^{d_e}$ and relation embedding $\boldsymbol{r} \in \mathbb{R}^{d_r}$ are initialized via random learnable matrices, serving as core base representations. The model then outputs standardized multimodal inputs $\{\boldsymbol{e}_s, \boldsymbol{e}_t, \boldsymbol{e}_v, \boldsymbol{r}\}$, providing a unified, semantically consistent foundation for relation-type-aware multimodal embedding learning. Unlike the projected multimodal features, the structure embedding $\boldsymbol{e}_s$ and relation embedding $\boldsymbol{r}$ are initialized directly as learnable parameters within the target dimension space. They are optimized end-to-end to align naturally with other modalities during training, requiring no additional projections.

4.2.2 Relation-type-Aware Multimodal Embedding. As shown in the left panel of Fig. 2, using the standardized multimodal representations described above, we further enhance entity embeddings by explicitly modeling the influence of different relation types.

Specifically, we first perform relation-type statistics on the training set to quantify the average number of entities associated with each relation. For each relation r, we define an *average entity cardinality function* $\alpha(r, type)$, where $type \in \{h, t\}$ denotes the head or tail entity role. Formally:

$$\alpha(r, h) = \frac{1}{|H_r|} \sum_{h \in H_r} |\{t \mid (h, r, t) \in \mathcal{F}\}|, \tag{3}$$

$$\alpha(r, t) = \frac{1}{|T_r|} \sum_{t \in T_r} |\{h \mid (h, r, t) \in \mathcal{F}\}|, \tag{4}$$

where H_r and T_r represent the head and tail entity sets associated with relation r, respectively. Based on these statistics, relations are categorized into four types using a threshold parameter θ. Theoretically, the ideal average cardinality for a one-to-one (1-1) relation is 1.0, but real-world datasets like Wikidata and YAGO often contain

noise or duplicate records. Therefore, we set $\theta = 1.5$, which is the arithmetic midpoint between the singular case (cardinality 1) and the minimal plural case (cardinality 2). This establishes a buffer that tolerates minor data noise while effectively distinguishing different relation cardinality types. Experimental analyses (see Section 5.4) validates that this setting is reasonable.

- $\alpha_h(r) \leq \theta$ and $\alpha_t(r) \leq \theta$: one-to-one (1-1)
- $\alpha_h(r) \leq \theta$ and $\alpha_t(r) > \theta$: one-to-many (1–N)
- $\alpha_h(r) > \theta$ and $\alpha_t(r) \leq \theta$: many-to-one (N–1)
- otherwise: many-to-many (N–N)

This relation-type profiling serves as the foundation for subsequent adaptive multimodal embedding fusion.

During the embedding learning phase, we first perform a basic integration of the entity's multimodal embeddings through average pooling to obtain an initial unified representation: $\tilde{\boldsymbol{e}} = \frac{\boldsymbol{e}_s + \boldsymbol{e}_t + \boldsymbol{e}_v}{3}$, where $\tilde{\boldsymbol{e}} \in \mathbb{R}^{d_e}$ serves as the entity's joint multimodal representation. While advanced mechanisms such as cross-attention [51] are popular in multimodal learning, our experiments suggest they tend to overfit spurious correlations in MMKGs due to data sparsity and semantic gaps. In contrast, average pooling serves as a parameter-free regularizer, effectively reducing noise and promoting robust semantic alignment across modalities, thus providing a stable foundation for relational modeling.

Subsequently, we dynamically incorporate relation-specific information according to the relation type. Since the dimensionalities of entity and relation embeddings may differ, the relation embedding $\boldsymbol{r} \in \mathbb{R}^{d_r}$ is first projected into the entity embedding space via a linear transformation:

$$\boldsymbol{r}_p = \boldsymbol{W}_r \boldsymbol{r} + \boldsymbol{b}_r, \tag{5}$$

where $\boldsymbol{W}_r \in \mathbb{R}^{d_e \times d_r}$ and $\boldsymbol{b}_r \in \mathbb{R}^{d_e}$ are learnable parameters of the projection layer.

To control the injection strength of relational information, a gating mechanism is employed. Given a triple (h, r, t), the final multimodal embeddings of the head and tail entities, denoted by $\boldsymbol{h}$ and $\boldsymbol{t}$, are computed according to the relation type as follows:
(i) 1–1 Relations: No relational information is injected, preserving the base embeddings:

$$\boldsymbol{h} = \tilde{\boldsymbol{h}}, \quad \boldsymbol{t} = \tilde{\boldsymbol{t}} \tag{6}$$

(ii) 1–N Relations: The relational information is fused into the tail entity, since the tail entity depends more on relation semantics:

$$\boldsymbol{h} = \tilde{\boldsymbol{h}}, \quad \boldsymbol{t} = \tilde{\boldsymbol{t}} + g(\tilde{\boldsymbol{t}}, \boldsymbol{r}_p) \odot \boldsymbol{r}_p \tag{7}$$

(iii) N–1 Relations: The relational information is fused into the head entity, since the head entity depends more on relation semantics:

$$\boldsymbol{h} = \tilde{\boldsymbol{h}} + g(\tilde{\boldsymbol{h}}, \boldsymbol{r}_p) \odot \boldsymbol{r}_p, \quad \boldsymbol{t} = \tilde{\boldsymbol{t}} \tag{8}$$

(iv) N–N Relations: The relational information is simultaneously fused into both embeddings, since both entities are semantically influenced by the relation:

$$\boldsymbol{h} = \tilde{\boldsymbol{h}} + g(\tilde{\boldsymbol{h}}, \boldsymbol{r}_p) \odot \boldsymbol{r}_p, \quad \boldsymbol{t} = \tilde{\boldsymbol{t}} + g(\tilde{\boldsymbol{t}}, \boldsymbol{r}_p) \odot \boldsymbol{r}_p \tag{9}$$

Here, $g(\cdot, \cdot)$ denotes the gating function, defined as:

$$g(\tilde{\boldsymbol{e}}, \boldsymbol{r}_p) = \sigma(\boldsymbol{W}_g[\tilde{\boldsymbol{e}}; \boldsymbol{r}_p] + \boldsymbol{b}_g), \tag{10}$$

Table 1: Intra-class Similarity Analysis of 1-N Relations on MKG-W

ID	Relation	Avg. Sim. (w/o RMEL)	Avg. Sim. (w/ RMEL)	Imp. (↑)
1	genre	0.334	0.621	0.287
2	occupation	0.392	0.685	0.293
3	performer	0.315	0.642	0.327
4	narrative location	0.276	0.586	0.310
5	producer	0.388	0.684	0.296
6	instrument	0.289	0.635	0.346
7	record label	0.341	0.685	0.344
8	member of sports team	0.435	0.711	0.276
9	contains administrative territorial entity	0.412	0.687	0.275
10	educated at	0.357	0.664	0.307
Avg.	/	0.354	0.660	0.306

where $[\cdot;\cdot]$ represents vector concateenation, $\boldsymbol{W}_g \in \mathbb{R}^{d_e \times 2d_e}$ and $\boldsymbol{b}_g \in \mathbb{R}^{d_e}$ are the parameters of the gating layer, $\sigma(\cdot)$ is the *sigmoid* activation function, and $\odot$ denotes element-wise multiplication. The gating output yields a vector with values between 0 and 1, adaptively weighting relational information during fusion.

Notably, the relation information is fused on the multiple ("N") side based on the Semantic Hub hypothesis. This hypothesis suggests that in asymmetric relations, the unique-side entity (e.g., director in a 1-N relation) serves as a stable semantic anchor, while entities on the multiple side (e.g., movies) are semantically diverse. By injecting relational information into the multiple side, we bring the diverse entities closer to the semantic hub, enhancing their discriminative power. To intuitively illustrate the effectiveness of this fusion strategy, we conduct an intra-class similarity analysis using MKG-W as an example. We randomly select 10 representative 1-N relations and compute the average pairwise cosine similarity among tail entities associated with the same head. As shown in Table 1, the intra-class similarity improves consistently in all cases. For instance, in the case of *instrument* (Case 6), visually distinct entities such as "guitar" and "bass" initially exhibit low similarity (0.289), but are aligned into a more coherent cluster (0.635) after RMEL processing. On average, the similarity increases from 0.354 to 0.660, showing that RMEL is able to effectively enforce relation-induced topological constraints.

With relation-type-aware embedding learning, entities involved in the same relation form more compact clusters in the embedding space, thereby preserving cardinality-specific semantic patterns and enhancing link prediction accuracy.

