PREDICTING WHAT HAPPENS NEXT: NEXT-EVENT TRAINING AND EVALUATION FOR TEMPORAL KNOWLEDGE GRAPH EXTRAPOLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Temporal Knowledge Graph Extrapolation (TKGE) aims to predict future events from historical events. Existing TKGE methods typically operate in a **relation-specified** setting, where the relation is given and the task is to predict the missing entity. However, in many real-world applications, only a single subject entity is available, and the goal is to predict both the missing relation and object entity of a future event. We refer to this setting as **relation-unspecified** and define the task as **S**ubject-conditioned **R**elation and **O**bject **P**rediction (**SROP**). This mismatch motivates our work. To address it, we propose **N**ext-**E**vent e**X**trapolation **T**raining (**NEXT**), a model-agnostic method that addresses SROP by shifting the optimization objective from predicting the missing entity to predicting the missing relation and then the object entity. NEXT applies to existing TKGE models equipped with an entity-history encoder and a relation-conditioned entity decoder. For such models, NEXT adapts them to SROP by adding an MLP-based prediction head that predicts the missing relation, after which the original decoder predicts the object under that relation. NEXT further introduces a new relation-aware training objective that supplements the original entity-prediction loss with a relation prediction loss. Experiments on four datasets show that NEXT raises SROP performance on ICEWS14, ICEWS18, ICEWS0515, and GDELT from 0.08%, 0.08%, 0.20%, and 0.20% to 9.35%, 4.57%, 10.87%, and 1.21%, respectively. An ablation study under identical architectures and parameter count further shows that this improvement is attributable to the change in the optimization objective rather than to any modification of the model architecture or parameters. These experiments demonstrate that NEXT enables existing models to solve SROP without compromising their original entity prediction capability.

1 INTRODUCTION

A Temporal Knowledge Graph (TKG) is a sequence of knowledge graphs at successive timestamps, where each event is a quadruple (s, r, o, t) recording that subject s takes relation r on object o at time t. Based on this historical sequence, Temporal Knowledge Graph Extrapolation (TKGE) aims to predict future events. TKGE supports a wide range of applications, including economic forewarning He et al. (2024), social computing Higashiguchi et al. (2025), recommendation systems Cao et al. (2025), temporal question answering Xue et al. (2024), and clinical risk analysis Postiglione et al. (2024). In principle, a query may supply all but one element of a future event, and the missing element may be an entity, a relation, or a timestamp. More generally, TKGE can be viewed as completing partially observed future events, where one or more elements are missing. In practice, however, most existing methods focus on predicting the missing entity.

A long line of methods has advanced this task by improving how an entity's history is encoded. RE-GCN Li et al. (2021) evolves entity states over successive snapshots; TiRGN Li et al. fuses each entity's local neighborhood with global history; LogCL W. Chen et al. selects historical facts via contrastive learning; and AdaTKG Lee et al. (2026) keeps a per-entity memory updated at every interaction. These methods, however, share a common assumption: the relation is given in advance. Since this assumption does not hold in many real-world scenarios, where only the subject entity is given, these methods cannot be directly applied to such cases.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Real-world applications, however, typically start from a single entity and require the model to predict the relation and then the object entity. We refer to this situation as **relation-unspecified** and define the task as **S**ubject-conditioned **R**elation and **O**bject **P**rediction (**SROP**). A few recent studies focus on relation prediction, which, given the subject and object entities, predicts the missing relation. HTKGP Chen et al. (2026) formalizes TKG relation prediction and models the hierarchy among relations in hyperbolic space; RME Zong et al. (2025) infers unknown or emerging relations from historical facts via relational multi-path reasoning; R-CTGCN Mirzaei & Kobti (2024) exploits timestamps and confidence for implicit relation prediction on temporal graphs. Although these methods also predict missing relations, they rely on the states of both the subject and the object entity. They therefore address relation prediction with both endpoints given, whereas SROP provides a single subject entity. SROP thus remains an unaddressed task.

To address this gap, we propose **N**ext-**E**vent e**X**trapolation **T**raining (**NEXT**), a model-agnostic method that addresses SROP by shifting the optimization objective from predicting the missing entity to predicting the missing relation and then the object entity. NEXT applies to any existing TKGE model that provides two components: an encoder that produces a per-entity state from the entity's history, and a decoder that scores object entities given a relation. On such a model, NEXT adapts it to SROP by adding an MLP-based prediction head that predicts the missing relation, after which the original decoder predicts the object under that relation. NEXT further introduces a new relation-aware training objective that supplements the original entity-prediction loss with a relation prediction loss. Our contributions are:

- We identify a new problem: predicting the missing relation and the object entity in a future event when the query provides only a subject entity. This situation arises in real-world applications yet remains unaddressed by existing methods. We formalize this task as SROP.
- We propose NEXT, a model-agnostic method that enables existing models to solve SROP without compromising their original entity prediction capability.
- Experiments on four datasets show that NEXT raises SROP performance on ICEWS14, ICEWS18, ICEWS0515, and GDELT from 0.08%, 0.08%, 0.20%, and 0.20% to 9.35%, 4.57%, 10.87%, and 1.21%, respectively. An ablation study under identical architectures and parameter count further shows that this improvement is attributable to the change in the optimization objective rather than to any modification of the model architecture or parameters.

