
Enhancing LLM Reasoning for Time Series Classification by Tailored Thinking and Fused Decision

Anonymous Author(s)

Affiliation

Address

email

Abstract

The reasoning capabilities of large language models (LLMs) have significantly advanced their performance by enabling in-depth understanding of diverse tasks. With growing interest in applying LLMs to the time series domain, this has proven nontrivial, as evidenced by the limited efficacy of straightforwardly adapting text-domain reasoning techniques. Although recent work has shown promise in several time series tasks, further leveraging advancements in LLM reasoning remains under-explored for time series classification (TSC) tasks, despite their prevalence and significance in many real-world applications. In this paper, we propose ReasonTSC, a novel framework designed to effectively leverage LLM reasoning for time series classification through both a multi-turn reasoning and a fused decision-making strategy tailored to TSC. Rather than straightforwardly applying existing reasoning techniques or relying solely on LLMs' built-in reasoning capabilities, ReasonTSC first steers the model to think over the essential characteristics of time series data. Next, it integrates predictions and confidence scores from plug-in classifiers, e.g., domain-specific time series models, as in-context examples. Finally, ReasonTSC guides the LLM through a structured reasoning process: it evaluates the initial assessment, backtracks to consider alternative hypotheses, and compares their merits before arriving at a final classification. Extensive experiments and systematic ablation studies demonstrate that ReasonTSC consistently outperforms both existing time series reasoning baselines and plug-in models, and is even capable of identifying and correcting plug-in models' false predictions. The code for ReasonTSC is available at <https://anonymous.4open.science/r/ReasonTSC-B737>.

1 Introduction

Time series classification (TSC) is a fundamental task with wide applications across diverse areas, including healthcare [1–3], finance [4, 5], speech recognition [6], and so on [7, 8]. The astounding performance of large language models (LLMs), especially boosted by recent advancements in their reasoning capabilities as epitomized by ChatGPT-o1 [9, 10], Deepseek-R1 [11], Gemini-2.5-Pro [12, 13], has sparked surging demand for leveraging them in domains well beyond the pure natural language processing (NLP) domain. The time series (TS) domain is no exception to such fevered explorations, with existing research promisingly discovering that LLMs have the capability to understand essential TS data characteristics, such as trend, cyclic behavior, stationarity, amplitude, rate of change, and outlier [14, 15]. Consequently, a variety of methods have been proposed to exploit LLMs for TS tasks [16–19], with a predominant focus on forecasting tasks that align more naturally with the autoregressive generation behavior of LLMs [20–23]. There are also efforts exploring LLMs for anomaly detection [24, 21, 25], imputation [26–28], and nascent but growing attempts at classification [29–31].

Propelled by the promise that advanced reasoning techniques can provide enhanced performance through in-depth understanding of complex tasks [32, 33], it has become a new frontier to leverage the reasoning capabilities of LLMs in the time series domain [34–36]. However, straightforwardly applying existing reasoning techniques, despite their effectiveness in the NLP domain, to the time series domain leads to minimal performance gains, suggesting it is a nontrivial task to leverage LLMs for effective reasoning about TS. For example, REC4TS [37] reports that reasoning LLMs (i.e., having built-in reasoning enhancements acquired during post-training), Chain-of-Thought (CoT), and self-correction all fail to consistently improve forecasting accuracy, with only self-consistency yielding modest gains. Merrill et al. [35] assess three reasoning styles, i.e., etiological reasoning, question answering, and context-aided forecasting, and find that the first two offer negligible benefit while the third produces only modest improvements when given highly relevant context in the form of descriptive text. Other authors conclude that introducing a visual module for understanding visualized TS patterns is essential for effective reasoning [38, 39]. Chow et al. [34] and Xie et al. [40] harness LLMs’ reasoning only after incorporating time series as an additional modality, whereby they train a dedicated encoder to convert TS into embeddings that are then fed to the LLM alongside text token embeddings. In particular, Liu et al. [41] show that vanilla CoT cannot even outperform random guessing, and that in-context learning can absurdly underperform no-context baselines. They also end up resorting to visualizing TS data to have effective reasoning and obtain performance improvement.

Research Gap. At first glance, these evaluations seem to conclude that neither LLMs with inference-time reasoning techniques such as CoT and in-context illustration nor even reasoning LLMs with built-in reasoning enhancements are capable of effective reasoning for time series tasks. This makes the multimodal and specialized encoder training approaches appear indispensable to enable LLMs to substantively understand and reason about TS tasks. However, this tentative conclusion somewhat contradicts existing evidence proving that LLMs can comprehend fundamental TS patterns [42–44], based on which they should be able to grasp essential TS task characteristics for sophisticated reasoning without relying on auxiliary vision modules or specialized encoders. Even more perplexing is the observation that providing LLMs with in-context examples [41], despite providing additional task-relevant information, often degrades classification accuracy rather than improving it, implying that current in-context strategies are ill-suited to TS reasoning. These contradictory phenomena raise the following tempting research questions (RQ):

RQ1: Is it possible to steer the reasoning process of LLMs to elicit their built-in understanding of time series patterns for effective reasoning?

RQ2: Is there a strategy suitable for fusing in-context knowledge into the LLMs’ reasoning process to enhance prediction performance?

Our work. In this paper, we focus on the time series classification task and answer both research questions in the affirmative by proposing ReasonTSC, which entails a thinking procedure tailored for time series (RQ1) and a fused decision strategy effectively exploiting in-context examples (RQ2).

Tailored thinking: We posit that the ineffectiveness of existing LLMs’ reasoning may stem from the fact that straightforwardly applying NLP-domain reasoning techniques or relying on the reasoning LLMs’ built-in reasoning enhancements is insufficient to guide the model to spontaneously think over TS data characteristics. LLMs acquire reasoning skills through training on mathematics and coding tasks [45], but rarely on time series tasks, which causes them to lack the spontaneous tendency to reason about TS patterns. Motivated by this, we propose a multi-turn thinking procedure tailored to TSC, featuring a more tightly guided reasoning strategy. ReasonTSC explicitly asks LLM to identify and think about key TS data patterns. Furthermore, after the LLM provides a preliminary prediction, ReasonTSC explicitly prompts it to reconsider whether alternative answers might be more feasible, drawing on a backtracking strategy shown to be useful in the NLP domain.

Fused decision: When few-shot examples are available for in-context knowledge, we devise a fused decision strategy. First, rather than directly feeding LLMs with context information in the form of text descriptions of the data characteristics, we find it is more effective to present few-shot examples from different classes and prompt the model to autonomously compare their TS data patterns. Moreover, instead of visualizing TS data for a vision module or training a specialized encoder for TS embeddings, we propose to introduce off-the-shelf and amply available time series foundation models (TSFM) into the reasoning process. This approach offers two key strengths: 1) TSFMs are pretrained on vast time series datasets, enabling them to provide more relevant information than vision module (e.g., ViT) trained on images or TS encoders trained on much smaller TS datasets; 2) TSFMs are generally more lightweight than vision foundation models, e.g., fusing MOMENT (341M parameters) with Chronos (710M parameters) substantially boosts the classification accuracy of LLMs. To integrate TSFM

95 outputs into the LLM’s reasoning pipeline, ReasonTSC explicitly interprets TSFM’s prediction and
 96 confidence score, then makes a fused decision by taking both the interpretation of TSFM’s outputs
 97 and the LLM’s own analysis of TS patterns into the reasoning process.

98 We conduct extensive experiments and systematic ablation studies on 15 TS benchmark datasets,
 99 using 2 TSFMs and 16 mainstream LLMs to validate the effectiveness of ReasonTSC. Our key
 100 findings are: 1) ReasonTSC achieves averagely 90% performance improvement compared with
 101 a vanilla CoT prompt adopted by existing work [24], demonstrating that its tailored reasoning
 102 procedure comprehends TS characteristics more thoroughly, thereby solving the classification task
 103 more effectively; 2) When applied across 16 mainstream LLMs, ReasonTSC consistently outperforms
 104 plain CoT prompting, suggesting its broad compatibility; 3) Notably, ReasonTSC can sometimes
 105 overturn TSFM’s incorrect predictions, indicating that its elicited thinking from LLMs regarding
 106 TS characteristics involves a nuanced and in-depth analysis essential for accurate predictions. In
 107 summary, the main contributions of this paper are:

- 108 • We critically investigate the emerging paradigm of leveraging LLMs reasoning for the time series
 109 domain and posit that LLMs are capable of effective reasoning, contrary to prior conclusions that
 110 they cannot achieve performance gains through time series reasoning;
- 111 • Through the lens of time series classification, we prove it is indeed possible to leverage LLMs for
 112 effective time series reasoning by proposing ReasonTSC, a novel framework featuring a tailored
 113 multi-turn thinking procedure to explicitly steer models to analyze key TS patterns and alternative
 114 predictions, alongside a fused decision strategy to enhance in-context example utility;
- 115 • We conduct extensive experiments and systematic ablation studies on 15 datasets, with 2 TSFM
 116 from different categories, across 16 mainstream LLMs to verify the effectiveness of ReasonTSC.

117 The *Supplementary Material* provides source code and an Appendix with detailed related work,
 118 experiment settings and additional results, and further details of the proposed method.

119 2 The Proposed ReasonTSC

120 2.1 Problem Formulation

121 Let $\mathcal{D} = \{(x_i, y_i), i = 0, 1, \dots, N - 1\}$ denotes a time series dataset with N samples, where $x_i \in$
 122 $\mathcal{R}^{m \times w}$ is a sample with m variables measured for w steps, $y_i \in \{1, 2, \dots, C\}$ is the corresponding
 123 label with C be the number of classes. The classical time series classification problem is to train a
 124 classification model on the training dataset \mathcal{D}^{train} , which can predict the labels of samples in the
 125 testing dataset \mathcal{D}^{test} ,

$$\hat{y}_t = f(x_t), t = 0, 1, \dots, M - 1, \quad (1)$$

126 where M is the number of samples in the testing dataset. In this work, we propose to adopt a reasoning
 127 LLM to enhance the time series classification task.

128 Let f_M be a reasoning language model that consists of a series of rationales obtained on condition of
 129 the time series \mathcal{X}_j and tailored prompts $\phi(\mathcal{X}_j)$ in a multi-turn manner, which is applied to enhance
 130 various time series classification tasks.

$$r_j \simeq p_\theta(r_j | r_{j-1}, \mathcal{X}_j, \phi(\mathcal{X}_j)), j = 0, 1, \dots, J - 1; \quad (2)$$

$$f_M \simeq p_\theta(r_0, r_1, \dots, r_{J-1}, \mathcal{X}, \phi(\mathcal{X})); \quad (3)$$

$$\hat{y}_t = f_M(x_t, \psi(x_t)), t = 0, 1, \dots, M - 1, \quad (4)$$

131 where J is the number of reasoning turns/steps, $\phi(\mathcal{X}_j)$ is the tailored prompt based on the correspond-
 132 ing input time series samples for the j th reasoning turn/step, p_θ is a LLM, f_M is the final reasoning
 133 language model based on all the intermediate rationales and input samples, x_t is the testing sample,
 134 M is the number of testing samples, and $\psi(x_t)$ is the tailored prompt designed for the testing time
 135 series sample x_t .

136 2.2 The ReasonTSC Framework

137 As illustrated in Figure 1, the proposed ReasonTSC framework comprises three reasoning turns:
 138 (1) TS Pattern Reasoning, where the language model is asked to think about the general patterns

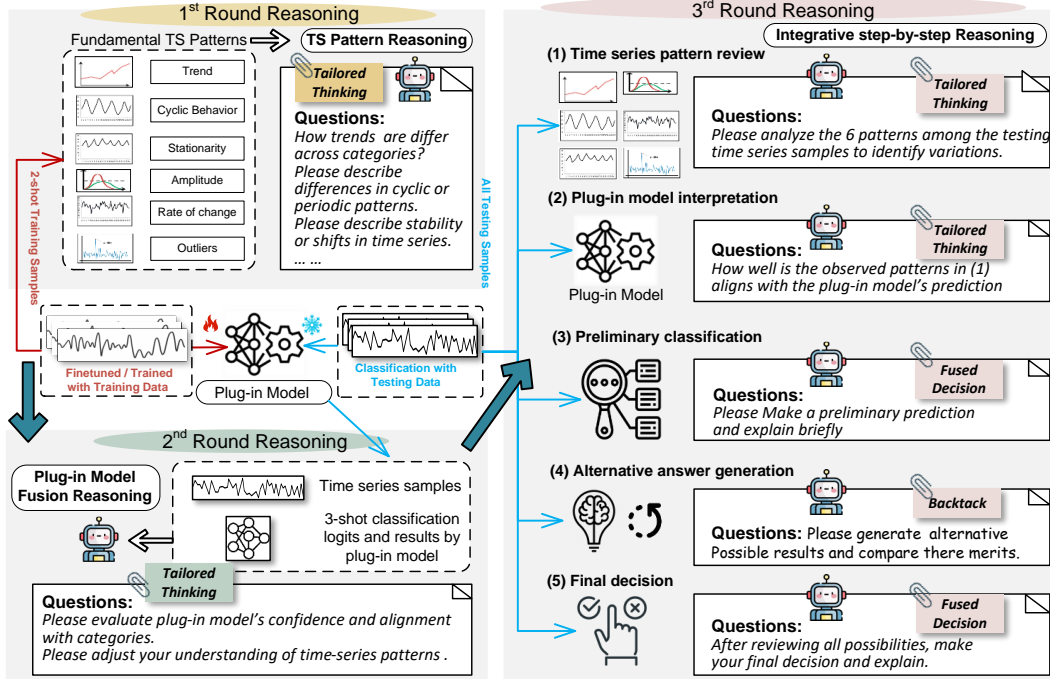


Figure 1: Architecture of the proposed ReasonTSC framework.

of time series data; (2) Plug-in Model Fusion Reasoning, where the classification logits of a finetuned/pretrained domain-specific time series model is plugged in the reasoning paradigm to enhance LLM’s understanding of the TSC task; and (3) Integrative Step-by-step Reasoning, where the reasoning paradigm is conducted step-by-step by evaluating the initial assessment, backtracking alternative hypotheses, and comparing different answers before reaching a final decision.

