

HTA: Temporal Reasoning Transformer with Temporal Context Encoding for Temporal Knowledge Graph Forecasting

Yingsong Ning, Fu Zhang*, Jingwei Cheng, Jiashun Peng, Xiaoke Wang

School of Computer Science and Engineering,
Northeastern University, Shenyang 110819, China
{2401887, 2472039, 2401911}@stu.neu.edu.cn
{zhangfu, chengjingwei}@neu.edu.cn

Abstract

Temporal knowledge graph (TKG) forecasting aims to predict future facts based on historical temporal information and still faces key challenges such as event sparsity and complex long-term dependencies. While graph-based methods effectively capture relational structure, they often fail to model nuanced long-range temporal patterns. Alternatively, standard Transformers, while potent for sequence modeling, are fundamentally time-agnostic; their generic positional encodings overlook critical TKG-specific temporal dynamics. As a result, many approaches struggle to capture fine-grained temporal patterns, limiting their effectiveness in forecasting sparse or complex events. To address these limitations, we propose **HTA** (Hybrid Transformer Architecture). HTA first constructs a query-specific Attention-Ready Temporal Sequence (ARTS) by aggregating historical context along a query-centered temporal reasoning path. Each historical timestamp is augmented with a relation-aware Temporal Context Encoding that jointly models its time difference to the query and the associated relation. At its core, HTA employs a stack of hybrid Attention-Mixer Transformer Blocks that combine bidirectional self-attention for capturing long-range dependencies with MLP-based layers for efficient feature refinement. Extensive experiments on four diverse benchmarks show that HTA achieves state-of-the-art performance. Notably, on the sparse and challenging GDELT dataset, HTA achieves an absolute MRR improvement of over 18 points, demonstrating its superior ability to model complex and noisy temporal dynamics.

1 Introduction

Forecasting future events based on historical data is a fundamental challenge with significant real-world applications, from political event monitoring to social dynamics analysis (Trivedi et al., 2017; Jin

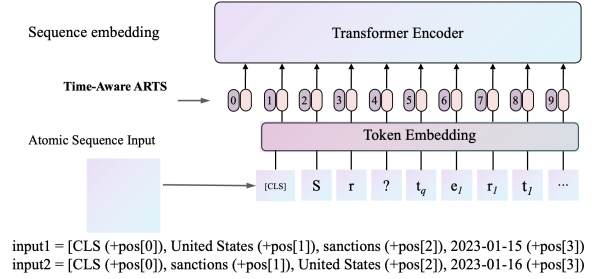


Figure 1: Illustration of HTA’s input embedding pipeline. While standard positional encodings (+pos[i]) distinguish semantic order (input1 vs. input2), they are insufficient for temporal modeling. Our Time-Aware ARTS explicitly encodes fine-grained temporal semantics between events and the query to capture timestamp relationships.

et al., 2020; Li et al., 2021). Temporal Knowledge Graphs (TKGs) provide a structured representation for such dynamic systems, capturing evolving entities and relations as timestamped facts (s, r, o, t) . Within TKG analysis, Temporal Knowledge Graph Forecasting—the task of predicting future facts (i.e., extrapolation) beyond the observed timeline—is particularly important because it requires models to reason over unseen timestamps while preserving both graph structure and temporal evolution. While interpolation (filling gaps within known history (Cai et al., 2023)) is also important, our focus is on the forecasting challenge.

Effective TKG forecasting hinges on modeling long-term temporal dependencies and handling sparse or unseen events. Real-world TKGs exhibit dynamic temporal patterns, where the relevance of historical facts decays or reappears based on complex time intervals and sequences (Li et al., 2021; Sun et al., 2022). Capturing these fine-grained patterns is essential for accurate forecasting, yet poses significant challenges because rare events provide limited observations and important dependencies may span long histories (Zhu et al., 2021; Cai et al., 2023).

* Corresponding author.

Existing TKG forecasting methods approach this challenge from complementary directions. Static KGE models such as TransE (Bordes et al., 2013) and RotatE (Sun et al., 2019) ignore temporal evolution; temporal KGE extensions such as TTransE (Leblay and Chekol, 2018) and TeRo (Xu et al., 2020) encode timestamps but remain limited in modeling long-range event dependencies; and RNN/GNN-based models such as RE-GCN (Li et al., 2021) and RE-NET (Jin et al., 2020) propagate information over graph snapshots but can miss fine-grained intervals between events (Sun et al., 2022). More recent Transformer and causal methods push sequence and structural modeling further through graph Transformers, temporal logical graphs, and causal enhanced representations (Sun et al., 2022; Lin et al., 2023; Sun et al., 2025). Among our LLM-assisted baselines, TiRGN+CoH, AnRe, and G2S introduce chain-of-history prompting, analogical replay, or general-to-specific LLM adaptation (Xia et al., 2024; Tang et al., 2025; Bai et al., 2025). These advances highlight the value of sequence and semantic reasoning, yet it remains challenging to build compact query-specific inputs that jointly preserve graph structure, timestamp semantics, and long-range history.

While Transformers (Vaswani et al., 2017) excel at modeling long-range dependencies, directly applying them to TKG forecasting remains difficult because their architecture is fundamentally time-agnostic. As illustrated in **Figure 1**, standard positional encodings capture sequence order but ignore crucial temporal information, such as absolute timestamps and the duration of intervals between events. This blindness to fine-grained temporal dynamics can curtail predictive accuracy.

To overcome this limitation, we propose **HTA** (Hybrid Transformer Architecture), a novel architecture specifically designed to understand temporal nuances. For a given query like $(s, r, ?, t_q)$, HTA first constructs a query-specific Attention-Ready Temporal Sequence (ARTS) from historical context, which explicitly embeds the fine-grained timing information that standard Transformers miss. Subsequently, a bidirectional Transformer encoder models the complex interdependencies within these temporally-aware ARTS, and an efficient MLP-Mixer architecture (Tolstikhin et al., 2021) refines these contextualized representations to facilitate robust future event prediction.

Our main contributions are summarized as follows:

- We propose HTA, a TKG forecasting architecture that converts query-specific graph history into temporally ordered ARTS inputs and processes them with a hybrid Attention-Mixer encoder.
- We introduce relation-aware Temporal Context Encoding (TC), which jointly models each historical event’s time difference to the query and its relation type through an MLP, capturing temporal semantics beyond generic positional embeddings.
- We validate HTA with extensive experiments and ablations on four TKG forecasting benchmarks. Notably, it achieves absolute MRR and Hits@1 improvements of 18.14 and 18.34 points, respectively, on the challenging, high-frequency (i.e., events occurring in rapid succession) GDELT dataset over strong prior methods.

2 Related Work

TKG forecasting aims to predict future facts (s, r, o, t) where $t > t_T$, given historical data $\mathcal{G}_{t_0:t_T}$. This extrapolation task (Jin et al., 2020; Li et al., 2021) requires models to understand temporal evolution and generalize to future timestamps.

Embedding-based and rule-based methods incorporate time into KGE scoring, tensor factorization, or symbolic constraints (Balazevic et al., 2019; Leblay and Chekol, 2018; García-Durán et al., 2018; Xu et al., 2020; Goel et al., 2020; Lacroix et al., 2020; Han et al., 2020; Park et al., 2022; Liu et al., 2022; Huang et al., 2024). They are efficient and often interpretable, but local scoring or rule structures make it difficult to represent query-specific long-range histories.

