`t-BEN`: A Temporal Logic Guided Approach for Temporal Reasoning Benchmark Generation

Dingmin Wang[*]
University of Oxford, United Kingdom
dingmin.wang@cs.ox.ac.uk

Bocheng Zou[*]
University of Wisconsin-Madison, USA
bochengz@cs.wisc.edu

Zhen Han
LMU Munich, Munich, Germany

Zhiqiang Xu
MBZUAI, Abu Dhabi, UAE

Abstract

In logic-based Artificial Intelligence (AI), temporal reasoning typically involves formalizing problems as logical rule expressions and employing symbolic reasoners to infer and derive new conclusions from structured knowledge. However, symbolic reasoners generally cannot process natural language directly and require manually constructed symbolic knowledge bases, which can be both time-consuming and resource-intensive to create and maintain. Given the recent widespread adoption of Large Language Models (LLMs) and their remarkable successes across diverse domains, we are motivated to explore to what extent LLMs can handle temporal logic tasks, dispensing with traditional symbolic reasoners.

We introduce `t-BEN`, a benchmark suite that strictly adheres to the semantics of temporal logic. It automatically synthesizes temporal reasoning datasets in both symbolic and natural language forms, enabling the evaluation of Large Language Models (LLMs) on temporal logic reasoning. `t-BEN` is a highly scalable benchmark that supports the generation of datasets with varying sizes and rule structures of varying complexity. Furthermore, each question in `t-BEN` is guaranteed to be unseen by LLMs during pretraining, effectively minimizing the risk of data leakage. Our results, along with a detailed ablation study of seven frontier LLMs, offer valuable insights into the capabilities and limitations of current models in temporal logic reasoning tasks. Our generated datasets are available at `https://huggingface.co/datasets/BochengZou/t-BEN`.

1 Introduction

Temporal logic reasoning problems, grounded on formal logical rules, have been studied for decades in the field of logic-based Artificial Intelligence Alur and Henzinger [1994], Venema [2017], Lamport [1980]. Predominant approaches to solving these problems typically rely on reasoners that are specific to particular logical languages, such as MeTeoR Wang et al. [2022] and NuSMV Cimatti et al. [1999]. A potential drawback of employing symbolic reasoners for temporal logic reasoning is that they often require specialized knowledge bases and rules tailored to a specific temporal logic language, which can be both time-consuming and resource-intensive to create and maintain. Additionally, the inability to support natural language expressions also limits their applicability to other domains.

In recent years, there has been growing evidence that contemporary Large Language Models (LLMs) have achieved remarkable performance across various domains, including automatic bug fixing Bouzenia et al. [2024], Wang et al. [2024], commonsense reasoning Wang and Zhao [2023a],

[*]Equal contribution.

Zhao et al. [2024], and mathematical reasoning Ahn et al. [2024]. Currently, a widely adopted approach to calibrating the diverse capabilities of Large Language Models (LLMs) is through the construction of well-designed and representative benchmarks. For example, HumanEval Chen et al. [2021] was introduced to evaluate the coding abilities of LLMs, while GSM8K Cobbe et al. [2021] was developed to assess their performance in mathematical reasoning. However, in traditional logic-based Artificial Intelligence (AI) domains, many tasks are still addressed using formal logical rules and symbolic reasoners. Despite the advancements of LLMs, relatively little effort has been made to explore their capabilities in solving such tasks—particularly the more challenging aspects of rule-based temporal logic reasoning. While some studies have benchmarked or evaluated the temporal reasoning abilities of LLMs Wang and Zhao [2023b], Xiong et al. [2024], they primarily focus on reasoning over temporal data expressed in natural language, without addressing the temporal logic, which is typically represented as logical rules with well-established syntax and semantics.

In this paper, we introduce `t-BEN`, a benchmark suite to evaluate the temporal reasoning capabilities of language models. Each question is constructed based on temporal logic and is guaranteed to be unseen during training, thereby requiring models to perform reasoning rather than rely on memorized knowledge. Specifically, we adopt DatalogMTL Brandt et al. [2018], a popular temporal logic language, as a proxy, and focus on the classic temporal logic reasoning task of *fact entailment* Cheng [1996], Brandt et al. [2018]. We consider temporal data of the symbolic form $P(a_1, \ldots, a_n)@\varrho$, where P denotes a predicate (relation), a_i is an entity, n denotes the arity[2] and ϱ represents a punctual time point or time interval. Given a set of temporal rules and a target temporal fact, the task is to determine whether the fact is entailed by the temporal data and logical rules. To provide better intuition, we use Example 1 togeter with Figure 1 to describe the problem.

Example 1. *There is growing evidence that individuals develop COVID-19 immunity if they were infected within the last 6 months (discounting the last ten days when they had no symptom) Feikin et al. [2022]. The condition can be captured by a DatalogMTL program* Π_{ex} *with the following rule:*

$$\mathit{Immune}(x) \leftarrow \diamondminus_{(10,183]}\mathit{Infect}(x), \boxminus_{[0,10]}\mathit{NoSym}(x)$$

The above rule checks whether an individual infected at some point in the last six months excluding the last 10 days (operator $\diamondminus_{(10,183]}$) remained continuously without symptoms in the last 10 days (using the 'box past' operator $\boxminus_{[0,10]}$).

