Timeline Self-Reflection: Enhancing Temporal Reasoning in Large Language Models

Xueyufei Zhang *s336472* | Dong Shi *s339024* | Xiangxi Li *s336719* | Chunyi Li *s338968* | Yuanzheng Ji *s344577*

GitHub Repository: https://github.com/xueyufeizhang/DNLP-TemporalReasoning

***Abstract*—Large language models (LLMs) perform well on factual question answering, yet they often fail when temporal constraints are essential, such as event ordering, interval overlap, and time-point validity. These errors frequently arise not from missing knowledge but from unstable handling of temporal boundaries and brittle output formats that complicate answer extraction. To address this, we reproduce and extend TISER (Timeline Self-Reflection), a structured generation framework that decomposes outputs into `<reasoning>`, `<timeline>`, `<reflection>`, and `<answer>` to improve controllability and reliability in temporal reasoning.**

Methodologically, we fine-tune two open-source instruction models, Qwen2.5-7B and Mistral-7B-v0.3, using LoRA-based supervised fine-tuning on TISER-formatted targets. We then evaluate both off-the-shelf and fine-tuned models on five in-domain benchmarks (TGQA, TempReason L2/L3, and TimeQA easy/hard) under Standard prompting and TISER prompting. Predictions are deterministically extracted from the final `<answer>` span and scored using Exact Match (EM) and token-level F1, with unweighted macro-averages.

We propose two inference-time extensions. First, a conservative Self-Consistency (SC) variant samples $k = 5$ candidates (temperature 0.6, top-p 0.9) and applies majority voting after string normalization, reverting to greedy decoding when vote confidence falls below $\tau = 0.6$. Second, we introduce a Timeline-Reflection mechanism with a Sys2 validator that detects malformed timelines and triggers a single regeneration pass only when needed, while logging trigger rates.

Results show that TISER prompting substantially improves LoRA-fine-tuned models by increasing format adherence and answer extractability, yielding large gains over Standard prompting across benchmarks. SC provides modest additional improvements but increases inference cost by about $6\times$, whereas Sys2-triggered regeneration offers targeted robustness gains by correcting structural timeline failures.

***Index Terms*—temporal reasoning, large language models, supervised fine-tuning, LoRA, self-reflection, self-consistency**

I. Problem statement

Large language models can be strong at factual QA, yet they often fail on questions where time boundaries matter (ordering, overlap/inclusion, or "what was true at year X"). The typical failure is not missing knowledge but mis-handling temporal constraints: misaligned start/end dates, treating the same year as the same phase, or drawing conclusions outside the valid time window. TISER addresses this by making the reasoning process explicit—first constructing a timeline, then performing a self-check (reflection), and only then producing the final answer—so temporal reasoning becomes more reliable and the generation is easier to audit and control.

In this project, we reproduce and extend the TISER paper by fine-tuning two open-source instruction models using LoRA to induce TISER-style structured generation, and evaluating them under the same benchmark protocol used in the paper. Specifically, we fine-tune Qwen2.5-7B-Instruct and Mistral-7B-Instruct-v0.3 on the TISER dataset, aiming to preserve general capabilities while improving temporal reasoning reliability and strict adherence to the required output template.

Given a temporal reasoning instance (c, q, a), where c denotes a context containing events and their temporal information (text evidence or textualized structured events), q is a query about c (e.g., ordering, overlap/inclusion, time-point queries), and a is the gold answer, a model M generates an output $y = M(c, q)$

The training data follow the TISER construction procedure: starting from standard temporal reasoning benchmarks (c, q, a), an external LLM generates intermediate structures (reasoning/timeline/reflection), and a filtering step keeps only samples whose regenerated answer matches the gold label, controlling noise and encouraging the stable behavior: *build timeline → reflect/check → answer.*

II. Methodology

A. Methodology: Pipeline Overview and Modules

a) Overview of the NLP pipeline/architecture: Our system performs temporal question answering with a structured prompting-and-decoding pipeline. Given an input instance, we (1) load the example and construct the input prompt (Standard or TISER), (2) run an instruction-tuned LLM (off-the-shelf or LoRA-fine-tuned) to generate a structured response, (3) optionally apply inference-time extensions (Self-Consistency voting or Sys2-triggered regeneration) to improve robustness, (4) deterministically extract the final answer from the generated text, and (5) normalize and evaluate predictions using Exact Match (EM) and token-level F1, reporting per-dataset scores and macro-averages over the selected subsets.

b) Module descriptions.: **(1) Data loader & prompt builder.** We load each example from JSON/JSONL and build the input using dataset-specific fields and a dataset tag. The module supports two prompting strategies: Standard prompting and TISER prompting, where TISER enforces a structured output template.