2 Related Work

This work lies at the intersection of temporal knowledge graph extrapolation and relation prediction. We group the most relevant studies into three lines and then position SROP relative to them.

2.1 Temporal Knowledge Graph Extrapolation under the relation-specified Setting

Most TKGE methods assume the relation is given and predict the missing entity. A long line of work improves this task by designing better encoders of an entity's history. RE-GCN Li et al. (2021) evolves entity states across successive snapshots with a recurrent architecture; TiRGN Li et al. fuses each entity's local neighborhood with global history via time-guided gates; LogCL W. Chen et al. selects informative historical facts with local-global history-aware contrastive learning; and AdaTKG Lee et al. (2026) keeps a per-entity memory updated at every interaction. Despite these advances, all of them operate under the relation-specified setting, where the query supplies the relation in advance. They therefore cannot be applied to SROP, where the relation itself is missing.

2.2 Relation Prediction in Temporal Graphs

A second, much smaller line treats the relation itself as the prediction target. HTKGP Chen et al. (2026) formalizes TKG relation prediction. It learns relation features in hyperbolic space to capture the hierarchy among relations and to fuse temporal dynamics with static attributes. RME Zong et al. (2025) infers unknown or emerging relations from historical facts through multi-path reasoning. R-CTGCN Mirzaei & Kobti (2024) exploit timestamps and confidence scores in a temporal graph

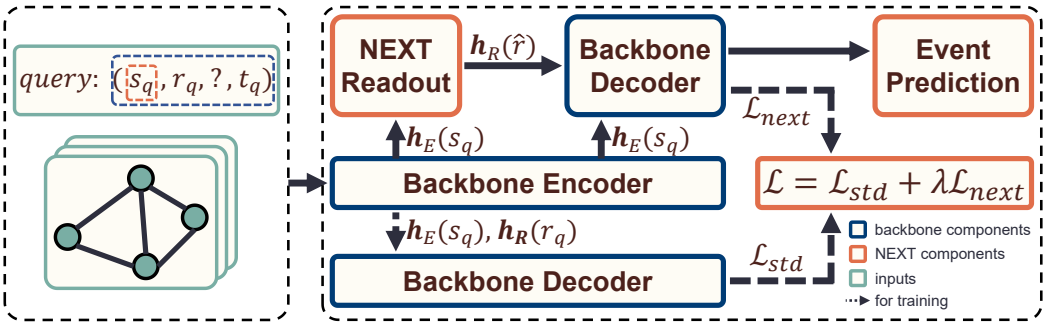


Figure 1: Overview of NEXT on an existing TKGE model. Blue boxes are the components the model already provides, and the two decoder boxes denote the same shared decoder; orange boxes are added by NEXT; teal elements are the inputs, the history and the query $(s_q, r_q, ?, t_q)$, of which the subject s_q is the only element the SROP path receives; dashed arrows are used during training only. Top path (the relation-unspecified setting, SROP): the MLP head predicts the missing relation $\hat{r}$ from the subject state $h_E(s_q)$, and the original decoder then predicts the object under the predicted relation $h_R(\hat{r})$, giving the SROP answer $(\hat{r}, \hat{o})$. Bottom path (the relation-specified setting): the unchanged decoder predicts the missing entity of the query from $h_E(s_q)$ and the query relation $h_R(r_q)$. The relation-aware objective supplements the original entity-prediction loss $\mathcal{L}_{\text{std}}$ with the relation prediction loss $\mathcal{L}_{\text{next}}$, that is, $\mathcal{L} = \mathcal{L}_{\text{std}} + \lambda\,\mathcal{L}_{\text{next}}$.

convolutional network to predict implicit relations between entities. These methods, however, require both endpoints of the query: they score the relation of a given subject-object pair or of a fully observed local context. In SROP, only the subject entity is given, so they cannot be applied to it.

2.3 Joint Prediction of Multiple Event Elements

A third line relaxes the assumption that only one element is missing. EvoKG Park et al. (2022) jointly predicts the link and its occurrence time by modeling event time and network structure together. GAttNHP Tian et al. (2026) models each (subject, relation) chain as a neural Hawkes process and jointly predicts the object entity and its time. GenTKG Liao et al. (2023) recasts extrapolation as conditional text generation with large language models. It produces object candidates never seen in history. These methods broaden prediction beyond a single missing entity, but none of them predicts both the relation and the object entity from a single subject entity.

2.4 Position of This Work

In summary, existing methods either assume the relation is given, require both endpoints of the query, or predict elements other than the relation and the object. SROP occupies this gap: given only a subject entity, it requires predicting both the relation and the object entity of its next event. NEXT addresses this gap by shifting the optimization objective of an existing TKGE model, rather than by changing the backbone architecture.

3 Methodology

This section formalizes the two settings of TKG extrapolation and the task studied in this paper, and then presents NEXT. Section 3.1 gives the notation and the task, Section 3.2 describes the method, and Section 3.3 gives the relation-aware training objective. Figure 1 gives the overview.

3.1 Problem Formulation

A temporal knowledge graph (TKG) is a sequence of knowledge graphs at successive timestamps, where each event is a quadruple (s, r, o, t) recording that subject s takes relation r on object o at time t. Let $\mathcal{E}$ denote the set of entities and $\mathcal{R}$ the set of relations, and let $\mathcal{H}_t$ denote the events before

time t. TKGE predicts future events from this historical sequence. A query may supply all but one element of a future event, and the missing element may be an entity, a relation, or a timestamp. We distinguish two settings by what the query supplies.

