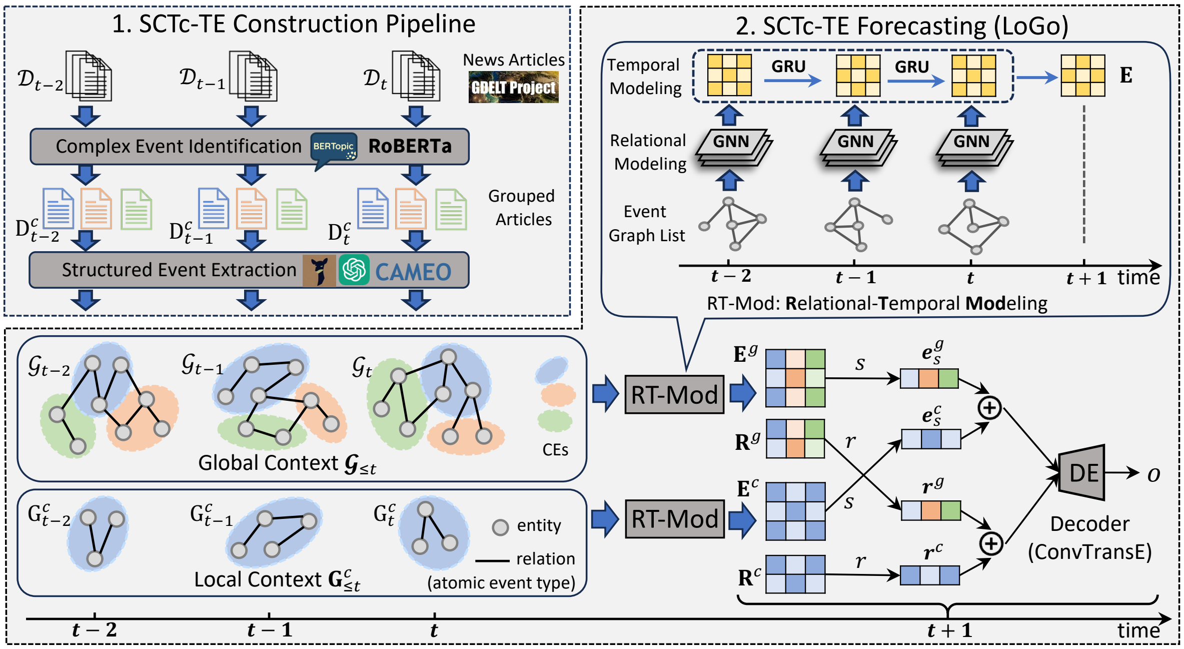


Figure 4: The overall framework of this work consists of two parts: 1) the SCTc-TE construction pipeline (the top left part), and 2) the SCTc-TE forecasting model LoGo (the right and bottom part).

Table 6: The overall performance of our LoGo and baselines. "%Improv." denotes the relative improvement.

Dataset	Metric	Static KG Methods				TKG Methods				Ours	
		DistMult	ConvE	ConvTransE	RGCN	RE-NET	RE-GCN	HisMatch	CMF	**LoGo**	%Improv.
GDELT-TE	MRR	0.1148	0.1140	0.1204	0.1277	0.1179	0.1403	0.1641	0.1387	**0.2533**	54.33
	HIT@1	0.0368	0.0387	0.0422	0.0503	0.0338	0.0507	0.0716	0.0499	**0.1344**	87.62
	HIT@3	0.1109	0.1107	0.1191	0.1285	0.1191	0.1379	0.1710	0.1397	**0.2899**	69.51
	HIT@10	0.2843	0.2734	0.2847	0.2874	0.3006	0.3445	0.3652	0.3345	**0.5053**	38.37
MidEast-TE	MRR	0.2227	0.2276	0.2358	0.2744	0.2504	0.2989	0.3354	0.2857	**0.4978**	48.44
	HIT@1	0.1177	0.1172	0.1283	0.1795	0.1447	0.1874	0.2226	0.1722	**0.3838**	72.42
	HIT@3	0.2568	0.2674	0.2731	0.3083	0.2833	0.3373	0.3826	0.3249	**0.5708**	49.18
	HIT@10	0.4421	0.4634	0.4658	0.4769	0.4740	0.5378	0.5642	0.5214	**0.6956**	23.31

where sotfmax(·) is the softmax function, ConvTransE(·) is the decoder, and $\hat{\mathbf{E}}$ is the candidate entity embedding, which is summed over $\mathbf{E}^c$ and $\mathbf{E}^g$. The prediction is denoted as:

$$\hat{o}_{(s,r,t+1,c)} = \arg\max_{\mathcal{E}} \hat{p}(\mathcal{E}|s,r,c,\mathbf{G}^c_{\le t},\mathcal{G}_{\le t}). \tag{5}$$