(2) Model backbone (base vs. LoRA fine-tuned). We use an instruction-tuned LLM as the backbone. Be-

sides off-the-shelf models, we evaluate LoRA SFT models trained on TISER-formatted targets that explicitly separate <reasoning>, <timeline>, <reflection>, and <answer>.

(3) Decoder / inference engine. We perform batched decoding under a fixed configuration. The default setting is greedy decoding (Greedy-TISER). For extensions, the inference engine can additionally run stochastic sampling (for Self-Consistency) or trigger a second-pass regeneration when validation fails.

(4) Extension A: Self-Consistency (SC) aggregator. SC generates multiple sampled outputs, normalizes candidate answers at the string level, groups them by normalized form, and selects the majority vote. A confidence threshold controls a fallback to the greedy answer when sampled answers disagree.

(5) Extension B: Sys2 timeline validator + targeted regeneration. This module validates whether the generated timeline is structurally well-formed. If validation fails, it performs one additional regeneration pass under the same prompt format and uses the regenerated output as the final response. It also logs how often regeneration is triggered as a diagnostic.

(6) Answer extractor. We deterministically extract the final answer from the generated text, preferring the last <answer>... </answer> span when present; otherwise, we fall back to simple heuristics such as boolean matching or taking the last line.

(7) Normalizer & evaluator. We apply consistent normalization to predictions and references (case, punctuation, articles, whitespace, and minor domain rules) and compute EM and token-level F1. We report per-subset results and macro-averages over the selected evaluation subsets.

III. Experiments

A. Model fine-tuning (LoRA SFT)

a) Training data and objective.: We fine-tune two open-source instruction models, **Qwen2.5-7B** and **Mistral-7B-v0.3**, using supervised fine-tuning with LoRA on the TISER-formatted training set. Each training sample pairs an instruction prompt with a structured target that explicitly separates <reasoning>, <timeline>, <reflection>, and <answer>. We optimize the standard causal language modeling loss to encourage both temporal reasoning reliability and strict adherence to the required output template.

b) Implementation details.: Qwen2.5-7B and Mistral-7B-v0.3 are loaded in full precision during training; in both cases, only LoRA parameters are updated. We track training loss and mean token accuracy to monitor convergence.

c) Training dynamics.: Fig. 1 shows the training curves for the Qwen2.5-7B-TISER run. The loss drops sharply at the beginning and then stabilizes, while mean token accuracy increases rapidly and plateaus around $\sim$0.96 by the end of training. This suggests the model has learned the structured TISER output format and can reproduce it consistently at the token level.

B. Reproduction

a) Data description: We evaluate five in-domain temporal QA test sets: TGQA, TempReason (L2/L3), and TimeQA (easy/hard). Each example is loaded from JSON/JSONL with a dataset tag (dataset_name), a reference answer, and fields used to build the input prompt; results are reported per dataset and as an unweighted macro-average.

b) Experimental design: We compare two inference prompting strategies (*Standard* vs. *TISER*) for both off-the-shelf and fine-tuned (LoRA) models under greedy decoding. Batched decoding is applied only to the Mistral model under TISER prompting for efficiency considerations; all other model–prompt combinations are processed sequentially. Hardware/software are standard: single-GPU when available, with PyTorch + Transformers (PEFT).

c) Validation method: For each example, we generate an output and deterministically extract the final answer (prefer the last <answer>...</answer> span; otherwise fall back to boolean match or the last line). Predictions and references are normalized (case/punctuation/articles/whitespace and minor domain-specific rules) before scoring; malformed or missing-answer items are skipped.

d) Performance metrics: We report Exact Match (EM) and token-level F1, plus the macro-average over the five datasets.

e) Results and analysis: As Table I shows, TISER yields a strong gain for the fine-tuned Qwen model, boosting the macro-average from 35.3/44.9 (EM/F1) under Standard to 92.5/94.1 under TISER. The improvement is consistent across all five evaluation splits and is particularly pronounced on harder temporal reasoning settings (e.g., TempReason (L2/L3) and TimeQA (hard)), suggesting that the fine-tuned model has learned to follow TISER-style output constraints. This makes answer extraction more reliable and reduces formatting-induced errors.

