From Document Layout to Causal Topology: An End-to-End Architecture for Temporal Causal Reasoning

Zhifeng Yang[1], **Wei Luo**[1], **Yongjian Chen**[1], **Shiqi Ou**[1,2†]
[1]School of Future Technology, South China University of Technology
[2]Guangdong Artificial Intelligence and Digital Economy Laboratory (Guangzhou)
ftyangzhifeng@mail.scut.edu.cn, 202364870991@mail.scut.edu.cn
ftyjchen@mail.scut.edu.cn, sou@scut.edu.cn

Abstract

Large language models (LLMs) generally lack deep causal reasoning capabilities in temporal reasoning. Inspired by document layout reordering, this paper proposes a paradigm shift from discrete narratives to causal topologies. It constructs two architectures: a training-free neuro-symbolic framework, temporal causal reasoning agent (TCR-Agent), which achieves interpretable causal deduction and counterfactual truncation through external causal graph instantiation and a symbolic intervention engine; and an end-to-end model, temporal causal reasoning former (TCR-Former), which internalizes these mechanisms into the Transformer attention space via a causal-topological attention mask and temporal span biases. Experiments on the TempoCausal benchmark demonstrate that TCR-Agent comprehensively surpasses existing training-free baselines. At the same time, TCR-Former achieves an accuracy 16.5% higher than that of state-of-the-art (SOTA) methods, outperforming large-scale models such as GPT-5.4 and DeepSeek-V3.2 with only approximately 8.6B parameters. Further analysis reveals a structure-reasoning disconnection phenomenon in existing methods: external symbolic constraints and chain-of-thought (CoT) reasoning can improve reasoning steps, yet fail to enhance deep causal deductive capability. By internalizing temporal causal reasoning capabilities into the model, TCR-Former effectively bridges this gap, providing an efficient and reliable end-to-end foundation for complex reasoning scenarios. The code and data can be obtained through https://github.com/Translab-SCUT/TCR-Former.

1 Introduction

Large language models (LLMs) have demonstrated strong generalization capabilities in tasks such as natural language understanding and instruction following [1, 2]. However, when tasks shift from time-independent semantic understanding to complex reasoning with temporal dependencies, existing models generally exhibit pronounced temporal blind spots [3, 4]. Studies indicate that LLMs perform poorly in temporal semantic parsing, long-context utilization, and complex event ordering, tending to rely on superficial textual order rather than genuine chronological order [5, 6]. To address this challenge, researchers have advanced the field along two dimensions: evaluation benchmarks and reasoning methods. On the evaluation side, early temporal question-answering benchmarks established foundational frameworks for task definition [7–9]. In contrast, subsequent evaluations have further revealed systematic deficiencies of LLMs in real-world temporality, multi-level dependencies, and

†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

cross-calendar reasoning [3, 5, 6, 10]. On the methodological side, recent work has partially improved performance in timeline ordering and explicit temporal question answering through techniques such as multi-hop temporal chain-of-thought (CoT) [11], temporal graph intermediate representations [4], and temporally-aware retrieval augmentation [12, 13].

Nevertheless, the evolution of the aforementioned benchmarks and methods has primarily centered on the explicit modeling of event temporal relations, essentially remaining at the level of superficial timeline ordering [4, 14, 15] and generally lacking attention to the deep causal relationships underlying temporality. In real-world decision-making scenarios involving implicit temporal windows, multi-entity narratives, and counterfactual reasoning, models must not only sequence events chronologically but also capture the causal topology among events to support abductive and hypothetical inference. However, reasoning within structural causal models involves complex operations such as abduction, intervention, and counterfactuals, which cannot be readily achieved through simple event ordering [16] or flat retrieval [17]. Some efforts have attempted to enhance reasoning reliability through causalized CoT [18, 19]. Still, these methods generally impose causal structure by externally controlling the reasoning process and do not genuinely internalize causal topology awareness and intervention mechanisms into the model's forward computation, thus remaining constrained in cross-scenario generalization and reasoning efficiency [15]. This predicament reveals that merely superimposing causal structure through external symbolic constraints does not address the essence of the problem. The crux of temporal causal reasoning lies in internalizing the topology-aware capacity for temporal causation into the model's reasoning process.

Inspired by visual document understanding, we observe a profound computational isomorphism between complex temporal causal reconstruction and layout structuring tasks. ROAP [20] and DeepSeek-OCR 2 [21] reorganize discrete, disordered visual blocks in two-dimensional space into logically ordered one-dimensional sequences through reading-order-aware biases and visual causal flow mechanisms. The core challenge faced by temporal reasoning is highly analogous; both essentially require reconstructing discrete, disordered perceptual fragments into a structured logical topology. Based on this insight, this paper aims to transfer the logic of layout reordering to temporal causal reconstruction by modeling the causal topology among events and internalizing causal topology awareness and temporal priors into the model's reasoning process and attention mechanisms.

Motivated by the above, we propose the temporal causal reasoning agent (TCR-Agent) framework, which explicitly reconstructs the causal topology in complex narratives through an external causal graph and a symbolic intervention engine. To further enhance end-to-end reasoning efficiency and integrate temporal causal priors into the underlying model representations, we extend this framework by proposing the temporal causal reasoning former (TCR-Former), which employs a causal-topological attention mask, explicit temporal span biases, and a native counterfactual intervention mechanism to achieve end-to-end temporal causal topology awareness.

The main contributions of this paper are as follows.

- We construct the TempoCausal-Bench to address the lack of deep causal reasoning evaluation in existing benchmarks.
- We propose the TCR-Agent framework, a training-free, model-agnostic external causal graph and symbolic intervention scheme compatible with various open-source and proprietary LLMs, providing interpretable and intervenable system-level support for complex temporal causal reasoning.
- We propose the TCR-Former model, an end-to-end architecture that internalizes causal topology awareness and intervention mechanisms into Transformer forward computation, providing an efficient and reliable temporal causal reasoning backbone for complex reasoning scenarios.

2 TCR-Agent Algorithm

TCR-Agent adopts a neuro-symbolic reasoning architecture and the core of TCR-Agent lies in leveraging an external graph memory mechanism to reconstruct the latent causal structure, enabling a pre-trained LLM to perform verifiable counterfactual reasoning and multi-hop temporal deduction without modifying its internal parameters. The framework formally integrates the structural causal model (SCM) and the potential outcome framework (POF). SCM provides a formal representation

and operations for the causal topology, while POF is utilized to evaluate evolutionary outcomes under counterfactual settings. The core algorithmic flow is presented in Algorithm 1 (Appendix D).

2.1 Causal Graph Extraction

This module aims to reconstruct the causal topology among events from unstructured text explicitly. We design a query-aware spatiotemporal extractor E_θ and a symbolic graph constructor B to work in concert, achieving end-to-end construction of a structured causal directed acyclic graph (DAG) from raw text. Temporal question answering typically involves one or more text segments describing the background of a question. Assume it is a sequence of N_c text chunks, $C = \{c_1, \cdots, c_{N_c}\}$. Let P denote the user query prompt, composed of a temporal question and other relevant hints, such as a CoT for solving the question. The spatiotemporal extractor E_θ takes C and P as input and extracts a causal graph by invoking the LLM. The extraction process is shown in Equations (1) and (2).

$$(S, G_{\text{raw}}) = E_\theta(C, P) \triangleq \text{LLM}_\theta(C, P) \tag{1}$$

$$G_{\text{raw}} = (\mathcal{V}_{\text{raw}}, \mathcal{E}_{\text{raw}}) \tag{2}$$

where, S records intermediate reasoning steps such as date differences, quantity accumulation, and temporal boundary determination to suppress information decay and hallucinations during long-context parsing. G_{raw} denotes the raw graph data, with $\mathcal{V}_{\text{raw}}$ being the set of extracted event nodes and $\mathcal{E}_{\text{raw}}$ the set of causal edges among these nodes.

Subsequently, the symbolic graph constructor B instantiates G_{raw} into a directed acyclic graph $G = (\mathcal{V}, \mathcal{E})$. For each node $v_i \in \mathcal{V}$, its original temporal anchor t_i is preserved, and its textual description and quantitative data are fused into a unified semantic representation, as formulated in Equation (3).

$$\tilde{v}_i = \text{desc}_i \oplus \text{Annotate}(d_i), \forall v_i \in \mathcal{V} \tag{3}$$

where desc_i is the textual description of the node v_i, d_i is the quantitative numerical field, $\text{Annotate}(\cdot)$ denotes a function that formats numerical data into explicit annotation text, and $\oplus$ denotes string concatenation. Consequently, each node simultaneously carries the temporal anchor t_i and the enriched semantic representation $\tilde{v}_i$, forming a three-dimensional representation space that fuses temporal, semantic, and numerical information.

For each directed edge $e_{ij} \in \mathcal{E}$, the associated attributes include the source node src_i, the target node tgt_j, the relation type $type_{ij}$, and the reason $reason_{ij}$, where $reason_{ij}$ records the causal derivation basis from the source event to the target event. During instantiation, the graph constructor B resolves potential contradictions among node timestamps via topological sorting, enforces the acyclicity constraint, and ultimately outputs a strictly structured logical trajectory G_{logic}.

To distinguish between causal and non-causal relationships among events, we categorize events in complex temporal question answering scenarios into three types and apply differentiated processing strategies:

(1) Irrelevant Events: Events with no logical intersection in semantics or time. The system directly filters these out during the causal graph extraction phase.

(2) Co-occurring Events: Events that merely overlap or follow a sequential order in time, but lack substantial causal drivers. The system retains only the independent states and temporal anchors of these nodes in the causal graph, without establishing any causal directed edge connections.

(3) Causal Events: Events with explicit causal dependencies. The system not only establishes directed edge connections but also generates attribute fields such as $reason$ to accurately characterize and record the specific causal derivation basis from the source event to the target event.

2.2 Counterfactual Intervention

This module aims to support what-if reasoning by performing symbolic-level structural perturbations to truncate and redirect causal chains. To this end, we design a graph intervention engine $\mathcal{I}$. Semantically, this intervention operation is equivalent to the do-operator in SCM, cutting off incoming edges

to specific variables via counterfactual truncation and observing system changes. The intervention operation is triggered by counterfactual assumptions in P. When P contains counterfactual statements such as "if an event had not occurred," the LLM parses the intervention parameters $\psi = (v_k, action)$, where v_k is the target node and $action \in \{\text{remove}, \text{cut_edges}\}$ specifies the type of intervention. Given the intervention parameters ψ, $\mathcal{I}$ applies structural perturbations to the current causal graph $G = (\mathcal{V}, \mathcal{E})$ to produce a modified causal graph $G' = (\mathcal{V}', \mathcal{E}')$, as computed in Equation (4).

$$G' = \mathcal{I}(G, \psi) = \begin{cases} G \setminus \{v_k\} \cup \mathcal{E}_{\text{preserve}}(v_k), & \text{if } action = \text{remove} \\ G \setminus \{(v_k, u) | (v_k, u) \in \mathcal{E}\}, & \text{if } action = \text{cut_edges} \end{cases} \tag{4}$$

where, u denotes the target node of a directed edge originating from v_k in the causal graph. When $action = \text{remove}$, the target node v_k is removed from the graph, while certain adjacent edges are explicitly preserved via $\mathcal{E}_{\text{preserve}}(v_k)$ to maintain the necessary connectivity of the narrative context. When $action = \text{cut_edges}$, all outgoing edges from v_k are deleted while the node itself is retained. The perturbed causal graph G' is linearized via a depth-first traversal into a topological description string $D(G')$, which explicitly encodes the temporal gaps, semantic descriptions, and causal justifications between event nodes.

2.3 Dual-Track Reasoning

This module aims to constrain autoregressive generation into a verifiable symbolic execution process through serialized cognitive procedures and explicit invariant checking. The core reasoning is executed by a dual-track reasoner R, which integrates the unstructured raw text C and the structured logical trajectory $D(G')$. To regulate the LLM's reasoning process, we constrain the generation trajectory using serialized tags organized into three parts: checking, reasoning, and reflection. Their specific contents are as follows.

(1) Checking: Describes $D(G')$ in natural language, and based on C checks the temporal computation correctness and causal derivation consistency of $D(G')$, and if incorrect, corrects $D(G')$.

(2) Reasoning: Performs reasoning according to C, P, and $D(G')$, and returns Unknown if the information is insufficient.

(3) Reflection: Reflects on whether the answer is correct, and if the answer is inconsistent with C and $D(G')$, corrects it and returns the correct answer.

3 TCR-Former Architecture

TCR-Former embeds the temporal causal modeling logic of the TCR-Agent paradigm into the Transformer's forward computation pipeline. TCR-Former consists of a temporal causal encoder and a reasoning decoder. The encoder extracts high-dimensional causal latent variables from fragmented text, and the decoder feeds these latent variables, along with the original factual information and user query instructions, into the autoregressive generation process. The architecture of the TCR-Former model is illustrated in Fig. 1. The training strategy of TCR-Former is detailed in Appendix F.

3.1 Temporal Causal Encoder

The temporal causal encoder uses Qwen-3-0.6B as its backbone, aiming to extract structured temporal causal latent variables from unstructured text fragments.

3.1.1 Input Sequence Processing

For a sequence of N_c text chunks $C = \{c_1, \cdots, c_{N_c}\}$, the model first converts them into continuous text token representations $\mathbf{V} \in \mathbb{R}^{B \times L_c \times d_{\text{enc}}}$ through a word embedding layer, where B is the batch size, L_c is the total length of the concatenated text tokens, and d_{enc} is the encoder hidden dimension. To discover the implicit causal structure within the text, we introduce a set of N learnable event anchors as a causal query token embedding matrix $\mathbf{Q} = \{\mathbf{q}_1, \cdots, \mathbf{q}_N\}$. Each event anchor $\mathbf{q}_k$ is used to query and aggregate a potential causal event from the text. In experiments, we set $N = 128$, which exceeds the maximum possible number of events in any sample, and the surplus slots are set to an empty event state during training. $\mathbf{q}_k$ consists of two components, as shown in Equation (5).