It should be noted that our implementation is a type of early fusion that we first fuse the representations from two branches and then feed them into the decoder. This early fusion strategy can fully take advantage of the power of the decoder ConvTransE, which is a multi-layer convolutional neural network, to capture the interactions between the two contexts. In this way, the model can adaptively learn an optimal combination of the two contextual representations to yield the accurate prediction for a certain query. We utilize the cross-entropy loss defined below to optimize our model:

$$\mathcal{L} = \sum_{c\in C}\sum_{(s,r)\in G^c_{t+1}} \mathbf{y}_{(s,r,t+1,c)} \log \hat{\mathbf{p}}(\mathcal{E}|s,r,c,\mathbf{G}^c_{\le t},\mathcal{G}_{\le t}). \tag{6}$$

4 EXPERIMENTS

We conduct experiments on the two SCTc-TE datasets MidEast-TE and GDELT-TE to justify the proposed model. We are particularly interested in answering these research questions:

- **RQ1:** Does our proposed LoGo outperform the SOTA methods, especially the TKG methods?
- **RQ2:** How do the key model designs contribute to the forecasting?
- **RQ3:** What are the effects of various hyper-parameter settings and how the model perform in concrete cases?

4.1 Experimental Settings

Following the typical settings of TKG [3], we employ the evaluation metrics of Mean Reciprocal Rank (MRR) and Hit Rate(HIT)@{1, 3, 10} and use time-aware filtering during testing. The MRR is used to select the best-performing models based on the validation set. The implementation details are described in Appendix **??**.

4.1.1 Compared Methods. We compare our method with two strands of baseline methods, *i.e.,* static KG methods and TKG methods, which are widely used in the event forecasting problem under the TKG formulation. It should be noted that the methods designed for TCE with schema do not apply to our SCTc-TE problem since we do not have a pre-defined CE schema. **Static KG Methods** ignore the timestamp of the atomic event and combine all the triplet-formatted atomic events. We consider several representative methods. DistMulti [45], ConvE [10], ConvTransE [36], and RGCN [35]. **TKG Methods** are designed for temporal atomic events forecasting. Specifically, we encompass several representative models, including RE-NET [19], RE-GCN [26], HisMatch [24], and CMF [9]. HisMatch is the SOTA method. CMF can utilize the content information (news article embedding) of historical atomic events.

4.1.2 Implementation Details. We implement all the static methods by ourselves. In terms of RE-NET, RE-GCN, and HisMatch, we reuse their officially released code and use the global context graph. We re-implement CMF by ourselves according to the original paper, and the textual embedding is extracted using the RoBERTa model fine-tuned by SimCSE. For all the baseline and our methods, only the atomic events within CEs are used as training samples and predicted during testing. For the outlier atomic events, they are just used as part of the input graph. To be fair, for all the methods, set the embedding dimensionality $d = 200$, apply cross-entropy loss, grid search learning rate from $\{10^{-2}, 10^{-3}, 10^{-4}\}$ and weight decay from $\{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$. We search the historical length D from $\{1, 3, 5, 7, 10, 14\}$ for all the TKG methods. The number of propagation layers of all the graph-based methods is grid searched from $\{1, 2, 3\}$. We use Adam [21] optimizer and Xavier [15] initialization for all the methods.

4.2 Performance Comparison (RQ1)

Table 6 presents the overall performance of our method and all the baselines. We have the following observations from the results. First, our method outperforms all the baselines of both static KG and SOTA TKG methods by a large margin, demonstrating the effectiveness of our method. Second, among the two types of baselines, TKG methods generally perform better than the static KG methods, indicating that there are salient temporal patterns within our datasets and properly capturing the temporal patterns is crucial for effective forecasting. Third, graph-based methods generally perform well, showing that GNN is still the leading technique for structured temporal event forecasting. Moreover, the text-based method CMF does not outperform purely structured methods, showing that there is still a large space to explore for incorporating textual information into event forecasting.

Table 7: Ablation Study.