Graph, sequential, and Transformer-based methods model evolving structure with snapshot encoders, recurrent propagation, copy/history mechanisms, temporal logic, causal representations, candidate-specific context, or self-attention (Jin et al., 2020; Li et al., 2021; Schlichtkrull et al., 2018; Zhu et al., 2021; Li et al., 2022a,b; Lin et al., 2023; Sun et al., 2025; Zhang et al., 2024; Sun et al., 2022; Gao et al., 2023). These models capture richer temporal signals than static scoring methods, yet they often rely on fixed snapshot-level context, task-specific retrieval, or generic positional encodings rather than explicitly preserving query-specific multi-hop temporal evidence as ordered sequence inputs.

Recent extrapolation methods further explore inductive temporal-path propagation and negative-aware diffusion, as exemplified by TiPNN and NADEx (Dong et al., 2024; Gan et al., 2026).

For the LLM baselines, TiRGN+CoH, AnRe, and G2S add semantic reasoning through chain-of-history prompting, analogical replay, or general-to-specific adaptation (Xia et al., 2024; Tang et al., 2025; Bai et al., 2025). They show that language-level reasoning can complement structural forecasting, but still depend on prompting, retrieval, or adaptation pipelines. In contrast, **HTA** directly constructs compact Attention-Ready Temporal Sequences (ARTS), contextualizes them with a Transformer encoder, and refines the representations with an efficient MLP-Mixer for extrapolated forecasting.

3 Problem Formulation

A Temporal Knowledge Graph (TKG) is a sequence of timestamped facts (s, r, o, t) involving entities $\mathcal{V}$, relations $\mathcal{R}$, and discrete timestamps $\mathcal{T}$. These are often viewed as graph snapshots $\mathcal{G}_t = (\mathcal{V}, \mathcal{R}, \mathcal{E}_t)$, where $\mathcal{E}_t$ contains facts at time t . To handle reciprocity, inverse relations (o, r^{-1}, s, t) are commonly included for each fact.

Temporal Knowledge Graph Forecasting (or extrapolation) aims to predict future facts. Given a TKG observed up to time t_T , $\mathcal{G}_{1:t_T} = \{\mathcal{G}_1, \dots, \mathcal{G}_{t_T}\}$, the task is to predict missing elements in queries (e.g., object o in $(s, r, ?, t_q)$ or subject s) for a future time $t_q > t_T$. This requires learning conditional probabilities like $p(o|s, r, t_q, \mathcal{G}_{1:t_T})$ (or $p(s|r, o, t_q, \mathcal{G}_{1:t_T})$). Unlike interpolation, forecasting necessitates reasoning about unseen timestamps and modeling complex evolutionary patterns from the observed history.

4 Methodology

We propose HTA, a multi-stage architecture designed to model long-term dependencies and fine-grained temporal dynamics (Figure 3). The model first builds an Attention-Ready Temporal Sequence (ARTS) by aggregating historical context and enriching it with temporal context encoding. Then, a stack of Attention-Mixer Transformer Blocks processes this sequence to learn deep representations. HTA is trained via a dual-objective function combining link prediction and an auxiliary masked timestamp prediction task.

4.1 Historical Context Aggregation (HCA)

The first stage constructs a query-specific historical context to capture relevant past events. HCA first follows a query-centered temporal path through historical facts, then enriches this core path with sampled neighborhoods of the path entities. The resulting context therefore contains both the path-level temporal trajectory and local structural evidence around each entity in the path (Figure 2); the full pseudocode is provided in Appendix A. To prevent temporal leakage, for a query at time t_q , both path construction and neighborhood enrichment are restricted to historical facts with timestamps $t < t_q$; facts at t_q or later are never used to construct the query context.

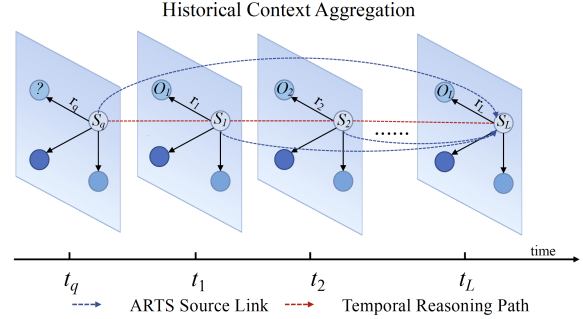


Figure 2: An illustration of the Historical Context Aggregation (HCA) process. The figure shows how a query entity expands into a temporal context path and how local neighbourhoods are aggregated along the path entities to form the final context set $\mathcal{N}_{ctx}$. All selected historical facts satisfy $t_1 \leq \dots \leq t_L < t_q$. The query anchor (s_q, t_q) is drawn at the left only to show the starting entity of HCA; its placement is not a temporal position. The remaining horizontal order shows the chronology after sampled facts are sorted for ARTS construction, not the order in which the random walk samples candidates.

4.2 Attention-Ready Temporal Sequence (ARTS) Construction

4.2.1 Temporal Context Path Construction

We construct a Temporal Context Path $\mathcal{P} = (F_1, \dots, F_\ell)$ to model the history of the query subject s , where $\ell \leq L_{\text{path}}$ if the candidate set becomes empty before reaching the maximum path budget. Starting from s , each step samples one historical fact from the current entity’s historical neighborhood and moves to the sampled counterpart entity. The sampled path facts form the core temporal trajectory. Before entering the encoder, the selected facts are sorted chronologically with respect to their timestamps, so the model observes

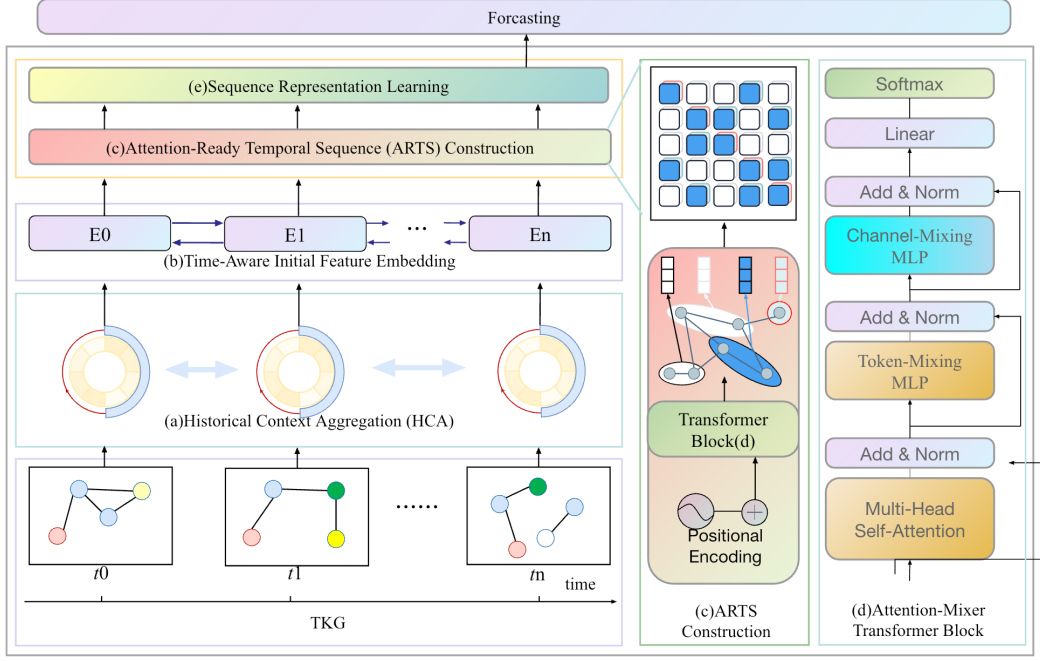


Figure 3: Overview of the HTA model architecture.

a time-ordered ARTS even though the context path is sampled from historical candidates.