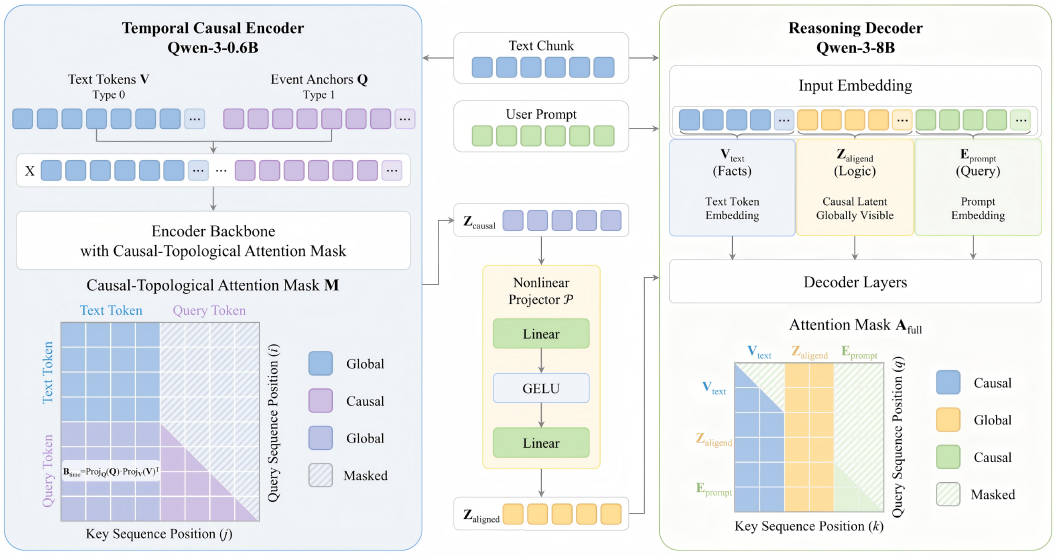


Figure 1: TCR-Former architecture.

$$\mathbf{q}_k = \mathbf{E}_{\text{anchor}}(k) + \mathbf{E}_{\text{span}}(k), \quad k \in [1, N] \tag{5}$$

where, $\mathbf{E}_{\text{anchor}} \in \mathbb{R}^{N \times d_{\text{enc}}}$ is a position-independent base anchor embedding that provides an independent basis vector for each potential event slot. $\mathbf{E}_{\text{span}} \in \mathbb{R}^{N \times d_{\text{enc}}}$ is a learnable time span bias embedding, which endows each event slot with a different temporal domain prior, for example, making certain anchors more inclined to attend to short-term events while others to long-range events. Both $\mathbf{E}_{\text{anchor}}$ and $\mathbf{E}_{\text{span}}$ are trainable parameters.

The text tokens and query tokens are finally concatenated into a one-dimensional input sequence $\mathbf{X} = [\mathbf{V}; \mathbf{Q}] \in \mathbb{R}^{B \times (L_c + N) \times d_{\text{enc}}}$. For differentiated treatment in subsequent attention computations, we label text tokens as type 0 and query tokens as type 1.

3.1.2 Causal-Topological Attention Mask

Temporal causal reasoning requires simultaneously handling two types of interactions with distinct natures: bidirectional semantic understanding among text tokens and strict causal ordering among event query tokens. To this end, we design a causal-topological attention mask $\mathbf{M}$.

Let the total sequence length be $L = L_c + N$, where positions 1 to L_c correspond to text tokens and positions $L_c + 1$ correspond to query tokens. The mask matrix $\mathbf{M} \in \mathbb{R}^{L \times L}$ is dynamically constructed before the attention computation of each layer. Its element assignment rule is given in Equation (6).

$$\mathbf{M}_{i,j} = \begin{cases} 0, & \text{if } i, j \le L_c \text{ (global bidirectional: semantic coherence)} \\ 0, & \text{if } i > L_c, j \le L_c \text{ (queries attend to all text tokens)} \\ 0, & \text{if } i, j > L_c \wedge j \le i \text{ (global causal)} \\ -\infty, & \text{otherwise} \end{cases} \tag{6}$$

The mask $\mathbf{M}$ defines a mixed attention space. Among text tokens, where $i, j \le L_c$, all text tokens are visible to each other, achieving global bidirectional attention and ensuring complete contextual understanding of the textual facts. For query tokens attending to text tokens, where $i > L_c, j \le L_c$, all query tokens can attend to all text tokens, thereby enabling retrieval of event-related information from the full text. Among query tokens, where $i, j > L_c \wedge j \le i$, the condition $j \le i$ ensures that the overall direction of event queries follows a causal order, meaning future queries cannot inform past ones.

3.1.3 Time-Aware Positional Bias

To further strengthen the encoder's perception of the physical time dimension, we introduce an explicit time bias matrix $\mathbf{B}_{\text{time}}$. For the query token embeddings $\mathbf{Q}$ and text token embeddings $\mathbf{V}$, we apply learnable linear projections $\text{Proj}_{\mathbf{Q}}$ and $\text{Proj}_{\mathbf{V}}$, respectively, and then compute their inner product to obtain the bias matrix $\mathbf{B}_{\text{time}}$, as shown in Equation (7).

$$\mathbf{B}_{\text{time}} = \text{Proj}_{\mathbf{Q}}(\mathbf{Q}) \cdot \text{Proj}_{\mathbf{V}}(\mathbf{V})^{\top} \tag{7}$$

$\mathbf{B}_{\text{time}}$ is added to the region where $i > L_c, j \leq L_c$ in Equation (6), enabling query tokens to adjust their attention weights based on temporal logic when aggregating textual information.

3.1.4 Auxiliary Timestamp Prediction

To enable the latent space representation to encode physical temporal positions explicitly, we attach a lightweight timestamp regression head to the encoder. This head consists of two fully connected layers with GELU activation and maps the final hidden state of each query token to a one-dimensional normalized timestamp prediction. The regression head is supervised only during training via mean squared error (MSE) loss, injecting temporal awareness into the latent variables.

3.1.5 Computation Flow

The overall computation flow of the encoder can be summarized as follows. The text embeddings $\mathbf{V}$ and event anchor embeddings $\mathbf{Q}$ are concatenated and fed into the backbone network. Before each attention layer computation, the model dynamically constructs the causal topological attention mask $\mathbf{M}$ based on the token type identifiers and adds the time bias matrix $\mathbf{B}_{\text{time}}$. After the network output, the hidden states corresponding to all query tokens are extracted to obtain the causal latent variable sequence $\mathbf{Z}_{\text{causal}} \in \mathbb{R}^{B \times N \times d_{\text{enc}}}$.

3.2 Reasoning Decoder

The reasoning decoder is built on the Qwen-3-8B model and aligns the encoder's causal latent variables with user query instructions while performing autoregressive text generation. Since the latent space dimensions and semantic distributions differ between the encoder and decoder, a nonlinear projector $\mathcal{P}$ is placed between them for dimension alignment and representation adaptation. The projector consists of two fully connected layers with GELU activation. To simultaneously leverage the fine-grained semantics of the original factual information and the topological priors in the causal graph structure during reasoning, the decoder adopts a fact-logic bypass fusion strategy. The input embedding sequence of the decoder is formed by concatenating three parts, as shown in Equation (8).

$$\mathbf{E}_{\text{input}} = [\mathbf{V}_{\text{text}}; \mathbf{Z}_{\text{aligned}}; \mathbf{E}_{\text{prompt}}] \tag{8}$$

where, $\mathbf{V}_{\text{text}}$ is the token embeddings of the original text chunks obtained through the decoder's word embedding layer, which constitutes the factual track. $\mathbf{Z}_{\text{aligned}}$ is the aligned causal latent variables from the encoder output via the projector, which constitutes the logical track. $\mathbf{E}_{\text{prompt}}$ is the embedding representation of the user query instruction.

The corresponding attention mask $\mathbf{A}_{\text{full}}$ is constructed in a block-diagonal form. Within $\mathbf{V}_{\text{text}}$, a standard causal mask is maintained. The $\mathbf{Z}_{\text{aligned}}$ part is globally visible to all positions. The $\mathbf{E}_{\text{prompt}}$ part follows the standard autoregressive mask.

3.3 Native Counterfactual Intervention

TCR-Former supports executing counterfactual interventions at the attention mechanism level, without re-encoding the input text or invoking an external graph manipulation engine. The intervention operation is implicitly triggered by counterfactual assumptions in the user query prompt P. When the user prompt contains counterfactual statements such as "if an event had not occurred," the system matches them with event anchors in the encoder input. It resolves the query token index k_{ψ} corresponding to the target event.

After obtaining the index k_ψ, the encoder performs topological pruning on the mask matrix $\mathbf{M}$ when constructing the causal-topological attention mask. The element-wise modification rule for the intervention mask $\mathbf{M}^\psi$ is given in Equation (9).

$$\mathbf{M}^\psi_{i,j} = \mathbf{M}_{i,j} \cdot \mathbb{I}(i \neq k_\psi) \cdot \mathbb{I}(j \neq k_\psi) \tag{9}$$

That is, all incoming and outgoing attention connections of the target query token are cut off by setting the mask values at its corresponding positions to $-\infty$. Through this operation, the causal latent variables $\mathbf{Z}_{\text{causal}}$ output by the encoder implicitly reflect the topological structural changes after the causal chain is broken. In the decoding stage, $\mathbf{Z}_{\text{aligned}}$ carries this intervention information into the autoregressive generation process, and the model reasons about the outcome under the counterfactual scenario based on the latent representations of the remaining causal paths.

3.4 Forward Computation Flow

The overall forward computation flow of TCR-Former is presented in Algorithm 2 (Appendix D). The input consists of the text chunk sequence C and the user query prompt P. First, the encoder generates text token embeddings $\mathbf{V}$ and event anchor embeddings $\mathbf{Q}$ with time span biases, concatenates them, computes the time bias matrix, and constructs the causal-topological attention mask. If P contains a counterfactual assumption, the intervention index is resolved, and topological pruning is applied to the mask. The encoder outputs hidden state sequences, from which the causal latent variables $\mathbf{Z}_{\text{causal}}$ corresponding to the query tokens are extracted, and timestamps $\hat{T}$ are predicted via the regression head. Subsequently, $\mathbf{Z}_{\text{causal}}$ is aligned to the decoder space by the projector, yielding $\mathbf{Z}_{\text{aligned}}$. The decoder concatenates the original text embeddings $\mathbf{V}_{\text{text}}$, the aligned latent variables $\mathbf{Z}_{\text{aligned}}$, and the prompt embedding $\mathbf{E}_{\text{prompt}}$, constructs the full attention mask, performs autoregressive decoding, and finally outputs the generated response Y.

4 TempoCausal-Bench

Existing mainstream temporal benchmarks mostly rely on explicit timestamps to anchor questions, allowing models to answer easily through shallow matching and making it difficult to reach deep temporal causal reasoning. However, complex decision-making and long-range narratives in real-world scenarios precisely involve characteristics such as implicit temporal boundary delineation, concurrent state evolution of multiple entities, and long-range causal lags. To systematically evaluate the topological causal reasoning capabilities of large language models in these scenarios, we build the TempoCausal-Bench by upgrading linear temporal question answering into a structured temporal causal reasoning task through a five-dimensional reconstruction paradigm, based on the difficult samples from TimeQA-Hard [8]. The detailed construction process of the TempoCausal-Bench is presented in Appendix E.

5 Experiments

5.1 Experimental Setup

Metrics. The evaluation metrics adopted are: **(1) Exact Match (EM):** requires string-identical matching with the standard answer. **(2) F1-Score:** computing based on token-level overlap to mitigate the strict matching bias for long answers. **(3) LLM-Acc:** employing DeepSeek-V3.2 [22] and Gemini-3.1-flash [23] as judges to perform semantic-level correctness assessment, the detailed evaluation procedures are provided in Appendix G.

Baseline. We select representative closed-source and open-source models for evaluation, including GPT-5.4 [24], DeepSeek-V3.2 [22], Qwen-3-8B [25], and Qwen-2.5-7B [26]. In addition, we include methods designed to enhance the temporal question-answering capability of LLMs for comparison, including TG-LLM [4], Time-CoT [14], and TISER [15]. For all experiments, these methods are used in their CoT configurations strictly following the prompts publicly disclosed in their papers, and no model training is conducted. Specific introductions to each method are given in Appendix H. In the TCR-Former experiments, when Qwen-3-8B serves as the decoder, the encoder adopts Qwen-3-0.6B, and when Qwen-2.5-7B serves as the decoder, the encoder adopts Qwen-2.5-0.5B.

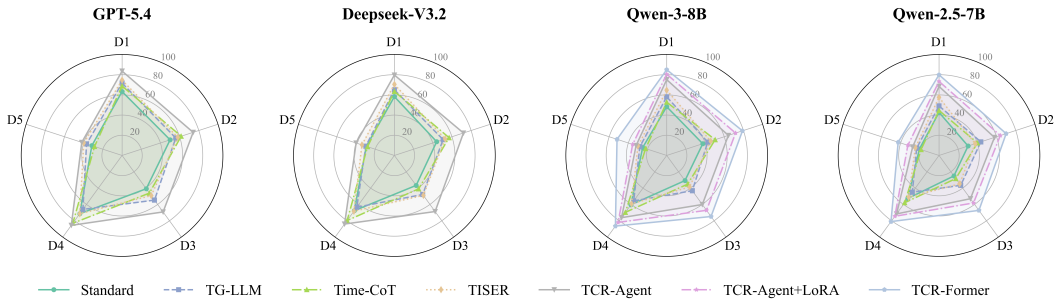


Figure 2: Experimental results on each dimension of the TempoCausal-Bench (LLM-Acc).

In the TCR-Agent+LoRA experiment, the TCR-Agent framework is employed, and the model is fine-tuned on question-answer pairs generated by TCR-Agent (GPT-5.4) on the TempoCausal-Bench training set.