Dataset	Model	MRR	HIT@1	HIT@3	HIT@10
GDELT-TE	$\text{Logo}_{\text{local}}$	0.2164	0.1067	0.2490	0.4460
	$\text{Logo}_{\text{global}}$	0.1471	0.0539	0.1517	0.3509
	$\text{Logo}_{\text{share}}$	0.2391	0.1200	0.2734	0.4940
	$\text{Logo}_{\text{late}}$	0.2082	0.1050	0.2328	0.4271
	LoGo	0.2414	0.1224	0.2759	0.4970
MidEast-TE	$\text{Logo}_{\text{local}}$	0.4230	0.3258	0.4784	0.6088
	$\text{Logo}_{\text{global}}$	0.2920	0.1765	0.3317	0.5367
	$\text{Logo}_{\text{share}}$	0.4850	0.3726	0.5570	0.6810
	$\text{Logo}_{\text{late}}$	0.4018	0.3080	0.4590	0.5720
	LoGo	0.4978	0.3838	0.5708	0.6956

4.3 Ablation Study (RQ2)

we design several ablated models to justify the key designs of LoGo. First, we just keep one of the two contexts and set up two model variants, *i.e.,* $\text{Logo}_{\text{local}}$ that only keeps the local context branch and $\text{Logo}_{\text{global}}$ that keeps the global context branch. The results on two datasets in Table 7 indicate that: 1) solely relying on either of the local and global contexts will result in a performance drop; and 2) between the two individual contexts, the local context is more valuable than the global context since $\text{Logo}_{\text{local}}$ outperforms $\text{Logo}_{\text{global}}$ on both datasets of all the metrics. This makes sense because the local context zoom in to the evolution of the specific CE, thus making the model focus more on the entities that appear in the CE. Nevertheless, the global context can provide auxiliary environmental information for those sparse entities that have few information in the local context, which is the reason why LoGo is more effective than $\text{Logo}_{\text{local}}$.

Second, we design $\text{Logo}_{\text{share}}$ that let the two contexts share parameters. The performance of $\text{Logo}_{\text{share}}$ is a little bit lower than that of LoGo, demonstrating that two separate branches with isolated parameters can better capture the characteristics in both contexts. However, the performance of $\text{Logo}_{\text{share}}$ is still much better than other baselines.

Third, we try late fusion strategy of combining the two contexts, *i.e.,* $\text{Logo}_{\text{late}}$. The default strategy of LoGo is early fusion, while $\text{Logo}_{\text{late}}$ lets the two branches first pass two separate decoders and then fuse the decoded representations for ranking. The results in Table 7 show that early fusion is superior to late fusion. It verifies our motivation that the convolutional layers in the following decoder (ConvTransE) can well capture the interactions of the early-fused representations for optimal forecasting.

4.4 Model Study (RQ3)

Effect of Historical Length. We vary the historical length of T^g and T^c, and the MRR curve is shown in Figure 5 (a), which illustrates that 5 is the best for MidEast-TE and 10 is the best for GDELT-TE. Then we keep one of T^g and T^c as the best setting and change the other one and show the MRR curves in Figure 5 (b,c). The results indicate that proper historical length is crucial for good performance.

Effect of Propagation Layers. We vary the number of propagation layers of RGCN and demonstrate the performance change in

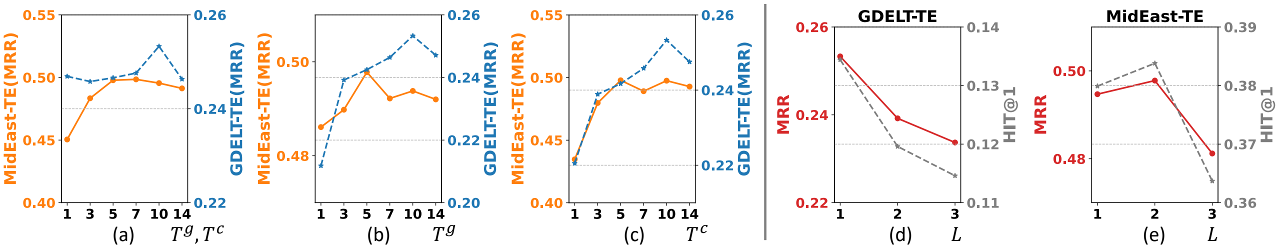
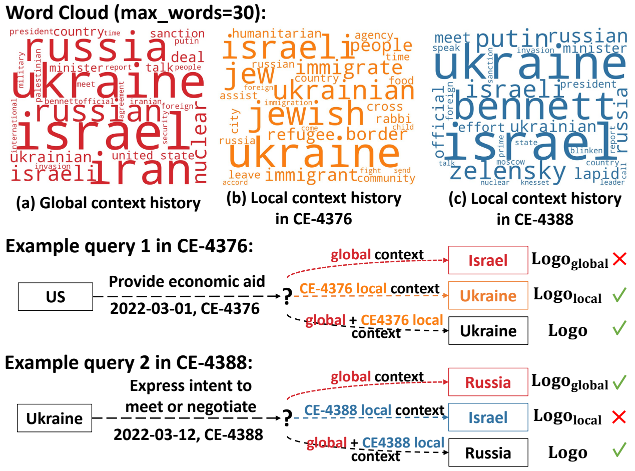


Figure 5: Sub-figure (a,b,c) show the effects of context historical length, and (d,e) study the effects of GNN layers.



