Diffusion Model with Selective Attention for Temporal Knowledge Graph Reasoning

Rushan Geng[1], Ge Chen[1], and Cuicui Luo[2] ✉

[1] School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China
{gengrushan23,chenge221}@mails.ucas.ac.cn
[2] International College, University of Chinese Academy of Sciences, Beijing, China
luocuicui@ucas.ac.cn

Abstract. Temporal knowledge graph reasoning aims to predict missing entities at future time steps, and as a critical task, it has attracted widespread attention in recent years due to its impressive ability to capture historical correlations and forecast future events. Although existing approaches, such as graph learning and logic rules, have partially addressed this problem, they still face limitations in modeling the uncertainty of future events—especially when predicting rare or unseen facts. To address these challenges, we propose a diffusion model based on a selective attention mechanism (DMSA) for temporal knowledge graph reasoning. In our method, the encoder incorporates selective attention to emphasize key information, while the diffusion module introduces noise to enhance the model's capability to predict unseen events. By integrating selective attention with the diffusion module, our model improves both its memory and its ability to predict future, unseen events. Experimental results on five public datasets demonstrate that our proposed model achieves state-of-the-art performance across multiple evaluation metrics.

Keywords: Temporal knowledge graph · Temporal knowledge graph reasoning · Diffusion model· Selective attention.

1 Introduction

Knowledge graphs (KGs) record facts about the real world as triples (s, r, o), where entities serve as nodes and relations as edges, forming a graph structure. However, traditional knowledge graphs only capture static facts and cannot reflect the dynamic evolution of events over time. For example, a fact such as $(\text{China}, \text{visit}, \text{USA})$ may convey significantly different information at different time points. To address this limitation, researchers have gradually incorporated temporal information to construct temporal knowledge graphs (TKGs), which are represented as quadruples (s, r, o, t), where s denotes the subject, r the relation, o the object, and t the timestamp. For instance, $(\text{China}, \text{visit}, \text{USA}, \text{2023-11-15})$ indicates that China visited the USA on November 15, 2023. TKGs not only

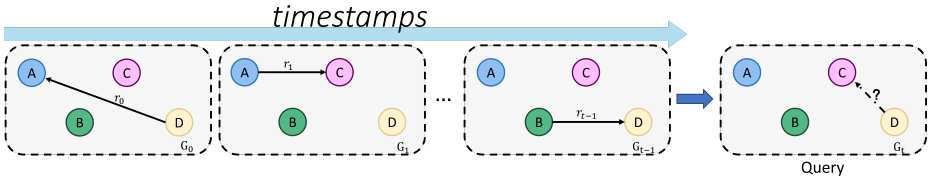


Fig. 1. One example for TKG reasoning.

maintain the simplicity and accuracy of traditional knowledge graphs but also dynamically capture the time-varying nature of facts, thereby playing an important role in various downstream natural language processing tasks, such as recommendation systems [1], question answering [2], and information retrieval [3].

Due to the inherent incompleteness of TKGs, researchers have been actively developing efficient reasoning methods to fill in missing information. TKG reasoning can typically be categorized into interpolation and extrapolation. Interpolation reasoning aims to predict missing facts within the observed time span—that is, given snapshots from time 0 to t, it predicts events that occurred within that period. In contrast, extrapolation reasoning leverages historical facts to forecast new events at future time steps (where $t_i > t$), as illustrated in Figure 1. Each snapshot corresponds to the facts occurring at a specific timestamp, and a TKG is composed of many such snapshots. This paper primarily focuses on extrapolation reasoning, which, compared with interpolation, not only poses greater challenges but also offers more practical value by completing future knowledge graphs and predicting emerging events [4].

Current TKG reasoning methods primarily rely on historical information to predict missing entities at future time steps, as illustrated in Figure 1. For example, many approaches use GNNs to learn the structural information from historical TKG snapshots and RNNs to capture temporal evolution patterns. Since past events often tend to reoccur, these methods have been widely adopted and studied. However, this reliance on historical data makes them less effective at predicting events that have never been observed before. Moreover, using the same historical information for all queries prevents the model from focusing on the most relevant details, as its attention gets dispersed by irrelevant data.

Based on this analysis, we propose DMSA, a diffusion model with selective attention for TKG reasoning, which is built on an encoder–decoder framework. By incorporating Gaussian noise through a diffusion process, DMSA enhances the model's ability to predict unseen events. In addition, a selective attention mechanism is introduced in the encoder to allow the model to autonomously choose the most pertinent information. Specifically, we employ CompGCN [10] as the encoder and integrate Relative Attention to select relation information that is closely linked to the current event. By converting entities, relations, and timestamps into a sequence prediction task based on historical snapshots—and

by injecting Gaussian noise into the sequence to introduce uncertainty—we improve the probability of accurately predicting unseen events. Finally, predictions are generated using Time-aware ConvTransE as the decoder. Extensive experiments on five public TKG datasets demonstrate that DMSA significantly outperforms state-of-the-art methods across multiple evaluation metrics.

In summary, the main contributions of this paper are as follows:

- We propose a novel approach that introduces the diffusion process into knowledge graph reasoning to increase the uncertainty in event representations, thereby enhancing the model's ability to predict unseen events.
- We incorporate a selective attention mechanism in the encoder, which enables the model to process information in a targeted manner within each snapshot.
- Extensive experiments on five public datasets show that DMSA significantly outperforms existing methods across multiple evaluation metrics.

2 Related work

Knowledge graph reasoning can be broadly categorized into static knowledge graph reasoning and temporal knowledge graph reasoning. Below, we briefly discuss some representative methods in each category. Next, we introduce research applications of diffusion models.