TS Pattern Reasoning. As mentioned in Section 1, LLM can learn to generate realistic time series by analyzing several fundamental time series characteristics such as trend, amplitude, stationarity, and so on [46, 47], which indicates that LLM can better understand the intrinsic time series patterns by thinking about these traits.

- **Trend:** A persistent, long-term directional movement (upward/downward) in the time series. It reveals fundamental shifts in data behavior at the macro-level.
- **Cyclic behavior:** Repeating patterns or periodic fluctuations. It enables the detection of seasonal or cyclical variations.
- **Stationarity:** The stability of time-invariant statistical properties (mean, variance) or their shifts. It is essential for assessing the underlying structure of time series.
- **Amplitude:** The maximal deviation magnitude during fluctuations. It quantifies the intensity of variations in the data.
- **Rate of change:** The speed at which the data changes (rapid/moderate/slow). It characterizes the temporal dynamics of the time series.
- **Outliers:** Data points that deviate significantly from normal values. It may indicate anomalies and data quality issues.

Thus, for the ReasonTSC framework, we first aim to obtain the LLM rationales by answering questions in terms of time series fundamental traits. To be specific, 2-shot time series samples are randomly selected per category from the training set. The LLM is prompted to compare the differences among various categories in terms of the selected fundamental traits. We also include domain-specific knowledge in the prompts and encourage the adopted LLM to decompose a series into

Table 1: Classification accuracy (%). MOMENT is plugged in for ReasonTSC.

Model	Dist. TW	Mid. TW	Mid. OA	Elec.	Med. Img	BME	Arr. Hd	Dod. LD
MOMENT (<i>reference and fused TSFM</i>)	62.59	51.30	60.39	57.89	76.97	74.00	65.71	31.17
Vanilla CoT (GPT-4o-mini)	33.81	23.38	41.56	36.84	9.87	42.34	45.14	15.58
ReasonTSC (GPT-4o-mini)	63.31	52.60	61.04	58.55	77.63	77.33	68.00	31.17
Improvement	+87.25%	+124.98%	+46.87%	+58.93%	+686.52%	+82.64%	+50.64%	+100.06%
Vanilla CoT (Llama-3.3-70B-instruct)	33.10	41.24	31.17	46.71	13.16	59.00	42.36	31.81
ReasonTSC (Llama-3.3-70B-instruct)	63.31	53.95	61.04	61.18	77.63	84.00	66.86	36.36
Improvement	+91.27%	+30.82%	+95.83%	+30.98%	+489.89%	+42.37%	+57.84%	+14.30%
Vanilla CoT (DeepSeek-R1)	52.52	47.08	33.11	51.98	37.17	76.66	54.86	28.57
ReasonTSC (DeepSeek-R1)	65.71	57.42	63.64	67.11	80.26	82.67	69.14	38.96
Improvement	+25.11%	+21.96%	+92.21%	+29.11%	+115.93%	+7.84%	+26.03%	+36.37%
Model	CBF	Rkt. Spt	ERing	Nt.Ops	Lbr.	Eplp.	Pen.	Avg
MOMENT (<i>reference and fused TSFM</i>)	66.00	59.21	72.59	65.56	48.49	88.40	85.62	64.39
Vanilla CoT (GPT-4o-mini)	45.67	34.26	36.67	38.61	22.78	51.45	21.92	33.33
ReasonTSC (GPT-4o-mini)	65.33	67.76	74.81	65.56	48.89	89.13	86.30	65.83
Improvement	+43.05%	+97.78%	+104.01%	+69.80%	+114.62%	+73.24%	+293.7%	+135.61%
Vanilla CoT (Llama-3.3-70B-instruct)	47.67	39.48	51.11	38.61	25.83	55.44	23.63	38.69
ReasonTSC (Llama-3.3-70B-instruct)	73.33	61.84	74.07	66.67	51.11	89.86	86.99	67.21
Improvement	+62.22%	+56.64%	+44.92%	+72.68%	+97.87%	+62.09%	+268.13%	+101.19%
Vanilla CoT (DeepSeek-R1)	65.00	47.04	55.56	46.11	38.89	63.41	40.76	49.25
ReasonTSC (DeepSeek-R1)	74.00	63.16	74.07	67.78	55.00	91.30	86.30	69.10
Improvement	+13.85%	+34.27%	+33.32%	+47.00%	+41.42%	+43.98%	+111.73%	+45.34%

semantically meaningful segments to enhance its understanding [15]. Please refer to the Appendix B for complete prompts.

Plug-in Model Fusion Reasoning. According to [48], classification results by a small model could enhance LLM’s ability on domain-specific tasks. Here, we propose to plug in a task-specific classifier to obtain further rationales about the TSC tasks by integrating the classification logits. Specifically, a task-specific time series classifier is first trained on the training dataset. Then, 3-shot time series samples are randomly selected from the testing set and fed to the trained classifier to obtain its classification logits and decision confidence. The logits, confidence, the ground truth labels, and the basic information (e.g., its training accuracy) of the trained task-specific plug-in model are fused as auxiliary references for the LLM to understand the TSC task. The LLM is asked to analyze cases where the plug-in model correctly or incorrectly identifies different classes to refine its understanding of how to conduct the TSC task. Please refer to the Appendix B for complete prompts.

Integrative Step-by-step Reasoning. For the third reasoning turn, we concatenate each testing time series sample with its corresponding predicted label and confidence scores from the plug-in model as input to the reasoning LLM. Rather than simply adopting the generic "think step by step" prompt prefix, we design a tailored CoT approach for the TSC task. The reasoning LLM, with its ability gained in the first two turns, is asked to analyze the patterns of the testing sample and the classification results provided by the plug-in model. Based on this analysis, the reasoning LLM generates a preliminary prediction with supporting rationale. Then, the LLM is asked to backtrack and explore alternative predictions and systematically compare their merits against the initial assessment. Finally, the reasoning LLM synthesizes all evidence to generate a refined final classification decision. Please refer to the Appendix B for complete prompts.

3 Experiments

3.1 Experimental Settings

Plug-in domain-specific time series models We select two prominent time series foundation models as the plug-in classifiers: (1) MOMENT [28], a T5-based encoder-only model, which is fully fine-tuned with our training data. (2) Chronos [49] is an encoder-decoder model primarily designed for TS forecasting, whose pretrained encoder is adopted to extract time series embeddings for training an SVM-based classifier with the training data.

Table 2: Classification accuracy (%). Chronos is plugged in for ReasonTSC.

Model	Dist. TW	Mid. TW	Mid. OA	Elec.	Med. Img	BME	Arr. Hd	Dod. LD
Chronos (<i>reference and fused TSFM</i>)	60.43	57.79	52.60	46.71	65.39	76.00	48.57	55.84
Vanilla CoT (GPT-4o-mini)	33.81	23.38	41.56	36.84	9.87	42.34	45.14	15.58
ReasonTSC (GPT-4o-mini)	61.15	57.79	57.14	45.39	69.74	78.00	54.29	58.44
Improvement	+80.86%	+147.18%	+37.49%	+23.21%	+606.59%	+84.22%	+20.27%	+275.10%
Vanilla CoT (Llama-3.3-70B-instruct)	33.10	41.24	31.17	46.71	13.16	59.00	42.36	31.81
ReasonTSC (Llama-3.3-70B-instruct)	64.03	59.09	53.90	48.03	71.05	86.00	50.29	57.14
Improvement	+93.44%	+43.28%	+72.92%	+2.83%	+439.89%	+45.76%	+18.72%	+79.63%
Vanilla CoT (DeepSeek-R1)	52.52	47.08	33.11	51.98	37.17	76.66	54.86	28.57
ReasonTSC (DeepSeek-R1)	64.75	61.69	54.55	53.95	73.03	85.33	54.29	62.34
Improvement	+23.29%	+31.03%	+64.75%	+3.79%	+96.48%	+11.31%	-1.04%	+118.20%
Model	CBF	Rkt. Spt	ERing	Nt.Ops	Lbr.	Eplp.	Pen.	Avg
Chronos (<i>reference and fused TSFM</i>)	90.89	54.61	53.33	62.22	42.22	91.30	68.49	61.76
Vanilla CoT (GPT-4o-mini)	45.67	34.26	36.67	38.61	22.78	51.45	21.92	33.33
ReasonTSC (GPT-4o-mini)	89.33	53.95	51.85	63.89	41.67	91.30	65.75	62.65
Improvement (%)	+95.60%	+57.47%	+41.40%	+65.48%	+82.92%	+77.45%	+199.95%	+126.35%
Vanilla CoT (Llama-3.3-70B-instruct)	47.67	39.48	51.11	38.61	25.83	55.44	23.63	38.69
ReasonTSC (Llama-3.3-70B-instruct)	95.33	55.26	57.04	66.67	45.00	92.03	69.18	64.67
Improvement	+99.98%	+39.97%	+11.60%	+72.68%	+74.22%	+66.00%	+192.76%	+90.25%
Vanilla CoT (DeepSeek-R1)	65.00	47.04	55.56	46.11	38.89	63.41	40.76	49.25
ReasonTSC (DeepSeek-R1)	93.33	61.84	62.96	67.78	57.22	94.93	61.64	67.31
Improvement	+43.58%	+31.46%	+13.32%	+47.00%	+47.13%	+49.74%	+51.23%	+42.08%

Reasoning LLMs The main body of experiments is conducted with three primary LLMs—GPT-4o-mini, Llama-3-70B-Instruct, and DeepSeek-R1, covering different parameter scales and reasoning training techniques. To further investigate how reasoning LLMs can enhance TSC tasks, we also evaluate the performance of ReasonTSC with six other mainstream LLMs on three selected UCR/UEA datasets, including ChatGPT, Claude, Gemini, Qwen [50, 51], Llama [52], and Grok, with a fixed temperature parameter of 0.2.

Datasets We select 15 datasets from the UCR/UEA classification archive [53, 54] that are commonly used for benchmarking classification algorithms, covering diverse scenarios and varying numbers of classes. We only use the first dimension of the multivariate UEA datasets to address the token limit restrictions imposed by LLM input queries. Given the typically long sequence lengths of time series samples, we retain values to three decimal places to optimize context window usage. Please refer to Appendix C for details about LLMs and datasets.

Implementation Details We maintain the original training-test splits from the UCR/UEA archive. All fine-tuning and training experiments are performed on an NVIDIA RTX 4090 GPU.

3.2 Main Results

As shown in Tables 1 and 2, the vanilla CoT with different LLMs presents consistently low accuracy values. This observation reveals that LLMs cannot enhance TSC tasks by adopting their built-in reasoning capabilities with CoT [24]. On the contrary, ReasonTSC achieves substantial performance improvements (+20%~+600%, average 90%) by incorporating a tailored thinking and fused decision strategy. With more scrutiny to compare ReasonTSC and the plug-in models, ReasonTSC outperforms the plug-in models across almost all the tested datasets. Specifically, ReasonTSC with DeepSeek as the reasoning language model surpasses the plug-in model MOMENT by over 10% on six datasets, including substantial performance improvement by 24.99% on DodgerLoopDay (Dod.LD) and 15.93% on ElectricDevices (Elec.). It is worth mentioning that the plug-in models are fine-tuned/trained on the whole training dataset, while the ReasonTSC is only shown with two samples per category, which indicates the efficiency of the proposed reasoning strategy.

To further investigate the proposed ReasonTSC’s reasoning capabilities, we show the average override rates of ReasonTSC compared with plug-in models as shown in Table 3. ReasonTSC with DeepSeek exhibits an override rate of 11.89% on average, which is higher than that by ReasonTS (Llama) (5.12%) and ReasonTSC (GPT) (4.23%). Regarding override accuracy, ReasonTSC (Llama) and ReasonTSC (DeepSeek) achieve average override accuracy of 77.41% and 65.68%, respectively.

Table 3: Results of ReasonTSC’s classification overrides against plug-in models. The Overriden (%) shows the percentage of classification results that are different from those by plug-in models. The Override Accuracy (%) shows the rate of correct classification results among these overrides.

