JKDM: A Joint Structural and Semantic Diffusion-Generated Knowledge Completion Model

Wendong Zhang[1], Haoqi Chen[1], and Song Yu[2(✉)]

[1] School of Software, Xinjiang University, Urumqi, 830000, Xinjiang, China
{wdzhang@xju.edu.cn,107552304905@stu.xju.edu.cn}
[2] School of Computer Science and Engineering, Central South University, Changsha, 410083, Hunan, China ys@csu.edu.cn

Abstract. Knowledge Graph Completion (KGC) aims to predict missing triples in a graph based on known relationships between entities. However, most KGC methods face the challenge of diversification representations among entities, making it difficult for models to link entities effectively. This article proposes a Joint Knowledge (Structure-Semantics) Diffusion Model (JKDM) to capture entity diversification relationships. By leveraging the probabilistic generative capabilities of diffusion models, JKDM generates diversification outputs that align with the distribution of target entities rather than producing a single deterministic result. Considering the insufficient structural information of sparse entities, which leads to their representations tending toward a smooth distribution, making it difficult for diffusion models to learn their probability distributions, we jointly enhance sparse entity representations using structural and semantic information. Structurally, a Dual-channel Graph Attention Network (DGAT) is introduced to capture structural embeddings of entities from different perspectives. Semantically, a contextual path strategy is applied to pre-trained language models (PLMs) to enrich entity semantics. Under the condition of joint embeddings, JKDM gradually denoises to generate the probability distribution of target entities. Experiments demonstrate that JKDM outperforms SOTA methods on the FB15k-237, WN18RR, and UMLS datasets, achieving improvements of 2.3%, 1.5%, and 0.43% in MRR scores, respectively.

Keywords: Diffusion Models · Knowledge Graph Completion · Attention Networks · Link Prediction.

1 Introduction

Knowledge Graphs (KG) store real-world facts in triples, represented as (h, r, t). Although KGs play a crucial role in numerous applications, such as KG-enhanced LLMs and recommendation systems, they still face the issue of incompleteness. Knowledge Graph Embedding (KGE) is an effective approach for inferring missing triples, such as TransE [1], which map entities and relations into

low-dimensional vector spaces and use scoring functions to calculate the plausibility of predicted triples. However, these methods struggle to handle the diverse representations among entities. To address this issue, most KGC methods focus on encoding tasks or aggregating more information structurally to enhance the diverse features of entities. For example, CompGCN [2] aggregate neighborhood information of entities to obtain diverse information. Alternatively, they aim to semantically understand the meaning behind entity texts, uncovering more hidden knowledge connections. For instance, KG-BERT [3] leverage PLMs to capture the contextual features of entities. However, in decoding tasks, using only KGE for deterministic result computation still struggles to handle the issue of diverse representations among entities. Diffusion models (DM), by combining randomness and multimodal modeling capabilities, can generate uncertain results, enabling them to produce multiple plausible outcomes from the same input. However, when faced with sparse distribution data, DM often struggles to generate ideal results due to insufficient precision in the denoising process. In KG, sparse entities tend to have sparse distributions due to insufficient structural information, making it difficult for diffusion models to learn their probability distributions, affecting the generated results.

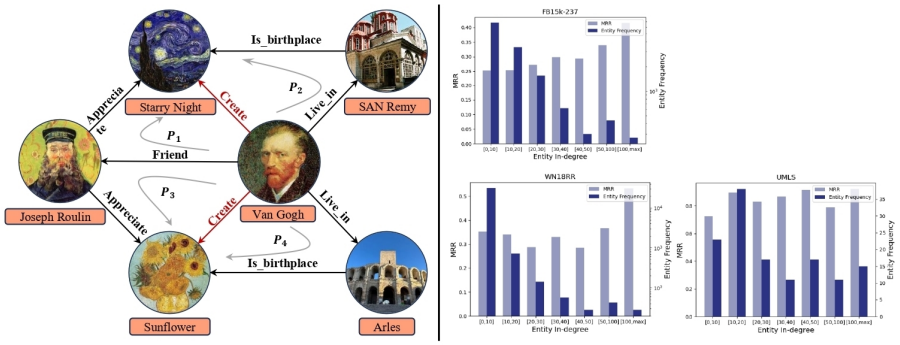

Fig. 1. The left side illustrates that the "Create" relationship carries multiple potential associations. The right side demonstrates the different contextual paths in which the triple exists. On the FB15k-237, WN18RR, and UMLS datasets, the results based on the RotatE [4] model for entities with different in-degrees and MRR reveal that sparse entities perform poorly in knowledge reasoning tasks.

Despite significant progress in KGC, limitations still exist. **Challenge 1:** Sparse entities in knowledge graphs suffer from insufficient structural information. We investigated several widely recognized KGs to explore the relationship between entity frequency and link prediction, as shown in Fig. 1. It was observed that there is a correlation between model performance and entity sparsity. To address sparsity, previous methods often introduce textual features for enhancement but lack a deep understanding of triple context. Some approaches map

neighboring entities to different representation spaces to enrich structural embeddings, yet the neighborhood information of sparse entities remains unchanged, and the structural features provided are still limited.**Challenge 2:** The issue of diverse representations among entities. Due to relationships often encompassing different semantic levels, entity pairs can exhibit one-to-many, many-to-one, and other associations. For example, as shown in Fig. 1, the relationship "$Create$" carries multiple potential associations. Existing methods primarily rely on KGE models for deterministic entity prediction, which can only capture specific semantics of relationships but lack the ability to handle diverse relationships between entities.