4.3 DINS Module

Given a positive triple (h, r, t), we take the process of generating a negative sample (h, r, t') by corrupting the tail entity as an example. Unless otherwise specified, all negative samples discussed below refer to (h, r, t'). The procedure for generating (h', r, t) is analogous. As shown in the middle panel in Fig. 2, the process integrates forward noising and reverse denoising stages.

4.3.1 Forward Noising and Interpolation. The forward process maps entity embeddings into a noise space and constructs transitional representations via interpolation, serving as inputs for the subsequent reverse denoising phase. It comprises two main steps: (1)

diffusion-based forward noising and (2) spherical linear interpolation.

Forward Noising. To adapt one-dimensional entity embeddings for diffusion models, we first preprocess the multimodal embedding $\boldsymbol{e}_m \in \mathbb{R}^{d_e}$ (from the RMEL module) by expanding its dimension to s^2 (where $s = \lceil \sqrt{d_e} \rceil$) and reshaping it into a $s \times s$ matrix. The reshaped embedding is then normalized to the range $[-1, 1]$ using min–max normalization to match the input distribution expected by the diffusion model.

Subsequently, Gaussian noise is progressively added following the Denoising Diffusion Probabilistic Model (DDPM) [16]. At time step t, the forward process is defined as follows:

$$q(\boldsymbol{z}_t \mid \boldsymbol{z}_0) = \mathcal{N}\left(\boldsymbol{z}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{z}_0, (1-\bar{\alpha}_t)\boldsymbol{I}\right), \tag{11}$$

where $\boldsymbol{z}_0$ denotes the initial embedding $\boldsymbol{e}_m^{2D}$, and $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$ represents the noise scheduling parameter with β_i being the variance schedule. In practice, by using the reparameterization trick [16], we can obtain:

$$\boldsymbol{z}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}) \tag{12}$$

Finally, the fully noised embedding $\boldsymbol{z}_T$ is obtained after iterating the diffusion process up to the maximum time step T. This gradual noising procedure progressively removes fine-grained semantic details while preserving the overall embedding distribution, establishing a well-structured noise space for subsequent interpolation.

Spherical Linear Interpolation. After obtaining $\boldsymbol{z}_T$, we aim to generate transitional samples via interpolation in the noise space. It is worth noting that interpolation needs to operate on a pair of samples (F_s, F_e). Given a positive triple (h, r, t), we treat it as the positive parent sample $F_s \in \mathcal{F}$. The corresponding negative parent sample F_e is constructed by replacing the tail entity t with a randomly selected entity $t' \in \mathcal{E} \setminus t$ such that $(h, r, t') \notin \mathcal{F}$. In other words, $F_e = (h, r, t')$ serves as a random negative counterpart of F_s.

Let the original multimodal embeddings of the parent tails t and t' be $\boldsymbol{t}_0$ and $\boldsymbol{t}'_0$ (i.e., $\boldsymbol{t}$, and $\boldsymbol{t'}$), respectively. Their fully noisy representations at step T based on Eq. 12 are:

$$\boldsymbol{z}_T^t = \sqrt{\bar{\alpha}_T}\boldsymbol{t}_0 + \sqrt{1-\bar{\alpha}_T}\boldsymbol{\epsilon}, \quad \boldsymbol{z}_T^{t'} = \sqrt{\bar{\alpha}_T}\boldsymbol{t}'_0 + \sqrt{1-\bar{\alpha}_T}\boldsymbol{\epsilon} \tag{13}$$

We then perform spherical interpolation as follows:

$$\boldsymbol{z}_T^{interp} = \frac{\sin((1-\omega)\psi)}{\sin(\psi)}\,\boldsymbol{z}_T^t + \frac{\sin(\omega\psi)}{\sin(\psi)}\,\boldsymbol{z}_T^{t'}, \tag{14}$$

where $\omega \in [0, 1]$ is the interpolation coefficient (its adaptive control is detailed in Section 4.3.2), and $\psi = \arccos\left(\frac{\boldsymbol{z}_T^t \cdot \boldsymbol{z}_T^{t'}}{\|\boldsymbol{z}_T^t\| \, \|\boldsymbol{z}_T^{t'}\|}\right)$ denotes the angular distance between the two vectors. To ensure numerical stability, when the dot product approaches 1, Slerp degenerates to linear interpolation. The resulting $\boldsymbol{z}_T^{interp}$ serves as the noise-space transitional representation, which will subsequently be reconstructed in the reverse denoising phase.

4.3.2 Reverse Denoising. The reconstruction process, labeled "Reverse Denoising" in Fig. 2, constitutes the core of the DINS module. Its goal is to reconstruct high-quality negative sample embeddings from the noisy latent representation $\boldsymbol{z}_T^{interp}$, which is generated by RMEL through the reverse diffusion process. This process consists of two main components: (i) the denoising paradigm, and (ii) the dynamic interpolation coefficient adjustment strategy, which together enable hardness-tunable negative sample generation.

Denoising Paradigm. The reverse denoising paradigm progressively refines a noisy latent representation $\boldsymbol{z}_t$ (i.e., $\boldsymbol{z}_T^{interp}$, the interpolated transitional sample obtained during forward diffusion) into a semantically coherent negative sample embedding.

At each reverse step $t = T, \ldots, 1$, a denoising network ϵ_θ predicts the injected noise and recursively restores the original embedding. The reverse denoising process is formulated as:

$$\boldsymbol{z}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(\boldsymbol{z}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(\boldsymbol{z}_t, t\right)\right) + \sigma_t \boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}), \tag{15}$$

where σ_t denotes the noise scale parameter. This iterative process gradually corrects $\boldsymbol{z}_t$ based on the predicted noise $\epsilon_\theta(\boldsymbol{z}_t, t)$, ultimately yielding the denoised embedding $\boldsymbol{z}_0$.

To optimize ϵ_θ, we employ the standard mean squared error (MSE) loss of the diffusion model, which minimizes the discrepancy between the predicted and actual noise:

$$\mathcal{L}_{DM} = \mathbb{E}_{t, \boldsymbol{z}_0, \boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \epsilon_\theta\left(\boldsymbol{z}_t, t\right)\|^2\right], \tag{16}$$

where t is uniformly sampled from $\{1, \ldots, T\}$, $\boldsymbol{z}_0$ is the preprocessed initial embedding, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ is the random noise added during the forward process, and $\boldsymbol{z}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ is the noisy embedding at time step t. By minimizing this objective, the denoising network learns to capture the statistical structure of the embedding space, enabling the generation of semantically meaningful hard negatives.

Notably, during training, $\mathcal{L}_{DM}$ is optimized alongside the overall negative sampling loss, but only updates the denoising network parameters, leaving the main embedding model unchanged.

Dynamic Interpolation Coefficient Adjustment. To enhance training stability and discriminative ability, we integrate curriculum learning to adaptively adjust negative sample hardness via a dynamic interpolation coefficient ω. Initially set to a high value to generate easy, random-like negatives, ω gradually decreases during training, producing harder samples that approach positive instances and aligning the sampling difficulty with the training progression. We compute ω as follows:

$$\omega = \omega_{final} + \left(\omega_{initial} - \omega_{final}\right) \cdot \max\left(0, 1 - \frac{E}{E_{warmup}}\right), \tag{17}$$

where $\omega_{initial}$ and ω_{final} are the initial and terminal interpolation coefficients, respectively. E denotes the current training epoch, and E_{warmup} specifies the warm-up duration.

The overall pipeline of DINS. For each positive triple (h, r, t), the generation of negatives follows a structured three-step workflow, as illustrated by labels ①–③ in Fig 2: ① **Parallel Forward Diffusion**: The positive tail entity and a randomly sampled negative entity are mapped into the noise space to obtain their noised representations $\boldsymbol{z}_T^t$ and $\boldsymbol{z}_T^{t'}$. ② **SLERP Interpolation**: Spherical Linear Interpolation is performed between the two noise vectors to construct a semantic trajectory. The interpolation position is determined by ω, which acts as a "hardness slider" to control the difficulty of the generated samples. ③ **Reverse Denoising**: The transitional embedding $\boldsymbol{z}_T^{interp}$ is processed in the reverse diffusion

phase to reconstruct a coherent, hard negative sample t_{hard}. Notably, during training, ω is adjusted dynamically to decrease as epochs increase (as shown at label ④ in Fig. 2), ensuring that the model is exposed gradually to increasingly complex patterns to refine the decision boundary. Finally, the entire module is optimized end-to-end via the overall loss function. Additionally, this process performs semantic synthesis rather than simple noise injection. By generating representations that lie geometrically between a positive anchor and a randomly sampled negative entity, DINS simulates real-world hard negatives that share plausible semantic characteristics with the target. This mechanism forces the model to learn fine-grained discriminative features, thereby sharpening the decision boundary between positive and negative instances.

