Highlights

Multi-Condition Latent Diffusion Network for Semantic-Aware Knowledge Graph Completion

Siyuan Li,Liang Zhang,Bo Jin,Xiaopeng Wei

- Link prediction can be modeled as a multi-condition entity generative task.
- Semantic context is a powerful source of guidance for entity generation.
- Filtering noisy neighbors significantly purifies the contextual guidance.
- Adapting this guidance for each fact is crucial for model performance.
- Sets strong performance records on major knowledge graph benchmarks.

Multi-Condition Latent Diffusion Network for Semantic-Aware Knowledge Graph Completion

Siyuan Li[a], Liang Zhang[a,c,*], Bo Jin[b,c] and Xiaopeng Wei[a,c]

[a]School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China
[b]School of Innovation and Entrepreneurship, Dalian University of Technology, Dalian 116024, China
[c]Key Laboratory of Social Computing and Cognitive Intelligence, Ministry of Education, Dalian 116024, China

ARTICLE INFO

Keywords:
Knowledge Representation
Knowledge Reasoning
Knowledge Graphs
Diffusion Process

ABSTRACT

Considering the rich and heterogeneous structured information of entities and relations, capturing diverse semantics is essential for accurately predicting missing facts in knowledge graph completion tasks. However, most existing approaches remain limited to learning connection patterns that are agnostic to semantic context. Although the performance of recent conditional diffusion-based methods has improved by modeling the probabilistic distribution of target entities, the critical role of semantic guidance is often overlooked. To address this limitation, we propose a Multi-Condition Latent Diffusion Network (MCDN), which formulates entity prediction as a multi-condition joint generation problem. Specifically, the MCDN incorporates a semantic context condition learning module and a connection pattern condition learning module to capture the semantic and structural contexts of each triplet. A specially designed multi-condition denoiser is further introduced to learn the reverse diffusion process, enabling the generation of target entity embeddings guided by these multiple contextual signals. Extensive experiments demonstrated that the MCDN achieved superior performance across several benchmarks. On the widely used FB15k-237 dataset, the MCDN achieved a competitive MRR of 55.1% and a Hits@1 of 52.9%. It also delivered robust results on WN18RR (MRR 57.4%) and Kinship (Hits@1 79.8%).

1. Introduction

Knowledge graphs (KGs), structured collections of objective information, are used to represent real-world entities and their relationships [Shen et al., 2022, Li et al., 2022a,b, Dong et al., 2023]. They are typically stored as triplets (h, r, t), where h and t represent the head and tail entities, respectively, and r denotes the relation between them. Despite their widespread adoption, real-world KGs are characterized by their vast scale and inherent incompleteness [Wang et al., 2017, Ji et al., 2021, Zhang et al., 2023a], which poses limitations for their application in downstream tasks. Consequently, automatic Knowledge Graph Completion (KGC) has attracted considerable attention within the KG community [Chen et al., 2020, Ji et al., 2021, Zhang et al., 2024a].

A key challenge in knowledge graph completion is that the answer distribution for a query $(h, r, ?)$ is often *multi-peaked*: multiple tail entities can be plausible and supported by different pieces of evidence [Wang et al., 2019, Long et al., 2024a]. This complexity is further compounded by the heterogeneity and semantic richness of real-world relations, where disambiguating among candidates requires signals beyond the local triplet [Wang et al., 2021, Zhang et al., 2023b]. Multi-hop neighborhoods around entities and relations offer such contextual evidence [Zhang et al., 2024b], but they are also noisy: indiscriminate aggregation may propagate irrelevant connections and amplify spurious correlations,

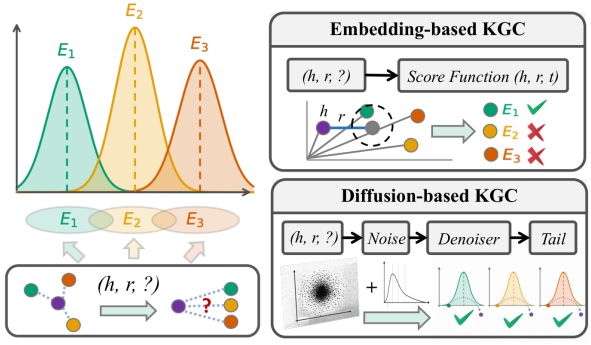


Figure 1: Two paradigms address multi-peaked answers in KGs. Diffusion-based methods, by leveraging the uncertainty of conditional denoising, naturally capture multi-modal distributions.

while overly rigid feature extraction may miss subtle semantic cues [Li et al., 2025a]. Meanwhile, relational connection patterns provide complementary structural priors that can further constrain plausible answers [Pavlović and Sallinger, 2023]. However, *how to jointly leverage semantic context and connection-pattern priors in a unified and query-adaptive manner* remains non-trivial.

Taken together, when the answer set is multi-peaked and evidence is distributed across noisy multi-hop context, collapsing prediction into a single deterministic score can be insufficient [Chen et al., 2024]. Instead, it is desirable to explicitly learn $p(t \mid h, r)$ and refine it under multiple sources of evidence. *Diffusion models* [Long et al., 2024b] provide a practical mechanism for such conditional distribution learning via iterative denoising. As shown in Figure 1, they can represent multi-peaked targets and support progressive refinement, but their success on KGs hinges on

*Corresponding author
liangzhang@dlut.edu.cn (L. Zhang)
ORCID(s): 0000-0002-5055-1527 (L. Zhang)

the conditioning design. Effective condition guidance in this setting should satisfy three requirements: (i) *explicitly capture semantic context* from both entity-side neighborhoods and relation-side evidence; (ii) *filter noisy neighborhoods* to retain only query-relevant contextual signals; and (iii) *adaptively integrate heterogeneous conditions*, since different queries may rely on different mixtures of semantic and structural evidence.

To address the above challenges, we propose a *Multi-Condition Latent Diffusion Network (MCDN)*, a unified latent diffusion framework for query-aware entity prediction. Instead of relying solely on deterministic scoring functions, MCDN models entity inference as a multi-condition generation process in latent space, where semantic contextual evidence and probabilistic connection patterns jointly guide representation refinement. By performing diffusion in a compact latent space, the framework ensures stable and efficient optimization. Through query-specific semantic conditioning and structural pattern modeling, MCDN enables adaptive contextual reasoning while mitigating neighborhood noise. The integrated multi-condition denoising process allows target entity embeddings to be progressively recovered under coordinated semantic and structural guidance. Our contributions are summarized as follows:

- We propose MCDN, a latent diffusion framework that models entity prediction as multi-condition conditional generation guided by semantic context and connection-pattern priors.
- We introduce a query-specific semantic conditioning mechanism that selectively aggregates entity–relation contextual evidence via a semantics-aware Top-K strategy, improving robustness against noisy neighborhood information.
- We propose a condition-adaptive denoising paradigm that dynamically integrates heterogeneous semantic and structural priors, allowing the reverse diffusion process to perform query-adaptive entity inference across diverse relational scenarios.
- Extensive experiments on both transductive and inductive benchmarks demonstrate that MCDN consistently improves prediction accuracy and exhibits stronger resilience in recovering reliable entity embeddings under noisy and sparse graph conditions.

2. Related Work

Knowledge Graph Completion Methods. Knowledge graph completion (KGC) aims to infer missing facts in incomplete KGs by predicting the missing entity in a query triplet $(h, r, ?)$ (or $(?, r, t)$) [Chen et al., 2020, Ji et al., 2021, Zhang et al., 2023a]. Existing KGC methods can be broadly categorized by how they represent triplets and exploit relational regularities [Zhang et al., 2024a].

Score-based embedding methods constitute a dominant paradigm, where entities and relations are embedded into continuous spaces and a scoring function is learned to rank candidate triplets [Bordes et al., 2013, Yang et al., 2015, Lin et al., 2015, Trouillon et al., 2016]. Representative models include translational approaches such as TransE [Bordes et al., 2013] and bilinear/complex-valued models such as ComplEx [Trouillon et al., 2016]. To capture richer relational patterns (e.g., symmetry, antisymmetry, inversion, and composition), subsequent work explores more expressive geometries and region-based representations, including RotatE [Sun et al., 2019] and Rot-Pro [Song et al., 2021] in complex space, HAKE [Zhang et al., 2020] for hierarchical structure, BoxE [Abboud et al., 2020] for logical constraints, and ExpressivE [Pavlović and Sallinger, 2023] for joint hierarchy and composition. Beyond point embeddings, distributional KGC models represent entities as probability distributions to account for uncertainty and multiplicity in relational facts [Wang et al., 2019]. Recent extensions further generalize beyond binary triplets, such as knowledge hypergraph embedding for higher-order relations (e.g., HyCubE [Li et al., 2025b]).

Beyond embedding-centric scoring, *generative formulations* have recently been explored for KGC [Liu et al., 2025]. In particular, diffusion-based methods (e.g., FDM [Long et al., 2024a] and KGDM [Long et al., 2024b]) model the probabilistic distribution of target entities via iterative denoising, offering a distribution-learning alternative to deterministic ranking. However, existing generative KGC methods are commonly conditioned primarily on the local query triplet, which limits their ability to leverage broader multi-hop semantic context. Complementary to model-centric advances, *feature-engineering and data-centric* strategies have shown promise: Intersection Features [Le et al., 2024] demonstrate strong performance by constructing neighbor-intersection features, while LLM-assisted methods such as KERMIT [Li et al., 2025c] enrich textual descriptions to alleviate data sparsity and inconsistency.

Despite substantial progress, a key limitation remains that many approaches either (i) rely mainly on local triplet scoring, or (ii) incorporate broader context through handcrafted features or weak condition signals, making it challenging to jointly exploit rich semantic context and structural priors in a unified, query-adaptive prediction mechanism.