5.2 Main Results

We conduct large-scale comparative experiments on the TempoCausal-Bench. All experiments are independently run three times under identical settings, and the average results are reported to eliminate stochastic fluctuations. Table 1 and Fig. 2 present the performance of each method on the TempoCausal-Bench. The results indicate that complex temporal causal reasoning poses a severe challenge to existing LLMs. Methods such as TG-LLM, Time-CoT, and TISER yield only limited improvements, suggesting that external prompt engineering cannot systematically resolve deep topological reasoning problems. In contrast, TCR-Agent consistently outperforms training-free baselines across all backbone models. When built on GPT-5.4, it attains an LLM-Acc of 70.4%, an 11.2% improvement over the best baseline TISER (LLM-Acc = 59.2%). Furthermore, TCR-Former internalizes structural awareness into model computation, achieving a secondary leap in performance. On Qwen-3-8B, its LLM-Acc, EM, and F1 reach 75.7%, 55.9%, and 68.9%, respectively, comprehensively outperforming all compared methods. To quantify the net gain from architectural internalization, we introduce a fine-tuned TCR-Agent + LoRA (Qwen-3-8B) baseline, which achieves an LLM-Acc of 67.4%; TCR-Former further improves upon this by 8.3%. These experimental results demonstrate that designs such as the causal-topological attention mask and time-span biases directly encode topological reasoning priors into the forward computation, effectively bridging the semantic gap in cross-modal and neuro-symbolic transformations.

Table 1: TempoCausal benchmark experimental results. The **Bold** numbers indicate the best performance among all models.

Method	Algorithm	EM (↑)	F1 (↑)	LLM-Acc (↑)
Train-Free				
GPT-5.4	Standard	28.8	44.4	51.3
	TG-LLM	32.0	49.3	57.1
	Time-CoT	30.0	47.8	54.5
	TISER	37.6	52.1	59.2
	TCR-Agent (Ours)	46.9	65.6	70.4
Deepseek-V3.2	Standard	25.7	42.1	47.4
	TG-LLM	30.3	46.1	50.4
	Time-CoT	31.2	46.4	50.6
	TISER	35.3	47.7	54.6
	TCR-Agent (Ours)	46.4	61.4	68.2
Qwen-3-8B	Standard	17.7	34.3	42.5
	TG-LLM	21.1	37.9	46.8
	Time-CoT	23.0	39.6	45.4
	TISER	25.2	39.1	49.2
	TCR-Agent (Ours)	39.9	51.7	60.1
Qwen-2.5-7B	Standard	11.9	24.5	35.2
	TG-LLM	14.6	28.5	39.6
	Time-CoT	13.5	28.8	37.5
	TISER	17.6	31.7	41.9
	TCR-Agent (Ours)	27.8	42.1	54.8
Train-Based				
Qwen-3-8B	TCR-Agent + LoRA	49.1	60.4	67.4
	TCR-Former (Ours)	**55.9**	**68.9**	**75.7**
Qwen-2.5-7B	TCR-Agent + LoRA	38.4	53.0	60.5
	TCR-Former (Ours)	45.8	59.3	67.8

Out-of-Domain Experiment. To evaluate the models' generalization capability, we conduct tests on the widely used temporal reasoning benchmarks TempReason (L3) [27] and TGQA [4]. The experimental results show that our proposed methods surpass existing SOTA methods, where TCR-Former achieves an EM of 95.4% on TempReason (L3) and an EM of 91.9% on TGQA. The complete experimental results are presented in Appendix C.1.

5.3 Ablation Study

Table 2: Ablation experiment results.

Component	Architecture Setting	EM (↑)	F1 (↑)	LLM-Acc (↑)
TCR-Agent Base: GPT-5.4	Full TCR-Agent	46.9	65.6	70.4
	w/o Scratchpad	43.2	60.1	66.8
	w/o DAG Instantiation	40.1	55.3	63.1
	w/o Counterfactual Intervention	44.0	62.7	67.5
	w/o Dual-Track Reasoning	41.3	58.6	65.2
	Standard	28.8	44.4	51.3
TCR-Former Base: Qwen-3-8B	Full TCR-Former	55.9	68.9	75.7
	w/o Time Span Bias	52.4	65.2	72.3
	w/o Fact-Logic Bypass Fusion	50.7	64.1	71.4
	w/o Causal-Topological Attention Mask	48.2	60.4	67.0
	Only Decoder	38.5	50.2	58.6
	Only Encoder + Decoder	34.2	47.7	55.4
	Standard	17.7	34.3	42.5

To quantify the contributions of each core component, we conduct ablation studies on the TempoCausal Benchmark. The results are presented in Table 2, and the detailed settings of the ablation study are provided in Appendix I. The experimental results show that, in the TCR-Agent architecture, removing DAG Instantiation leads to an LLM-Acc drop of 7.3%, the largest decline, indicating that converting text into a rigorous topological structure is a crucial prerequisite for overcoming temporal blind spots. In the TCR-Former architecture, removing the Causal-Topological Attention Mask causes an 8.7% drop in LLM-Acc, the largest decline among component ablations, proving that severing the bidirectional context of concurrent events severely damages the native perception of causal logic. Notably, Only Decoder achieves 58.6% LLM-Acc, substantially outperforming the 42.5% obtained by the Standard baseline. Whereas Only Encoder + Decoder falls to 55.4%, confirming that the superior performance of Full TCR-Former stems from architectural design rather than training data or encoder parameters. The detailed analysis of the Causal-Topological Attention Mask is presented in Appendix C.5.

6 Analysis

6.1 Reasoning Efficiency Analysis

To evaluate the computational efficiency of each method in practical deployment, we test the inference overhead of various methods on the TempoCausal-Bench using Qwen-3-8B. The results are shown in Fig. 3(a). TCR-Agent incurs the highest end-to-end latency and token consumption, as it requires LLM calls for graph extraction and dual-track reasoning. In contrast, TCR-Former internalizes causal-topological awareness and time biases entirely into the model's forward computation, reducing inference latency by 62% and output token count by approximately 44% compared with TCR-Agent. Its efficiency is comparable to that of TISER. A complete experimental analysis is provided in Appendix C.2.

6.2 Causal Graph Robustness Analysis

To evaluate the impact of causal graph extraction errors on downstream reasoning, we randomly select 500 samples from TempoCausal and conduct noise injection experiments based on TCR-Agent (GPT-5.4) and its ablation variant (w/o Dual-Track Reasoning). Specifically, we design two perturbation paradigms:

(1) Edge Deletion: randomly removing directed edges to simulate dependency omission.

(2) Node Interference: replacing node text with irrelevant, distracting text to simulate entity semantic distortion.

The experimental results are shown in Fig. 3(b). The results indicate that interference affecting nodes has a

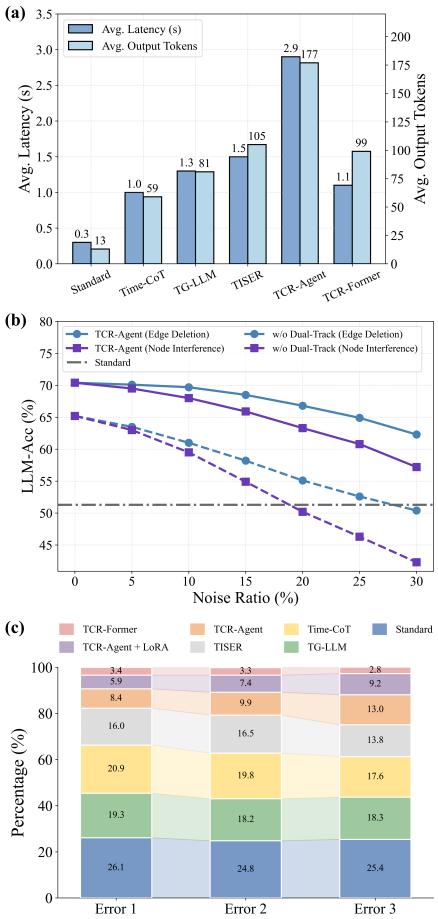


Figure 3: Indicator visualization. (a): Comparison of reasoning efficiency. (b): Robustness analysis of causal graphs. (c): Error Percentage.

greater impact than interference affecting edges, and that the Dual-Track Reasoning mechanism of TCR-Agent helps improve the system's robustness. A complete experimental analysis is provided in Appendix C.3.

6.3 Error Attribution Analysis

To systematically analyze the failure mechanisms of models in complex temporal tasks, we categorize error attributions into four primary error patterns: Temporal Boundary Misjudgment (Error 1), Causal Chain Breakage (Error 2), Rule Inference Mismatch (Error 3), and Other Errors (Error 4). Detailed explanations of the various errors and a detailed analysis of this section are provided in Appendix C.4.

Based on Qwen-3-8B, we randomly sample 200 instances per method and perform error attribution. Fig. 3(c) presents the error distribution for each method. Upon deeper analysis of the experimental results, we observe a noteworthy **structure-reasoning disconnection phenomenon**: external graph structures and CoT reasoning can make the model's reasoning structure more precise, but they fail to fundamentally address the model's deficiency in deductive capability. As shown in Fig. 3(c), compared with existing methods, TCR-Agent drastically reduces the error proportions of temporal boundary misjudgment (8.4%) and causal chain breakage (9.9%). However, rule inference mismatch remains as high as 13.0%, nearly on par with TISER (13.8%). This indicates that combining external graph structures with CoT for reasoning still struggles to handle error types such as Rule Inference Mismatch, which demand deeper deductive capabilities. In contrast, TCR-Former keeps the proportions of all three error types below 3.4%, with rule inference mismatch at only 2.8%, indicating that internalizing temporal causal priors into the Transformer attention space can effectively alleviate the structure-reasoning disconnection problem.

7 Related Work

Existing research on temporal causal reasoning primarily centers on two main paradigms. In neuro-symbolic reasoning, approaches leveraging graph structures and symbolic working memory [28–30] effectively enhance multi-hop compositional stability, yet they predominantly target static commonsense domains, lacking dedicated mechanisms for dynamic temporal evolution and counterfactual deduction. In temporal reasoning for large language models, existing approaches, including timestamp injection, temporal graphs, and chain-of-thought prompting [5, 8, 11, 31, 32], predominantly remain confined to superficial chronological ordering, failing to internalize temporal span biases and causal topological constraints into the foundational self-attention computation of the model. Furthermore, at the underlying computational architecture level, this work draws critical inspiration from layout rearrangement studies in visual document understanding. This field has progressed from early spatial coordinate modeling [33] to recent paradigms that directly encode reading order topology into forward computation via attention biases, such as ROAP [20] and DeepSeek-OCR 2 [21]. This motivates us to design the Causal-Topological Attention Mask and temporal span biases, transferring topology awareness into the self-attention layer to achieve end-to-end temporal causal modeling. The complete related work is detailed in Appendix A.

8 Conclusion

We propose TCR-Agent and TCR-Former, two systematic approaches that, respectively, follow the training-free neuro-symbolic framework and the natively internalized causal architecture to tackle temporal causal reasoning in complex narratives. On the TempoCausal-Bench, TCR-Agent consistently surpasses existing training-free baselines across multiple models. Moreover, TCR-Former achieves 75.7% accuracy with approximately 8.6B parameters, outperforming the previous best method by 16.5%, and breaking through the structure-reasoning disconnection bottleneck under external symbolic constraints. Ablation studies validate the effectiveness of each component in TCR-Agent and TCR-Former, providing an efficient and reliable reasoning paradigm for real-world decision-making scenarios such as hypothesis deduction with strong temporal dependencies and long-range lag prediction.

Acknowledgements

This work was supported by the Development Agreement between Saudi Aramco Technologies Company and South China University of Technology (SCUT) (SATC 2025-024) and by the Introduced Innovative R&D Team of Guangdong (2023ZT10L145), and it used the facilities of Pazhou Laboratory. All opinions expressed in this paper are the authors' and do not necessarily reflect the policies and views of the sponsors.

References

[1] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[2] Renze Lou, Kai Zhang, and Wenpeng Yin. Large language model instruction following: A survey of progresses and challenges. *Computational Linguistics*, 50(3):1053–1095, 2024.

[3] Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1204–1228, 2024.

[4] Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. Large language models can learn temporal reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10452–10470, 2024.

[5] Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. Test of time: A benchmark for evaluating llms on temporal reasoning. *arXiv preprint arXiv:2406.09170*, 2024.

[6] Shaohang Wei, Wei Li, Feifan Song, Wen Luo, Tianyi Zhuang, Haochen Tan, Zhijiang Guo, and Houfeng Wang. Time: A multi-level benchmark for temporal reasoning of llms in real-world scenarios. *arXiv preprint arXiv:2505.12891*, 2025.

[7] Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. Tempquestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, pages 1057–1062, 2018.

[8] Wenhu Chen, Xinyi Wang, and William Yang Wang. A dataset for answering time-sensitive questions. *arXiv preprint arXiv:2108.06314*, 2021.

[9] Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6663–6676, 2021.

[10] Zhongjian Miao, Hao Fu, and Chen Wei. Span: Benchmarking and improving cross-calendar temporal reasoning of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 35715–35723, 2026.

[11] Qingyu Tan, Hwee Tou Ng, and Lidong Bing. Towards robust temporal reasoning of large language models via a multi-hop qa dataset and pseudo-instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6272–6286, 2024.

[12] Zhang Siyue, Xue Yuxiang, Zhang Yiming, Wu Xiaobao, Luu Anh Tuan, and Zhao Chen. Mrag: A modular retrieval framework for time-sensitive question answering. *arXiv preprint arXiv:2412.15540*, 2024.

[13] Byeongjeong Kim, Jeonghyun Park, Joonho Yang, and Hwanhee Lee. Chronological passage assembling in rag framework for temporal question answering. *arXiv preprint arXiv:2508.18748*, 2025.