Then, we assume a dataset contains some historical data about a person called Ben in the form of facts stamped with validity intervals, where the first day of the year is given by the interval $(0, 1]$, the second day by $(1, 2]$, and so on. Ben got vaccinated at July 19 (represented as 199). Moreover, Ben had no symptoms since July 1 (i.e., 181) until August 30 (i.e., 242). This is represented by a dataset $\mathcal{D}_{ex}$ with the following facts:

$$\mathit{Infect}(\mathsf{Ben})@199, \quad \mathit{NoSym}(\mathsf{Ben})@(181, 242]$$

If we want to know whether Ben is immune between September 8 and September 9, represented as a temporal fact $Ben@(251, 252]$, we can formulate this as a *fact entailment* problem: Is $Ben@(251, 252]$ entailed by $\mathcal{D}_{ex}$ and Π_{ex}?

Traditionally, a symbolic reasoner Bellomarini et al. [2018], Fionda and Greco [2018], Wang et al. [2022] is used to check entailment by applying temporal rules to temporal data, deriving new facts, and verifying if the given fact is among the derived ones. However, there are two key challenges in using symbolic reasoners for temporal reasoning tasks: 1) *symbolic reasoners cannot directly process natural language descriptions and instead require inputs to be formalized as logical rules*; 2) *generating these logically consistent and error-free rule representations is a non-trivial task that demands significant domain expertise and manual effort*.[3] In this paper, we explore whether Large Language Models (LLMs) can solve temporal reasoning tasks in both symbolic and natural language forms, potentially serving as an alternative to, or a complementary tool for, traditional symbolic reasoners. Our contributions are summarized as follows:

[2]If the arity is 0, then P is treated as a statement that is either true or false. This differs from temporal knowledge graphs, which consist solely of quadruples (arity=2).

[3]Although prior work has explored converting natural language expressions into logical rules Chen et al. [2023], Tammet et al. [2024], the accuracy of such conversions remains an open question. The two-stage pipeline may suffer from error propagation, which complicates the reasoning process.

- `t-BEN` is the first temporal reasoning benchmark constructed based on the formal semantics of temporal logic, while supporting evaluation in both symbolic and natural language forms.
- `t-BEN` provides a scalable and verifiable testbed for the creation of datasets with varying sizes and rule structures of different complexities. Moreover, the questions in `t-BEN` are guaranteed to be unseen by LLMs during pretraining, thereby mitigating the risk of data leakage and enabling a more rigorous and trustworthy evaluation setting.
- We conduct extensive experiments to evaluate the performance of several frontier Large Language Models (LLMs), including both open-source and proprietary models, on `t-BEN`. Our results reveal an interesting observation: among all evaluated models, only DeepSeek-R1 delivers impressive results on `t-BEN`, while other LLMs—including GPT-4o—perform poorly, often nearing random chance. Additionally, our analysis of other distilled variants of DeepSeek-R1 reveals consistent performance gains, which we attribute to DeepSeek's unique training strategy—specifically, the inclusion of instruction-following data during the final stages of supervised fine-tuning and reinforcement learning training.

2 Related Works

Temporal logic reasoning Knowledge representation languages, such as Linear Temporal Logic (LTL) Huth and Ryan [2004] and DatalogMTL Brandt et al. [2018], have become the de facto standard for specifying temporal properties in both formal verification and artificial intelligence. Many temporal reasoning problems have proven to be PSPACE-complete Wałęga et al. [2019], Fionda and Greco [2018], Bauland et al. [2009]. *Satisfiability checking*, that is, the problem of deciding whether a given formula admits a satisfying model, is one of the most important computational tasks associated with the logic, and one of the first that have been carefully studied Sistla and Clarke [1985]. Similarly, the reasoning tasks considered in DatalogMTL are *fact entailment* and *consistency checking*. These problems polynomially reduce to the complements of each other Brandt et al. [2018]. Despite this theoretically high computational complexity, numerous techniques and tools are developed to solve different temporal reasoning problems, ranging from tableau systems Goré and Widmann [2009], Bertello et al. [2016] to reductions to model checking Cavada et al. [2014], to automata techniques Li et al. [2014], Wang et al. [2022].

Benchmarking and Reasoning in Large Language Models Although the aforementioned temporal reasoning problems have been widely explored in the traditional logic-based AI domain, they remain underexplored in the regime of LLMs. In recent years, benchmarking reasoning capabilities in LLMs is a problem of pressing interest to the field Plaat et al. [2024], Chang et al. [2024], Huang and Chang [2022]. There is a substantial body of research evaluating the reasoning abilities of LLMs, covering areas such as arithmetic reasoning, logical reasoning, and commonsense reasoning. Notably, simple math problem datasets like AQUA Ling et al. [2017], GSM8K [Cobbe et al., 2021], and SVAMP [Patel et al., 2021] are frequently used to assess arithmetic reasoning [Touvron et al., 2023, Shi et al., 2023]. Welleck et al. [2021] developed NaturalProofs, a multi-domain dataset for studying mathematical reasoning in natural language, while Welleck et al. [2022] investigated LLMs' abilities to generate the next step in mathematical proofs and complete full proofs. Additionally, LLMs have been evaluated on logical reasoning tasks, including symbolic tasks like Coin Flip and Last Letter Concatenation [Wei et al., 2022], and Logic Grid Puzzles on the BIG-BENCH [Srivastava et al., 2023]. Commonsense reasoning datasets [Talmor et al., 2019] have also been proposed for evaluating LLMs. Most relevant to our work are various approaches to evaluating and enhancing the algorithmic reasoning abilities of LLMs [Zhou et al., 2022, Fatemi et al., 2024].