Diffusion Models. Diffusion models are generative models that employ diffusion processes. Ho et al. [Ho et al., 2020] and Sohl-Dickstein et al. [Sohl-Dickstein et al., 2015] were among the pioneers who proposed defining data sampling as a gradual denoising process from pure Gaussian noise. Recently, these diffusion-based generative models demonstrated remarkable performance in modeling high-dimensional, multi-modal distributions. They also exhibited the ability to generate high-quality and diverse samples across various benchmarks, particularly in computer vision Guo et al. [2025], as showcased by Dhariwal and Nichol [Dhariwal and Nichol, 2021] and Rombach et al. [Rombach et al., 2022]. However, their iterative sampling process is often computationally intensive, which has motivated research into more efficient generative frameworks. For instance, Flow Matching [Lipman et al., 2023] has emerged

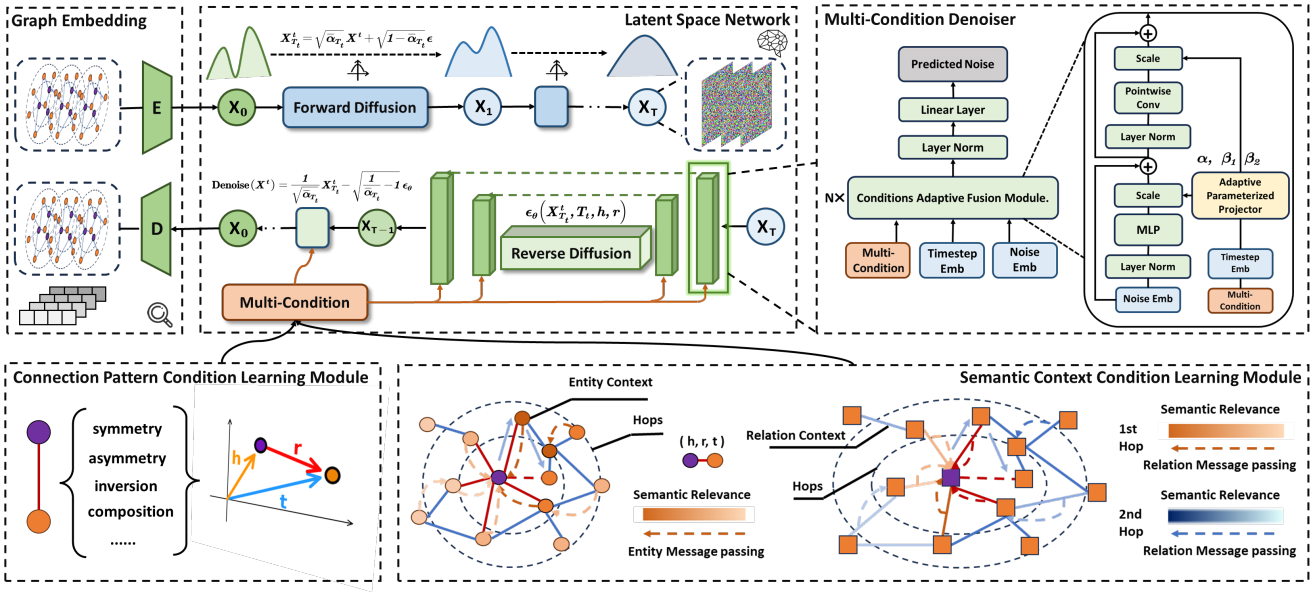


Figure 2: The architecture of the Multi-Condition Latent Diffusion Network (MCDN). The system comprises: (1) a Graph Embedding Module (Encoder E, Decoder D); (2) a Latent Space Network performing forward and reverse diffusion ($X_0 \to X_T$ and $X_T \to X_0$); and (3) a Multi-Condition Denoiser. The reverse diffusion and denoiser are guided by a **Multi-Condition** input. This input is synthesized from a **Connection Pattern Condition Learning Module** and a **Semantic Context Condition Learning Module**. Notably, the Semantic Context Condition incorporates both Entity Context (circular nodes) and Relation Context. In the Relation Context, nodes (square shapes) do not consider their own intrinsic node messages; rather, their role is to facilitate the propagation and aggregation of messages associated with their incident edges, which represent the relation semantics.

as a simulation-free method for training continuous normalizing flows, offering a more direct and stable training objective. Concurrently, Consistency Models [Song et al., 2023] have been developed to enable fast, single-step generation by distilling the knowledge of a pre-trained diffusion model.

The success in continuous domains has spurred efforts to extend these models to discrete data. Austin et al. [Austin et al., 2021] proposed investigating diffusion on discrete state spaces by defining a corruption process for discrete data. Some studies, for instance, Li et al. [Li et al., 2022c] and Gong et al. [Gong et al., 2023], explored continuous diffusion models for text generation, proposing the use of neural backbones like U-Nets [Dhariwal and Nichol, 2021, Ho et al., 2020] or Transformers [Li et al., 2022c, Gong et al., 2023, Vaswani, 2017] to parameterize the conditional distribution $p(x_{t-1}|x_t)$. However, knowledge graphs, stored as triplets (h, r, t), have a fundamentally different structure than images or text, with shorter lengths and reduced long-range dependencies. Therefore, we introduce an MLP-based Multi-Condition Denoiser, specifically tailored to learn the reverse diffusion process.

3. Methodology

3.1. Preliminaries

A knowledge graph is in the form of $K = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$, where $\mathcal{V}, \mathcal{E}, \mathcal{F} = \{(h, r, t) \mid (h, t \in \mathcal{V}, r \in \mathcal{E})\}$ are the sets of entities, relations, and fact triplet, respectively. Let h be the query entity, r be the query relation, and t be the answer entity. Given the query $(h, r, ?)$, the completion task is to predict the missing answer entity t. Generally, all the entities in $\mathcal{V}$ are considered candidates for t. Our goal is to model the distribution of the tail entity: $p(t|h, r)$ [Wang et al., 2021]. This is equivalent to modeling the following term based on Bayes' theorem:

$$p(t|h, r) \propto p(h, r|t) \cdot p(t), \tag{1}$$

where $p(t)$ is the prior distribution of the tail entity. Then, the first term can be further decomposed as:

$$p(h, r|t) = \frac{1}{2}\left[p(h|t) \cdot p(r|h, t) + p(r|t) \cdot p(h|r, t)\right], \tag{2}$$

which guides our model design. The terms $p(h|t)$ or $p(r|t)$ measure the likelihood of a tail entity t given a specific (h, r) pair. Since our model needs to capture the rich semantic information between entities and relations, we utilize neighborhood structures to represent entities and relations themselves, i.e., $p(C^{\mathcal{V}}(h)|t)$ and $p(C^{\mathcal{E}}(r)|t)$, where $C^{\mathcal{V}}(\cdot)$ represents the neighboring entity subgraph, also known as the entity context, and $C^{\mathcal{E}}(\cdot)$ represents the local relation subgraph, also known as the relation context. The terms $p(r|h, t)$ or $p(h|r, t)$ measure how likely h is linked to t through r. In the following sections, we detail how these three factors are modeled within our approach and integrated as multiple conditions within the MCDN framework.

Notation. For clarity, we summarize the key symbols and notations used throughout our methodology in Table 1.

3.2. Multi-Condition Latent Diffusion Network

Figure 2 illustrates the architecture of the Multi-Condition Latent Diffusion Network (MCDN). It consists of two stages:

Table 1
Main notation in MCDN.

Symbol	Meaning
(h, r, t)	A factual triple; given $(h, r, ?)$, predict the tail entity t.
$\text{EMB}^v, \text{EMB}^e$	Embedding for entities/relations.
X^v, X^e	Generic entity/relation embeddings: $X^v = \text{EMB}^v(v)$, $X^e = \text{EMB}^e(e)$.
X^h, X^r, X_0^t	Clean embeddings: $X^h = \text{EMB}^v(h)$, $X^r = \text{EMB}^e(r)$, $X_0^t = \text{EMB}^v(t)$.
τ, T	Diffusion timestep $\tau \in \{1, \dots, T\}$ and total steps T.
X_τ^t	Noised tail embedding at timestep τ.
$\beta_\tau, \alpha_\tau, \bar{\alpha}_\tau$	Noise schedule: $\alpha_\tau = 1 - \beta_\tau$, $\bar{\alpha}_\tau = \prod_{s=1}^{\tau} \alpha_s$.
X_τ^T	Timestep embedding: $X_\tau^T = \text{EMB}^T(\tau)$.
$\epsilon, \epsilon_\theta(\cdot)$	Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$; denoiser ϵ_θ predicts noise.
$\hat{X}_0^t$	Reconstructed clean tail embedding.
$\mathbf{S}^c = \{s^v; s^e\}$	Semantic-context condition: entity context s^v and relation context s^e.
$\mathbf{X}^c$	Connection-pattern condition derived from (X^h, X^r).
$\mathcal{N}(v)$, $\mathcal{N}'_K(v)$	Neighbors of node v and the selected Top-K neighbors.
$\mathcal{E}(v)$, $\mathcal{E}'_K(v)$	Incident edges of node v and the selected Top-K incident edges.
X_i^v, X_i^e	Node/edge representations at iteration i (with $X_0^v = X^v$, $X_0^e = X^e$).
m_i^v, m_i^e	Aggregated messages for node v and edge e at iteration i.
λ	Temperature in Top-K semantic scoring.

a forward diffusion process and a reverse denoising process conditioned on multiple factors. The objective is to realize a Markovian transition from the Gaussian distribution to the target entity distribution in the latent embedding space. Next, we provide a detailed explanation of these two processes.

3.2.1. Forward Process

To apply continuous diffusion models to discrete triplet, we introduce two learnable embedding functions: EMB^v and EMB^e. These embedding functions map high-dimensional identifiers of entities and relations into a lower-dimensional latent vector space, represented as $X^v \in \mathbb{R}^{d_v}$ and $X^e \in \mathbb{R}^{d_e}$. During training, for a given triple (h, r, t), under the multiple conditions where the semantic context and connection patterns corresponding to the head entity h and relation r are considered, the distribution of the target entity $q(X^t \mid h, r)$ in the vector space is assumed to be unknown. Initially, we define the forward diffusion Markov chain $\{X_\tau^t\}_{\tau=0}^T$ starting from the clean tail embedding $X_0^t = \text{EMB}^v(t)$, and progressively add Gaussian noise until $\tau = T$. This process gradually transforms the embedding vector of the tail entity into pure noise. The transition of the forward process is parameterized by the following equation:

$$q(X_\tau^t \mid X_{\tau-1}^t) = \mathcal{N}\left(X_\tau^t; \sqrt{1-\beta_\tau}\, X_{\tau-1}^t, \beta_\tau I\right), \quad (3)$$

where $\{\beta_\tau\}_{\tau=1}^T$ denotes the variance schedule of the forward process. By progressively adding noise in this manner, the model effectively diffuses the embedding vector of the target entity into a pure noise distribution, thereby establishing a foundation for the subsequent reverse multi-conditional generation process.