We propose the Joint Knowledge Diffusion Model (JKDM) to address these issues. The diverse representations among entities lead to relationships in KGs that may be nonlinear and complex. As a probabilistic generative model, DM enables the generation of complex distributions. Therefore, we designed a Knowledge Diffusion Generation (KDG) module using DM for multi-relational semantic reasoning, gradually adding noise and learning the reverse process to generate complex probability distributions. However, sparse entities lack sufficient neighborhood information, causing their representations to tend toward sparse distributions, making it difficult for DM to learn their probability distributions. While densification techniques can supplement sparse structures, the introduced dense relationships may affect the original relational features. Therefore, we designed a Dual-channel Graph Attention Network (DGAT) module consisting of two independent GAT networks: RGAT and EGAT. RGAT operates on the non-densified graph, focusing on neighborhood relation aggregation and capturing interaction patterns between entities and specific relations. EGAT emphasizes aggregating neighboring entities in the densified graph, computing the intrinsic interactions between the central entity and its original neighboring entities and similar entities. Additionally, we designed a Contextual Path Enhancement for Semantics (CPES) module to uncover hidden connections behind knowledge. This module formulates questions about the target entity using question-answering templates and answers them using the reasoning paths of the target entity, thereby enriching the semantics of the target entity through contextual paths. For example, as shown in Fig. 1, the triple $(VanGogh, Create, Sunflower)$ is transformed into $(What\ did\ VanGogh\ create?\ \ Sunflower)$, and the semantic representation of the entity "$Sunflower$" is enriched through the path $(VanGogh, Friend, JosephRoulin, Admire, Sunflower)$. The main contributions are as follows:

- We propose a Joint Knowledge Diffusion Model (JKDM). The DGAT and CPES modules enhance the representations of sparse entities from structural and semantic perspectives, while the KDG module leverages structural and semantic features to diffusively generate representations of predicted entities.
- To address the issue of incomplete structural information, DGAT enhances the structural information of sparse entities by leveraging Dense Graph information in a dual-channel GAT manner. CPES employs a question-answering

template format and trains PLMs using a contextual path strategy to enhance entity semantics.
- To address the issue of diverse representations among entities, the Knowledge Diffusion Generation (KDG) module is designed. By leveraging diffusion models to gradually add noise and learn the reverse process, it enables the generation of complex probability distributions.
- The model JKDM is evaluated on three datasets. The experimental results demonstrate that JKDM achieves superior performance.

2 Related Work

2.1 Knowledge Completion

Structure-based methods: Early research applied CNNs to KGC, such as ConvE [5]. These methods focus on individual triples, overlooking the topological structure. Therefore, GNN-based KGC methods, such as CompGCN, have been proposed, which jointly embed entities and relations into the knowledge graph using composition operators. SACN [6] designs a weighted GCN by assigning different weights to different relation types.
Semantics-based methods: The sparsity of knowledge graphs makes it difficult for models to learn high-quality entity embeddings. Therefore, researchers have begun to introduce textual information to enhance entity embeddings. AATE [7] encodes text and combines it with topological structures to enhance entity embeddings, but this method lacks the utilization of contextual representations of triples. In contrast, KG-BERT treats triple text sequences as input to PLMs to obtain contextual semantic representations of entities.
Structure-Semantics-based methods: The model leverages structural and semantic information to enhance entity features. For instance, GS-InGAT [8] considers neighbourhood interactions and global semantics, while SEA-KGC [9] uses PLMs to learn unified representations from entity structures and text. However, existing methods lack deep contextual understanding and fail to address incomplete structural information for sparse entities. These methods rely on deterministic entity prediction, capturing only specific relationship semantics and struggling with diverse entity relationships. This article's JKDM enhances sparse entity neighbourhoods with densified graphs and enriches semantics via contextual paths. It also introduces diffusion models to generate complex probability distributions by adding noise and learning the reverse process.

2.2 Diffusion Model

The diffusion process of diffusion models is implemented through two Markov chains: the forward noising process gradually corrupts samples into Gaussian noise, while the reverse denoising process progressively restores the data. In the past, diffusion models have been primarily applied in computer vision, such as visual generation, multimodal generation, and image restoration.

In recent years, researchers have begun to introduce diffusion models for knowledge graph tasks. For example, FDM [10] utilizes diffusion models to directly learn the distribution of credible facts from known knowledge graphs. KGDM [11] transforms entity prediction tasks into conditional fact generation tasks using diffusion models. However, the methods above overlook the knowledge graph's topological structure and semantic information, insufficiently representing entity features. Therefore, we propose the Joint Knowledge Diffusion Model (JKDM), which enhances entity representations by combining structural and semantic features of the knowledge graph. Using the designed Joint Embedding Condition Denoising (JECD) module, JKDM learns the reverse diffusion process under the joint embedding conditions of entities and relations.

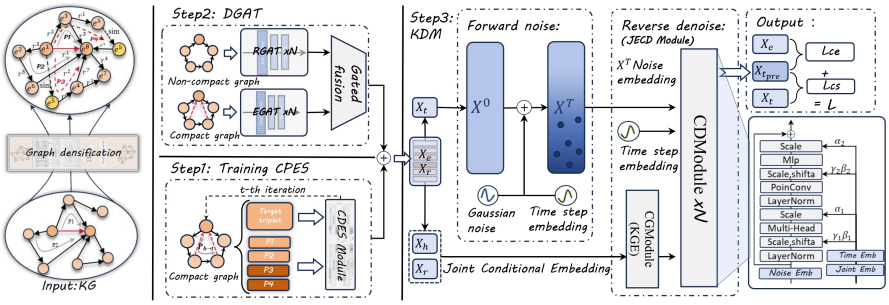

Fig. 2. Step 1: The CPES module aims to obtain contextual semantic; Step 2: The DGAT module aims to obtain structural; Step 3: The KDM module implements diffusion generation through a forward noising process and a reverse denoising process.

3 Methodology

We treat the KG as a set of triples $\mathcal{G} = \left\{\left(e^i, r^u, e^j\right)\right\}$, where e^i and e^j denote the head and tail entities, and r^u represents the relation. The architecture of the JKDM model is shown in Fig. 2. JKDM mainly consists of three modules:

3.1 Graph Densification

Dense Graph(DG): The sparsity of knowledge graphs leads to missing structural information. Malaviya et al. [12] introduced graph densification techniques to address this issue. The graph densification process is achieved solely through semantic similarity between entity texts, neglecting the contextual information of entities. Liu X et al. [13] indicate that extracting entity semantic embeddings without context or solely from the [CLS] token is suboptimal. Inspired by this, We calculate contextual semantic similarity to densify sparse entities. For example, the context of the head entity e^i in the triple $\left(e^i, r^u, e^j\right)$ is defined as the set

of triples containing e^i, i.e.$C_{e^i} = \{(e^i, r^u, e^j)|(e^i, r^u, e^j) \in G \cup (e^j, r^u, e^i) \in G\}$. To obtain the semantics of e^i in the context of C_{e^i}, e^i is replaced with $[S]e^i[/S]$, and markers are added at both ends of e^i to capture its textual semantic embeddings. Using Bert, all marker pairs $[S]e^i[/S]$ captured from the context of the target entity are aggregated through average pooling to obtain the full semantics of e^i in C_{e^i}.