	Overriden (%)			Override Accuracy (%)		
	MOMENT	Chronos	Average	MOMENT	Chronos	Average
ReasonTSC (GPT-4o-mini)	2.77	5.68	4.23	65.34	29.37	47.36
ReasonTSC (Llama-3.3-70b-instruct)	4.23	6.00	5.12	83.30	71.51	77.41
ReasonTSC (Deepseek-R1)	9.42	14.36	11.89	68.47	62.88	65.68

This suggests that ReasonTSC can effectively leverage LLMs’ understanding of time series patterns through multi-turn reasoning to correct incorrect predictions by plug-in models.

Besides, we also evaluate the proposed ReasonTSC with other mainstream LLMs as its reasoning language models on three datasets. As illustrated in Figure 2, the horizontal black dashed line marks the performance of the plug-in model MOMENT. In Figure 2 (a), we compare ReasonTSC’s performance in terms of the model sizes of different language models. Here, ReasonTSC’s performance does not show an obvious correlation with the sizes and architectures of language models. On the other hand, Gemini-2.5-pro (175B parameters) and Deepseek-v3 (671B parameters) achieve the best and second-best performance. The red and blue solid lines represent the performance of Vanilla CoT reasoning with Gemini-2.5-pro and Deepseek-v3, respectively. It is shown that even for the recently newly released LLMs with strong reported built-in reasoning ability, the proposed ReasonTSC shows much performance improvement over the Vanilla CoT reasoning strategy. Please refer to Appendix D for complete experimental results.

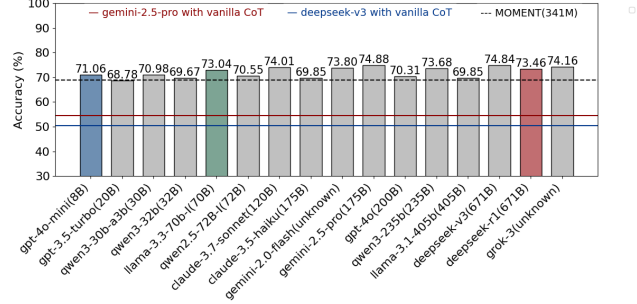


Figure 2: Average performance of ReasonTSC with mainstream LLMs as reasoning language models on three selected UCR/UEA datasets (MiddlePhalanxOutlineAgeGroup, BME, and ERing).

3.3 Analysis of Key Thinking Steps

Thinking TS patterns In the first round of reasoning, ReasonTSC thinks about the fundamental TS patterns by showing few-shot training samples of each category. We examine how the number of few-shot examples affects reasoning performance. As shown in Figure 3, with one or two examples, ReasonTSC achieves average classification performance of 61.39% and 62.92%, respectively, surpassing the performance of the plug-in model (MOMENT). ReasonTSC’s performance slightly declines when shown three examples, which is potentially caused by information overload in prompt-based inputs that hinders the language model’s ability to process excessive information (the full multi-round prompt combined with three samples exceeds the 10K context length in most subsets).

Backtracking During the integrative step-by-step reasoning process (third reasoning turn), the *alternative answer generation* step guides ReasonTSC to backtrack to consider alternative hypotheses and compares their merits before arriving at a final classification decision. Figure 4 illustrates the counts of cases where ReasonTSC ultimately adopts alternative candidates in their final predictions. ReasonTSC with Llama shows higher sensitivity than ReasonTSC s with GPT and DeepSeek, where 58 successful corrections out of 109 alternative adoptions are presented. ReasonTSC s with DeepSeek and GPT present successful correction rates of 75% and 42.31%, respectively. This reveals that with a step-by-step integrative reasoning strategy, the proposed ReasonTSC could comprehensively consider the TS patterns and plug-in model’s auxiliary information, and correct its primary decision.

3.4 Research Questions

3.4.1 TS Pattern Interpretation (RQ1)

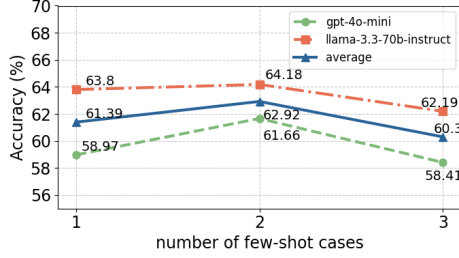


Figure 3: ReasonTSC’s performance based on the number of few-shot examples provided in the 1st turn of reasoning.

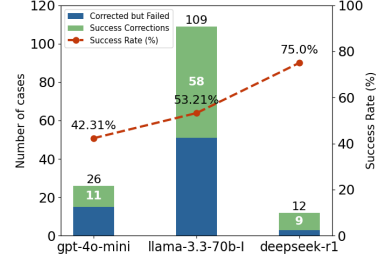


Figure 4: Effectiveness of the *alternative answer generation* step in the 3rd turn of reasoning.

To further answer **RQ1**, we evaluate ReasonTSC’s ability to think about time-series patterns in this section. We first construct four synthetic time series datasets, where the first three individually exhibit distinct trend, frequency, and amplitude patterns, while the last one integrates these three patterns. We present each time series sample alongside randomly generated noise sequences in a multiple-choice format, questioning the ReasonTSC to identify the sequence with the most discernible patterns. Choice positions are randomized to eliminate positional bias. Notably, ReasonTSC s with GPT, Llama, and Deepseek achieve satisfactory accuracy across all the tested datasets,

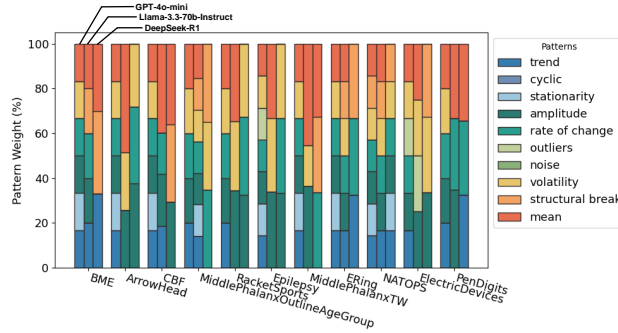


Figure 5: Evaluation of ReasonTSC’s ability to reason about time series patterns using real-world datasets. We select 11 datasets from UCR and UEA archives, and ask the model to identify the 10 typical time series patterns across different datasets. For each dataset, the predominant patterns identified by GPT-4o-mini, Llama3.3-70b-instruct, and DeepSeek-R1 are shown in the bars in a left-to-right order.

demonstrating ReasonTSC’s ability to generate rationales about fundamental time series patterns. Details of dataset construction, question design, and related prompts are provided in Appendix E. We further evaluate ReasonTSC’s ability to reason about time-series patterns using the realistic UCR/UEA archives. Here we evaluate ten fundamental patterns as mentioned in Section 2: *trend*, *cyclic*, *stationarity*, *amplitude*, *rate of change*, *outliers*, *noise*, *volatility*, *structural break*, and *mean shift* [46]. For each sample, we randomly select one unique instance per category and ask the ReasonTSC to identify significant pattern differences across categories. We quantitatively summarize the responses by counting the top three most frequently identified patterns (including ties) and calculating their relative weights. As shown in Figure 5, ReasonTSC with GPT-4o-mini consistently identifies similar TS patterns (e.g., trend, amplitude, rate of change, volatility, and mean shift) across all datasets, suggesting it tends to present more generalized interpretations (cannot discern different datasets), which aligns with the final classification performance where it shows relatively lower classification accuracy. On the contrary, ReasonTSC with DeepSeek-R1 (which also shows the best overall classification performance) shows superior performance in identifying category-discriminative patterns: it recognizes trend, structural break, and mean shift as distinctive features in the BME dataset, while recognizing amplitude, rate of change, and volatility as predominant in the ArrowHead dataset. **These observations indicate that a better understanding of the time series patterns could enhance the reasoning process of LLMs and the TSC accordingly.** Details of prompts and corresponding answers are provided in Appendix E.

3.4.2 Ablation of Fusion Strategy (RQ2)

To answer **RQ2**, we conduct ablation studies to evaluate the impact of fused decision strategy: (1) reasoning about the category-wise confidence scores (logits) of the plug-in model (w/o logits), and (2) the complete outputs (logits & final predictions) of the plug-in model (w/o plug-in model).

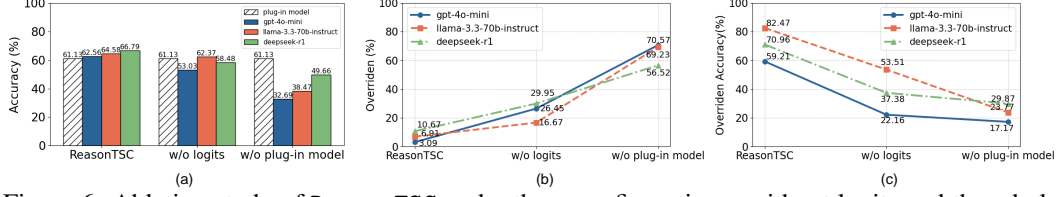


Figure 6: Ablation study of ReasonTSC under three configurations: without logits and the whole plug-in model. Three merits are compared under these conditions: classification performance (a), overridden rate (b), and override accuracy (c).

As illustrated in Figure 6 (a), removing the plug-in model’s logits leads to an 8.31% performance decline in ReasonTSC with DeepSeek; Completely removing outputs of the plug-in model leads to a significant performance decrease. **This indicates the importance of the fused decision strategy.**

As shown in Figure 6 (b) and (c), the override rates of ReasonTSC s increase while their overall override accuracy decreases with reduced reasoning supports. When the plug-in model’s logits are removed, we observe higher override rates and bigger accuracy degradation, which also **shows that the fused decision strategy with the plug-in model enhances ReasonTSC’s performance in TSC.** Please refer to Appendix D for more ablation studies.

3.4.3 Decision Interpretation (RQ1&2)

Since the ReasonTSC is asked to explain its final decision, we can count for each override case which information drives the model to make different classification results. As shown in Figure 7, ReasonTSC with GPT relies on the plug-in model’s logits and time series patterns in all the override cases. ReasonTSC s with Llama and DeepSeek partially rely on the plug-in model’s accuracy for their override decisions. Specifically, ReasonTSC with GPT relies on the TS patterns only for the majority of override cases(63.49%). As discussed in Section 3.4.1, ReasonTSC with GPT cannot discern the TS patterns among different categories. Its heavy reliance on the TS patterns for final decision can also explain its relatively low classification performance compared to the other two scenarios (ReasonTSC s with Llama and DeepSeek). This interpretation analysis shows that both the TS patterns and the fused plug-in model influence the final performance of the proposed ReasonTSC .

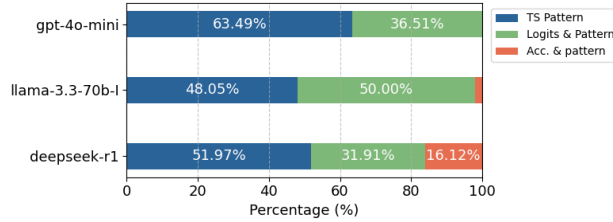


Figure 7: Reasons for ReasonTSC override: (i) primary reliance on typical time series patterns, (ii) consideration of both the plug-in model’s logits and time series patterns, (iii) combined assessment of the plug-in model’s accuracy and time series patterns.

4 Conclusion

The paper presents ReasonTSC, a novel framework that effectively leverages reasoning LLMs for time series classification through a multi-turn reasoning and fused decision-making strategy. It first guides the LLM to analyze the intrinsic patterns of time series data. It then incorporates predictions and category-wise confidence scores from the plug-in model as in-context examples to enhance its understanding of the TSC task. Finally, ReasonTSC orchestrates a structured reasoning pipeline: the LLM evaluates its initial assessment, backtracks to consider alternative hypotheses, and compares their merits before determining the final classification. Extensive experiments and ablation studies demonstrate that ReasonTSC consistently outperforms both LLMs with Vanilla CoT reasoning and plug-in models, and is even capable of identifying plug-in models’ false predictions and correcting them accordingly. This reveals significant potential for leveraging reasoning LLMs to enhance time series classification tasks in various domains. However, the proposed ReasonTSC remains constrained by the inherent context length limitations of LLMs when processing long time series sequences. Future work could explore alternative tokenization methods to improve time series representation for LLMs.