2.1 Static Knowledge Graph Reasoning

Static knowledge graph reasoning methods for temporal knowledge graphs generally ignore timestamps and directly process triples, mainly modeling the structural and semantic information of entities. TransE [5] treats relations as translation transformations when projecting entity embeddings into a latent space. They use distance functions (such as L1 and L2 norms) to score factual triples. DistMult [6] and ComplEx [7] represent knowledge graphs as three-dimensional tensors and decompose them into low-dimensional vectors to learn embeddings for entities and relations. Methods like ConvE [8] utilize transformations, bilinear objectives, complex embeddings, and convolutional operations to capture relational semantics. Graph Convolutional Networks (GCNs) are representative methods characterized by their ability to learn structural features of knowledge graphs. R-GCN [9] is a typical graph neural network model that integrates relational information through specialized message passing and aggregation mechanisms, enabling effective capture of complex patterns and dependencies in graph-structured data. CompGCN [10] generalizes several multi-relational GCN methods and employs various composition operations to handle multi-relational graph data.

2.2 Temporal Knowledge Graph Reasoning

Temporal Knowledge Graph (TKG) reasoning models the dynamic evolution of entities and relations over time. Early methods such as TA-DistMult [11]

incorporate time embeddings, while TeMP [12] leverages GNNs and RNNs to mitigate temporal sparsity via message passing. Building on these, RE-NET [13] uses an encoder-aggregator structure for fixed-length historical subgraphs, and CyGNet [14] employs a replication-based mechanism to capture repetitive patterns. EvoKG [15], RE-GCN [16], and TiRGN [17] further model dynamic and long-term dependencies through subgraph evolution, temporal gating, and time embeddings. Other approaches, including HGLS [18], SMiFY [19], CENET [20], PLEASING [21], LSEN [22], and HIP [23], enhance reasoning by constructing global graphs, simplifying architectures, or mining both short- and long-term patterns. PPT [24] converts temporal knowledge graph completion into a masked prediction task on pre-trained language models by designing dedicated prompts for entities, relations, and time intervals, enabling explicit modeling of temporal and relational semantics. LLM-DA [25] leverages large language models to extract interpretable temporal logical rules from historical data and dynamically updates these rules with recent events, enabling accurate and adaptive temporal knowledge graph reasoning without fine-tuning the LLMs. Yuan et al. [26] introduces the first explainable temporal reasoning task, accompanied by the ExplainTemp instruction-tuning dataset and the TimeLlaMA model series, enabling large language models to predict future events with step-by-step explanations derived from temporal knowledge graphs.

2.3 Diffusion Models on Discrete Data

Diffusion models were first introduced by [27] and have since been widely applied in generative tasks such as image generation [28] and audio generation [29], where they have demonstrated excellent performance. Diffusion-LM [30] applied diffusion models to text processing tasks, while DiffuSeq [31] introduces partial noise during the forward diffusion process. In the field of named entity recognition, DiffusionNER [32] uses diffusion models by treating the entity recognition task as a boundary denoising process. DiffCLR [33] brings diffusion models into knowledge graph reasoning by leveraging its multi-step generation process to inject uncertainty and generate distributions, thus better capturing the multi-dimensional semantic information in queries. The DiffTGK [34] model redefines temporal knowledge graph reasoning as a sequence prediction task by encoding historical events as conditional inputs and gradually adding Gaussian noise to target facts in the forward process to simulate the uncertainty of future events, followed by restoring the target facts through a reverse denoising process. Although many researchers have started exploring the application of diffusion models in the field of knowledge graphs, effectively integrating diffusion models with historical information remains an important area of study.

3 Preliminaries

3.1 Diffusion Models on Discrete Data

Diffusion models are a type of probabilistic model composed of a forward process and a reverse process. The core idea is to represent the input data x_0 as a Markov

chain $\{\mathbf{x}_T, \mathbf{x}_{T-1}, \ldots, \mathbf{x}_0\}$, where each state lies in the real space $\mathbb{R}$ and $\mathbf{x}_T$ follows a Gaussian distribution.

In the forward diffusion process, we first convert w into a continuous embedding $\mathbf{x}_0 \in \mathbb{R}^d$. This is expressed as:

$$\mathbf{x}_0 = \sqrt{\beta_0}\ \mathrm{Embed}(w) + \sqrt{1-\beta_0}\,\epsilon, \tag{1}$$

where $\mathrm{Embed}(\cdot)$ denotes the embedding operation, β_0 controls the amount of noise added in the initial step, and $\epsilon \sim \mathcal{N}(0,1)$ is a random noise drawn from a Gaussian distribution. Then, Gaussian noise is gradually added to the original data $\mathbf{x}_0$ until at diffusion step T the generated sample $\mathbf{x}_T$ approximately follows a Gaussian distribution. Each transition from $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ in the forward process is given by:

$$\begin{aligned} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) &= \mathcal{N}\Big(\mathbf{x}_t;\ \sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\ \beta_t\,\mathbf{I}\Big) \\ &= \sqrt{\beta_t}\,\mathbf{x}_{t-1} + \sqrt{1-\beta_t}\,\epsilon, \quad t \in \{1, \cdots, T\}. \end{aligned} \tag{2}$$

In the reverse process, the model starts from the initial state $\mathbf{x}_T$ and uses a neural network to reconstruct the original data $\mathbf{x}_0$. This is expressed as:

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) = \mathcal{N}\Big(\mathbf{x}_{t-1};\ \mu_\theta(\mathbf{x}_t, t),\ \Sigma_\theta(\mathbf{x}_t, t)\Big), \tag{3}$$

where μ_θ and Σ_θ represent the mean and variance parameters computed by a neural network.

