Fine-tuning foundation models for temporal knowledge graph reasoning

Manuel Dileo[1][0000−0002−4861−455X] (✉), Pasquale Minervini[2,3][0000−0002−8442−602X], and Matteo Zignani[1][0000−0002−4808−4106]

[1] Department of Computer Science, University of Milan, Milan, Italy
{manuel.dileo,matteo.zignani}@unimi.it
[2] School of Informatics, University of Edinburgh, Edinburgh, UK
[3] miniml.AI, UK
p.minervini@miniml.ai

Abstract. Foundation models have recently demonstrated strong performance in various knowledge graph reasoning tasks. However, their applicability to temporal knowledge graphs (TKGs), where facts evolve over time, remains underexplored. In this work, we investigate whether a foundation model designed for knowledge graph reasoning can be adapted to temporal reasoning through fine-tuning. Specifically, we extend and fine-tune ULTRA [5] for temporal knowledge graph forecasting tasks. To this end, we adapt the training and evaluation setting of the model, originally designed to perform KG completion tasks, to KG forecasting tasks. Furthermore, we allow ULTRA to incorporate temporal information of facts and queries, in the form of quadruples, via positional encoding of timestamps. Experimental results on standard TKG benchmarks reveal that fine-tuned ULTRA achieves competitive performance with state-of-the-art (SOTA) supervised competitors, particularly on the ICEWS datasets. These datasets emphasize entity-driven prediction over time, where relational patterns are sparse and events such as diplomatic visits or negotiations often occur once without strong temporal regularities. However, on more structurally and temporally rich datasets like YAGO, GDELT, and WIKI, ULTRA falls short of SOTA supervised models, which leverage relational temporal dynamics and evolving patterns more effectively. These findings suggest that while static foundation models can be effectively fine-tuned for certain types of temporal reasoning, they lack the inductive biases necessary to fully capture evolving relational structures. This underscores the development of foundation models explicitly tailored for temporal knowledge graphs as a promising research direction for mining and learning complex patterns from these systems.

Keywords: Temporal Knowledge Graphs · Graph Neural Networks · Foundation Models.

1 Introduction

Knowledge Graphs (KGs) have emerged as a powerful abstraction for representing structured knowledge across a wide range of domains, enabling tasks

such as link prediction, question answering, and reasoning [10], which allow the extraction of complex and semantically rich patterns from heterogeneous and interlinked data. Recent advances in foundation models have shown promise in unifying various KG reasoning tasks under a shared framework, achieving strong generalization across datasets and tasks [5, 4]. These models, pre-trained on large-scale static KGs and fine-tuned for specific downstream applications, aim to offer scalability and adaptability without requiring training architectures for every task. One prominent example is ULTRA [5], a graph neural network (GNN)-based architecture that frames knowledge graph completion as a unified classification problem over $(\texttt{subject}, \texttt{relation}, \texttt{?})$ and $(\texttt{?}, \texttt{relation}, \texttt{object})$ queries, being able to extend its reasoning over unseen entities and relations, demonstrating competitive performance across multiple static KG benchmarks.

However, in many real-world applications, knowledge is not static but evolves over time. Temporal Knowledge Graphs (TKGs) extend traditional KGs by associating each fact with a timestamp or a temporal interval, enabling reasoning over dynamic and time-dependent information [1]. In particular, TKGs provide a natural and expressive framework for modeling complex and evolving event data, such as social interactions, geopolitical events, or dynamic biomedical records, making them highly relevant for the analysis of massive and time-sensitive data sources [12, 9]. Yet, reasoning over TKGs introduces unique challenges, including temporal consistency, evolving relational patterns, and the sparsity of recurring events [6, 7]. Unlike static KGs, temporal reasoning requires not only structural understanding but also the ability to capture complex, time-sensitive patterns. Despite the growing importance of TKGs, the applicability of foundation KG models to temporal reasoning tasks remains underexplored. In this work, we address this gap by investigating whether a foundation model designed for static KGs can be adapted for TKG forecasting through fine-tuning. We choose ULTRA as a case study and modify its design and code to perform multistep link forecasting, a task that requires predicting future quadruples over multiple timestamps based on historical information of a temporal knowledge graph.