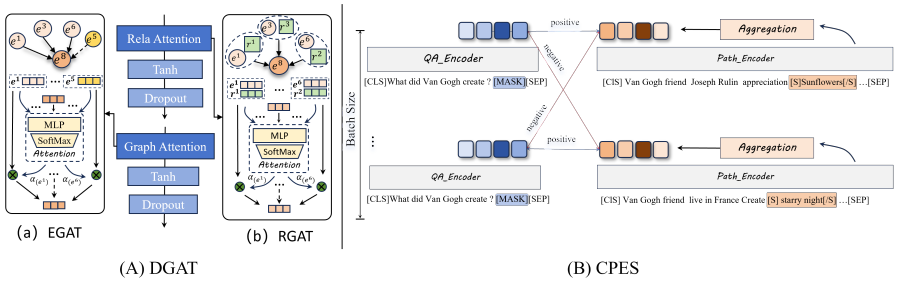

Fig. 3. A.(a) EGAT performs attention-based weighted aggregation of the DG from the entity perspective. (b) RGAT performs weighted aggregation on the KG from the entity-relation perspective. B.The CPES module encodes $\mathcal{T}_{QA}$ and $\mathcal{T}_{Path}$ using the QA_Encoder and Path_Encoder, respectively.

3.2 Context Path Enhance Semantics

Inspired by Pan Y et al. [14], we designed the CPES module to fine-tune PLMs using contextual paths for semantic enhancement. CPES employs a dual-encoder structure combining QA_Encoder and Path_Encoder. It formulates questions and answers in the form of questions-answers templates (QA) for the predicted entities, transforming the triple into question and answer sequences, denoted as $\mathcal{T}_{QA}$ and $\mathcal{T}_{Path}$, respectively, as shown in Fig. 3(B).

QA_Encoder: Converts $\left(e^i, r^u, e^j\right)$ into a question sequence $\mathcal{T}_{QA}$. For example, $(VanGogh, Create, StarryNight)$ is transformed into $(What\ did\ VanGogh\ Create\ ?)$. Then, the target entity $(StarryNight)$ is masked, and the question and mask are concatenated to form the question sequence $(What\ did\ VanGogh\ Create?\ [MASK])$. The formula is as follows:

$$\mathcal{T}_{QA} = Q_r(e^i) \oplus [MASK]_{e^j} \tag{1}$$

where $\oplus$ denotes the concatenation operation, $Q_r(e^i)$ denotes the transformation of triple into a question sequence, and $[MASK]_{e^j}$ masks the target entity.

Path_Encoder: Converts the set of contextual paths from entity e^i to entity e^j into an answer sequence $\mathcal{T}_{Path}$. The set of contextual paths is searched using the Breadth-First Search (BFS) algorithm. However, the sparsity of the knowledge graph may make it difficult to find contextual paths within k hops

for certain target triples. Therefore, this article conducts path searches on the DG. The formula for $\mathcal{T}_{Path}$ is as follows:

$$\mathcal{T}_{Path} = (e^i, r^u, e^j) \oplus \mathcal{P}_{e^i \to e^j} \ , \ h_e = Pool\big(\big[h_{e,1}, ..., h_{e,m}\big]\big) \tag{2}$$

where $\mathcal{P}_{e^i \to e^j}$ denote the paths from entity e^i to entity e^j. To obtain the representation of the answer entity e^j in different paths, visual marker pairs [S][\S] are added to the answer entity in the path set. For example, $(e^i, r^1, ..., e^j)$ is marked as $(e^i, r^1, ..., [S]e^j[\backslash S])$. The semantic representations of entity e^j in different contextual paths, denoted as $h_{e,1}, \ldots, h_{e,m}$, are aggregated using average pooling *Pool* to obtain the comprehensive contextual path semantics of entity.

CPES utilizes PLM to encode the $\mathcal{T}_{QA}$ sequence and $\mathcal{T}_{Path}$ sequence separately. In a question-answering format, the masked semantics in $\mathcal{T}_{QA}$ are semantically aligned with the comprehensive contextual path semantics in $\mathcal{T}_{Path}$ for fine-tuning the PLM. The main idea is to leverage the intrinsic connection between questions and answers to enhance the semantic understanding capability of the PLM. A semantic alignment loss $\mathcal{L}_{aligned}$ is designed in CPES for contrastive learning between positive and negative samples. It calculates the distance between the embedding of $[MASK]_{e^j}$ and the contextual path embeddings in $\mathcal{T}_{Path}$, as well as the distance to negative sample embeddings as follows.

$$\mathcal{L}_{aligned} = \max\left(0, d(h_{[MASK]_e}, h_{\overline{e}}\right) - d(h_{[MASK]_e}, h_e) + \eta) \tag{3}$$

where d calculates the distance between the positive and negative samples and the masked embedding. $h_{[MASK]_e}$ represents the masked embedding in $\mathcal{T}_{QA}$, $h_{\overline{e}}$ denotes the comprehensive contextual path semantics, $h_{\overline{e}}$ represents the comprehensive contextual path semantics of negative samples, and η is the hyperparameter for the margin loss. After training, the entity text (e, r) is input into the trained CPES module, and the hidden state of the [CLS] token is used as the node semantic embedding, denoted as $h_e^S = ([cls]|e)$,$h_r^S = ([cls]|r)$.