3 DatalogMTL

DatalogMTL Brandt et al. [2018], Wałęga et al. [2019] is a temporal logic language, which extends Datalog Abiteboul et al. [1995] with operators from metric temporal logic (MTL) Koymans [1990]. Different Datalog designed to handle static facts and rules due to lack of built-in temporal constructs, DatalogMTL equipped with MTL operators is enabled to reasoning about properties of systems that evolve over time. These operators build upon the standard linear temporal logic (LTL) Huth and Ryan [2004] operators, such as $\diamondminus$ standing for "sometime in the past", $\boxminus$ for "always in the past", and $\mathcal{S}$ for "since", as well as their future counterparts $\diamondplus$ for "sometime in the future", $\boxplus$ for "always in the future", and $\mathcal{U}$ for "until". In MTL, however, these LTL operators are annotated with intervals; for instance, the expression $\diamondminus_{[1,2]} \mathit{LiveIn}(x, y)$ is true at time t if entity x lived in location y sometime between times $t-1$ and $t-2$. Similarly, $\boxminus_{[1,2]} \mathit{LiveIn}(x, y)$ holds at time t if x continuously lived in y throughout the aforementioned time interval. In this section, we recapitulate the syntax, semantics, and key temporal tereasoning problems in DatalogMTL.

Syntax We consider a *signature* consisting of pairwise disjoint countable sets of constants, variables, and predicates with non-negative integer arities. A term is either a constant or a variable. A *relational atom* is an expression of the form $P(\mathbf{s})$, with P a predicate and $\mathbf{s}$ a tuple of terms whose length matches the arity of P. In this paper, we restrict ourselves to a fragment in which metric atoms are generated by the following grammar, where $P(\mathbf{s})$ is a relational atom and ϱ an interval including only non-negative numbers:

$$M ::= P(\mathbf{s}) \mid \diamondminus_{\varrho} M \mid \diamondplus_{\varrho} M \mid \boxminus_{\varrho} M \mid \boxplus_{\varrho} M$$

A rule in this fragment is an expression of the form

$$P(\mathbf{s}) \leftarrow M_1 \wedge \cdots \wedge M_n, \qquad \text{for } n \geq 1, \tag{1}$$

where the body atoms $M_1, \ldots, M_n$ are metric atoms and the head atom $P(\mathbf{s})$ is relational. A program is a finite set of rules.

Semantics An interpretation $\mathfrak{I}$ is a function assigning truth values to ground relational atoms $P(\mathbf{c})$ and time points $t \in \mathbb{Z}$. It determines if $P(\mathbf{c})$ is satisfied at t, denoted as $\mathfrak{I}, t \models P(\mathbf{c})$, or not, denoted as $\mathfrak{I}, t \not\models P(\mathbf{c})$. This notion of truth assignment extends to other ground metric atoms in the considered fragment as follows:

$$
\begin{array}{lll}
\mathfrak{I}, t \models \diamondminus_{\varrho} M & \text{iff} & \mathfrak{I}, t' \models M \text{ for some } t' \text{ with } t - t' \in \varrho, \\
\mathfrak{I}, t \models \diamondplus_{\varrho} M & \text{iff} & \mathfrak{I}, t' \models M \text{ for some } t' \text{ with } t' - t \in \varrho, \\
\mathfrak{I}, t \models \boxminus_{\varrho} M & \text{iff} & \mathfrak{I}, t' \models M \text{ for all } t' \text{ with } t - t' \in \varrho, \\
\mathfrak{I}, t \models \boxplus_{\varrho} M & \text{iff} & \mathfrak{I}, t' \models M \text{ for all } t' \text{ with } t' - t \in \varrho.
\end{array}
$$

For example, an interpretation making atom $\mathit{LiveIn}(Ann, Paris)$ true everywhere within $[10, 30]$ and false elsewhere makes $\boxminus_{[1,2]} \mathit{LiveIn}(Ann, Paris)$ true at the time point 31, but false at 32. An interpretation can be alternatively seen as the (possibly infinite) set of facts that it satisfies, which yields a natural meaning to containment and minimality of interpretations.

3.1 Major Temporal Reasoning Problems

According to Brandt et al. [2018], Wałęga et al. [2019], temporal logic reasoning involves two major problems: *consistency checking* and *fact entailment*. *Consistency checking* is the task of determining whether a given program and dataset admit a common model Emerson [1990], Schnoebelen [2002]. *Fact entailment* involves checking whether a program and dataset together entail a specific relational fact. Brandt et al. [2018] note that in DatalogMTL, consistency checking and fact entailment are complementary problems. Consequently, this paper focuses solely on the *fact entailment* problem to evaluate the temporal reasoning capabilities of large language models.