As a concrete illustration, consider the triple *(Christopher Nolan, director, Inception)*. While random sampling often yields semantically distant negatives like *Apple* (which are trivial to distinguish), RelDINS generates synthetic negatives via interpolation that may resemble other science-fiction movies by different directors. These samples share high-level semantic context with the target while remaining factually incorrect, forcing the model to learn fine-grained distinctions and tightening the decision boundary.

4.4 Training Objective and Loss Function

For MMKGC task, our training objective is to maximize the scores of positive triples while minimizing the scores of negative triples. The score reflects the plausibility of a triple (h, r, t) and is computed by a score function $\phi(h, r, t)$. To this end, we optimize a joint loss consisting of three parts: a standard KGC loss with self-adversarial negative sampling, a hard negative sampling loss, and a regularization term.

Knowledge graph completion loss. To enable the model to distinguish positive triples from negatives, we use a margin-based logistic loss with self-adversarial weighting. For each positive triple (h, r, t), we sample k negative triples by corrupting either the head or the tail, and optimize:

$$\mathcal{L}_{KGC} = -\log\sigma(\gamma - \phi(h,r,t)) - \sum_{i=1}^{k} w_i \log\sigma\left(\phi\left(h,r,t_i'\right) - \gamma\right), \quad (18)$$

where $\sigma(\cdot)$ is the sigmoid activation function, γ is a margin hyperparameter controlling the separation between positives and negatives, and t_i' is the i-th randomly sampled negative tail (the head can be corrupted analogously). Following the self-adversarial negative sampling [38], each negative is assigned a weight:

$$w_i = \frac{\exp\left(\alpha\phi\left(h,r,t_i'\right)\right)}{\sum_{j=1}^{k}\exp\left(\alpha\phi\left(h,r,t_j'\right)\right)}, \quad (19)$$

where α is a temperature parameter. This weighting scheme upweights negatives that the model currently finds more plausible ("harder" negatives), which accelerates learning.

Unless stated otherwise, we adopt the RotatE [38] scoring function, which embeds entities and relations in the complex space and models diverse relation patterns via element-wise rotation. But note that our framework is agnostic to the specific scoring function and can be instantiated with alternative choices such as TransE [2] or DistMult [53].

Algorithm 1: Training Procedure of RelDINS

Input: MMKG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{F})$, Modality features $\mathcal{T}, \mathcal{I}$, Hyperparameters T (diffusion steps), E_{warmup}, $\omega_{initial}$, ω_{final}.
Output: Learned embeddings $\mathbf{E}, \mathbf{R}$, Denoising network ϵ_θ.

Initialization: Randomly initialize entity structure embedding matrix $\mathbf{E}$ (containing $\mathbf{e}_s$ for all $e \in \mathcal{E}$) and relation embedding matrix $\mathbf{R}$ (containing $\mathbf{r}$ for all $r \in \mathcal{R}$).
while *not converged* **do**
  for *each batch* $\mathcal{B} = \{(h,r,t)\} \in \mathcal{F}$ **do**
    /*Update interpolation coefficient ω based on current epoch using Eqs. (17)*/
    $\omega \leftarrow \omega_{final} + (\omega_{initial} - \omega_{final}) \cdot \max\left(0, 1 - \frac{E}{E_{warmup}}\right)$
    /*RMEL: Relation-aware Embedding Learning*/
    for *each entity* $e \in \{h, t\}$ *in* $\mathcal{B}$ **do**
      /*Extract raw features $\mathbf{e}_t^{raw}, \mathbf{e}_v^{raw}$ via Eqs. (1–2)*/
      $\boldsymbol{e}_t^{raw} \leftarrow \frac{1}{|\mathcal{T}(e)|}\sum_{text\in\mathcal{T}(e)} \mathrm{BERT}(text)$
      $\boldsymbol{e}_v^{raw} \leftarrow \frac{1}{|\mathcal{I}(e)|}\sum_{img\in\mathcal{I}(e)} \mathrm{VGG\text{-}16}(img)$
      $\tilde{\mathbf{e}} \leftarrow \mathrm{AvgPool}(\mathbf{e}_s, P_t(\mathbf{e}_t^{raw}), P_v(\mathbf{e}_v^{raw}))$ /*Project and Fuse*/
      if *relation type requires fusion (e.g., 1-N for tail)* **then**
        $\mathbf{e} \leftarrow \tilde{\mathbf{e}} + g(\tilde{\mathbf{e}}, \mathbf{r}_p) \odot \mathbf{r}_p$ (Eqs. 6–9) /*Apply Gating*/
      else
        $\mathbf{e} \leftarrow \tilde{\mathbf{e}}$.
    /*DINS: Diffusion-Based Negative Sampling*/
    Sample random negatives (h, r, t') for each positive (h, r, t)
    /*Add noise to obtain $z_T(\mathbf{t})$ and $z_T(\mathbf{t}')$ via Eqs. (12–13)*/
    $\boldsymbol{z}_t \leftarrow \sqrt{\bar{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
    $\boldsymbol{z}_T^t \leftarrow \sqrt{\bar{\alpha}_T}\boldsymbol{t}_0 + \sqrt{1-\bar{\alpha}_T}\boldsymbol{\epsilon}, \quad \boldsymbol{z}_T^{t'} \leftarrow \sqrt{\bar{\alpha}_T}\boldsymbol{t}_0' + \sqrt{1-\bar{\alpha}_T}\boldsymbol{\epsilon}$
    $\boldsymbol{z}_T^{interp} \leftarrow \mathrm{Slerp}(z_T(\mathbf{t}), z_T(\mathbf{t}'), \omega)$ /*Interpolation*/
    /*Reverse Denoising (Generation)*/
    $\boldsymbol{z}_{t-1} \leftarrow \frac{1}{\sqrt{1-\beta_t}}\left(\boldsymbol{z}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{z}_t, t)\right) + \sigma_t \boldsymbol{z}, \ \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$
    /*Optimization (Dual Optimizers)*/
    $\min \mathcal{L}_{DM} \leftarrow \mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(z_t, t)\|^2]$ /*Update ϵ_θ*/
    /*KGC Loss*/
    $\mathcal{L}_{KGC} \leftarrow -\log\sigma(\gamma - \phi(h,r,t)) - \sum_{i=1}^{k} w_i \log\sigma\left(\phi\left(h,r,t_i'\right) - \gamma\right)$
    /*Hard Negative Sampling Loss*/
    $\mathcal{L}_{Hard} \leftarrow -\log\sigma(\gamma - \phi(h_{hard}, r, t)) - \log\sigma(\gamma - \phi(h, r, t_{hard}))$
    $\min \mathcal{L} \leftarrow \mathcal{L}_{KGC} + \beta\mathcal{L}_{Hard} + \lambda\mathcal{R}$ /*Total Loss*/
    Update $\mathbf{E}, \mathbf{R}$, and RMEL parameters by minimizing $\mathcal{L}$
  return $\mathbf{E}, \mathbf{R}$, *Denoising network* ϵ_θ

Hard negative sampling loss. To tighten the decision boundary, we introduce a hard negative loss constructed from high-quality adversarial negatives synthesized by the DINS module. For each positive triple (h, r, t), DINS generates challenging corrupted heads h_{hard} and tails t_{hard} via a diffusion-based interpolation process (Section 4.3). We penalize the hard negatives as follows:

$$\mathcal{L}_{Hard} = -\log\sigma(\gamma - \phi(h_{hard}, r, t)) - \log\sigma(\gamma - \phi(h, r, t_{hard})) \quad (20)$$

Overall objective. The final training objective combines the standard completion loss $\mathcal{L}_{KGC}$, the hard negative sampling loss $\mathcal{L}_{Hard}$, and

a regularization term $\mathcal{R}$ used for preventing overfitting:

$$\mathcal{L} = \mathcal{L}_{KGC} + \beta \mathcal{L}_{Hard} + \lambda \mathcal{R}, \tag{21}$$

where β balances the impact of hard negatives, and λ is the regularization coefficient.

We summarize the overall training procedure of RelDINS in Algorithm 1. The time complexity of RelDINS is $O(B \cdot [d^2 + T \cdot C_{\text{net}} + k \cdot d])$, where B is the batch size, d the embedding dimension, and C_{net} the denoising network cost, consisting of three main components: (i) RMEL encoding, which involves feature projection and gating mechanisms with a complexity of $O(B \cdot d^2)$, (ii) DINS hard negative generation, the primary cost, requiring T iterations of the denoising network with a complexity of $O(B \cdot T \cdot C_{\text{net}})$, and (iii) scoring, which involves calculating scores for k negatives, with a complexity of $O(B \cdot k \cdot d)$. The space complexity of $O((|\mathcal{E}| + |\mathcal{R}|) \cdot d)$ is dominated by the entity and relation embedding tables, while the additional parameters introduced by the RMEL and DINS modules remain constant with respect to the graph size.