4.2.2 Neighborhood Definition (Multigraph View)

For neighborhood sampling, we treat each time-stamped fact as a distinct edge instance in a temporal multigraph. Thus, repeated interactions between the same entity pair are not collapsed, allowing HCA to preserve both frequency and recency signals.

4.2.3 Context Set Definition

The final context set $\mathcal{N}_{ctx}(s)$ has two parts. The path facts $\{F_1, \dots, F_\ell\}$ provide the core temporal trajectory. In addition, for each entity on this path, HCA samples up to M historical neighborhood facts and adds them to $\mathcal{N}_{ctx}(s)$. This distinction resolves the notation in Appendix Algorithm 1: $\mathcal{P}$ denotes the sampled path facts, while $\mathcal{N}_{ctx}(s)$ denotes the neighborhood-enriched context used to form ARTS.

4.2.4 Sequence Formulation

We flatten the query and the neighborhood-enriched historical context into an atomic sequence. Let L_{ctx} be the maximum number of historical facts retained after chronological sorting and truncation. Each query keeps $n_s \leq L_{ctx}$ non-padding context facts $\tilde{\mathcal{N}}_{ctx}(s) = \{\tilde{F}_1, \dots, \tilde{F}_{n_s}\}$ and pads the remaining slots. Each fact is oriented from

its anchor/path entity a_k to the counterpart entity e_k , using a reciprocal relation when needed; thus $\tilde{F}_k = (a_k, r_k, e_k, t_k)$ and $\phi(\tilde{F}_k) = [e_k, r_k, t_k]$. The anchor a_k is used to orient the sampled fact and is omitted from $\phi(\tilde{F}_k)$ for compactness; the counterpart entity, direction-aware relation, and timestamp are retained, while reciprocal relations preserve traversal direction.

$$\mathbf{A}_s = [\text{CLS}] \oplus \phi(F_0) \oplus \phi(\tilde{F}_1) \oplus \dots \oplus \phi(\tilde{F}_{L_{ctx}}) \quad (1)$$

where $\tilde{F}_k = \tilde{F}_k$ for $k \leq n_s$ and $\tilde{F}_k$ is a padding fact for $k > n_s$.

For an object-forecasting query $F_0 = (s, r, ?, t_q)$, the query token is $\phi(F_0) = [s, r, t_q]$. Expanding the tokens into their components, the sequence becomes:

$$\begin{aligned} \mathbf{A}_s = & [\text{CLS}] \oplus [s, r, t_q] \\ & \oplus [\bar{e}_1, \bar{r}_1, \bar{t}_1] \oplus \dots \oplus [\bar{e}_{L_{ctx}}, \bar{r}_{L_{ctx}}, \bar{t}_{L_{ctx}}] \end{aligned} \quad (2)$$

Subject forecasting is handled by the same object-forecasting form after reciprocal conversion: an original subject query $(?, r, o, t_q)$ becomes $(o, r^{-1}, ?, t_q)$. The context facts are ordered chronologically. We employ a standard bidirectional self-attention mechanism without causal masking, allowing the model to leverage the full historical context collectively for prediction rather than imposing a causal next-token objective over the context.

4.2.5 Time-Aware Initial Feature Embedding

Each atomic element in the sequence $\mathbf{A}_s$ is mapped to a dense feature vector by summing three components. First, lookup embeddings map entities, relations, and timestamps to d -dimensional vectors. Second, a learnable type embedding $\mathbf{P}_{\text{type}}$ distinguishes among the four token types ([CLS], entity, relation, and time). Third, Temporal Context Encoding (TC) injects fine-grained, relation-aware temporal dynamics. For a time token t_k associated with a historical fact F_k , we compute

$$\mathbf{tc}_k = \text{MLP}([\Delta t_k; \mathbf{e}_{r_k}]), \quad \Delta t_k = t_q - t_k, \quad (3)$$

where Δt_k is the time difference relative to the query and $\mathbf{e}_{r_k}$ is the embedding of the associated relation. The resulting relation-aware temporal vector is added to the corresponding time-token representation. Finally, the input matrix $\mathbf{X}_s$ is processed by pre-layer normalization.

4.3 Attention-Mixer Transformer Block

The core of our HTA model comprises a stack of N identical Attention-Mixer Transformer Blocks. We denote the input to the k -th block as $\mathbf{Z}^{(k-1)}$, where the initial input is set to $\mathbf{Z}^{(0)} = \mathbf{X}_s$. Each block processes this representation through three sub-layers to learn progressively richer features.

4.3.1 Multi-Head Self-Attention

The first sub-layer employs a standard multi-head self-attention (MHSA) mechanism. Given the input $\mathbf{Z}^{(k-1)}$, we compute Query (Q), Key (K), and Value (V) matrices using learnable projections. The attention output is calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

This mechanism captures global dependencies between all elements in the sequence.

4.3.2 Token-Mixing MLP

The second sub-layer is a Token-Mixing MLP module, which operates across the sequence dimension and facilitates direct information exchange between different atomic tokens.

4.3.3 Channel-Mixing MLP

The final sub-layer is a Channel-Mixing MLP, which operates on the feature vector of each token independently to refine channel-wise information. The final contextualized representation after N blocks is denoted as $\mathbf{X}_{\text{final}} = \mathbf{Z}^{(N)}$.

4.4 Prediction and Training

Our model is trained end-to-end using a dual-objective loss function to enhance both forecasting accuracy and temporal reasoning.

4.4.1 Link Prediction

For the primary task, we obtain a query context vector $\mathbf{z}_s$ by applying global average pooling to the final sequence representation $\mathbf{X}_{\text{final}}$. We then compute the score vector $\mathbf{s} \in \mathbb{R}^{|\mathcal{V}|}$ over the entity vocabulary by taking the dot product with the entity embedding matrix:

$$\mathbf{s} = \mathbf{z}_s \cdot \text{Emb}_e^T \quad (5)$$

The link prediction loss is the standard Cross-Entropy loss computed between the output logits and the ground-truth target index:

$$\mathcal{L}_{\text{link}} = \text{CrossEntropy}(\mathbf{s}, y_{\text{true}}) \quad (6)$$

where $\mathbf{s} \in \mathbb{R}^{|\mathcal{V}|}$ is the vector of output logits from the model, and y_{true} is the index of the ground-truth target entity.