Since the parameterization of the forward process $q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$ does not include any trainable parameters, a training objective is needed to allow the model to learn how to reverse this process using the noise data generated in the forward process, thereby reconstructing the original data. The training objective of the diffusion model is to maximize a variational lower bound on the marginal likelihood $\log p_\theta(x_0)$, which can be expressed as:

$$\mathcal{L}_{\mathrm{vlb}}(\mathbf{x}_0) = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big[\log \frac{q(\mathbf{x}_T \mid \mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t)}{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)\Big]. \tag{4}$$

However, in practice, this objective is often unstable, and various optimization techniques are required for convergence. Therefore, Ho et al. [35] proposed a simplified alternative objective by expanding and reweighting the KL divergence terms in L_{vlb}, which is eventually transformed into a mean squared error (MSE) loss:

$$\mathcal{L}_{\mathrm{simple}}(\mathbf{x}_0) = \sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \Big\| \mu_\theta(\mathbf{x}_t, t) - \hat{\mu}(\mathbf{x}_t, \mathbf{x}_0) \Big\|^2, \tag{5}$$

where $\hat{\mu}(\mathbf{x}_t, \mathbf{x}_0)$ is the mean of the posterior $q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t)$, and $\mu_\theta(\mathbf{x}_t, t)$ is the mean of $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$. To better suit the task of TKG reasoning, Cai et al. [34] extend this simplified MSE loss to the case where the continuous values of $\mathbf{x}_0$ are

approximated or mapped to discrete representations. The objective is expressed as:

$$\begin{aligned}\mathcal{L}_{\text{simple}}^{\text{e2e}}\left(\mathbf{x}_0\right) &= \mathbb{E}_{q_\phi(\mathbf{x}_{0:T}|\mathbf{x}_0)}\left[\sum_{t=2}^{T}\left\|\mathbf{x}_0 - f_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right] \\ &+\mathbb{E}_{q_\phi(\mathbf{x}_{0:1}|\mathbf{x}_0)}\left[\left\|\mathbf{x}_0 - f_\theta\left(\mathbf{x}_1, 1\right)\right\|^2 - \log p_\theta\left(\mathbf{w} \mid \mathbf{x}_0\right)\right],\end{aligned} \tag{6}$$

where the first expectation term is used to train the prediction model $f_\theta(\mathbf{x}_t, t)$ to accurately recover $\mathbf{x}_0$ from steps 2 to T, effectively reducing errors in practice; the second expectation term contains two parts: the first part ensures that the predicted $\mathbf{x}_0$ is close to the embedding Embed(w), while the second part focuses on accurately mapping $\mathbf{x}_0$ back to the discrete text w.

3.2 Task Definition

Temporal knowledge graph extrapolation aims to predict entities at future timestamps. Given a query $q = (s, r, ?, t)$, where $q \in \mathcal{Q}_t$, and the event at timestamp t is unknown, the task is formally defined as computing the conditional probability of the missing object o given the known subject s, relation r, timestamp t, and historical information $G_{t_0:t_i}$ before t:$p(o|s, r, t, G_{t_0:t_i})$ where $t_i < t$.

In this paper, we represent a temporal knowledge graph as $G = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F}\}$, where $\mathcal{E}$, $\mathcal{R}$, $\mathcal{T}$, and $\mathcal{F}$ denote the entity types, relation types, timestamp types, and fact set, respectively. Additionally, a TKG can be viewed as a series of snapshots $\{G_0, G_1, \ldots, G_t, \ldots\}$, where G_t contains all quadruples occurring at timestamp t. The query set is denoted as $\mathcal{Q}$, and a TKG consists of quadruples of the form (s, r, o, t), where $s, o \in \mathcal{E}$, $r \in \mathcal{R}$, and $t \in \mathcal{T}$. The embedding dimension is denoted as d.

4 Methodology

In this section, we provide a detailed description of our DMSA architecture, shown in Figure 2. It mainly consists of three modules: the Selective Attention with CompGCN module, the diffusion module, and the decoding module. The Selective Attention with CompGCN module uses CompGCN and an attention extraction mechanism to capture information from historical snapshots, with a focus on information that is relevant to the current snapshot. The diffusion module comprises two processes: the forward noise propagation process and the reverse denoising process. The decoder employs Time-aware ConvTransE. In the following, we describe each module in detail.

4.1 Selective Attention with CompGCN

Relying solely on static entity representations in TKG reasoning may cause significant temporal information loss. Inspired by [36], we integrate static and dynamic components using timestamp information. Given a sequence of l temporal

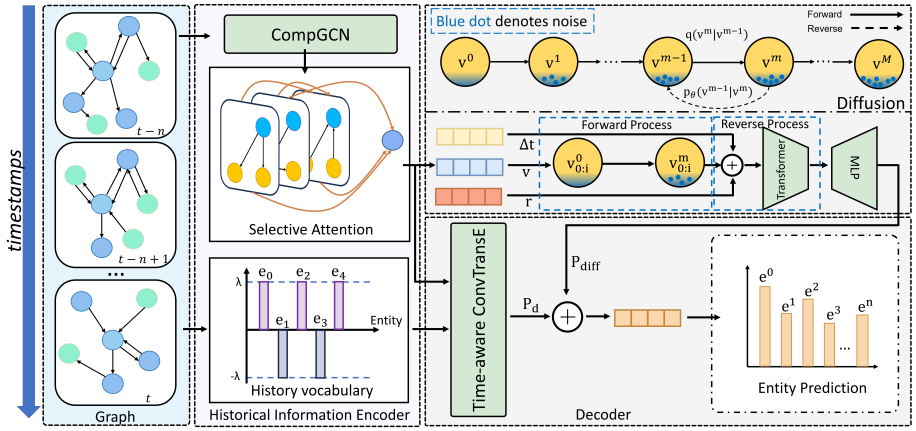


Fig. 2. Overall architecture of DMSA for temporal knowledge graph reasoning. From the input TKG, we first extract the most recent n snapshots as short-term context. Each snapshot is encoded by CompGCN, and Selective Attention filters out irrelevant information. The filtered representations are then combined by the Selective Attention with CompGCN to produce entity and relation embeddings. In parallel, the Diffusion module applies forward noise injection and reverse denoising to model uncertainty in the target entity embedding. Finally, the diffusion-based prediction is concatenated with the Time-aware ConvTransE output, and the fused vector is used for the final fact prediction.

snapshots with query timestamp t, the static embedding $\mathbf{e}_t^s$ captures invariant features of entity s, while the dynamic embedding $\mathbf{e}_t^d$ models temporal variations:

$$\mathbf{e}_t^d = \mathbf{W}_1^e t + \sin(2\pi \mathbf{W}_2^e t), \tag{7}$$

where $\mathbf{W}_1^e \in \mathbb{R}^{1\times d}$ and $\mathbf{W}_2^e \in \mathbb{R}^{1\times d}$ are learnable parameters capturing linear changes and periodic fluctuations, respectively.