4 `t-BEN`: A Benchmark Suite for Generating Temporal Reasoning Datasets

DatalogMTL is a temporal logic language that can characterize complex temporal conditions by defining various rules using combinations of different atoms and temporal operators ($\diamondminus, \diamondplus, \boxminus, \boxplus$) whose semantics has been described in Section 3. To some extent, the complexity of a *fact entailment* problem is largely determined by the complexity of associated temporal rules.

4.1 Leveling DatalogMTL Rules

To address the aforementioned challenge and provide a more comprehensive evaluation of the temporal reasoning abilities of large language models, we aim to create a new synthetic benchmark with flexible configurations for customizing rule structures and task complexity. We classify DatalogMTL rules into six classes (**S-Atom**, . . . , **Recursive**) based on their structural representations, considering

Zero-shot Prompt Prefix

Given a dataset, temporal rules and a temporal fact, you need to apply the rules to the dataset and then judge whether the given fact is entailed by the dataset and rules.

The rules are expressed as DatalogMTL, a language of temporal logic that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows:

If $\diamondminus_{[a,b]} A$ is true at the time t, it requires that A needs to be true at some time between t-b and t-a.

If $\boxminus_{[a,b]} A$ is true at the time t, it requires that A needs to be true continuously between t-b and t-a.

If $\diamondplus[a,b] A$ is true at the time t, it requires that A needs to be true at some point between t+a and t+b.

If $\boxplus[a,b] A$ is true at the time t, it requires that A needs to be true continuously between t+a and t+b.

Now, we have a data, some DatalogMTL rules and a fact entailment question. You should only output true or false, and please do not output other words.

SingleAtom

R: $\mathrm{A} \leftarrow \diamondminus_{[1,2]}\mathrm{B}$
D: $\{\mathrm{B}@[4,5]\}$
Q: A@6 is entailed?

MultiAtoms

R: $\mathrm{A} \leftarrow \boxminus_{[3]}\mathrm{B} \wedge \boxminus_{[2,\ 3]}\mathrm{C}$
D: $\{\mathrm{B}@[1], \mathrm{C}@[1,\ 3]\}$
Q: A@3 is entailed?

Rational

R: $\mathrm{A} \leftarrow \boxminus_{[1.2,\ 2.1]}\mathrm{B} \wedge \boxminus_{[4.2,\ 5.1]}\mathrm{B}$
D: $\{\mathrm{A}@[1.1]\}$
Q: A@[2.4, 2.8]) is entailed?

MixedOperators

R: $\mathrm{A} \leftarrow \diamondminus_{[1,\ 2.4]}\mathrm{B} \wedge \boxplus_{[1,\ 2]}\mathrm{C}$
D: $\{\mathrm{B}@[1], \mathrm{C}@[2,\ 4]\}$
Q: A@2.3 is entailed?

MultiRules

R: $\mathrm{D} \leftarrow \boxminus_{[2]}\mathrm{B} \wedge \boxplus_{[1,2]}\mathrm{C}$
$\mathrm{A} \leftarrow \diamondminus_{[1.5,\ 2]}\mathrm{D} \wedge \diamondminus_{[2]}\mathrm{C}$
D: $\{\mathrm{B}@[1], \mathrm{C}@[2,5]\}$
Q: A@[4.5, 5] is entailed ?

Recursive

R: $\mathrm{A} \leftarrow \diamondminus_{[1,2]}\mathrm{A} \wedge \boxminus_{[1,10]}\mathrm{C}$
D: $\{\mathrm{A}@[1], \mathrm{C}@[1,100]\}$
Q: A@99 is entailed?

Figure 1: Six levels of temporal reasoning problems with varying complexity. We present an intuitive example representing each level, along with the corresponding rule, dataset, and fact entailment problem. A zero-shot-prompt prefix is also provided (see Appendix for additional prompt prefixes used in this paper). For better demonstration, we use the symbols $\diamondminus$, $\diamondplus$, $\boxminus$, and $\boxplus$, which are replaced by $<->$, $<+>$, $[-]$, and $[+]$, respectively, in the actual prompts due to typing constraints.

factors such as the number of body atoms, the number of temporal operators used, the number of rules involved, and whether the rules are recursive. While we are unable to quantify the degree of complexity of each level, we assume that higher levels correspond to greater complexity. This assumption is based on the observation that more complex rule structures require additional temporal reasoning steps when using a symbolic reasoner like MeTeoR Wang et al. [2022].

S-Atom The most simplest form of a rule is $A \leftarrow \oslash_{[\rho]} B$, where $\oslash$ could be one of the four metric temporal operators ($\boxminus$, $\boxplus$, $\diamondminus$ and $\diamondplus$). We ensure that A and B are two different atoms, so only one calculation operation. A **S-Atom** example is given in Figure 1, where we can derive $A@[5,7]$ based on the given dataset and the rule, entailing that $A@6$ is true. In particular, we consider the integer timeline, a fragment of DatalogMTL Wałęga et al. [2020] and use one type of MTL operator.