4.4.2 Masked Timestamp Prediction

We employ an auxiliary task to predict the masked timestamps. For each masked index $j \in \mathcal{M}$, the model uses the representation $\mathbf{X}_{\text{final}}[j, :]$ to predict a probability distribution $\mathbf{p}_j$ over timestamp values. The loss is the cross-entropy between $\mathbf{p}_j$ and the true timestamp t_j :

$$\mathcal{L}_{\text{mlm}} = \frac{1}{|\mathcal{M}|} \sum_{j \in \mathcal{M}} \text{CE}(\mathbf{p}_j, t_j) \quad (7)$$

4.4.3 Total Loss Function

The total training objective combines the link prediction loss and the auxiliary loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{link}} + \lambda_{\text{mlm}} \cdot \mathcal{L}_{\text{mlm}} \quad (8)$$

The weight λ_{mlm} is dynamically adjusted based on the context dropout rate δ (the proportion of masked timestamps):

$$\lambda_{\text{mlm}} = \frac{\delta}{1 + \delta} \quad (9)$$

5 Experiments

5.1 Experimental Setup

5.1.1 Datasets

We use four standard TKG datasets for evaluation: ICEWS14 (Trivedi et al., 2017),

ICEWS18 (Boschee et al., 2015), ICEWS05-15 (García-Durán et al., 2018), and GDELT (Lee-taru and Schrodtt, 2013). The ICEWS series datasets capture daily political events, whereas GDELT consists of high-frequency (15-minute interval) global events. For all datasets, we adhere to established temporal splitting procedures for training, validation, and testing, following prior work (Li et al., 2021; Sun et al., 2022). Detailed statistics for each dataset, including counts of entities, relations, and facts per split, are provided in Table 8 of Appendix F.

5.1.2 Baselines

We compare HTA against a comprehensive set of baselines from four main categories, with results shown in Table 1. The categories are: **1) Static KGE models** that ignore time, such as TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), and R-GCN (Schlichtkrull et al., 2018); **2) Temporal KGE models** that extend static methods, including TTransE (Leblay and Chekol, 2018), TA-DistMult (García-Durán et al., 2018), DE-Simple (Goel et al., 2020), TNTComplEx (Lacroix et al., 2020), xERTE (Han et al., 2020), Evo-KG (Park et al., 2022), TITer (Sun et al., 2021), TLogic (Liu et al., 2022), and TempValid (Huang et al., 2024); **3) GNN- and path-based sequential models** like RE-NET (Jin et al., 2020), RE-GCN (Li et al., 2021), CyGNet (Zhu et al., 2021), CEN (Li et al., 2022a), HiSMatch (Li et al., 2022b), and TiPNN (Dong et al., 2024); and **4) Transformer-, diffusion-, and LLM-based architectures**, such as GHT (Sun et al., 2022), TECHS (Lin et al., 2023), CEGRL-TKGR (Sun et al., 2025), NADEX (Gan et al., 2026), TiRGN+CoH (Xia et al., 2024), AnRe (Tang et al., 2025), and G2S (Bai et al., 2025). Following common practice in TKG forecasting benchmarks, baseline scores are copied from the corresponding papers or their reported benchmark settings.

5.1.3 Evaluation Metrics

For all datasets, HTA uses 1-to-All scoring: each query $(s, r, ?, t_q)$ is scored against every entity in $\mathcal{V}$. Subject prediction is evaluated through reciprocal queries, so both directions reduce to the same object-ranking protocol. Following standard protocols (Li et al., 2021; Sun et al., 2022), evaluation uses the time-wise filtered setting: rank the ground truth o_{gt} against all $e \in \mathcal{V}$ where (s, r, e, t_q) is ab-

sent in the full graph at the same timestamp. We report Mean Reciprocal Rank (MRR) and Hits@k ($k=1, 3, 10$); higher is better.

5.1.4 Implementation Details

We implement HTA in PyTorch and run experiments on four NVIDIA RTX4090 GPUs. Hyperparameters are selected by validation MRR: embedding dimension $d = 320$ for ICEWS and $d = 256$ for GDELT, batch size 256, $N = 6$ Mixer blocks, maximum path budget $L_{\text{path}} = 50$, final ARTS context length $L_{\text{ctx}} = 50$, neighbor budget $M = 50$, hidden/attention dropout 0.1, and learning rate 0.01 with Adamax (Kingma and Ba, 2014). We fix Python, PyTorch, and NumPy seeds to 333, 339, and 333, respectively, and propagate them to data-loader workers. Appendix E reports training time, inference time, and hardware efficiency details.

5.2 Main Results

We perform TKG forecasting experiments on four datasets, and the main results are shown in Table 1.

Overall Comparison: HTA achieves new state-of-the-art results on most metrics and all MRR columns (bolded). The Improve (abs.) row reports HTA’s absolute difference from the strongest prior result in each metric column.

Performance on ICEWS Datasets: On the three ICEWS datasets, which are characterized by more regular and structured daily event patterns, HTA demonstrates robust performance across ranking metrics. On ICEWS14, HTA achieves an MRR of 52.04 and Hits@1 of 45.99, improving over NADEX by **3.01** and **6.54** points, respectively. Its Hits@3 and Hits@10 trail NADEX by 2.96 and 6.80 points, indicating that HTA is strongest at ranking the correct entity near the top while remaining competitive at broader cutoffs.

Performance on GDELT Dataset: The most compelling evidence of HTA’s capabilities is its exceptional performance on GDELT, which is difficult due to high data sparsity, 15-minute granularity, and noisy event patterns. As shown in Table 1, HTA improves over the strongest prior result by 18.14 MRR points and 18.34 Hits@1 points. This gain is stable across runs: five different GDELT seeds produce MRR 42.42 ± 0.31 , Hits@1 34.91 ± 0.28 , Hits@3 45.98 ± 0.35 , and Hits@10 55.25 ± 0.27 .

Table 2 further shows that the improvement is not concentrated in only frequent relations. HTA