The final entity representation is obtained by concatenating the static and dynamic embeddings and transforming them:

$$\mathbf{e}_t = \mathbf{W}_3^e(\mathbf{e}_t^s \oplus \mathbf{e}_t^d), \tag{8}$$

with $\mathbf{W}_3^e \in \mathbb{R}^{d\times 2d}$ adjusting the balance between the two.

To capture evolution, we process l consecutive snapshots using CompGCN:

$$\mathbf{x}_t = \text{CompGCN}(\mathbf{e}_t, \mathbf{r}), \tag{9}$$

yielding node representations $\{\mathbf{X}_{t-l}, \ldots, \mathbf{X}_t\}$ that reflect changes in entities and relations. A GRU then encodes the temporal sequence to capture hidden dependencies:

$$\mathbf{v}_t = \text{GRU}(\mathbf{x}_t, \mathbf{v}_{t-1}), \tag{10}$$

where $\mathbf{v}_t$ is the updated representation of entity s at timestamp t.

Since not all adjacent snapshots are relevant, we need to filter useful information. First, mean pooling is applied to the relation embeddings associated with entity e at time t to form a reference vector:

$$\mathbf{r}_m = \frac{1}{|\mathcal{R}(e_t)|} \sum_{r \in \mathcal{R}(e_t)} \mathbf{r}. \tag{11}$$

Then $\mathbf{r}_m$ is then combined with embeddings from the past l timestamps and passed through a feedforward layer to produce attention weights:

$$\mathbf{B}_j = \text{softmax}\left(\mathbf{W}_b \left(\mathbf{v}_{t-j} + \mathbf{r}_m\right)\right), \quad j \in [0, l], \tag{12}$$

where $\mathbf{W}_b \in \mathbb{R}^{d \times d}$ and $\mathbf{B}_0$ is initialized to zero. Finally, a weighted sum of the current and past embeddings is computed:

$$\mathbf{v}_t^e = \mathbf{B}_0 \mathbf{v}_t + \sum_{j=1}^{l} \mathbf{B}_j \mathbf{v}_{t-j}. \tag{13}$$

Selective attention with CompGCN effectively combines recent interactions with periodic patterns to refine the entity representation over time.

4.2 Diffusion Module

The diffusion process consists of a forward process and a reverse process. We treat the historical information as input to predict the current missing entity. The representations of entities in the historical snapshots are denoted as $\mathbf{v}_{0:l-1}^e \in \mathbb{R}^{(l-1)\times d}$, and the representation of the target object is denoted as $\mathbf{v}_l^e \in \mathbb{R}^{1\times d}$. To better capture the time information, we use relative time representations by calculating the time interval between each snapshot in $Q_{0:l-1} = \{(\mathbf{v}_0^e, \mathbf{r}_0, \mathbf{t}_0), \cdots, (\mathbf{v}_{l-1}^e, \mathbf{r}_{l-1}, \mathbf{t}_{l-1})\}$ and the current snapshot, and encoding these intervals with an embedding function.

Forward Process Following the method in [34], after obtaining the embedding $\mathbf{v}^{e,0}$ of the object sequence, we gradually add randomness to the target object $\mathbf{v}_l^{e,0}$ during the forward process. Specifically, the forward process is constructed as a Markov chain with Gaussian transitions. For each object $\mathbf{v}_i^{e,0}$, we define:

$$q\left(\mathbf{v}_i^{e,m} \mid \mathbf{v}_i^{e,0}\right) = \begin{cases} \mathbf{v}_i^{e,0}, & \text{if } i < l, \\ \sqrt{\bar{\beta}_m}\, \mathbf{v}_i^{e,0} + \sqrt{1-\bar{\beta}_m}\, \epsilon, & \text{if } i = l, \end{cases} \tag{14}$$

$$\bar{\beta}_m = 1 - \delta \cdot (\beta_{\min} + \frac{m-1}{M-1} (\beta_{\max} - \beta_{\min})), \tag{15}$$

where $\epsilon \sim \mathcal{N}(0, 1)$ is a random Gaussian noise,$\delta \in [0, 1]$ controls the noise scale, and $\bar{\beta}_m$ is the cumulative product that controls the noise level at each diffusion step. The diffusion process is performed for $m \in \{1, 2, \cdots, M\}$, where M is the maximum number of forward steps. $\beta_{\min}$ and $\beta_{\max}$ denote the lower and upper bounds of the noise, respectively, with $\beta_{\min} < \beta_{\max} \in (0, 1)$.

Reverse Process In the reverse process, we denoise the noisy representation while using the time and relation information from historical snapshots as conditions. Specifically, we incorporate the encoded information of the relations $\mathbf{r}$ and the time intervals Δt into the denoising process, as expressed by:

$$p_\theta\left(\hat{\mathbf{v}}^{m-1} \mid \hat{\mathbf{v}}^m, \mathbf{r}, \mathbf{t}, m\right) = \mathcal{N}(\hat{\mathbf{v}}^{m-1}; \mu_\theta(\hat{\mathbf{v}}^m, \mathbf{r}, \mathbf{t}, m), \Sigma_\theta(\hat{\mathbf{v}}^m, \mathbf{r}, \mathbf{t}, m)), \quad (16)$$

where $\hat{\mathbf{v}}^{m-1} = \mathbf{v}^m_{0:n-1} \oplus \hat{\mathbf{v}}^{m-1}_n$. In the first step of the reverse process, we set $\hat{\mathbf{v}}^m = \mathbf{v}^m$. At this stage, we use a Transformer architecture to compute $\mu_\theta(\hat{\mathbf{v}}^m, \mathbf{r}, \mathbf{t}, m)$ and $\Sigma_\theta(\hat{\mathbf{v}}^m, \mathbf{r}, \mathbf{t}, m)$. This is represented as:

$$\begin{aligned} f_\theta(\hat{\mathbf{v}}^m, \mathbf{r}, \mathbf{t}, m) &= \hat{\mathbf{v}}^0, \\ \bar{\mathbf{v}}^m &= \hat{\mathbf{v}}^m + \mathbf{r} + \mathbf{t} + \mathbf{m}, \end{aligned} \quad (17)$$

where f_θ denotes the Transformer, and $\mathbf{m}$ represents the step embedding used to adjust the impact of different noise levels [31]. Finally, the final prediction is generated through a fully connected layer:

$$P_{\text{diff}} = \text{softmax}((\text{MLP}(\mathbf{v}_t \oplus \mathbf{r} \oplus \bar{\mathbf{v}}^m))\mathbf{V}^\top + \mathbf{H}_{\text{history}}), \quad (18)$$

where $\mathbf{V}^\top$ represents the evolving representations of all entities output by the selective attention with CompGCN at each moments, $\mathbf{H}_{\text{history}}$ is an embedding representation that records the frequency of entity and relation occurrences in the historical data. We assign a value of λ to the subject and relation pairs with a frequency greater than 0, and λ to the subject and relation pairs with a frequency less than 0. This operation is similar to previous methods [20] [21] [22].

4.3 Time-aware ConvTransE

Prior work has shown that Time-aware ConvTransE is effective as a temporal knowledge graph decoder [17]. We thus adopt it as our decoder backbone. Time-aware ConvTransE takes three inputs: entity embeddings, relation embeddings, and timestamp embeddings. Since the entity and relation embeddings come from previous modules, we next introduce the timestamp embedding.

Timestamp Embedding We represent timestamps by considering both relative and absolute aspects. Specifically, we use a sine function for periodic (relative) changes and a linear function for non-periodic (absolute) changes. These features are fused via element-wise addition:

$$\mathbf{h}_t = \mathbf{h}^p_t + \mathbf{h}^{np}_t, \quad (19)$$

$$\mathbf{h}^p_t = \sin(\mathbf{W}_{1,\omega} t + \mathbf{b}_p), \quad (20)$$

$$\mathbf{h}^{np}_t = \mathbf{W}_{2,\omega} t + \mathbf{b}_{np}, \quad (21)$$

where $\mathbf{W}_{1,\omega}\mathbb{R}^{d\times d}$ and $\mathbf{b}_p$ are learnable parameters for periodic features, and $\mathbf{W}_{2,\omega}\mathbb{R}^{1\times d}$ and $\mathbf{b}_{np}$ are for non-periodic features. The resulting $\mathbf{h}_t$ is used as the timestamp embedding.

Prediction We feed the entity embedding $\mathbf{v}_t$, relation embedding $\mathbf{r}_t$, and timestamp embedding $\mathbf{h}_t$ into Time-aware ConvTransE to get:

$$P_{\mathrm{d}} = \mathrm{softmax}(f(\mathbf{v}_t, \mathbf{r}_m, \mathbf{h}_t)), \tag{22}$$

where $f(\cdot)$ is the mapping function of Time-aware ConvTransE.

To balance the diffusion module and the Time-aware ConvTransE, we fuse their outputs as follows:

$$P = \gamma P_{\mathrm{d}} + (1-\gamma) P_{\mathrm{diff}}, \tag{23}$$

where γ is a hyperparameter, and P_{diff} is the prediction from the diffusion.

4.4 Model Training

The main training goal is to minimize a combined loss:

$$\mathcal{L} = \alpha \mathcal{L}_1 + (1-\alpha)\mathcal{L}_2, \tag{24}$$

where $\alpha \in [0,1]$ balances the losses from the Selective Attention with CompGCN module and the Time-aware ConvTransE. The Selective Attention with CompGCN module minimizes the cross-entropy loss:

$$\mathcal{L}_1 = -\sum_{i=1}^{|\mathcal{Q}(t)|} y_i \log\left(P_{\mathrm{diff}}\right), \tag{25}$$

where $|\mathcal{Q}(t)|$ as the number of queries at timestamp t, y_i the true label for the i-th query, and P_{diff} the predicted probability.

Similarly, the Time-aware ConvTransE minimizes a cross-entropy loss:

$$\mathcal{L}_2 = -\sum_{i=1}^{|\mathcal{Q}(t)|} y_i \log\left(P_{\mathrm{d}}\right), \tag{26}$$

where P_{d} as the predicted probability for the target entity. Jointly optimizing these losses allows the model to capture both historical information and unseen events information, enhancing reasoning on temporal knowledge graphs.

5 Experiments

In this section, we present a series of experiments to evaluate the performance of DMSA. We compare DMSA with various state-of-the-art TKG models. Then we conduct an ablation study to evaluate the effectiveness of different components of the model. Lastly, we explore the impact of hyperparameters on the overall model performance.

Table 1. Statistics of the datasets.

Dataset	Entities	Relations	Training	Validation	Test	Time gap	Snapshots
ICEWS14	12,498	260	323,895	-	341,409	1 day	365
ICEWS18	23,033	256	373,018	45,995	49,545	1 day	304
GDELT	7,691	240	1,734,399	238,765	305,241	15 mins	2,976
YAGO	10,623	10	161,540	19,523	20,026	1 year	189
WIKI	12,554	24	539,286	67,538	63,110	1 year	232

5.1 Setup

Datasets We utilize three real-world event-driven temporal knowledge graph datasets—ICEWS14[11], ICEWS18 [4], and GDELT[37]. Meanwhile, we utilize two widely used public knowledge graph datasets, YAGO [38] and WIKI[39]. ICEWS14 and ICEWS18 are derived from the Integrated Crisis Early Warning System (ICEWS), capturing a wide range of international political events across different periods. GDELT is a dataset sourced from global news media, recording human societal behaviors, while YAGO and WIKI are subsets of YAGO3 and Wikipedia, respectively.

To ensure fair comparisons with baseline models, we follow the dataset partitioning strategies employed in previous studies. For all datasets except ICEWS14, we divide the data into training, validation, and test sets with an 8:1:1 ratio. Since the original ICEWS14 dataset does not provide a validation set, we split it into training and test sets only. The detailed statistics for each dataset are presented in Table 1.

