A Generative Adaptive Replay Continual Learning Model for Temporal Knowledge Graph Reasoning

Zhiyu Zhang[1,3], Wei Chen[2*], Youfang Lin[1,3], Huaiyu Wan[1,3]
[1]School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
[2]Guilin University of Electronic Technology,
School of Computer Science and Information Security, Guangxi, China
[3]Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing, China
{zyuzhang, yflin, hywan}@bjtu.edu.cn, w_chen@guet.edu.cn

Abstract

Recent Continual Learning (CL)-based Temporal Knowledge Graph Reasoning (TKGR) methods focus on significantly reducing computational cost and mitigating catastrophic forgetting caused by fine-tuning models with new data. However, existing CL-based TKGR methods still face two key limitations: (1) They usually one-sidedly reorganize individual historical facts, while overlooking the historical context essential for accurately understanding the historical semantics of these facts; (2) They preserve historical knowledge by simply replaying historical facts, while ignoring the potential conflicts between historical and emerging facts. In this paper, we propose a **D**eep **G**enerative **A**daptive **R**eplay (DGAR) method, which can generate and adaptively replay historical entity distribution representations from the whole historical context. To address the first challenge, historical context prompts as sampling units are built to preserve the whole historical context information. To overcome the second challenge, a pre-trained diffusion model is adopted to generate the historical distribution. During the generation process, the common features between the historical and current distributions are enhanced under the guidance of the TKGR model. In addition, a layer-by-layer adaptive replay mechanism is designed to effectively integrate historical and current distributions. Experimental results demonstrate that DGAR significantly outperforms baselines in reasoning and mitigating forgetting.

1 Introduction

Temporal Knowledge Graphs (TKGs) extend traditional Knowledge Graphs (KGs) by associating triples with timestamps (Leblay and Chekol, 2018; Dasgupta et al., 2018; Lacroix et al., 2020), providing dynamic and structured time-sensitive knowledge for various downstream applications (Chen et al., 2023; Gutiérrez et al., 2024; Wang et al., 2024; Zhao et al., 2025), such as Large Language Models reasoning, event prediction, and financial forecasting (Guan et al., 2022). Unfortunately, TKGs often suffer from incompleteness, hindering the capability of dynamic knowledge representation in downstream applications. Temporal knowledge Graph Reasoning (TKGR) is proposed to address this issue by inferring missing temporal facts based on historical knowledge.

In real-world scenarios, TKGs are continuously updated with unseen entities, relations, and new facts. Existing TKGR studies (Leblay and Chekol, 2018; Li et al., 2021, 2022b; Xu et al., 2023a) update model parameters by retraining on the entire TKG when new data arrives. This process is computationally expensive and impractical for dynamic settings, especially in the transportation and finance domains where frequent knowledge updates are required (Liu et al., 2024a). Continual Learning(CL) fine-tuning models with new data may seem intuitive, but often results in catastrophic forgetting, where prior knowledge is lost (Mirtaheri et al., 2023).

To mitigate catastrophic forgetting, recent CL-based TKGR studies (Wu et al., 2021; Mirtaheri et al., 2023) rely on the mechanisms of replaying prior knowledge, further employing regularization techniques to preserve old knowledge. These studies integrate new knowledge while preserving previously acquired information, thus enabling reasoning over both historical and emerging data.

Despite notable progress, the currently CL-based TKGR methods still face two primary challenges:

(1) These methods often reorganize and replay the historical data (e.g., based on frequency or clustering) to mitigate catastrophic forgetting. However, such methods solely focus on the statistical properties of individual historical events, which fails to correctly understand the historical semantics of these facts combining the necessary histor-

[*]Corresponding author

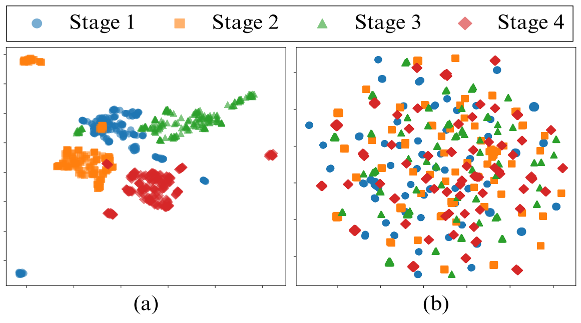


Figure 1: The distributions of the same set of entity features at different timestamps are visualized using U-MAP visualization. Entities involved in all facts at a randomly selected time i are extracted. Stage 1 represents the feature distribution of these entities at time i, while stage 2, 3, and 4 respectively correspond to their feature distributions at times j, m and n, where $i < j < m < n$. (a) depicts the distributions learned by the base model across these timestamps, and (b) shows the distributions learned by DGAR. The results demonstrate that our approach effectively resolves distribution conflicts while preserving historical knowledge.

ical context. Besides, this fragmented approach makes it difficult to capture the overarching trends of entity behavior, thus limiting the model's capacity in complex reasoning tasks.

(2) Current approaches typically replay historical data directly overlooking potential conflicts between the distributions of historical and current data (e.g., Figure 1). As entities associate with different neighbors over time, semantic differences arise, which in turn cause conflicts in the distributions of entities at different times. This oversight hinders the effectiveness of mitigating catastrophic forgetting.

To address these challenges, we propose a Deep Generative Adaptive Replay (DGAR) method for TKGR, which can continually and adaptively replay historical information by generating the historical distribution representation of entities from the whole historical context. For the first challenge, instead of using individual facts as sampling units, we build Historical Context Prompts (HCPs) as sampling units to retain the context information of historical data. For the second challenge, we enhance the common features across different distributions and introduce a deep adaptive replay mechanism to mitigate distribution conflicts. Specifically, we design a Diffusion-Enhanced Historical Distribution Generation (Diff-HDG) strategy that generates entity historical distribution representations. During the generation process, the features of the entity's historical distribution that are common to the entity's current distribution are enhanced. In addition, a layer-by-layer Deep Adaptive Replay (DAR) mechanism is introduced to inject the entity's historical distribution representation into its current distribution representation.

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

- We propose a novel Generative Adaptive Replay Continual Learning method for TKGR, which effectively addresses the issue of knowledge forgetting by incorporating the entire historical context and mitigating distribution conflicts.
- A sophisticated historical context prompt is designed for replay data sampling, ensuring the semantic integrity of the historical context information in the sampled facts.
- A Diff-HDG strategy is proposed to generate historical distribution representations by enhancing the common features. In addition, a DAR mechanism is designed to efficiently integrate historical and current distributions.
- Extensive experiments conducted on widely used TKGR datasets demonstrate the superiority of our approach, consistently outperforming all baseline methods across various metrics.