3.3 Dual-Channel GAT Module

DGAT consists of two independent GAT Networks: the RGAT Network and the EGAT Network. In the form of a dual-channel GAT, it captures entity structural features in different feature spaces and employs a gating network for feature fusion. The overall architecture of this part is shown in Fig. 3(A). The formula is as follows:

$$h_{e^i}^G = \left(gate * LR\left(W_{fuse}\left(h_{e^i}^{(R)}, h_{e^i}^{(E)}\right)\right) + (1 - gate)h_{e^i}^{(R)}\right) + h_{e^i}^{(R)} \tag{4}$$

$$gate = Sigmoid\left(W_{gate}\left(h_{e^i}^{(R)}, h_{e^i}^{(E)}\right)\right) \tag{5}$$

where $h_{e^i}^G$ represents the entity structural embedding, $h_{e^i}^{(R)}$ denotes the entity embedding of the RGAT layer, $h_{e^i}^{(E)}$ denotes the entity embedding of the EGAT

layer, and *gate* represents the gating vector. LR represents the LeakyReLU activation function. $W_{gate} \epsilon \mathbb{R}^{2\times d}$,$W_{fuse} \epsilon \mathbb{R}^{2\times d}$respectively represent the weight matrices of the gating vector generator and the feature fusion module.

RGAT channel: The edges in KG are relationships with specific semantics. During the message-passing process, RGAT updates the features of the central entity by considering the features of neighboring entities and the specific relationship features. The formula is as follows:

$$h_{e^i}^{(R)(l)} = \delta\left(\sum_{(u,j)\in N_i} \alpha_{iuj} f(h_{e^j}, h_{r^u}) + W^l h_{e^i}^{(R)(l-1)}\right) \tag{6}$$

where δ denotes the Tanh activation function, α_{iuj} denotes the relational attention weight of the triple. $f\left(h_{e^j}^{(R)(l-1)}, h_{r^u}^{(R)(l-1)}\right)$ denotes the local topological context aggregation message of the triple, specifically denoting the fusion operation between $h_{e^j}^{(R)(l-1)}, h_{r^u}^{(R)(l-1)}$, such as $h_{e^j} - h_{r^u}$. α_{iuj} is used to distinguish the importance of different triples in the neighborhood. α_{iuj} defined as:

$$\alpha_{iuj} = Softmax\left(W_1^{(l)}\left(\delta\left(W_2^{(l)}\left(h_{e^i}^{(l-1)}, h_{r^u}^{(l-1)}, h_{e^j}^{(l-1)}\right)\right)\right)\right) \tag{7}$$

where δ represents the LeakyReLU activation function, $W_1^{(l)} \epsilon \mathbb{R}^{1\times d}$ and $W_2^{(l)} \epsilon \mathbb{R}^{3\times d}$ are learnable parameters in RGAT. $h_{e^i}^{(R)(l-1)}, h_{r^u}^{(R)(l-1)}$ and $h_{e^j}^{(R)(l-1)}$ denote the embedding of the head entity, relation, and tail entity at the *l-1*-th layer. Then, the attention score are normalized using *Softmax*. The relation representation is also transformed as follows:

$$h_{r^u}^{G} = h_{r^u}^{(l)(R)} = W_{rela}^{(l)} \cdot h_{r^u}^{(l-1)(R)} \tag{8}$$

where $h_{r^u}^{G}$ represents the structural embedding of the relation, and $W_{rela}^{(l)} \in \mathbb{R}^{1\times d}$ is the trainable weight matrix for the relation embedding at layer l.

EGAT channel: The EGA module operates on the DG, capturing the intrinsic interactions between the central node and its neighboring nodes, as well as similar nodes, without considering the relational features within the neighborhood. Specifically, it aggregates the dense neighborhood information of the central entity in the DG to enhance the structural features of sparse entities. The formula is as follows:

$$h_{e^i}^{(E)(l)} = \delta^1\left(\sum_{e^j\in N_i} \alpha_{ij} \cdot h_{e^j}^{(E)(l-1)} + W^l h_{e^i}^{(E)(l-1)}\right) \tag{9}$$

$$\alpha_{ij} = Softmax\left(W_1^{(l)}\left(\delta^2\left(W_2^{(l)}\left(h_{e^i}^{(l-1)}, h_{e^j}^{(l-1)}\right)\right)\right)\right) \tag{10}$$

where δ^1 represents the Tanh activation function, δ^2 represents the LeakyReLU activation function, α_{ij} denotes the relational attention weight of the triple to distinguish their importance.

3.4 Knowledge Diffusion Generation

The diffusion model consists of the noise-adding process and the conditional denoising process, as shown in Fig. 2. Specifically, the noise-adding process disrupts the vector distribution of the target entity by adding noise. In contrast, the denoising process learns the reverse denoising process step by step, enabling it to understand the underlying distribution of the target entity.

Noise-adding process: The entity and relation features are defined as $X_e = Cat(h_e^G, h_e^S)$,$X_r = Cat(h_r^G, h_r^S)$, whereX_h, X_t, and X_r represent the head entity, tail entity, and relation, respectively. The noise-adding process gradually adds Gaussian noise to the tail entity X_t until the time step $T_p = T$, at which point X_t is corrupted by Gaussian noise and mapped to a pure noise embedding, denoted as $X_t^{T_p}$. The formula is as follows:

$$q\left(X_t^{T_p}|X_t^{T_p-1}\right) = \mathcal{N}\left(X_t^{T_p}; \sqrt{1-\beta^{T_p}}X_t^{T_p-1}, \beta^{T_p}I\right) \tag{11}$$

where T_p is the total number of time steps in the diffusion process, and β^{T_p} represents the variance in the diffusion process. After reparameterization, the output X_t^k at any time step $T_p = k$ takes the form:

$$X_t^k = \sqrt{\bar{\alpha}^k}X_t^0 + \sqrt{1-\bar{\alpha}^k}\varepsilon^k \tag{12}$$

where $\alpha^k = 1-\beta^k, \bar{\alpha}^k = \prod_{z=1}^{k}\alpha^z$, and ε^k represents as Gaussian noise,$\varepsilon^k \sim \mathcal{N}(0, I)$.