[14] Baosheng Yin and Naiyu Hu. Time-cot for enhancing time reasoning factual question answering in large language models. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

[15] Adrián Bazaga, Rexhina Blloshmi, Bill Byrne, and Adrià de Gispert. Learning to reason over time: Timeline self-reflection for improved temporal reasoning in language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28014–28033, 2025.

[16] Judea Pearl. *Causality*. Cambridge university press, 2009.

[17] Yuefei Chen, Vivek K Singh, Jing Ma, and Ruixiang Tang. Counterbench: Evaluating and improving counterfactual reasoning in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30350–30358, 2026.

[18] Xiangning Yu, Zhuohan Wang, Linyi Yang, Haoxuan Li, Anjie Liu, Xiao Xue, Jun Wang, and Mengyue Yang. Causal sufficiency and necessity improves chain-of-thought reasoning. *arXiv preprint arXiv:2506.09853*, 2025.

[19] Yubin Ge, Salvatore Romeo, Jason Cai, Raphael Shu, Yassine Benajiba, Monica Sunkara, and Yi Zhang. Tremu: Towards neuro-symbolic temporal reasoning for llm-agents with memory in multi-session dialogues. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18974–18988, 2025.

[20] Tingwei Xie, Jinxin He, and Yonghong Song. Roap: A reading-order and attention-prior pipeline for optimizing layout transformers in key information extraction. *arXiv preprint arXiv:2601.05470*, 2026.

[21] Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr 2: Visual causal flow. *arXiv preprint arXiv:2601.20552*, 2026.

[22] Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

[23] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[24] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[25] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[26] Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024. URL `https://api.semanticscholar.org/CorpusID:274859421`.

[27] Qingyu Tan, Hwee Tou Ng, and Lidong Bing. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, 2023.

[28] Junjie Wang, Mingyang Chen, Binbin Hu, Dan Yang, Ziqi Liu, Yue Shen, Peng Wei, Zhiqiang Zhang, Jinjie Gu, Jun Zhou, et al. Learning to plan for retrieval-augmented large language models from knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7813–7835, 2024.

[29] Zhili Shen, Chenxin Diao, Pavlos Vougiouklis, Pascual Merita, Shriram Piramanayagam, Enting Chen, Damien Graux, Andre Melo, Ruofei Lai, Zeren Jiang, et al. Gear: Graph-enhanced agent for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12049–12072, 2025.

[30] Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. Symbolic working memory enhances language models for complex rule application. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17583–17604, 2024.

[31] Zeyu Wang. Causalbench: A comprehensive benchmark for evaluating causal reasoning capabilities of large language models. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 143–151, 2024.

[32] Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2278–2288, 2018.

[33] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1192–1200, 2020.

[34] Xin Li, Yunfei Wu, Xinghua Jiang, Zhihao Guo, Mingming Gong, Haoyu Cao, Yinsong Liu, Deqiang Jiang, and Xing Sun. Enhancing visual document understanding with contrastive learning in large visual-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15546–15555, 2024.

[35] Sachin Raja, Ajoy Mondal, and CV Jawahar. Visual understanding of complex table structures from document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2299–2308, 2022.

[36] Yihao Ding, Soyeon Caren Han, Jean Lee, and Eduard Hovy. Deep learning based visually rich document content understanding: A survey. *Artificial Intelligence Review*, 2026.

[37] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, 2021.

[38] Qiming Peng, Yinxu Pan, Wenjin Wang, Bin Luo, Zhenyu Zhang, Zhengjie Huang, Yuhui Cao, Weichong Yin, Yongfeng Chen, Yin Zhang, et al. Ernie-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3744–3756, 2022.

[39] Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. Structurallm: Structural pre-training for form understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6309–6318, 2021.

[40] Junlong Li, Yiheng Xu, Lei Cui, and Furu Wei. Markuplm: Pre-training of text and markup language for visually rich document understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6078–6087, 2022.

[41] Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. Layoutreader: Pre-training of text and layout for reading order detection. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 4735–4744, 2021.

[42] Chong Zhang, Ya Guo, Yi Tu, Huan Chen, Jinyang Tang, Huijia Zhu, Qi Zhang, and Tao Gui. Reading order matters: information extraction from visually-rich documents by token path prediction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13716–13730, 2023.

[43] Chong Zhang, Yi Tu, Yixi Zhao, Chenshu Yuan, Huan Chen, Yue Zhang, Mingxu Chai, Ya Guo, Huijia Zhu, Qi Zhang, et al. Modeling layout reading order as ordering relations for visually-rich document understanding. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 9658–9678, 2024.

[44] Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. Docllm: A layout-aware generative language model for multimodal document understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8529–8548, 2024.

[45] Xiangnan Chen, Qian Xiao, Juncheng Li, Duo Dong, Jun Lin, Xiaozhong Liu, and Siliang Tang. Global structure knowledge-guided relation extraction method for visually-rich document. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1587–1598, 2023.

[46] Yixin Ji, Kaixin Wu, Juntao Li, Wei Chen, Mingjie Zhong, Xu Jia, and Min Zhang. Retrieval and reasoning on kgs: Integrate knowledge graphs into large language models for complex question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7598–7610, 2024.

[47] Yading Li, Dandan Song, Changzhi Zhou, Yuhang Tian, Hao Wang, Ziyi Yang, and Shuhao Zhang. A framework of knowledge graph-enhanced large language model based on question decomposition and atomic retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11472–11485, 2024.

[48] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. Grag: Graph retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4145–4157, 2025.

[49] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Learning Representations*, volume 2024, pages 3868–3898, 2024.

[50] Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, et al. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 163–184, 2024.

[51] Zheng Wang, Zhongyang Li, Zeren Jiang, Dandan Tu, and Wei Shi. Crafting personalized agents through retrieval-augmented generation on editable memory graphs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4891–4906, 2024.

[52] Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. Meminsight: Autonomous memory augmentation for llm agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33124–33140, 2025.

[53] Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8416–8439, 2025.

[54] Chunhui Zhang, Yiren Jian, Zhongyu Ouyang, and Soroush Vosoughi. Working memory identifies reasoning limits in language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16896–16922, 2024.

[55] Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6750–6774, 2023.

[56] Yuqing Wang and Yun Zhao. Tram: Benchmarking temporal reasoning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6389–6415, 2024.

[57] Sigang Luo, Yinan Liu, Dongying Lin, Yingying Zhai, Bin Wang, Xiaochun Yang, and Junpeng Liu. Etrqa: A comprehensive benchmark for evaluating event temporal reasoning abilities of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23321–23339, 2025.

[58] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, 2024.

[59] Zixi Jia, Qinghua Liu, Hexiao Li, Yuyan Chen, and Jiqiang Liu. Evaluating the long-term memory of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19759–19777, 2025.

[60] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

[61] Sen Yang, Xin Li, Lidong Bing, and Wai Lam. Once upon a time in graph: Relative-time pre-training for complex temporal reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11879–11895, 2023.

[62] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

A Complete Related Work

Structural Rearrangement and Topology Modeling in Document Visual Understanding. In document visual understanding, the core challenge lies in reconstructing discrete and disordered visual elements on a two-dimensional layout into a one-dimensional sequence conforming to a logical reading order [34–36]. LayoutLM [33] introduces spatial coordinate embeddings to provide preliminary modeling of document layout, yet subsequent studies indicate that relying solely on geometric position information is insufficient to capture complex layout structures [37, 38]. To address this, researchers have introduced hierarchical structures [39], markup languages [40], and other techniques for structured modeling, and have gradually established reading order as a key variable linking spatial layout and semantic logic [41–45]. Recent methods such as ROAP [20] and DeepSeek-OCR 2 [21] further encode topology priors related to reading order directly into the attention computation of Transformers via structure-aware attention biases and query mechanisms, achieving end-to-end rearrangement of unordered visual blocks into an ordered logical sequence. These studies collectively demonstrate that explicitly encoding latent structures into attention [21] is more effective than simple spatial modeling, inspiring us to transfer topology-aware attention design to temporal causal reasoning.

Graph Structures and Neuro-Symbolic Methods for Complex Reasoning. To alleviate the lost-in-the-middle problem in long-context reasoning [8], researchers have begun exploring graph structures and neuro-symbolic methods. In retrieval-augmented generation, path planning and subgraph retrieval based on knowledge graphs have effectively enhanced multi-hop reasoning capabilities [28, 46–48, 29, 49, 50]. In broader complex reasoning tasks, explicitly encoding entity relations via graph structures [51–53] or introducing symbolic working memory [30, 54] has improved the stability of models in handling cross-task compositional rules. However, the above methods are primarily oriented towards non-temporal scenarios, such as general reasoning and commonsense question answering, and lack targeted designs for complex temporal dependencies and counterfactual reasoning mechanisms. In contrast, the TCR-Agent framework proposed in this paper explicitly models temporal causal topology through an external causal graph and interventional topology pruning.

Temporal Perception and Causal Reasoning in LLMs. Standard retrieval-augmented generation (RAG) systems typically treat temporal information as shallow textual features, struggling to handle the dynamic evolution of events and long-range dependencies [1, 2, 7, 8]. Early methods relying on timestamp injection or multi-turn prompting [7, 8] remain at the level of superficial temporal ordering and cannot support deep causal modeling [5, 8, 55], a limitation also revealed by multiple benchmarks such as TRAM [56], ETRQA [57], and CausalBench [31]. Although subsequent studies have introduced strategies such as temporal graphs [11, 32], multi-hop data [58], and instruction tuning [59] to enhance structured temporal representations, these methods still essentially depend on external prompts or intermediate structures and do not internalize temporal biases and causal constraints into the core attention mechanism of the model. In contrast to these works, TCR-Former introduces temporal span biases and a causal-topological attention mask into the attention layers, thereby fundamentally reconstructing a nonlinear causal attention mechanism with topology-awareness.

B Limitation and Future Work

Although the proposed TCR-Agent and TCR-Former demonstrate significant performance advantages in complex temporal causal reasoning tasks and effectively alleviate the core "structure-reasoning disconnection" dilemma in existing methods, this study still has several limitations that warrant further exploration and refinement in future work.

Limitation. In terms of computational efficiency and training paradigms, TCR-Agent relies on multiple calls to large language models and cascaded scheduling of an external symbolic engine, resulting in high end-to-end inference latency that makes it difficult to meet high-throughput, low-latency real-time decision-making requirements. While TCR-Former reduces inference overhead by approximately 62% through architectural internalization, its three-stage training strategy is relatively complex. It has currently been validated mainly on Qwen-series backbones, leaving its cross-architecture generalization yet to be further verified. Second, regarding the causal representation construction mechanism, the causal graph instantiation in this paper heavily relies on large language models' text-extraction capabilities, resulting in an essentially "narrative-driven" topological reconstruction. In long texts with implicit information, ambiguous references, or conflicting

expressions, hallucinations or structural biases during extraction may propagate downstream through the DAG, thereby affecting the reliability of counterfactual intervention. Third, regarding the benchmark's multimodal coverage, the proposed TempoCausal-Bench currently focuses solely on the plain-text modality. However, numerous temporal causal reasoning tasks in real-world scenarios involve multimodal signals such as text, charts, and numerical sequences. The existing benchmark cannot measure the model's ability to fuse cross-modal information and perform collaborative causal inference, limiting the extrapolation and validation of architectures such as TCR-Former for practical multimodal applications.

Future Work. To address the above limitations, future work will proceed along the following directions. To improve efficiency, we will explore end-to-end joint optimization or representation distillation strategies to simplify the training pipeline and introduce sparsification or linear-complexity variants to reduce the computational overhead of dynamic masking. Meanwhile, we will attempt to establish formal connections between the Causal-Topological Attention mechanism and structural causal models, thereby providing theoretical bounds on the identifiability and error propagation in counterfactual deduction. We will further investigate causal discovery and uncertainty modeling, extending beyond the pure text extraction paradigm. By integrating symbolic causal discovery algorithms or incorporating Bayesian graph structure learning, we will probabilistically model causal edge weights and intervention paths, thereby quantifying the structural uncertainty during topology construction and enhancing the robustness of counterfactual truncation. To expand the multimodal benchmark, we plan to extend the evaluation dimensions of the TempoCausal-Bench to include multimodal scenarios, constructing a temporal causal reasoning evaluation dataset that includes modalities such as images, structured tables, and time-series plots. By designing tasks such as cross-modal event alignment, multi-source information conflict resolution, and chart-based trend reasoning, we will systematically assess the model's capability in causal topology reconstruction and counterfactual deduction in complex multimodal environments, providing a standardized testbed for developing a unified multimodal temporal causal reasoning backbone.

C Experiments and Analysis

C.1 Out-of-Domain Experiment

To evaluate the models' generalization capability, we conduct extended tests on two widely used temporal reasoning benchmarks, TempReason (L3) [27] and TGQA [4]. TCR-Agent uses GPT-5.4 as the base model, and TCR-Former employs Qwen-3-8B as the reasoning decoder. The detailed experimental results are presented in Table 3.

On the TempReason (L3) benchmark, the experimental results indicate that both methods proposed in this paper surpass existing strong baselines. As a classic temporal question answering evaluation benchmark, TempReason (L3) primarily relies on explicit temporal anchors, focusing on assessing chronological ordering and temporal commonsense computation, while lacking deep causal topology and counterfactual deduction. The training-free TCR-Agent achieves an EM of 93.2% and an F1 score of 95.6%, outperforming the previous best baseline, TISER (EM = 91.5%, F1 = 94.9%). Building upon this, the end-to-end architecture TCR-Former further attains an EM of 95.4% and an F1 score of 97.1%, achieving optimal performance across all metrics.

On the TGQA benchmark, which places greater emphasis on complex multi-entity temporal relation determination, our proposed methods likewise demonstrate significant performance advantages. TGQA requires models to accurately resolve overlapping events, lifespans, and relative durations within long-context multi-entity narratives. Experiments show that the EM and F1 of TCR-Agent reach 91.4% and 96.6%, representing improvements of 6.9% and 2.4% over TISER (EM = 84.5%, F1 = 94.2%), respectively. Furthermore, TCR-Former refreshes the state-of-the-art performance with an EM of 91.9% and an F1 of 97.5%. This demonstrates that the proposed architectures possess exceptional generalizability beyond complex causal deduction, confirming that effectively injecting causal topology and temporal biases into the model's reasoning mechanism fundamentally reinforces the language model's temporal causal perception and reasoning capabilities.