For Qwen-Base, the effect of TISER is mixed: it substantially improves TimeQA (easy) (EM/F1 8.6/34.0 $\rightarrow$ 36.2/37.8), but degrades TGQA and TempReason (e.g., TempReason (L2) EM/F1 16.2/30.2 $\rightarrow$ 7.4/8.6). This pattern is consistent with a prompt–model mismatch: without explicit training to produce the TISER format, the base model may overemphasize structural constraints, leading to brittle outputs or incomplete answers on datasets requiring more free-form temporal reasoning.

For Mistral-Finetuned, Standard inference achieves a macro-average of 69.2/72.9, while the TISER row is higher at 85.0/87.0, with noticeable gains on TGQA and TimeQA. Overall, the table supports the conclusion that the best performance occurs when the inference prompting strategy matches the output format the model was optimized to follow, and that TISER is most effective when the model has been trained or adapted to produce TISER-compatible answers.

C. Extension: Self-Consistency

a) Data description: We evaluate the same in-domain temporal QA benchmarks as in the baseline (TGQA, Tem-

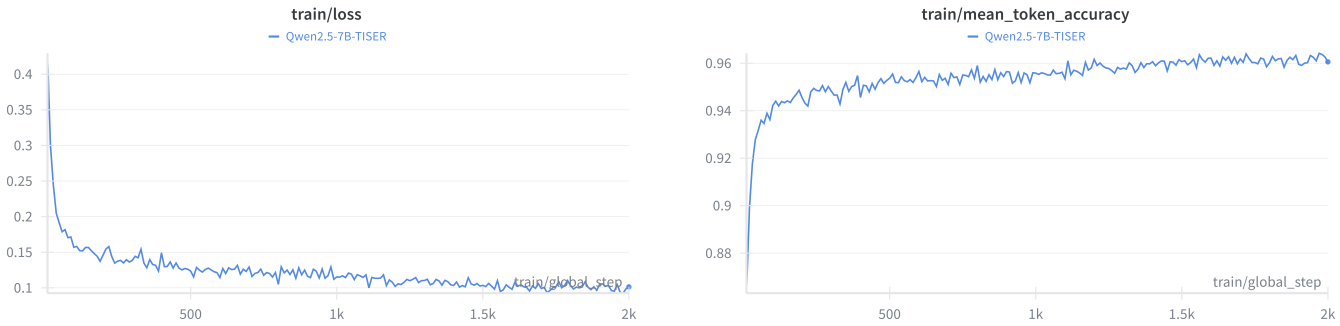

Fig. 1. LoRA SFT training curves for Qwen2.5-7B-TISER: (left) training loss, (right) mean token accuracy.

TABLE I
REPRODUCTION

Model	Inference	TGQA		TempReason (L2)		TempReason (L3)		TimeQA(easy)		TimeQA(hard)		Macro Avg.	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Open LLMs - Off-the-Shelf													
Qwen2.5-7B	Standard	3.8	21.6	16.2	30.2	14.4	31.4	8.6	34.0	2.4	19.5	9.1	27.4
Qwen2.5-7B	TISER	8.2	8.5	7.4	8.6	9.4	10.9	36.2	37.8	8.8	9.6	14.0	15.1
Open LLMs - Fine-tuned													
Mistral-7B	Standard	42.0	55.3	74.0	75.7	60.0	61.4	85.0	86.9	85.0	85.0	69.2	72.9
Qwen2.5-7B	Standard	16.4	26.3	26.9	35.0	38.0	43.6	44.0	61.1	51.0	58.9	35.3	44.9
Mistral-7B	TISER	70.8	80.6	84.2	83.2	88.2	87.1	92.8	94.0	89.0	89.9	85.0	87.0
Qwen2.5-7B	TISER	**76.1**	**81.1**	**92.0**	**93.7**	**97.2**	**97.9**	**99.0**	**99.6**	**98.0**	**98.0**	**92.5**	**94.1**