M-Atoms In the **S-Atom**, the body contains only one atom, so a single rule application is sufficient to complete the derivation. In **M-Atoms**, we increase the number of atoms in the rule body, requiring not only the validation of each atom but also an intersection operation to obtain the final valid interval. As the example shown in Figure 1, the rule contains two atoms. First, we calculate the valid intervals for each atom. Based on the provided facts, $\boxminus_{[3]} B$ holds only at the punctual time point $[4,4]]$, and $\boxminus_{[2,3]} A$ holds at the interval [4,5]. The intersection of these intervals, $[4,4]$ and $[4,5]$, is $[4,4]$. Thus, we derive that A is true at the time point 4, so $A@4$ is entailed. As with **S-Atom**, we consider DatalogMTL over the integer timeline Wałęga et al. [2020] and use only one type of MTL operator.

Rational Both **S-Atom** and **M-Atoms** focus solely on the integer timeline, which represents a relatively limited time space and simplifies reasoning due to the integer semantics Wałęga et al. [2020]. In **Rational**, we build on top of **M-Atoms** by expanding the timeline to include the rational numbers, incorporating decimal time points. Intuitively, rational-based numerical operations are more complex than their integer-based counterparts, and we aim to determine if large language models exhibit similar behavior. We continue to use only one type of MTL operator at this level.e type of MTL operator in the level.

M-Operators Using only one operator limits the expressiveness of DatalogMTL, preventing the definition of complex temporal conditions. Thus, a natural expansion is to allow the use of MTL operators. The four types of MTL operators can be used to define temporal conditions associated with both the past and the future. A **M-Operators** example is shown in Figure 1, which involves

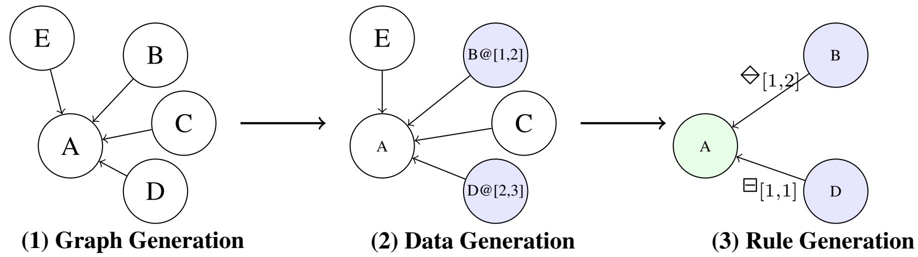

Figure 2: An example of generating temporal data and rules. First, we randomly generate a graph. Next, our program selects specific nodes to assign time points. In our example, nodes B and D are chosen, resulting in two temporal facts: $\{$B@[1,2], D@[2,3]$\}$; Finally, we select a node as the head atom, with body atoms derived from the previous step. We then randomly assign temporal operators to these body atoms, resulting in the rule: A $\leftarrow \diamondminus_{[1,2]}$B $\wedge \boxminus_{[1,1]}$D. The number of body atoms, the time range, and the temporal operators are specified as input parameters.

two MTL operators ($\diamondminus$ and $\boxplus$). To complete the derivation, we first calculate the valid interval where $\diamondminus_{[1,2.4]}B$ with the past operator ($\diamondminus$) holds, which is $[2, 3.4]$. Then, we calculate $\boxplus_{[1,2]C}$, whose valid interval is $[1, 2]$. After performing the interval intersection, we obtain that A holds at the time interval $[2, 2]$. Thus, the temporal $A@2.3$ is not entailed.

M-Rules In the previous four levels, fact entailment is associated with only one temporal rule. However, in more practical scenarios, multiple temporal rules may be required to express complex temporal conditions. In this level, we consider a multi-rule temporal reasoning case, where fact entailment involves multiple temporal rules and rule applications must be executed across these rules to complete the derivation. As the example in Figure 1, to derive the target atom A, we need to know both D and C. However,the dataset only provides the information about C. We can derive the D holds at 3 according to the first temporal rule $D \leftarrow \boxminus_{[2]} \wedge \boxplus_{[1,2]}C$; then, we can derive that A holds at the interval $[4.5, 5]$ according to the second rule. Hence, $A@[4.5, 5]$ is entailed.

Recursive The fact entailment problem at this level is considered the hardest because it involves recursion. Unlike static knowledge representation languages (e.g., Datalog), where all facts can be derived after a certain number of rule applications, some recursive rules in DatalogMTL may require an infinite number of applications. Even for symbolic-based approaches, this presents a significant challenge, and researchers have devoted considerable effort to addressing it Wałęga et al. [2021, 2023]. According to Wałęga et al. [2023], in the recursive scenarios, periodic structures will ultimately occur repeatedly, but calculating these periodic structures is challenging. From a human perspective, however, identifying such periodic structures can be straightforward. For instance, consider a recursive rule $\boxplus_{1year}$Bday$(x) \leftarrow$ Bday(x), which states that anyone having their birthday at a time point t will also be having their birthday at the same time the following year. If we know that Ben has his birthday on Jun 8, 1991, it is easy to know that he will have his birthday on Jun 8, 1992, Jun 8, 1993 and so on. However, this is difficult for traditional symbolic-based approaches to handle. Therefore, we design fact entailment problems associated with recursive rules to test whether large language models can perform well in this setting.

Specifically, we use facts from both propositional logic Klement [2004] and first-order logic Barwise [1977]. The former contains declarative statements that are either 'true' or 'false', while the latter includes expressions with one or more variables. For example, we allow both forms of temporal facts: $Raining$ and $Immune(x)$. The former states that an event (raining) is occurring, while the latter denotes that a property (immune) is associated with an entity, where x acts as a placeholder that can be instantiated to any entity, such as $Immune(Ben)$, indicating that Ben is immune.