To this end, we make two key adaptations. First, we modify the training and evaluation setup and the loss function of ULTRA, originally intended for static KG completion, to fit the temporal forecasting setting. This distinction is crucial because knowledge graph completion assumes access to the entire KG, including both past and future quadruples relative to any given query. The model operates under the assumption that the graph is fully available, and it learns to infer missing links within that static snapshot. In contrast, temporal forecasting imposes a stricter and more realistic constraint: the model can only observe the TKG up to a certain timestamp, and must predict quadruples that occur in the future. This makes forecasting inherently more challenging, as it requires the model to learn temporal dynamics and anticipate unseen events without relying on future context. Second, we augment ULTRA's input representation with explicit temporal signals by integrating positional encodings of timestamps into the model. This enhancement enables the architecture to account not only for the structural information of the graph but also for the temporal context

in which events occur. In the temporal setting, it is crucial that the model distinguishes between different instantiations of the same query across time. For example, a quadruple $(\texttt{subject}, \texttt{predicate}, \texttt{object}, \texttt{t})$ may hold true at time `t` but be invalid at another timestamp `t'`. By incorporating timestamp encodings, ULTRA can learn to assign different scores to the same structural triple depending on its temporal position.

Our empirical evaluation is conducted on widely-used TKG benchmarks, including ICEWS14, ICEWS18 [6], YAGO [3], GDELT [12], and WIKI [8]. Results show that the fine-tuned ULTRA model achieves competitive performance with supervised SOTA baselines on datasets like ICEWS, which are characterized by entity-centric predictions and sparse, non-repetitive relational patterns (e.g., diplomatic meetings, protests). These scenarios benefit from the model's capacity to generalize from entity-level dynamics without relying heavily on complex temporal dependencies. In contrast, ULTRA underperforms on datasets such as YAGO, WIKI, and GDELT, which feature richer temporal dynamics and more structured, repeating relational patterns [15]. These results indicate that while fine-tuning static foundation models can offer a promising starting point for temporal reasoning, such models lack the inductive biases required to effectively model evolving relational structures and time-dependent behaviors.

Our findings emphasize the limitations of naïvely extending static models to temporal settings and highlight the need for dedicated foundation models tailored to temporal reasoning. We argue that gathering a rich collection of temporal knowledge graphs for pre-training, and designing inductive temporal representations and sequence modeling techniques for a dynamic graph foundational model, is a promising research direction for advancing the state of learning complex patterns from temporal knowledge graphs.

Main Contributions. This paper makes the following key contributions: *i)* **Adapting a foundation model for temporal reasoning**. We extend the ULTRA model, originally developed for static knowledge graph completion, to handle temporal knowledge graph forecasting tasks. This includes modifications to the training and evaluation pipeline, design of the loss function, and the integration of temporal signals via positional encodings of timestamps, enabling the model to reason over time-sensitive facts. Code is available on a Github repository[4]; *ii)* **Comprehensive benchmarking on temporal forecasting tasks.** We evaluate the fine-tuned ULTRA on a multi-step forecasting task across several TKG datasets, using the standard evaluation protocol for temporal link prediction.

2 Background

Problem definition. A Temporal Knowledge Graph (TKG) can be defined as an ordered sequence of timestamped knowledge graph snapshots $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_t, \ldots)$. Each snapshot $\mathcal{G}_t = (\mathcal{V}, \mathcal{R}, \mathcal{E}_t)$ represents the state of the knowledge graph at a discrete time step t, where $\mathcal{V}$ is the set of entities, $\mathcal{R}$ is the set of relations,

[4] https://anonymous.4open.science/r/ULTRA-D157/README.md

and $\mathcal{E}_t$ denotes the set of timestamped factual triples (or *quadruples*) observed at time t. Each fact $(h, r, v, t) \in \mathcal{E}_t$ consists of a subject entity $h \in \mathcal{V}$, a relation $r \in \mathcal{R}$, an object entity $v \in \mathcal{V}$, and a timestamp t. For example, the quadruple (`Angela Merkel`, `meet`, `Pope Francis`, `2018-06-08`) captures a temporal fact indicating that a visit took place on the given date. In the context of TKG forecasting, the goal is typically formulated as a link prediction task: given a TKG observed up to time t and a query with missing subject or object, such as $(h, r, ?, t + k)$ or $(?, r, v, t + k)$ for a future timestamp $t + k$, where $k \in \mathbb{N}^+$, the model is expected to predict the most likely missing entity [13].