Conditional denoising process: Conditioned on the known head entity embedding X_h, relation embedding X_r, and the time step T_p, it gradually denoises the pure noise embedding $X_t^{T_p}$ generated by the noise-adding process. Finally, the predicted entity X_t^{pre} is generated in the vector space. The formula is as follows:

$$p_\theta\left(X_t^{T_p-1} \mid X_t^{T_p}, T_p, X_h, X_r\right) = \mathcal{N}\left(X_t^{T_p}; \mu_\theta\left(X_t^{T_p}, T_p, X_h, X_r\right), \sigma_{T_p}^2 I\right) \tag{13}$$

where $\sigma_{T_p}^2$ is the constant variance, μ_θ represented as the computed mean of the normal distribution, θ represented as the reverse conditional denoising process parameters.

$$\mu_\theta\left(X_t^{T_p}, T_p, X_h, X_r\right) = \frac{1}{\sqrt{\alpha^{T_p}}}X_t - \frac{1-\alpha^{T_p}}{\sqrt{\alpha^{T_p}}\sqrt{1-\overline{\alpha}^{T_p}}}\varepsilon_\theta\left(X_t^{T_p}, T_p, X_h, X_r\right) \tag{14}$$

where β^{Tp} is the variance of the forward process, $\beta^{T_p} = 1-\alpha^{T_p}$, $\alpha^{T_p} = \prod_{s-1}^{T_p}\alpha^s$ is used to predict the additional noise ε_{pre}^k at any time step $T_p = k$. It consists of the Condition Generation Module (CGModule) and the Condition Denoising Module (CDModule), as shown in the following formula:

$$\varepsilon_\theta\left(X_t^{T_p}, T_p, X_h, X_r\right) = \text{CDModule}\left(X_t^{T_p}, T_p, \text{CGModule}(X_h, X_r)\right) \tag{15}$$

CGModule: To ensure that the entities generated during the denoising process align more closely with the actual knowledge graph, we employ Knowledge

Graph Embedding (KGE) methods for deterministic conditional embedding generation (CGModule), such as TransE. This ensures that the generation process adheres to the structural constraints of the knowledge graph. For instance, at each step of the denoising process in the diffusion model, the embeddings of X_h and X_r are used to compute gradients to adjust the generation direction.

CDModule: It aims to combine noise, conditional embeddings, and time step embeddings to achieve the denoising process. To fully leverage conditional embeddings for guidance, parameters α, β, γ are set for scaling and shifting operations to adjust the vector distribution. Additionally, multi-head attention mechanisms and PoinConv extract and process features at different levels, helping the model better understand and utilize conditional embedding information. Finally, residual connections enhance the model's stability and performance.

3.5 Training and Reasoning

Supervised contrastive learning [15] aims to bring the generated embeddings of the same entity closer together and push the generated embeddings of different entities farther apart. At the same time, entity category labels guide the model in learning more discriminative feature representations. The loss is calculated as follows:

$$\mathcal{L}_{\mathrm{CL}} = \sum_{t \in \mathcal{T}} \frac{-1}{|\mathcal{T}_t|} \sum_{X_t^{pro} \epsilon \mathcal{T}_t} \log \frac{\exp(X_t^{pre} \cdot X_t / \tau)}{\sum_{k \in \mathcal{T}_t} \exp(X_k^{pre} \cdot X_t / \tau)} \tag{16}$$

where $\mathcal{T}$ represents the entity embeddings, $\mathcal{T}_t$ is the set of entity t, and t is a learnable parameter that controls the balance between uniformity and tolerance. X_t^{pre}represents the generated entity embedding from the reverse conditional denoising process and scores all candidate entities using the dot product. The cross-entropy loss is as follows:

$$\mathcal{L}_{CE} = -\frac{1}{|\mathcal{T}|} \sum_{(h,r) \in \mathcal{T}} \sum_{t \in E} y_{(h,r)}^t \cdot \log \hat{y}_{(h,r)}^t \tag{17}$$

where $\mathcal{T}$ represents the training triples in the batch, E denotes all entities present in the KG, $y_{(h,r)}^t$ represents the true label, $\hat{y}_{(h,r)}^t$ represents the plausibility score between the generated entity and the candidate entity set, and $X_e \epsilon \mathcal{R}^{2 \times d \times |\mathrm{E}|}$ represents the joint embedding representation of all entities. The final loss is optimized by combining the contrastive and cross-entropy losses.

We jointly optimize the two loss objectives $\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{\mathrm{CL}}$, computing the similarity of diffusion-generated embeddings for the same entity and calculating the contrastive loss with embeddings of other entities. Simultaneously, we compute the similarity scores with the candidate entity set and calculate the cross-entropy loss. During the inference phase, the trained DGAT and CPES perform coarse-grained aggregation of entities. Then, KDG iteratively denoises random Gaussian noise conditioned on the entity feature (X_e, X_r) until $T_p = T$, generating the predicted entity X_t^{pre} in the vector space.

4 Experiment

4.1 Experimental Setup

Dataset: We considered three widely recognized datasets to evaluate the JKDM. FB15k-237 [16]: A subset of FB15k with inverse relations removed. WN18RR [17]: A subgroup of WN18, primarily featuring symmetric/antisymmetric and compositional relation patterns. UMLS [18]: A collection of medical vocabularies and standards. For detailed statistical data, please refer to Appendix A.1.
Evaluation metrics: The JKDM is evaluated using link prediction task metrics, including Mean Reciprocal Rank (MRR) and Hits@N. MRR represents the average reciprocal rank of the correct predicted entity from the candidate entity set, while Hits@k indicates the proportion of correctly predicted entities within the top k ranks. This work sets k to (1, 3, 10).
Baselines: We evaluate JKDM against the latest knowledge completion models: GS-InGAT [8], SEA-KGC [9], FDM [10], KGDM [11], TDS [19], KRACL [20], DRR-GAT [21], MGTCA [22], LCA-KGC [23], SimKGC [24], CSProm-KG [25], FTL-LM [26], C-LMKE [27], BMKGC [28], HONARL [29], PEMLM-F [30], Relphormer [31].
Experimental details: The model was trained using a single A40 GPU, with some hyperparameters set as follows: the time step was set to 40, and the dimension was set to 400. The url is https://github.com/Irreproachability/JKDM.

Table 1. The experimental results on FB15k-237, WN18RR, and UMLS are as follows. The best results are highlighted in bold, the second-best results are underlined, and "-" indicates no result.