The relation-specified setting. The query supplies the relation, and the task is to predict the missing entity. A query $(s, r, ?, t)$ supplies the subject s, the relation r, and the time t, and asks for the object o. Existing TKGE methods answer such a query with two components. An encoder produces a per-entity state from the entity's history,

$$h_s = \mathrm{Enc}(s, \mathcal{H}_t), \tag{1}$$

and a decoder scores object entities given a relation,

$$\mathrm{score}(o \mid s, r) = \mathrm{Dec}\big(h_s, h_R(r)\big), \tag{2}$$

where $h_R(r)$ is the representation of relation r that the decoder conditions on. Training minimizes the entity-prediction loss over these scores. This is the setting in which existing methods are trained and evaluated (Section 2).

The relation-unspecified setting. In many real-world applications, only a single subject entity is available, and the goal is to predict both the missing relation and object entity of a future event. We refer to this setting as relation-unspecified and define the task as follows.

Definition (Subject-conditioned Relation and Object Prediction, SROP). Given a subject entity s and its history $\mathcal{H}_t$, SROP asks for the missing relation and object entity of the next event in which s participates as subject.

SROP provides a single subject entity. It therefore differs from relation prediction, which is given both endpoints of the event and predicts the relation that connects them (Section 2). Neither component of a relation-specified model answers SROP on its own: the encoder produces the state h_s for entity prediction, and the decoder requires a relation, which the query no longer supplies.

3.2 NEXT

Figure 1 overviews NEXT. NEXT addresses SROP by shifting the optimization objective from predicting the missing entity to predicting the missing relation and then the object entity. It is model-agnostic: it applies to any existing TKGE model that provides the two components above, an encoder that produces a per-entity state from the entity's history, and a decoder that scores object entities given a relation. NEXT adapts such a model to SROP by adding an MLP-based prediction head that predicts the missing relation, after which the original decoder predicts the object under that relation.

SROP asks for the missing relation and the missing object entity together, and NEXT predicts them in two steps. It factorizes the joint prediction as

$$p(r, o \mid s) = p_r(r \mid s)\; p_o(o \mid s, r), \tag{3}$$

where the conditioning on $\mathcal{H}_t$ is omitted for brevity, and assigns the two factors to the head and to the decoder. The head, the only added component, is an MLP that maps the subject state to a distribution over the relations:

$$p_r(r \mid s) = \mathrm{softmax}\big(\mathrm{MLP}(h_s)\big). \tag{4}$$

The object factor is the original decoder, conditioned on a relation:

$$p_o(o \mid s, r) = \mathrm{softmax}\big(\mathrm{Dec}(h_s, h_R(r))\big). \tag{5}$$

The predicted relation is $\hat{r} = \arg\max_{r \in \mathcal{R}} p_r(r \mid s)$, and the SROP answer is the pair

$$\big(\hat{r},\ \hat{o}\big) = \Big(\arg\max_{r \in \mathcal{R}} p_r(r \mid s),\ \arg\max_{o \in \mathcal{E}} p_o(o \mid s, \hat{r})\Big). \tag{6}$$

The head predicts the missing relation, and the original decoder predicts the object under that relation. In this path the object is scored under the predicted relation rather than a given one, which is the only difference from the relation-specified path of Eq. equation 2. The two maximizations in

Eq. equation 6 are carried out greedily and in sequence: the head first selects its most probable relation, and the decoder then ranks the objects under that relation. This is a greedy two-step decoding, not a joint maximization of Eq. equation 3 over all relation-object pairs, and we do not claim it to be. We adopt it for two reasons. First, a joint search would require scoring every entity under every relation, multiplying the cost of a single object ranking by $|\mathcal{R}|$ on vocabularies in which entities outnumber relations 31 to 90 times (Section 4.1). Second, Table 2 shows that the relation step is the bottleneck of the SROP answer, which limits the gain a joint search could offer over the greedy rule. Greedy decoding is therefore an intentional efficiency design choice of NEXT, acknowledged among the limitations in Section 5.

The relation-specified setting is preserved. A query $(s, r, ?, t)$ is answered by the same decoder conditioned on the relation supplied by the query, as in Eq. equation 2, and neither the components nor the evaluation of the original task are modified.

3.3 Relation-Aware Training Objective

NEXT introduces a relation-aware training objective that supplements the original entity-prediction loss with a relation prediction loss. We write $\mathcal{L}_{\text{std}}$ for the original entity-prediction loss, the loss the model already minimizes, and $\mathcal{L}_{\text{next}}$ for the added loss, so that the model is trained on

$$\mathcal{L} = \mathcal{L}_{\text{std}} + \lambda \mathcal{L}_{\text{next}}, \tag{7}$$

where λ is a single weight. The added loss is a cross-entropy of the head's distribution against the relation of the entity's next event,

$$\mathcal{L}_{\text{next}} = \text{CE}\big(p_r(\cdot \mid s),\ r\big), \tag{8}$$

computed on the same events as the original loss. It is the only signal that trains the head, and the decoder continues to be trained by the entity-prediction loss under the relations the queries supply.