Method	ICEWS14				ICEWS18				ICEWS05-15				GDELT			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	22.48	13.36	25.63	41.23	12.24	5.84	12.81	25.10	22.55	13.05	25.61	42.05	-	-	-	-
DistMult	27.67	18.16	31.15	46.96	10.17	4.52	10.33	21.25	28.73	19.33	32.19	47.54	8.61	3.91	8.27	17.04
ComplEx	30.84	21.51	34.48	49.59	21.01	11.87	23.47	39.97	31.69	21.44	35.74	52.04	9.84	5.17	9.58	18.23
R-GCN	28.03	19.42	31.95	44.83	15.05	8.31	16.49	29.00	27.13	18.83	30.41	43.16	12.17	8.64	12.37	20.63
TTransE	13.43	3.11	17.32	34.55	8.31	1.92	8.56	21.89	15.71	5.00	19.72	38.02	5.50	0.49	4.99	15.18
TA-DistMult	26.47	17.09	30.22	45.41	16.75	8.61	18.41	33.59	24.31	14.58	27.92	44.21	10.34	6.25	10.44	21.63
DE-SimplE	32.67	24.43	35.69	49.11	19.30	11.53	21.86	34.80	35.02	25.91	38.99	52.75	-	-	-	-
TNTComplEx	32.12	23.35	36.03	49.13	21.23	13.28	24.02	36.91	27.54	19.52	30.80	42.86	19.53	12.41	20.75	33.42
CygNet	32.73	23.69	36.31	50.67	24.93	15.90	28.28	42.61	34.97	25.67	39.09	52.94	18.05	11.13	19.11	31.50
Evo-KG	26.90	16.69	30.57	47.39	25.46	16.25	29.15	43.21	26.32	15.82	31.96	50.80	18.94	11.31	20.08	34.01
xERTE	40.79	32.70	45.67	57.30	29.31	21.03	33.51	46.48	46.62	37.84	52.31	63.92	19.45	11.92	20.84	34.18
TTTer	36.06	27.51	40.16	52.05	25.34	18.09	28.17	38.95	38.11	26.83	44.43	59.44	15.75	10.94	15.74	25.37
TLogic	41.80	31.93	47.23	60.53	28.41	18.74	32.71	47.97	45.99	34.49	52.89	67.39	-	-	-	-
RE-NET	35.71	26.79	39.57	52.81	26.88	17.70	30.35	45.04	39.11	29.04	44.10	58.90	-	-	-	-
RE-GCN	36.08	27.30	39.89	53.01	27.34	18.44	30.48	44.85	39.89	30.10	44.63	56.98	19.07	12.16	20.20	32.39
CEN	41.93	31.71	46.86	61.36	29.41	19.60	33.91	49.97	47.04	36.58	52.60	67.18	19.96	12.68	20.97	34.42
GHT	37.40	27.77	41.66	56.19	27.40	18.08	30.76	45.76	41.50	30.79	46.85	62.73	20.04	12.68	21.37	34.42
TempValid	45.78	35.50	51.34	65.06	33.50	23.91	37.89	52.33	50.31	39.46	56.71	70.55	21.88	14.37	24.40	37.00
TiRGN + CoH	43.94	33.07	49.64	64.90	32.98	21.83	37.79	54.92	49.71	38.01	56.40	71.25	-	-	-	-
TECHS	43.88	34.59	49.36	61.95	30.85	21.81	35.39	49.82	48.38	38.34	54.69	68.92	-	-	-	-
CEGRL-TKGR	42.74	32.32	48.05	62.68	32.90	22.70	36.91	52.95	48.35	37.63	54.22	68.47	20.11	12.73	21.46	34.51
HiSMatch	46.42	35.91	51.63	66.84	33.99	23.91	37.90	53.94	<u>52.85</u>	<u>42.01</u>	<u>59.05</u>	<u>73.28</u>	-	-	-	-
TiPNN	-	-	-	-	32.17	22.74	36.24	50.72	-	-	-	-	21.17	14.03	22.98	34.76
NADEx	49.03	39.45	57.48	70.55	<u>36.84</u>	<u>27.58</u>	<u>45.12</u>	60.58	-	-	-	-	23.67	16.10	<u>28.35</u>	<u>43.05</u>
AnRe	47.40	36.90	51.10	65.70	35.50	26.00	39.20	56.70	50.90	39.10	58.00	69.60	<u>24.30</u>	<u>16.60</u>	26.60	37.50
G2S	-	38.33	54.12	<u>68.58</u>	-	23.04	35.32	46.62	-	41.39	58.15	73.34	-	-	-	-
HTA	52.04	45.99	<u>54.52</u>	63.75	46.35	39.44	49.43	58.89	59.14	52.27	62.77	71.72	42.44	34.94	46.01	55.29
Improve (abs.)	3.01 ↑	6.54 ↑	-2.96 ↓	-6.80 ↓	9.51 ↑	11.86 ↑	4.31 ↑	-1.69 ↓	6.29 ↑	10.26 ↑	3.72 ↑	-1.62 ↓	18.14 ↑	18.34 ↑	17.66 ↑	12.24 ↑

Table 1: Experimental results of TKG forecasting on three ICEWS series datasets and the GDELT dataset. The best result in each column is boldfaced and the second best result is underlined. The ‘Improve (abs.)’ row shows HTA’s absolute difference from the best prior result, with ↑ indicating a positive improvement. A dash indicates that the original paper does not report the metric.

Relation group	HTA	Best baseline	Δ
High-frequency	45.82	24.15	+21.67
Mid-frequency	41.35	21.42	+19.93
Low-frequency	38.17	17.63	+20.54

Table 2: GDELT per-relation-frequency MRR breakdown.

maintains an approximately 20-point MRR advantage over the strongest baseline in high-, mid-, and low-frequency relation groups. The component ablation in Appendix D shows the same pattern mechanistically: on GDELT, removing TC drops MRR from 42.44 to 26.18, while using minimal ARTS context drops MRR to 25.73. These results support our main interpretation that time-aware event encoding and path-based ARTS construction are especially valuable for sparse high-frequency event streams.

5.3 Ablation Study

To rigorously evaluate the contributions of HTA’s key architectural components, we conducted ablation studies on all four datasets. Table 3 reports the ICEWS14 component ablation, while Appendix D gives full metrics for GDELT, ICEWS18, and ICEWS05-15. We designed variants that disable the Self-Attention sub-layer, the MLP Token-Mixing sub-layer, our Temporal Context (TC) en-

coding, the auxiliary MLM loss, and variants that reduce the retained ARTS context (L_{ctx}) or HCA neighborhood scope (M).

Variant Configuration	MRR	Hits@1	Hits@3	Hits@10
HTA w/o Self-Attention	39.47	29.90	44.10	58.20
HTA w/o MLP Token-Mixing	40.59	31.10	45.30	59.50
HTA w/o Temporal Context (TC)	36.41	26.78	41.32	54.50
HTA w/o Aux-Loss (MLM)	48.39	41.50	51.50	62.00
HTA w/ Min ARTS Context ($L_{ctx} = 1$)	37.42	26.03	43.07	53.18
HTA w/ Reduced HCA Scope (M)	48.52	41.82	51.15	62.06
HTA (Full)	52.04	45.99	54.52	63.75

Table 3: Ablation study on ICEWS14 (values in %).

The ablations show that HTA relies most strongly on temporal-context modeling and sufficient historical context: removing TC and collapsing ARTS to minimal context reduce MRR by 15.63 and 14.62 points, respectively. Self-attention and token mixing are also complementary, each causing over an 11-point MRR drop when removed, while the auxiliary MLM loss and broader HCA scope provide smaller but consistent gains. The same trend holds across the additional datasets in Appendix D.

5.4 Hyperparameter Analysis

We tuned HTA’s key hyperparameters on the ICEWS14 validation set to establish an optimal balance between performance and computational cost. Our main configuration employs an encoder depth of $N = 6$ Attention-Mixer Transformer Blocks and

a maximum ARTS context length of $L_{ctx} = 50$, analyzed in Section 5.5. For the embedding dimension, our analysis (shown in Figure 4) revealed that performance on the ICEWS datasets generally peaked at $d = 320$. However, for the sparser and higher-frequency GDELT dataset, a more compact dimension of $d = 256$ yielded optimal results. Accordingly, we used $d = 320$ for the ICEWS datasets and $d = 256$ for GDELT in our final experiments, while keeping the remaining architecture and training settings shared across datasets. A transfer check using the ICEWS14 configuration on the other datasets changes MRR by at most 1.26 points, suggesting that HTA does not rely on heavy per-dataset tuning.