References

- [1] Yihe Wang, Nan Huang, Taida Li, Yujun Yan, and Xiang Zhang. Medformer: A multi-granularity patching transformer for medical time-series classification. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [2] Xiaofeng Liu, Zhihong Liu, Jie Li, and Xiang Zhang. Semi-supervised contrastive learning for time series classification in healthcare. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.
- [3] Qi An, Saifur Rahman, Jingwen Zhou, and James Jin Kang. A comprehensive review on machine learning in healthcare industry: classification, restrictions, opportunities and challenges. *Sensors*, 23(9):4178, 2023.
- [4] Mengxia Liang, Xiaolong Wang, and Shaocong Wu. Improving stock trend prediction through financial time series classification and temporal correlation analysis based on aligning change point. *Soft Computing*, 27(7):3655–3672, 2023.
- [5] Sourav Majumdar and Arnab Kumar Laha. Clustering and classification of time series using topological data analysis with applications to finance. *Expert Systems with Applications*, 162:113868, 2020.
- [6] Chao-Han Huck Yang, Yun-Yun Tsai, and Pin-Yu Chen. Voice2series: Reprogramming acoustic models for time series classification. In *International conference on machine learning*, pages 11808–11819. PMLR, 2021.
- [7] Ashish Gupta, Hari Prabhat Gupta, Bhaskar Biswas, and Tanima Dutta. Approaches and applications of early classification of time series: A review. *IEEE Transactions on Artificial Intelligence*, 1(1):47–61, 2020.
- [8] Will Ke Wang, Ina Chen, Leeor Hershkovich, Jiamu Yang, Ayush Shetty, Geetika Singh, Yihang Jiang, Aditya Kotla, Jason Zisheng Shang, Rushil Yerrabelli, et al. A systematic review of time series classification techniques used in biomedical applications. *Sensors*, 22(20):8016, 2022.
- [9] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [10] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [11] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-rl: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [12] 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.
- [13] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.
- [14] Baiting Chen, Zhimei Ren, and Lu Cheng. Conformalized time series with semantic features. *Advances in Neural Information Processing Systems*, 37:121449–121474, 2024.
- [15] Jinliang Deng, Feiyang Ye, Du Yin, Xuan Song, Ivor Tsang, and Hui Xiong. Parsimony or capability? decomposition delivers both in long-term time series forecasting. *Advances in Neural Information Processing Systems*, 37:66687–66712, 2024.
- [16] Harshavardhan Prabhakar Kamarthi and B Aditya Prakash. Large pre-trained time series models for cross-domain time series analysis tasks. *Advances in Neural Information Processing Systems*, 37:56190–56214, 2024.
- [17] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *Advances in Neural Information Processing Systems*, 37:122154–122184, 2024.
- [18] Qingxiang Liu, Xu Liu, Chenghao Liu, Qingsong Wen, and Yuxuan Liang. Time-ffm: Towards Im-empowered federated foundation model for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, pages 94512–94538. Curran Associates, Inc., 2024.

- [19] Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam Nguyen, Wesley M Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. *Advances in Neural Information Processing Systems*, 37:74147–74181, 2024.
- [20] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024.
- [21] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.
- [22] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations (ICLR)*, 2024.
- [23] Jiecheng Lu, Yan Sun, and Shihao Yang. In-context time series predictor. *arXiv preprint arXiv:2405.14982*, 2024.
- [24] Zihao Zhou and Rose Yu. Can llms understand time series anomalies?, 2024.
- [25] Zhihao Dai, Ligang He, Shuanghua Yang, and Matthew Leeke. Sarad: Spatial association-aware anomaly detection and diagnosis for multivariate time series. *Advances in Neural Information Processing Systems*, 37:48371–48410, 2024.
- [26] Shuyu Wang, Wengen Li, Hanchen Yang, Jihong Guan, Xiwei Liu, Yichao Zhang, Rufu Qin, and Shuigeng Zhou. Llm4hrs: Llm-based spatio-temporal imputation model for highly-sparse remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 2025.
- [27] Zhixian Wang, Linxiao Yang, Liang Sun, Qingsong Wen, and Yi Wang. Task-oriented time series imputation evaluation via generalized representers. *Advances in Neural Information Processing Systems*, 37:137403–137431, 2024.
- [28] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *International Conference on Machine Learning*, 2024.
- [29] Yunshi Wen, Tengfei Ma, Ronny Luss, Debarun Bhattacharjya, Achille Fokoue, and Anak Agung Julius. Shedding light on time series classification using interpretability gated networks. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [30] Vasilii Feofanov, Songkang Wen, Marius Alonso, Romain Ilbert, Hongbo Guo, Malik Tiomoko, Lujia Pan, Jianfeng Zhang, and Ievgen Redko. Mantis: Lightweight calibrated foundation model for user-friendly time series classification. *arXiv preprint arXiv:2502.15637*, 2025.
- [31] Xiaoyu Tao, Tingyue Pan, Mingyue Cheng, and Yucong Luo. Hierarchical multimodal llms with semantic space alignment for enhanced time series classification, 2024.
- [32] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.
- [33] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [34] Winnie Chow, Lauren Gardiner, Haraldur T. Hallgrímsson, Maxwell A. Xu, and Shirley You Ren. Towards time series reasoning with llms. *ArXiv*, abs/2409.11376, 2024.
- [35] Mike A Merrill, Mingtian Tan, Vinayak Gupta, Tom Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. *arXiv preprint arXiv:2404.11757*, 2024.
- [36] Ming Jin, Yifan Zhang, Wei Chen, Kexin Zhang, Yuxuan Liang, Bin Yang, Jindong Wang, Shirui Pan, and Qingsong Wen. Position: What can large language models tell us about time series analysis. In *Forty-first International Conference on Machine Learning*, 2024.
- [37] Haoxin Liu, Zhiyuan Zhao, Shiduo Li, and B. Aditya Prakash. Evaluating system 1 vs. 2 reasoning approaches for zero-shot time-series forecasting: A benchmark and insights, 2025.

- [38] Yaxuan Kong, Yiyuan Yang, Shiyu Wang, Chenghao Liu, Yuxuan Liang, Ming Jin, Stefan Zohren, Dan Pei, Yan Liu, and Qingsong Wen. Position: Empowering time series reasoning with multimodal llms, 2025.
- [39] Jialin Chen, Aosong Feng, Ziyu Zhao, Juan Garza, Gaukhar Nurbek, Cheng Qin, Ali Maatouk, Leandros Tassioulas, Yifeng Gao, and Rex Ying. Mtbench: A multimodal time series benchmark for temporal reasoning and question answering. *arXiv preprint arXiv:2503.16858*, 2025.
- [40] Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *arXiv preprint arXiv:2412.03104*, 2024.
- [41] Haoxin Liu, Chenghao Liu, and B. Aditya Prakash. A picture is worth a thousand numbers: Enabling LLMs reason about time series via visualization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7486–7518, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6.
- [42] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36:19622–19635, 2023.
- [43] Mingtian Tan, Mike Merrill, Vinayak Gupta, Tim Althoff, and Tom Hartvigsen. Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems*, 37: 60162–60191, 2024.
- [44] Haoxin Liu, Zhiyuan Zhao, Jindong Wang, Harshavardhan Kamarthi, and B Aditya Prakash. LSTPrompt: Large language models as zero-shot time series forecasters by long-short-term prompting. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7832–7840, August 2024.
- [45] Yiheng Liu, Hao He, Tianle Han, Xu Zhang, Mengyuan Liu, Jiaming Tian, Yutong Zhang, Jiaqi Wang, Xiaohui Gao, Tianyang Zhong, et al. Understanding llms: A comprehensive overview from training to inference. *Neurocomputing*, page 129190, 2024.
- [46] Yifu Cai, Arjun Choudhry, Mononito Goswami, and Artur Dubrawski. Timeseriesexam: A time series understanding exam. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.
- [47] Willa Potosnak, Cristian Ignacio Challu, Mononito Goswami, Michał Wiliński, Nina Żukowska, and Artur Dubrawski. Implicit reasoning in deep time series forecasting. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.
- [48] Linyi Yang, Shuibai Zhang, Zhuohao Yu, Guangsheng Bao, Yidong Wang, Jindong Wang, Ruochen Xu, Wei Ye, Xing Xie, Weizhu Chen, and Yue Zhang. Supervised knowledge makes large language models better in-context learners. In *The Eighteenth International Conference on Learning Representations (ICLR 2024)*, 2024.
- [49] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.
- [50] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.
- [51] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- [52] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [53] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019. doi: 10.1109/JAS.2019.1911747.
- [54] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.
- [55] Jonathan D Cryer. *Time series analysis*. Springer, 2008.

- [56] Hao Xue and Flora D Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6851–6864, 2023.
- [57] Hua Tang, Chong Zhang, Mingyu Jin, Qinkai Yu, Zhenting Wang, Xiaobo Jin, Yongfeng Zhang, and Mengnan Du. Time series forecasting with llms: Understanding and enhancing model capabilities. *ACM SIGKDD Explorations Newsletter*, 26(2):109–118, 2025.
- [58] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Forty-first International Conference on Machine Learning*, 2024.
- [59] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.
- [60] Tuo An, Yunjiao Zhou, Han Zou, and Jianfei Yang. Iot-llm: Enhancing real-world iot task reasoning with large language models. *arXiv preprint arXiv:2410.02429*, 2024.
- [61] Mingyue Cheng, Yiheng Chen, Qi Liu, Zhiding Liu, Yucong Luo, and Enhong Chen. Instructtime: Advancing time series classification with multimodal language modeling. In *WSDM '25*, 2025. doi: 10.1145/3701551.3703499.
- [62] Wen Ye, Wei Yang, Defu Cao, Yizhou Zhang, Luminyuan Tang, Jie Cai, and Yan Liu. Domain-oriented time series inference agents for reasoning and automated analysis, 2024.
- [63] Willa Potosnak, Cristian Challu, Mononito Goswami, Kin G. Olivares, Michał Wiliński, Nina Żukowska, and Artur Dubrawski. Investigating compositional reasoning in time series foundation models, 2025.
- [64] Willa Potosnak, Cristian Challu, Mononito Goswami, Michał Wiliński, Nina Żukowska, and Artur Dubrawski. Implicit reasoning in deep time series forecasting, 2024.
- [65] Mike A. Merrill, Mingtian Tan, Vinayak Gupta, Tom Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. *CoRR*, abs/2404.11757, 2024.
- [66] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *International Conference on Learning Representations*, 2024.
- [67] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Yunzhong Qiu, Haoran Zhang, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 2024.
- [68] Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. *KDD '24*, page 6555–6565, 2024. doi: 10.1145/3637528.3671451.
- [69] Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Trans. Intell. Syst. Technol.*, 2025. ISSN 2157-6904. doi: 10.1145/3719207.
- [70] Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM Web Conference 2024*, 2024.
- [71] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. TEMPO: Prompt-based generative pre-trained transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.
- [72] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.
- [73] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 2021.
- [74] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. 39th International Conference on Machine Learning (ICML 2022)*, 2022.

- 556 [75] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itrans-
557 former: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*,
558 2023.
- 559 [76] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words:
560 Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.
- 561 [77] David Campos, Miao Zhang, Bin Yang, Tung Kieu, Chenjuan Guo, and Christian S. Jensen. Lights:
562 Lightweight time series classification with adaptive ensemble distillation. *Proc. ACM Manag. Data*, 1(2):
563 171:1–171:27, 2023. doi: 10.1145/3589316.
- 564 [78] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting?
565 *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, Jun. 2023.

Appendix: Enhancing LLM Reasoning for Time Series Classification by Tailored Thinking and Fused Decision

This appendix contains: 1) Section A: deferred related work. 2) Section B: reasoning details of our proposed ReasonTSC framework; 3) Section C: implementation details of ReasonTSC, including datasets, adopted LLMs, and adopted time series foundation models; 4) Section D: additional experimental results and analysis; 5) Section E: comprehensive interpretation study of ReasonTSC on time series patterns; 6) Section F: limitations and discussions of ReasonTSC. Our source code for replicating the experiments can be found in <https://anonymous.4open.science/r/ReasonTSC-B737>.

A Related Work

In addition to the existing works discussed in Section 1, we provide a brief review of related works on leveraging LLM for time series analysis, as follows.

Time-Series Tasks Using LLMs. The history of time series analysis can be traced back to signal processing and solving the first-order differential equations [55]. With the advent of LLMs, a variety of works have shown the potential for leveraging LLMs for TS tasks. PromptCast [56] transforms numerical time series into prompts in a natural language generation manner. Gruver et al. [42] demonstrates that LLMs like GPT-3 and LLaMA-2 can perform zero-shot time series extrapolation. Tang et al. [57] finds that LLMs perform well in forecasting time series with clear patterns but face challenges with datasets lacking periodicity. Jin et al. [22] proposes to reprogram time series data for general forecasting. GPT4TS [21] fine-tunes the GPT-2 backbone to perform time series analysis tasks such as classification and anomaly detection. Zhou and Yu [24] evaluate LLMs’ capabilities for conducting time series anomaly detection and conclude that LLMs can understand trivial time series anomalies. LLMs are adopted in [26] to impute the highly sparse remote sensing data, where LLMs are applied to capture the spatiotemporal dependencies buried in data sequences. Tao et al. [31] adopt a multi-modal approach to leveraging LLMs for enhancing the time series classification task, which trains two additional encoders to convert time series data into embeddings. They do not explore reasoning techniques at the LLM end. In contrast, the primary goal of our work is to investigate whether dedicated reasoning strategies can enhance LLMs’ ability to understand time series tasks without relying on additional visual modalities or the training of specialized encoders. Besides, various Time Series Foundation Models (TSFMs), inspired by the corresponding architectures and pretraining strategies from LLM literature, are pretrained with a large scale of time series data [28, 58, 20, 49, 59] targeting to solve common time series analysis tasks, including forecasting, anomaly detection, imputation, and classification.