3.2.2. Reverse Process with Multi-Condition Denoising

To perform iterative denoising from pure Gaussian noise to generate target entities in latent vector space, we define the reverse diffusion process $p_\theta(X_{\tau-1}^t \mid X_\tau^t, h, r)$ which is conditioned on both the semantic context and connection patterns, represented by (h, r). Formally, the process is defined as follows:

$$p_\theta(X_{\tau-1}^t \mid X_\tau^t, h, r) = \mathcal{N}\left(X_{\tau-1}^t; \mu_\theta(X_\tau^t, \tau, h, r), \sigma_\tau^2 I\right), \quad (4)$$

where σ_τ is a fixed variance, μ_θ is predicted by a neural network, and θ denotes the denoiser parameters. We reparameterize the mean to enable the neural network to directly learn the noise added at time step τ. This reparameterization is expressed as:

$$\mu_\theta(X_\tau^t, \tau, h, r) = \frac{1}{\sqrt{\alpha_\tau}}\left(X_\tau^t - \frac{\beta_\tau}{\sqrt{1-\bar{\alpha}_\tau}}\epsilon_\theta(X_\tau^t, \tau, h, r)\right), \quad (5)$$

where $\alpha_\tau = 1 - \beta_\tau$, $\bar{\alpha}_\tau = \prod_{s=1}^{\tau} \alpha_s$, and $\epsilon_\theta(\cdot)$ (MCDenoiser) predicts the injected noise conditioned on (h, r) and the current timestep τ. Next, we provide a detailed introduction to MCDenoiser in the following section.

Multi-Condition Denoiser(MCDenoiser). In the reverse diffusion process, the key to achieving a Markovian transition from the Gaussian distribution to the target entity distribution lies in designing an effective denoising model. Considering that knowledge graphs are usually stored in the form of fact triplet (h, r, t), which have short lengths and fewer long-range dependencies, and that previous research such as FDM [Long et al., 2024a] has confirmed the efficiency of MLP as the denoising backbone network, rather than the traditional Transformer[1]. To capture rich semantic and structural information between entities and relations, we develop multiple conditions: **(1) semantic context**; **(2) connection patterns**, and dynamically fuse them to design an efficient MLP-based denoiser (MCDenoiser). The specific implementation of this architecture can be represented in the following four steps:

First, we extract the semantic condition $\mathbf{S}^c$ from the neighborhoods of h and r:

$$\mathbf{S}^c = \{s^v; s^e\} = \mathit{SemanticContext}(h, r). \quad (6)$$

[1] We provide a detailed comparison between MLP and Transformer in Section 4.3.1.

Second, we capture the connection pattern $\mathbf{X}^c$ from the embeddings X^h and X^r:

$$\mathbf{X}^c = ConnectionPattern(X^h, X^r). \tag{7}$$

Third, the noised entity $X^t_{T_t}$, timestep $X^T_{T_t}$, and conditions $\mathbf{S}^c, \mathbf{X}^c$ are fused into a unified representation vector E:

$$E = ConditionsFusion(X^t_\tau, X^T_\tau, \mathbf{S}^c, \mathbf{X}^c). \tag{8}$$

Finally, the noise ϵ is predicted from the fused feature E via a final projection layer:

$$\epsilon = LinearLayer(LN(E)). \tag{9}$$

In the following, we provide a detailed description of the architectural design for each of these modules.

Semantic Context Condition Learning Module. In knowledge graphs, the semantic context surrounding a given triplet (h, r, t) provides crucial cues for identifying valid links from the head entity h to the tail entity t [Wang et al., 2021]. Inspired by the remarkable success of Graph Neural Networks (GNNs) in representation learning [Kipf and Welling, 2016, Hamilton et al., 2017, Yun et al., 2019], we devise a message-passing mechanism to explicitly model two complementary types of semantic contexts: entity context and relation context, as illustrated in Figure 2.

Entity Context Modeling. We employ the classic Message Passing Neural Network (MPNN) framework [Gilmer et al., 2017] to aggregate node information, thereby constructing the entity context. Each node v is endowed with an initial embedding $X^v = \text{EMB}^v(v)$ and updates its hidden state iteratively. To mitigate issues such as noise, information dilution, or over-smoothing that can arise from naively aggregating all neighboring nodes, we introduce a **semantics-aware Top-K mechanism**. Specifically, when selecting information sources for a target node v (anchored by its initial representation X^v), we assess the semantic relevance between v and each of its neighboring nodes u (represented by its hidden state X^u_i at iteration i). This is achieved via a learnable mapping function $g(\cdot)$ that projects X^v and X^u_i into a shared latent semantic space. Within this space, inspired by energy-based models [Cao et al., 2024], we compute their similarity score using:

$$Score(v, u) = \exp(-\|g(X^v) - g(X^u_i)\|^2/\lambda), \tag{10}$$

where $\|\cdot\|^2$ denotes the squared Euclidean distance, and λ is a temperature hyperparameter. The K neighboring nodes with the highest scores are considered most semantically relevant to node v and form the selected neighborhood $\mathcal{N}'_K(v) \subseteq \mathcal{N}(v)$.

Subsequently, message passing is performed over multiple iterations on the graph. At each iteration i, the hidden state X^v_i of node v is updated as follows:

$$m^v_i = Aggregator_1\left(\{X^u_i\}_{u\in\mathcal{N}'_K(v)}\right), \tag{11}$$

$$X^v_{i+1} = \sigma\left([m^v_i, X^v_i]\cdot W^i_v + b^i_v\right), \tag{12}$$

where $X^v_0 = X^v$ is the initial hidden state, m^v_i is the aggregated message received by node v from its Top-K neighbors, $Aggregator_1(\cdot)$ is an entity message mean aggregation function. $[\cdot]$ represents vector concatenation, W^i_v and b^i_v are learnable transformation matrix and bias at iteration i, and $\sigma(\cdot)$ is a non-linear activation function. The resulting X^v_{i+1} constitutes the entity context representation with respect to the query (h, r). During training, the direct link from h to t must be treated as unobserved (i.e., masked) when predicting the tail entity t, to ensure the model learns to rely on contextual information rather than the target itself.

Relation Context Modeling. Given that edge features (i.e., relation embeddings) in KGs also carry significant information, and the number of relation types is typically much smaller than the number of entities, we posit that message passing at the relation level can more efficiently capture indicative contextual cues. To further enhance the model's expressive power, we also integrate a **semantics-aware Top-K mechanism** into the relation message passing. We adapt and enhance the alternating relation message passing scheme from PathCon [Wang et al., 2021]. Specifically, when selecting relevant incident edges e (represented by its current state X^e_i at iteration i) for a node v, the semantic relevance score is defined as:

$$Score(v, e) = \exp(-\|g(X^v) - g(X^e_i)\|^2/\lambda), \tag{13}$$

where the functions of $g(\cdot)$ and λ are consistent with Eq. (10), and they are projected into the same latent semantic space. The K edges with the highest scores are selected to form the set $\mathcal{E}'_K(v) \subseteq \mathcal{E}(v)$, where $\mathcal{E}(v)$ denotes the set of edges incident to node v. Then, these selected Top-K edges, $\{X^e_i\}_{e\in\mathcal{E}'_K(v)}$, are aggregated via a mean aggregator $Aggregator_2$ to form the relation-context message m^v_i for node v:

$$m^v_i = Aggregator_2\left(\{X^e_i\}_{e\in\mathcal{E}'_K(v)}\right). \tag{14}$$

Subsequently, the message m^e_i for each edge $e = (v', u')$ is generated from the relation-context messages $m^{v'}_i$ and $m^{u'}_i$ received by its two endpoint nodes v' and u'. This is accomplished using a MLP-based aggregator $Aggregator_3$:

$$m^e_i = Aggregator_3\left([m^{v'}_i, m^{u'}_i]\right). \tag{15}$$

Finally, the hidden state X^e_i of edge e is updated as:

$$X^e_{i+1} = \sigma\left([X^e_i, m^e_i]\cdot W^i_e + b^i_e\right), \tag{16}$$

where $X^e_0 = X^e = \text{EMB}^e(e)$ is the initial feature vector for the edge, and W^i_e, b^i_e are learnable parameters for this step.

Through this dual-context modeling approach, combined with a semantics-aware Top-K selection mechanism, our model effectively leverages both the structural and feature information of knowledge graphs (KGs), while more accurately capturing and propagating the most relevant contextual information for the specific link prediction task $(h, r, ?)$.

Connection Pattern Condition Learning Module. For triplets in KGs, entities and relations often exhibit rich connection patterns and are not independent of each other. Drawing inspiration from TransE and its successors [Bordes et al., 2013, Yang et al., 2015, Sun et al., 2019] that a linking function should hold between the tail entity, the head entity, and the relation: $t = f_r(h, r)$. Thus, we utilize this functional idea to learn different patterns, which allows for better guidance of the generation process. The Connection Pattern Condition Learning Module can be represented as:

$$\mathit{ConnectionPattern}(h, r) = X^h \oplus X^r. \tag{17}$$

Following [Long et al., 2024b], we define $\oplus$ as a dataset-specific composition operator. In our implementation, we use element-wise addition on FB15k-237, i.e., $\mathbf{X}_c = \mathbf{X}_h + \mathbf{X}_r$. For WN18RR, Kinship, and UMLS, we use the Hadamard product, i.e., $\mathbf{X}_c = \mathbf{X}_h \odot \mathbf{X}_r$. This choice is fixed per dataset and selected based on validation performance.