Temporal Knowledge Graph Forecasting. In recent years, the task of forecasting future links in temporal knowledge graphs has gained increasing attention, with a wide range of methods being proposed [8]. A significant line of research has focused on combining graph neural networks with temporal modeling techniques to capture both the structural and dynamic aspects of evolving knowledge graphs. These methods typically encode the graph topology at each timestamp while modeling temporal dependencies across snapshots using sequential architectures. Examples include approaches that use recurrent networks, such as RE-Net [11] and RE-GCN [13], message-passing, and attention mechanisms to capture evolving patterns in entity and relation interactions. In addition to sequential GNN-based approaches, some works have explored alternative paradigms. Reinforcement learning has been used to model multi-hop reasoning over time-aware graphs, allowing the agent to simulate temporal exploration paths for forecasting future events [17]. Other approaches, such as TLogic [14], rely on logic-based inference, employing symbolic reasoning to discover temporal rules that govern entity behavior over time. Furthermore, CyGNet [20] is based on pattern repetition and frequency of past events and offers simpler yet effective strategies by exploiting regularities in the temporal evolution of quadruples.

Graph Foundation Models. Recent years have seen a growing interest in developing graph foundation models (GFMs), that is, machine learning architectures capable of generalizing across previously unseen graphs, tasks (such as node or graph classification), and semantic vocabularies, i.e., meaning, related to links between nodes. The overarching goal is to build unified models that can operate out-of-the-box on arbitrary relational data, enabling transferability and reducing the need for task-specific training. One notable example is ULTRA [5], which introduces a framework for inductive reasoning over knowledge graphs with disjoint entity and relation vocabularies. ULTRA learns relational representations conditioned on their structural interactions, making it possible to perform zero-shot inference on unseen KGs and further improve through fine-tuning. GraphAny [19] addresses the fully-inductive node classification setting by modeling inference on new graphs as combinations of solutions from a family of LinearGNNs, guided by an attention mechanism designed to generalize to new feature and label spaces. In the domain of structured enterprise data, KumoRFM [4] proposes a relational foundation model that can perform predictive tasks directly on arbitrary relational databases without task-specific supervision. It combines

table-agnostic encoding with a relational graph transformer and demonstrates competitive performance across diverse applications. While most of these efforts focus on static graphs, a first step toward temporal generalization is represented by MiNT [16], a method designed for transfer learning on temporal graphs using multi-network training. However, MiNT targets graph classification and does not yet support inductive inference at the link or entity level. In our work, we contribute to this emerging landscape by exploring the adaptation of a knowledge graph foundation model, ULTRA, to the task of temporal knowledge graph forecasting. We investigate whether pre-trained static KG representations can be extended and fine-tuned to support inductive reasoning over evolving temporal data, bridging the gap between foundation models and time-aware relational inference.

3 Methodology

3.1 Adapting the ULTRA Algorithm to TKG

How ULTRA works. Given a query $(h, q, ?)$ over a static graph $\mathcal{G}$, ULTRA employs a three-step algorithm to obtain the scores $p(h, q, v)$ for each possible node v to be a tail of the initial query:

1. *Lift the original graph $\mathcal{G}$ to the graph of relations $\mathcal{G}_r$.* Starting from a graph $\mathcal{G} = (V, \mathcal{R}, \mathcal{E})$, ULTRA first constructs a lifted representation (i.e., a higher-level abstraction of the original graph) $\mathcal{G}_r = \textsc{Lift}(\mathcal{G})$, yielding a relation graph $\mathcal{G}_r = (\mathcal{R}, \mathcal{R}_{fund}, \mathcal{E}_r)$ in which nodes correspond to distinct relation types[5]. The edge set $\mathcal{E}_r \in (\mathcal{R} \times \mathcal{R}_{fund} \times \mathcal{R})$ captures the interactions between relations present in the original graph $\mathcal{G}$. Four fundamental interaction types are considered: *tail-to-head (t2h)*, *head-to-head (h2h)*, *head-to-tail (h2t)*, and *tail-to-tail (t2t)*.
2. *Obtain relative relation representations $R_q|(q, \mathcal{G}_r)$ conditioned on the query relation q in the relation graph $\mathcal{G}_r$.* Given a query $(h, q, ?)$ and the relation graph $\mathcal{G}_r$, ULTRA derives d-dimensional node embeddings $\boldsymbol{R}_q \in \mathbb{R}^{|\mathcal{R}| \times d}$ representing all relations in $\mathcal{G}$, conditioned on the specific query relation q. This conditioning is realized by initializing the node q in $\mathcal{G}_r$ using the $\textsc{Indicator}_r$ function, and applying message passing through a GNN defined over $\mathcal{G}_r$:

$$\boldsymbol{h}^0_{v|q} = \textsc{Indicator}_r(v, q) = \mathbb{1}_{v=q} * \mathbf{1}^d, \quad v \in \mathcal{G}_r$$

$$\boldsymbol{h}^{t+1}_{v|q} = \textsc{Update}\Big(\boldsymbol{h}^t_{v|q}, \textsc{Aggregate}\big(\textsc{Message}(\boldsymbol{h}^t_{w|q}, \boldsymbol{r}) \mid w \in \mathcal{N}_r(v), r \in \mathcal{R}_{fund}\big)\Big)$$

   The graph neural network used at this stage, denoted GNN_r, follows the NBFNet [21] framework and employs a non-parametric DistMult [18] message function combined with sum aggregation.
3. *Using the relation representations R_q as starting relation features, run inductive link prediction on the original graph $\mathcal{G}$.* With the query $(h, q, ?)$ and

[5] Resulting in $2|\mathcal{R}|$ nodes when including inverse relations.

the conditioned relation embeddings $\boldsymbol{R}_q$ obtained previously, ULTRA proceeds to run inductive link prediction on the original graph $\mathcal{G}$. This step leverages another GNN (again, based on NBFNet), where the initial feature for the head node h is set to the corresponding query vector from $\boldsymbol{R}_q$, while other nodes are initialized to zero:

$$\boldsymbol{h}^0_{v|u} = \textsc{Indicator}_e(u, v, q) = \mathbb{1}_{u=v} * \boldsymbol{R}_q[q], \quad v \in \mathcal{G}$$

$$\boldsymbol{h}^{t+1}_{v|u} = \textsc{Update}\Big(\boldsymbol{h}^t_{v|u}, \textsc{Aggregate}\big(\textsc{Message}(\boldsymbol{h}^t_{w|u}, g^{t+1}(\boldsymbol{r})) \mid w \in \mathcal{N}_r(v), r \in \mathcal{R}\big)\Big)$$

Each t-th GNN layer applies a non-linear function $g^t(\cdot)$ to transform original relation representations to layer-specific relation representations as $\boldsymbol{R}^t = g^t(\boldsymbol{R}_q)$ from which the edge features are taken for the Message function. $g(\cdot)$ is implemented as a 2-layer MLP with ReLU. After message passing, a final MLP $s : \mathbb{R}^d \to \mathbb{R}^1$ produces logits $p(h, q, v)$ indicating the likelihood that node v completes the query $(h, q, ?)$ as its tail entity.

Adding temporal information. Given a timestamped query $(h, q, ?, q_t)$, we add a new Step 2b in the ULTRA Algorithm to take into account the temporal information:

– (2b) *Obtain relative timestamp representations $T_q|(q_t)$ conditioned on the query timestamp q_t.* We introduce a time-encoding function $\cos(t\boldsymbol{\omega})$, which utilizes features $\boldsymbol{\omega} = \{\alpha^{-(i-1)/\beta}\}_{i=1}^d$ to encode each timestamps into a d-dimensional vector. More specifically, we first map each timestamp t to a vector with monotonically exponentially decreasing values $t\boldsymbol{\omega} \in (0, t]$ among the feature dimension, then use cosine function to project all values to $\cos(t\boldsymbol{\omega}) \in [-1, +1]$. Following the work that proposes this encoding function [2], we set $\alpha = \beta = \sqrt{d}$ for all the datasets, and d equal to the embedding dimension of ULTRA. Notice that $\boldsymbol{\omega}$ is fixed and will not be updated during training. As shown in Figure 1, the output of this time-encoding function has two main properties that could help distinguish different timestamps: similar timestamps have similar time-encodings (e.g., the plot of t_1, t_2) and the larger the timestamp the later the values in time-encodings converge to $+1$ (e.g., the plot of t_1, t_3 or t_1, t_4). To obtain relative timestamp representations conditioned on the query timestamp q_t, we compute $T_q[z]$ as the embedding of the time difference between timestamp z and the query timestamp q_t, that is $T_q[z] = \cos((q_t - z)\boldsymbol{\omega})$.