Word Cloud (max_words=30):
(a) Global context history
(b) Local context history in CE-4376
(c) Local context history in CE-4388
Example query 1 in CE-4376:
US
Provide economic aid
2022-03-01, CE-4376
global context
CE-4376 local context
global + CE4376 local context
Israel
Ukraine
Ukraine
Logo$_{global}$ ✗
Logo$_{local}$ ✓
Logo ✓
Example query 2 in CE-4388:
Ukraine
Express intent to meet or negotiate
2022-03-12, CE-4388
global context
CE-4388 local context
global + CE4388 local context
Russia
Israel
Russia
Logo$_{global}$ ✓
Logo$_{local}$ ✗
Logo ✓

Figure 6: Example queries, corresponding news word clouds, and forecasting results from MidEast-TE.

Figure 5 (d,e). MidEast-TE performs best when $L = 2$, while $L = 1$ is best for GDELT-TE, showing that MidEast-TE is more sensitive to higher-order connections.

Case Study. As shown in Figure 6, for each query $(s, r, ?, t+1, c)$, we list the object o predicted by Logo, Logo$_{\text{global}}$, and Logo$_{\text{local}}$. We plot the word clouds based on tf-idf scores of the news articles in local and global context, *i.e.*, $\mathbf{D}^c_{\leq t}$ and $\mathcal{D}_{\leq t}$. In query 1 [4], Logo$_{\text{global}}$ wrongly predicts that *US* would provide economic aid to *Israel*, but Logo$_{\text{local}}$ and Logo correctly predict the true object *Ukraine* by incorporating the local context of *CE-4376*. From word clouds (a) and (b), we could observe that the global context involves a much larger number of international events and major countries, while *CE-4376* mainly depicts events about Ukraine refugees and the aid from other parties. There are indeed many positive past interactions between *US* and *Israel*, but the local context makes the model focus on the Ukraine refugee problem which leads to a better prediction. This demonstrates that the local context information is crucial. On another hand, in query 2 [5], Logo$_{\text{global}}$ and Logo correctly predict that *Ukraine* would express intent to negotiate with *Russia* while Logo$_{\text{local}}$ wrongly predict to *Israel*. One possible reason is that *CE-4388* mainly focuses on how Israel was interacting with Ukraine as shown in word cloud (c), but the global context contains more related past events, and thus provides a larger picture of the Russia-Ukraine conflict and its progress. This shows the importance of global context information.

5 RELATED WORK

Event Extraction. Typical EE tasks include sentence-level EE [41] and more complex document-level EE [39]. However, all of these works focus on small-scale individual events while ignoring dependencies between events including their temporal ordering. Recently, several studies have been conducted for complex event-level EE. One line of work focuses on structured CE extraction, such as IED [23] and RESIN-11 [12]. They first design some complex event schema, and then extract complex events that follow such schema. However, such extraction is highly dependent on schema construction, which is a non-trivial step and difficult to scale up to various CE types. Another line of work explores unstructured complex event analysis in a formulation of storyline discovery from multiple documents [4, 46]. They employ unsupervised clustering methods to construct storylines from a large set of news corpus, getting rid of schema construction. Therefore, in this work, we follow this clustering-based approach to identify complex events without a rigid schema, and we further extract out the structured events from the unstructured news text for downstream analysis and forecasting. A more recent work [34] generates natural language report analysis for temporal complex events using LLMs, but lacks quantitative analysis of the generated report. With the recent explosive progress of LLMs, researchers also take advantage of their zero-shot or few-shot capabilities to serve as a structured event extractor [20, 38]. Therefore, We also leverage LLMs [6, 32] for the EE in a zero-shot paradigm.

Temporal Event Forecasting. Temporal event forecasting aims to predict future events at future timestamps based on observed historical events. Many efforts have been devoted to TE forecasting [47] and different formulations have been raised. One line of work represents TE with time series [1, 31, 44], but they fail to capture the multi-relational relation between entities, not to say to model complex events. Several studies also explore the unstructured textual formulation for TE, where a CE is a storyline [5], and a TE is either in the form of summaries [14] or phrases [17]. However, redundant information is largely introduced due to the natural language expression and hinders the downstream forecasting task formulation. The benchmark ForecastQA [18] also falls in this unstructured formulation by providing only the textual historical events during forecasting, but the construction of this QA

[4] Here is the source url of the referred news article.
[5] Here is the source url of the referred news article.