In summary, as illustrated in Fig. 2, the synergy between RMEL and DINS provides a principled and scalable solution for enhancing semantic consistency and completion accuracy in MMKGC.

5 EXPERIMENTAL STUDY

We evaluate RelDINS on three real-world datasets, aiming to answer the following research questions:

- **Q1:** How does RelDINS perform on standard MMKGC benchmarks compared to state-of-the-art methods?
- **Q2:** How well does RelDINS generalize across scoring functions?
- **Q3:** How does RelDINS perform when training with different sample sizes?
- **Q4:** How sensitive is RelDINS to key hyperparameters?
- **Q5:** How do the individual modules contribute to the overall performance of RelDINS?
- **Q6:** Do the visualization results confirm that RelDINS-generated negatives lie near the decision boundary?

5.1 Experimental Settings

5.1.1 Datasets. We evaluate RelDINS on three MMKG benchmark datasets: **MKG-W**, **MKG-Y**, and **DB15K** [25, 51].

- **MKG-W** is derived from Wikidata [42], covering general-domain knowledge like persons, locations, and events, with a balanced distribution of image and textual descriptions.
- **MKG-Y** is based on the YAGO knowledge base [36], focusing on semantic relationships and hierarchical structures, with rich multimodal data.
- **DB15K** extends FB15K-237 [25] to the multimodal setting, emphasizing complex relational patterns (e.g., symmetric and antisymmetric relations) and integrating visual and textual data.

Table 2: Dataset Statistics

Dataset	Entity	Relation	Image	Text	1-1	1-*N*	*N*-1	*N*-*N*
DB15K	12,842	279	12,818	9,904	131	83	24	41
MKG-W	15,000	169	14,463	14,123	80	79	6	4
MKG-Y	15,000	28	12,444	12,305	13	7	5	3

These datasets span different domains, providing a comprehensive test of RelDINS's generalization and robustness. Each follows an 8:1:1 train-validation-test split, ensuring a fair comparison. Table 2 shows dataset statistics, including entity/relation counts and multimodal data volume, along with the distribution of triples across relation types. Notably, non−1-to-1 relations account for over half of the total, highlighting the complex relational structure.

5.1.2 Baselines. The baselines are grouped into three categories:

- **Unimodal (Traditional) KGC Methods:** These approaches rely on the structural information of knowledge graphs, without incorporating multimodal features, and emphasize symbolic reasoning. The methods include *TransE* [2], *TransD* [18], *DistMult* [53], *ComplEx* [40], *RotatE* [38], *PairRE* [5], and *GC-OTE* [39].
- **Multimodal KGC Methods:** These approaches explicitly integrate multimodal information (e.g., visual or textual features) to enhance entity and relation embeddings. The methods are *IKRL* [50], *TBKGC* [29], *TransAE* [46], *MMKRL* [26], *RSME* [43], *VBKGC* [63], *OTKGE* [4], and *AdaMF* [60].
- **Negative Sampling (NS)–based Methods:** These approaches focus on designing advanced negative sampling strategies to improve training efficiency and embedding quality, serving as direct baselines for **RelDINS**. The methods include *KBGAN* [3], *MANS* [58], *MMRNS* [51], *DHNS* [31], and *MIRANS* [57].

5.1.3 Evaluation Metrics. We focus on the core task of MMKGC-*link prediction* [6]. Given a query triple $(h, r, ?)$ or $(?, r, t)$, the goal is to predict the missing entity. Following prior studies [9, 38], we adopt the following ranking-based evaluation metrics:

- **MRR (Mean Reciprocal Rank):** The average of the reciprocal ranks.
- **Hit@k ($k = 1, 3, 10$):** The proportion of cases where the correct entity appears in the top-k predictions, measuring the model's accuracy at different recall rates.

5.1.4 Implementation Details. RelDINS is implemented based on the OpenKE [15] and executed on Ubuntu 24.04.1. All experiments use two NVIDIA RTX 4070 GPUs (16GB VRAM). Empirical observations show that the model trains stably with a batch size of 512 on MKG-W/Y and 256 on DB15K, with a peak memory usage at around 14–16GB, confirming its compatibility with mainstream hardware. Training utilizes the *Adam* optimizer [19], and all parameters in the sampling network are initialized using Xavier normalization [11]. Hyperparameters are tuned on the validation set via grid search, and they have two main categories:

- **Standard MMKGC hyperparameters** are typically employed to regulate the optimization dynamics and determine the embedding capacity of the model. Following the standard settings adopted in prior studies [31, 38], the corresponding search ranges are defined as follows: embedding dimensionality of entities $d_e \in \{150, 175, 200, 225, 250, 275\}$; learning rate $lr \in \{1e{-}3, 5e{-}3, 1e{-}4, 5e{-}4, 1e{-}5\}$; margin parameter $\gamma \in \{2, 4, 6, 8, 10, 12\}$; temperature parameter $\alpha \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$; and relation cardinality threshold $\theta \in \{1.1, 1.3, 1.5, 1.7, 1.9\}$.
- **Negative sampling–specific hyperparameters** are introduced to control the dynamics of the proposed negative sampling process. Specifically, the ratio of warm-up epochs to total training epochs (E_{warmup}/E_{total}) is tuned within $\{0.3, 0.4, 0.5, 0.6, 0.7\}$,

Table 3: Performance Comparison between RelDINS and Representative Baselines

Category	Model	DB15K				MKG-W				MKG-Y			
		MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
Unimodal KGC Models	TransE	24.86	12.78	31.48	47.07	29.19	21.06	33.20	44.23	30.73	23.45	35.18	43.37
	TransD	21.52	8.34	29.93	44.24	26.84	19.65	31.48	42.68	26.39	17.01	33.05	40.41
	DistMult	23.03	14.78	26.98	40.59	20.95	15.89	22.88	36.80	25.04	19.32	27.80	39.95
	ComplEx	27.48	18.13	31.57	45.37	28.09	21.45	30.89	44.77	28.94	23.11	31.07	43.48
	RotatE	29.28	17.87	36.12	49.66	30.79	21.98	36.42	46.73	29.96	20.70	36.38	46.49
	PairRE	31.13	21.67	36.91	51.98	33.29	25.00	38.67	46.71	32.18	25.24	37.58	44.98
	GC-OTE	31.85	22.11	36.52	51.18	33.92	26.55	35.96	46.05	32.95	26.77	36.44	44.01
MMKGC Models	IKRL	26.82	14.09	34.93	49.09	32.36	26.11	34.75	44.07	33.22	30.37	34.28	38.26
	TBKGC	28.08	15.61	37.03	49.86	31.80	25.31	33.91	44.58	33.39	30.47	33.74	37.92
	TransAE	28.09	21.25	37.17	49.17	30.01	21.23	34.91	44.72	28.10	25.31	29.19	33.03
	MMKRL	26.81	13.85	35.01	49.39	29.42	22.54	31.49	43.44	30.94	27.00	32.53	36.07
	RSME	29.76	24.15	32.12	49.23	29.23	23.31	31.09	40.83	34.44	31.78	36.07	39.79
	VBKGC	30.61	19.75	37.18	49.41	30.69	24.55	33.07	44.62	34.03	31.76	35.73	37.72
	OTKGE	23.86	18.45	25.89	34.23	34.36	28.85	36.25	44.88	35.51	31.97	37.18	41.38
	AdaMF	32.51	21.31	39.67	51.68	34.27	27.21	37.86	47.21	38.06	33.49	40.40	45.48
NS-based Models	KBGAN	25.73	9.91	36.95	51.93	29.47	22.21	34.87	40.64	29.71	22.81	34.88	40.21
	MANS	28.82	16.87	38.54	51.51	32.04	27.48	37.48	41.62	29.93	25.25	31.35	34.49
	MMRNS	29.67	17.89	36.86	51.01	34.47	28.93	38.63	47.48	33.32	30.50	35.37	45.47
	DHNS	34.36	23.34	41.56	**53.72**	35.68	28.98	38.68	48.12	39.11	34.70	41.23	46.66
	MIRANS	35.17	24.76	41.24	52.83	35.76	29.33	38.97	48.78	38.45	34.58	40.95	46.03
Ours	RelDINS	**36.61**	**27.14**	**42.23**	53.70	**37.71**	**30.51**	**41.20**	**50.57**	**39.45**	**35.13**	**41.36**	**46.88**
	Improve	**4.09%**	**9.61%**	**1.61%**	/	**5.45%**	**4.02%**	**5.72%**	**3.67%**	**0.87%**	**1.24%**	**0.32%**	**0.47%**

while the initial and final interpolation coefficients ($\omega_{initial}$ and ω_{final}) are selected from $\{0.4, 0.6, 0.8\}$.