Conditions Adaptive Fusion Module. Building on the effectiveness of Transformer frameworks [Vaswani, 2017] in the graph data domain [Hu et al., 2020], our module adopts an architecture that mirrors that of the DiT Block with adaLN-Zero [Peebles and Xie, 2023]. Given the inherently concise nature of triplet and the paucity of evident long-range dependencies, we replace multi-head self-attention layers with alternating MLP layers. LayerNorm is applied as a pre-processing step to each layer, and residual connections are implemented around each sub-layer [He et al., 2016]. Furthermore, rather than learning dimension-wise scale α and shift β parameters directly, we instead learn these parameters using **Adaptive Parameterized Projector**, separately concatenating the embedding vector X_τ^T corresponding to time step τ with the embedding vectors of multi-conditions $\mathbf{S}^c$ and $\mathbf{X}^c$ respectively, and then feeding them into the projector. The heterogeneity in connection patterns $\mathbf{X}^c$ permits the learning of distinct α values, and variations in entity contexts s^v and relation contexts s^e allow for the discrimination of β_1 and β_2. The parameter automation process is detailed as follows:

$$\alpha^{(\ell)} = \mathcal{F}_\Omega([\mathbf{X}^c, X_\tau^T]); \quad \{\beta_1^{(\ell)}, \beta_2^{(\ell)}\} = \mathcal{F}_\Omega([\mathbf{S}^c, X_\tau^T]), \tag{18}$$

where $\mathcal{F}_\Omega$ is a fully-connected layer with parameters Ω and $[\cdot]$ represents the concatenation operation. The Adaptive Parameterized Projector enables MCDN to adapt to heterogeneous graphs with various graph heterogeneity settings. Finally, we incorporate these parameters into the noise embedding X_τ^t by scaling them as:

$$X_\tau^t = \alpha^{(\ell)} \cdot X_\tau^t + \beta_1^{(\ell)} + \beta_2^{(\ell)}. \tag{19}$$

This modulation mechanism allows the model to finely guide the update of $X_{T_t}^t$ based on the specific connection patterns, semantic context of the input triplet, and the current timestep T_t, thereby enhancing the accuracy of target generation.

Algorithm 1 MCDenoiser $\epsilon_\theta(X_\tau^t, \tau, h, r)$

Input: Noisy tail embedding X_τ^t, timestep τ, head entity h, relation r;
Parameters: EMB^T, EMB^v, EMB^e, ϵ_θ;
Output: Predicted noise ϵ_{pred};
Calculate timestep, head and relation embeddings:
  $X_\tau^T = EMB^T(\tau)$;
  $X^h = EMB^v(h)$;
  $X^r = EMB^e(r)$;
Learn multi-condition features:
  $\mathbf{S}^c = \mathit{SemanticContext}(h, r)$;
  $\mathbf{X}^c = X^h \oplus X^r$;
Conditions Adaptive Fusion Module:
  Learn modulation parameters via Adaptive Parameterized Projector:
    $\alpha = \mathcal{F}_\Omega([\mathbf{X}^c, X_\tau^T])$;
    $\{\beta_1, \beta_2\} = \mathcal{F}_\Omega([\mathbf{S}^c, X_\tau^T])$;
  Modulate the noisy input:
    $E = X_\tau^t + \alpha \cdot \text{MLP}(\text{LayerNorm}(X_\tau^t)) + \beta_1 + \beta_2$;
    $E = E + \alpha \cdot \text{Pointwise}(\text{LayerNorm}(E)) + \beta_1 + \beta_2$;
Final projection to predict noise:
  $\epsilon_{\text{pred}} = \mathit{LinearLayer}(\text{LayerNorm}(E))$;
return ϵ_{pred};

3.3. Training and Inference

During training, for each positive triple (h, r, t), we sample $\tau \sim \text{Uniform}(\{1, \ldots, T\})$ and noise $\epsilon \sim \mathcal{N}(0, I)$, construct X_τ^t, and let the denoiser predict ϵ_{pred}. We then reconstruct the clean embedding $\hat{X}_0^t$ and apply the negative log-likelihood loss [Mikolov et al., 2013]:

$$\begin{aligned} L = &-\log \sigma(\gamma - d_1(X_0^t, \hat{X}_0^t)) \\ &-\frac{1}{n}\sum_{i=1}^{n} \log \sigma(d_1(X_0^{t_i'}, \hat{X}_0^t) - \gamma), \end{aligned} \tag{20}$$

where γ is a fixed margin, σ is the sigmoid function, and d_1 is the $L1$ distance. Here, $X_0^t = \text{EMB}^v(t)$ is the clean tail embedding, $X_0^{t_i'} = \text{EMB}^v(t_i')$ are negative tail embeddings, and $\hat{X}_0^t$ is reconstructed from X_τ^t. The predicted noise and the final denoised result are mutually convertible through the function:

$$\hat{X}_0^t = \frac{1}{\sqrt{\bar{\alpha}_\tau}} X_\tau^t - \sqrt{\frac{1}{\bar{\alpha}_\tau} - 1}\, \epsilon_\theta(X_\tau^t, \tau, h, r). \tag{21}$$

The detailed procedure of MCDenoiser $\epsilon_\theta(X_\tau^t, \tau, h, r)$ is presented in Algorithm 1. Furthermore, the training process of MCDN is illustrated in Algorithm 2. In the inference phase, for predicting $(h, r, ?)$, MCDN iteratively denoises from Gaussian noise towards the latent vector of the target tail entity. This denoising process is guided by a trained MCDenoiser ϵ_θ, and multi-conditional inputs. Finally, the real tail entity is retrieved from the latent space via nearest neighbor search. This procedure is detailed in Algorithm 3.

Algorithm 2 Training Stage

Input: (h, r, t), $\{(h, r, t'_i)\}_{i=1}^{n}$;
Parameters: EMB^v, EMB^e, MCDenoiser ϵ_θ;
repeat
  Calculate $X^h \leftarrow EMB^v(h)$, $X^r \leftarrow EMB^e(r)$, $X_0^t \leftarrow EMB^v(t)$, and $X_0^{t'_i} \leftarrow EMB^v(t'_i)$;
  $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$;
  $\tau \sim \text{Uniform}(\{1, \dots, T\})$;
  $X_\tau^t = \sqrt{\bar{\alpha}_\tau}\, X_0^t + \sqrt{1-\bar{\alpha}_\tau}\, \epsilon$;
  $\hat{X}_0^t = \frac{1}{\sqrt{\bar{\alpha}_\tau}} X_\tau^t - \sqrt{\frac{1}{\bar{\alpha}_\tau} - 1}\, \epsilon_\theta(X_\tau^t, \tau, h, r)$;
  Take the gradient descent steps for alternating positive and negative:
  $L = -\log \sigma(\gamma - d_1(X_0^t, \hat{X}_0^t)) -$
    $\frac{1}{n} \sum_{i=1}^{n} \log \sigma(d_1(X_0^{t'_i}, \hat{X}_0^t) - \gamma)$;
until converged;

Algorithm 3 Inference Stage

Input: Incomplete triplet $(h, r, ?)$;
Output: Predicted target entity t;
$X^h \leftarrow EMB^v(h)$; $X^r \leftarrow EMB^e(r)$;
$X_\tau^t \sim \mathcal{N}(0, \boldsymbol{I})$;
for $\tau = T, \dots, 1$ **do**
  $z \sim \mathcal{N}(0, \boldsymbol{I})$ if $\tau > 1$, else $z = 0$;
  $\epsilon_{\text{pred}} = \epsilon_\theta(X_\tau^t, \tau, h, r)$;
  $X_{\tau-1}^t = \frac{1}{\sqrt{\alpha_\tau}} \left(X_\tau^t - \frac{\beta_\tau}{\sqrt{1-\bar{\alpha}_\tau}} \epsilon_{\text{pred}} \right) + \sigma_\tau z$;
end for
Search for $t = \arg\min_{t \in \mathcal{V}} \|\hat{X}_0^t - EMB^v(t)\|_1$;
Return t;

4. Experiment

To systematically validate the MCDN and the proposed multi-condition design, we organize a series of experiments to address the following research questions:

- **RQ1:** How does MCDN perform compared to mainstream knowledge graph completion models?
- **RQ2:** What are the distinct contributions of MCDN's key components to its performance? How does its performance adapt to variations in hyperparameter settings? Furthermore, How effective is the diffusion process?
- **RQ3:** How well does MCDN model complex multi-relation semantics, especially for multi-peaked (e.g., one-to-many) query distributions?
- **RQ4:** How sensitive is MCDN to different levels of knowledge graph data sparsity?
- **RQ5:** How does MCDN perform in inductive learning on knowledge graphs with unseen entities?

4.1. Experiment Settings

Datasets. We conduct experiments on four standard knowledge graph datasets: (i) FB15k-237, a subset of FB15k with inverse relations removed [Toutanova and Chen, 2015]; (ii) WN18RR, a subset of WN18 with inverse relations removed [Dettmers et al., 2018]; (iii) Kinship, a knowledge graph describing kinship relations, focusing on family and blood relations [Hinton, 1986]; (iv) UMLS, a biomedical knowledge graph encompassing medical terminology, concepts, and their interrelationships [Bodenreider, 2004]. The statistics above are summarized in Table 2.

Table 2
Statistics of transductive learning datasets. $|\mathcal{V}|$ denotes the number of entities, $|\mathcal{R}|$ the number of relations, $|\mathcal{F}|$ the number of training triples, $|\mathcal{T}_{\text{val}}|$ the number of validation triples, and $|\mathcal{T}_{\text{tst}}|$ the number of test triples. $\mathbb{E}[d]$ and $\text{Var}[d]$ represent the mean and variance of the node degree, respectively.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{T}_{\text{val}}\rvert$	$\lvert\mathcal{T}_{\text{tst}}\rvert$	$\mathbb{E}[d]$	$\text{Var}[d]$
FB15k-237	14,541	237	272,115	17,535	20,466	42.6	16307
WN18RR	40,943	11	86,835	3,034	3,134	4.2	64.3
Kinship	104	7	10,686	2,500	2,500	205.5	2.8
UMLS	135	46	5,327	569	633	37.4	1205.3

Baselines. We categorize all baseline methods into four groups: **(i) Embedding-based:** TransE [Bordes et al., 2013], DistMult [Yang et al., 2015], ComplEx [Trouillon et al., 2016], RotatE [Sun et al., 2019], ConvE [Dettmers et al., 2018], TDN [Wang et al., 2023], KERMIT [Li et al., 2025c]; **(ii) Rule-based:** RNNlogic [Qu et al.], DRUM [Sadeghian et al., 2019]; **(iii) GNN-based:** CompGCN [Vashishth et al., 2020], Pathcon [Wang et al., 2021], NBFNet [Zhu et al., 2021], NORAN [Zhang et al., 2024b], AdaGCN [Li et al., 2025d]; **(iv) Diffusion-Based:** FDM [Long et al., 2024a], KGDM [Long et al., 2024b], DiffusionE [Cao et al., 2024].

Evaluation Metrics. For each test triplet (h, r, t), we construct two queries: $(h, r, ?)$ and $(?, r, t)$, where the answers are t and h, respectively. Mean Reciprocal Rank (MRR) and Hits@N are selected as evaluation metrics.