TABLE II
TISER + SELF-CONSISTENCY

Model	Inference	TGQA		TimeQA(easy)		TimeQA(hard)		Macro Avg.	
		EM	F1	EM	F1	EM	F1	EM	F1
Mistral + LoRA	TISER	70.8	80.6	92.8	94.0	89.0	89.9	84.2	88.2
Mistral + LoRA	TISER + Self-Consistency	73.2	82.4	94.6	95.6	92.6	93.5	86.8	90.5

pReason L2/L3, and TimeQA easy/hard) using identical test splits and the same TISER prompt template. For the self-consistency (SC) extension, we report results on the same comparison subset used in Table1 for Mistral, focusing on TGQA and TimeQA (easy/hard); the macro-average is computed over these three subsets only.

b) Experimental design: We start from the Greedy-TISER baseline and add an inference-time SC module without changing the prompt format or the output schema. For each input, we first run greedy decoding to produce a fallback answer. We then sample $k = 5$ additional completions using temperature 0.6 and top-p 0.9. Candidate answers are normalized at the string level (lowercasing, removing punctuation and articles, and applying a simple year-to-number normalization), grouped by their normalized form, and aggregated by majority vote. If the winning class has vote share below a confidence threshold $\tau = 0.6$, we return the greedy answer; otherwise, we return the shortest raw string within the top-vote class. This conservative design aims to reduce variance on difficult temporal reasoning cases while preventing regressions when sampled answers diverge.

c) Validation method: We keep the same answer extraction and normalization pipeline as the baseline. The only difference is that the final answer can come from either (i) greedy decoding (fallback), or (ii) the SC majority-vote result when the confidence threshold is satisfied.

d) Performance metrics: We report Exact Match (EM) and token-level F1 for each evaluated subset, and an unweighted macro-average over TGQA, TimeQA (easy), and TimeQA (hard) for this extension setting.

e) Execution times: On the evaluated subset, Greedy-TISER takes 5687.11 seconds, while SC takes 34911.63 seconds, i.e., approximately $6.1\times$ slower due to multiple sampled generations per input.

TABLE III
PERFORMANCE COMPARISON: BASELINE VS. SYSTEM 2.

Model	Inference	TGQA			TempReason (L2)			Macro Avg.	
		EM	F1	Sys2	EM	F1	Sys2	EM	F1
Mistral + LoRA	TISER	70.8	80.6	-	84.2	83.2	-	77.5	81.9
Mistral + LoRA	TISER + Sys2 Timeline Reflection	80.6	87.3	182/500	88.0	90.1	500/500	84.3	88.7

f) Results and analysis: As Table II shows, Greedy-TISER is already strong because the fine-tuned model produces structured `<reasoning>`, `<timeline>`, `<reflection>`, and `<answer>` in a single pass, which provides an internal consistency constraint within one decoding trajectory. Consequently, SC yields only modest but consistent gains, increasing the macro-average by +2.6 EM and +2.3 F1 points, consistent with its role as a variance-reduction mechanism rather than introducing new modeling capacity. Overall, these gains align with SC behaving primarily as variance reduction rather than adding new capability: when the baseline EM is already high (e.g., 92.8% and 89.0% EM on TimeQA easy/hard), the remaining headroom is limited. The τ-based fallback is important in this regime, as it makes the method more conservative and prevents degradations when sampled answers are inconsistent. From a cost–benefit perspective, the improvements come at a substantial computational cost: inference time increases by approximately 6.1× due to multiple sampled generations per input. Given the already strong Greedy-TISER baseline, the marginal gains from SC must be weighed against this overhead. In latency- or resource-constrained settings, Greedy-TISER may offer a more favorable efficiency–accuracy trade-off. Conversely, SC is better suited to robustness-sensitive or high-stakes scenarios, where small but consistent accuracy gains justify additional computation. Overall, SC functions as a robustness-enhancing but compute-intensive extension.