Table 3: Out-of-domain experimental results. Results with † are reported in the original papers [15].

Method	TempReason (L3)		TGQA	
	EM	F1	EM	F1
T5-large[60]†	28.8	46.8	54.8	71.3
Temp-T5[27]†	26.1	43.0	64.0	77.8
REMEMO-large[61]†	33.4	49.3	46.1	66.0
TG-LLM[4]†	35.6	46.9	79.7	85.0
TISER[15]†	91.5	94.9	84.5	94.2
TCR-Agent	93.2	95.6	91.4	96.6
TCR-Former	95.4	97.1	91.9	97.5

C.2 Reasoning Efficiency Analysis

To systematically evaluate the computational efficiency of each method in practical deployment, we test the inference overhead of various methods on the TempoCausal-Bench using Qwen-3-8B, with a batch size of 1 to simulate real-world deployment scenarios, running on 2 × NVIDIA A100-40G GPUs. The results are shown in Fig. 3(a). The standard prompt exhibits the lowest latency and output token consumption. Time-CoT, TG-LLM, and TISER introduce additional steps such as timeline explicitation, structured generation, and multi-stage self-reflection, leading to progressively increasing token outputs and correspondingly higher latency. TCR-Agent incurs the highest end-to-end latency and token consumption, as it requires LLM calls for graph extraction and dual-track reasoning, along with topological sorting and intervention pruning executed by an external symbolic engine. In contrast, TCR-Former internalizes causal-topological awareness and time biases entirely into the model's forward computation, reducing inference latency by 62% and output token count by approximately 44% compared with TCR-Agent. Its efficiency is comparable to that of TISER, effectively overcoming the efficiency bottleneck of training-free neuro-symbolic frameworks in large-scale deployment.

C.3 Causal Graph Robustness Analysis

To evaluate the impact of causal graph extraction errors on downstream reasoning, we randomly select 500 samples from TempoCausal and conduct noise injection experiments based on TCR-Agent (GPT-5.4) and its ablation variant (w/o Dual-Track Reasoning). Specifically, we design two perturbation paradigms:

(1) Edge Deletion: randomly removing directed edges to simulate dependency omission.

(2) Node Interference: replacing node text with irrelevant, distracting text to simulate entity semantic distortion.

Fig. 3(b) illustrates the performance degradation curves under the two types of noise. After removing dual-track reasoning, the system's sensitivity to noise increases significantly. Taking the 15% noise ratio as an example, under edge deletion, the accuracy of the complete TCR-Agent decreases from 70.4% to 68.5% (a drop of 1.9%), whereas that of the ablation variant drops sharply from 65.2% to 58.2% (a drop of 7.0%). The contrast is even more pronounced under node interference: at a high noise ratio of 30%, the complete TCR-Agent still maintains an accuracy of 57.2%, while the ablation variant plummets to 42.3%, far below the Standard baseline without a causal graph (51.3%). These results indicate that the original text pathway in dual-track reasoning provides critical fault tolerance against causal graph noise. Through fact-logic dual-track synergy, TCR-Agent can correct deviations by referencing the original text even when the causal graph is corrupted, thereby achieving significantly enhanced robustness.

C.4 Error Attribution Analysis

To systematically analyze the failure mechanisms of models in complex temporal tasks, we summarize error attribution into four primary error patterns.

(1) Temporal Boundary Misjudgment: inability to accurately locate the start and end ranges of implicit temporal windows or concurrent states, corresponding to typical error types in dimensions D1 and D2.

(2) Causal Chain Breakage: omitting critical nodes, reversing chronological order, or incurring intervention pruning failure in multi-hop dependencies, predominantly manifested in dimension D3.

(3) Rule Inference Mismatch: when faced with hypothetical questions or long-cycle lagged deduction, the model ignores explicit calculation rules and quantitative constraints given in the text, instead relying on superficial word associations or general commonsense to make erroneous inferences, leading to deduction results that severely violate the original textual logic; concentrated in dimensions D4 and D5.

(4) Other Errors: errors not falling into the above categories.

Based on Qwen-3-8B, we randomly sample 200 instances per method and perform error attribution. Fig. 3(c) presents the distribution of error types for each method. The Standard prompt and the three prompt-enhanced baselines all exhibit high proportions across the three primary error types. Although TG-LLM and TISER achieve slight improvements by introducing graph structures or reflection mechanisms, the overall gains are limited. Upon deeper analysis of the experimental results, we observe a noteworthy structure-reasoning disconnection phenomenon: external graph structures and CoT reasoning can make the model's reasoning structure more precise, but they fail to address the model's deficiency in deductive capability fundamentally. As shown in Fig. 3(c), compared with existing methods, TCR-Agent drastically reduces the error proportions of temporal boundary misjudgment (8.4%) and causal chain breakage (9.9%). Yet, rule inference mismatch remains as high as 13.0%, nearly on par with TISER (13.8%). Even when LoRA fine-tuning is applied on top of TCR-Agent, rule inference mismatch only drops to 9.2%, significantly higher than the other two error types (5.9% and 7.4%), further corroborating the structure-reasoning disconnection phenomenon. This indicates that relying solely on external graph structures for reasoning still struggles to handle error types like Rule Inference Mismatch that demand deeper deductive capabilities. In contrast, TCR-Former keeps the proportions of all three error types below 3.4%, with rule inference mismatch at only 2.8%, achieving balanced, substantial convergence across all dimensions. The above results indicate that relying solely on external structural constraints can achieve only static reconstruction of the causal topology, whereas embedding temporal causal priors into the attention flow and representation update mechanisms can simultaneously overcome the dual challenges of precise reasoning structure and insufficient deductive capability.

C.5 Causal-Topological Attention Mask Analysis

This section further examines the underlying mechanism of the Causal-Topological Attention Mask to verify whether its performance gains stem from actionable, structured causal topological representations rather than a superficial temporal ordering prior. Building upon the ablation study and error attribution analysis, we conduct a dimension-level ablation analysis and an internal failure mode investigation on the variant without this mask.

Table 4 summarizes the answering accuracy (LLM-Acc) across the five core evaluation dimensions of the TempoCausal-Bench under the ablation of the Causal-Topological Attention Mask. Compared with the Full TCR-Former, removing the causal-topological mask leads to performance drops of 2.9%, 5.5%, 9.1%, 16.2%, and 9.3% across dimensions D1 to D5, respectively, exhibiting the most pronounced declines in D4 and D5. If the mask merely provided a temporal ordering prior, its performance advantages would primarily concentrate on shallow tasks that heavily rely on temporal anchor alignment, specifically D1 and D2. However, the performance degradation on D1 and D2 upon mask removal remains relatively moderate, at 2.9% and 5.5%, respectively. In contrast, accuracy on the deep causal task D4 drops sharply by 16.2%, accompanied by substantial degradation in D3 and D5. This pronounced asymmetry in performance degradation indicates that the standard autoregressive mask is sufficient for handling pure event sequencing, whereas the performance advantage of the causal-topological mask primarily arises from sustaining the causal pathways required for counterfactual deduction and supporting structured intervention capabilities.

Regarding internal failure modes, building upon the error attribution analysis in Section 6.3, we supplement the error distribution for this ablation variant to dissect the specific role of the Causal-Topological Attention Mask, as shown in 5. After removing the causal-topological mask, Error 2 and

Table 4: Ablation of Causal-Topological Attention Mask (LLM-Acc). D1: Implicit temporal windows and conditional filtering. D2: Concurrent multi-entity state tracking. D3: Temporal lags and multi-hop dependencies. D4: Counterfactual cascading deduction. D5: Lagged amplification of minute perturbations.

Component	Architecture Setting	D1	D2	D3	D4	D5
TCR-Former Base: Qwen-3-8B	Full TCR-Former	85.0	79.0	74.8	86.5	51.8
	w/o Causal-Topological Attention Mask	82.1	73.5	65.7	70.3	42.5
	Standard	48.3	37.6	30.8	60.4	25.1

Error 3 increase substantially from 3.1% and 2.6% to 11.1% and 12.3%, respectively. In contrast, Error 1, which primarily reflects event boundary localization and associative ordering capabilities, increases only modestly from 3.3% to 6.1%. The marginal increase in Error 1 indicates that sequential ordering degradation is not the primary factor driving overall performance deterioration. Conversely, the substantial surge in Error 2 and Error 3 corresponds to the severance of causal dependencies and the impairment of deep deductive capabilities. This demonstrates that the intrinsic function of the causal-topological mask is to preserve directed causal connections among events within the attention space, thereby supporting the model in executing causally consistent deduction under interventional and counterfactual conditions, rather than merely enforcing chronological event constraints.

Table 5: Error proportions under mask ablation (%). Error 1: Temporal Boundary Misjudgment. Error 2: Causal Chain Breakage. Error 3: Rule Inference Mismatch.

Component	Architecture Setting	Error 1	Error 2	Error 3
TCR-Former Base: Qwen-3-8B	Full TCR-Former	3.3	3.1	2.6
	w/o Causal-Topological Attention Mask	6.1	11.1	12.3
	Standard	24.3	22.1	22.3

D TCR-Agent and TCR-Former Algorithm

The core algorithm process of TCR-Agent is shown in Algorithm 1.

Algorithm 1 TCR-Agent Temporal-Causal Reasoning

Input: Text chunks C, Prompt P
Output: Final answers $\hat{\mathbf{A}}_q$

▷ **Stage 1: Causal Graph Extraction**
$(S, G_{\text{raw}}) \leftarrow LLM_\theta(C, P)$
$G \leftarrow B(G_{\text{raw}})$
▷ **Stage 2: Counterfactual Intervention**
$\psi \leftarrow P$
if $\psi \neq \emptyset$ **then**
    $G' \leftarrow \mathcal{I}(G, \psi)$
else
    $G' \leftarrow G$
end if
$D(G') \leftarrow \text{GraphInterventionEngine.serialize}(G')$
▷ **Stage 3: Dual-Track Reasoning**
$\hat{\mathbf{A}}_q \leftarrow LLM_\varphi(C, D(G'), P)$
return $\hat{\mathbf{A}}_q$

The core algorithm process of TCR-Former is shown in Algorithm 2.

Algorithm 2 TCR-Former Temporal-Causal Forward Pass

Input: Text chunks C, Prompt P
Output: Generated response Y, Predicted timestamps $\hat{T}$

▷ **Stage 1: Temporal Causal Encoding**
$\mathbf{V} \leftarrow \text{EmbedText}(C)$; $\mathbf{Q} \leftarrow \text{AnchorEmbedding} + \text{TimeSpanBias}$
$\mathbf{X} \leftarrow \text{Concatenate}(\mathbf{V}, \mathbf{Q})$
$\mathbf{B}_{\text{time}} \leftarrow \text{ComputeTemporalBias}(\text{Proj}_Q(\mathbf{Q}), \text{Proj}_V(\mathbf{V}))$
$\mathbf{M} \leftarrow \text{BuildDynamicCausalMask}(L_c, N)$
$\mathbf{M} \leftarrow \text{InjectTemporalBias}(\mathbf{M}, \mathbf{B}_{\text{time}})$
if P contains counterfactual assumption **then**
    $k_\psi \leftarrow \text{ParseInterventionTarget}(P, \text{event_list})$
    $\mathbf{M} \leftarrow \text{ExecuteIntervention}(\mathbf{M}, k_\psi)$
end if
$\mathbf{H}_{\text{enc}} \leftarrow \text{ForwardEncoder}(\mathbf{X}, \text{attention_mask} = \mathbf{M})$ // Causal-Topological Attention
$\mathbf{Z}_{\text{causal}} \leftarrow \text{ExtractQueryStates}(\mathbf{H}_{\text{enc}}, \text{offset} = L_c)$ // Latent causal sequence
$\hat{T} \leftarrow \text{RegressTimestamp}(\mathbf{Z}_{\text{causal}})$ // Auxiliary timestamp prediction head
▷ **Stage 2: Reasoning & Bypass Fusion**
$\mathbf{Z}_{\text{aligned}} \leftarrow \text{ProjectToDecoderSpace}(\mathbf{Z}_{\text{causal}})$ // Nonlinear alignment
$\mathbf{V}_{\text{text}} \leftarrow \text{EmbedText}(C)$; $\mathbf{E}_{\text{prompt}} \leftarrow \text{EmbedPrompt}(P)$
$\mathbf{E}_{\text{input}} \leftarrow \text{Concatenate}(\mathbf{V}_{\text{text}}, \mathbf{Z}_{\text{aligned}}, \mathbf{E}_{\text{prompt}})$
$\mathbf{A}_{\text{full}} \leftarrow \text{MergeAttentionMasks}(\mathbf{V}_{\text{text}}, \mathbf{Z}_{\text{aligned}}, \mathbf{E}_{\text{prompt}})$
$Y \leftarrow \text{AutoregressiveDecode}(\mathbf{E}_{\text{input}}, \text{attention_mask} = \mathbf{A}_{\text{full}})$ // Constrained generation
return $Y, \hat{T}$

E TempoCausal-Bench

E.1 Five-Dimensional Evaluation

Existing mainstream temporal benchmarks mostly rely on explicit timestamps to anchor questions, allowing models to answer easily through shallow matching and making it difficult to reach deep temporal causal reasoning. We build the TempoCausal-Bench by upgrading linear temporal question answering into a structured temporal causal reasoning task through a five-dimensional reconstruction paradigm, based on the difficult samples from TimeQA-Hard [8]. The specific dimensions are introduced as follows.