2 Related Work

2.1 Reasoning on TKGs

TKGR aims to infer missing facts by utilizing known facts. Recent advancements in this field fall into four main approaches. The distribution-based approaches (Leblay and Chekol, 2018; Lacroix et al., 2020) perform reasoning by training a scoring function that can evaluate the distance or semantic similarity between entities. The Graph Neural Network (GNN)-based methods (Li et al., 2021, 2022c; Xu et al., 2023b; Chen et al., 2024b; Wu et al., 2023; Chen et al., 2024c) capture structural and temporal patterns in graph sequence to enhance reasoning accuracy. Rule-based temporal knowledge graph reasoning methods (Huang et al., 2024; Chen et al., 2024a) follow a symbolic paradigm that emphasizes interpretability, logical consistency, and low resource requirements. These methods typically mine temporal logical rules from historical fact sequences, then use these rules to

[1]The source code of DGAR is available at: https://github.com/zyzhang11/DGAR.

infer future events or fill in missing historical facts. When new data arrives, these methods often require retraining. Given the strong performance of GNN-based methods in TKGR, our approach builds upon this category of methods.

2.2 CL for Knowledge Graphs

Compared to existing approaches that necessitate repeated retraining, CL adaptively incorporates sequentially evolving knowledge. Recently, several methods have applied CL to knowledge graph embedding (KGE) and TKGR. For instance, some approaches (Wu et al., 2021; Mirtaheri et al., 2023) integrate experience replay with regularization techniques to address catastrophic forgetting in TKGR. TIE's (Wu et al., 2021) overly restrictive regularization leads to a decline in overall performance. The regularization method restricts the applicability of DEWC (Mirtaheri et al., 2023) to a limited number of tasks. (Cui et al., 2023; Liu et al., 2024a,b) apply CL to KGE by employing regularization constraints to retain historical knowledge, effectively mitigating catastrophic forgetting.

2.3 Diffusion Models

Diffusion models are generative frameworks that reconstruct structured data from Gaussian noise through a stepwise reverse denoising process (Sohl-Dickstein et al., 2015; Ho et al., 2020). **In continuous domains** like image synthesis, DDPM and its variants effectively model complex distributions and generate high-quality outputs (Ho et al., 2020; Rombach et al., 2022). Applying diffusion models to **discrete domains** is challenging due to Gaussian noise's incompatibility with discrete structures. Text generation employs polynomial diffusion or continuous-to-discrete mapping to link continuous processes with discrete data (Austin et al., 2021; Gong et al., 2022; Li et al., 2022a). **In KGs**, (Long et al., 2024; Cai et al., 2024) restore knowledge from noise by mapping discrete KG data to a continuous space and applying conditional constraints.

3 Preliminaries

3.1 The Task of TKGR

TKG can be represented as a sequence of snapshots partitioned by time, denoted as $\mathcal{G} = \{G_1, G_2, G_3, ..., G_T\}$. Each snapshot $G_t = (\mathcal{V}, \mathcal{R}, \mathcal{F}_t)$ is a directed multi-relational graph at timestamp t. $(s, r, o, t) \in \mathcal{F}_t$ is denoted as a fact, where $s \in \mathcal{V}$ and $o \in \mathcal{V}$ are subject entity and object entity, respectively, $r \in \mathcal{R}$ is denoted as a relation that connects the subject entity and the object entity.

The task of TKGR aims to predict the missing object entity (or subject entity) given a query $(s_q, r_q, ?, t_q)$ or $(?, r_q, o_q, t_q)$. To be consistent with common representation, the inverse quadruple of a fact (s, r, o, t) is (o, r^{-1}, s, t) which is added to the dataset. The TKG reasoning goal can be expressed as the prediction of object entities.

3.2 Continual Learning for TKGR

Under the CL setting, TKGs can be viewed as a sequence of KG snapshots arriving as a stream over time. A set of tasks can be denoted as $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T\}$, where each task is denoted as $\mathcal{T}_t = (D^t_{train}, D^t_{valid}, D^t_{test})$, where $G_t = [D^t_{train} : D^t_{valid} : D^t_{test}]$. The model parameters are updated sequentially for each task as the task stream $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T\}$ arrives. The trained model parameters at each step can be represented as $\{\theta_1, \theta_2, ..., \theta_T\}$. At time t, the parameters θ_t are initialized by the parameters θ_{t-1} at the previous time. Then the model is trained on D^t_{train} to update the parameters.

During CL for TKGR, we mitigate catastrophic forgetting based on KG snapshot sequence reasoning models, such as RE-GCN (Li et al., 2021). We focus primarily on entity representations, as the semantics of entities tend to evolve more frequently over time, in contrast to the relatively negligible changes in the semantics of relations (Goel et al., 2020).

3.3 Denoising Diffusion Probabilistic Model

The Diffusion Model (DM) consists of a forward diffusion process and a reverse diffusion process. In the forward process, a continuous DM is adapted to handle discrete facts $\mathcal{G}$. Given discrete data x, we first project x into a continuous embedding, denoted as $X_0 = \text{Embedding}(x)$, where $X_0 \in \mathbb{R}^d$. $\text{Embedding}(\cdot)$ is a function that can map a word to a vector in $\mathbb{R}^d$. Then a Markov chain of latent variables $X_1, X_2, ..., X_n$ is generated in the forward process by gradually adding small amounts of standard Gaussian noise to the sample. This process can be obtained by:

$$q(X_n \mid X_{n-1}) = \mathcal{N}\left(X_n; \sqrt{1-\beta_n} X_{n-1}, \beta_n I\right), \quad (1)$$

where $\beta_n, n \in [1, ..., N]$ is a noise schedule used to control the step size of the added noise and I is an identity matrix. $\mathcal{N}$ is the Gaussian distribution.

In the reverse process, the standard Gaussian representation X_t progressively approximates the

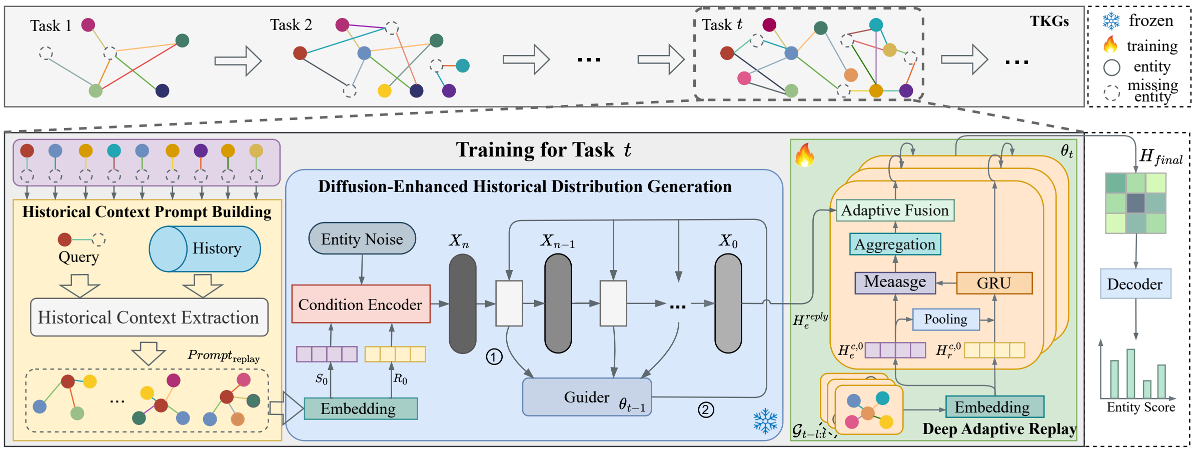


Figure 2: The overall architecture diagram of DGAR. Following the CL paradigm, each snapshot of the TKGs is treated as a separate task.