Implementation Details. We optimize all models using Adam with a learning rate of 1×10^{-4} and a batch size of 128. The embedding dimension is set to 128 and we train for 30 epochs. For the semantic context condition module, the semantic space dimension is 32 and the temperature in the semantics-aware Top-K mechanism is 0.95. For the conditions adaptive fusion module, we use two fusion blocks and set the hidden size of the MLP to 1600. All experiments are conducted on a single NVIDIA RTX 4090 (24GB). Notably, we observe that the optimal context-hop number and Top-K value are dataset-dependent. Therefore, we tune key hyperparameters independently for each dataset using a grid search over: context hops $\{1, 2, 3, 4\}$, number of sampled neighbors $\{8, 16, 24, 32\}$, and Top-K $\{3, 4, 5, 8, 10\}$. The best dataset-specific hyperparameter choices are summarized in Table 5.

4.2. Overall Performance Comparison: RQ1

To answer RQ1, we conducted extensive experiments comparing MCDN with state-of-the-art baseline methods. The results, presented in Table 3, demonstrate the strong performance of MCDN, with the highest mean reciprocal

Table 3
Performance comparison on FB15k-237, WN18RR, Kinship and UMLS datasets. All metrics (MRR and Hits@N) are reported in percentage. Best results are in **bold**, second best are underlined.

Type	Model	FB15k-237			WN18RR			Kinship			UMLS		
		MRR	H@1	H@3	MRR	H@1	H@3	MRR	H@1	H@3	MRR	H@1	H@3
Emb-based	TransE	28.9	19.8	32.4	22.6	-	-	31.0	0.9	-	69.0	52.3	-
	DistMult	24.1	15.5	26.3	43.0	39.0	44.0	35.4	18.9	40.0	39.1	25.6	44.5
	ComplEx	24.7	15.8	27.5	44.0	41.0	46.0	41.8	24.2	49.9	41.1	27.3	46.8
	RotatE	33.7	24.1	37.5	44.7	42.8	49.2	65.1	50.4	75.5	74.4	63.6	82.2
	ConvE	32.5	23.7	35.6	43.0	40.0	44.0	68.5	55.2	78.6	75.6	69.7	80.7
	TDN	35.0	26.3	39.5	48.1	43.9	50.2	78.0	67.7	86.7	86.0	82.4	87.3
	KERMIT	35.9	26.6	39.6	**70.0**	**62.9**	**73.8**	-	-	-	92.1	85.6	98.3
Rule-based	RNNLogic	34.4	25.2	38.0	48.3	44.6	49.7	72.2	59.8	81.4	84.2	77.2	89.1
	DRUM	23.8	17.4	26.1	38.2	36.9	38.6	33.4	18.3	37.8	54.8	35.8	69.9
GNN-based	CompGCN	35.5	26.4	39.0	47.9	44.3	49.4	-	-	-	92.7	86.7	-
	PathCon	48.3	42.5	49.9	52.2	46.2	54.6	-	-	-	-	-	-
	NBFNet	41.5	32.1	45.4	55.1	49.7	57.3	60.6	43.5	72.5	77.8	68.8	84.0
	NORAN	49.9	45.1	51.9	54.0	46.7	56.0	-	-	-	-	-	-
	AdaGCN	37.0	27.5	39.6	47.2	45.0	50.1	-	-	-	97.7	96.6	99.2
Diffusion-based	FDM	48.5	38.6	52.9	50.6	45.6	51.8	83.7	76.1	89.7	92.2	89.3	94.4
	KGDM	52.0	42.3	56.6	51.6	45.7	51.9	78.9	68.7	87.0	90.9	87.2	93.7
	DiffusionE	37.6	29.4	-	55.7	50.4	-	-	-	-	**97.0**	**95.7**	-
Ours	MCDN	**55.1**	**52.9**	**57.2**	57.4	55.0	58.5	**85.4**	**79.8**	**91.1**	94.8	91.3	**98.6**

Table 4
Hit@10 performance comparison on FB15k-237, WN18RR, Kinship and UMLS datasets. Best results are in **bold**, second best are underlined.

Model	FB15k-237	WN18RR	Kinship	UMLS
	H@10	H@10	H@10	H@10
TransE	46.5	50.1	75.7	98.9
DistMult	41.9	49.0	74.3	72.9
ComplEx	42.8	51.0	76.0	75.2
RotatE	53.3	57.1	93.2	93.9
ConvE	50.1	52.0	92.8	91.9
TDN	54.6	48.1	96.5	96.8
KERMIT	54.7	**83.3**	-	99.5
RNNLogic	53.0	55.8	94.9	96.5
DRUM	36.4	41.0	67.5	85.4
CompGCN	53.5	54.6	-	99.4
PathCon	**71.6**	62.4	96.3	98.9
NBFNet	59.9	66.6	93.7	93.8
NORAN	-	-	-	-
AdaGCN	54.6	54.8	-	99.6
FDM	68.1	59.2	96.8	97.0
KGDM	70.8	59.3	97.2	97.3
DiffusionE	53.9	65.8	-	99.2
MCDN	65.3	70.7	**97.9**	**99.8**

rank (MRR) scores on two out of the four datasets and ranks second on the remaining two datasets. Specifically, on FB15k-237, the MCDN achieves an MRR score of 55.1, which is 3.1 percentage points (6.0% relative) more than the strongest competitor KGDM (52.0). On WN18RR, the MRR of the MCDN is 57.4, representing a 1.7 percentage points (3.1% relative) improvement over the next best model of the same type DiffusionE (55.7). On the Kinship dataset, the MCDN achieves an MRR of 85.4, marking a 1.7 percentage points (2.0% relative) gain over the second-best performing model FDM (83.7). On the UMLS dataset, the MCDN achieves a highly competitive MRR of 94.8, ranking second to DiffusionE (97.0). Compared with embedding-based models (best MRRs: KERMIT (35.9, 70.0, and 92.1 on FB15k-237, WN18RR, and UMLS, respectively)), the MCDN achieves better MRR on nearly all datasets.

Table 4 further shows the Hits@10 metric for the models across all datasets. These results demonstrate the effectiveness and robustness of the MCDN model. Although specialized models such as PathCon and the KERMIT exhibit outstanding performance on individual datasets of FB15k-237 and WN18RR, respectively, the MCDN consistently

Table 5
Dataset-specific hyperparameter configurations of MCDN.

Hyperparameter	FB15k-237	WN18RR	Kinship	UMLS
Context hops	2	3	3	3
Neighbor samples	16	8	8	8
Top-K	10	4	4	4

Table 6
Training complexity and memory requirements. n_{pos} is positive samples, n_{neg} is the negative samples, k is the number of neighbors sampled per node, $hops$ is the number of hops on the graph, and d is the embedding dimension.

Metric	FB15k-237	WN18RR	Kinship	UMLS
Training Complexity	$\mathcal{O}(n_{pos} \cdot (k^{hops} \cdot d^2 + n_{neg} \cdot d))$			
VRAM (MB)	5284	5260	5814	6443

Table 7
Ablation study on key components of MCDN.

Ablation Settings	FB15k-237		WN18RR	
	MRR	Hit@1	MRR	Hit@1
MCDN	**55.1**	**52.9**	**57.4**	**55.0**
MCDN w/o Entity Context	50.7	47.9	53.8	51.7
MCDN w/o Relation Context	48.2	46.3	48.7	47.0
MCDN w/o Top-K	52.2	50.1	56.0	54.2
MCDN w/o Connection Patterns	49.3	47.7	51.2	48.4
MCDN w/ Transformer	51.7	49.9	54.6	52.7

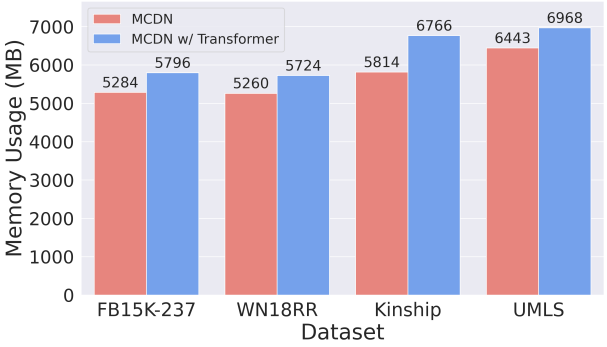


Figure 3: Comparison of Peak Memory Usage between MLP-based and Transformer-based Backbones.

ranks among the top performers across multiple benchmarks. Additionally, training time complexity and memory requirements of the MCDN are presented in Table 6.

In conclusion, these results demonstrate that an effective modeling of multiple relation semantics and the distribution of target entities during prediction by the MCDN significantly enhances the performance of embedding-based methods and improves their effectiveness in KGC.

4.3. Ablation Study: RQ2

4.3.1. Key Modules Ablation Study

To verify the effectiveness of our proposed modules, we developed five distinct model variants:

- **w/o Entity Context**: the entity context, a neighborhood structural component, was removed from the semantic context condition learning module, retaining only the relation context.
- **w/o Relation Context**: the relation context, a neighborhood structural component, was removed from the semantic context condition learning module, retaining only the entity context.
- **w/o Top-K**: the Top-K selection mechanism was replaced with a simple random sampling strategy in the semantic context condition learning module.
- **w/o Connection Patterns**: the connection pattern condition learning module was replaced by simply concatenating the entity and relation context.
- **w/ Transformer**: MLP layers in the conditions adaptive fusion module were replaced with Transformer layers.

The ablation study results, presented in Table 7, provide crucial insights into the individual contributions of MCDN's key components, leading to the following observations.

Firstly, the Relation Context emerges as the most critical component for MCDN's performance. Its removal (MCDN w/o Relation Context) results in the most substantial performance degradation, with MRR decreasing by 6.9 (from 55.1 to 48.2) and Hit@1 by 6.6 points (from 52.9 to 46.3) on FB15k-237, and even more significantly on WN18RR, with the MRR decreasing by 8.7 (from 57.4 to 48.7) and Hit@1 by 8.0 points (from 55.0 to 47.0). This corresponds to an average relative performance decrease of approximately 13.68% across these metrics and datasets, underscoring the paramount importance of relation-specific contextual information for accurate link prediction.