We intentionally use random initialization for structure embeddings to ensure alignment with baseline methods (e.g., TransAE and DHNS), which follow the same protocol. While graph pre-training methods such as Node2Vec [12] could offer potential benefits, our choice guarantees that observed performance improvements can be attributed to the proposed RMEL and DINS modules rather than to initialization techniques. Additionally, performance scores for most baseline methods are cited directly from the original literature in which these datasets were benchmarked. For negative sampling approaches like KBGAN [3], NSCaching [62], and SANS [1], which were not evaluated on multimodal datasets, we adopt results reported in DHNS [31], using official open-source implementations and adhering to the recommended hyperparameter configurations.

5.2 Overall Accuracy Evaluation (RQ1)

We compare RelDINS with three baseline categories, as shown in Table 3. RelDINS ranks best or second-best across datasets, demonstrating substantial improvements. To further assess the robustness of these improvements, we conduct paired t-tests over five independent runs. The results show that the RelDINS improvements over the second-best baseline are statistically significant.

On DB15K, RelDINS achieves an MRR of 36.61%, surpassing the best baseline by 4.09%. Hits@1 increases by 9.61 points to 27.14%. On MKG-W, MRR is 37.71% (+5.45%), with gains of +5.72 and +3.67 on Hits@3 and Hits@10, respectively. The substantial improvements in Hit@1 shows that although the DINS module is applied exclusively during training, the resulting refined decision boundaries are internalized effectively. Thus, the model exhibits an enhanced ability to resolve ambiguity among true hard negatives at inference time. On MKG-Y, RelDINS leads with an MRR of 39.45%, maintaining an edge across all ranking cutoffs.

Unimodal KGC methods lag behind RelDINS. On DB15K, RelDINS outperforms GC-OTE by nearly 15 points in MRR, showing the advantage of multimodal cues. Multimodal methods like IKRL, TBKGC, and AdaMF improve over unimodal methods but still trail RelDINS (e.g., Hits@1 +12.13 over AdaMF on MKG-W). This is due to: (i) relation-agnostic fusion diluting discriminative signals, and (ii) alignment issues when multimodal features aren't matched to relation semantics. Advanced samplers (e.g., DHNS, MIRANS) perform well at higher cutoffs but fall short of RelDINS in Hits@1/MRR due to limitations in sampling strategies. These results demonstrate the effectiveness of RelDINS in improving multimodal representation learning and hard negative discrimination.

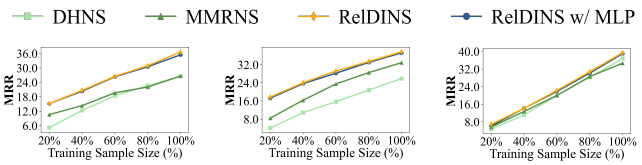
5.3 Model Generalization Analysis (RQ2, RQ3)

Generalization across scoring functions. To answer RQ2, we test three representative scoring functions: TransE (distance-based) [2], DistMult (bilinear) [53], and RotatE (complex-space) [38], combining with six negative sampling strategies: traditional (Uniform [52], Bernoulli [47]), adversarial (KBGAN [3]), self-adversarial (SANS [1]), caching-based (NSCaching), and diffusion-based (DHNS [31]). MIRANS [57] is excluded due to unavailable code.

As shown in Table 4, RelDINS outperforms or matches the best results across all scoring functions, showing strong cross-model generalization. The results demonstrate the ability of RelDINS to generalize well across different embedding geometries. RelDINS generally outperforms DHNS, the top-performing baseline, across most datasets. However, the performance gap narrows on the MKG-Y dataset (e.g., +0.34% in MRR vs. +2.25% on DB15K). This is likely due to MKG-Y's richer multimodal semantics and simpler relational

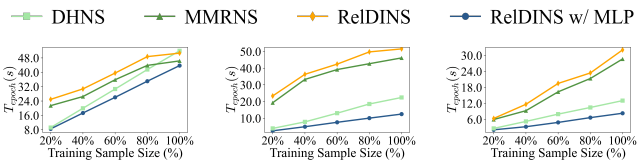
Table 4: Model Generalization across Scoring Functions (Including Efficiency Analysis)

KGE Models	NS Strategies	DB15K					MKG-W					MKG-Y				
		MRR	Hit@1	Hit@10	Train(s)	Infer(s)	MRR	Hit@1	Hit@10	Train(s)	Infer(s)	MRR	Hit@1	Hit@10	Train(s)	Infer(s)
TransE	Uniform	24.86	12.78	47.07	9.63	38.46	29.19	21.06	44.23	3.61	13.35	30.73	23.45	43.37	2.45	9.08
	Bern	30.11	19.04	49.86	9.91	39.38	28.98	22.99	40.11	3.92	13.52	31.12	25.69	43.61	2.94	9.24
	KBGAN	24.00	5.47	50.36	44.87	43.78	24.21	14.51	40.45	21.03	22.06	26.92	19.90	37.41	14.25	16.89
	NSCaching	33.03	**23.31**	50.29	26.48	40.83	28.90	22.24	40.97	15.37	14.03	29.51	24.78	38.02	8.93	10.01
	SANS	26.33	13.87	48.80	24.82	46.82	30.22	23.64	44.61	14.26	26.70	27.51	22.31	42.71	8.21	21.31
	MMRNS	26.61	13.40	50.60	41.56	41.64	33.51	26.25	47.11	43.43	18.06	34.81	28.59	44.76	27.68	10.31
	DHNS	32.63	20.95	52.86	46.87	94.67	33.83	26.46	46.85	18.32	45.32	36.68	32.12	43.62	12.06	27.36
	RelDINS	**33.83**	22.14	**52.95**	43.83	52.65	**36.40**	**29.43**	**48.78**	45.86	15.54	**38.06**	**33.31**	**45.89**	30.24	10.32
DistMult	Uniform	20.99	14.78	39.59	8.16	38.45	20.99	15.93	38.06	3.07	13.26	25.04	19.33	35.95	2.13	9.09
	Bern	27.05	20.04	40.37	8.64	39.17	25.76	21.02	36.16	3.47	13.39	32.00	30.42	38.11	2.66	9.21
	KBGAN	23.36	16.43	36.30	42.35	42.89	19.11	7.69	37.44	20.36	22.03	16.99	9.25	33.33	13.42	16.31
	NSCaching	24.68	19.33	39.74	24.61	40.22	24.70	20.31	37.24	14.23	13.98	27.30	23.55	35.67	8.26	9.91
	SANS	23.62	16.82	39.76	23.07	46.27	23.21	18.46	32.20	13.01	26.38	23.62	16.82	39.76	7.98	20.90
	MMRNS	25.92	14.06	40.42	39.32	41.36	25.92	19.73	**38.20**	40.68	18.05	32.78	27.99	41.13	27.06	10.32
	DHNS	27.20	19.07	**43.01**	42.56	93.76	25.86	20.33	36.53	16.48	45.04	**37.01**	**38.72**	42.53	11.85	27.85
	RelDINS	**28.02**	**20.06**	42.48	40.21	51.97	**27.64**	**22.34**	37.34	43.72	15.45	35.86	32.65	**42.84**	29.96	10.41
RotatE	Uniform	29.28	17.87	49.66	10.72	53.12	33.67	26.80	46.73	6.45	24.82	34.95	29.10	45.30	4.31	14.86
	Bern	20.46	12.83	34.37	11.18	53.98	29.65	24.58	38.80	6.96	24.89	33.63	30.39	39.29	4.88	14.97
	NSCaching	20.32	12.89	34.05	33.52	54.36	32.86	27.86	41.86	20.09	25.38	32.03	29.57	36.32	11.01	16.03
	SANS	30.51	19.13	50.72	31.37	63.10	33.32	27.35	44.67	18.83	39.07	35.28	29.29	44.93	10.31	27.37
	MMRNS	29.67	17.89	51.01	46.34	56.62	34.12	27.37	47.48	46.15	26.93	35.93	30.53	45.47	28.66	16.34
	DHNS	34.36	23.34	**53.72**	51.95	119.63	35.68	28.98	48.12	22.43	58.64	39.11	34.70	46.66	13.02	35.33
	RelDINS	**36.61**	**27.14**	53.70	50.66	83.42	**37.71**	**30.51**	**50.57**	51.55	30.38	**39.45**	**35.13**	**46.88**	32.07	19.07



(a) DB15K (b) MKG-W (c) MKG-Y

Figure 3: MRR under varying sample sizes.