Then, we modify the entity-level link prediction component of ULTRA (Step 3) to obtain scores for queries $(h, q, ?, q_t)$:

$$\boldsymbol{h}^0_{v|u} = \textsc{Indicator}_e(u, v, q) = \mathbb{1}_{u=v} * \boldsymbol{R}_q[q] * \boldsymbol{T}_q[q_t], \quad v \in \mathcal{G}$$

$$\boldsymbol{h}^{t+1}_{v|u} = \textsc{Update}\Big(\boldsymbol{h}^t_{v|u}, \textsc{Aggregate}\big(\textsc{Message}(\boldsymbol{h}^t_{w|u}, g^{t+1}(\boldsymbol{r} * \boldsymbol{T}_q[z])) \mid (w, r, v, z) \in \mathcal{E}_z\big)\Big)$$

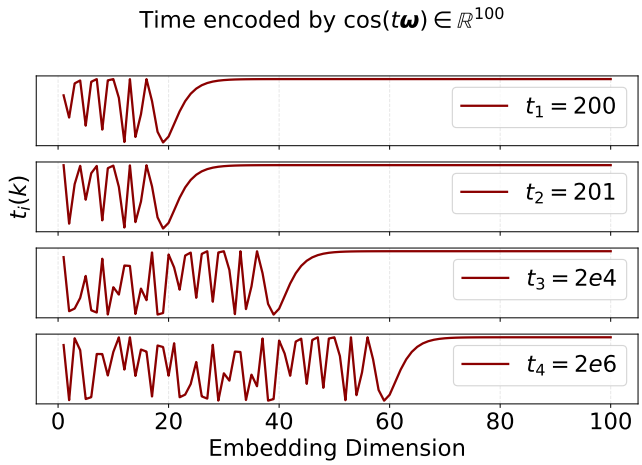

Fig. 1. Visualization of cosine-based time encoding across embedding dimensions for different timestamp values. Time-encoding function that pre-process timestamp t into a vector $\cos(t\boldsymbol{\omega})$. The x-axis is the vector dimension, and the y-axis is the cosine value.

3.2 Fine-tuning ULTRA on TKG forecasting

Training and evaluation setting. To adapt ULTRA for temporal knowledge graph forecasting, we transition from a static completion task to a dynamic forecasting setup. Specifically, instead of predicting missing entities in observed quadruples, the model is trained to assign scores to potential future facts, quadruples, that may occur at upcoming timestamps. To simulate this forecasting scenario, following previous works [13], we chronologically split the train set, reserving the last 10% of the temporal snapshots as the prediction interval, while training is performed on the preceding 90%. During training, it is essential to treat the same fact occurring at different times as distinct instances. Accordingly, negative samples are generated by corrupting the head or tail entity of positive quadruples at the same timestamp, ensuring the model learns time-specific representations. The loss is computed independently at each timestamp t in the prediction interval $[t_s, t_e]$, capturing the temporal evolution of the knowledge graph (see below). For evaluation, we follow the time-aware filtering protocol proposed in [8]. Unlike traditional filtering adopted by ULTRA, which removes all known true triples regardless of time, time-aware filtering only excludes quadruples occurring at the same timestamp as the test query. This avoids unfair penalization of the model for ranking temporally valid quadruples higher than the target. Furthermore, to prevent an information gap during testing, we allow the model to access true quadruples from the validation set when making predictions on the test set, as done in [8].

Loss function. Following established practices in the literature [13, 11], ULTRA is fine-tuned by minimizing the binary cross-entropy loss over both positive and negative quadruples across time:

$$\mathcal{L} = \sum_{t=t_s}^{t_e} \left(-\log p(u, q, v, t) - \sum_{i=1}^{n} \frac{1}{n} \log(1 - p(u'_i, q, v'_i, t)) \right)$$

Here, (u, q, v, t) denotes a positive quadruple observed in the graph, while $\{(u'_i, q, v'_i, t)\}_{i=1}^{n}$ are negative samples generated by corrupting either the head entity u or the tail entity v. These negative samples represent non-factual statements at the current timestamp.