true representation X_0 by iterative denoising. It can be learned by a parameterized model:

$$p_\phi\left(\mathrm{X}_{n-1} \mid X_n, n\right)=\mathcal{N}\left(X_{n-1} ; \mu_\phi\left(\mathrm{X}_n, n\right), \Sigma_\phi\left(\mathrm{X}_n, n\right)\right), \tag{2}$$

where μ_ϕ and Σ_ϕ are generally implemented by a deep neural $f_\phi(\cdot)$, such as Transformer or U-Net. Inspired by the success of Transformer encoders in the field of graph data(Hu et al., 2020), we opt for the Transformer architecture in this work. The pretraining objectives are defined as follows:

$$\mathcal{L}=\mathbb{E}_q\left[\sum_{n=2}^N\left\|X_0-f_\phi\left(X_n, n\right)\right\|^2\right]-\log p_\phi\left(x \mid X_0\right), \tag{3}$$

where $\mathbb{E}_q$ is the expectation over joint distribution. It is important to emphasize that the data employed in the testing phase remains entirely unseen during the pretraining process.

4 The DGAR Method

The overall architecture of DGAR is shown in Figure 2, primarily consisting of three parts: Historical Context Prompt Building, Diffusion-Enhanced Historical Distribution Generation, and Deep Adaptive Replay.

Initially, when a new query arrives for the t-th task, DGAR builds HCPs based on the queried entity (Section 4.1). Based on the obtained HCPs, we adopt the latest TKGR model parameters θ_{t-1} to guide the historical distribution representation generation of entities (Section 4.2). To support the following reasoning, DAR injects generated historical entity distribution into the current entity distribution representation (Section 4.3). Finally, we present the final loss function of DGAR (Section 4.4).

4.1 Historical Context Prompt Building

Historical context prompts, serving as sampling units of replay data, aim to accurately preserve entities' complete historical semantics. Before constructing an HCP, it is critical to determine which entities at time t are most relevant to achieving the ultimate goal of mitigating catastrophic forgetting. As a new query $(e_q, r_q, ?, t)$ arrives, e_q is an involved entity and its semantics will be directly influenced after fine-tuning at the current timestamp t (Zhang et al., 2023a).

To correctly reflect the historical context semantics of e_q, we construct an HCP for entity e_q. If e_q appears at time $t-1$ or earlier, it might be associated with one or more entities in the past. The historical distribution of e_q is determined by the entities historically associated with e_q (Xing et al., 2024). The HCP building for e_q can be formalized as follows:

$$\begin{aligned} Prompt_{\text{replay}}^i= & \left\{(s, r, e_q) \mid (s, r, e_q) \in G_i\right. \\ & \left.\text{or}\left(e_q, r^{-1}, s\right) \in G_i, G_i \in \mathcal{G}\right\}, \end{aligned} \tag{4}$$

where $Prompt_{\text{replay}}^i$ is the HCP of entity e_q at time i, which denotes the set of triples associated with e_q at historical moment i. The triples in $Prompt_{\text{replay}}^i$ consist of the entity e_q, the neighbor s associated with e_q at time i, and the relation r between e_q and s at time i. When no triple containing e_q appears at time i, $Prompt_{\text{replay}}^i$ is empty.

To reduce the computational and storage burden, we do not select the HCP of e_q across its entire history. Instead, we treat a HCP as the sampling unit, and randomly select HCPs of k distinct time to enhance the generalizability of the replay data. The discussion about k is provided in Appendix B.4. The set of HCP after sampling is denoted as $Prompt_{\text{replay}}$, which serves as the prompt for generating entity historical distribution in Section 4.2. The entities involved in $Prompt_{\text{replay}}$ are represented as a set V_{replay} , these entities are directly or indirectly influenced by newly arrived data:

$$\begin{aligned} V_{\text{replay}} = & \left\{e \mid \forall (e,r,o) \in Prompt_{\text{replay}}\right\} \\ & \cup \left\{e \mid \forall (s,r,e) \in Prompt_{\text{replay}}\right\}, \end{aligned} \tag{5}$$

4.2 Diffusion-enhanced Historical Distribution Generation

The target of Diff-HDG strategy is to generate the historical distribution of entity with minimal conflicts against the current distribution of entity. For this purpose, during the generation process, common features between the historical and current distributions need to be enhanced. In addition, features in the historical distribution of entity that differ from the current distribution of entity need to be weakened. Motivated by previous work (Yang et al., 2023; Voynov et al., 2023; Zhang et al., 2023b), pre-trained DMs have demonstrated exceptional capabilities in reproducing knowledge from prompt texts. DMs possess a robust ability to generate generalized expressions. This capability is crucial for resolving conflicts that arise between different distributions. Thus, we generate historical distribution representations of entities through a pre-trained DM based on HCP. The generation of entity historical distribution primarily relies on the inverse diffusion process, which can be outlined as follows:

$$H_e^{\text{replay}} = p_\phi \left(X_n, f_{\theta_t}, Prompt_{replay}\right), \tag{6}$$

where X_n denotes the object to denoising, which is processed iteratively to yield the historical distribution of entity H_e^{replay} . The function f_{θ_t} represents the parameters of the TKGR model at the current time t.

In detail, for a fact $(s, r, e_q) \in Prompt_{replay}$, we treat the entity s and the relation r as generation conditions. This condition-based generation method integrates information from historical neighbors and relations, enabling a more precise modeling of the historical distribution of entity:

$$X_n = \text{Condition}\left(S_0, R_0, Z\right), Z \sim \mathcal{N}(\mathbf{0}, I), \tag{7}$$

where $S_0 = \text{Embedding}\,(s)$, $R_0 = \text{Embedding}\,(r)$, and $\text{Condition}\,(\cdot)$ represents concatenation. The condition in X_n can guide the DM to generate distributions that reflect the historical semantics of entities.