Because NEXT adds the head without changing the encoder or the decoder, the only added parameters are those of the MLP. Because $\mathcal{L}_{\text{std}}$ is the loss the model already minimizes, the relation-specified setting is trained exactly as before. Because the head is present in both cases and only its training signal differs, the experiments use the case $\lambda = 0$ to attribute the gain on SROP to the change in the optimization objective rather than to a modification of the model architecture or parameters (Section 4).

4 Experiments

The claims of Section 1 raise three questions, which organize this section.

- **Q1:** What happens to the original entity-prediction performance in the relation-specified setting when the relation prediction loss is added?
- **Q2:** How much does NEXT improve SROP performance on the four datasets and across the four models?
- **Q3:** What does the improvement on SROP stem from, the change in the optimization objective or a modification of the model architecture or parameters?

Section 4.1 describes the setup of these questions, and Sections 4.2, 4.3, and 4.4 answer them in turn.

4.1 Experimental Setup

Datasets and splits. We use four benchmarks, ICEWS14, ICEWS18, ICEWS0515, and GDELT Boschee et al.; gde, under the chronological splits used in prior work Jin et al.: the events are partitioned by time, so that all training events precede the validation events, which precede the test events, and every prediction is made strictly from past events.

Models. NEXT is instantiated on four existing TKGE models, RE-GCN Li et al. (2021), TiRGN Li et al., LogCL W. Chen et al., and AdaTKG Lee et al. (2026), each of which provides the two components that NEXT requires, an entity-history encoder and a relation-conditioned entity decoder:

Table 1: Q1, the relation-specified setting. Filtered MRR and Hits@k of the four models, trained as the backbone on the original entity-prediction loss and as the backbone w/NEXT. The Δ column per model block reports the change w/NEXT $-$ backbone, $\uparrow x$ for an increase and $\downarrow x$ for a decrease.

Dataset	Metric	RE-GCN			TiRGN			LogCL			AdaTKG		
		backbone	w/NEXT	Δ	backbone	w/NEXT	Δ	backbone	w/NEXT	Δ	backbone	w/NEXT	Δ
ICEWS14	MRR	41.50	41.56	↑0.06	44.37	44.44	↑0.07	48.87	48.35	↓0.52	45.60	45.11	↓0.49
	Hits@1	30.86	30.83	↓0.03	34.03	33.87	↓0.16	37.76	37.52	↓0.24	32.47	32.18	↓0.29
	Hits@3	46.60	46.48	↓0.12	49.14	49.78	↑0.64	54.71	55.18	↑0.47	49.09	48.27	↓0.82
	Hits@10	62.47	62.82	↑0.35	64.41	64.66	↑0.25	70.26	70.57	↑0.31	67.32	66.38	↓0.94
ICEWS18	MRR	30.55	30.56	↑0.01	33.68	33.13	↓0.55	35.76	35.70	↓0.06	31.25	30.62	↓0.63
	Hits@1	20.00	19.92	↓0.08	23.04	22.60	↓0.44	24.53	24.16	↓0.37	20.14	19.22	↓0.92
	Hits@3	34.73	34.65	↓0.08	38.23	37.45	↓0.78	40.32	39.87	↓0.45	35.18	35.23	↑0.05
	Hits@10	51.46	51.33	↓0.13	54.37	53.90	↓0.47	57.74	57.92	↑0.18	55.74	55.25	↓0.49
ICEWS0515	MRR	46.41	46.46	↑0.05	49.25	48.66	↓0.59	57.04	56.91	↓0.13	47.02	46.38	↓0.64
	Hits@1	35.17	34.98	↓0.19	38.40	37.82	↓0.58	46.07	45.63	↓0.44	35.66	34.98	↓0.68
	Hits@3	52.76	52.68	↓0.08	55.24	54.64	↓0.60	63.72	63.45	↓0.27	56.11	55.20	↓0.91
	Hits@10	67.64	67.50	↓0.14	69.80	69.32	↓0.48	77.87	77.66	↓0.21	70.24	70.41	↑0.17
GDELT	MRR	19.31	19.26	↓0.05	21.42	21.32	↓0.10	23.75	23.14	↓0.61	21.37	21.12	↓0.25
	Hits@1	11.99	12.08	↑0.09	13.38	13.28	↓0.10	14.64	14.69	↑0.05	11.12	10.79	↓0.33
	Hits@3	20.61	20.70	↑0.09	23.05	22.96	↓0.09	25.60	24.81	↓0.79	22.33	22.04	↓0.29
	Hits@10	33.59	33.58	↓0.01	37.39	37.20	↓0.19	42.33	41.86	↓0.47	43.06	42.51	↓0.55