Evaluation Metrics We employ Mean Reciprocal Rank (MRR) and Hits@N as the evaluation metrics, which are standard indicators used to assess the performance of temporal knowledge graph models. In order to maintain consistency with baseline methods, we adopt the same evaluation standards. MRR calculates the mean reciprocal of the rank of the correct answer, while Hits@N measures the proportion of correct predictions ranked within the top N positions. A higher ranking of the correct entity leads to higher MRR and Hits@N values.

Baselines We compare DMSA with several recent approaches, which fall into two main categories: static knowledge graph reasoning methods and temporal knowledge graph reasoning methods. The static reasoning methods include TransE [5], DistMult [6], ComplEx [7], ConvE [8], R-GCN [9], and CompGCN [40]. Temporal knowledge graph extrapolation methods include RE-NET [13], xERTE [41], EvoKG [15], CyGNet [14], HIP [23], RE-GCN [16], TiRGN [17], HGLS [18], CENET [20], LSEN [22], SiMFy [19], and PLEASING [21].

Implementation Details For all experiments, we employ the Adam optimizer with a learning rate of 0.001, a batch size of 1024, and an embedding dimen-

sion of 200. A dropout rate of 0.2 is applied uniformly across all modules to mitigate overfitting, while the number of GNN layers is chosen from $\{1, 2, 3, 4\}$ based on validation performance. In line with prior work [20], we constrain the hyperparameter λ to values in $\{2, 3, 4\}$ and set the hyperparameter α to 0.1, and the length of historical information l is set to match the number of diffusion steps M, chosen from $\{1, 2, 4\}$. All experiments are conducted on an NVIDIA Tesla A100 GPU (40GB) using PyTorch, with 128GB of memory and 100GB of storage. Required environments, codes, and details of commands are available at https://github.com/AAristotle/DMSA.

Table 2. Model performance comparison on five TKG datasets. All values are in percentage (%). The best results are in **bold**, and the second-best are underlined.

Model	ICEWS14			ICEWS18			GDELT			YAGO			WIKI		
	MRR	H@1	H@3	MRR	H@1	H@3	MRR	H@1	H@3	MRR	H@1	H@3	MRR	H@1	H@3
TransE	18.65	1.12	31.34	17.56	2.48	26.95	16.05	0.00	26.10	48.97	46.23	62.45	46.68	36.19	49.71
DistMult	19.06	10.09	22.00	22.16	12.13	26.00	18.71	11.59	20.05	59.47	52.97	60.91	46.12	37.24	49.81
ComplEx	24.47	16.13	27.49	30.09	21.88	34.15	22.77	15.77	24.05	61.29	54.88	62.28	47.84	38.15	50.08
ConvE	40.73	33.20	43.92	36.67	28.51	39.80	35.99	27.05	39.32	62.32	56.19	63.97	47.57	38.76	50.10
R-GCN	26.31	18.23	30.43	23.19	16.36	25.34	23.31	17.24	24.96	41.30	32.56	44.44	37.57	28.15	39.66
CompGCN	26.46	18.38	30.64	23.31	16.52	25.37	23.46	16.65	25.54	41.42	32.63	44.59	37.64	28.33	39.87
EvoKG	18.30	6.30	19.43	29.67	12.92	33.08	11.29	2.93	10.84	55.11	54.37	81.38	50.66	12.21	63.84
xERTE	32.92	26.44	36.58	36.95	30.71	40.38	» 1 day			58.75	58.46	58.85	» 1 day		
RE-NET	45.71	38.42	49.06	42.93	36.19	45.47	40.2	32.43	43.40	65.16	63.29	65.63	51.97	48.01	52.07
CyGNet	48.63	41.77	52.50	46.69	40.58	49.82	50.29	44.53	54.69	63.47	64.26	65.71	45.50	50.48	50.79
RE-GCN	41.61	33.81	44.76	37.92	28.90	41.44	28.66	21.52	30.50	65.69	59.98	68.70	44.86	39.82	46.75
TiRGN	45.13	37.03	48.80	39.58	30.41	43.41	31.58	23.78	33.69	-	-	-	-	-	-
HGLS	40.63	31.97	43.90	39.22	28.96	43.34	» 1 day			59.02	48.17	65.73	49.63	39.62	55.17
HIP	50.57	45.73	54.28	48.37	43.51	51.32	52.76	46.35	55.31	67.55	66.32	68.49	54.71	53.82	54.73
CENET	53.35	49.61	54.07	51.06	47.10	51.92	58.48	55.99	58.63	84.13	84.03	84.23	68.39	68.33	68.36
LSEN	54.82	51.15	55.53	52.12	48.37	52.95	59.47	57.44	59.38	88.07	86.70	88.61	76.13	74.01	76.82
SiMFy	54.81	47.99	58.54	46.87	39.29	51.00	47.40	40.17	50.81	-	-	-	-	-	-
PLEASING	55.82	51.50	56.99	54.98	50.09	56.66	59.12	55.96	59.85	84.36	84.27	84.38	68.13	67.97	68.28
DMSA	**58.41**	**53.54**	**59.62**	**57.08**	**51.40**	**58.95**	**62.49**	**58.29**	**63.44**	**89.20**	**88.10**	**89.50**	**79.32**	**77.31**	**80.08**

5.2 Results

Table 2 presents the entity prediction results on five TKG datasets, where DMSA outperforms most baselines. Static reasoning methods perform poorly because they cannot effectively model temporal dynamics. Although xERTE offers interpretability, it struggles with computational efficiency on large datasets like GDELT. SiMFy, despite its simple structure and fast convergence, delivers suboptimal overall results. LSEN focuses solely on historical data and thus misses future trends, even though it performs well on GDELT where historical correlations are strong. Notably, DMSA improves MRR by 3.02% on GDELT compared to LSEN. Both CENET and PLEASING achieve excellent results through contrastive learning, but their two-stage training leads to high computational overhead. Compared with PLEASING, DMSA boosts MRR by 2.59% on ICEWS14 and by 2.1% on ICEWS18. On two public TKG datasets (YAGO and WIKI),

DMSA achieves the best performance, while CENET and PLEASING show similar results. The YAGO and WIKI datasets rely less on historical snapshots, which poses a challenge for models that depend solely on past data. For example, the low frequency of relevant facts in the WIKI dataset makes it difficult for many models to correctly infer subject entities.