To enhance the common features between historical and current distribution representations, we propose a novel method to guide the generation process of historical entity representations:

$$X_{n-1} = p_\phi \left(X_n\right), \tag{8}$$

$$X_{n-1} = X_{n-1} + \gamma \frac{\partial \sigma}{\partial X_{n-1}}, \tag{9}$$

$$\frac{\partial \sigma}{\partial X_{n-1}} = \nabla_{X_{n-1}} \sigma \left(f_{\theta_t}\left(X_{n-1}, (s, r, e_q)\right)\right), \tag{10}$$

where σ denotes the softmax function, γ is a hyperparameter, and X_{n-1} is the result of the first denoising step performed on X_n. This process produces a cleaner representation of X_n. After acquiring X_{n-1}, we evaluate the scores of historical facts in $Prompt_{replay}$ with the current TKGR model f_{θ_t}. The gradient of scores is applied to optimize the generated historical distribution, ensuring that the scores of these historical facts are maximized at the current time. Based on our empirical observations, adjacent timestamps in TKGs show only minor distribution differences. Since the model parameters θ_t for the current time can only be obtained after being updated at the current time, we approximate θ_t using θ_{t-1}.

After n iterations of denoising with p_ϕ, we obtain the generated representation $X_0^{e_q}$ for the query entity. Similarly, the historical neighboring entity s receives an updated representation X_0^s, influenced by the query entity $X_0^{e_q}$. Mean pooling is used to aggregate information from multiple neighbors across different timestamps, as shown below:

$$H_e^{\text{replay}} = \frac{\sum_{i=1}^{k} \sum_{\epsilon \in M_e^i} H_\epsilon^i}{\sum_{i=1}^{k} |M_e^i|}, \tag{11}$$

where H_e^{replay} represents the final historical distribution representation of entity $e \in V_{\text{replay}}$, capturing its historical characteristics in the TKGs. H_ϵ^i represents the entity representation X_0^e generated from the facts ϵ. M_e^i refers to the set of facts that contain entity e at the i-th time slice. After iterative denoising, the features in H_e^{replay} that are the same as the current distribution are enhanced, and the features in H_e^{replay} that are different from the current distribution are weakened. These historical entity representations are generated in parallel to improve computational efficiency.

4.3 Deep Adaptive Replay

In this section, we introduce a DAR mechanism that effectively integrates the historical and current distributions of entities. Building upon the historical distribution representation of entities obtained in section 4.2, these representations are incorporated into the current distribution representation of entities.

We identify that overly complex historical knowledge injection mechanisms impose a considerable learning burden, whereas excessively simplistic approaches result in significant knowledge loss. To overcome these issues, we propose DAR for historical knowledge replay, which performs the following operations at each layer of the KG snapshot sequence reasoning model:

$$H_e^l = \begin{cases} H_e^{\text{current},l}, & e \notin V_{\text{replay}} \\ \alpha H_e^{\text{replay}} + (1-\alpha) H_e^{\text{current},l}, e \in V_{\text{replay}} \end{cases}, \quad (12)$$

where $\alpha \in [0, 1]$, which adaptively balances new and old knowledge. $H_e^{\text{current},l}$ denotes the entity distribution representation of the current task at layer l. To preserve the evolutionary characteristics in the temporal sequence, deep replay is conducted within the L evolution units; the final entity representation, denoted as H_{final} , is obtained.

4.4 Model Training

After obtaining the final representation, the decoder computes scores for candidate entities. We treat entity prediction as multi-class classification, and model parameters θ_t for task t are optimized as follows:

$$\mathcal{L}_{t,c} = - \sum_{(s,r,o,t) \in D_{\text{train}}^t} y_t^e f_{\theta_t}(s, r, o, t), \quad (13)$$

where y_t^e represents the label vector. During experiments, we observe that although we attempt to preserve historical knowledge by enhancing common features between the representations of current and historical distributions, the model still suffers from historical information loss. This is primarily due to the constraints of the guidance function and subsequent optimization for current data. To address this issue, we incorporate facts from the historical context prompt as a regularization term into the loss function. The final loss calculation is formulated as follows:

$$\mathcal{L}_t = \mathcal{L}_{t,c} + \mu \mathcal{L}_{t,r}, \quad (14)$$

where $\mathcal{L}_{t,c}$ represents the training loss for the current task, $\mathcal{L}_{t,r}$ denotes the loss associated with replaying historical facts, and μ is a hyperparameter, typically set to 1. The computation of $L_{t,r}$ is similar to that of $L_{t,c}$. The difference is that $L_{t,c}$ calculates the loss based on current facts, while $L_{t,r}$ uses historical fact in $Prompt_{replay}$.

5 Experiments

5.1 Experimental Setup

Datasets. We adopt four widely used benchmark datasets for TKGR tasks: ICE14, ICE18, ICE05-15, and GDELT. The first three datasets originate from the Integrated Crisis Early Warning System (Jin et al., 2020), which records geopolitical events. The statistical details of these datasets are summarized in Table B.1.

Metrics. We utilize two evaluation metrics: Mean Reciprocal Rank (MRR) and Hits@k (k=1,10), both of which are widely adopted to assess the performance of TKGR methods. Following the approach (Mirtaheri et al., 2023), we evaluate the model's ability to mitigate catastrophic forgetting. We evaluate the model trained on the final task t by testing its performance on the current test set (Current) and calculating its average performance across all previous test sets (Average).

Baselines. We adopt the following baselines: FT, ER (Rolnick et al., 2019), TIE (Wu et al., 2021), LKGE (Cui et al., 2023) and IncDE (Liu et al., 2024a). Details about these baselines are provided in Appendix B.2. In the experiments, we use RE-GCN as the base model.

5.2 Main Results

The results of main experiments are shown in Table 1 and Table 2. Each dataset is tested five times and the average results are reported. The same procedure is also followed in subsequent experiments.

DGAR achieves consistent performance improvements compared to Fine-tuning. For historical tasks, it achieves an average increase of 11.34% in MRR. This demonstrates that, compared to direct fine-tuning, our approach more effectively retains historical knowledge.

Moreover, DGAR consistently outperforms all baselines. Compared to the strongest baseline, it achieves an average MRR improvement of 4.01% in the current task across all evaluated datasets. For historical tasks, the average improvement is 8.23% in MRR, and 9.79% in Hits@10. DGAR demonstrates improvements across various datasets by preserving historical knowledge.

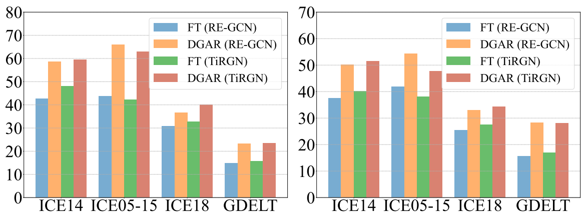
In contrast, while the TIE model performs well on current tasks, it exhibits poor average performance across all historical tasks. This result is

Algo.	ICE14				ICE18				ICE05-15			
	Current	Average			Current	Average			Current	Average		
	MRR	MRR	Hits@1	Hits@10	MRR	MRR	Hits@1	Hits@10	MRR	MRR	Hits@1	Hits@10
FT	42.66	37.46	26.95	58.20	30.76	25.35	15.97	44.36	43.80	41.88	30.55	63.82
ER	48.75	42.14	31.03	63.80	30.39	27.20	16.88	48.19	52.50	45.55	33.34	69.07
TIE	53.74	41.07	30.28	62.39	34.45	28.73	18.40	49.60	60.77	42.56	30.90	64.67
LKGE	43.56	37.51	27.13	58.51	31.12	25.56	16.12	44.70	43.28	42.46	30.99	64.51
IncDE	45.03	36.57	26.20	56.95	31.83	25.52	16.07	44.74	46.33	40.56	29.34	62.17
DGAR	**58.59**	**50.12**	**39.36**	**70.48**	**36.53**	**33.00**	**21.74**	**55.63**	**66.01**	**54.33**	**43.11**	**75.13**

Table 1: The main experimental results on the ICE14, ICE18, and ICE05-15 datasets are presented. Bolded scores indicate the best results.