4.2 Generating Temporal Data and Rules

The benchmark generation process can be mainly divided into the following three steps: 1) Graph construction, 2) Data generation, and 3) Rule generation. The pseudocode for this benchmark generation algorithm can be found in Appendix F.

Graph construction We employ a general-purpose random graph generator to generate a connected directed random graph. The nodes in the random graph represent predicates, such as A, B, and C. Each edge in this graph represents a body atom of a rule pointing to the corresponding head in the rule. In particular, a predicate can appear in bodies of multiple different rules.

Data generation After the construction of the graph, the program will traverse each nodes in the graph and randomly assign time points or time intervals to the chosen nodes. The time points or intervals are generated based on a given range.

Rule Generation Once the temporal data is generated, the rule generator traverses the edges of the graph, assigning random operators and intervals to the edges. To ensure the generated graph is non-trivial, a reasoning process is performed across the entire graph after completing this step to ensure new facts can be inferred. If multiple rules are required, the program repeats previous steps until a sufficient number of rules are generated.

An example Figure 2 shows an example of generating temporal data and rules. In particular, our program will have a post-processing operation to scan all the data and rules to ensure they have been utilized and removes any data and rules (in the ablation study, we will explore the impact of irrelevant data and rules) that are not participated in the the temporal reasoning process. We define the following flags for the samples to be generated based on their characteristics: `rational number`, `multiple body atoms`, `recursive` and `mixed operators`. These flags control the rule structures during the generation process.

	Prompt type	**S-Atom**	**M-Atoms**	**Rational**	**M-Operators**	**M-Rules**	**Recursive**
GPT-4o	Zero-shot	45.8	43.2	37.1	57.3	53.3	37.7
	Few-shot	40.4	38.0	27.2	51.6	36.7	32.2
	Zero-shot-CoT	85.6	85.1	85.7	90.3	74.0	58.0
Llama-3-8B	Zero-shot	40.7	44.0	43.9	60.5	39.1	8.7
	Few-shot	38.4	44.3	44.4	47.1	36.1	30.2
	Zero-shot-CoT	59.9	58.4	68.2	64.1	59.0	48.5
Qwen2.5-32B	Zero-shot	47.0	46.0	33.0	49.5	38.5	16.0
	Few-shot	41.5	48.0	31.0	56.0	42.5	21.5
	Zero-shot-CoT	80.0	80.0	78.4	89.0	61.6	51.5
Distill-Qwen-7B	Zero-shot	80.7	75.9	70.0	79.9	65.6	45.5
Distill-Qwen-14B	Zero-shot	95.0	92.0	97.0	95.5	88.4	57.6
Distill-Qwen-32B	Zero-shot	96.9	87.9	97.5	90.4	86.2	64.0
DeepSeek-R1	Zero-shot	**100.0**	**96.0**	**99.5**	**99.5**	**97.5**	**88.9**

Table 1: Model performance measured by accuracy on the synthetic benchmarks across six rule structures, as defined in Section 4.1.

5 Experiments and Results

Baselines We evaluate the performance of seven LLMs on `t-BEN`. These models include GPT-4o Achiam et al. [2023], DeepSeek-R1 Liu et al. [2024] and three DeepSeek-R1 distilled models (DS-R1-Distill-Qwen-7B, DS-R1-Distill-Qwen-14B and DS-R1-Distill-Qwen-32B), Llama-3 Dubey et al. [2024] and Qwen2.5-32B-Instruct Yang et al. [2024]. Specifically, we conduct experiments on GPT-4o, Llama-3-8B-Instruct, and Qwen2.5-32B-Instruct using three different prompting strategies: zero-shot prompting, few-shot in-context learning [Brown et al., 2020], and chain-of-thought prompting [Wei et al., 2022]. Due to the unique nature of DeepSeek, which inherently incorporates a reasoning process, we consider only the zero-shot prompting setting for DeepSeek-R1 and three distilled variants.

Benchmark statistics and experimental settings Unless otherwise specified, each benchmark level contains 200 samples selected from the facts derived using the chosen data and rule(s). For negative samples, a random interval is chosen, ensuring that these intervals do not overlap with those of the derived facts. Specifically, for all baselines, the temperature value is set to 0. For few-shot prompting techniques, the input prompt includes two manually constructed exemplars. In this paper, we use both the F1 score and the accuracy as the evaluation metric. Single-run results are reported.

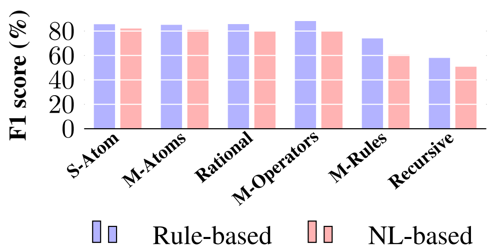

Figure 3: Comparison of symbolic and natural language (NL) based temporal logic reasoning using GPT-4o with Zero-shot-CoT.