4 Experiments

Experimental Setup. We conduct our experiments following the well-established evaluation protocol and dataset versions introduced in Gastinger *et al.* [8]. To ensure a fair comparison, we adopt their dataset splits and report state-of-the-art supervised model results from their work as baselines. Our evaluation focuses on the multi-step forecasting setting, which is a more general and challenging task compared to single-step forecasting, as it requires predicting quadruples of multiple timestamps into the future. For fine-tuning ULTRA, we train the model for a maximum of 10 epochs with early stopping based on validation performance. We restrict training to the last 10% of temporal snapshots in the training data. This approach balances the need for sufficient temporal context with computational efficiency. Notice that we empirically observe that increasing the training window beyond 10% does not lead to significant performance gains, as also highlighted in previous works [13, 11]. Following [8], during inference, ULTRA is allowed to leverage valid quadruples from the validation set as part of the candidate pool when predicting on the test set, thus avoiding an information gap between training and testing time intervals.

Datasets. We evaluate the fine-tuned ULTRA model and supervised baselines on five widely-used temporal knowledge graph datasets: WIKI, YAGO, ICEWS14, ICEWS18, and GDELT [8]. These datasets span diverse domains and temporal characteristics. WIKI and YAGO are derived from encyclopedic sources and capture long-term, structured relational quadruples evolving over time, such as biographical or organizational events. In contrast, ICEWS14 and ICEWS18 are extracted from the Integrated Crisis Early Warning System and contain geopolitical event sequences, often characterized by sparse relations and irregular, entity-centric patterns. GDELT, a large-scale open event dataset, records real-time global news events and exhibits high temporal granularity and dynamic relational structures. Table 1 summarizes key statistics of each dataset, including the number of entities, relations, and temporal quadruples. It also reports the number of temporal snapshots used for training, validation, and testing (Tr/Val/Te TS), based on the standard timestep-based data splits introduced in [8].

Table 1. Datasets statistics, timestep interval, and splitting points.

Dataset	#Nodes	#Rels	#Train	#Valid	#Test	Time Int.	#Tr/Val/Te TS
ICEWS14	7128	230	74845	8514	7371	24 hours	304/30/31
ICEWS18	23033	256	373018	45995	49545	24 hours	239/30/34
GDELT	7691	240	1734399	238765	305241	15 min.	2303/288/384
YAGO	10623	10	161540	19523	20026	1 year	177/5/6
WIKI	12554	24	539286	67538	63110	1 year	210/11/10

Results. Table 2, Table 3, and Table 4 report the performance of ULTRA and several supervised baselines on five standard TKG forecasting benchmarks under the multi-step prediction setting with time-aware filtering. Across all datasets, we observe that fine-tuning ULTRA leads to substantial gains over the zero-shot version, confirming the benefit of adapting foundation models to the temporal setting. This is particularly evident in ICEWS14, ICEWS18, and GDELT, where fine-tuning closes a significant portion of the performance gap with supervised methods. Moreover, integrating timestamp-based positional encodings (TE) further boosts performance on all datasets except YAGO. Notably, YAGO is the only dataset in our benchmark that includes both timestamped and non-timestamped facts, which likely explains why incorporating temporal encodings does not yield improvements and suggests that temporal encodings may introduce noise when applied to partially atemporal graphs. The ULTRA (TE + FT) variant achieves results that are either on par with or very close to the best supervised baselines on ICEWS14 and ICEWS18 datasets, where temporal reasoning is primarily entity-driven, and relational patterns are sparse and irregular [15]. In these cases, ULTRA proves highly competitive, occasionally even outperforming all supervised competitors in metrics such as Hits@1 or MRR. In contrast, on structurally richer and temporally regular datasets like GDELT and WIKI, supervised models like RE-GCN and CyGNet maintain a notable lead.

In summary, these results show that while ULTRA is not yet able to fully match SOTA supervised methods on datasets with strong relational temporal regularities, it can reach comparable performance on more entity-driven temporal KGs. The consistent benefit of temporal encoding across most benchmarks also highlights the value of explicitly modeling time in foundation models adapted for TKG reasoning.

5 Conclusion

In this work, we investigated the potential of adapting knowledge graph foundation models to temporal knowledge graph (TKG) forecasting tasks. By fine-tuning ULTRA, a recent foundation model for KG reasoning, we demonstrated that such models can achieve competitive results with supervised approaches on several benchmarks, particularly on datasets where temporal reasoning is largely driven by entities and exhibits sparse relational patterns. Our approach involved

Table 2. Results for temporal knowledge graph forecasting on ICEWS14 and ICEWS18 in the multi-step setting with time-filter triples. TE stands for time encoding, FT for fine-tuned.