Algo.	GDELT			
	Current	Average		
	MRR	MRR	Hits@1	Hits@10
FT	14.74	15.60	8.73	29.05
ER	15.42	16.21	8.97	30.42
TIE	15.56	16.40	8.94	30.98
LKGE	14.43	15.52	8.69	28.90
IncDE	15.14	15.49	8.64	28.86
DGAR	**23.25**	**28.30**	**17.38**	**51.39**

Table 2: The main experimental results on the GDELT.



(a) Current (b) Average

Figure 3: Performance of different base TKGR models.

attributed to the strict regularization of TIE, which limits its ability to retain historical knowledge. IncDE and LKGE only use the embedding constraint model of entities and relations at the previous moment to retain old knowledge, which leads to the decline of IncDE's performance on historical tasks compared to FT. LKGE additionally considers the constraints of cumulative weights, so it has a slight improvement on certain datasets compared to FT.

5.3 Ablation Study

In this section, we examine the impact of various components of the model on the final result, as shown in Table 3. To thoroughly evaluate their roles, we implement the following model variants: (1) w/o HP, where the HCP is replaced with ER; (2) w/o GR, where the variant discards DAR and Diff-HDG, relying only on the facts within the HCP for regularization; (3) w/o AR, where the historical and current entity distributions are merged through direct addition instead of DAR as specified in Eq. 15; and (4) w/o Guider, where the operation of enhancing common features across different distributions in Diff-HDG is discarded; (5) w/o L_r, where the loss $\mathcal{L}_{t,r}$ in Section 4.4 is removed during training. The analysis of w/o L_r in Appendix B.6.

Effect Analysis of Historical Context Prompt. In the w/o HP variant, the model's performance noticeably declined, demonstrating that HCP effectively ensures the semantic integrity of the historical information. It prevents catastrophic forgetting during CL and enhances predictions for the current task. In contrast, ER merely replays partial historical information.

Effect Analysis of Diffusion-enhanced Historical Distribution Generation. In the w/o Guider variant, different datasets show varying degrees of performance drop. This demonstrates that incorporating the guider aids in capturing common features between historical and current distributions, thereby mitigating performance losses caused by distribution conflicts. The smaller drop observed on the GDELT dataset likely results from its shorter temporal gaps and less pronounced distribution shifts compared to other datasets.

Effect Analysis of Deep Adaptive Replay. Removing the adaptive parameter α in the w/o AR variant causes varying levels of performance decrease across datasets, demonstrating the effectiveness of our adaptive fusion method in balancing historical and current distribution representations. The limited decrease observed on the GDELT dataset can be attributed to the minimal difference between the current distribution and that of GDELT, which restricts the adaptive parameter α in its capacity to adjust effectively.

Combined Effect Analysis of DAR and Diff-HDG. DAR and Diff-HDG are proposed to resolve the conflict between historical and current

	ICE14			ICE18			ICE05-15			GDELT		
	Current	Average		Current	Average		Current	Average		Current	Average	
	MRR	MRR	Hits@10	MRR	MRR	Hits@10	MRR	MRR	Hits@10	MRR	MRR	Hits@10
w/o HP	53.43	46.74	67.58	31.67	25.89	45.31	53.53	45.71	68.18	15.49	17.18	32.73
w/o GR	48.18	39.15	59.71	30.32	27.62	47.74	52.97	38.71	59.19	16.24	16.67	31.65
w/o AR	49.27	45.55	65.27	32.28	29.79	50.90	58.48	51.93	72.57	22.04	26.98	49.76
w/o Guider	55.23	49.32	70.29	35.16	32.25	54.67	58.70	52.53	72.85	22.18	27.99	50.93
w/o $\mathcal{L}_r$	52.17	44.43	64.13	36.20	30.62	52.98	58.64	51.98	72.01	22.84	25.95	48.75
Ours	**58.59**	**50.12**	**70.48**	**36.53**	**33.00**	**55.63**	**66.01**	**54.33**	**75.13**	**23.25**	**28.30**	**51.39**

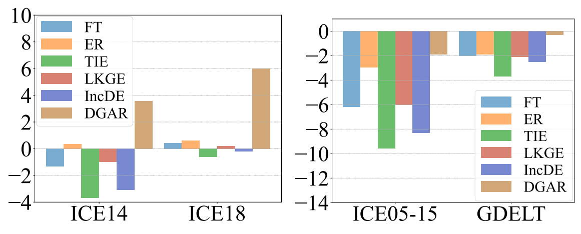
Table 3: Ablation experimental results on all datasets.

distribution. The w/o GR variants show a clear performance drop, likely due to memory contention caused by distribution conflicts arising from the simple replay of historical data. This suggests that Dif-HDG and DAR alleviate distribution conflicts, thereby preserving historical knowledge more efficiently.

Effect Analysis of Different Base Models. Although we choose the most typical model, RE-GCN, as our base model, our method can still be extended to other GNN-based TKGR models. To verify the scalability of our method, we extend DGAR to TiRGN (Li et al., 2022b) and conduct experiments on four benchmark datasets. The experimental results of MRR in Figure 3 indicate that DGAR consistently outperforms direct fine-tuning on both current tasks and historical tasks, demonstrating that DGAR has robust scalability and effectiveness for GNN-based TKGR models.

5.4 Effect of Memorizing in CL

To further validate DGAR's ability to retain historical knowledge during forward learning, we evaluate the mean difference between $p_{n,i}$ and $p_{i,i}$ ($1 < i \leq n$) in Figure 4. Here, $p_{i,j}$ represents the MRR score of the j-th task after training the model on the i-th task. A higher mean value indicates better retention of prior knowledge during CL. When the value exceeds zero, it indicates reverse transfer of newly learned knowledge, whereas a value below zero reflects the loss of prior knowledge during CL (Lin et al., 2022). Experiments show that DGAR outperforms the best baseline, confirming its effectiveness in mitigating catastrophic forgetting. Since the data in TKGs are highly correlated, we find that when the number of tasks is small, a reasonable strategy can help prevent catastrophic forgetting and facilitate the reverse transfer of new knowledge. This is evident in the performance of DGAR and ER on ICE14.



(a) ICE14 and ICE18 (b) ICE05-15 and GDELT

Figure 4: Effect of memorizing old knowledge in CL.