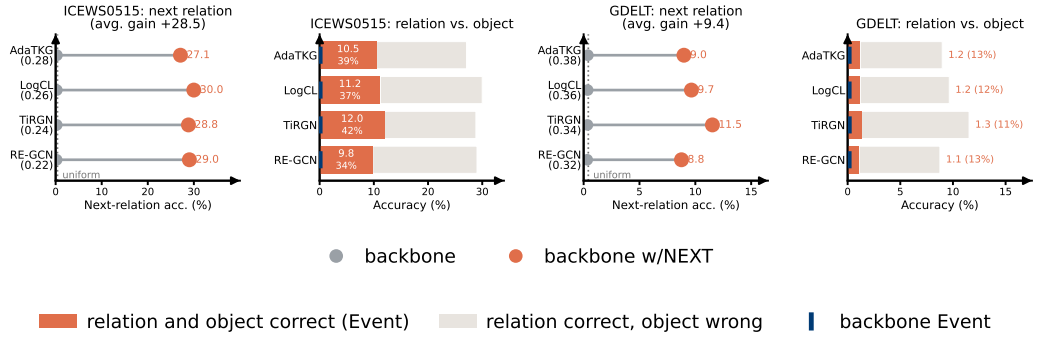
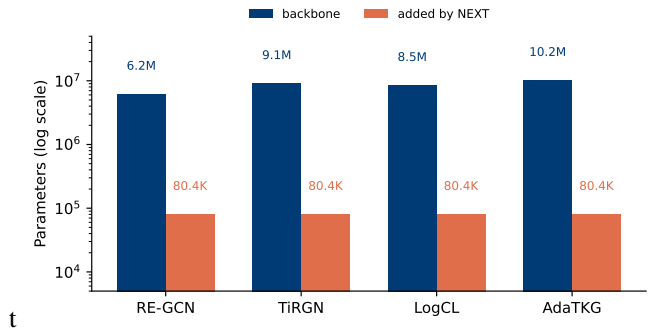


Figure 2: The relation-unspecified setting on ICEWS0515 (panels 1–2) and GDELT (panels 3–4); rows are the four backbones and the legend below the row applies to all four panels. *Panels 1 and 3*: next-relation accuracy; gray is the backbone, orange the backbone w/NEXT, the dotted line the uniform guess, and the bracket under each backbone name its own accuracy. *Panels 2 and 4*: the decomposition of the next-event accuracy, where the bar is the next-relation accuracy, its orange part the next-event accuracy, its light part the events whose relation is right but whose object is not, and the blue tick the next-event accuracy of the backbone; the percentage is the object accuracy among the correctly predicted relations.

RE-GCN evolves entity states over successive snapshots; TiRGN fuses each entity's local neighborhood with global history; LogCL selects historical facts via contrastive learning; and AdaTKG keeps a per-entity memory updated at every interaction. On every model, NEXT adds only the MLP-based prediction head of Section 3.2 and leaves the model itself unchanged.

Protocols and metrics. The relation-specified setting keeps the standard evaluation unchanged: queries of the form $(s, r, ?, t)$ are answered by the decoder under the query relation, and performance is the filtered MRR and Hits@1, Hits@3, and Hits@10. The relation-unspecified setting follows the SROP definition of Section 3.1: for every test event (s, r, o, t) in the subject role, the model receives only the subject s and the history $\mathcal{H}_t$, and outputs the pair $(\hat{r}, \hat{o})$ as in Eq. equation 6 of Section 3. At each test timestamp, $\mathcal{H}_t$ consists of the training and validation events and the test events before t. We report three quantities. Next-relation accuracy is the fraction of events with $\hat{r} = r$, and we accompany it with the MRR, Hits@3, and Hits@10 of the relation ranking produced by $p_r(r \mid s)$. The object ranking is evaluated under the predicted relation: the decoder scores all entities under $h_R(\hat{r})$, and we report the MRR, Hits@1, Hits@3, and Hits@10 of the resulting ranking, filtered with respect to $(s, \hat{r})$. Next-event accuracy is the fraction of events with $(\hat{r}, \hat{o}) = (r, o)$, the headline metric of the relation-unspecified setting.



t

Figure 3: Parameters of the four backbones and the parameters that NEXT adds, on a logarithmic scale.

Training. For every model we train two configurations that share the same architecture and the same parameter set: the backbone, trained on the original entity-prediction loss alone, which is the case $\lambda = 0$ of Eq. equation 7, and the backbone w/NEXT, trained on Eq. equation 7 with $\lambda > 0$. The relation head is present in both configurations; in the backbone it receives no training signal. The two configurations of a model are trained with the same budget and selected by the same rule, the checkpoint with the best validation filtered MRR, so that the only difference between them is the optimization objective. The head adds only the parameters of its MLP, 80.4K parameters, 0.8% to 1.3% of each model, and the two configurations of every model have identical parameter counts.

Implementation. We train with Adam at learning rate 10^{-3}, embedding dimension 200, and dropout 0.2; the history window and the remaining hyperparameters follow each model's original setting.

Efficiency. NEXT adds only the parameters of the MLP head. The head scores the relation vocabulary, while the decoder scores the entity vocabulary, which is 31 to 90 times larger on the four datasets, so the computation added by the head is a small fraction of one pass of the decoder. Figure 3 reports the parameters of the four backbones and of the head on a logarithmic scale, where the head is about two orders of magnitude smaller than the backbone, and training time increases by at most 0.4%.

4.2 Q1: What Happens to the Original Entity-Prediction Performance?

Table 1 reports the filtered MRR and Hits@k of the four models, trained as the backbone and as the backbone w/NEXT, with one Δ column per model. The two configurations agree closely: 47 of the 64 cells move by less than 0.5 points and only 15 of them increase. The largest deviation anywhere in the table is 0.94 Hits@10 points (AdaTKG on ICEWS14), and the largest MRR move is 0.64 (AdaTKG on ICEWS0515). The answer to Q1 is that the added relation prediction loss maintains the original entity-prediction performance in the relation-specified setting.