Secondly, Connection Patterns are also highly influential. Excluding this module (MCDN w/o Connection Patterns) results in considerable performance drops: the MRR decreases by 5.8 points on FB15k-237 and 6.2 points on WN18RR, while Hit@1 scores decrease by 5.2 and 6.6 points, respectively. Thus, average relative performance decreases by approximately 10.79%. This finding robustly supports the hypothesis that entities and relations within KGs are not independent, and explicit modeling of their connection patterns enables highly effective use of conditional information within triplets to guide the generation process.

Furthermore, the entity context is indispensable. Its absence (MCDN w/o Entity Context) results in MRR reductions of 4.4 on FB15k-237 and 3.6 on WN18RR, and Hit@1 reduction of 5.0 and 3.3 points, respectively. Although entity and relation contexts are vital, the data clearly indicate that the relation context has a significant impact.

The ablation of the Top-K selection mechanism (MCDN w/o Top-K) also affects performance, with the MRR decreasing by 2.9 on FB15k-237 and 1.4 on WN18RR. This indicates its utility in refining candidate predictions, although its impact is less pronounced than the contextual components.

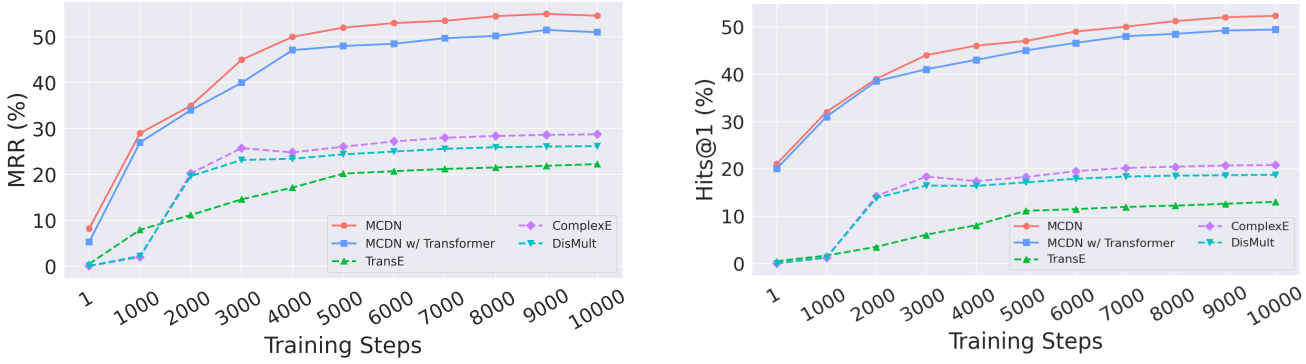
Finally, we evaluate the MLP layers within the conditions adaptive fusion module. Their removal (MCDN w/ Transformer) results in MRR decrease of 3.4 on FB15k-237 and 2.8 on WN18RR, and Hit@1 decrease of 3.0 and 2.3 points, respectively. This translates to an average relative performance decrease of approximately 5.23% in the MCDN. This observation suggests that the simpler MLP architecture is highly adept at modeling the relatively straightforward factual structures in these KGs. Conversely, a highly complex fusion mechanism, such as a Transformer explored in a few designs [Peebles and Xie, 2023] may not be as well-suited for this particular form of KG data, potentially owing to the risk of overfitting or a mismatch in complexity.

To substantiate our backbone choice, we compare MLP and Transformer variants of MCDN from the perspectives of complexity, memory footprint, and optimization efficiency.

- **Complexity.** Self-attention incurs $O(n^2 d)$ complexity with sequence length n, while our denoiser input consists of a small number of fixed-size condition vectors, making the quadratic term unnecessary. In contrast, an MLP scales approximately as $O(d^2)$ and is therefore a better fit for this setting.

Table 8
Cumulative training time (in minutes) corresponding to the training steps in Figure 3. This table quantifies the real-world time cost of the training process, providing context for the convergence speed and overall efficiency of each model.

Model	Training Steps										
	Start	1	1000	2000	3000	4000	5000	6000	7000	8000	10000
MCDN - MLP	0	9.97	25.87	43.35	60.03	75.68	91.63	108.32	125.48	143.25	175.05
MCDN - Transformer	0	11.28	28.47	45.63	62.95	80.32	97.53	114.97	132.88	150.05	184.50
TransE	0	1.45	2.78	4.13	5.48	6.85	8.22	9.57	10.92	12.28	16.28
ComplEx	0	1.65	3.20	4.77	6.30	7.87	9.43	11.02	12.57	14.13	18.77
DistMult	0	1.38	2.73	4.07	5.40	6.78	8.13	9.52	10.90	12.28	16.30



(a) MRR Convergence.

(b) Hit@1 Convergence.

Figure 4: Training dynamics and convergence comparison on the FB15k-237 dataset. The plots show the evolution of (a) MRR and (b) Hits@1 metrics over training steps. Our proposed MCDN not only achieves superior final performance but also demonstrates significantly faster convergence compared to both the Transformer-based variant and other baseline models.

- **Memory footprint.** As shown in Figure 3, the Transformer variant consistently consumes more GPU memory across all benchmarks. For example, on Kinship the peak memory increases from 5814 MB (MLP) to 6766 MB (Transformer), indicating the additional cost of attention parameters and intermediate activations.
- **Training and convergence.** Table 8 indicates that the Transformer variant incurs a modest **5.4%** overhead over the MLP backbone at 10,000 steps (184.50 vs. 175.05 min), while Figure 4 provides no evidence of improved optimization from attention. More broadly, despite a higher per-step cost than classical embedding models (e.g., TransE and ComplEx), MCDN is substantially more sample-efficient: it exceeds the best final performance of all baselines within **2,000 steps**, corresponding to approximately **43–46** minutes of cumulative training time, whereas the baselines plateau and fail to reach comparable performance even after 10,000 steps.

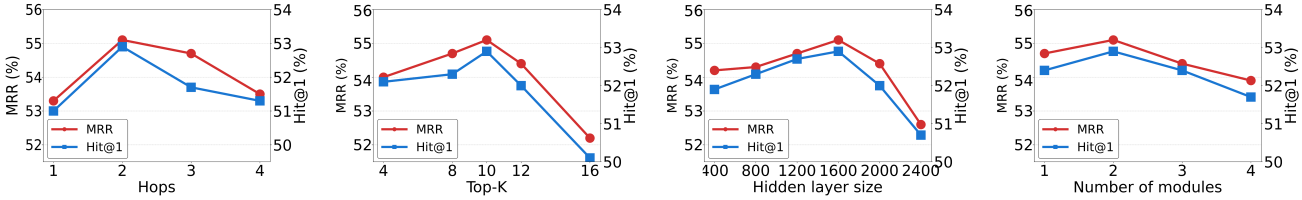
4.3.2. Sensitivity to Key Hyperparameters

We conducted a hyperparameter sensitivity analysis to investigate the impact of crucial settings within our model. Figure 5 presents the results on the FB15k-237 dataset.

Figure 5(a) illustrates the model's performance on varying numbers of hops for contextual information aggregation. Both MRR and Hit@1 metrics peak at 2 hops, indicating that expanding the contextual neighborhood up to this depth is beneficial. However, increasing the hop count to 3 or 4 degrades performance, likely owing to the introduction of noisy or less relevant information, which dilutes the signal from highly pertinent local structures.

Figure 5(b) evaluates the influence of the Top-K parameter. This parameter controls the number of neighbors selected for the current node at each hop, specifically filtering the neighbors with the highest semantic similarity. This selection focuses on the most relevant contextual information. The optimal performance is achieved with a K-value of 10. Using a smaller value of K (e.g., 4 or 8) may insufficiently capture of contextual richness, as fewer highly relevant and semantically similar neighbors are considered. Conversely, a larger K (e.g., 16) can dilute the signal by incorporating less semantically similar neighbors, introducing noise and causing a notable performance drop. This highlights the critical balance in selecting an appropriate neighborhood size to maximize relevant information while minimizing interference from less pertinent neighbors.

Figure 5(c) shows the effect of the hidden layer size in the conditions adaptive fusion modules. The performance steadily improves as the hidden dimension increases from 400 to 1600. However, further increasing the hidden size (e.g., to 2000 or 2400) degrades performance. This suggests that although a sufficiently large hidden dimension is necessary for capturing complex conditional patterns, excessively



(a) Number of Hops. (b) Number of Top-K. (c) Hidden layer size. (d) Number of modules.

Figure 5: Hyperparameter sensitivity analysis on FB15k-237, reporting the impact of key settings on MRR and Hit@1.

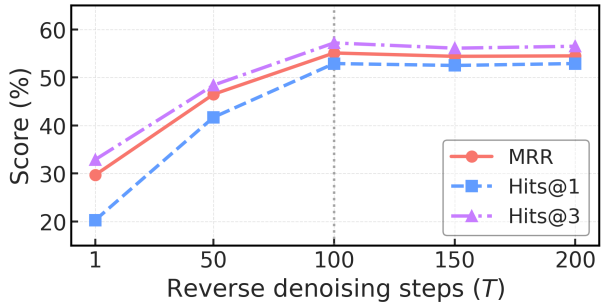


Figure 6: Effect of the reverse denoising steps T on FB15k-237.

large sizes may increase model complexity and potential overfitting without commensurate gains in expressive power.

Figure 5(d) illustrates the effect of the number of stacked conditions adaptive fusion modules. The model exhibits its best results with 2 modules. Using a single module appears insufficient to fully leverage conditional information, whereas stacking 3 or 4 modules decreases performance. This decline in deep configurations might be attributed to optimization or overfitting challenges as the network depth increases for this particular component.

4.3.3. Effectiveness of the Diffusion

To quantify the role of the diffusion process in KGC, we conduct a controlled comparison that isolates the impact of iterative denoising while keeping the condition encoders, the denoiser backbone, and the training objective unchanged. Concretely, we vary the number of reverse denoising steps $T \in \{1, 50, 100, 150, 200\}$ at inference time and report MRR, Hits@1, and Hits@3 on FB15k-237. Figure 6 shows a clear improvement when increasing T from 1 to 100. This indicates that iterative refinement is essential: a degenerated one-step generation (T=1) is insufficient to recover high-quality target embeddings, whereas multi-step denoising progressively concentrates the prediction toward valid entities under multi-condition guidance. Beyond T=100, the performance saturates and slightly fluctuates, suggesting diminishing returns from further increasing denoising steps. Therefore, we adopt T=100 as the default setting in all experiments for an accuracy-efficiency trade-off.