5.5 Case Study

We conduct a case study on ICE14 and ICE18 datasets to assess whether DGAR can handle conflicts in entity distributions and retain old knowledge effectively in Figure 5. At a randomly chosen time i, we extract all entities from the facts, save their feature distributions at time i (Stage 1), time j (Stage 2), time m (Stage 3) and at a later time n ($n > m > j > i$) (Stage 4), and analyzed them using U-MAP.

Figures 5(a) and 5(b) compare the entity distribution representations of the FT model and DGAR on ICE14 across four stages. The entity distribution learned by the FT model at the same time is more clustered, while the entity distributions at different times are more distinct, showing a clearer difference. In contrast, DGAR learns a more general and consistent distribution, allowing it to share the feature space between tasks more effectively, thereby enhancing knowledge retention and reducing forgetting. A similar pattern is observed in Figures 5(c) and 5(d) on ICE18, further supporting these findings.

6 Conclusion

This paper introduces a deep generative adaptive replay method to mitigate catastrophic forgetting in TKGR models during CL. A historical context prompt integrating contextual information is designed to generate historical distribution representations of entities via a pre-trained DM. The generation process is guided by current model parameters

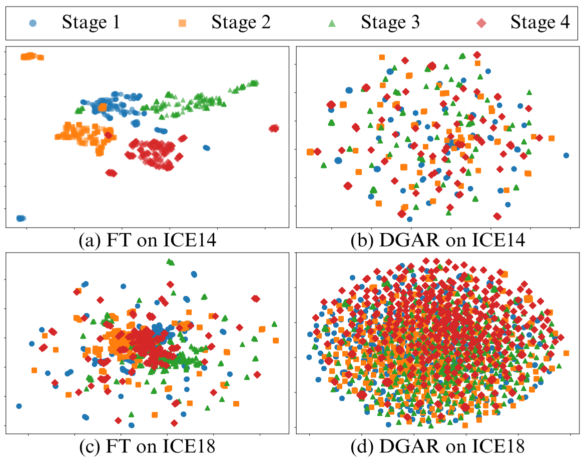


Figure 5: Visualization case study of entity distribution.

to reinforce common features, minimizing conflicts between historical and current entity distributions. In addition, a deep adaptive replay strategy derives entity distribution representations with historical knowledge. These combined techniques enable the proposed method to achieve outstanding performance across various datasets.

7 Limitations

In this section, we examine the limitations of our approach. DGAR is designed to retain previously acquired knowledge through CL, facilitating TKGR. Although DGAR is more time-efficient than retraining and surpasses other models in mitigating catastrophic forgetting, it still faces several challenges.

Firstly, the model addresses newly emerging entities and relations using Xavier initialization without further analysis or dedicated modeling. Such a simplistic approach may constrain the model's ability to learn new knowledge effectively, particularly when complex interrelations exist between new and previously learned knowledge. This highlights the need for more sophisticated strategies to handle new entities and relations in CL scenarios.

Secondly, while DGAR demonstrates strong performance in reducing catastrophic forgetting, it introduces additional learnable parameters. These parameters enhance adaptability to new knowledge but also pose a potential risk of forgetting previously learned information. This risk arises since the increased number of parameters may lead the model to prioritize new knowledge, thereby compromising the retention of older knowledge. Furthermore, the inclusion of additional parameters inherently increases model complexity, making the training and reasoning process more cumbersome. Such complexity necessitates careful consideration during design to strike a balance between knowledge retention and model complexity.

8 Ethics Statement

Firstly, this study fully complies with the ethical guidelines in the ACL Code of Ethics. Secondly, all datasets involved in this study are from previous studies. The datasets we used do not contain individual privacy data. Finally, DGAR focuses on the research and experiments of TKGR tasks. Like other TKGR methods, the results of our method reasoning may be toxic or erroneous, so manual inspection of the results may be required in the applications.

References

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

Yuxiang Cai, Qiao Liu, Yanglei Gan, Changlin Li, Xueyi Liu, Run Lin, Da Luo, and JiayeYang JiayeYang. 2024. Predicting the unpredictable: Uncertainty-aware reasoning over temporal knowledge graphs via diffusion process. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5766–5778.

Kai Chen, Ye Wang, Yitong Li, Aiping Li, Han Yu, and Xin Song. 2024a. A unified temporal knowledge graph reasoning model towards interpolation and extrapolation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 117–132, Bangkok, Thailand. Association for Computational Linguistics.

Wei Chen, Huaiyu Wan, Yuting Wu, Shuyuan Zhao, Jiayaqi Cheng, Yuxin Li, and Youfang Lin. 2024b. Local-global history-aware contrastive learning for temporal knowledge graph reasoning. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 733–746.

Wei Chen, Yuting Wu, Shuhan Wu, Zhiyu Zhang, Mengqi Liao, Youfang Lin, and Huaiyu Wan. 2024c. Cogntke: A cognitive temporal knowledge extrapolation framework. *ArXiv*, abs/2412.16557.

Yubo Chen, Shaoru Guo, Kang Liu, and Jun Zhao. 2023. (large language models and knowledge graphs). In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 2: Frontier Forum)*, pages 67–76.

Yuanning Cui, Yuxin Wang, Zequn Sun, Wenqiang Liu, Yiqiao Jiang, Kexin Han, and Wei Hu. 2023. Lifelong embedding learning and transfer for growing knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4217–4224.

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2001–2011.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3988–3995.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*.

Saiping Guan, Xueqi Cheng, Long Bai, Fujun Zhang, Zixuan Li, Yutao Zeng, Xiaolong Jin, and Jiafeng Guo. 2022. What is event knowledge graph: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):7569–7589.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *arXiv preprint arXiv:2405.14831*.

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

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

Rikui Huang, Wei Wei, Xiaoye Qu, Shengzhe Zhang, Dangyang Chen, and Yu Cheng. 2024. Confidence is not timeless: Modeling temporal validity for rule-based temporal knowledge graph forecasting. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10783–10794, Bangkok, Thailand. Association for Computational Linguistics.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6669–6683, Online. Association for Computational Linguistics.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion proceedings of the the web conference 2018*, pages 1771–1776.

Eunhae Lee. 2024. The impact of model size on catastrophic forgetting in online continual learning. *Preprint*, arXiv:2407.00176.

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

Yujia Li, Shiliang Sun, and Jing Zhao. 2022b. Tirgn: Time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. In *IJCAI*, pages 2152–2158.

Yujia Li, Shiliang Sun, and Jing Zhao. 2022c. Tirgn: Time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 2152–2158.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutional representation learning.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. 2022. Beyond not-forgetting: Continual learning with backward knowledge transfer. *Advances in Neural Information Processing Systems*, 35:16165–16177.

Jiajun Liu, Wenjun Ke, Peng Wang, Ziyu Shang, Jinhua Gao, Guozheng Li, Ke Ji, and Yanhe Liu. 2024a. Towards continual knowledge graph embedding via incremental distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8759–8768.