(a) DB15K (b) MKG-W (c) MKG-Y

Figure 4: Training time under varying sample sizes.

structures, where DHNS already captures the global semantic distribution effectively. Nevertheless, RelDINS still maintains an advantage by leveraging relation-cardinality constraints to further refine decision boundaries.

Generalization with different training sample size. To answer RQ3, we compare RelDINS with two top-performing NS methods: DHNS and MMRNS, by varying training sample sizes (20%, 40%, 60%, 80%, 100%). We also evaluate a lightweight RelDINS variant, RelDINS w/ MLP, where the U-Net backbone is replaced with two MLP layers to assess the diffusion-based architecture's impact on performance and efficiency. Figures 3 and 4 show MMKGC performance (MRR) and training time across three datasets.

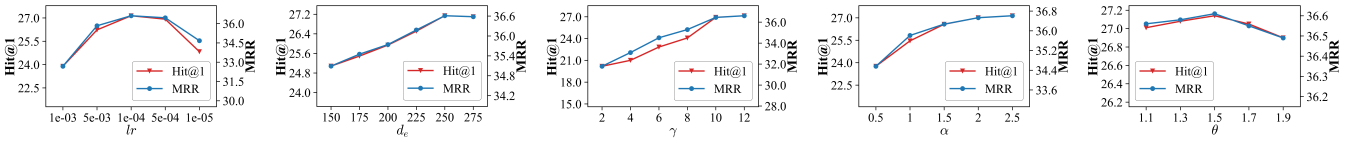
The key observations are summarized as follows: (i) As training sample size increases, MRR improves for all models. (ii) RelDINS and RelDINS w/ MLP outperform DHNS and MMRNS across all datasets, demonstrating the effectiveness of relation-type-aware sampling and adaptive hardness control. RelDINS slightly outperforms its MLP variant (metrics in Table 5), highlighting the benefit of diffusion-based interpolation for generating semantically consistent negatives. (iii) While RelDINS achieves the best performance, it requires longer training due to the iterative denoising process. RelDINS w/ MLP offers a good balance between accuracy and efficiency, reducing training time while maintaining competitive performance. RelDINS is ideal for maximizing accuracy, while RelDINS w/ MLP is more practical for large-scale scenarios.

We proceed to study the computational efficiency of RelDINS empirically. As shown in Section 4.4, the training time complexity per iteration is $O(B \cdot (d^2 + T \cdot C_{\text{net}} + k \cdot d))$, meaning that the total training time per epoch scales linearly with the number of triples $|F|$ when hyperparameters are fixed. This analytical finding is validated empirically in Fig. 4, where the training time exhibits a linear growth trend as the training sample size increases from 20% to 100% across all three datasets. The consistent analytical and empirical findings provide evidence that RelDINS possesses excellent scalability. Moreover, as shown in Table 4, despite the marginal training overhead introduced by the iterative diffusion process, the inference time of RelDINS remains identical to that of standard non-generative models (e.g., 83.42s for the full DB15K test set), as the DINS module is discarded during deployment. This ensures that RelDINS is practical in large-scale, real-time MMKGC scenarios.

5.4 Parameter Sensitivity Analysis (RQ4)

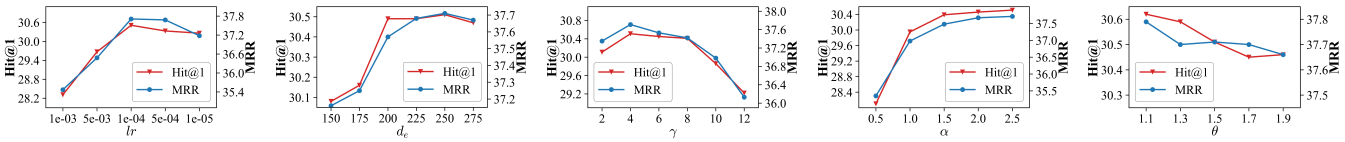
Common parameters for MMKGC. We analyze five standard MMKGC hyperparameters, including the learning rate (lr), entity embedding dimension (d_e), margin (γ), temperature (α), and relation cardinality threshold θ, as shown in Figs. 5–7.

RelDINS demonstrates stable performance under moderate parameter variations. Specifically, the best performance is achieved with learning rates in [1e−4, 5e−4]. As d_e increases from 150 to 275, performance improves initially and then saturates around $d_e = 250$. Small γ values weaken discrimination, whereas large values cause unstable optimization, making $\gamma = 4$ a balanced choice. Similarly, performance improves as α increases from 0.5 to 1.5 and



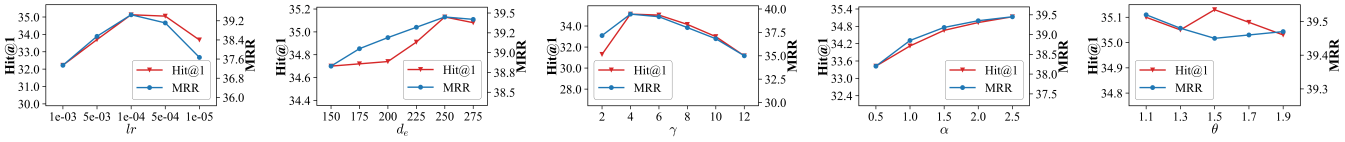
(a) lr (b) d_e (c) γ (d) α (e) θ

Figure 5: Sensitivity analysis on varying common parameters over DB15K.



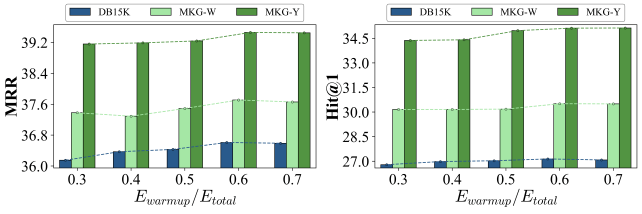
(a) lr (b) d_e (c) γ (d) α (e) θ

Figure 6: Sensitivity analysis on varying common parameters over MKG-W.



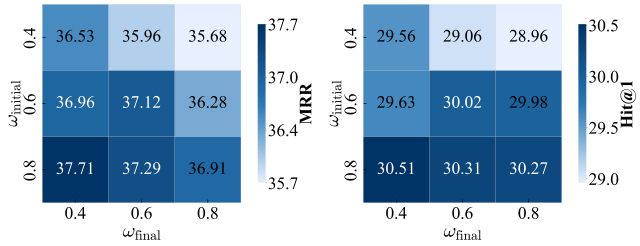
(a) lr (b) d_e (c) γ (d) α (e) θ

Figure 7: Sensitivity analysis on varying common parameters over MKG-Y.



(a) MRR performance (b) Hit@1 performance

Figure 8: Sensitivity analysis on varying (E_{warmup}/E_{total}).



(a) MRR performance (b) Hit@1 performance

Figure 9: Sensitivity analysis on varying $(\omega_{initial}, \omega_{final})$.

then plateaus. In addition, RelDINS remains robust under different θ values, indicating that $\theta = 1.5$ effectively balances relation cardinality modeling and noise resilience.

Negative sampling–specific parameters. We analyze negative sampling parameters, including the warmup epoch ratio (E_{warmup}/E_{total}) and interpolation coefficients $(\omega_{initial}, \omega_{final})$, across all datasets.

As shown in Fig. 8, both MRR and Hit@1 improve with an increasing warmup ratio, peaking at $(E_{warmup}/E_{total}) \approx 0.6$, after which gains plateau. The reason is that a longer warmup phase helps the model transition from easy to hard negatives, preventing overfitting and promoting smoother optimization. Finally, Fig. 9 shows the coupling effect between interpolation coefficients via a heatmap. The optimal configuration, $\omega_{initial} = 0.8$ and $\omega_{final} = 0.4$, follows a "high-initial, low-final" pattern. This aligns with curriculum learning principles: early training focuses on easier negatives, while later stages emphasize harder negatives to refine decision boundaries.

5.5 Ablation Study (RQ5)

We conduct an ablation study across the three datasets. The settings are divided into two parts.

RMEL-related settings. We evaluate five variants of the RMEL module: (i) removing the relation-type-aware embedding mechanism (w/o RMELM) and replacing it with a simple feature concatenation strategy for multimodal representation learning; (ii) retaining relation-type awareness but removing the two-layer projection network (w/o $P_m(\cdot)$), directly using raw features for embedding computation; (iii) removing all multimodal features (w/o Multimodal Info), which tests the importance of multimodal inputs on performance; (iv) replacing average pooling with a gating mechanism (w/ Gate), where modality weights are assigned dynamically through a Softmax-normalized linear layer; (v) replacing average pooling with an inter-modal attention mechanism (w/ Attention), where structural embeddings are used as queries to compute attention weights for visual and textual features.