Model	FB15K-237				WN18RR				UMLS			
	MRR	H@10	H@3	H@1	MRR	H@10	H@3	H@1	MRR	H@10	H@3	H@1
TDN	.358	.561	.403	.273	.499	.579	.523	.455	.938	.997	**.983**	.891
KRACL	.360	.548	.395	.266	.527	.613	.547	.482	-	-	-	-
DRR-GAT	.361	.549	.415	.268	.468	.579	.508	.421	-	**1.00**	-	-
MGTCA	.393	.583	.428	.291	.511	.593	.525	.475	-	-	-	-
LCA-KGC	.372	.554	.407	.276	.492	.585	.510	.456	-	-	-	-
SimKGC	.336	.511	.362	.249	.666	.800	.717	.587	-	-	-	-
CSProm-KG	.358	.538	.393	.269	.575	.678	.596	.522	-	-	-	-
C-LMKE	.306	.484	.331	.218	.619	.789	.671	.523	-	-	-	-
CP-KGC	.329	.503	.353	.243	.648	.773	.683	.580	.780	.951	.857	.678
BMKGC	.332	.514	.365	. 247	.669	.807	.720	**.590**	-	-	-	-
SEA-KGC	.367	.553	.401	.275	.653	.795	.696	.577	-	-	-	-
GS-InGAT	.382	.567	.416	.283	.546	.625	.556	.491	-	-	-	-
HONARL	.367	.568	.406	.287	.513	.611	.541	.473	.907	.990	.951	.856
PEMLM-F	.355	.538	.389	.264	.556	.648	.573	.509	-	.997	-	-
Relphormer	.371	.481	-	.314	.401	.591	-	.448	-	.992	-	-
FDM	.485	.681	.529	.386	.506	.592	.518	.456	.922	.970	.944	.893
KGDM	.520	.708	.566	**.423**	.516	.593	.519	.457	.909	.973	.937	.872
Our (JKDM)	**.532**	**.786**	**.639**	.367	**.679**	**.892**	**.770**	.557	**.942**	.984	.967	**.913**

4.2 Performance Comparison

To demonstrate the effectiveness of JKDM, experiments were conducted on the WN18RR, FB15k-237, and UMLS datasets, and the performance was compared with existing models, as shown in Table 1.

Based on observations, JKDM outperforms other baseline models across the three datasets on most metrics. Specifically, on the FB15k-237, WN18RR, and UMLS datasets, the MRR scores improved by 1.2% (a 2.3% improvement relative to KGDM), 1% (a 1.5% improvement relative to BMKGC), and 0.4% (a 0.43% improvement relative to TDN), respectively, compared to the SOTA models.

Additionally, it is observed that JKDM significantly improves the performance on the Hits@10 and Hits@3 metrics, while the improvement on the Hits@1 metric is less pronounced. We analyze that JKDM, by leveraging structural and semantic enhancements to entity embeddings, captures more patterns and features of entities during the diffusion process. This enhanced generalization capability contributes to improved performance on Hits@3 and Hits@10. However, as the number of diffusion steps increases, noise may be introduced, or important information may be lost. This means that even if the initial embeddings contain rich information, this information may become blurred after multiple diffusion steps, affecting the final ranking accuracy (Hits@1).

Table 2. Ablation results on FB15k-237 and WN18RR. The best results are highlighted in bold, the second-best results are underlined.

Model	FB15K-237				WN18RR			
	MRR	H@10	H@3	H@1	MRR	H@10	H@3	H@1
JKDM w/o K	.298	.478	.327	.210	.457	.535	.470	.418
JKDM w/o D	.506	.764	.618	.352	.601	.824	.680	.481
JKDM w/o C	.497	.753	.616	.338	.628	.818	.699	.524
JKDM w/o C, D	.462	.687	.558	.325	.530	.706	.587	.438
JKDM w/o RGAT	.512	**.793**	**.651**	.346	.648	.868	.739	.525
JKDM w/o EGAT	.510	.770	.626	.357	.657	.880	.756	.547
JKDM w/o DG	<u>.519</u>	.779	.631	<u>.361</u>	<u>.662</u>	<u>.882</u>	<u>.762</u>	<u>.549</u>
JKDM w/o PE	.516	.780	.634	.358	.651	.860	.736	.537
JKDM	**.532**	<u>.786</u>	<u>.639</u>	**.367**	**.679**	**.892**	**.770**	**.557**

4.3 Ablation Experiment

Table 2 presents the ablation results of modules. "w/o C" and "w/o D" indicate the removal of the CPES and DGAT modules, respectively. It is observed that the MRR scores of "w/o C" and "w/o D" decreased by (4.9%, 6.8%) and (3.3%,10.8%), respectively. This demonstrates that the CPES and DGAT modules enhance entity features and improve model performance. By observing the decline ratios, it is noted that on the WN18RR dataset, the MRR metric of "w/o D" shows a more significant relative decline. In contrast, on the FB15k-237

dataset, the MRR metric of "w/o C" shows a more significant relative decline. We analyze that the WN18RR dataset is sparser (as shown in Table 1), making the structural feature enhancement by DGAT more significant for WN18RR. Therefore, the performance decline is more pronounced when the DGAT module is removed. We also removed CPES and DGAT ("w/o C, D"). It is observed that all metrics show a significant decline, with the MRR scores decreasing by 11.7% and 21.4%, respectively. This proves that combining structural and semantic features can effectively enhance entity embeddings. We conducted a case study to validate the effectiveness of the multi-model JKDM incorporating DGAT and CPES. For a detailed introduction, please refer to Appendix A.2.

Next, by removing the Knowledge Diffusion Generation (KDG) module ("w/o K"), it is observed that the MRR scores decrease by 43.0% and 32.2%, respectively. This indicates that using only KGE cannot effectively address the issue of diverse representations among entities. It demonstrates that diffusion models can effectively handle diverse entity representations and generate the distribution of target entities. Additionally, we conducted further ablation analysis. For details, please refer to Section 4.4.