4.4. Multi-Relation Semantics Analysis: RQ3

The 1-N prediction task [Wang et al., 2017] serves as a critical benchmark for evaluating the capacity of a model for semantic understanding. We analyze MCDN by relation categories [Wang et al., 2014] on FB15k-237. Table 9 reports the MRR for each category. The results indicate that MCDN exhibits more improvement on 1-1, 1-N, N-1, and N-N relations. These categories typically exhibit more complex and potentially multi-peaked query distributions, for which probabilistic modeling of target entities is beneficial.

Table 9
Entity prediction by relation category on FB15k-237. The metric is MRR in percentage.

Category	TransE	RotatE	NBFNet	FDM	MCDN
Head Prediction					
1-1	49.8	48.7	57.8	56.9	**59.5**
1-N	7.9	8.1	16.5	20.3	**21.7**
N-1	45.5	46.7	49.9	55.9	**58.8**
N-N	22.4	23.4	34.8	42.3	**44.1**
Tail Prediction					
1-1	48.8	48.4	60.0	51.9	**61.7**
1-N	74.4	74.7	79.0	82.6	**84.4**
N-1	7.1	7.0	12.2	16.7	**18.5**
N-N	33.0	33.8	45.6	54.3	**54.6**

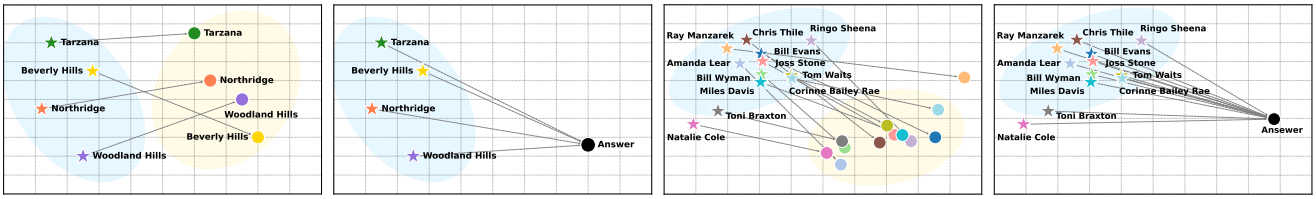
Furthermore, We conducted a qualitative analysis using illustrative examples from the FB15k-237 test dataset (see Table 10). The t-SNE visualization presented in Figure 7 clearly highlights the contrasting behaviors of the two models. Figures 7(a) and (c) show that the MCDN-generated embeddings form a distinct cluster that effectively captures the distribution of the multiple ground-truth entities. This indicates the effectiveness of the model in capturing the inherent one-to-many ambiguity present in the query.

In contrast, Figures 7(b) and (d) show that TransE produces a single point estimate exclusively for its prediction. This discrepancy stems from fundamental architectural differences. The deterministic nature of TransE, which relies on the translation $t = h + r$, forces it to compute a single vector that represents an average of all potential correct answers, thereby failing to accurately capture any single answer.

Our MCDN is designed as a conditional generative model. Instead of predicting a single point, it learns the underlying probability distribution of target entities. Guided by rich semantic contexts and connection patterns, MCDN's reverse diffusion process generates a diverse set of candidate entities. These predictions are not collapsed into an uninformative average; instead, they are strategically positioned in close proximity to the actual ground-truth

Table 10
Knowledge Graph Query Examples on the FB15k-237 Dataset. The table shows two distinct 1-N queries for Los Angeles locations and Jazz artists. Square brackets [...] enclose Freebase Machine IDs.

Component	Example Instance			
Query (h, r, ?)	(Los Angeles [/m/030qb3t], location contains [/location/location/contains], ?)			
Reference Answers (Tail Entities)	Tarzana [/m/0281rb]	Beverly Hills [/m/0k049]	Northridge [/m/0k_mf]	Woodland Hills [/m/0k_p5]
Query (h, r, ?)	(Jazz [/m/03_d0], music genre artists [/music/genre/artists], ?)			
Reference Answers (Tail Entities)	Bill Evans [/m/014g91]	Amanda Lear [/m/01l_vgt]	Chris Thile [/m/01m15br]	Bill Wyman [/m/01vsksr]
	Joss Stone [/m/01wk7ql]	Ringo Sheena [/m/01wy61y]	Ray Manzarek [/m/021r7r]	Natalie Cole [/m/026spg]
	Toni Braxton [/m/02h9_l]	Tom Waits [/m/03h_fqv]	Miles Davis [/m/053yx]	Corinne Bailey Rae [/m/0b68vs]



(a) MCDN embeddings. (b) TransE embeddings. (c) MCDN embeddings. (d) TransE embeddings.

Figure 7: T-SNE visualization of embeddings. The first two figures on the left are about the first example in the table 6, and the two on the right are about the second example.

entities, demonstrating a highly sophisticated and practical understanding of one-to-many semantics.

4.5. Sensitivity to Data Sparsity: RQ4

We observe that MCDN performs slightly worse on UMLS than on other datasets. Table 2 shows that UMLS has a limited size and semantic diversity, which may provide insufficient contextual signals for MCDN. In contrast, DiffusionE may be robust on such sparse graphs due to its emphasis on semantic consistency via adaptive propagation.

To empirically validate this hypothesis, we design a controlled sparsity experiment by sub-sampling FB15k-237, which exhibits similar node-degree characteristics to UMLS. Given the absence of an official implementation for DiffusionE[2], we focus on evaluating how data volume impacts message-passing mechanisms. By randomly sampling FB15k-237 at various proportions, we quantify the performance of MCDN on smaller-scale graphs, and the results are reported in Figure 8.

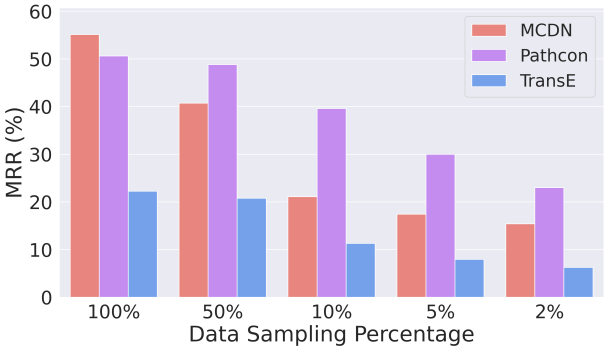


Figure 8: Performance analysis on FB15k-237 under varying training data sparsity. The results demonstrate the models' sensitivity to the volume of training data. While MCDN achieves strong performance on the full dataset, its performance degrades more rapidly than the more robust PathCon model as data becomes sparse. This provides empirical evidence that MCDN's architecture is more data-hungry, justifying its relatively lower performance on the small UMLS dataset.

4.6. Inductive Reasoning: RQ5

Building on the demonstrated superiority of the MCDN in the transductive setting, we further validated its inductive learning power. This direction is motivated by concurrent studies, such as DKGC [Chen et al., 2024], which have already confirmed the guiding role of relational semantics for diffusion-based models in inductive tasks. To better adapt the proposed model to this task, we introduced a variant named **MCDN w/ Relation**. Specifically, we modify the scaling and fusion mechanism within our conditions adaptive fusion module as follows:

$$X_\tau^t = \beta_2^{(\ell)} \cdot X_\tau^t + \beta_1^{(\ell)} + \alpha^{(\ell)}, \tag{22}$$

[2]As DiffusionE's adaptive message passing and PathCon's relation-based message passing are conceptually similar, we use PathCon as a proxy to evaluate the impact of data volume on message-passing mechanisms.

Table 11
Inductive results. The best results are shown in bold while the second-best results are shown underlined.

Type	Model	FB15k-237								NELL-995							
		v1		v2		v3		v4		v1		v2		v3		v4	
		MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10
GNN-based	GraIL	27.9	42.9	27.6	42.4	25.1	42.4	22.7	38.9	48.1	56.5	29.7	49.6	32.2	51.8	26.2	50.6
	NBFNet	30.7	51.7	36.9	63.9	33.1	58.8	30.5	55.9	58.4	79.5	41.0	63.5	42.5	60.6	28.7	59.1
	RED-GNN	36.9	48.3	46.9	62.9	44.5	60.3	44.2	62.1	63.7	86.6	41.9	60.1	43.6	59.4	36.3	55.6
	AdaProp	31.0	55.1	47.1	65.9	47.1	63.7	45.4	63.8	64.4	**88.6**	45.2	65.2	43.5	61.8	36.6	**60.7**
Diffusion-based	DiffusionE	31.0	52.5	47.4	66.1	45.6	62.7	44.6	62.5	**67.3**	87.3	42.4	66.0	45.8	62.0	30.8	52.3
Ours	MCDN	25.4	40.3	35.2	60.5	35.9	59.9	34.7	60.1	-	-	33.2	55.2	30.7	51.3	26.5	51.2
	MCDN w/ Relation	30.6	52.2	40.8	60.8	42.3	60.4	41.5	61.2	-	-	39.4	61.1	38.8	58.8	33.7	56.8

where $\beta_2^{(\ell)}$ represents the scaling parameter learned from the relation contexts after processing by the adaptive parameterized projector.

4.6.1. Experiments Settings

Datasets. We proceed with four subsets each from the FB15k-237 and NELL-995 datasets, for a detailed examination of the dataset specifics, refer to Table 12.

Baselines. Traditional embedding-based models are incapable of generalizing to unseen entities. Therefore, our selection of baselines is drawn from the following two categories: **(i) GNN-based:** GraIL [Teru et al., 2020], NBFNet [Zhu et al., 2021], RED-GNN [Zhang and Yao, 2022], and AdaProp [Zhang et al., 2023b]. **(ii) Diffusion-based:** DiffusionE [Cao et al., 2024].

Implementation Details. The setup and parameter selection are consistent with the transductive experiments.

4.6.2. Overall Performance

Table 11 presents the results of our inductive learning. The MCDN w/ Relation demonstrates strong and robust generalization capabilities, achieving highly competitive performance against state-of-the-art GNN-based and diffusion-based models across eight distinct inductive settings on FB15k-237 and NELL-995.