Jiajun Liu, Wenjun Ke, Peng Wang, Jiahao Wang, Jinhua Gao, Ziyu Shang, Guozheng Li, Zijie Xu, Ke Ji, and Yining Li. 2024b. Fast and continual knowledge graph embedding via incremental lora. *arXiv preprint arXiv:2407.05705*.

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

Mehrnoosh Mirtaheri, Mohammad Rostami, and Aram Galstyan. 2023. History repeats: Overcoming catastrophic forgetting for event-centric temporal knowledge graph completion. *arXiv preprint arXiv:2305.18675*.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. *Advances in neural information processing systems*, 32.

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

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

Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. 2023. Sketch-guided text-to-image diffusion models. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11.

Jiapu Wang, Kai Sun, Linhao Luo, Wei Wei, Yongli Hu, Alan Wee-Chung Liew, Shirui Pan, and Baocai Yin. 2024. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. *arXiv preprint arXiv:2405.14170*.

Jiapeng Wu, Yishi Xu, Yingxue Zhang, Chen Ma, Mark Coates, and Jackie Chi Kit Cheung. 2021. Tie: A framework for embedding-based incremental temporal knowledge graph completion. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 428–437.

Shuhan Wu, Huaiyu Wan, Wei Chen, Yuting Wu, Junfeng Shen, and Youfang Lin. 2023. Towards enhancing relational rules for knowledge graph link prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10082–10097, Singapore. Association for Computational Linguistics.

Yujie Xing, Xiao Wang, Yibo Li, Hai Huang, and Chuan Shi. 2024. Less is more: on the over-globalizing problem in graph transformers. *arXiv preprint arXiv:2405.01102*.

Yi Xu, Junjie Ou, Hui Xu, and Luoyi Fu. 2023a. Temporal knowledge graph reasoning with historical contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4765–4773.

Yi Xu, Junjie Ou, Hui Xu, and Luoyi Fu. 2023b. Temporal knowledge graph reasoning with historical contrastive learning. In *AAAI*.

Zhengyuan Yang, Jianfeng Wang, Zhe Gan, Linjie Li, Kevin Lin, Chenfei Wu, Nan Duan, Zicheng Liu, Ce Liu, Michael Zeng, et al. 2023. Reco: Region-controlled text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14246–14255.

Jiasheng Zhang, Jie Shao, and Bin Cui. 2023a. Streame: Learning to update representations for temporal knowledge graphs in streaming scenarios. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 622–631.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023b. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847.

Shuyuan Zhao, Wei Chen, Boyan Shi, Liyong Zhou, Shuohao Lin, and Huaiyu Wan. 2025. Spatial-temporal knowledge distillation for takeaway recommendation. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 13365–13373. AAAI Press.

A Details about DGAR

A.1 Details about Deep Adaptive Replay

While initially exploring more complex mechanisms for this integration, it is observed that the additional parameters introduced hinders the model's convergence due to increased fitting complexity. As a result, a direct injection approach is adopted to integrate the historical distributions into the current representations, as detailed below:

$$H_{\text{final}} = \begin{cases} H_e^{\text{current}}, & e \notin V_{\text{replay}} \\ H_e^{\text{replay}} + H_e^{\text{current}}, & e \in V_{\text{replay}} \end{cases} . \quad (15)$$

However, it is observed that such a simple and direct fusion approach results in performance degradation. To address this issue, a straightforward parameter is introduced to balance the historical distribution representation H_e^{replay} and the current distribution representation H_e^{current}, thereby generating the final entity representation H_{final}. This parameter effectively adjusts the weighting of the two distributions, mitigating the performance loss caused by direct fusion while preserving the expressive power of historical knowledge and the dynamic characteristics of the current distribution:

$$H_{\text{final}} = \alpha H_e^{\text{replay}} + (1-\alpha) H_e^{\text{current}}, e \in V_{\text{replay}}, \quad (16)$$

where $\alpha \in [0, 1]$. H_{final} denotes the final entity representation, combining the most recent and historical information of the entity.

To prevent the loss of entity evolution patterns over time caused by direct injection, we integrate distribution representation from the KG snapshot sequence reasoning model's evolution unit to achieve a deeper incorporation of historical distribution representations without introducing additional parameters. After applying the relation-aware GCN in the l-th evolution unit, we obtain the current distribution representation of entity $H_e^{\text{current},l}$. The historical distribution representation of entities is then injected into the current distribution representation of entities as follows:

$$H_e^{\text{current},l} = \alpha H_e^{\text{replay}} + (1-\alpha) H_e^{\text{current},l}, e \in V_{\text{replay}}. \quad (17)$$

After passing through multiple evolution units, the final entity representation H_{final}, which incorporates historical distributions, is obtained.

A.2 Pre-train for DM

In this section, we will discuss how we obtain the pre-trained DM. Considering that using a large amount of knowledge to train diffusion will not only increase the risk of data leakage, but also fail to adapt to new data arriving over time, we adopt CL to train DM. For example, at the t-th moment, we can obtain the DM ϕ_{t-1} pre-trained at the previous moment, and ϕ_{t-1} is used as a pre-trained DM to assist in generating the entity history distribution in Diff-HDG. After completing all the operations in Section 4, ϕ_t is initialized with ϕ_{t-1}, and the model parameters are updated on D^t_{train}. Eq. 16 is employed to preserve the historical knowledge of the entity for DM. Upon completion of the training, a new pre-trained DM, ϕ_t, is obtained and will be used in the subsequent learning process.

B Further Analysis

B.1 Datasets Details

	ICE14	ICE18	ICE05-15	GDELT
Entities	6869	23033	10094	7,691
Relations	230	256	251	240
Tasks	365	304	4017	2,751
Task granularity	24 hours	24 hours	24 hours	15mins
Total number of train	74,845	373,018	368,868	1,734,399
Total number of valid	8,514	45,995	46,302	238,765
Total number of test	7,371	49,545	46,159	305,241

Table 4: Details of the TKG datasets.

We follow the common division ratio of TKGR tasks: The facts in each task are partitioned into train, valid, and test sets in a ratio of 8:1:1 (Li et al., 2021; Xu et al., 2023a). The statistical details of the datasets are shown in Table 4.

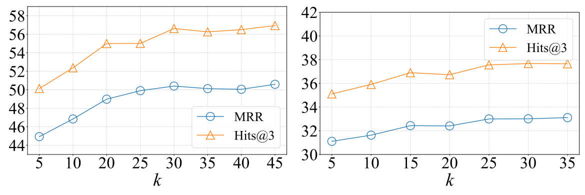
[2]AI such as GPT only assists us in translation and grammar checking.