(1) Implicit Temporal Windows and Conditional Filtering. Existing benchmarks often directly provide explicit time ranges, weakening the model's autonomous ability to perceive event boundaries. This dimension aims to test the model's reasoning ability to delineate temporal boundaries through event anchors without explicit dates.

(2) Concurrent Multi-Entity State Tracking. Traditional temporal data mostly unfold linearly along a single timeline. This dimension aims to test the model's ability to synchronously track the independent state evolution of multiple interacting entities within the same time window.

(3) Temporal Lags and Multi-Hop Dependencies. Causal chains in real narratives often span multiple paragraphs. This dimension tests the model's ability to concatenate and reconstruct scattered event fragments with relative time offsets from across the text into a complete long-range causal chain.

(4) Counterfactual Cascading Deduction. This dimension tests the model's understanding of a system's intrinsic rules. By reversing or removing key historical nodes, the model must deduce subsequent timeline evolution based on the quantitative rules provided in the text.

(5) Lagged Amplification of Minute Perturbations. Inspired by the bullwhip effect in supply chains, this dimension aims to test the model's ability to capture long-range lag propagation and asymmetric causal relationships.

TempoCausal-Bench devises rigorous construction logics and standardized question templates across five core dimensions, as presented in Table 6.

Table 6: Construction of the five-dimensional test for TempoCausal-Bench.

Dimension	Core Tested Capability	Standard Question Template
D1: Implicit Temporal Windows and Conditional Filtering	Guide the model away from explicit date matching; require it to delineate boundaries through event anchors and complete conditional filtering.	"During the period from [Event A's occurrence] to [Event B's occurrence], how many [specific entities/states] satisfy [Condition C]?"
D2: Concurrent Multi-Entity State Tracking	Break the single-entity linear narrative structure; synchronously track the states of multiple entities and identify logical conflicts.	"Throughout the entire period when [Entity X] is in [continuous State A], what complete [State B] transitions does [Entity Y] undergo?"
D3: Temporal Lags and Multi-Hop Dependencies	Test the model's ability to assemble scattered event fragments into a DAG, examining its topological reasoning level in reconstructing long-range causal chains across paragraphs.	"If the most critical precondition leading to [final Event Z] is [Event Y], then at the exact moment of [Event Y]'s occurrence (e.g., [time offset] prior), what action is [Entity X] performing?"
D4: Counterfactual Cascading Deduction	Examine the model's understanding of the text's intrinsic rules and require it to infer a counterfactual timeline based on those rules.	"Had [historical turning point Event A] not occurred, according to [the text's established rules/values], how would [Event B], originally scheduled at [Time T], have evolved?"
D5: Lagged Amplification of Minute Perturbations	Extract asymmetric causality and long-lag effects; identify early weak triggering signals.	"What was the most fundamental local triggering event that caused [Entity X] to make [a major structural change] at [late Time T_2]? From the first appearance of this weak signal to the eventual materialization of the macroscopic change, how long was the incubation period?"

E.2 Benchmark Construction Pipeline

The benchmark construction adopts a three-stage pipeline to ensure logical rigor and data reliability. Prompt examples are provided in Tables 7-9.

(1) Initial Generation Stage. The original Story from TimeQA-Hard is fed into an LLM, and combined with the five-dimensional specifications and construction logic, question-answer pairs corresponding to each dimension are generated in batches.

(2) Correctness Verification. GPT-5.4 and DeepSeek-V3.2 are used as verifiers to review the generated results, evaluating the correctness of answers and determining whether the questions are out of scope. When both LLMs simultaneously judge an answer as incorrect, GPT-5.4 is invoked to correct the answer. If both simultaneously judge the question as out of scope, the answer of that sample is marked as Unknown. If the two models disagree, the case is referred to human arbitration and corrected manually. According to statistics, the preliminary accuracy of the question-answer pairs constructed in this stage is 77.21%, and another 19.53% of the samples are marked as Unknown due to being out of scope.

(3) Sample Secondary Reconstruction. Samples marked as Unknown are reconstructed, with the model constrained to regenerate questions based solely on the events described in the original text. Subsequently, the newly generated samples are again passed through the aforementioned correctness verification process, and those that fail are manually corrected. After inspection, the construction accuracy of the question-answer pairs in this stage reaches 98.2%, with the remaining 1.8% of samples manually corrected, ultimately ensuring the reliability of the question-answer pairs.

Based on the above pipeline, we reconstruct 6,000 complex reasoning samples from 1,500 samples in a subset of TimeQA-Hard and split them into training and test sets at a ratio of 4:1. During dataset splitting, all variants generated from the same original sample are guaranteed to belong to the same set, thereby preventing data leakage at the source.

Table 7: Question-answer pair construction prompt example.

Stage 1: Question-Answer Pairs Construction Prompt Example

Please generate 5 complex questions based on the provided story, following the methods outlined below. Provide the standard answer for each question. Do NOT provide explanations.

Requirements for questions and answers:
- Use the original text content as much as possible for both questions and answers.
- The generated questions will be used to test other models (evaluated by EM and F1 metrics), so the answers MUST be concise. Do not ask questions that require long textual answers.

Construction Methods:
Dimension 1: Implicit Temporal Windows and Conditional Filtering
Core Testing Capability: Reject direct date matching. Require the model to define time boundaries via "event anchors" (Interval Bounding) and perform extraction and counting of specific entities or actions within those implicit boundaries.
Construction Steps:
Step 1: Select two non-simultaneous causal or sequential events from the text (Event A as the start, Event B as the end).
Step 2: Specify a state or entity type that changes within this interval.
Step 3: Add a distractor condition to be excluded (e.g., a person from the previous term, goods that were not successfully delivered).
Question Template: "Between [the occurrence of Event A] and [the occurrence of Event B], how many [specific entities/states] satisfy [Condition C]?"

Dimension 2: Concurrent Multi-Entity State Tracking
Core Testing Capability: Break away from single-entity linear tracking. Require the model to simultaneously track the independent states of two or more entities within the same time slice or interval, and find their intersections or logical conflicts.
Construction Steps:
Step 1: Identify two or more interacting entities in the text (e.g., a politician and their party, competitors, or a company and a regulator).
Step 2: Lock onto a persistent state of one entity (e.g., "during the period of being out of office" or "while holding a certain position").
Step 3: Ask what key state transition the other entity underwent during this period.
Question Template: "Throughout the entire period when [Entity X] was in [persistent state A], what complete [state B] transitions did [Entity Y] experience?"

Dimension 3: Temporal Lags and Multi-Hop Dependencies
Core Testing Capability: Test the model's ability to assemble independent events scattered across multiple paragraphs—with relative time offsets (e.g., "two years later," "subsequently," "upon the expiration of X's term")—into a complete directed acyclic graph (DAG).
Construction Steps:
Step 1: Locate an outcome event in the text.
Step 2: Trace backward to identify multiple preceding steps leading to that event, ensuring these steps are not adjacent in the text but described with time offsets.
Step 3: Ask about a hidden node somewhere in the middle of this chain.
Question Template: "If the most critical prerequisite leading to [final event Z] is [event Y], then exactly [time offset, e.g., two years earlier / the previous term] when [event Y] occurred, what action was [Entity X] performing?"

Dimension 4: Counterfactual Cascading Deduction
Core Testing Capability: Examine the model's understanding of the text's internal logical rules. Remove or invert an already occurred historical node, and require the model to deduce the collapsed timeline outcome based on rules provided in the text (e.g., electoral seat mathematics, eligibility rules for holding office).
Construction Steps:
Step 1: Find a historical turning point in the text that involves explicit quantitative rules or prior qualification constraints (e.g., a very close vote, a narrow win in a bid).
Step 2: Assume that this turning point did NOT happen (i.e., invert the variable).
Step 3: Ask how the immediate subsequent historical event on the counterfactual timeline would be rewritten, without making the question too complex.
Question Template: "If [historical turning point event A] had not happened, given [the established rules/numbers in the text], what would have happened to [event B], which originally occurred at [time T]?"

Table 7: Question-answer pair construction prompt example. (Continued)

Stage 1: Question-Answer Pairs Construction Prompt Example

Dimension 5: Lagged Amplification of Minute Perturbations
Core Testing Capability: Extract asymmetric causality and long temporal lags. The model must identify, within a grand narrative, a seemingly trivial local event early on that—after a long delay—becomes the root cause of a major global change.
Construction Steps:
Step 1: Find a major structural change at the end or later part of the text (e.g., complete withdrawal from politics, company bankruptcy, system overhaul).
Step 2: Locate the earliest trigger in the text that led to this outcome (e.g., a minor scandal involving a subordinate years ago, an early slight mistake).
Step 3: Establish this lagged causal link and require the model to calculate the time span from the emergence of the "minor disturbance" to the occurrence of the "global collapse."
Question Template: "When did the most fundamental local trigger that led [Entity X] to make [major structural change] at [late time T_2] occur? From the first appearance of that minor signal to the eventual macro-level change, how long was the incubation period?"

Story:
{Economic and Financial Committee (European Union). The Economic and Financial Committee (EFC) is an advisory body of the European Union, as defined by Article 134 of the Treaty on the Functioning of the European Union. Its president is also the president of the Eurogroup Working Group, which prepares dossiers for approval by the Eurogroup, whose decisions are generally ratified by ECOFIN. History: Before the third stage of the Economic and Monetary Union (EMU), which began on 1 January 1999, the Economic and Financial Committee was preceded by the Monetary Committee, which was established by Article 105 of the Treaty of Rome in 1957. In May, as part of the European Semester 2016 Spring Package, the committee was tasked with making reports on Belgium, Finland, and Italy's government debt. Agenda and composition of the EFC...}

Please return the output in JSON format only. The root must be a JSON object with a key "qa_list", and its value is a list of 5 objects. Each object should have the following keys:
- "dimension": The dimension number (e.g., "Dimension 1")
- "question": The generated question
- "answer": The concise standard answer

Table 8: Question-answer pair checking prompt example.

Stage 2: Question-Answer Pair Checking Prompt Example

Please evaluate the following question-answer pairs based on the provided story.
For each question, check if the provided answer is correct according to the story.
If the answer is incorrect, please correct it.
If the question is out of the scope of the story and cannot be answered, set the corrected answer to "Unknown".

Requirements for the corrected answers:
- The generated questions will be used to test other models (evaluated by EM and F1 metrics), so the answers MUST be concise. Do not ask questions that require long textual answers.

Story:
{Economic and Financial Committee (European Union). The Economic and Financial Committee (EFC) is an advisory body of the European Union, as defined by Article 134 of the Treaty on the Functioning of the European Union. Its president is also the president of the Eurogroup Working Group, which prepares dossiers for approval by the Eurogroup, whose decisions are generally ratified by ECOFIN. History: Before the third stage of the Economic and Monetary Union (EMU), which began on 1 January 1999, the Economic and Financial Committee was preceded by the Monetary Committee, which was established by Article 105 of the Treaty of Rome in 1957. In May, as part of the European Semester 2016 Spring Package, the committee was tasked with making reports on Belgium, Finland, and Italy's government debt. Agenda and composition of the EFC...}

QA Pairs:
{qa_json_str}

Table 8: Question-answer pair checking prompt example. (Continued)

Stage 2: Question-Answer Pair Checking Prompt Example

Please return the output in JSON format only. The root must be a JSON object with a key "checked_qa_list", and its value is a list of objects corresponding to the input QA pairs in the same order. Each object should have the following keys:
- "dimension": The original dimension
- "question": The original question
- "original_answer": The original answer
- "is_correct": true or false, whether the original answer was correct
- "answer": The concise standard answer (use the original if it was correct, the corrected one if incorrect, or "Unknown" if out of scope)

Table 9: Sample reconstruction prompt example.

Stage 3: Sample Reconstruction Prompt Example

You are a professional assistant for constructing and correcting datasets.
You need to strictly follow the specified dimensions and rules to rewrite the questions that cannot be answered (Unknown) based on the given original story (Story). Please ensure to return in the legal JSON format.

Dimension Construction Rules:
{dimension_rules}

Original Story:
{Economic and Financial Committee (European Union). The Economic and Financial Committee (EFC) is an advisory body of the European Union, as defined by Article 134 of the Treaty on the Functioning of the European Union. Its president is also the president of the Eurogroup Working Group, which prepares dossiers for approval by the Eurogroup, whose decisions are generally ratified by ECOFIN. History: Before the third stage of the Economic and Monetary Union (EMU), which began on 1 January 1999, the Economic and Financial Committee was preceded by the Monetary Committee, which was established by Article 105 of the Treaty of Rome in 1957. In May, as part of the European Semester 2016 Spring Package, the committee was tasked with making reports on Belgium, Finland, and Italy's government debt. Agenda and composition of the EFC...}

Pending Q&A Pairs:
{qa_json_str}

Task Requirements:
1. Carefully read the 5 pairs of questions and answers provided above.
2. For the pairs where the "answer" is "Unknown", explain that the original question cannot be answered within the story (beyond the scope of the text). Please, based on the original story and the corresponding dimension rules, rewrite a new and challenging question and provide a concise answer.
3. The rewritten question must be able to find a precise answer within the original text. It must strictly adhere to the "core testing ability" and "question template" in the corresponding dimension rules.
4. For the pairs where the "answer" is not "Unknown", please keep the original text unchanged.
Please return the result in a strict JSON format. The root node must be an array named "rewritten_qa_list", and this array must contain 5 objects, maintaining the same order as the input question-answer pairs.

The format of each object is as follows:
{
"dimension": "The name of the dimension, for example, Dimension 5",
"question": "The original question (if not modified) or the new question after rewriting",
"answer": "The original answer (if not modified) or the concise answer after rewriting",
"is_rewritten": true or false (if rewriting has been done, it is true; otherwise, it is false)
}

F TCR-Former Training Strategy

To equip the model with end-to-end reasoning capability from fragmented narratives to topological causal reconstruction while avoiding gradient conflicts and representation degradation commonly encountered in multi-task joint optimization, we propose a three-stage training strategy. During training, the optimization objectives for temporal perception, cross-modal alignment, and causal

reasoning are decoupled, enabling synergistic incremental improvement while ensuring stable convergence of each module. In the first stage, the temporal causal encoder is trained, with the base language model's parameters frozen and only the event anchor embeddings, time span biases, and the timestamp regression head updated. In the second stage, the cross-modal projector is trained, with all encoder and decoder parameters frozen and only the nonlinear projection layer optimized. In the third stage, end-to-end instruction fine-tuning is performed, with the encoder and projector frozen and the decoder base efficiently fine-tuned using LoRA. TCR-Former is trained on 4 × NVIDIA A100-40G GPUs.