On FB15k-237, MCDN w/ Relation consistently performs similarly to leading baselines such as AdaProp and DiffusionE, particularly in the more challenging v2, v3, and v4 versions. Notably, the proposed model exhibits remarkable stability across these settings, with MRR scores remaining consistently high (40.8 to 42.3). On the highly sparse NELL-995, the model remains highly competitive, surpassing several strong baselines and exhibiting comparable performance with other examined models.

However, the most crucial insight is obtained from the ablation study comparing MCDN w/ Relation with its vanilla counterpart MCDN. The results demonstrate that the model achieves substantial and consistent performance gains across all evaluated settings after refining the fusion mechanism. For instance, on the v3 and v4 versions of NELL-995, the MRR is boosted by **26.7%** and **27.2%**, respectively, whereas the H@10 score on FB15k-237 v1 improves by nearly **relative 30%**. This demonstrates that the proposed relation-context scaling mechanism is the key contributor to the strong inductive reasoning capability of the model, effectively guiding the diffusion process for unseen entities.

In summary, although MCDN w/ Relation does not consistently outperform the top model on every metric, it remains a robust and competitive method for inductive KGC.

Table 12
Statistics of inductive learning datasets. $|\mathcal{R}|$ and $|\mathcal{R}_{\text{tst}}|$ denote the number of relations in the training and test sets, respectively. Similarly, $|\mathcal{V}|$ and $|\mathcal{V}_{\text{tst}}|$ represent the number of entities, and $|\mathcal{F}|$ and $|\mathcal{T}_{\text{tst}}|$ represent the number of triples.

Dataset		$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{R}_{\text{tst}}\rvert$	$\lvert\mathcal{V}_{\text{tst}}\rvert$	$\lvert\mathcal{T}_{\text{tst}}\rvert$
FB15k-237	v1	180	1,594	5,226	142	1,093	2,404
	v2	200	2,608	12,085	172	1,660	5,092
	v3	215	3,668	22,394	183	2,501	9,137
	v4	219	4,707	33,916	200	3,051	14,554
NELL-995	v1	14	3,103	5,540	14	225	1,034
	v2	88	2,564	10,109	79	2,086	5,521
	v3	142	4,647	20,117	122	3,566	9,668
	v4	76	2,092	9,289	61	2,795	8,520

5. Conclusion

Herein, we introduced the MCDN, which reframed KGC as a conditional generation task. By jointly conditioning on semantic context and connection patterns, the MCDN effectively modeled the distribution of target entities with an accurate prediction process. Extensive experiments demonstrate that the MCDN achieved superior performance across several challenging benchmark datasets. Despite its empirical success, a primary limitation of the MCDN was in its computational efficiency. The iterative sampling process inherent to diffusion models exhibited more training latency and memory footprint than single-step, nongenerative approaches, highlighting a fundamental trade-off between the expressive power of conditional generative modeling and its associated computational cost. In future studies, we aim to improve the training efficiency of the MCDN by adopting

advanced and efficient diffusion frameworks such as the recently proposed mean-flow method. We plan to extend the MCDN to large-scale KGs to thoroughly evaluate its completion capability when facing the vast knowledge semantics of the real world. We might enhance the proposed model by exploring new condition fusion mechanisms and context aggregation techniques to further enhance its performance.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China under Grant Nos. 62202085, 61772110; the Fundamental Research Funds for the Central Universities under Grant No. DUT23RC(3)063.

Declaration of Competing Interest

The authors report no potential conflicts of interest.

CRediT authorship contribution statement

Siyuan Li: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - Original Draft. **Liang Zhang:** Conceptualization, Methodology, Supervision, Writing - Review & Editing, Funding Acquisition, Project Administration. **Bo Jin:** Supervision. **Xiaopeng Wei:** Supervision.

References

Tong Shen, Fu Zhang, and Jingwei Cheng. A comprehensive overview of knowledge graph completion. *Knowledge-Based Systems*, 255:109597, 2022. ISSN 0950-7051.

Qing Li, George Baciu, Jiannong Cao, Xiao Huang, Richard Chen Li, Peter HF Ng, Junnan Dong, Qinggang Zhang, Zackary PT Sin, and Yaowei Wang. Kcube: A knowledge graph university curriculum framework for student advising and career planning. In *International Conference on Blended Learning*, pages 358–369. Springer, 2022a.

Wentao Li, Huachi Zhou, Junnan Dong, Qinggang Zhang, Qing Li, George Baciu, Jiannong Cao, and Xiao Huang. Constructing low-redundant and high-accuracy knowledge graphs for education. In *International Conference on Web-Based Learning*, pages 148–160. Springer, 2022b.

Junnan Dong, Qinggang Zhang, Xiao Huang, Keyu Duan, Qiaoyu Tan, and Zhimeng Jiang. Hierarchy-aware multi-hop question answering over knowledge graphs. In *Proceedings of the ACM Web Conference 2023*, pages 2519–2527, 2023.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE transactions on knowledge and data engineering*, 29(12):2724–2743, 2017.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021.

Qinggang Zhang, Junnan Dong, Qiaoyu Tan, and Xiao Huang. Integrating entity attributes for error-aware knowledge graph embedding. *IEEE Transactions on Knowledge and Data Engineering*, 2023a.

Xiaojun Chen, Shengbin Jia, and Yang Xiang. A review: Knowledge reasoning over knowledge graph. *Expert systems with applications*, 141: 112948, 2020.

Yuchao Zhang, Xiangjie Kong, Zhehui Shen, Jianxin Li, Qiuhua Yi, Guojiang Shen, and Bo Dong. A survey on temporal knowledge graph embedding: Models and applications. *Knowledge-Based Systems*, 304: 112454, 2024a.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

Xiao Long, Liansheng Zhuang, Aodi Li, Houqiang Li, and Shafei Wang. Fact embedding through diffusion model for knowledge graph completion. In *Proceedings of the ACM on Web Conference 2024*, pages 2020–2029, 2024a.

Hongwei Wang, Hongyu Ren, and Jure Leskovec. Relational message passing for knowledge graph completion. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1697–1707, 2021.

Yongqi Zhang, Zhanke Zhou, Quanming Yao, Xiaowen Chu, and Bo Han. Adaprop: Learning adaptive propagation for graph neural network based knowledge graph reasoning. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3446–3457, 2023b.

Qinggang Zhang, Keyu Duan, Junnan Dong, Pai Zheng, and Xiao Huang. Logical reasoning with relation network for inductive knowledge graph completion. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4268–4277, 2024b.

Siyuan Li, Ruitong Liu, Te Sun, Yan Wen, Ruihao Zhou, Jingyi Kang, and Yunjia Wu. Context-driven knowledge graph completion with semantic-aware relational message passing. In *International Conference on Neural Information Processing*, pages 567–580. Springer, 2025a.

Aleksandar Pavlović and Emanuel Sallinger. Expressive: A spatio-functional embedding for knowledge graph completion. In *International Conference on Learning Representations*, 2023.

Zhiyu Chen, Guojiang Shen, Yuqi Shen, Zhi Liu, and Xiangjie Kong. A diffusion model for inductive knowledge graph completion. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2024. doi: 10.1109/IJCNN60899.2024.10651154.

Xiao Long, Liansheng Zhuang, Aodi Li, Jiuchang Wei, Houqiang Li, and Shafei Wang. Kgdm: A diffusion model to capture multiple relation semantics for knowledge graph embedding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(8):8850–8858, Mar. 2024b.

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

Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. 2015.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. 2019.

Tengwei Song, Jie Luo, and Lei Huang. Rot-pro: Modeling transitivity by projection in knowledge graph embedding. In *Proceedings of the Thirty-Fifth Annual Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3065–3072, 2020.

Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. Boxe: A box embedding model for knowledge base completion. *Advances in Neural Information Processing Systems*, 33:9649–9661, 2020.

Zhao Li, Xin Wang, Jun Zhao, Wenbin Guo, and Jianxin Li. Hycube: Efficient knowledge hypergraph 3d circular convolutional embedding.

IEEE Transactions on Knowledge and Data Engineering, 37(4):1902–1914, 2025b.

Zedong Liu, Jianxin Li, Yongle Huang, Ningning Cui, and Lili Pei. Knowledge-based natural answer generation via effective graph learning. *Knowledge-Based Systems*, 316:113288, 2025.

Duy Le, Shaochen (Henry) Zhong, Zirui Liu, Shuai Xu, Vipin Chaudhary, Kaixiong Zhou, and Zhaozhuo Xu. Knowledge graphs can be learned with just intersection features. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Haotian Li, Bin Yu, Yuliang Wei, Kai Wang, Richard Yi Da Xu, and Bailing Wang. Kermit: Knowledge graph completion of enhanced relation modeling with inverse transformation. *Knowledge-Based Systems*, 324: 113500, 2025c. ISSN 0950-7051.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851, 2020.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

Keyu Guo, Yongle Huang, Tinglei Jia, Xiangyu Song, Shijie Sun, Hongkai Wei, Xian-Feng Han, Shuwen Huang, Nicola Strisciuglio, and Shuyan Li. Visual grounding in 2d and 3d: A unified perspective and survey. *Information Fusion*, page 103625, 2025.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 32211–32252, 2023.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022c.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. 2023.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

Zongsheng Cao, Jing Li, Zigan Wang, and Jinliang Li. Diffusione: Reasoning on knowledge graphs via diffusion-based graph neural networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 222–230, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710, 2020.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China, July 2015. Association for Computational Linguistics.

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

Geoff Hinton. Kinship. UCI Machine Learning Repository, 1986.

Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl_1): D267–D270, 2004.

Jiapu Wang, Boyue Wang, Junbin Gao, Xiaoyan Li, Yongli Hu, and Baocai Yin. Tdn: Triplet distributor network for knowledge graph completion. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):13002–13014, 2023.

Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs.

Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in Neural Information Processing Systems*, 32, 2019.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P Talukdar. Composition-based multi-relational graph convolutional networks. 2020.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 34:29476–29490, 2021.

Zhifei Li, Lifan Chen, Yue Jian, Han Wang, Yue Zhao, Miao Zhang, Kui Xiao, Yan Zhang, Honglian Deng, and Xiaoju Hou. Aggregation or separation? adaptive embedding message passing for knowledge graph completion. *Information Sciences*, 691:121639, 2025d. ISSN 0020-0255.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

Komal Teru, Etienne Denis, and Will Hamilton. Inductive relation prediction by subgraph reasoning. In *International conference on machine learning*, pages 9448–9457. PMLR, 2020.

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