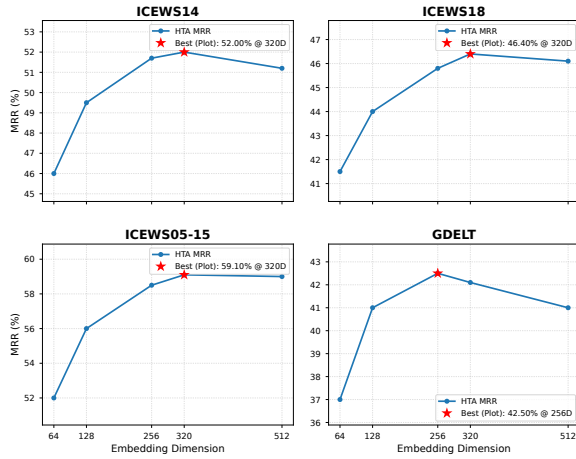


Figure 4: Performance (MRR) of HTA with varying embedding dimensions across different datasets.

5.5 Analysis of ARTS Context Length

The maximum ARTS context length controls how much historical evidence each query can attend to. Figure 5 shows that performance on ICEWS14 peaks at $L_{ctx} = 50$: increasing the context from 25 to 50 improves MRR from 49.35 to 52.04, while longer contexts (75 or 100) bring a mild decline. This trend indicates that HTA benefits from sufficiently long temporal evidence, but indiscriminately adding more historical facts can introduce noise and dilute attention. We attribute the decline beyond $L_{ctx} = 50$ to two effects: longer sequences are more likely to include outdated or weakly related events, and the fixed attention budget is spread across more tokens. Consistently, the entropy diagnostic in Appendix C shows final-layer attention entropy rising from 1.92 at $L_{ctx} = 25$ to 2.58 at $L_{ctx} = 75$, indicating less concentrated evidence selection. We thus treat $L_{ctx} = 50$ as an empirically validated trade-off that captures sufficient

long-term dependencies while limiting contextual noise and attention dilution.

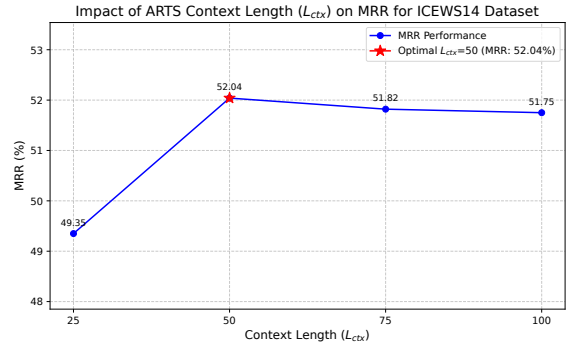


Figure 5: Impact of ARTS context length (L_{ctx}) on MRR performance for HTA on the ICEWS14 dataset.

5.6 Analysis of GPU Memory Consumption

We analyzed peak GPU memory consumption as a function of key architectural parameters: model depth (MLP-Mixer layers N) and ARTS context length (L_{ctx}). As illustrated in Figure 6, the memory-profile sweep was measured on NVIDIA A800 GPUs and reports per-GPU peak memory. Peak memory consumption clearly increases with both a greater number of layers and longer ARTS sequences. This highlights the necessity of balancing model complexity against available hardware resources, particularly for large-scale applications. Our selected parameters ($N = 6$ and $L_{ctx} = 50$) aim for a strong performance-to-resource efficiency. In contrast, the front-end HCA sampling cost is bounded by $O(L_{path}M)$, and the encoder input is capped at L_{ctx} facts. With a pre-built neighbor index, ARTS preparation adds only a near-constant per-query overhead across datasets (Appendix E), so the dominant cost of scaling HTA comes from the encoder stack rather than from sequence preparation.

6 Conclusions

We introduce HTA, a Hybrid Transformer Architecture for Temporal Knowledge Graph forecasting that constructs query-specific Attention-Ready Temporal Sequences (ARTS) via Historical Context Aggregation, augments historical timestamps with relation-aware temporal context based on query-relative time differences, and refines the sequence with Attention-Mixer blocks. Across four benchmarks HTA achieves state-of-the-art performance, with especially large gains on the sparse, high-frequency GDELT dataset. Ablations confirm that temporal-context encoding and sufficient

Measured Per-GPU Peak Memory vs. Model Depth and Context Length

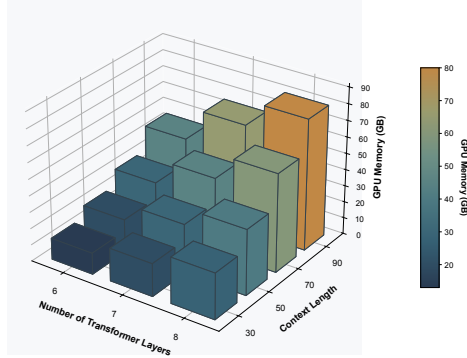


Figure 6: Per-GPU peak memory (GB) of HTA under different model depths and ARTS context lengths, measured on NVIDIA A800 GPUs.

ARTS context drive most of the improvement, while self-attention, token mixing, and the auxiliary loss are complementary.

Limitations

Despite its strong performance, our model has limitations. Its computational requirements for ARTS construction and sequence processing can be demanding, particularly for very long histories or massive graphs, potentially impacting scalability and requiring substantial GPU resources. Key areas for future enhancement include improving model interpretability, learning adaptive context-selection policies, extending the architecture to handle continuous timestamps instead of discrete ones, and incorporating mechanisms for inductive forecasting to manage unseen entities. Additionally, although our main configuration transfers well across datasets, extreme context lengths and dropout settings can still affect performance and should be selected with validation data. Addressing these limitations presents avenues for future research.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (62276057) and the Fundamental Research Funds for the Central Universities (No. N26GFZ009).

References

Long Bai, Zixuan Li, Xiaolong Jin, Jiafeng Guo, Xueqi Cheng, and Tat-Seng Chua. 2025. [G2S: A general-to-specific learning framework for temporal knowledge graph forecasting with large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20927–20938, Vienna, Austria. Association for Computational Linguistics.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. [TuckER: Tensor factorization for knowledge graph completion](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194, Hong Kong, China. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS’13*, page 2787–2795, Red Hook, NY, USA. Curran Associates Inc.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O’Brien, Steve Shellman, James Starz, and Michael Ward. 2015. [ICEWS Coded Event Data](#).

Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2023. [Temporal knowledge graph completion: A survey](#). In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-2023*, page 6545–6553. International Joint Conferences on Artificial Intelligence Organization.

Hao Dong, Pengyang Wang, Meng Xiao, Zhiyuan Ning, Pengfei Wang, and Yuanchun Zhou. 2024. Temporal inductive path neural network for temporal knowledge graph reasoning. *Artificial Intelligence*, 329:104085.

Yanglei Gan, Peng He, Yuxiang Cai, Run Lin, Guanyu Zhou, and Qiao Liu. 2026. [Negative-aware diffusion process for temporal knowledge graph extrapolation](#). In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 3352–3367, Rabat, Morocco. Association for Computational Linguistics.