F.1 First Stage Temporal Causal Encoder Pretraining

The training objective of the first stage is to establish the modeling capacity of the unified temporal causal encoder E_θ for implicit temporal dynamics and event structures. The encoder receives a sequence of original text chunks, extracts high-dimensional causal latent variables via learnable event anchors and time span biases, and predicts the physical temporal boundaries corresponding to each anchor. During training, all parameters of the base language model remain frozen, and only the event anchor embeddings, time bias parameters, and the timestamp regression head are updated, thereby reducing memory overhead and avoiding catastrophic forgetting. Three loss functions jointly provide the training signal.

(1) Timestamp Regression Loss. A lightweight regression head $h_\tau : \mathbb{R}^d \to \mathbb{R}$ is attached to the end of the encoder, mapping the hidden state $\mathbf{z}_k \in \mathbb{R}^d$ of the k-th query token to a normalized timestamp prediction $\hat{t}_k$. Given ground-truth time labels $\{t_k\}_{k=1}^M$, this loss applies mean squared error constraints as defined in Equation (10).

$$\mathcal{L}_{\text{time}} = \frac{1}{M} \sum_{k=1}^{M} \| h_\tau(\mathbf{z}_k) - t_k \|_2^2 \tag{10}$$

The timestamp regression loss drives the encoder to align event semantics with the physical timeline in the latent space, providing temporal priors for subsequent causal ordering.

(2) Semantic Reconstruction Loss. To prevent factual information decay during latent variable compression, we introduce a cosine similarity constraint, requiring the causal latent variables $\mathbf{Z}_{\text{causal}}$ to retain the global semantic kernel of the original text tokens. Denote the global average pooling representations of the causal latent variables and text tokens in the batch b as $\bar{\mathbf{z}}^{(b)}$ and $\bar{\mathbf{v}}^{(b)}$, respectively. The semantic reconstruction loss is then defined as Equation (11).

$$\mathcal{L}_{\text{sem}} = 1 - \frac{1}{B} \sum_{b=1}^{B} \frac{\langle \bar{\mathbf{z}}^{(b)}, \bar{\mathbf{v}}^{(b)} \rangle}{\| \bar{\mathbf{z}}^{(b)} \|_2 \cdot \| \bar{\mathbf{v}}^{(b)} \|_2} \tag{11}$$

The semantic reconstruction loss ensures that the causal latent variables carry not only topological structure information but also necessary factual semantics, laying a foundation for cross-modal alignment in the second stage training.

(3) Counterfactual Contrastive Loss. To enable the encoder to learn to distinguish genuine causal dependencies from pure temporal co-occurrences, we construct a contrastive learning objective based on topological interventions. During training, a subset of samples in a batch is selected with probability p_{cf} for intervention augmentation. For each selected sample, a proportion ρ_{cf} of nodes are randomly sampled from their non-empty event anchors as the intervention set k_ψ. The encoder computes in parallel the factual latent representation $\mathbf{Z}_{\text{factual}}$ and the counterfactual latent representation $\mathbf{Z}_{\text{cf}}$ after severing all attention connections of the nodes in k_ψ. These two representations are required to maintain a Euclidean distance of no less than a preset margin m in the feature space, explicitly modeling the representational shift induced by causal chain breakage. The computation is given in Equation (12).

$$\mathcal{L}_{\text{cf}} = \frac{1}{B} \sum_{b=1}^{B} \text{ReLU}\left(m - \| \mathbf{Z}_{\text{factual}}^{(b)} - \mathbf{Z}_{\text{cf}}^{(b)} \|_2 \right) \tag{12}$$

For unselected samples, $\mathcal{L}_{\text{cf}}$ is set to zero. To prevent intervention noise from disrupting early representation learning, a curriculum schedule is introduced. Specifically, $\mathcal{L}_{\text{cf}}$ is completely turned

off during the first 30% of training steps. For the 31%–70% steps, $p_{\text{cf}} = 0.1$ and $\rho_{\text{cf}} = 0.1$ are used, and for the final 30% of steps, $p_{\text{cf}} = 0.2$ and $\rho_{\text{cf}} = 0.2$ are set.

(4) First Stage Loss Function. The overall optimization objective of the first stage is a weighted sum of the three losses, as given in Equation (13).

$$\mathcal{L}_{\text{phase1}} = \alpha\mathcal{L}_{\text{time}} + \beta\mathcal{L}_{\text{sem}} + \gamma\mathcal{L}_{\text{cf}} \tag{13}$$

where α, β, γ are weighting coefficients. Considering the importance of each term, we set $\alpha = 1.0$, $\beta = 0.5$, and $\gamma = 0.2$.

F.2 Second Stage Cross-Modal Projection Alignment

The second stage focuses on training the nonlinear projector $\mathcal{P} : \mathbb{R}^{d_{\text{enc}}} \to \mathbb{R}^{d_{\text{dec}}}$, whose function is to map the causal latent variables $\mathbf{Z}_{\text{causal}}$ output by the encoder into the semantic space of the LLM decoder. In this stage, all parameters of both the encoder and decoder base models are frozen, and only the projector module is trained. The training follows the standard autoregressive language modeling objective, and a bypass-aware label masking strategy is introduced.

The decoder input sequence $\mathbf{E}_{\text{input}}$ is concatenated in the order of $[\mathbf{V}_{\text{text}}; \mathcal{P}(\mathbf{Z}_{\text{causal}}); \mathbf{E}_{\text{prompt}}]$. When computing the cross-entropy loss, supervision is only applied to the target response portion following the prompt text, and the conditional prefix composed of $\mathbf{V}_{\text{text}}$ and $\mathcal{P}(\mathbf{Z}_{\text{causal}})$ does not participate in the loss calculation. Formally, let T_{target} be the set of token indices of the target response after the prompt text. The language modeling loss is defined as Equation (14).

$$\mathcal{L}_{LM} = -\sum_{t \in T_{\text{target}}} \log p_\theta(y_t | \mathbf{E}_{\text{input}}, y_{<t}) \tag{14}$$

This masking strategy ensures that the projector focuses on transforming the causal latent variables into structured instructions interpretable by the LLM.

F.3 Third Stage End-to-End Instruction Tuning

The third stage performs end-to-end instruction tuning across the entire TCR-Former architecture, strengthening its reasoning capabilities for complex temporal question-answering and counterfactual deduction tasks. In this stage, all encoder and projector parameters are frozen, and low-rank adaptation (LoRA) [62] is used for parameter-efficient fine-tuning of the decoder base. The rank hyperparameter r of the low-rank adaptation matrices $\Delta\mathbf{W} = \mathbf{AB}$ is set to 64, and the scaling factor α is set to 128. The adaptation modules are applied to the attention projection layers and the linear layers in the feed-forward networks. This strategy efficiently injects temporal causal reasoning priors while preserving the base model's general language capabilities.

F.4 Training Dataset Construction

To support the three-stage training of TCR-Former, we constructed a structured training dataset based on multiple public temporal reasoning benchmarks and the TempoCausal-Bench.

F.4.1 First Stage Dataset

The first stage aims to train the encoder for temporal causal perception and structured modeling. We extracted and integrated 60,000 long-text narrative paragraphs from public datasets, including TGQA [4], TempReason [27], and TimeQA [8]. These texts cover multiple domains, including biographies, historical events, and organizational changes. The core of data construction lies in extracting event sequences that contain causal associations from the text along with their corresponding times. Specifically, for each text, we utilized DeepSeek-V3.2 to annotate explicit and implicit temporal and causal cues, forming a set of ordered event-timestamp pairs. Each event is summarized in a concise description and paired with the time point at which it occurs. The physical time is first uniformly converted to a floating-point number with years as the unit. For time points specified only with a year, the fractional part is zero. Subsequently, all timestamps are normalized to the [0, 1] interval, enabling the encoder's regression head to predict relative temporal positions stably. Through this process, each

training sample contains a sequence of original text chunks, a sequence of extracted causal event descriptions, and a corresponding sequence of normalized target timestamps. Table 10 provides a simplified example of a training sample.

Table 10: Examples of samples from the first stage dataset.

Examples of Samples From the First Stage Dataset

{
  "text_chunks": [
    "Jaroslav Pelikan ... was born on December 17, 1923, in Akron, Ohio ... He earned a seminary degree from Concordia Seminary in 1946 ... He completed his Ph.D. at the University of Chicago in 1953. Subsequently, in 1962, he joined the faculty at Yale University ..."
  ],
  "prompt_texts": [
    "Extract the chronological timeline of causal events."
  ],
  "target_events": [
    "Jaroslav Pelikan earned a degree from Concordia Seminary.",
    "Jaroslav Pelikan completes Ph.D. at the University of Chicago.",
    "Jaroslav Pelikan joins the faculty at Yale University."
  ],
  "target_timestamps": [0.35, 0.52, 0.78]
}

F.4.2 Second Stage Dataset

The second stage focuses on training the projector to align the causal latent variables with the LLM's semantic space. This dataset shares the same textual sources as the first stage but is constructed in a form that serves the autoregressive language modeling objective. From 40,000 texts, we used DeepSeek-V3.2 to organize further the causal event sequences generated in the first stage into a coherent natural-language passage. A training sample consists of the original text chunks, an instruction prompt that triggers the decoder's behavior, and the structured timeline text serving as the generation target. In this stage, the projector loss is applied only to the target part's token sequence, enabling the projector to learn to convert the fixed-length causal latent-variable sequence output by the encoder into a dynamic-length semantic representation that the decoder can parse. Table 11 provides a simplified example of a training sample.

Table 11: Examples of samples from the second stage dataset.

Examples of Samples From the Second Stage Dataset

{
  "text_chunks": [
    "Jaroslav Pelikan ... was born on December 17, 1923, in Akron, Ohio ... He earned a seminary degree from Concordia Seminary in 1946 ... He completed his Ph.D. at the University of Chicago in 1953. Subsequently, in 1962, he joined the faculty at Yale University ..."
  ],
  "prompt_texts": [
    "Please reconstruct the exact timeline of events based on the text."
  ],
  "target": "Jaroslav Pelikan earns his degree from Concordia Seminary around 1946.\nJaroslav Pelikan completes his Ph.D. at the University of Chicago around 1953.\nJaroslav Pelikan joins the faculty at Yale University around 1962."
}

F.4.3 Third Stage Dataset

The third stage dataset is used for end-to-end instruction tuning, aiming to finally unlock the model's reasoning capabilities for complex temporal causal question answering and counterfactual deduction. Based on the construction pipeline of the TempoCausal-Bench, we first utilized GPT-5.4 as the

base model. We ran the TCR-Agent framework on the TempoCausal-Bench training set to generate high-quality question-answer pairs. Each pair contains a complex causal reasoning question and its structured CoT answer, which strictly follows the tag sequence of <checking> → <reasoning> → <reflection>. Through this approach, we obtained 4,000 instruction-tuning samples that embody deep causal topology and counterfactual deduction logic, providing a learnable reasoning paradigm for the model.

G LLM-Acc

Since EM and F1 lack semantic judgment, we introduce a semantic-level accuracy metric, LLM-Acc. This metric employs two LLMs, DeepSeek-V3.2 and Gemini-3.1-flash, as independent evaluators. For a generated answer, each evaluator compares the predicted answer with the standard answer for semantic equivalence and gives a binary judgment of correct or incorrect. The final LLM-Acc is the average of the two evaluators' scores. During evaluation, the generation temperature of both evaluators is set to 0 to ensure determinism and reproducibility. The prompt template for the LLM is shown in Table 12.

Table 12: LLM judgment prompt example.

LLM Judgment Prompt Example

You are an expert evaluator for temporal causal reasoning tasks.

Your job is to judge whether a model's predicted answer is semantically equivalent to the ground-truth answer. Do not focus on superficial wording differences. Focus on the core facts, logical consistency, and causal implications. If the prediction conveys the same meaning as the ground truth, it should be considered correct.

Question: {question}
Ground-Truth Answer: {ground_truth}
Model's Response Answer: {model_response}

Based on the above, is the Response Answer semantically correct? Please first provide a brief reasoning for your decision, then output your final verdict. Use the following format exactly:

<reasoning>
[Your reasoning here]
</reasoning>
<verdict>
Correct or Incorrect
</verdict>

Fig. 4 illustrates the confusion matrix of LLM judgments based on samples from Qwen-3-8B, and we use Cohen's Kappa to measure the inter-judge agreement between the two LLMs. The average Kappa value across different methods is 0.826, indicating good agreement. To further verify the reliability of this automatic evaluation metric, we conducted a human review on 200 randomly sampled instances. The absolute deviation between human accuracy and LLM-Acc is 1.5%, validating that the automatic evaluation scheme using LLMs as evaluators can effectively replace human evaluation with sound robustness.