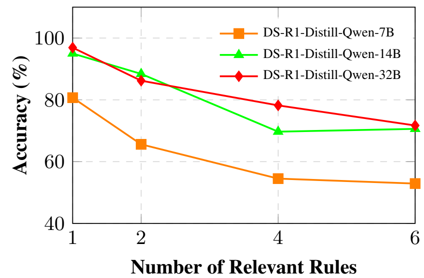

Figure 4: Performance of DeepSeek-R1 distilled models across different numbers of relevant rules.

5.1 Main Results

From Table 1, we observe a striking phenomenon: compared to DeepSeek-R1 and its distilled models, GPT-4o, Llama-3, and Qwen-32B-Instruct perform poorly on the temporal logic reasoning problems of `t-BEN`, even with chain-of-thought prompting (CoT). This suggests that these models lack the advanced reasoning capabilities necessary for truly understanding symbolic representations involving time. Notably, in the **M-Rules** and **Recursive** benchmarks, we observe a significant performance drop across all evaluated models. These results indicate that recursive rules pose a particular challenge, as they require not only an understanding of language semantics and step-by-step reasoning but also strong *inductive abilities*. However, a surprising finding is that, apart from the task involving recursive structures, DeepSeek-R1 achieves an accuracy of 88.9%, and for all five other levels, it surpasses 96% accuracy—demonstrating exceptionally strong symbolic reasoning abilities. One possible explanation for DeepSeek's strong performance lies in its distinctive training strategy—namely, the incorporation of instruction-following data during the final stages of supervised fine-tuning and reinforcement learning. This approach may improve the model's ability to adhere to prompts, such as our system-provided instructions, thereby enhancing its temporal reasoning capabilities. In addition, we evaluated several smaller DeepSeek-R1 distilled models, which also exhibited remarkable performance. These findings suggest that integrating instruction-following data into the training process may be an effective strategy for strengthening a model's temporal reasoning abilities.

5.2 Symbolic v.s. Natural Language

In addition to evaluating the temporal reasoning capabilities of LLMs in symbolic forms—where traditional symbolic reasoners excel—it is also valuable to assess their performance in natural language scenarios, which symbolic reasoners cannot handle. To this end, we adopt a common strategy of verbalizing logical rules before presenting them to the LLMs, following the approach explored in prior works Saxena et al. [2021], Ismayilzada et al. [2023]. Given that manually converting each rule into its corresponding natural language expression is a labor-intensive process, we adopt a template-based approach to automate this verbalization. Although this method may result in unnatural expressions, it provides a practical alternative to manual translation.

From Figure 3, we observe that both the rule-based and natural language-based settings achieve similar results, with the rule-based approach performing slightly better. The comparison indicates that *LLMs are also capable of understanding the semantics of input expressed in rules*, provided that each notation is clearly explained in the instructions. Notably, both settings struggle with the **M-Rules** and **Recursive** cases. One possible reason for this is that, while LLMs can understand the semantics of temporal logic language, they still face significant challenges in executing multiple deductions, retaining intermediate results, and recognizing repeated patterns—tasks that require delicate algorithms to accomplish effectively.

5.3 Ablation study

To explore which component of the rule structure most significantly impact the reasoning complexity for LLMs, we designed four sets of ablation study experiments using GPT-4o. These experiments explored the effects of the number of relevant rules , the number of operators considered, the

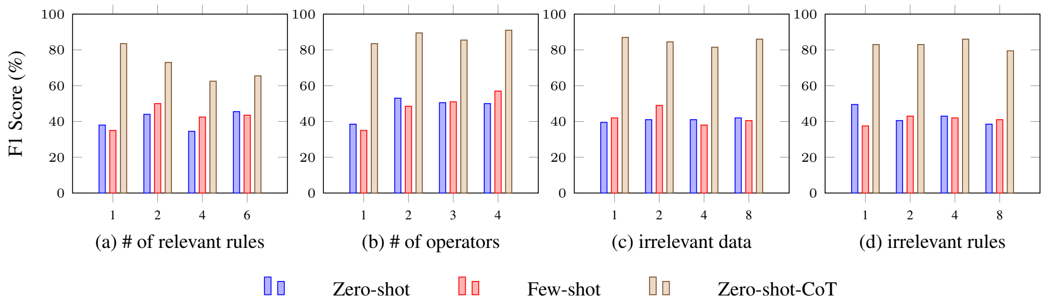

Figure 5: Results of ablation study for GPT-4o with three different prompting strategies.

percentage of irrelevant data, and the percentage of irrelevant rules. From Figure 5 (a), we observe that as the lengths of dependent rules increase, the model's performance noticeably degrades.One possible reason is that when multiple rules are mutually dependent, the model needs to store intermediate results during the derivation process to complete subsequent steps that rely on previously derived outcomes. Unlike symbolic reasoners, which can explicitly store intermediate results, it may be challenging for large language models (LLMs) to retain such information in an auto-regressive manner. Additionally, Figure 5 (b) demonstrates that using more types of operators does not affect reasoning complexity, indicating that understanding the semantics of the temporal logic language is not a major issue for the model. Results in Figures 5 (c) and (d) show that the model's performance is minimally affected by irrelevant information, demonstrating its ability to correctly select relevant rules and remain resistant to distracting information.

Furthermore, in Figures 5, we observe that the number of relevant rules has the most significant impact. To further explore its impact, we experiment with the three DeepSeek-R1 distilled models, which have demonstrated strong performance in the single-rule setting (Table 1). In Figure 4 , it shows that as the number of relevant rules increases, performance declines, suggesting that reasoning over multiple rules remains a significant challenge.