DINS-related settings. We compare four variants: (i) removing the entire dynamic interpolation diffusion module (w/o DINSM) and adopting uniform negative sampling as a baseline; (ii) removing the dynamic interpolation coefficient mechanism (w/o ω) by fixing $\omega = 0.5$ to evaluate the role of tunable hardness; (iii) replacing spherical linear interpolation with conventional linear interpolation (w/ LI) to test the necessity of geometric consistency; and (iv) replacing the U-Net–structured diffusion model with a two-layer MLP (w/ MLP) to analyze the effect of network architecture on denoising quality.

Table 5: Ablation Study Results

Module	Model	DB15K				MKG-W				MKG-Y			
		MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
	RelDINS	**36.61**	**27.14**	**42.23**	**53.70**	**37.71**	**30.51**	**41.20**	**50.57**	**39.45**	**35.13**	**41.36**	**46.88**
RMEL	w/o RMELM	33.85	22.58	38.57	50.86	35.66	26.37	38.45	48.65	37.25	31.05	39.50	45.11
	w/o $P_m(\cdot)$	35.45	26.13	41.82	53.01	36.31	29.22	39.59	48.91	38.14	34.05	40.22	45.50
	w/o Multimodal Info	34.61	26.02	39.09	51.25	36.03	29.78	38.68	47.82	38.16	34.64	40.01	44.67
	w/ Gate	36.64	27.31	42.00	53.40	36.93	30.08	40.00	49.36	38.73	34.47	40.57	46.00
	w/ Attention	36.48	27.34	41.79	53.07	37.56	30.50	40.80	50.56	39.38	35.04	41.33	46.88
DINS	w/o DINSM	32.85	22.77	37.47	50.51	33.74	25.83	36.79	47.98	35.32	29.80	37.55	44.38
	w/o ω	34.72	25.86	41.63	53.62	35.89	29.24	40.58	50.76	37.58	33.67	40.52	46.60
	w/ LI	36.52	26.95	42.10	53.60	37.63	30.35	41.05	50.97	39.35	34.95	41.25	46.81
	w/ MLP	36.15	26.65	41.77	53.55	37.23	30.03	40.68	50.54	38.92	34.61	40.92	46.75

Results and analysis. As shown in Table 5, removing the relation-type-aware embedding and the projection network yields a significant performance drop, indicating reduced discriminability of multimodal representations. Although performance decreases slightly when multimodal information is removed, the model does not collapse and still maintains competitive predictive accuracy. This indicates that RelDINS does not depend overly on auxiliary modalities and retains robust inference capabilities even in the presence of incomplete data. Interestingly, the empirical findings on the MMKGC task reveal a counterintuitive yet consistent observation: more sophisticated, learnable fusion mechanisms do not outperform simple average pooling on key evaluation metrics. In some cases, they even result in slight performance degradation.

For the DINS module, removing the entire component (w/o DINSM) or fixing the interpolation coefficient (w/o ω) both result in substantial degradation, highlighting the importance of dynamically tunable negative sampling hardness. In contrast, replacing the interpolation or the diffusion model architecture (w/ LI or w/ MLP) leads to relatively minor decreases, suggesting that the overall performance is more sensitive to the dynamic scheduling mechanism than to architectural simplification.

5.6 Qualitative Visualization Analysis (RQ6)

To further analyze the sampling quality of DINS, we employ the t-SNE [41] technique to visualize the distributions of negative samples through interpolation between parent triples. The parent triples (F_s, F_e) are constructed as described in Section 4.3.1, with 60 positive samples (green circles in Fig. 10) selected from the MKG-W dataset for visualization.

During training, the interpolation coefficient ω decreases gradually from 0.8 to 0.2 in the warm-up phase, simulating a curriculum learning process that transitions from easy to hard negatives. We visualize four representative training stages: early training ($\omega = 0.8$), mid-training ($\omega = 0.6$ and $\omega = 0.4$), and late training ($\omega = 0.2$). As shown in Fig. 10, randomly sampled negatives (orange circles) are scattered widely and located far from positive anchors, corresponding to semantically trivial cases that provide limited supervision. In contrast, the visualization captures the dynamic evolution of the sampling process: as ω decreases, the interpolated negatives (blue circles) progressively migrate from regions dominated by random negatives toward the vicinity of positive samples. When ω reaches 0.2, the negatives generated by RelDINS are concentrate densely at the borders of the positive cluster, forming a boundary-like structure in the embedding space. This behavior confirms visually that RelDINS incrementally tightens the decision boundary by synthesizing increasingly difficult negatives.

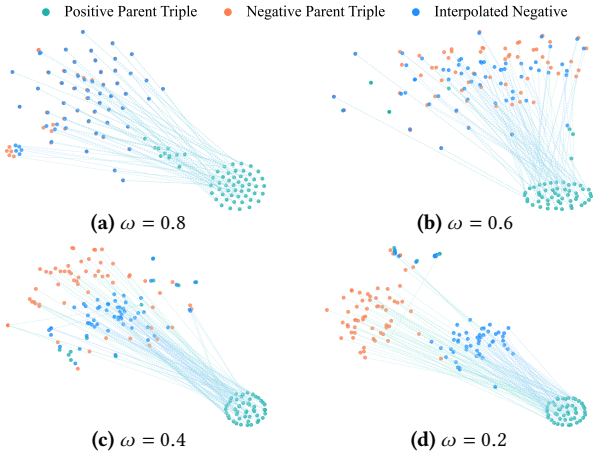


Figure 10: Qualitative visualization example

6 CONCLUSION

We present RelDINS, a unified framework for MKGC that addresses relation-type semantics and negative sample quality. By integrating the RMEL module for relation-aware embedding learning and the DINS module for diffusion-based negative sampling, RelDINS significantly improves link prediction accuracy and robustness across diverse datasets. The adaptive interpolation mechanism introduces curriculum learning to ensure stable optimization and efficient convergence. Visualization confirms that DINS-generated samples are high-quality hard negatives, lying near decision boundaries. Experiments show that RelDINS consistently outperforms state-of-the-art methods and generalizes well across different scoring functions. In the future work, it is of interest to explore extending RelDINS to dynamic, large-scale multimodal knowledge graphs, focusing on its adaptability in evolving environments and heterogeneous data.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62572108, 62002039).

REFERENCES

[1] Kian Ahrabian, Aarash Feizi, Yasmin Salehi, William L. Hamilton, and Avishek Joey Bose. 2020. Structure Aware Negative Sampling in Knowledge Graphs. In *EMNLP*. 6093–6101.

[2] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NeurIPS*. 2787–2795.

[3] Liwei Cai and William Yang Wang. 2018. KBGAN: Adversarial Learning for Knowledge Graph Embeddings. In *NAACL-HLT*. 1470–1480.

[4] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Yuan He, Xiaochun Cao, and Qingming Huang. 2022. OTKGE: Multi-modal knowledge graph embeddings via optimal transport. In *NeurIPS*. 39090–39102.

[5] Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. 2021. PairRE: Knowledge Graph Embeddings via Paired Relation Vectors. In *ACL/IJCNLP (1)*. 4360–4369.

[6] Filip Cornell, Yifei Jin, Jussi Karlgren, and Sarunas Girdzijauskas. 2025. Are We Wasting Time? A Fast, Accurate Performance Evaluation Framework for Knowledge Graph Link Predictors. In *ICDE*. 1650–1663.

[7] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*. 1811–1818.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. 4171–4186.

[9] Yunjun Gao, Xiaoze Liu, Junyang Wu, Tianyi Li, Pengfei Wang, and Lu Chen. 2022. ClusterEA: Scalable Entity Alignment with Stochastic Training and Normalized Mini-batch Similarities. In *KDD*. 421–431.

[10] Yuxia Geng, Jiaoyan Chen, Jeff Z. Pan, Mingyang Chen, Song Jiang, Wen Zhang, and Huajun Chen. 2023. Relational Message Passing for Fully Inductive Knowledge Graph Completion. In *ICDE*. 1221–1233.

[11] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*. 249–256.

[12] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *KDD*. 855–864.

[13] Yu Gu, Kaiqiang Yu, Zhen Song, Jianzhong Qi, Zhigang Wang, Ge Yu, and Rui Zhang. 2022. Distributed Hypergraph Processing Using Intersection Graphs. *IEEE Trans. Knowl. Data Eng.* 34, 7 (2022), 3182–3195.