Table 3. Performance comparison on ICEWS14, ICEWS18, YAGO, and WIKI datasets. All values are in percentage. The best score is in bold. w/o denotes without. SA denotes Selective Attention with CompGCN. TC denotes Time-aware ConvTransE.

Model	ICEWS14			ICEWS18			YAGO			WIKI		
	MRR	**H@1**	**H@3**	**MRR**	**H@1**	**H@3**	**MRR**	**H@1**	**H@3**	**MRR**	**H@1**	**H@3**
DMSA	**58.41**	**53.54**	**59.62**	**57.08**	**51.40**	**58.95**	**88.14**	**86.80**	**88.59**	**79.32**	**77.31**	**80.04**
-w/o SA	54.22	50.44	54.98	51.61	47.00	53.20	80.66	78.82	81.59	69.04	68.58	69.13
-w/o Diffusion	57.67	52.58	59.02	55.77	49.93	57.71	85.58	84.77	85.49	78.32	76.23	79.13
-w/o TC	50.96	47.95	50.77	45.14	42.48	44.83	84.47	84.21	84.38	68.78	68.39	68.57

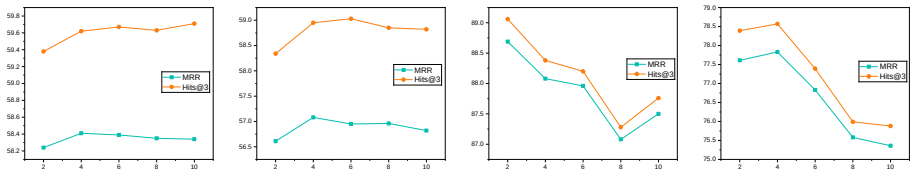
5.3 Ablation

Table 3 shows the ablation study results on four datasets. ICEWS18 and ICEWS14 are event-based knowledge graphs, while YAGO and WIKI are public knowledge graphs, each with its own characteristics. We systematically removed key modules from DMSA to assess their contributions, and overall, the removal of any module led to a drop in performance.

Removing the Selective Attention with CompGCN reduced the model's ability to capture recent historical details, which negatively affected its performance in modeling recent events. This module's removal caused the largest drop in performance, highlighting the critical role of historical data in TKG reasoning. Similarly, removing the Diffusion module resulted in only a small overall decrease in MRR; however, its impact was more noticeable on the YAGO and WIKI datasets than on ICEWS14 and ICEWS18. This difference reflects the varying data characteristics of the datasets and supports our design goal of introducing uncertainty to improve the recognition of unseen entities. In addition, removing the Time-aware ConvTransE module significantly hurt DMSA's prediction performance, further confirming its effectiveness.

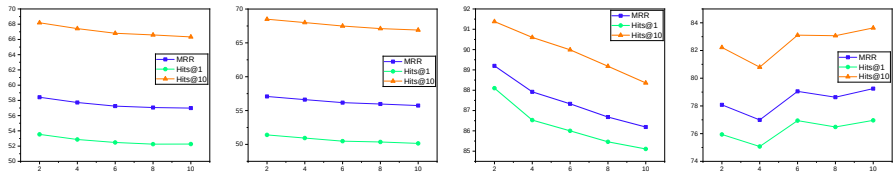
5.4 Sensitivity Analysis

To explore the importance of historical facts on prediction performance, we conducted a sensitivity analysis. First, we examined the impact of historical information length on model performance (as shown in Figure 3). The results indicate that, in the YAGO and WIKI datasets, the model's performance significantly declines as the length of historical information increases, while in the ICEWS



(a) ICEWS14 (b) ICEWS18 (c) YAGO (d) WIKI

Fig. 3. Performance of DMSA under different length of history length l in terms of MRR and Hits@3 (%).



(a) ICEWS14 (b) ICEWS18 (c) YAGO (d) YAGO

Fig. 4. Performance of DMSA under different length of CompGCN layers in terms of MRR Hits@1 and Hits@10 (%).

datasets, performance remains relatively stable with no obvious decline. This suggests that historical information plays a vital role in temporal knowledge graph reasoning.

Next, we analyze the impact of the number of CompGCN layers on model performance, as shown in Figure 4. As the number of layers increases, the performance of DMSA decreases. This is due to the problem of over-smoothing caused by too many CompGCN layers. On the YAGO dataset, the model performance declines significantly as the number of CompGCN layers increases. In ICEWS18, the performance decreases more slowly. The WIKI dataset requires multiple CompGCN layers to extract features, and as the number of layers increases, performance improves. By adjusting the number of CompGCN layers on the validation set, DMSA can achieve the best performance.

6 Conclusion

In this paper, we propose DMSA, a novel model for temporal knowledge graph reasoning. DMSA leverages historical information and introduces noise to enhance the model's ability to predict unseen facts in knowledge graphs. By incorporating a selective attention mechanism, the model can focus on the historical information that is most relevant to the current query. Entities are predicted through decoding using Time-aware ConvTransE. Experimental results show that DMSA significantly outperforms existing methods, highlighting its promise for advancing temporal knowledge graph reasoning. In future work, we will focus on addressing the uncertainty in entity representations within temporal graphs

to better capture event evolution and improve the modeling of unseen events. Additionally, we plan to explore the use of pre-trained language models to further enhance the semantic representation of entities.

Acknowledgments.

This work was supported by the National Natural Science Foundation of China under Grant No. 72210107001, the Beijing Natural Science Foundation under Grant No. IS23128, the Fundamental Research Funds for the Central Universities, and the CAS PIFI International Outstanding Team Project (Grant No. 2024PG0013).