4.3 Q2: How Much Does NEXT Improve SROP?

Figure 2 reports SROP on ICEWS0515 and GDELT: for each dataset, the first of its two panels gives the next-relation accuracy of the four backbones and the second decomposes the next-event accuracy that follows from it into the relation step (the bar) and the object step (its orange part). Figure 4 summarizes the next-event accuracy of all sixteen backbone-by-dataset settings, including the two datasets not shown in Figure 2. Four observations follow. First, the backbone answers SROP at chance: on every model and dataset its next-relation accuracy is at most 0.41%, the level of a uniform guess over the relation vocabulary, and its next-event accuracy is at most 0.21%. A model trained as usual, therefore, cannot answer the query that supplies only the subject, which is the numerical form of the claim that existing methods cannot be directly applied to SROP. Second, the backbone w/NEXT changes this output. Averaged over the four models, the next-relation accuracy improves from 0.30, 0.34, 0.25, and 0.35 to 29.0, 26.6, 28.7, and 9.7 on ICEWS14, ICEWS18, ICEWS0515, and GDELT, and the next-event accuracy improves from 0.08, 0.08, 0.20, and 0.20

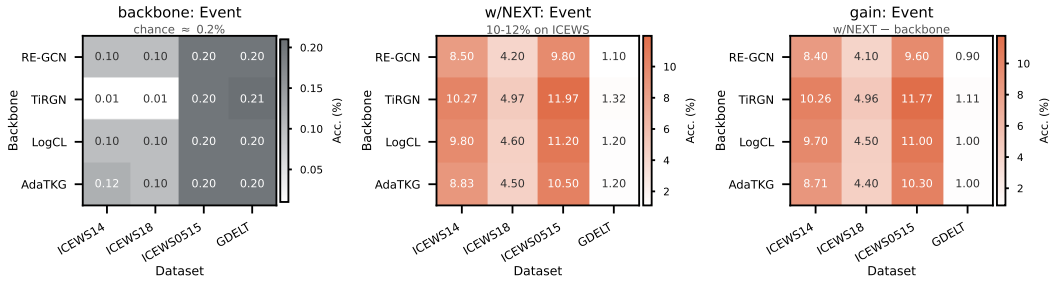


Figure 4: Next-event accuracy of the sixteen backbone-by-dataset settings, and its gain: gray is the backbone trained on the original entity-prediction loss, orange the backbone w/NEXT, and the right matrix the gain of each; rows are backbones and columns datasets.

Backbone	Dataset	backbone			backbone w/NEXT		
		Rel Hits@1	Obj acc.	Event	Rel Hits@1	Obj acc.	Event
RE-GCN	ICEWS14	0.32	30.86	0.10	27.91	30.46	8.50
	ICEWS18	0.31	20.00	0.10	26.92	15.60	4.20
	ICEWS0515	0.22	35.17	0.20	29.03	33.76	9.80
	GDELT	0.32	11.99	0.20	8.76	12.56	1.10
TiRGN	ICEWS14	0.12	34.03	0.01	29.23	35.14	10.27
	ICEWS18	0.33	23.04	0.01	26.78	18.56	4.97
	ICEWS0515	0.24	38.40	0.20	28.83	41.52	11.97
	GDELT	0.34	13.38	0.21	11.53	11.45	1.32
LogCL	ICEWS14	0.34	37.76	0.10	29.85	32.83	9.80
	ICEWS18	0.35	24.53	0.10	27.54	16.70	4.60
	ICEWS0515	0.26	46.07	0.20	29.99	37.35	11.20
	GDELT	0.36	14.64	0.20	9.65	12.44	1.20
AdaTKG	ICEWS14	0.41	32.47	0.12	28.84	30.62	8.83
	ICEWS18	0.37	20.14	0.10	25.04	17.97	4.50
	ICEWS0515	0.28	35.66	0.20	27.09	38.76	10.50
	GDELT	0.38	11.12	0.20	8.98	13.36	1.20

Table 2: Decomposition of the next-event accuracy into its two steps, for the backbone and for the backbone w/NEXT. For the backbone, Obj acc. is the filtered Hits@1 under the true relation (Table 1); for the backbone w/NEXT, it is the object accuracy on the queries whose predicted relation is correct, so Rel Hits@1 $\times$ Obj acc. equals Event up to rounding. The Rel Hits@1 and Event columns are the two metrics defined in Section 4.1; this table adds the two Obj acc. columns.

to 9.35, 4.57, 10.87, and 1.21, by 9.3, 4.5, 10.7, and 1.0 percentage points, respectively; Figure 4 shows the same gains in next-event accuracy for all sixteen settings. Third, the object ranking of the SROP answer is produced by the same decoder that serves the relation-specified setting, so the object side inherits the entity-prediction ability that Table 1 shows to be maintained.

Fourth, the decomposition follows the two steps of Eq. equation 6. Panels 2 and 4 of Figure 2 show it for ICEWS0515 and GDELT, and Table 2 gives both factors of all sixteen settings. For the backbone, the relation factor is at chance, 0.12 to 0.41, so its next-event accuracy collapses to 0.01 to 0.21 even though the object step under the true relation is the filtered Hits@1 of Table 1, 11.1 to 46.1. For the backbone w/NEXT the object accuracy is measured on the queries whose predicted relation is correct, so the product of the two factors is the next-event accuracy up to rounding, and the object step stays between 11.4 (GDELT, TiRGN) and 41.5 (ICEWS0515, TiRGN): it inherits the entity-prediction ability that Table 1 reports for the relation-specified decoder, and the next-relation accuracy alone is the bottleneck. That is exactly the quantity the added relation prediction loss trains.