[14] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *TKDE* 34, 8, 3549–3568.

[15] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *EMNLP*. 139–144.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *NeurIPS*. 6840–6851.

[17] Jing Huang. 2022. Knowledge graph representation learning and graph neural networks for language understanding. In *GRADES-NDA@SIGMOD*. 1–1.

[18] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *ACL*. 687–696.

[19] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*. 1–15.

[20] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. VisualBERT: A Simple and Performant Baseline for Vision and Language. *CoRR* abs/1908.03557.

[21] Ran Li, Shimin Di, Lei Chen, and Xiaofang Zhou. 2024. SimDiff: Simple Denoising Probabilistic Latent Diffusion Model for Data Augmentation on Multi-modal Knowledge Graph. In *SIGKDD*. 1631–1642.

[22] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*. 2181–2187.

[23] Bin Liu and Bang Wang. 2023. Bayesian negative sampling for recommendation. In *ICDE*. 749–761.

[24] Xiaoze Liu, Junyang Wu, Tianyi Li, Lu Chen, and Yunjun Gao. 2023. Unsupervised Entity Alignment for Temporal Knowledge Graphs. In *WWW*. 2528–2538.

[25] Ye Liu, Hui Li, Alberto Garcia-Duran, Mathias Niepert, Daniel Oñoro-Rubio, and David S. Rosenblum. 2019. MMKG: multi-modal knowledge graphs. In *ESWC*. 459–474.

[26] Xinyu Lu, Lifang Wang, Zejun Jiang, Shichang He, and Shizhong Liu. 2022. MMKRL: A robust embedding approach for multi-modal knowledge graph representation learning. *Appl. Intell.* 52, 7, 7480–7497.

[27] Chuangtao Ma, Yongrui Chen, Tianxing Wu, Arijit Khan, and Haofen Wang. 2025. Unifying Large Language Models and Knowledge Graphs for Question Answering: Recent Advances and Opportunities. In *EDBT*. 1174–1177.

[28] Christian Meilicke, Melisachew Wudage Chekol, Patrick Betz, Manuel Fink, and Heiner Stuckenschmidt. 2025. Correction: Anytime bottom-up rule learning for large-scale knowledge graph completion. *VLDBJ* 34, 4, 48.

[29] Hatem Mousselly-Sergieh, Teresa Botschen, Iryna Gurevych, and Stefan Roth. 2018. A multimodal translation-based approach for knowledge graph representation learning. In *SEM@NAACL-HLT*. 225–234.

[30] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *ICML*. 809–816.

[31] Guanglin Niu and Xiaowei Zhang. 2025. Diffusion-based hierarchical negative sampling for multimodal knowledge graph completion. *CoRR* abs/2501.15393.

[32] Andrea Rossi, Donatella Firmani, Paolo Merialdo, and Tommaso Teofili. 2022. Explaining Link Prediction Systems based on Knowledge Graph Embeddings. In *SIGMOD*. 2062–2075.

[33] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*. 1–14.

[34] Zhen Song, Yu Gu, Jianzhong Qi, Zhigang Wang, and Ge Yu. 2022. EC-Graph: A Distributed Graph Neural Network System with Error-Compensated Compression. In *ICDE*. 648–660.

[35] Zhen Song, Yu Gu, Qing Sun, Tianyi Li, Yanfeng Zhang, Yushuai Li, Christian S. Jensen, and Ge Yu. 2024. DynaHB: A Communication-Avoiding Asynchronous Distributed Framework with Hybrid Batches for Dynamic GNN Training. *Proc. VLDB Endow.* 17, 11 (2024), 3388–3401.

[36] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*. 697–706.

[37] Ye Sun, Lei Shi, and Yongxin Tong. 2025. eXpath: Explaining Knowledge Graph Link Prediction with Ontological Closed Path Rules. *PVLDB* 18, 9, 2818–2830.

[38] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR (Poster)*.

[39] Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, and Bowen Zhou. 2020. Orthogonal Relation Transforms with Graph Context Modeling for Knowledge Graph Embedding. In *ACL*. 2713–2722.

[40] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *ICML*, Vol. 48. 2071–2080.

[41] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR* 9, 11, 2579–2605.

[42] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10, 78–85.

[43] Meng Wang, Sen Wang, Han Yang, Zheng Zhang, Xi Chen, and Guilin Qi. 2021. Is Visual Context Really Helpful for Knowledge Graph? A Representation Learning Perspective. In *ACM MM*. 2735–2743.

[44] Peifeng Wang, Shuangyin Li, and Rong Pan. 2018. Incorporating GAN for Negative Sampling in Knowledge Representation Learning. In *AAAI*. 2005–2012.

[45] Yuxiang Wang, Arijit Khan, Tianxing Wu, Jiahui Jin, and Haijiang Yan. 2020. Semantic guided and response times bounded top-k similarity search over knowledge graphs. In *ICDE*. 445–456.

[46] Zikang Wang, Linjing Li, Qiudan Li, and Daniel Zeng. 2019. Multimodal Data Enhanced Representation Learning for Knowledge Graphs. In *IJCNN*. 1–8.

[47] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*. 1112–1119.

[48] Yuyang Wei, Wei Chen, Xiaofang Zhang, Pengpeng Zhao, Jianfeng Qu, and Lei Zhao. 2024. Multi-Modal Siamese Network for Few-Shot Knowledge Graph Completion. In *ICDE*. 719–732.

[49] Junyang Wu, Tianyi Li, Lu Chen, Yunjun Gao, and Ziheng Wei. 2023. SEA: A Scalable Entity Alignment System. In *SIGIR*. 3175–3179.

[50] Ruobing Xie, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2017. Image-embodied Knowledge Representation Learning. In *IJCAI*. 3140–3146.

[51] Derong Xu, Tong Xu, Shiwei Wu, Jingbo Zhou, and Enhong Chen. 2022. Relation-enhanced Negative Sampling for Multimodal Knowledge Graph Completion. In *ACM MM*. 3857–3866.

[52] Derong Xu, Jingbo Zhou, Tong Xu, Yuan Xia, Ji Liu, Enhong Chen, and Dejing Dou. 2023. Multimodal Biological Knowledge Graph Completion via Triple Co-Attention Mechanism. In *ICDE*. 3928–3941.

[53] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*. 1–12.

[54] Yueji Yang, Divyakant Agrawal, H. V. Jagadish, Anthony K. H. Tung, and Shuang Wu. 2019. An Efficient Parallel Keyword Search Engine on Knowledge Graphs. In *ICDE*. 338–349.

[55] Zhen Yang, Ming Ding, Chang Zhou, Hongxia Yang, Jingren Zhou, and Jie Tang. 2020. Understanding negative sampling in graph representation learning. In *SIGKDD*. 1666–1676.

[56] Jiasheng Zhang, Deqiang Ouyang, Shuang Liang, and Jie Shao. 2025. Towards Pattern-aware Data Augmentation for Temporal Knowledge Graph Completion. *PVLDB* 18, 10, 3573–3586.

[57] Ruijia Zhang, Zheng Dong, Jianqiang Zhang, Gongpeng Song, and Qin Lu. 2025. Multimodal Knowledge Graph Completion Method Based on Integrated Modality Adversarial Training and Relation-Enhanced Attention Mechanism. In *CSCWD*. 1794–1799.

[58] Yichi Zhang, Mingyang Chen, and Wen Zhang. 2023. Modality-aware negative sampling for multi-modal knowledge graph embedding. In *IJCNN*. 1–8.

[59] Yufeng Zhang, Wei Chen, Xi Chen, Qingzhi Ma, and Lei Zhao. 2024. Inductive Link Prediction for Sequential-emerging Knowledge Graph. In *ICDE*. 5753–5766.

[60] Yichi Zhang, Zhuo Chen, Lei Liang, Huajun Chen, and Wen Zhang. 2024. Unleashing the Power of Imbalanced Modality Information for Multi-modal Knowledge

Graph Completion. In *LREC/COLING*. 17120–17130.
[61] Yufeng Zhang, Weiqing Wang, Hongzhi Yin, Pengpeng Zhao, Wei Chen, and Lei Zhao. 2023. Disconnected Emerging Knowledge Graph Oriented Inductive Link Prediction. In *ICDE*. 381–393.
[62] Yongqi Zhang, Quanming Yao, Yingxia Shao, and Lei Chen. 2019. NSCaching: Simple and Efficient Negative Sampling for Knowledge Graph Embedding. In *ICDE*. 614–625.
[63] Yichi Zhang and Wen Zhang. 2022. Knowledge graph completion with pre-trained multimodal transformer and twins negative sampling. *CoRR* abs/2099.07084.