Yifu Gao, Yongquan He, Zhigang Kan, Yi Han, Linbo Qiao, and Dongsheng Li. 2023. [Learning joint structural and temporal contextualized knowledge embeddings for temporal knowledge graph completion](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 417–430, Toronto, Canada. Association for Computational Linguistics.

Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. [Learning sequence encoders for temporal knowledge graph completion](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821, Brussels, Belgium. Association for Computational Linguistics.

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.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. xerte: Explainable reasoning on temporal

- knowledge graphs for forecasting future links. *arXiv e-prints*, pages arXiv–2012.
- Rikui Huang, Wei Wei, Xiaoye Qu, Shengzhe Zhang, Danyang 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 inference over 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.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. *arXiv preprint arXiv:2004.04926*.
- Julien Leblay and Melisachew Wudage Chekol. 2018. [Deriving validity time in knowledge graph](#). In *Companion Proceedings of the The Web Conference 2018*, WWW ’18, page 1771–1776, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Kalev Leetaru and Philip A. Schrod. 2013. [Gdelt: Global data on events, location, and tone](#). *ISA Annual Convention*.
- Zixuan Li, Saiping Guan, Xiaolong Jin, Weihua Peng, Yajuan Lyu, Yong Zhu, Long Bai, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022a. [Complex evolutionary pattern learning for temporal knowledge graph reasoning](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 290–296, Dublin, Ireland. Association for Computational Linguistics.
- Zixuan Li, Zhongni Hou, Saiping Guan, Xiaolong Jin, Weihua Peng, Long Bai, Yajuan Lyu, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022b. [HiSMATCH: Historical structure matching based temporal knowledge graph reasoning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7328–7338, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutionary representation learning. In *SIGIR*.
- Qika Lin, Jun Liu, Rui Mao, Fangzhi Xu, and Erik Cambria. 2023. [TECHS: Temporal logical graph networks for explainable extrapolation reasoning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1281–1293, Toronto, Canada. Association for Computational Linguistics.
- Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4120–4127.
- Namyong Park, Fuchen Liu, Purvanshi Mehta, Dana Cristofor, Christos Faloutsos, and Yuxiao Dong. 2022. [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, WSDM ’22*, page 794–803. ACM.
- Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. 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*, pages 593–607. Springer.
- Haohai Sun, Shangyi Geng, Jialun Zhong, Han Hu, and Kun He. 2022. [Graph Hawkes transformer for extrapolated reasoning on temporal knowledge graphs](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7481–7493, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8306–8319.
- Jinze Sun, Yongpan Sheng, Lirong He, Yongbin Qin, Ming Liu, and Tao Jia. 2025. [CEGRL-TKGR: A causal enhanced graph representation learning framework for temporal knowledge graph reasoning](#). In *Proceedings of Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning @ COLING 2025*, pages 6–17, Abu Dhabi, UAE. ELRA and ICCL.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.
- Guo Tang, Zheng Chu, Wenxiang Zheng, Junjia Xi, Yizhuo Li, Weihao Zhang, Ming Liu, and Bing Qin. 2025. [AnRe: Analogical replay for temporal knowledge graph forecasting](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

4632–4650, Vienna, Austria. Association for Computational Linguistics.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, and 1 others. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272.

Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *international conference on machine learning*, pages 3462–3471. PMLR.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiao-Yu Zhang. 2024. [Chain-of-history reasoning for temporal knowledge graph forecasting](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16144–16159, Bangkok, Thailand. Association for Computational Linguistics.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020. [TeRo: A time-aware knowledge graph embedding via temporal rotation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1583–1593, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*.

Shengzhe Zhang, Wei Wei, Rikui Huang, Wenfeng Xie, and Danyang Chen. 2024. [Modeling historical relevant and local frequency context for representation-based temporal knowledge graph forecasting](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7675–7686, Miami, Florida, USA. Association for Computational Linguistics.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *AAAI*.

Algorithm 1 Historical Context Aggregation (HCA) Algorithm

```

1: Input: Query  $(s_q, r_q, ?, t_q)$ , TKG history  $\mathcal{G}_{<t_q}$ , maximum path budget  $L_{\text{path}}$ , neighborhood size  $M$ .
2: Output: Path facts  $\mathcal{P}$  and aggregated context  $\mathcal{N}_{\text{ctx}}$ .
3: Initialize path facts  $\mathcal{P} \leftarrow []$ , path entities  $\mathcal{E}_{\text{path}} \leftarrow [s_q]$ , context set  $\mathcal{N}_{\text{ctx}} \leftarrow \emptyset$ .
4:  $s_{\text{current}} \leftarrow s_q$ .
5: for  $i = 1$  to  $L_{\text{path}} - 1$  do
6:   Let  $\mathcal{C}$  be historical facts incident to  $s_{\text{current}}$ , oriented as  $(s_{\text{current}}, r', e', t')$  via reciprocal conversion when needed.
7:   if  $\mathcal{C}$  is empty then break
8:   end if
9:   Sample a fact  $F_i = (s_{\text{current}}, r', e', t')$  uniformly from  $\mathcal{C}$ .
10:   $\mathcal{P} \leftarrow \mathcal{P} \cup \{F_i\}$ ,  $\mathcal{N}_{\text{ctx}} \leftarrow \mathcal{N}_{\text{ctx}} \cup \{F_i\}$ .
11:   $s_{\text{current}} \leftarrow e'$ .
12:  Append  $s_{\text{current}}$  to  $\mathcal{E}_{\text{path}}$ .
13: end for
14: for each entity  $s_k \in \mathcal{E}_{\text{path}}$  do
15:   Let  $\mathcal{N}_k$  be historical facts incident to  $s_k$ , oriented as  $(s_k, r_j, e_j, t_j)$ .
16:   Sample a subset  $\mathcal{S}_k \subseteq \mathcal{N}_k$  of size at most  $M$ .
17:    $\mathcal{N}_{\text{ctx}} \leftarrow \mathcal{N}_{\text{ctx}} \cup \mathcal{S}_k$ .
18: end for
19: return  $\mathcal{P}, \mathcal{N}_{\text{ctx}}$ .

```

A Historical Context Aggregation Algorithm

B Extended Ablation Study on ARTS Construction

This appendix provides additional ablation studies to further validate the design choices of our proposed Attention-Ready Temporal Sequence (ARTS).

B.1 Analysis of Sequence Ordering

To empirically validate the importance of the chronological ordering within ARTS, we conducted an ablation study comparing our full model against two carefully designed baselines on the ICEWS14 dataset.

Methodology We evaluate the following variants. **ARTS (Ours):** The standard model as presented in the main paper, where historical facts selected

by the Historical Context Aggregation (HCA) module are sorted chronologically. **ARTS-random:** A variant that uses the exact same set of historical facts selected by HCA, but the chronological order of these facts within the sequence is randomly shuffled. This isolates the impact of temporal ordering. **Random Sequences:** A more basic baseline where the HCA module is bypassed entirely. Instead, historical facts are randomly sampled from the TKG history to form the input sequence. This baseline demonstrates the value of the HCA selection process itself.