NEXT thus improves performance on SROP on all four datasets and across all four models.

4.4 Q3: What Does the Improvement Stem From?

The objective, not the architecture or parameters. The backbone columns of Table 2 are the experiment required by Q3: the backbone and the backbone w/NEXT share the same architecture, including the same relation head, and the same parameter set, and the only difference between them is the objective of Eq. equation 7. At $\lambda = 0$ the head receives no training signal—its parameters

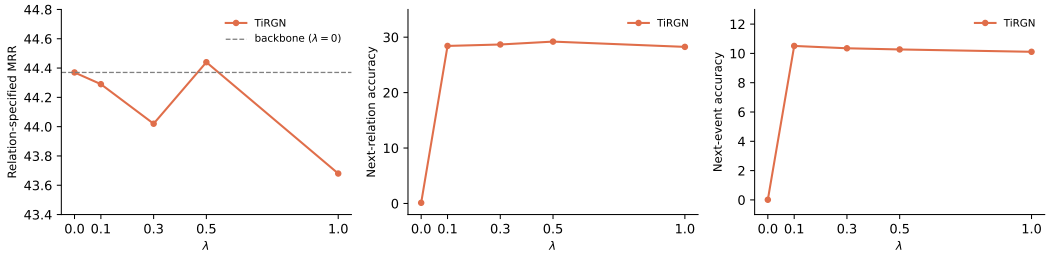


Figure 5: The λ sweep n ICEWS14 for TiRGN: (a) the relation-specified filtered MRR, with the backbone's $\lambda = 0$ score as the dashed reference; (b) the next-relation accuracy; (c) the next-event accuracy.

never receive gradients and it does not influence the encoder or the decoder—and its predictions remain essentially random, as the chance-level backbone columns of that table confirm. In this controlled comparison, the improvement on SROP is therefore attributable to the change in the optimization objective rather than to any modification of the model architecture or parameters. Because the reported numbers come from single runs, we state this attribution within the present comparison and treat the training objective as the decisive contributor, not the only conceivable one.

The weight λ. λ is the only hyperparameter that NEXT introduces. We sweep λ over $\{0, 0.1, 0.3, 0.5, 1.0\}$ on ICEWS14 and report the relation-specified MRR, the next-relation accuracy, and the next-event accuracy in Figure 5. The sweep is run on TiRGN, the backbone with the largest next-event gain on every dataset, because its purpose is to verify that the gain comes from the presence of the relation prediction loss rather than from a finely tuned weight.

Two observations follow. First, the relation-specified score stays at the level of the backbone for every λ, so the original entity prediction capability does not depend on the weight of the added objective. Second, every $\lambda > 0$ yields the SROP improvement, so the gain comes from the presence of the relation prediction loss rather than from a finely tuned weight. We therefore use $\lambda = 0.5$ for all backbones and datasets.

Taken together, the three answers show that NEXT enables existing models to solve SROP without compromising their original entity prediction capability.

5 Conclusion

In this paper, we identified a new problem in temporal knowledge graph extrapolation: predicting both the missing relation and object entity of a future event when only a single subject entity is given. We formalized this task as Subject-conditioned Relation and Object Prediction (SROP) and showed that existing TKGE methods, which assume a relation-specified setting, cannot directly address it. To bridge this gap, we proposed NEXT, a model-agnostic method that adapts existing TKGE models to SROP by shifting the optimization objective from predicting the missing entity to predicting the missing relation and then the object entity. NEXT applies to models equipped with an entity-history encoder and a relation-conditioned entity decoder, adding only an MLP-based prediction head and a relation-aware training objective that supplements the original entity-prediction loss with a relation prediction loss. Experiments on four datasets show that NEXT raises SROP performance on ICEWS14, ICEWS18, ICEWS0515, and GDELT from 0.08%, 0.08%, 0.20%, and 0.20% to 9.35%, 4.57%, 10.87%, and 1.21%, respectively. An ablation study under identical architectures and parameter count further shows that this improvement is attributable to the change in the optimization objective rather than to any modification of the model architecture or parameters. These experiments demonstrate that NEXT enables existing models to solve SROP without compromising their original entity prediction capability. A limitation of NEXT is the greedy two-step decoding, which does not in general find the most probable relation-object pair under the factorization of Eq. equation 3; we adopt it for the efficiency reasons discussed in Section 3.2, and Table 2 indicates that the relation step, which the greedy rule optimizes exactly, is the main bottleneck of the SROP answer.

AI USE STATEMENT

Generative AI tools were used to provide feedback on the experimental design and the alignment between claims and evidence, and to assist with method implementation, research-code development, statistical result interpretation, figure and artifact preparation, and manuscript drafting and editing. The authors manually verified the AI-assisted code and analyses against the source code and the recorded training logs, checked every figure and numerical summary against the generated artifacts, and reviewed and edited the full manuscript. The authors take responsibility for the final content of this work.