D. Extension: System 2 Logic Calibration

a) Data Description: We evaluate the same in-domain temporal QA benchmarks as the baseline. For this extension, we report results on the specific subsets shown in Table III: TGQA and TempReason (L2). We also compute the unweighted macro-average over these two subsets. Each instance follows the TISER prompt format and uses the same test splits as the baseline.

b) Experimental Design: Building upon the Greedy-TISER baseline, we introduce a *Timeline-Reflection* module equipped with a System 2 (Sys2) validator. Unlike unconditional self-consistency which blindly resamples, this approach uses a verify-and-correct mechanism. For each input, we first generate a response using greedy decoding. A rule-based validator then inspects the output for two types of validity: (1) **Structural Integrity** (presence of valid `<timeline>` tags and parsing format), and (2) **Logical Consistency** (e.g., ensuring event end times do not precede start times). If an issue is detected, the system appends an intervention prompt (e.g., "[System Alert]: ... Step 1: Extract the events...") to trigger a targeted regeneration. We log the frequency of these interventions as the *Sys2 trigger count*.

c) Performance Metrics: We report Exact Match (EM) and token-level F1 per subset, alongside the macro-average. The answer normalization pipeline remains identical to the baseline. The final answer is extracted from the original output if it passes validation, or from the regenerated output if the Sys2 mechanism is triggered.

d) Results and Analysis: As shown in Table III, the System 2 logic calibration significantly improves performance over the Mistral-Finetuned TISER baseline. On TGQA, EM/F1 increases from 70.8/80.6 to 80.6/87.3, and on TempReason (L2), it rises from 84.2/83.2 to 88.0/90.1. Consequently, the macro-average improves from 77.5/81.9 to 84.3/88.7.

The trigger rates reveal the mechanism's distinct behavior across datasets. On TGQA, regeneration is triggered in 182/500 cases, acting as a selective repair mechanism. In contrast, on TempReason (L2), the mechanism is triggered for all instances. This indicates that the baseline model consistently failed to produce valid timeline structures or logical temporal flows for this dataset in the zero-shot pass. The System 2 intervention effectively forced the model to explicitly reason via timeline construction, thereby correcting the reasoning path. Overall, results suggest that conditional, logic-driven regeneration is a highly effective inference-time strategy, offering robustness superior to simple stochastic sampling.

IV. CONCLUSION

We reproduced and extended TISER for structured temporal reasoning in large language models. LoRA-based fine-tuning on TISER-formatted targets substantially improves answer reliability when inference prompting matches the learned output schema. While Self-Consistency offers modest gains through variance reduction at higher computational cost, Sys2-triggered timeline validation provides more targeted robustness improvements via selective regeneration. These results underscore the importance of structure-aware alignment between training and inference for reliable temporal reasoning.

REFERENCES

[1] 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 (ACL 2025)*, pages 28014–28033, 2025.

[2] 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 (ACL 2023)*, pages 14820–14835, 2023.
[3] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations (ICLR)*, 2023.
[4] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
[5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2022.

APPENDIX

SYSTEM 2 ALGORITHM

Algorithm 1 System 2 Logic Calibration Inference

Require: Question Q, Context C, LLM $\mathcal{M}$
Ensure: Final answer string A_{final}

$P \leftarrow \text{BUILDPROMPT}(Q, C)$
$R_{init} \leftarrow \mathcal{M}(P)$
$is\_valid, error\_msg \leftarrow \text{TRUE}, \text{None}$
$Events \leftarrow \text{PARSETIMELINE}(R_{init})$
if $Events = \emptyset$ **or** $\text{PARSEERROR}(R_{init})$ **then**
    $is\_valid \leftarrow \text{FALSE}$
    $error\_msg \leftarrow$ Missing or malformed timeline tags.
else
    for each event $e \in Events$ **do**
        if e.end $<$ e.start **then**
            $is\_valid \leftarrow \text{FALSE}$
            $error\_msg \leftarrow$ Event ends before it starts.
            break
        end if
    end for
end if
if not is_valid **then**
    $P_{fix} \leftarrow P \oplus R_{init} \oplus error\_msg$
    $R_{final} \leftarrow \mathcal{M}(P_{fix})$
else
    $R_{final} \leftarrow R_{init}$
end if
$A_{final} \leftarrow \text{EXTRACTANSWER}(R_{final})$
return A_{final}