Time-Series Tasks Using LLMs Reasoning. Recently, it has become a new frontier to leverage the reasoning capabilities of LLMs in the time series domain [34–36]. LSTPrompt [44] introduces a Chain-of-Thought (CoT) approach that guides LLMs to decompose forecasting tasks into short-term and long-term subtasks. REC4TS [37] evaluates several reasoning strategies, such as Chain-of-Thought and self-correction, to enhance LLMs in forecasting tasks. However, forecasting metrics like MSE only require LLMs to generate approximate extrapolations, whereas reasoning typically demands that LLMs pinpoint definitive answers through grasping the underlying patterns. Thus, Zhou and Yu [24] investigates the time series anomaly detection capabilities of LLMs, while An et al. [60] proposes IoT-LLM to explore the potential for IoT task reasoning. InstructTime [61] redefines time series classification by framing it as a multimodal generative task. TS-Reasoner [62] enables precise multi-step time series analysis by integrating LLM-based task decomposition using in-context learning and program-aided execution with self-correcting feedback loops. Potosnak et al. [63] and Potosnak et al. [64] demonstrate that patch-based Transformers exhibit robust generalization to systematic out-of-distribution scenarios, suggesting their intrinsic reasoning abilities surpass mere pattern memorization. However, Merrill et al. [65] evaluate LLMs on tasks such as etiological reasoning and context-aided forecasting, revealing that time-series reasoning remains a critical yet severely underdeveloped direction.

B Reasoning Details of Our Proposed ReasonTSC Framework

This Section presents the full details of the prompt templates devised by ReasonTSC in Subsection B.1, followed by the responses of three LLMs in the third reasoning round in Subsection B.2.

618 B.1 Full Prompt Templates for Three-turn Reasoning Rounds of ReasonTSC

619 The proposed ReasonTSC develops a multi-turn reasoning approach with a fused decision-making
620 strategy tailored to TSC. The framework consists of three key reasoning stages: (1) TS Pattern Reason-
621 ing. ReasonTSC guides the LLM to analyze typical patterns across time series categories. (2) Plug-in
622 Model Fusion Reasoning. Predictions and confidence scores from domain-specific time series models
623 are incorporated as in-context examples. (3) Integrative Step-by-step Reasoning. ReasonTSC guides
624 the LLM through a structured reasoning process. It evaluates the initial assessment, backtracks to
625 consider alternative hypotheses, and compares their merits before arriving at a final classification.
626 The complete prompt template for this process is presented below.

1st Round Reasoning Prompt: TS Pattern Reasoning

Task Description

You are given a time series classification task with the [dataset name] dataset, [domain-specific knowledge of the dataset]. You will be provided with two time series samples from each category of this dataset. Your first task is to analyze and compare the significant pattern differences across these categories.

Dataset Details

- Categories: [class count]
- Sequence Length: [sample length] time points

Time Series Samples (2 samples per category):

Category 1:

- Sample 1: [sample for category 1]
- Sample 2: [sample for category 1]

...

Category k :

- Sample 1: [sample for category k]
- Sample 2: [sample for category k]

Analysis Task

Compare and summarize the significant differences in the time series patterns across categories based on the following characteristics. Explicitly state if no differences are observed. Break the series into meaningful segments (e.g. early, middle, late) if applicable.

Answer Format

- Trend Differences: [Describe trends (upward/downward) and how trends differ across categories, or state if no trends are observed.]
- Cyclic Behavior Differences: [Describe differences in cyclic or periodic patterns, or state if none are found.]
- Stationarity Differences: [Describe stability or shifts in the time series, or state if none are found.]
- Amplitude Differences: [Compare constant or fluctuating amplitudes, or state if no differences]
- Rate of Change Differences: [Describe the speed of change across categories (rapid, moderate, slow), or state if none are found.]
- Outliers Differences: [Identify distinct outliers or anomalies, or state if none are found.]

627

2nd Round Reasoning Prompt: Plug-in Model Fusion Reasoning

Task Description

You are given a time series classification task with the [dataset name] dataset, [domain-specific knowledge of the dataset]. Your second task is to analyze the time series data and refine your understanding based on the classification results and logits (model probabilities for each category) provided by a domain-specific model.

Dataset Details

- Categories: [class count]
- Sequence Length: [sample length] time points

Model Details

- Classification Accuracy: [performance of plug-in model (%)]

628

Classification Examples

- Case 1: True Label: [ground truth], Model Result: [plug-in model's prediction], Category Logits: [plug-in model's logits], Time Series Sample: [time series sample]
- Case 2: True Label: [ground truth], Model Result: [plug-in model's prediction], Category Logits: [plug-in model's logits], Time Series Sample: [time series sample]
- Case 3: True Label: [ground truth], Model Result: [plug-in model's prediction], Category Logits: [plug-in model's logits], Time Series Sample: [time series sample]

Analysis Task

Refine your understanding of the time series patterns, considering the model's classification results and logits. Identify any necessary adjustments to your initial analysis. **### Answer Format**

- Classification Analysis: [Evaluate the logits' confidence and alignment with categories.]
- Time Series Understanding Adjustment: [Adjust your understanding of time series patterns based on the model's results.]

629

3rd Round Reasoning Prompt: Integrative Step-by-step Reasoning

Task Description

Based on your refined understanding, your third task is to perform the time series classification task on the new data sample. You will use your updated analysis of time series patterns along with the result and category logits (model probabilities for each category) from the domain-specific model to make a final classification decision.

Dataset Details

- Categories: [class count]
- Sequence Length: [sample length] time points

Model Details

- Classification Accuracy: [accuracy of plug-in model %]

Classification Task

- Task: Model Result: [plug-in model's prediction], Category Logits: [plug-in model's logits], Time Series Sample: [time series sample]

Please think step by step:

- Analyze the Time Series Pattern: [Examine the time series data for trends, cyclic behavior, stationarity, amplitude, rate of change, and outliers. Compare these characteristics across the categories to identify any significant patterns or differences.]
- Interpret the Model's Results: [Evaluate the model's classification result and logits. Assess the confidence level of the model's prediction and how well it aligns with the observed time series patterns.]
- Make a Preliminary Prediction: [Based on your analysis of the time series pattern and the model's results, make an initial classification decision. Provide a brief explanation for this decision.]
- Review Alternative Classifications: [Consider if there are any other plausible categories that could fit the observed time series pattern. Evaluate the strengths and weaknesses of these alternative classifications compared to your initial prediction.]
- Final Classification Decision: [After reviewing all possibilities, make your final classification decision.]
- True Label: [Your Final Classification Result]

630

631 B.2 Illustrative Rationales Generated by ReasonTSC with DeepSeek, Llama, and GPT

632 In this subsection, we demonstrate illustrative generations in the third reasoning round from
633 ReasonTSC integrated with DeepSeek, Llama, and GPT. Specifically, these illustrative genera-
634 tions cover three representative cases: (1) ReasonTSC with DeepSeek identifies and corrects the
635 plug-in model's biased prediction by analyzing its behavioral tendency; (2) ReasonTSC with Llama
636 initially agrees with the plug-in model's prediction but subsequently overrides it after detecting closer
637 logit values and more representative temporal patterns in category 6; (3) ReasonTSC with GPT main-
638 tains consistency with the plug-in model's final prediction after analysis of temporal characteristics
639 and the category-wise logit distributions.

ReasonTSC with DeepSeek identifies and corrects the plug-in model's biased prediction by analyzing its behavioral tendency.

Ground Truth = 6, Plug-in Model Prediction = 4, LLM Prediction = 6

=====Analyze the Time Series Pattern=====

– Spike Timing: The significant spike (9.134) occurs at positions 15–23 (mid-series), characteristic of Category 6* (initial analysis showed mid-series spikes for Cat 6 vs. late spikes for Cat 4). – Amplitude: The spike magnitude (9.134) aligns with Category 6's highest observed outlier (9.13 in initial samples). – Post-Spike Behavior: Smaller oscillations (1.156, 0.745, etc.) resemble Category 6's gradual decline after a spike.

=====Interpret the Model's Results=====

– The model predicts Category 4 (logit=2.0) but shows moderate support for Category 6 (logit=0.98). This mirrors previous misclassifications (e.g., Case 3) where the model prioritized amplitude over spike timing.

=====Make a Preliminary Prediction=====

– The mid-series spike strongly matches Category 6's defining pattern. **The model's bias toward Category 4 likely stems from overemphasizing spike magnitude rather than temporal position.**

=====Review Alternative Classifications=====

– Category 4: Plausible due to high spike amplitude, but its defining late-spike pattern (e.g., positions 80–90 in initial samples) is absent here. – Category 3: Rejected due to lack of irregular scattered spikes. – Category 1/7: No sustained flat baseline or post-spike oscillations.

=====Final Classification Decision=====

– Despite the model's output, the spike timing and post-spike dynamics align unambiguously with Category 6.

640

ReasonTSC with Llama initially aligns with the plug-in model's prediction but subsequently overrides it after detecting category 6's closer logit values and more representative temporal patterns.

Ground Truth = 6, Plug-in Model Prediction = 3, LLM Prediction = 6

=====Analyze the Time Series Pattern=====

The given time series data shows a constant trend until around the 50th time point, where it significantly increases (to 9.607) and then decreases back to the baseline level. This pattern is indicative of a specific event or anomaly. The amplitude of the time series is generally low except for the mentioned spike. The rate of change is moderate, with a rapid increase and decrease around the 50th time point. There are no clear cyclic patterns, and the series appears to be stationary except for the outlier.

=====Interpret the Model's Results=====

The model predicts Category 3 with a logit of 1.02, which is the highest among all categories but not significantly higher than Category 6 (logit of 0.78) or Category 4 (logit of 0.81). This suggests that while the model has some confidence in its prediction, there is ambiguity, and other categories could also be plausible.

=====Make a Preliminary Prediction=====

Based on the time series pattern and the model's results, my initial classification decision is Category 3. The reasoning behind this decision is the model's prediction and the presence of a significant spike in the time series, which might align with patterns observed in Category 3. However, the confidence in this prediction is not high due to the model's ambiguity and the similarity of the time series pattern to patterns that could be observed in other categories.

=====Review Alternative Classifications=====

Considering the time series pattern, Category 6 could also be a plausible classification. **The pattern of a significant increase followed by a return to baseline could fit the characteristics of Category 6, as observed in previous analyses. Additionally, the model's logit for Category 6 (0.78) is not significantly lower than that of Category 3, suggesting that the model considers Category 6 as a possible classification as well.**

=====Final Classification Decision=====

641

After reviewing the time series pattern, the model’s results, and considering alternative classifications, my final classification decision is Category 6. The reason for choosing Category 6 over Category 3 is the closer alignment of the time series pattern with the characteristics observed in Category 6, such as the initial increase followed by a decrease back to the baseline level. Although the model’s prediction was Category 3, the ambiguity in the model’s results and the closer match of the time series pattern to Category 6 characteristics lead to this final decision.

ReasonTSC with GPT maintains consistency with the plug-in model’s final prediction after analysis of temporal characteristics and the category-wise logit distributions.

Ground Truth = 1, Plug-in Model Prediction = 1, LLM Prediction = 1

===== Interpret the Model’s Results =====

The model predicted Category 1. The logits are [2.62, -1.15, -1.37]. The positive logit for Category 1 (2.62) indicates strong confidence in this classification, while the negative logits for Categories 2 and 3 suggest that the model sees these categories as unlikely given the observed time series.

===== Make a Preliminary Prediction =====

Based on the analysis of the time series pattern, which shows high fluctuations, rapid changes, and cyclic behavior consistent with Category 1, along with the model’s strong confidence in this classification, I would initially classify this time series as Category 1.

===== Review Alternative Classifications =====

Category 2: This category is characterized by more stable and linear patterns. Given the high fluctuations and rapid changes in the observed time series, Category 2 does not seem plausible. Category 3: While there are some similarities in trends, Category 3 is generally more stable with less pronounced fluctuations. The model’s logits indicate that Category 3 is unlikely, as evidenced by the negative logit value. Strengths and Weaknesses: Strengths of Category 1: The observed time series aligns well with the characteristics of Category 1, and the model’s logits support this classification. Weaknesses of Category 3: Although there are some overlapping patterns, the overall behavior of the time series is more consistent with Category 1.

===== Final Classification Decision =====

After reviewing all possibilities and considering the strong alignment of the time series with the characteristics of Category 1, I confirm the classification as Category 1.

C Implementation Details of Our Proposed ReasonTSC

This section presents the implementation details of ReasonTSC. Specifically, Subsection C.1 describes the representative datasets selected from the UCR/UEA archive. Subsection C.2 describes the mainstream LLMs employed for evaluating ReasonTSC, and Subsection C.3 introduces existing time series foundation models.

C.1 Dataset Details of UCR/UEA Archive

We evaluate ReasonTSC on 15 representative time series datasets, 9 from the UCR archive and 6 from the UEA archive. These datasets cover diverse domains and vary in key characteristics (e.g., number of classes, time series length) to assess the performance stability of ReasonTSC (Subsection D.4). Detailed information about these datasets is summarized in Table 4.