REFERENCES

GDELT: The global database of events, language, and tone. 52(1):52–0044–52–0044. ISSN 0009-4978, 1523-8253. doi: 10.5860/CHOICE.52-0044. URL http://choicereviews.org/review/10.5860/CHOICE.52-0044.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. ICEWS coded event data. URL https://doi.org/10.7910/DVN/28075. Section: 2015-03-27 14:44:26.045.

Yang Cao, Shuo Shang, Jun Wang, and Wei Zhang. Explainable session-based recommendation via path reasoning. *IEEE Trans. Knowl. Data Eng.*, 37(1):278–290, 2025. doi: 10.1109/TKDE.2024.3486326. URL https://doi.org/10.1109/TKDE.2024.3486326.

Jianrui Chen, Peijie Wang, Maoguo Gong, and Xuehui Zhao. Hyperbolic-based feature learning for temporal knowledge graph relation prediction. *ACM Transactions on Knowledge Discovery from Data*, 20(2):1–23, 2026.

Hui He, Qi Zhang, Shoujin Wang, Kun Yi, Zhendong Niu, and Longbing Cao. Learning informative representation for fairness-aware multivariate time-series forecasting: A group-based perspective. *IEEE Transactions on Knowledge and Data Engineering*, 36(6):2504–2516, 2024. doi: 10.1109/TKDE.2023.3323956.

Shingo Higashiguchi, Yasuko Matsubara, Koki Kawabata, Taichi Murayama, and Yasushi Sakurai. D-tracker: Modeling interest diffusion in social activity tensor data streams. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 460–471, 2025.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6669–6683. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.541. URL https://aclanthology.org/2020.emnlp-main.541/.

Seunghan Lee, Jun Seo, Jaehoon Lee, Sungdong Yoo, Minjae Kim, Tae Yoon Lim, Dongwan Kang, Hwanil Choi, SoonYoung Lee, and Wonbin Ahn. Adatkg: Adaptive memory for temporal knowledge graph reasoning. *CoRR*, abs/2605.07121, 2026. doi: 10.48550/ARXIV.2605.07121. URL https://doi.org/10.48550/arXiv.2605.07121.

Yujia Li, Shiliang Sun, and Jing Zhao. TiRGN: Time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 2152–2158. International Joint Conferences on Artificial Intelligence Organization. doi: 10.24963/ijcai.2022/299. URL https://doi.org/10.24963/ijcai.2022/299.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal knowledge graph reasoning based on evolutional representation learning. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (eds.), *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pp. 408–417. ACM, 2021. doi: 10.1145/3404835.3462963. URL https://doi.org/10.1145/3404835.3462963.

Ruotong Liao, Xu Jia, Yunpu Ma, and Volker Tresp. Gentkg: Generative forecasting on temporal knowledge graph. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*, 2023.

Lida Mirzaei and Ziad Kobti. Temporal graph convolutional network for implicit relation prediction: Leveraging timestamps and confidence. *Procedia Computer Science*, 246:940–949, 2024.

Namyong Park, Fuchen Liu, Purvanshi Mehta, Dana Cristofor, Christos Faloutsos, and Yuxiao Dong. Evokg: Jointly modeling event time and network structure for reasoning over temporal knowledge graphs. In K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang (eds.), *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pp. 794–803. ACM, 2022. doi: 10.1145/3488560.3498451. URL https://doi.org/10.1145/3488560.3498451.

Marco Postiglione, Daniel Bean, Zeljko Kraljevic, Richard J. B. Dobson, and Vincenzo Moscato. Predicting future disorders via temporal knowledge graphs and medical ontologies. *IEEE J. Biomed. Health Informatics*, 28(7):4238–4248, 2024. doi: 10.1109/JBHI.2024.3390419. URL https://doi.org/10.1109/JBHI.2024.3390419.

Xiangni Tian, Kaixian Yu, Runpeng Dai, Niansheng Tang, and Hongtu Zhu. Gattnhp: Group attention neural hawkes process for extrapolation reasoning in temporal knowledge graphs. *arXiv preprint arXiv:2607.14733*, 2026.

W. Chen, H. Wan, Y. Wu, S. Zhao, J. Cheng, Y. Li, and Y. Lin. Local-global history-aware contrastive learning for temporal knowledge graph reasoning. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 733–746. ISBN 2375-026X. doi: 10.1109/ICDE60146.2024.00062. Journal Abbreviation: 2024 IEEE 40th International Conference on Data Engineering (ICDE).

Chao Xue, Di Liang, Pengfei Wang, and Jing Zhang. Question calibration and multi-hop modeling for temporal question answering. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 19332–19340. AAAI Press, 2024. doi: 10.1609/AAAI.V38I17.29903. URL https://doi.org/10.1609/aaai.v38i17.29903.

Linlin Zong, Chi Ma, Jiahui Zhou, Xinyue Liu, Wenxin Liang, Xianchao Zhang, and Bo Xu. Relational multi-path enhancement for extrapolative relation reasoning in temporal knowledge graph. In Meeyoung Cha, Chanyoung Park, Noseong Park, Carl Yang, Senjuti Basu Roy, Jessie Li, Jaap Kamps, Kijung Shin, Bryan Hooi, and Lifang He (eds.), *Proceedings of the 34th ACM International Conference on Information and Knowledge Management, CIKM 2025, Seoul, Republic of Korea, November 10-14, 2025*, pp. 4592–4601. ACM, 2025. doi: 10.1145/3746252.3761033. URL https://doi.org/10.1145/3746252.3761033.