H Baseline Methods

H.1 TG-LLM

The CoT method in TG-LLM [4] aims to guide the LLM to perform rigorous temporal reasoning with the aid of explicit event context and external knowledge through a highly structured prompt design. As shown in Table 13, in its specific prompt construction, the model input first contains a natural-language timeline that clearly defines the start and end times of all relevant events in a list format. Next is the temporal logic question to be answered. To compensate for the LLM's deficiencies in mathematical and commonsense computation, this method explicitly introduces a Useful Information module in the prompt, directly injecting key external precomputed knowledge, including the chronological order of time points and the results of precise numerical difference

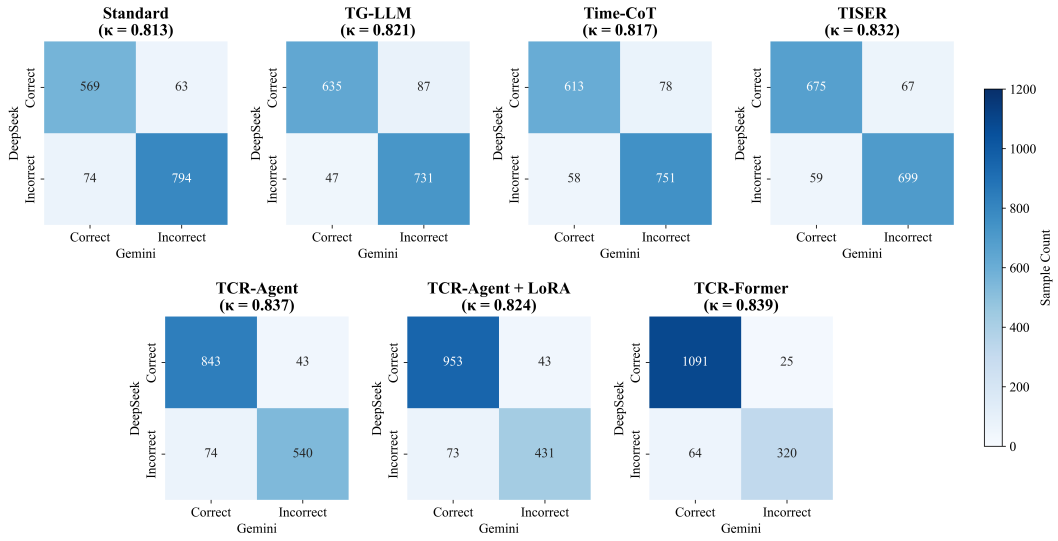


Figure 4: Confusion matrix of the LLM judgment (Qwen-3-8B).

calculations. After providing these enriched premises, the method employs "Let's think step by step" as a trigger statement to regulate the model's output of a complete reasoning path. Under this paradigm, the model can strictly follow a logical chain, first locating the specific time boundaries of the target event, then substituting the provided numerical values for span comparison, and finally deriving the conclusion based on the preceding explicit deduction steps.

Table 13: TG-LLM prompt example.

TG-LLM Prompt Example

Example:
Timeline:
(Liam Mitchell was born in Harrison) starts at 1885
(Maddox Reynolds was born in Glenville) starts at 1893
(Liam Mitchell was married to Emma Scott) starts at 1916
(Emma Scott was married to Liam Mitchell) starts at 1916
(Liam Mitchell was married to Emma Scott) ends at 1918
(Emma Scott was married to Liam Mitchell) ends at 1918
(Maddox Reynolds was married to Liam Mitchell) starts at 1922
(Liam Mitchell was married to Maddox Reynolds) starts at 1922
(Maddox Reynolds was married to Liam Mitchell) ends at 1928
(Liam Mitchell was married to Maddox Reynolds) ends at 1928
(Liam Mitchell died in Boston) starts at 1941
(Emma Scott died in Oceanview) starts at 1984

Question: True or false: Event (Liam Mitchell was married to Maddox Reynolds) was longer in duration than event (Emma Scott was married to Liam Mitchell)?

Useful information:
1885 before 1893 before 1916 before 1918 before 1922 before 1928 before 1941
$1918 - 1916 = 2$
$1928 - 1893 = 35$
$1928 - 1922 = 6$
$1941 - 1918 = 23$
$2 < 6 < 23 < 35$

Table 13: TG-LLM prompt example. (Continued)

TG-LLM Prompt Example

Answer: Let's think step by step.
To determine whether the statement is true or false, we need to compare the durations of (Liam Mitchell was married to Maddox Reynolds) and (Emma Scott was married to Liam Mitchell). The duration for each event can be calculated as follows:
(Liam Mitchell was married to Maddox Reynolds):
(Liam Mitchell was married to Maddox Reynolds) starts at 1922
(Liam Mitchell was married to Maddox Reynolds) ends at 1928
$1928 - 1922 = 6$
(Emma Scott was married to Liam Mitchell):
(Emma Scott was married to Liam Mitchell) starts at 1916
(Emma Scott was married to Liam Mitchell) ends at 1918
$1918 - 1916 = 2$
Now, we compare the duration of the two events: 6 is greater than 2; thus, the answer is True.

Test:
Timeline:
(Chris Evans was born in Bristol, Connecticut) starts at 1948
(Oliver Jensen was born in Bristol, Connecticut) starts at 1948
(Chris Evans was married to Oliver Jensen) starts at 1970
(Oliver Jensen was married to Chris Evans) starts at 1970
(Chris Evans created Pearl Network) starts at 2005
(Chris Evans won prize Victory Achievement Award) starts at 2007
(Chris Evans was married to Oliver Jensen) ends at 2010
(Oliver Jensen was married to Chris Evans) ends at 2010
(Chris Evans created Pearl Network) ends at 2013

Question: True or false: Event (Chris Evans owned Pearl Network) was longer in duration than event (Oliver Jensen was married to Chris Evans)?

Useful information:
1948 before 1970 before 2005 before 2007 before 2010
$1970 - 1948 = 22$
$2007 - 1948 = 59$
$2007 - 2005 = 2$
$2010 - 1970 = 40$
$2 < 22 < 40 < 59$

Answer: Let's think step by step.

H.2 Time-CoT

Time-CoT [14] is a CoT method designed to enhance the ability of LLMs to handle factual question answering involving temporal reasoning. As shown in Table 14, Time-CoT guides the model to perform step-by-step reasoning through a structured prompt template. First, the prompt requires the model to detail and describe the Standard Question Entity along the timeline dimension, and then focus on the specific Time Parameter for reasoning and matching. For example, when dealing with queries such as "What movie did Zoe Cassavetes direct in 2007 ?" or "Who was the quarterback of the New York Giants as of December 2010 ?", Time-CoT forces the LLM first to construct the historical timeline of the entity, and then focus on the specific target time point to derive the exact final answer based on the intersection of the timeline.

Table 14: Time-CoT prompt example.

Time-CoT Prompt Example

Response Template:
Let's think step by step. First, we need to find out the <Standard Question Entity> according to the timeline. Then, we need to focus on the time <Explicit Time Parameter>. So we can obtain the answer is <Gold Answer>. Therefore, the answer(s) is/are ...

Table 14: Time-CoT prompt example. (Continued)

Time-CoT Prompt Example

Example:
Question: Which movie did Zoe Cassavetes direct in 2007 ?
Answer: Let's think step by step. First, we need to find out which movie did Zoe Cassavetes direct according to timeline. 2000: Men Make Women Crazy Theory, 2007: Broken English, 2012: The Powder Room, 2015: Day Out of Days..... Then, we need to focus on the time in 2007. So we can obtain the answer in Broken English. Therefore, the answer(s) is/are Broken English.

Question: Who is the quarterback for the New York Giants as of December 2010 ?
Answer: Let's think step by step. First, we need to find out who is the quarterback for the New York giants according to timeline. 1999-2003: Kerry Collins, 2003: Jesse Palmer, 2004: Kurt Warner, 2004-2019: Eli Manning, 2017: Geno Smith, 2019-2023: Daniel Jones, 2020: Colt McCoy, etc. Then, we need to focus on the time December 2010. So we can obtain the answer is Eli Manning. Therefore, the answer(s) is/are Eli Manning.

Test:
Question: Who was the president of the US when Adele Faccio was a teenager?
Answer: Let's think step by step.

H.3 TISER

TISER [15] is a framework designed to enhance the temporal reasoning capabilities of large language models. It extends the traditional CoT into a four-stage cyclic inference paradigm comprising preliminary reasoning, timeline construction, reflection and verification, and answer generation, by scaling reasoning traces at test time. As shown in Table 15, the specific workflow of TISER is as follows:

(1) Stage 1 – Initial Reasoning. The model first generates a preliminary, raw reasoning trajectory based on the provided question and temporal context.

(2) Stage 2 – Timeline Construction. Relevant temporal events are extracted from the context and the initial reasoning trajectory and organized into an explicit, ordered timeline.

(3) Stage 3 – Reflection and Verification. The model compares its initial reasoning trajectory with the newly constructed timeline to self-verify its logic. This stage is responsible for detecting and correcting inconsistencies, ambiguities, or omissions in the initial reasoning, thereby generating a refined reasoning trajectory.

(4) Stage 4 – Answer Generation. Relying on the error-corrected reasoning trajectory and the structured timeline, the model ultimately generates a concise, accurate, and logically coherent answer.

Table 15: TISER prompt example.

TISER Prompt Example

You are an AI assistant that uses a Chain of Thought (CoT) approach with reflection to answer queries. Follow these steps:
1. Reason through the problem step by step within the <reasoning> tags.
2. Given your previous reasoning, identify relevant temporal events in the given context for answering the given question within <timeline> tags. Assume relations in the context are unidirectional.
3. Reflect on your reasoning and the timeline to check for any errors or improvements within the <reflection> tags.
4. Make any necessary adjustments based on your reflection. If additional reasoning is required, go back to Step 1 (reason through the problem step by step); otherwise, move to the next step (Step 5).
5. Provide your final, concise answer within the <answer> tags. If the answer is a number, just output the number, nothing else. Otherwise, output the entity or event without any additional comments.

Additional Instructions:
1. [IF during data generation] Perform your reasoning knowing that the answer is: X.
2. The <reasoning>, <reflection>, and <timeline> sections are for your internal reasoning process. All the reflection and the timeline have to be contained inside the thinking section.
3. Do not use enumerations or lists when writing; use plain text such as paragraphs.
4. The response to the query must be entirely contained within the <answer> tags.

Table 15: TISER prompt example. (Continued)

TISER Prompt Example

Response Format:
<reasoning>
[Your step-by-step reasoning goes here. This is your internal thought process.]
<timeline> [Relevant temporal events for answering the given question.] </timeline>
<reflection> [Your reflection on your reasoning, checking for errors or changes required.] </reflection>
[Any adjustments to your thinking based on your reflection.]
</reasoning>
<answer> [Your final, concise answer to the query.] </answer>

I Ablation Study Setup

To systematically analyze the actual contributions of each design element in TCR-Agent and TCR-Former, we conduct component-level ablation on the two frameworks on the TempoCausal-Bench.

I.1 TCR-Agent Ablation Setup

The complete TCR-Agent pipeline based on GPT-5.4 contains three key stages: causal graph extraction, counterfactual intervention, and dual-track reasoning. By removing critical components from each stage individually, we obtain the following ablation variants.

w/o Scratchpad: During the causal graph extraction stage, the LLM is required to directly output the raw graph G_{raw} with nodes and edges, without generating intermediate reasoning steps. The instruction regarding the scratchpad is removed from the extraction prompt, and the content of the scratchpad S is completely excluded from subsequent reasoning.

w/o DAG Instantiation: The structured DAG instantiation step is skipped. The extracted G_{raw} is directly serialized into a textual description as a list of natural language edges, without performing multi-dimensional attribute projection, timestamp consistency verification, or topological sorting with the directed acyclic constraint, thereby degenerating into a flat relational chain.

w/o Counterfactual Intervention: When processing counterfactual-type queries, the graph intervention engine is deactivated, and the original causal graph G_{raw} is used for reasoning, meaning no intervention operation is performed on the causal structure.

w/o Dual-Track Reasoning: The dual-track reasoning is downgraded to a single-track mode. Only the serialized description of the structured causal graph $D(G')$ is provided to the LLM as context, while the original narrative text is excluded, thereby isolating the auxiliary effect of unstructured factual details on structured logical deduction.

Standard: All external symbolic modules of TCR-Agent are completely removed, and only the original text and question are directly fed to GPT-5.4, which is required to output an answer, serving as the upper-bound reference for the pure language model baseline.

I.2 TCR-Former Ablation Setup

The complete TCR-Former architecture introduces core designs, including the causal-topological attention mask, explicit time-span bias, and fact-logic bypass fusion, on top of the Qwen-3-8B decoder. The role of each mechanism is examined through the following structural degradations.

w/o Causal-Topological Attention Mask: The causal-topological attention mask is replaced with the standard strictly lower-triangular causal mask.

w/o Time Span Bias: The learnable time span bias embeddings are frozen and removed, causing the initial representation of query tokens to degrade to relying solely on the position-independent base anchor embeddings. Meanwhile, the time bias matrix is no longer added during attention computation.

w/o Fact-Logic Bypass Fusion: At the decoder input, the word embedding track $\mathbf{E}_{\text{text}}$ of the original text is removed, and only the aligned causal latent variables $\mathbf{Z}_{\text{aligned}}$ from the encoder and

the instruction prompt embedding $\mathbf{E}_{\text{prompt}}$ are concatenated and fed into the decoder. This setup forces the generation process to rely entirely on the topology representation compressed in the latent space to examine the degree of factual information retention during compression.

Only Decoder: The encoder and the projection layer are removed, retaining only the native Qwen-3-8B. Since it cannot execute the first two training stages, we perform instruction tuning on it using the high-quality causal question-answering data generated in the third stage.

Only Encoder + Decoder: Only Qwen-3-0.6B is adopted as the encoder and Qwen-3-8B as the decoder, with the projection layer between the two models, undergoing the full three-stage training pipeline of TCR-Former.

Standard: Qwen-3-8B directly performs reasoning on the original text and question, serving as the performance reference for the pure backbone model.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitation of our work in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper has some theoretical analysis. We provide a complete and correct analysis.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer to Experiments Section and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have made our code and dataset publicly available, along with comprehensive instructions to enable easy reproduction of the experimental results reported in our paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to Experiments Section and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Please refer to Experiments Section and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Experiments Section, Analysis Section and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: When applied as intended and functioning correctly, our methods do not pose any negative social impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all the papers associated with the datasets and codes used in our work. And we have obtained the license for their use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have released the code and the detailed documentation to GitHub.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.