Model	ICEWS14				ICEWS18			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RE-GCN [13]	**37.82**	27.86	**42.14**	**57.50**	**29.03**	**19.52**	**32.66**	47.50
RE-Net [11]	37.00	27.80	40.80	54.92	27.86	18.47	31.43	46.19
CyGNet [20]	36.12	26.66	40.28	54.54	26.01	16.69	29.59	44.43
TLogic [14]	35.48	26.54	39.59	53.11	24.01	15.59	27.23	41.20
ULTRA (Zero-shot)	26.86	17.99	30.07	44.60	11.11	06.13	12.07	21.42
ULTRA (Fine-tuned)	35.37	26.14	39.20	53.56	26.71	16.88	30.32	46.26
ULTRA (TE + FT)	37.72	**27.89**	42.08	57.13	28.33	18.02	32.34	**49.14**

Table 3. Results for temporal knowledge graph forecasting on YAGO and WIKI in the multi-step setting with time-filter triples. TE stands for time encoding, FT for fine-tuned.

Model	YAGO				WIKI			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RE-GCN [13]	**75.40**	**71.75**	**77.67**	81.70	62.72	59.48	64.89	67.87
RE-Net [11]	58.21	53.44	61.31	66.26	49.47	47.21	50.70	53.04
CyGNet [20]	69.02	61.38	74.29	**83.42**	58.26	52.51	62.41	67.56
TLogic [14]	66.93	63.14	70.63	71.58	**63.99**	**61.31**	**66.36**	**68.22**
ULTRA (Zero-shot)	63.11	55.33	69.11	76.71	45.68	37.25	51.41	61.91
ULTRA (Fine-tuned)	65.26	56.11	70.71	82.24	50.15	40.10	57.60	68.53
ULTRA (TE + FT)	63.64	54.05	69.55	82.35	53.43	45.10	58.74	68.73

extending ULTRA with temporal encodings and adapting its training and evaluation to the forecasting scenario, where the model must predict future events without access to future snapshots. Results highlight both the strengths and limitations of static foundation models in temporal contexts: while fine-tuned ULTRA performs well on certain benchmarks, it still lags behind state-of-the-art methods on temporally structured datasets like GDELT, YAGO, and WIKI. This can be partly attributed to the simplicity of our extension, which enables fine-tuning without introducing any new learnable parameters. Although this design ensures efficiency and compatibility with pretrained ULTRA checkpoints, more expressive alternatives remain unexplored. Additionally, as state-of-the-art methods often combine GNNs with recurrent modules, integrating ULTRA with RNN-based components is a promising direction for future work. Looking ahead, gathering large-scale, heterogeneous TKGs for pre-training, especially from less

Table 4. Results for temporal knowledge graph forecasting on GDELT in the multi-step setting with time-filter triples. TE stands for time encoding, FT for fine-tuned.

Model	GDELT			
	MRR	H@1	H@3	H@10
RE-GCN [13]	<u>19.64</u>	<u>12.47</u>	<u>20.85</u>	<u>33.62</u>
RE-Net [11]	**19.71**	**12.48**	**20.90**	**33.93**
CyGNet [20]	19.08	11.88	20.29	33.07
TLogic [14]	17.68	11.26	18.90	30.29
ULTRA (Zero-shot)	06.53	02.94	06.03	11.46
ULTRA (Fine-tuned)	16.42	08.96	17.31	31.35
ULTRA (TE + FT)	18.56	12.03	20.17	33.54

explored domains like socio-financial systems or biomedicine, may unlock new capabilities for temporal reasoning and enable the development of temporal graph foundation models.