C.2 Details of Adopted LLMs

This subsection introduces the mainstream LLMs evaluated in our study. We examine six representative model series: Gemini, Llama3, GPT, DeepSeek, and Grok3, which span diverse parameter scales from 8B (GPT-4o-mini) to 671B (DeepSeek). To assess the impact of ReasonTSC’s TSC-tailored CoT, we categorize these models into two groups: those enhanced with reasoning training techniques and those without. It is worth noting that powerful reasoning LLMs such as GPT-o1 are excluded from evaluation due to high per-token pricing. However, based on experimental results from the

Table 4: The dataset details of the UCR/UEA Archive.

Dataset	Type	Train Size	Test Size	Classes	Length	Domain
DistalPhalanxTW	IMAGE	400	139	6	80	Medical
MiddlePhalanxTW	IMAGE	399	154	6	80	Medical
MiddlePhalanxOutline	IMAGE	400	154	3	80	Medical
AgeGroup						
MedicalImages	IMAGE	381	760	10	99	Medical
ElectricDevices	DEVICE	8926	7711	7	96	Energy
BME	SIMULATED	30	150	3	128	Shape
ArrowHead	IMAGE	36	175	3	251	Cultural
DodgerLoopDay	SENSOR	78	80	7	288	Traffic
CBF	SIMULATED	30	900	3	128	Shape
RacketSports	HAR	151	152	4	30	Sports
ERing	HAR	30	270	6	65	Gesture
NATOPS	HAR	180	180	6	51	Gesture
Libras	HAR	180	180	15	45	Gesture
Epilepsy	HAR	137	138	4	207	Medical
PenDigits	MOTION	7494	3498	10	8	Handwriting

16 evaluated LLMs, GPT-o1 has the potential to achieve comparable or even superior classification accuracy when integrated with ReasonTSC. An overview of these LLMs is provided in Table 5.

Table 5: Overview of the sixteen mainstream LLMs integrated with ReasonTSC for evaluation.

LLM	Parameters	Reasoning-enhancing post-training	Developer	Release
Gemini-2.5-pro	175B	✓	Google	2025-03
Gemini-2.0-flash	Unknown	✓	Google	2024-12
Llama-3.3-70b-instruct	70B	✗	Meta	2024-12
Llama-3.1-405b-instruct	405B	✗	Meta	2024-07
GPT-4o-mini	8B	✗	OpenAI	2024-07
GPT-4o	200B	✗	OpenAI	2024-11
GPT-3.5-turbo	20B	✗	OpenAI	2024-01
DeepSeek-V3	671B	✗	DeepSeek	2024-12
DeepSeek-R1	671B	✓	DeepSeek	2024-01
Grok3	Unknown	✗	xAI	2025-02
Claude-3.7-sonnet	120B	✓	Anthropic	2025-02
Claude-3.5-haiku	175B	✗	Anthropic	2024-10
Qwen3-235b-a22b	235B	✓	Alibaba	2025-04
Qwen3-30b-a3b	30B	✓	Alibaba	2025-04
Qwen3-32b	32B	✓	Alibaba	2025-04
Qwen2.5-72B-Instruct	72B	✗	Alibaba	2024-09

C.3 Details of Adopted Time-Series Foundation Models

In this subsection, we briefly introduce the current time series foundation models (TSFMs), with detailed comparisons provided in Table 6. Our analysis reveals three observations: (1) TSFMs generally adopt large language models as their backbone and primarily process single-modality time series data for both input and output, lacking natural language interaction capabilities. (2) Most TSFMs focus on time series forecasting, while tasks such as classification, anomaly detection and imputation remain unexplored. (3) The reasoning ability of TSFMs are largely unexplored. Current research has not adequately determined whether the success of TSFMs stems from memorizing training patterns or genuine reasoning abilities

Transformer architectures have demonstrated state-of-the-art performance in diverse time series tasks [66, 67]. Inspired by the success of pretrained LLMs, researchers have shifted toward developing time series foundation models (TSFMs) with zero-shot or few-shot capabilities [68], which generally fall into three architectural categories: encoder-only, decoder-only, and encoder-decoder structures. For instance, MOMENT [28], an encoder-only TSFM based on T5, supports diverse time series tasks such

as forecasting, classification and anomaly detection. Similarly, MOIRAI [20] is a masked encoder-based universal time series forecaster designed for zero-shot tasks. In contrast, decoder-only models possess the ability for iterative generation. Given sequential input patches, they autoregressively predict the next patch conditioned on all preceding ones. Representative examples include TimesFM [59], featuring a decoder-style attention architecture with input patching, and Timer [58], which leverages an autoregressive approach for generative pre-training. Besides, Chronos [49] primarily focus on the variants of the encoder-decoder T5 model, and uses simple scaling and quantization to tokenize time series into discrete bins.

Given that pre-trained TSFMs incur substantial computational costs, some studies leverage language models pre-trained on billions of tokens for time series analysis. GPT4TS [21] and LLM4TS [69] freeze the pretrained blocks and fine-tune positional embeddings and layer normalization, achieving comparable performance in time series tasks. Time-LLM [22] introduces a reprogramming framework that bridges the modality gap between time series data and natural language. Similarly, UniTime [70] is a cross-domain learning approach that utilizes domain instructions and a Language-TS Transformer to provide identification information. Additionally, TEMPO [71] introduces an interpretable, prompt-tuning-based GPT architecture to focus on leveraging knowledge from distinct temporal semantics components.

Table 6: Overview of time series foundation models.

Model	Base Architecture	Text as Input	Text Generation	Forecasting	Classification	Anomaly Detection	TS Reasoning
MOMENT [28]	T5 encoder	✗	✗	✓	✓	✓	✗
Chronos [49]	T5 (encoder-decoder)	✗	✗	✓	✗	✗	✗
MOIRAI [20]	Encoder-only Transformer	✗	✗	✓	✗	✗	✗
TimesFM [59]	Decoder-only	✗	✗	✓	✗	✗	✗
Timer [58]	Decoder-only	✗	✗	✓	✗	✓	✗
GPT4TS [21]	GPT2	✗	✗	✓	✓	✓	✗
LLM4TS [69]	GPT2	✗	✗	✓	✗	✗	✗
Time-LLM [22]	Llama-7B	✓	✗	✓	✗	✗	✗
UniTime [70]	GPT2	✓	✗	✓	✗	✗	✗
TEMPO [71]	GPT2	✗	✗	✓	✗	✗	✗

D Additional Experimental Results

In this section, we present extended experimental analyses to complement the discussions in Subsections 3.3 and 3.4. Subsection D.1 compares ReasonTSC’s few-shot performance with traditional full-shot baselines. Subsection D.2 evaluates the impact of ReasonTSC’s tailored Chain-of-Thought on performance improvement. Subsection D.3 investigates how the plug-in model’s ICL examples affect ReasonTSC. Subsection D.4 analyzes the impact of dataset characteristics (e.g., number of classes, tokens, and series length) on ReasonTSC. Subsection D.5 provides the full results of ReasonTSC with mainstream LLMs on three selected datasets.

D.1 Comparing ReasonTSC with Traditional Full-shot Baselines

We compare the performance of ReasonTSC with the retraining/finetuning-from-scratch baselines that train/finetune time series classification models on the entire dataset. Unlike traditional task-specific methods that rely on the entire downstream task’s dataset for training and lack generalization, ReasonTSC requires only a few in-context learning demonstrations, enabling it to handle diverse time series tasks more effectively. As shown in Table 7, ReasonTSC with Llama and DeepSeek achieves comparable performance to traditional baselines in most subsets. Moreover, since ReasonTSC can reason about the plug-in model’s prediction behavior, it can outperform the plug-in model when a baseline model is used as the plug-in.

D.2 Comparing ReasonTSC’s Tailored CoT with Vanilla CoT

The proposed ReasonTSC employs a multi-turn reasoning with a fused decision-making strategy tailored to TSC. It steers the LLM to analyze time series patterns in the first reasoning round and guides the LLM to examine the prediction behavior of the plug-in model in the second reasoning round. In the third reasoning round, it reevaluates the initial assessment, backtracks to consider

Table 7: Performance comparison between ReasonTSC and traditional baselines. Note that all baseline methods are task-specific models trained on the entire training set for **100** epochs, whereas ReasonTSC utilizes only a few demonstration examples for in-context learning.

Model	Dist. TW	Mid. TW	Mid. OA	Elec.	Med. Img	BME	Arr. Hd	Dod. LD
Full-shot (# Training samples):	400	399	400	8926	381	30	36	67
TimesNet [72]	69.78	59.74	65.58	65.3	70.26	94.00	82.28	58.75
Autoformer [73]	64.74	57.79	64.28	66.78	62.89	84.00	55.43	32.50
FEDformer [74]	69.06	58.44	66.23	68.27	67.10	<u>94.67</u>	75.42	41.25
iTransformer [75]	70.50	62.98	64.28	60.03	73.16	95.33	80.00	55.00
PatchTST [76]	<u>71.94</u>	61.03	66.23	65.30	72.76	<u>94.67</u>	66.85	52.50
LightTS [77]	74.10	59.74	62.33	62.40	69.87	86.00	65.71	<u>61.25</u>
DLinear [78]	69.78	61.03	64.28	47.72	58.28	84.00	69.71	55.00
Two-shot (# Training samples):	12	12	6	14	20	6	6	14
ReasonTSC (Llama-3.3-70B-instruct)	64.03	59.09	61.04	61.18	<u>77.63</u>	86.00	66.86	57.14
ReasonTSC (DeepSeek-R1)	65.71	<u>61.69</u>	63.64	<u>67.11</u>	80.26	85.33	69.14	62.34

Model	CBF	Rkt. Spt	ERing	Nt.Ops	Lbr.	Eplp.	Pen.	Avg
Full-shot (# Training samples):	30	151	30	180	180	137	7494	
TimesNet [72]	<u>94.44</u>	76.97	71.85	69.44	58.33	79.71	90.10	73.76
Autoformer [73]	42.22	73.02	51.48	51.11	48.33	65.21	88.47	60.55
FEDformer [74]	51.44	72.36	64.07	65.56	58.33	71.73	88.82	67.51
iTransformer [75]	90.55	72.36	71.48	66.67	56.11	73.18	86.70	71.88
PatchTST [76]	90.00	71.05	75.18	58.33	53.33	90.57	87.79	71.83
LightTS [77]	83.55	68.42	70.00	<u>68.89</u>	52.22	62.31	<u>89.05</u>	69.05
DLinear [78]	78.44	57.89	68.89	65.55	35.00	36.22	71.09	61.52
Two-shot (# Training samples):	6	8	12	12	30	8	20	
ReasonTSC (Llama-3.3-70B-instruct)	95.33	61.84	<u>74.07</u>	66.67	51.11	<u>92.03</u>	86.99	70.73
ReasonTSC (DeepSeek-R1)	93.33	63.16	<u>74.07</u>	67.78	57.22	94.93	86.30	<u>72.80</u>

alternative hypotheses, and compares their merits before arriving at a final classification. As illustrated in Figure 8, ReasonTSC substantially outperforms vanilla CoT, with performance gains of 8.08% for GPT, 3.82% for Llama, and 7.18% for DeepSeek. Note that the performance gains mean the TSC performance improvements by ReasonTSC and Vanilla CoT strategies compared to plain LLMs.

We further analyze the performance of ReasonTSC with Llama compared to vanilla CoT by evaluating the impact of removing components from the plug-in model, as depicted in Figure 9. For both methods, removing either the plug-in model’s logits or the full model (including predictions and logits) results in significant performance improvements. Notably, our TSC-tailored CoT achieves improvement ratios over twice those of vanilla CoT, reaching 55.63% without logits and 51.45% without the plug-in model. This substantial performance gap further validates the effectiveness of ReasonTSC’s customized reasoning strategy for time series classification tasks.

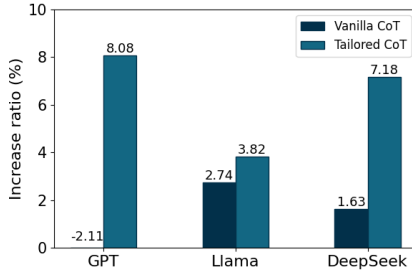


Figure 8: Comparison between ReasonTSC’s tailored CoT and vanilla CoT by GPT, Llama, and DeepSeek.

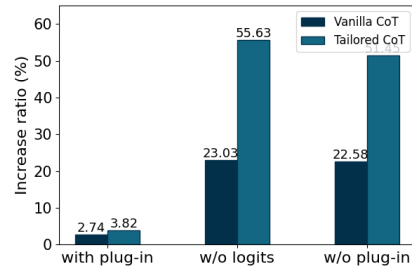


Figure 9: Comparison between ReasonTSC’s tailored CoT with Llama and vanilla CoT in terms of with/without plug-in model.