4.4 Further analysis of ablation experiment

Explore the effectiveness of DGAT module: "w/o RGAT" indicates the removal of the RGAT channel. In FB15k-237, the metrics H@10 and H@3 increase while H@1 decreases. We analyze that since FB15k-237 is denser than WN18RR and contains more relational information, removing RGAT weakens the model's ability to capture fine-grained relational features, leading to a decline in the accuracy of precise matching. "w/o EGAT" indicates the removal of the EGAT channel, and all metrics show a decline. After removing EGAT, we analyze that sparse entities lack densified neighborhoods, resulting in lower utilization of neighborhood information and insufficient intrinsic interaction information between entities. To validate this point, we conducted a densification ablation experiment ("w/o DG"), and the results show a decline in all metrics. This result proves that densified graph structures can effectively supplement the structural information of sparse entities, enhancing the model's ability to utilize neighborhood information and thereby improving overall performance.

Additionally, to avoid performance improvements solely due to the stacking of RGAT and EGAT layers, we conducted hyperparameter experiments on the number of GAT layers, as shown in Fig. 4. The experiments demonstrate that, with the same number of layers, the model using DGAT consistently outperforms the "w/o EGAT" and "w/o RGAT" models. This result validates the effectiveness of DGAT in better capturing complex relationships in knowledge graphs. DGAT not only enhances the ability to model relationships but also supplements the neighborhood information of sparse entities.

Explore the effectiveness of CPES module: "w/o PE" indicates replacing entity contextual enhancement semantics with entity text semantics. The results show a decline in all metrics, proving the effectiveness of contextual paths in

enhancing entity semantics. Specifically, contextual paths contain structured relational information between entities, which helps the model better understand the semantics and associations of entities. In contrast, entity text semantics typically only include surface-level descriptions of entities and lack relational information between entities, resulting in weaker performance when modeling complex relationships. We conducted experiments on different PLM. For a detailed introduction, please refer to Appendix A.3.

Explore the impact of diffusion parameters on KDG: experiments were conducted on different time step parameters and the number of CDModules. In Fig. 4(3), we tested the performance under different time step parameters and observed that the model performs best when the time step parameter T=40. Increasing the time step may cause the model to overly rely on specific conditional feature details, reducing its robustness. In Fig. 4(4), we tested different numbers of CDModules and found that the performance is optimal when the number of layers is 3. Deeper networks may lead to overfitting, resulting in a decline in performance. We conducted experiments on different CGModules. For a detailed introduction, please refer to Appendix A.4.

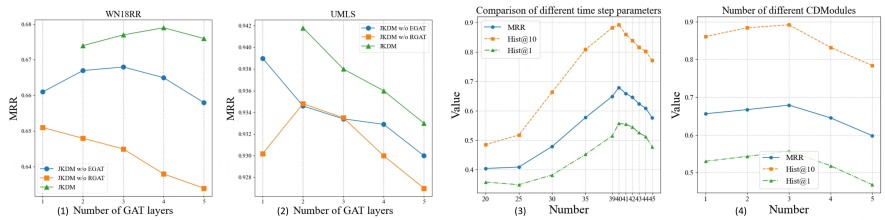

Fig. 4. (1) and (2) illustrate the performance comparison of different numbers of GAT layers in the channels of the DGAT. (3) and (4) represent the hyperparameter experiments conducted with varying time steps and numbers of CDModules.

4.5 Performance Evaluation By Relation Type

Next, we analyze the performance of JKDM in FB15k-237 by relation categories (Wang et al. 2014) [32]. Table 3 shows the MRR for different categories, and the results indicate that JKDM significantly improves performance when predicting N-1 tail entities and 1-N head entities. The experimental results suggest that the JKDM exhibits stronger robustness and generalization capabilities when handling complex relations, with its performance advantages being particularly pronounced when dealing with sparse entities. We conducted a case study to further demonstrate the JKDM's diverse generative capabilities. For a detailed introduction, please refer to Appendix A.5.

Table 3. MRR scores by relation type in FB15k-237, with the best results are highlighted in bold, the second-best results are underlined, and "-" indicates no result.

Model	Tail Pred				Head Pred			
	1-1	1-N	N-1	N-N	1-1	1-N	N-1	N-N
TransE	<u>.476</u>	.536	.060	.287	<u>.484</u>	.080	.329	.219
DisMult	.257	.575	.032	.184	.255	.038	.322	.131
ConvE	.366	.762	.069	.375	.374	.091	.444	.261
CompGCN	.453	<u>.779</u>	<u>.076</u>	<u>.395</u>	.457	<u>.112</u>	<u>.471</u>	<u>.275</u>
JKDM	**.522**	**.782**	**.424**	**.525**	**.523**	**.263**	**.623**	**.528**

4.6 Knowledge Sparsity Research

To further validate JKDM's sensitivity to sparse knowledge graphs, we randomly removed triples from the FB15k-237 training set at varying proportions (while maintaining the connectivity of the KG) and compared our experiments with TransE, RotatE, and w/o C, D, as shown in Fig. 5. As the entity degree increases, the performance of all models improves, with JKDM consistently outperforming the baseline models. This result demonstrates that the JKDM exhibits remarkable robustness when handling sparse entities, effectively addressing the challenges posed by incomplete information in the KG.

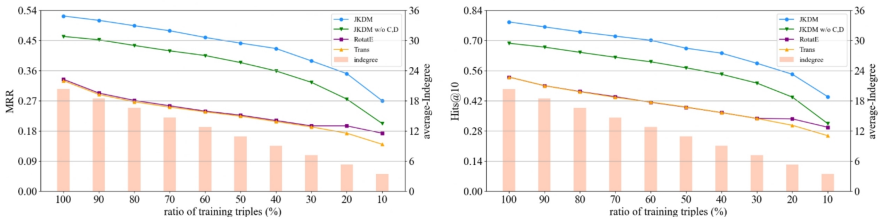

Fig. 5. Performance comparison between JKDM on sparse knowledge graphs and baseline models on the FB15k-237 dataset.