Acknowledgments. This work has been partially funded by the Italian Ministry of University and Research (MUR) and the European Union – NextGenerationEU in the framework of the PRIN 2022 project "AWESOME: Analysis framework for WEb3 SOcial MEdia" – CUP: I53D23003680006; and by the National Center for Gene Therapy and Drugs Based on RNA Technology—MUR (Project no. CN 00000041) funded by NextGeneration EU program

References

1. Cai, B., Xiang, Y., Gao, L., Zhang, H., Li, Y., Li, J.: Temporal knowledge graph completion: A survey. In: IJCAI. pp. 6545–6553. ijcai.org (2023)
2. Cong, W., Zhang, S., Kang, J., Yuan, B., Wu, H., Zhou, X., Tong, H., Mahdavi, M.: Do we really need complicated model architectures for temporal networks? In: ICLR. OpenReview.net (2023)
3. Dileo, M., Minervini, P., Zignani, M., Gaito, S.: Enhancing neural link predictors for temporal knowledge graphs with temporal regularisers. In: ESANN (2025)
4. Fey, M., Kocijan, V., Lopez, F., Lenssen, J., Leskovec, J.: Kumorfm: A foundation model for in-context learning on relational data (2025), preprint, https://kumo.ai/research/kumo_relational_foundation_model.pdf
5. Galkin, M., Yuan, X., Mostafa, H., Tang, J., Zhu, Z.: Towards foundation models for knowledge graph reasoning. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=jVEoydFOl9
6. García-Durán, A., Usbeck, R., Paulheim, H.: Learning sequence encoders for temporal knowledge graph completion. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020), introduced ICEWS datasets (e.g., ICEWS14) for temporal KG tasks

7. Gastinger, J., Meilicke, C., Errica, F., Sztyler, T., Schülke, A., Stuckenschmidt, H.: History repeats itself: A baseline for temporal knowledge graph forecasting. In: IJCAI. pp. 4016–4024. ijcai.org (2024)
8. Gastinger, J., Sztyler, T., Sharma, L., Schuelke, A., Stuckenschmidt, H.: Comparing apples and oranges? on the evaluation of methods for temporal knowledge graph forecasting. In: Koutra, D., Plant, C., Gomez Rodriguez, M., Baralis, E., Bonchi, F. (eds.) Machine Learning and Knowledge Discovery in Databases: Research Track. pp. 533–549. Springer Nature Switzerland, Cham (2023)
9. Hu, F., et al.: Temporal knowledge graph reasoning with dynamic hypergraph transformer. In: LREC 2024 (2024), introduces WIKI (and YAGO) as temporal KG benchmarks
10. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Trans. Neural Networks Learn. Syst. **33**(2), 494–514 (2022)
11. Jin, W., Qu, M., Jin, X., Ren, X.: Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6669–6683. Association for Computational Linguistics, Online (Nov 2020)
12. Leetaru, K., Schrodt, P.A.: Gdelt: Global data on events, location and tone, 1979–2012. ISA Annual Convention (2013), introduced GDELT event database
13. Li, Z., Jin, X., Li, W., Guan, S., Guo, J., Shen, H., Wang, Y., Cheng, X.: Temporal knowledge graph reasoning based on evolutional representation learning. Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (2021)
14. Liu, Y., Ma, Y., Hildebrandt, M., Joblin, M., Tresp, V.: Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In: AAAI. pp. 4120–4127. AAAI Press (2022)
15. Messner, J., Abboud, R., Ceylan, İ.İ.: Temporal knowledge graph completion using box embeddings. In: AAAI. pp. 7779–7787. AAAI Press (2022)
16. Shamsi, K., Ngo, T.G.B., Shirzadkhani, R., Huang, S., Poursafaei, F., Azad, P., Rabbany, R., Coskunuzer, B., Rabusseau, G., Akcora, C.G.: Mint: Multi-network training for transfer learning on temporal graphs (2025), https://arxiv.org/abs/2406.10426
17. Sun, H., Zhong, J., Ma, Y., Han, Z., He, K.: Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In: EMNLP (1). pp. 8306–8319. Association for Computational Linguistics (2021)
18. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. International Conference on Learning Representations (2015)
19. Zhao, J., Zhu, Z., Galkin, M., Mostafa, H., Bronstein, M.M., Tang, J.: Fully-inductive node classification on arbitrary graphs. In: ICLR. OpenReview.net (2025)
20. Zhu, C., Chen, M., Fan, C., Cheng, G., Zhang, Y.: Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In: AAAI. pp. 4732–4740. AAAI Press (2021)
21. Zhu, Z., Zhang, Z., Xhonneux, L.P., Tang, J.: Neural bellman-ford networks: A general graph neural network framework for link prediction. Advances in Neural Information Processing Systems **34**, 29476–29490 (2021)