D.3 The Influence of Different ICL Examples in the Second Reasoning Round

In the second reasoning round of ReasonTSC, the plug-in model’s predictions and logits are served as in-context learning (ICL) examples for the LLM. Figure 10 illustrates the performance improvement ratio of ReasonTSC when incorporating ICL examples in the second reasoning turn compared to that when ICL examples are omitted. ReasonTSC randomly sampled one successful case and two failed cases. Figure 11 further investigates the influence of the success-failure ratio of the selected cases on ReasonTSC with Llama. The success case means the plug-in classifier’s prediction aligns with the ground truth, while the failed case means otherwise. We test four configurations: all-successful, majority-successful, majority-failed, and all-failed cases.

As illustrated in Figure 10, the prediction behaviors of plug-in models can significantly enhance ReasonTSC’s performance on certain datasets. ReasonTSC with Llama exhibits increase ratios of 37.23% (BME) and 12.59% (CBF), suggesting that ReasonTSC with Llama is more strongly influenced by plug-in model predictions than ReasonTSC with DeepSeek.

The success-failure ratio of ICL examples also influences ReasonTSC’s performance. By using three success cases as ICL examples, ReasonTSC with Llama outperforms the zero-shot setting. Notably, gradually introducing failure cases leads to consistent slight performance improvements, suggesting that the LLM enhances its understanding of time series patterns by analyzing the plug-in model’s biased behaviors. Conversely, relying solely on failure cases causes a substantial performance drop. This decline likely arises because negative examples mislead the LLM to reject the plug-in model’s valid predictions, yielding counterproductive results.

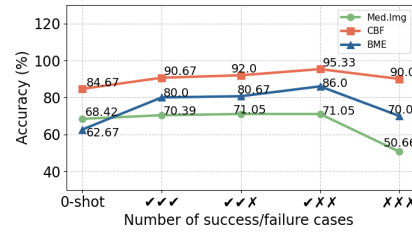
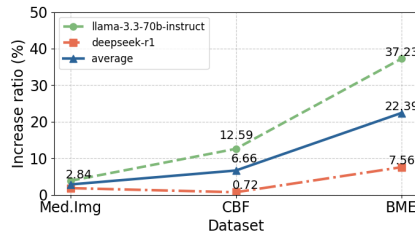


Figure 10: Performance improvement ratio of ICL examples in ReasonTSC’s second reasoning round. Figure 11: Impact of success-failure ratio in ICL examples on ReasonTSC’s second reasoning round

D.4 ReasonTSC’s Stability In Terms of Numbers of Classes and Tokens, and Series Length

We also investigate the impact of three key factors on ReasonTSC’s reasoning performance: category count, time series length, and token count, as shown in Figure 12. Regarding the number of classification categories, both ReasonTSC s with Llama and DeepSeek present stable performance. ReasonTSC with GPT exhibits a performance decline as the category count increases, suggesting that a smaller-scale language model faces limitations as the volume of the processed information increases. On the other hand, for sequence lengths less than 80 timestamps, ReasonTSC s with Llama and DeepSeek achieve only 3.38% and 8.19% performance improvements, respectively. This is because shorter time series samples contain fewer discernible patterns, which provides less information for LLM to understand TS. In terms of the number of tokens, ReasonTSC with GPT performs best with less than 6,000 tokens. ReasonTSC s with Llama and DeepSeek achieve the best performance with an input token count between 6,000 and 10,000, with improvement ratios of 6.39% and 12.42%, respectively.

D.5 Performances of ReasonTSC with Mainstream LLMs

Table 8 presents the complete performance of ReasonTSC integrated with sixteen mainstream LLMs, as discussed in Subsection 3.2. Notably, ReasonTSC substantially enhances DeepSeek-V3’s performance in the Mid.OA dataset, improving its accuracy from 27.92% to 66.88% compared to the vanilla CoT approach without plug-in models. Additionally, it enables DeepSeek-V3 to achieve performance comparable to DeepSeek-R1. Besides, gemini-2.5-pro (175B parameters) and claude-

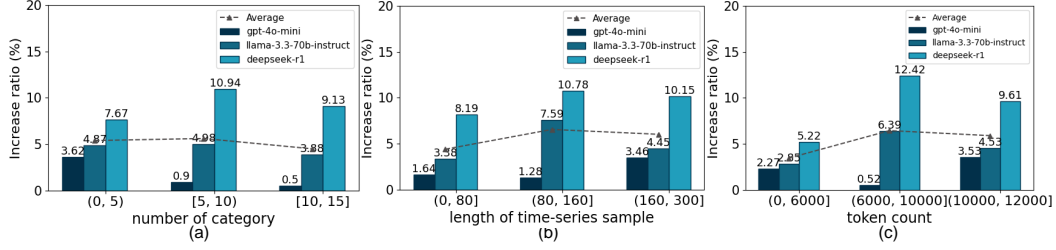


Figure 12: Average performance improvement of ReasonTSC compared to TSFMs across all the tested datasets. Three influence factors are considered: category count (a), time series length (b), and token count(c).

766 3.7-sonnet (120B parameters) demonstrate superior performance owing to their inherent reasoning
 767 capabilities acquired during reasoning-enhancing post-training.

Table 8: The performance of ReasonTSC with mainstream LLMs as reasoning language models on three UCR/UEA datasets (MiddlePhalanxOutlineAgeGroup, BME and ERing).

LLM	Parameter	Mid.OA	BME	ERing	Average
MOMENT (<i>plug-in</i>)	341M	60.39	74.00	72.59	68.99
Vanilla CoT (gemini-2.5-pro)	175B	37.01	79.33	57.78	58.04
Vanilla CoT (deepseek-v3)	671B	27.92	63.23	60.00	50.38
gpt-4o-mini	8B	61.04	77.33	74.81	71.06
gpt-3.5-turbo	20B	59.74	74.00	72.59	68.78
qwen3-30b-a3b	30B	61.69	77.18	74.07	70.98
qwen3-32b	32B	59.09	77.33	72.59	69.67
llama-3.3-70b-Instruct	70B	61.04	84.00	74.07	73.04
qwen2.5-72B-Instruct	72B	60.39	78.67	72.59	70.55
claude-3.7-sonnet	120B	62.99	82.00	77.04	74.01
claude-3.5-haiku	175B	60.53	77.18	71.85	69.85
gemini-2.0-flash	unknown	58.44	86.67	76.30	73.80
gemini-2.5-pro	175B	62.34	86.00	76.30	74.88
gpt-4o	200B	62.34	76.00	72.59	70.31
qwen3-235b	235B	63.64	83.33	74.07	73.68
llama-3.1-405b	405B	61.04	76.67	71.85	69.85
deepseek-v3	671B	66.88	88.00	69.63	<u>74.84</u>
deepseek-r1	671B	63.64	82.67	74.07	73.46
grok-3	314B	<u>64.29</u>	81.33	<u>76.87</u>	74.16

768 E Interpretation Study – Can ReasonTSC Reason About TS Patterns?

769 This section presents a comprehensive evaluation of how our proposed ReasonTSC reasons about TS
 770 patterns as discussed in Subsection 3.4.1. In the following, Subsection E.1 evaluates ReasonTSC’s
 771 ability to think over TS patterns with four synthetic datasets; Subsection E.2 conducts this investi-
 772 gation with the real-world UCR/UEA datasets. Additionally, Subsection E.3 provides illustrative
 773 rationales generated by ReasonTSC integrated with two mainstream LLMs.

774 E.1 Evaluation on Synthetic Datasets

775 **Synthetic Datasets Generation.** To evaluate ReasonTSC’s ability to interpret general TS patterns.
 776 For illustration, we generate four synthetic datasets, each containing 200 TS samples with one of
 777 the four pattern types (i.e., *Trend*, *Frequency*, *Amplitude*, and *Mixed Patterns*) with a fixed sequence
 778 length of 100. Following [46, 47], we devise a series of particular linear functions to simulate time
 779 series with the desired patterns. To be specific, the *trend* pattern is simulated by linear plots that
 780 exhibit obvious slopes $y(t) = \beta_0 + \beta_1 \cdot t + \epsilon(t)$; the *frequency* pattern is simulated by sine functions

781 with fixed frequencies and amplitudes $y(t) = A \cdot \sin(2\pi f \cdot t + \phi) + \epsilon(t)$; the *amplitude* pattern
 782 is reflected by sine functions with varying amplitudes $y(t) = A \cdot \sin(2\pi f \cdot t) + \epsilon(t)$; the *mixed*
 783 *patterns* pattern is captured by integrating multiple sine functions with varying frequencies and
 784 phases to illustrate cases where various complex TS patterns are presented within adjacent samples
 785 $y(t) = \beta_0 + \beta_1 \cdot t + A_1 \cdot \sin(2\pi f_1 \cdot t + \phi_1) + A_2 \cdot \cos(2\pi f_2 \cdot t + \phi_2) + \epsilon(t)$. Besides, we also construct
 786 plain time series samples without distinguishable features compared with the aforementioned patterns
 787 by introducing Gaussian noise to the time series, serving as negative counterparts.

788 **Evaluation Process.** We present each time-series sample in a multiple-choice format alongside a
 789 randomly generated noise sequence, prompting the ReasonTSC to identify the sequence with the
 790 most discernible patterns. The choice positions are randomized to eliminate positional bias. The
 791 prompt templates are illustrated below.

Trend

Compare the two provided time series samples and select the one that demonstrates a more typical and well-defined trend pattern, specifically a sustained and clear directional trend (either upward or downward) throughout the series.

- Case A: [time series sample]
- Case B: [time series sample]

Answer Format:

- Option: [Case A or Case B]
- Explanation: [Reason for choosing this time series sample and the specific pattern observed]

792

Frequency

Compare the two provided time series samples and select the one that exhibits a more typical and well-defined frequency or cyclical pattern, characterized by consistent and regular periodic behavior or repetitive cycles throughout the series.

- Case A: [time series sample]
- Case B: [time series sample]

Answer Format:

- Option: [Case A or Case B]
- Explanation: [Reason for choosing this time series sample and the specific pattern observed]

793

Amplitude

Compare the two provided time series samples and select the one that demonstrates a more typical and well-defined amplitude pattern, characterized by consistent and pronounced variations in value range, indicative of strong oscillations or signal intensity.

- Case A: [time series sample]
- Case B: [time series sample]

Answer Format:

- Option: [Case A or Case B]
- Explanation: [Reason for choosing this time series sample and the specific pattern observed]

794

Mixed Patterns

Compare the two provided time series samples and select the one that exhibits more typical and well-defined patterns, such as trends, seasonality, or cyclical behavior.

- Case A: [time series sample]
- Case B: [time series sample]

Answer Format:

- Option: [Case A or Case B]
- Explanation: [Reason for choosing this time series sample and the specific pattern observed]

795

796 **ReasonTSC’s Performance of Understanding TS patterns.** Table 9 presents the experimental
 797 results. Notably, ReasonTSC with GPT, Llama, and Deepseek achieve satisfactory accuracy across

all the tested datasets, demonstrating ReasonTSC’s ability to generate rationales about fundamental time series patterns.

Table 9: Pattern Recognition Accuracy of LLMs employed in ReasonTSCon synthetic data

LLM	Trend	Frequency	Amplitude	Mixed Patterns
GPT-4o-mini	100%	100%	100%	100%
Llama-3.3-70b-instruct	100%	100%	100%	99.50%
DeepSeek-R1	100%	100%	100%	100%

E.2 Evaluation on the UCR/UEA Archive

Data Preparation. We further employ the UCR/UEA archives for interpretation analysis. For each sample, we randomly select one unique time series instance per category, generating a maximum of 100 samples per dataset. Datasets with fewer than 30 unique samples are excluded.

Evaluation Process. In this study, we extend six fundamental time-series patterns to ten: *trend*, *cyclic*, *stationarity*, *amplitude*, *rate of change*, *outliers*, *noise*, *volatility*, *structural break*, and *mean shift* [46]. We then prompt the ReasonTSC to identify significant pattern variations across categories. The complete prompt details are provided below.

Pattern Interpretation Prompt Template on the UCR/UEA Archive

Task Description

You are given a time series analysis task with the [dataset name] dataset, [domain-specific knowledge of the dataset]. Your task is to analyze and determine whether there are any highly pronounced and distinctly typical temporal patterns across these categories. Only if such patterns are exceptionally clear and consistently representative, mark it as 1; otherwise, mark it as 0.

Dataset Details

- Categories: [class count]
- Sequence Length: [sample length] time points

Time Series Samples by Category

- Category 1: [sample for category 1]
- Category 2: [sample for category 2]

...

- Category k : [sample for category k]

Analysis Task

Compare and summarize the significant differences in the time series patterns across categories based on the following characteristics. Break the series into meaningful segments (e.g. early, middle, late) if applicable. Only mark a characteristic as 1 if the differences are very clear and typical. Explicitly state if no differences are observed.