B.2 Baselines Details

FT (fine-tuning) is a naive baseline where the model is fine-tuned using newly added facts without any mechanism to alleviate catastrophic forgetting. FT is set up following the previous works such as TIE (Wu et al., 2021), LKGE (Wu et al., 2021), and IncDE (Wu et al., 2021). FT enables the base model (e.g., RE-GCN and TiRGN) to perform CL without applying any additional strategies. Specifically, the base model inherits the parameters from the previous time step $i-1$ and continual training on the training data D^t_{train} at time step i. ER (Rolnick et al., 2019) mitigates forgetting by replaying a subset of previously stored events alongside newly added facts during training. TIE (Wu et al., 2021) incorporates temporal regularization, experience replay with positive facts, and the use of deleted facts as negative examples to effectively address both catastrophic forgetting and intransigence. LKGE (Cui et al., 2023) preserves historical knowledge by leveraging historical weights and embeddings through the L2 paradigm. We incorporate the reconstruction loss and embedding regularization from LKGE into our objective function. When initializing new entities, their embedding transfer strategies are adopted. Based on the method proposed by IncDE (Liu et al., 2024a), we leverage the hierarchical ordering measure of IncDE and incorporate the distillation loss proposed by IncDE into our objective function.

B.3 Implementation Details

For all datasets, the embedding size d is set to 200, the learning rate lr is set to 0.001, and the batch size is determined by the number of facts at each time step. The number of layers of the Transformer encoder for all datasets is set to 2. The temperature coefficient τ for all datasets is set to 0.5. The parameters of DAGR are optimized by using Adam during the training process. The optimal coefficient γ in the Diff-HDG is set to 1. The optimal number of layers L in the DAR is set to 3. The optimal loss coefficient μ in model training is set to 1. The number of samples of the best HCP k for ICE14, ICE18, ICE05-15, and GDELT is set to 35, 25, 40, and 32, respectively. We conduct hyperparameter search experiments on the primary parameters of DGAR using control variables. The number of parameters in ICE14, ICE18, ICE05-15, and GDELT is 17.55 MB, 33.71 MB, 21.44 MB, and 21.38 MB, respectively. All experiments are



(a) ICE14 (b) ICE18

Figure 6: Sensitivity Analysis.

conducted on NVIDIA A40.

B.4 Sensitivity Analysis

In this section, we conduct experiments on ICE14 and ICE18 to further analyze the impact of hyperparameter k in DGAR. The hyperparameter k means the replay data consists of HCPs at k different times.

To explore how the number of k affects the model's ability to retain historical knowledge, we set different values for k. The results are shown in Figure 6, including MRR and Hits@3 results on the historical tasks. A larger k means the replay data consists of HCPs at more different times. The results reveal that on the ICE14 dataset, DGAR demonstrates an initial improvement followed by a plateau as k increases. These results indicate that selecting HCPs at more different times for historical distribution representations replay does not significantly enhance the final performance but instead increases computational costs. Notably, even with only 5 recall time slices, DGAR outperforms all baseline models. This demonstrates that our approach can effectively help the model retain historical knowledge, even under limited memory constraints.

B.5 Inference Efficiency

	ICE14			ICE18		
	k	Time(s) / Task	MRR	k	Time(s) / Task	MRR
Retrain	—	553.48	49.80	—	530.48	31.35
DGAR	5	4.00	44.93	5	13.12	31.11
	25	5.21	49.90	15	16.32	32.43
	35	5.42	50.12	25	18.83	33.00
	45	5.68	50.05	35	19.41	33.11
FT	—	2.53	37.46	—	5.48	25.35
ER	—	2.86	42.14	—	5.16	27.20
TIE	—	4.38	41.07	—	10.71	29.31
LKGE	—	2.70	37.51	—	5.03	25.56
IncDE	—	4.07	36.57	—	9.71	25.52

Table 5: Inference efficiency analysis.

We report the average time cost of each task and the MRR on historical tasks for DGAR at different k values in Table 5. The hyperparameter k means the replay data consists of HCPs at k different times. We further report the average time consumption of each task and the MRR on historical tasks under different baselines and retaining settings in Table 5. Unlike DGAR and the baselines, retraining involves reprocessing the entire dataset whenever new data arrives. By comparing the results, our method outperforms retraining and requires less time. This shows the powerful ability of our model in dealing with catastrophic forgetting. Because our model uses more complex operations than the baseline in order to retain more historical knowledge, it is more time-consuming. Future research will focus on enhancing reasoning efficiency while preserving the accuracy of historical knowledge retention.

B.6 Effect Analysis of $\mathcal{L}_r$

In the w/o $\mathcal{L}_r$ variant, the performance of the historical tasks drops significantly. This shows that the $\mathcal{L}_{r,t}$ loss in Section 4.4 effectively alleviates the loss of historical information caused by current data optimization and Diff-HDG.

B.7 Random Selection of HCPs

In order to verify whether the generalization of replay data is enhanced, we added an additional experiment to prove. Compared to replaying historical data from the specified k time slices, randomly selecting across the entire history provides more global and generalizable information. Therefore, we specifically added an experiment where the historical context prompts from the k nearest time slices was selected as the replay data. The experimental results are in the Table 6 :

ICE14			ICE18		
k	Average (MRR)		k	Average (MRR)	
	nearest	random		nearest	random
25	46.64	49.90	15	31.23	32.43
35	48.11	50.12	25	31.89	33.00
45	47.86	50.05	35	32.03	33.11

Table 6: Effect of Random Selection

The nearest refers to replaying with historical context prompts sampled from the nearest k time slices. As shown in the Table 6, regardless of different k values, the random outperforms the nearest on the historical tasks. The experimental results are consistent with our expectations. Mainly because the random selection of historical context prompts provides the model with more generalized data, thus improving the model's performance on the test set.

B.8 Compare Base on LogCL

To further verify whether DGAR can enhance the performance of recent GNN-based models under the CL setting, we conducted the following experiment.

We selected LogCL (Chen et al., 2024b) , a representative GNN-based TKGR model from recent works, as the base model. Below, we report its performance under the CL setting (FT) and its performance when combined with DGAR in the same setting on two datasets.

	ICE14			ICE18		
Algo.	Current	Average		Current	Average	
	MRR	MRR	Hits@10	MRR	MRR	Hits@10
FT	31.63	28.93	48.61	38.46	30.47	55.52
DGAR	**37.12**	**33.08**	**53.83**	**43.20**	**32.88**	**58.88**

Table 7: Performance base on LogCL

The above experiments show that DGAR significantly enhances LogCL's reasoning performance under the CL setting. Interestingly, LogCL performs worse on the ICE14 dataset than on ICE18, which contrasts with its performance in the full-training setting. This discrepancy occurs because ICE14 has a simpler data distribution compared to ICE18, while LogCL's complex model structure makes it prone to overfitting on ICE14. Under the CL setting, highly complex models struggle to maintain stable learned features as new data is introduced (Lee, 2024). DGAR mitigates this issue by replaying historical information, helping LogCL retain its learned features more effectively. Consequently, TKGR methods that excel in full-training scenarios may not necessarily achieve better reasoning performance under CL setting.