Results and Analysis The results, presented in Table 4, confirm that both the selection and the ordering of historical context are critical. The substantial performance gap between ARTS-random and Random Sequences (e.g., an MRR improvement of 18.94 points) highlights the crucial role of our HCA module in identifying relevant historical context. Furthermore, the superior performance of the original ARTS model over ARTS-random demonstrates that preserving the chronological sequence of these events provides vital temporal cues that the model leverages for more accurate forecasting.

Method	MRR	Hits@1	Hits@3	Hits@10
ARTS (Ours)	52.04	45.99	54.52	63.75
ARTS-random	50.48	43.99	53.15	63.13
Random Sequences	31.54	26.58	32.96	40.59

Table 4: Ablation study on the impact of sequence ordering and context selection on the ICEWS14 dataset. Results validate the necessity of both the HCA process and chronological ordering.

C Detailed Analysis of ARTS Context Length (L_{ctx})

This section provides additional diagnostics for the final ARTS context length L_{ctx} summarized in Section 5.5. It is distinct from the HCA path budget L_{path} in Algorithm 1; both are set to 50 in the main configuration.

Factors Influencing Performance Degradation for $L_{\text{ctx}} > 50$ We attribute the performance decline for sequences longer than the optimal length to two primary factors. First, the introduction of contextual noise occurs as excessively long contexts risk incorporating outdated or irrelevant historical events. This noise can obscure the signal from more causally important facts, hindering the model’s ability to generalize. Second, attention

dilution becomes problematic because the self-attention mechanism operates with a fixed probability mass. As the sequence length increases, this attention budget is spread more thinly across a larger number of tokens, making it difficult for the model to assign sufficiently high weights to the few truly decisive historical events.

To provide quantitative evidence for attention dilution, we measured the mean attention entropy of the final layer across different context lengths. As shown in Table 5, attention entropy—a measure of the uniformity of the attention distribution—steadily increases with L_{ctx} . A higher entropy indicates a more spread-out, less focused attention mechanism. This finding supports our hypothesis that longer sequences can lead to a less decisive model.

Context Size (L_{ctx})	Mean Attention Entropy	Std Dev of Entropy
25	1.92	0.52
50	2.33	0.57
75	2.58	0.59

Table 5: Analysis of mean attention entropy as a function of ARTS context length (L_{ctx}) on ICEWS14. Higher entropy at longer sequence lengths suggests increased attention dilution.

The selection of $L_{\text{ctx}} = 50$ represents an empirically validated trade-off, providing an optimal balance that captures sufficient long-term dependencies while mitigating the adverse effects of contextual noise and attention dilution.

D All-Dataset Component Ablations

Table 6 reports the additional component ablations on the three datasets beyond the ICEWS14 main-table analysis. The trends are consistent across datasets: removing Temporal Context (TC) or reducing ARTS to minimal context causes the largest degradation, while the auxiliary loss and HCA scope provide smaller but consistent gains.

E Efficiency and Robustness Details

Table 7 reports wall-clock training time, full test-set inference time, and average ARTS construction time per query. Training and inference are measured on four NVIDIA RTX4090 GPUs, while the memory-profile sweep in Figure 6 is measured on NVIDIA A800 GPUs. HCA neighborhood sampling is bounded by $O(L_{\text{path}}M)$ with $L_{\text{path}} = 50$ and $M = 50$, while the retained encoder context is capped at $L_{\text{ctx}} = 50$ facts. With a pre-built

Dataset	Variant	MRR	Hits@1	Hits@3	Hits@10
GDELT	HTA (Full)	42.44	34.94	46.01	55.29
	HTA w/o Temporal Context (TC)	26.18	18.45	28.72	40.33
	HTA w/o Aux-Loss (MLM)	39.15	31.82	42.58	52.47
	HTA w/ Min ARTS Context (L_{ctx})	25.73	18.02	28.15	39.87
	HTA w/ Reduced HCA Scope (M)	38.92	31.58	42.25	52.13
ICEWS18	HTA (Full)	46.35	39.44	49.43	58.89
	HTA w/o Temporal Context (TC)	30.59	21.34	35.14	47.88
	HTA w/o Aux-Loss (MLM)	42.93	35.19	46.28	57.01
	HTA w/ Min ARTS Context (L_{ctx})	30.21	21.03	34.72	47.63
	HTA w/ Reduced HCA Scope (M)	42.78	35.07	46.10	56.92
ICEWS05-15	HTA (Full)	59.14	52.27	62.77	71.72
	HTA w/o Temporal Context (TC)	43.76	33.92	48.90	61.46
	HTA w/o Aux-Loss (MLM)	55.96	48.22	59.92	70.10
	HTA w/ Min ARTS Context (L_{ctx})	43.44	33.64	48.55	61.24
	HTA w/ Reduced HCA Scope (M)	56.14	48.37	60.04	70.18

Table 6: Ablation study on GDELT, ICEWS18, and ICEWS05-15 (values in %).

neighbor index, its per-query time remains nearly constant across datasets. The GPU memory analysis reports per-GPU peak memory consumption rather than average memory.

E.1 HCA Sampling Robustness

Randomness in HCA is controlled by the fixed seeds described in Section 5.1.4. On GDELT, five runs with different seeds give MRR 42.42 ± 0.31 , Hits@1 34.91 ± 0.28 , Hits@3 45.98 ± 0.35 , and Hits@10 55.25 ± 0.27 . As an additional ICEWS14 check, random neighbor selection (52.04 MRR) slightly outperforms deterministic most-recent selection (51.38 MRR), suggesting that the stochastic HCA policy is not a source of instability.

For neighborhood coverage, ICEWS14 has an average of 127.5 neighbors per entity (median 89), with 28.3% of entities having fewer than 50 neighbors and a 3.2% early path-termination rate. GDELT has an average of 358.2 neighbors per entity (median 215), with 12.7% of entities having fewer than 50 neighbors and a 1.1% early termination rate.

F Datasets

We evaluate our model on four widely used benchmark datasets: ICEWS14, ICEWS18, ICEWS05-15, and GDELT. These datasets consist of time-stamped events that record interactions between entities. The ICEWS datasets (Integrated Crisis Early Warning System) capture international political events with a time granularity of 24 hours. In contrast, GDELT (Global Database of Events, Language, and Tone) is much larger and records events at 15-minute intervals. Detailed statistics for each

dataset, including the counts of entities, relations, and data splits, are summarized in Table 8.

Dataset	#Entities	#Facts (all splits)	ARTS Time (ms/query)	Train Time (h)	Inference Time (s)
ICEWS14	7,128	90,740	2.35	6.2	42.5
ICEWS18	23,033	468,558	2.42	19.5	135.8
ICEWS05-15	10,488	460,819	2.38	18.8	128.4
GDELT	7,691	2,278,405	2.48	48.5	856.3

Table 7: Scalability and efficiency analysis across datasets.

Dataset	# Entity	# Relation	# Train	# Valid	# Test	Interval
ICEWS14	7,128	230	72,837	8,940	8,963	24 hours
ICEWS18	23,033	256	373,018	45,995	49,545	24 hours
ICEWS05-15	10,488	251	368,672	46,093	46,054	24 hours
GDELT	7,691	240	1,734,399	238,765	305,241	15 mins

Table 8: Statistics of datasets used in the experiments.