Answer Format

- Trend Differences: 0/1. [Describe clear and typical trends (upward/downward) and how they differ across categories, or state if none are found.]
- Cyclic Behavior Differences: 0/1. [Describe clear and typical differences in cyclic or periodic patterns, or state if none are found.]
- Stationarity Differences: 0/1. [Describe clear and typical stability or shifts in the time series, or state if none are found.]
- Amplitude Differences: 0/1. [Describe clear and typical constant or fluctuating amplitudes, or state if none are found.]
- Rate of Change Differences: 0/1. [Describe clear and typical differences in the speed of change (rapid, moderate, slow), or state if none are found.]
- Outliers Differences: 0/1. [Identify clear and typical distinct outliers or anomalies, or state if none are found.]
- Noise Level Differences: 0/1. [Describe clearly and typically the amount of random fluctuations or noise across categories, or state if none are found.]
- Volatility Differences: 0/1. [Describe clear and typical differences in variability or fluctua-

tions, or state if none are found.]

- Structural Break Differences: 0/1. [Identify clear and typical significant shifts or breaks in the time series, or state if none are found.]
- Mean Level Differences: 0/1. [Identify clear and typical average values across categories, or state if none are found.]

809

810 **ReasonTSC’s Performance of Thinking Over TS Patterns.** Table 10 presents our experimen-
811 tal results, with the top three most frequently identified patterns (including ties) highlighted in
812 bold. GPT-4o-mini consistently identifies similar temporal patterns (e.g., trend, amplitude, rate
813 of change, volatility, and mean shift) across all datasets, suggesting that smaller-scale LLMs
814 tend to generate more generalized interpretations. This observation aligns with its performance
815 within the ReasonTSC framework. DeepSeek-R1 shows superior capability in identifying category-
816 discriminative patterns. Llama3.3-70b-instruct shows comparable capability, with significant pattern
817 recognition overlap between Llama and DeepSeek, further validating LLMs’ temporal reasoning
818 capacities. These observations suggest that an in-depth understanding of the time series patterns
819 could enhance the reasoning process of LLMs and time series classification performance.

Table 10: Pattern Interpretation of LLMs in ReasonTSC on UCR/UEA Archive

Dataset	Samples	LLM	Trend	Cyclic	Stationarity	Amplitude	Rate of Change	Outliers	Noise	Volatility	Structural Break	Mean
BME	60	GPT	60	0	60	60	60	4	37	60	25	60
		Llama	60	0	52	60	60	1	0	59	60	60
		DeepSeek	53	0	20	37	47	6	2	22	59	48
Arr. Hd	65	GPT	65	0	65	65	65	0	30	65	15	65
		Llama	7	0	1	8	7	0	0	8	2	15
		DeepSeek	15	1	3	52	47	2	25	39	8	31
CBF	100	GPT	100	0	100	100	100	76	55	100	61	100
		Llama	45	1	3	56	45	10	1	43	41	96
		DeepSeek	59	0	23	81	67	16	0	15	95	99
Mid. OA	84	GPT	84	0	83	84	84	0	2	84	40	84
		Llama	31	2	31	31	31	1	2	31	31	34
		DeepSeek	23	3	32	44	59	32	27	51	59	34
Rkt. Spt	68	GPT	68	0	67	68	68	58	10	68	16	68
		Llama	42	2	9	68	61	25	4	61	37	68
		DeepSeek	50	0	10	56	60	40	27	56	55	49
Eplp.	60	GPT	60	0	60	60	60	60	43	60	25	60
		Llama	35	2	4	60	43	10	5	58	8	59
		DeepSeek	18	56	57	60	60	54	41	60	54	58
Mid. TW	34	GPT	34	0	34	34	34	0	3	34	7	34
		Llama	0	0	0	4	0	0	0	2	1	5
		DeepSeek	31	1	26	30	34	17	22	31	34	33
ERing	50	GPT	50	0	50	50	50	44	8	50	43	50
		Llama	50	0	35	50	50	0	1	50	50	50
		DeepSeek	48	0	21	45	50	10	22	45	49	47
Nt.Ops	60	GPT	60	0	60	60	60	59	2	60	60	60
		Llama	60	1	50	60	60	19	2	60	60	60
		DeepSeek	59	0	59	59	59	38	7	59	59	56
Elec.	100	GPT	85	34	84	85	85	85	36	85	48	85
		Llama	78	34	73	99	97	98	28	98	86	99
		DeepSeek	15	75	65	99	69	88	56	100	96	94
Pen.	100	GPT	99	0	98	99	99	97	0	99	25	99
		Llama	10	8	8	94	87	8	8	61	9	90
		DeepSeek	93	0	9	12	94	1	46	90	70	98

820 E.3 Illustrative Rationales about TS Patterns by ReasonTSC on the UCR/UEA Archive

821 We further provide illustrative rationales for time-series patterns generated by Llama-3-70b-instruct
822 and DeepSeek-R1 on the ElectricDevices (Figure 13) and ERing datasets (Figure 14), respectively.

Time-series pattern rationales of Llama-3.3-70b-instruct on the ElectricDevices dataset

To analyze the time series patterns across the given categories, we’ll break down the characteristics as requested. – **Trend Differences:** 0. Upon examining the series, no clear

823

and consistent upward or downward trends are observed across all categories that distinctly differentiate them. – **Cyclic behavior Differences:** 0. There’s no evident cyclic or periodic behavior that clearly and typically differentiates the categories. – **Stationarity Differences:** 0. The series generally appear to be non-stationary due to the presence of spikes and shifts, but the differences in stationarity are not clear and typical across categories. – **Amplitude Differences:** 1. There are clear differences in amplitude across categories. **Category 4 has a very low amplitude until a significant spike, whereas Category 6 shows a mix of low and high amplitude values.** – **Rate of change Differences:** 1. The speed of change varies significantly across categories. For instance, **Category 1 remains almost constant, while Category 5 shows rapid changes between two distinct levels.** Category 7 has segments with slow changes followed by rapid spikes. – **Outliers Differences:** 1. Outliers or anomalies are present and differ across categories. For example, **Category 3 has a significant outlier at the 25th position, and Category 6 has a large spike.** – **Noise level Differences:** 0. While there are fluctuations, the level of noise does not clearly and typically differentiate the categories. – **Volatility Differences:** 1. Volatility, or the variability in the series, differs across categories. **Category 1 is very stable, whereas Category 5 shows high variability between its two main states.** – **Structural Break Differences:** 1. Significant shifts or breaks in the time series are observed and differ across categories. For example, **Category 4 has a structural break with a large spike after a long period of stability.** – **Mean Level Differences:** 1. **The average values across categories differ significantly.** For instance, Category 1 has a mean close to -0.179, while Category 6 has a higher mean due to its positive spikes. Category 7’s mean is influenced by its mix of negative and positive values.

824

Time-series pattern rationales of DeepSeek-R1 on the ERing dataset

– **Trend Differences:** 1. Categories exhibit distinct trend patterns. For example, **Category 1 shows a V-shaped trend (sharp decline followed by recovery), Category 4 has a single pronounced peak and trough, and Category 5 displays a gradual decline followed by a sharp drop.** – **Cyclic behavior Differences:** 0. No consistent cyclic or periodic patterns (e.g., repeated oscillations) are observed across categories. While Category 3 has multiple abrupt shifts, these are not cyclical but rather irregular structural breaks. – **Stationarity Differences:** 0. All categories exhibit non-stationary behavior (e.g., shifts in mean/variance). – **Amplitude Differences:** 1. Amplitude ranges vary significantly. For instance, **Category 2 spans 3.8 (from 1.191 to -1.89), while Category 4 spans 2.0 (from -0.759 to 1.285).** These differences in magnitude are pronounced and category-specific. – **Rate of change Differences:** 1. Categories differ in how abruptly values change. **Category 1 and Category 4 have rapid drops, Category 5 shows a slow initial decline followed by a sharp drop, and Category 3 has frequent abrupt shifts.** – **Outliers Differences:** 0. No distinct outliers are observed; abrupt changes align with structural breaks rather than anomalies. – **Noise level Differences:** 0. All categories exhibit low noise, with values often plateauing for multiple time points. Random fluctuations are minimal and consistent across categories. – **Volatility Differences:** 1. Variability differs: **Category 3 and Category 2 show high volatility (frequent large shifts), while Category 6 and Category 4 have stable segments followed by single sharp changes.** – **Structural Break Differences:** 1. Structural breaks (sudden shifts) are prominent but differ in timing and frequency. For example, **Category 3 has multiple breaks, while Category 6 has one dominant drop.** – **Mean Level Differences:** 1. Mean values differ clearly. Categories like Category 1 and 6 start at higher values (2.69 and 1.085), while Category 3 and 5 spend significant time at lower means (e.g., -1.384 in Category 5).

825

F Limitations and Discussions

826

ReasonTSC is a novel framework designed to effectively leverage LLM reasoning for time series classification through a multi-turn tailored reasoning and fused decision-making strategy. However, we also recognize several limitations of this work. First, unlike NLP and vision datasets, time series data typically consists of long sequences with sparse semantics and inherent noise. Due to the context length limits of LLMs, effectively conveying the original time series sequence within a multi-turn

827

828

829

830

831

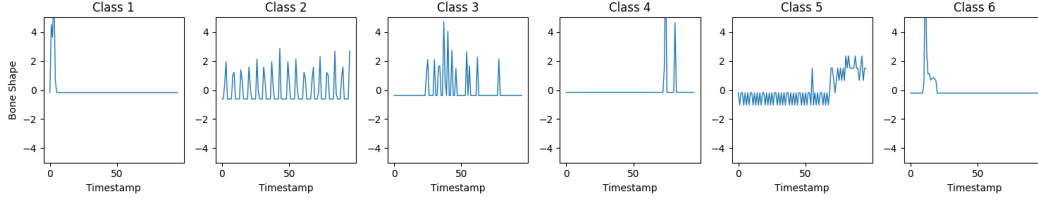


Figure 13: Visualization of Class Distribution in the ElectricDevices Dataset.

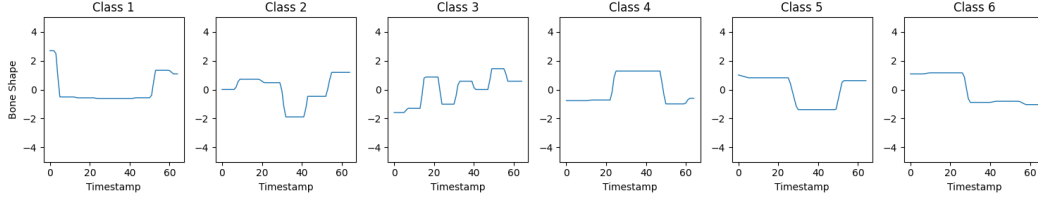


Figure 14: Visualization of Class Distribution in the ERing Dataset.

832 prompting framework remains challenging. Second, although the proposed ReasonTSC is more
833 cost-efficient than pretraining a time series LLM from scratch, calling APIs of reasoning LLMs still
834 incurs computational overhead. Third, our experiments adhere to the original training-test splits
835 of the UCR/UEA archive. In contrast, different dataset split ratios may impact the performance of
836 plug-in models, which could also indirectly influence the performance of ReasonTSC.

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: The abstract and introduction clearly state the paper’s contribution: the ReasonTSC framework, which uses multi-turn reasoning and fused decision-making to adapt LLMs for time series classification. The claims are validated by experimental results.

Guidelines:

- The answer NA 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 NA 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: The paper discusses limitations in the Appendix.

Guidelines:

- The answer NA 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: [No]

Justification: The paper focuses on applying LLMs to time series reasoning and does not present theoretical results. Therefore, it includes no theoretical assumptions or proofs. The work is empirically validated, with experimental results supporting the proposed framework.

Guidelines:

- The answer NA 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: We clearly documents the experimental settings, including data sources, evaluation metrics, and pre-training details for the ReasonTSC framework. We also list our full prompt and details in the Appendix.

Guidelines:

- The answer NA 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: The paper provides open access to the code and data via an anonymous GitHub repository, as stated in the abstract. Detailed instructions for reproduction are included in the supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper clearly specifies the experimental setup, including data splits, hyperparameters, model configurations, and evaluation protocols.

Guidelines:

- The answer NA 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: The paper reports detailed experimental conditions (including datasets and model configurations), with complete results provided in the Appendix.

Guidelines:

- The answer NA 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: Yes. The paper specifies the GPU training environment and details for fine-tuning time-series foundation models.

Guidelines:

- The answer NA 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 strictly follows the NeurIPS Code of Ethics. All experiments comply with ethical standards regarding data usage, privacy, and fairness.

Guidelines:

- The answer NA 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: [No]

Justification: The paper focuses on the reasoning framework design and novel applications. As such, the paper does not directly address societal implications.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: This work does not release new models, but utilizes existing open-source pretrained language models within our reasoning framework.

Guidelines:

- The answer NA 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: All assets (datasets, code, models) are properly cited, and their licenses/terms of use are respected, as documented in the paper.

Guidelines:

- The answer NA 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: The paper introduces a new codebase implementing the ReasonTSC framework, which is fully documented with instructions for reproduction, training, and evaluation. The documentation is provided alongside the released code.

Guidelines:

- The answer NA 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: [No]

Justification: This work does not involve any human subject experiments or crowdsourcing studies.

Guidelines:

- The answer NA 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: [No]

Justification: This study does not involve human participants, so no risk assessment or IRB approval was required.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper provides detailed descriptions of LLM usage as it constitutes a core methodological component of this research.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.