References

1. X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 950–958.
2. A. Saxena, A. Tripathi, and P. Talukdar, "Improving multi-hop question answering over knowledge graphs using knowledge base embeddings," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 4498–4507.
3. M. Zamiri, Y. Qiang, F. Nikolaev, D. Zhu, and A. Kotov, "Benchmark and neural architecture for conversational entity retrieval from a knowledge graph," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 1519–1528.
4. W. Jin, M. Qu, X. Jin, and X. Ren, "Recurrent event network: Autoregressive structure inference over temporal knowledge graphs," *arXiv preprint arXiv:1904.05530*, 2019.
5. A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.
6. B. Yang, S. W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015.
7. T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.
8. T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.
9. M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*. Springer, 2018, pp. 593–607.
10. S. Vashishth, S. Sanyal, V. Nitin, and P. P. Talukdar, "Composition-based multi-relational graph convolutional networks," *ArXiv*, vol. abs/1911.03082, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:207847719
11. A. Garcia-Duran, S. Dumančić, and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4816–4821.

12. J. Wu, M. Cao, J. C. K. Cheung, and W. L. Hamilton, "Temp: Temporal message passing for temporal knowledge graph completion," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 5730–5746.
13. W. Jin, M. Qu, X. Jin, and X. Ren, "Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6669–6683.
14. C. Zhu, M. Chen, C. Fan, G. Cheng, and Y. Zhang, "Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 4732–4740.
15. N. Park, F. Liu, P. Mehta, D. Cristofor, C. Faloutsos, and Y. Dong, "Evokg: Jointly modeling event time and network structure for reasoning over temporal knowledge graphs," in *Proceedings of the fifteenth ACM international conference on web search and data mining*, 2022, pp. 794–803.
16. Z. Li, X. Jin, W. Li, S. Guan, J. Guo, H. Shen, Y. Wang, and X. Cheng, "Temporal knowledge graph reasoning based on evolutional representation learning," in *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 2021, pp. 408–417.
17. Y. Li, S. Sun, and J. Zhao, "Tirgn: Time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning." in *IJCAI*, 2022, pp. 2152–2158.
18. M. Zhang, Y. Xia, Q. Liu, S. Wu, and L. Wang, "Learning long-and short-term representations for temporal knowledge graph reasoning," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2412–2422.
19. Z. Liu, L. Tan, M. Li, Y. Wan, H. Jin, and X. Shi, "Simfy: A simple yet effective approach for temporal knowledge graph reasoning," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 3825–3836.
20. Y. Xu, J. Ou, H. Xu, and L. Fu, "Temporal knowledge graph reasoning with historical contrastive learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 4765–4773.
21. J. Zhang, M. Sun, Q. Huang, and L. Tian, "Pleasing: Exploring the historical and potential events for temporal knowledge graph reasoning," *Neural Networks*, vol. 179, p. 106516, 2024.
22. F. Wang, G. Zhu, H. Hou, C. Yuan, and Y. Huang, "Mining long short-term evolution patterns for temporal knowledge graph reasoning," in *International Conference on Pattern Recognition*. Springer, 2024, pp. 227–242.
23. Y. He, P. Zhang, L. Liu, Q. Liang, W. Zhang, and C. Zhang, "Hip network: Historical information passing network for extrapolation reasoning on temporal knowledge graph," *arXiv preprint arXiv:2402.12074*, 2024.
24. W. Xu, B. Liu, M. Peng, X. Jia, and M. Peng, "Pre-trained language model with prompts for temporal knowledge graph completion," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 7790–7803.
25. J. Wang, S. Kai, L. Luo, W. Wei, Y. Hu, A. W.-C. Liew, S. Pan, and B. Yin, "Large language models-guided dynamic adaptation for temporal knowledge graph reasoning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 8384–8410, 2024.
26. C. Yuan, Q. Xie, J. Huang, and S. Ananiadou, "Back to the future: Towards explainable temporal reasoning with large language models," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 1963–1974.

27. J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. pmlr, 2015, pp. 2256–2265.
28. N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 500–22 510.
29. Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, "Audiolm: a language modeling approach to audio generation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 31, pp. 2523–2533, 2023.
30. X. Li, J. Thickstun, I. Gulrajani, P. S. Liang, and T. B. Hashimoto, "Diffusion-lm improves controllable text generation," *Advances in neural information processing systems*, vol. 35, pp. 4328–4343, 2022.
31. S. Gong, M. Li, J. Feng, Z. Wu, and L. Kong, "Diffuseq: Sequence to sequence text generation with diffusion models," *arXiv preprint arXiv:2210.08933*, 2022.
32. Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "Diffusionner: Boundary diffusion for named entity recognition," *arXiv preprint arXiv:2305.13298*, 2023.
33. Y. Liu, Y. Cao, S. Wang, Q. Wang, and G. Bi, "Generative models for complex logical reasoning over knowledge graphs," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 492–500.
34. Y. Cai, Q. Liu, Y. Gan, C. Li, X. Liu, R. Lin, D. Luo, and J. JiayeYang, "Predicting the unpredictable: Uncertainty-aware reasoning over temporal knowledge graphs via diffusion process," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 5766–5778.
35. J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
36. K. Wang, S. C. Han, and J. Poon, "Re-temp: Relation-aware temporal representation learning for temporal knowledge graph completion," *arXiv preprint arXiv:2310.15722*, 2023.
37. K. Leetaru and P. A. Schrodt, "Gdelt: Global data on events, location, and tone, 1979–2012," in *ISA annual convention*, vol. 2, no. 4. Citeseer, 2013, pp. 1–49.
38. F. Mahdisoltani, J. Biega, and F. Suchanek, "Yago3: A knowledge base from multilingual wikipedias," in *7th Biennial Conference on Innovative Data Systems Research*. CIDR Conference, 2014.
39. J. Leblay and M. W. Chekol, "Deriving validity time in knowledge graph," in *Companion proceedings of the the web conference 2018*, 2018, pp. 1771–1776.
40. S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in *International Conference on Learning Representations*.
41. Z. Han, P. Chen, Y. Ma, and V. Tresp, "Explainable subgraph reasoning for forecasting on temporal knowledge graphs," in *International conference on learning representations*, 2020.