5 Conclusion

In this article, we propose the Joint Knowledge Diffusion Model (JKDM), which leverages diffusion models to capture diverse relationships between entities and generate the probability distribution of target entities. Additionally, we introduce a dual-channel GAT and a contextual path strategy to enhance the features of sparse entities, improving the generative capability of diffusion models for such entities. However, PLMs have limited understanding of entity contextual logic, while LLMs often increase the model's time cost. Therefore, in future work, we will explore knowledge distillation from LLMs to enhance PLMs' contextual logic comprehension while maintaining their time efficiency.

Acknowledgments. We thank Professor Song Yu and Professor Zhang Wendong for their valuable discussions and insights. We acknowledge the GPU equipment support provided by the Research Center of Xinjiang University. We also thank the anonymous reviewers for their constructive feedback, which has greatly improved this paper.

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: *Advances in Neural Information Processing Systems*, vol. 26. (2013)
2. Vashishth, S., Sanyal, S., Nitin, V., Talukdar, P.: Composition-based multi-relational graph convolutional networks. arXiv preprint arXiv:1911.03082 (2019)
3. Yao, L., Mao, C., Luo, Y.: KG-BERT: BERT for knowledge graph completion. arXiv preprint arXiv:1909.03193 (2019).
4. Sun, Z., Deng, Z. H., Nie, J. Y., et al.: RotatE: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)
5. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32(1) (2018)
6. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29(1) (2015)
7. Sun, Z., Deng, Z.-H., Nie, J.-Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)
8. Yin, H., Zhong, J., Wang, C., Li, R., Li, X.: GS-InGAT: An interaction graph attention network with global semantic for knowledge graph completion. *Expert Systems with Applications* **228**, 120380 (2023)
9. Je, S.-H., Choi, W., Oh, K.: Unifying structure and language semantic for efficient contrastive knowledge graph completion with structured entity anchors. arXiv preprint arXiv:2311.04250 (2023)
10. Long, X., Zhuang, L., Li, A., et al.: Fact embedding through diffusion model for knowledge graph completion. In: *Proceedings of the ACM Web Conference 2024*, pp. 2020–2029 (2024)
11. Nguyen, T.-K., Fang, Y.: Diffusion-based negative sampling on graphs for link prediction. In: *Proceedings of the ACM Web Conference 2024*, pp. 948–958 (2024)
12. Cai, Y., Liu, Q., Gan, Y., Li, C., Liu, X., Lin, R., Luo, D., JiayeYang, J.: Predicting the unpredictable: Uncertainty-aware reasoning over temporal knowledge graphs via diffusion process. In: *Findings of the Association for Computational Linguistics ACL 2024*, pp. 5766–5778 (2024)
13. Jiang, Y., Yang, Y., Xia, L., Huang, C.: Diffkg: Knowledge graph diffusion model for recommendation. In: *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 313–321 (2024)
14. Pan, Y., Liu, J., Zhao, T., Zhang, L., Wang, Q.: Context-aware commonsense knowledge graph reasoning with path-guided explanations. *IEEE Transactions on Knowledge and Data Engineering* **36**(8), 3725–3738 (2024)
15. Gunel, B., Du, J., Conneau, A., Stoyanov, V.: Supervised contrastive learning for pre-trained language model fine-tuning. arXiv preprint arXiv:2011.01403 (2020)

16. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: *Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality*, pp. 57–66 (2015)
17. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32(1) (2018)
18. Kok, S., Domingos, P.: Statistical predicate invention. In: *Proceedings of the 24th International Conference on Machine Learning*, pp. 433–440 (2007)
19. Wang, J., Wang, B., Gao, J., Li, X., Hu, Y., Yin, B.: TDN: Triplet distributor network for knowledge graph completion. *IEEE Transactions on Knowledge and Data Engineering* **35**(12), 13002–13014 (2023)
20. Tan, Z., Chen, Z., Feng, S., Zhang, Q., Zheng, Q., Li, J., Luo, M.: KRACL: Contrastive learning with graph context modeling for sparse knowledge graph completion. In: *Proceedings of the ACM Web Conference 2023*, pp. 2548–2559 (2023)
21. Xin, Z., Zhang, C., Guo, J., Peng, C., Niu, Z., Wu, X.: Graph attention network with dynamic representation of relations for knowledge graph completion. *Expert Systems with Applications* **219**, 129684 (2023)
22. Shang, B., Zhao, Y., Liu, J., Wang, D.: Mixed geometry message and trainable convolutional attention network for knowledge graph completion. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38(8), pp. 8966–8974 (2024)
23. Shang, B., Zhao, Y., Liu, J.: Learnable convolutional attention network for knowledge graph completion. *Knowledge-Based Systems* **285**, 111360 (2024)
24. Wang, L., Zhao, W., Wei, Z., Liu, J.: Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. arXiv preprint arXiv:2203.02167 (2022)
25. Chen, C., Wang, Y., Sun, A., Li, B., Lam, K.: Dipping plms sauce: Bridging structure and text for effective knowledge graph completion via conditional soft prompting. arXiv preprint arXiv:2307.01709 (2023)
26. Lin, Q., Mao, R., Liu, J., Xu, F., Cambria, E.: Fusing topology contexts and logical rules in language models for knowledge graph completion. *Information Fusion*, pp. 253–264 (2023)
27. Wang, X., He, Q., Liang, J., Xiao, Y.: Language models as knowledge embeddings. arXiv preprint arXiv:2206.12617 (2022)
28. Kong, Y., Fan, C., Chen, Y., Zhang, S., Lv, Z., Tao, J.: Bilateral masking with prompt for knowledge graph completion. In: *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 240–249 (2024)
29. Yin, H., Zhong, J., Li, R., Shang, J., Wang, C., Li, X.: High-order neighbors aware representation learning for knowledge graph completion. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1234–1245 (2024)
30. Qiu, C., Qian, P., Wang, C., Yao, J., Liu, L., Wei, F., Eddie, E.: Joint pre-encoding representation and structure embedding for efficient and low-resource knowledge graph completion. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15257–15269 (2024)
31. Bi, Z., Cheng, S., Chen, J., Liang, X., Xiong, F., Zhang, N.: Relphormer: Relational graph transformer for knowledge graph representations. *Neurocomputing* **566**, 127044 (2024)
32. Wang, Z., et al.: Knowledge graph embedding by translating on hyperplanes. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28(1), pp. 613–619 (2014)