Robustness to the input formats We investigate the impact of the input formats to the LLM-based approach through three evaluation settings: 1 *error-free symbolic input*,2 *symbolic input with errors*, and 3 *natural language input*. We construct a subset of 100 questions, each represented in all three formats. For 3, we introduce syntactic errors by randomly removing notation elements that cause parsing issues—for example, altering $\boxminus_{[1,2]}$ to $\boxminus_{1,2]}$ by removing the opening bracket. Symbolic reasoners can only handle the error-free symbolic input. In contrast, the LLM demonstrates strong accuracy across all three settings (95.0%, 94.5% and 94.4%)[4]. This suggests that the LLM not only exhibits effective temporal reasoning capabilities but also shows robustness to imperfect input.

Human analysis of errors We do a manual analysis of the reasoning processes of two models of the same size—Qwen2.5-32B-Instruct and DeepSeek-R1-Distill-Qwen-32B—in the most challenging recursive setting, we observe a key difference. Qwen2.5-32B-Instruct performs only shallow inference step, failing to recognize the recursive nature of the problem and its potential for infinite expansion. In contrast, DeepSeek-R1-Distill-Qwen-32B correctly identifies the recursive structure, explicitly acknowledging it with statements such as "... applying the rule again, A at 8 would imply A at 10, and so on." This deeper understanding enables the model to arrive at the correct result.

6 Conclusion

We introduce T-BENCH, a benchmark suite designed to systematically evaluate the temporal reasoning capabilities of large language models (LLMs) in a controlled setting. Preliminary results suggest that certain LLMs, such as DeepSeek-R1, may serve as viable alternatives or complementary tools to traditional symbolic reasoners, though further investigation is needed. By open-sourcing our codes and datasets, we hope to stimulate further research and development in this field, thereby better facilitating the potential application of LLMs in traditional logic-based AI domains.

[4]Evaluated on S-Atom using DeepSeek-R1-Distill-Qwen-14B.

References

Rajeev Alur and Thomas A Henzinger. A really temporal logic. *Journal of the ACM (JACM)*, 41(1): 181–203, 1994.

Yde Venema. Temporal logic. *The Blackwell guide to philosophical logic*, pages 203–223, 2017.

Leslie Lamport. " sometime" is sometimes" not never" on the temporal logic of programs. In *Proceedings of the 7th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 174–185, 1980.

Dingmin Wang, Pan Hu, Przemysław Andrzej Wałęga, and Bernardo Cuenca Grau. Meteor: Practical reasoning in datalog with metric temporal operators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5906–5913, 2022.

Alessandro Cimatti, Edmund Clarke, Fausto Giunchiglia, and Marco Roveri. Nusmv: A new symbolic model verifier. In *Computer Aided Verification: 11th International Conference, CAV'99 Trento, Italy, July 6–10, 1999 Proceedings 11*, pages 495–499. Springer, 1999.

Islem Bouzenia, Premkumar Devanbu, and Michael Pradel. Repairagent: An autonomous, llm-based agent for program repair. *arXiv preprint arXiv:2403.17134*, 2024.

Dingmin Wang, Jinman Zhao, Hengzhi Pei, Samson Tan, and Sheng Zha. Fine-tuning language models for joint rewriting and completion of code with potential bugs. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15854–15868, 2024.

Yuqing Wang and Yun Zhao. Gemini in reasoning: Unveiling commonsense in multimodal large language models. *arXiv preprint arXiv:2312.17661*, 2023a.

Zirui Zhao, Wee Sun Lee, and David Hsu. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems*, 36, 2024.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Yuqing Wang and Yun Zhao. Tram: Benchmarking temporal reasoning for large language models. *arXiv preprint arXiv:2310.00835*, 2023b.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. Large language models can learn temporal reasoning. *arXiv preprint arXiv:2401.06853*, 2024.

Sebastian Brandt, Elem Gužel Kalaycı, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. Querying log data with metric temporal logic. *Journal of Artificial Intelligence Research*, 62:829–877, 2018.

Jingde Cheng. The fundamental role of entailment in knowledge representation and reasoning. *Journal of Computing and Information*, 2(1):853–873, 1996.

Daniel R. Feikin, Melissa M. Higdon, Laith J. Abu-Raddad, Nick Andrews, Rafael Araos, Yair Goldberg, Michelle J. Groome, Amit Huppert, Katherine L. O'Brien, Peter G. Smith, Annelies Wilder-Smith, Scott Zeger, Maria Deloria Knoll, and Minal K. Patel. Duration of effectiveness of vaccines against sars-cov-2 infection and covid-19 disease: Results of a systematic review and meta-regression. *The Lancet*, 399(10328):924–944, 2022. doi: 10.1016/S0140-6736(22)00152-0.

Luigi Bellomarini, Emanuel Sallinger, and Georg Gottlob. The vadalog system: datalog-based reasoning for knowledge graphs. *Proceedings of the VLDB Endowment*, 11(9):975–987, 2018.

Valeria Fionda and Gianluigi Greco. Ltl on finite and process traces: Complexity results and a practical reasoner. *Journal of Artificial Intelligence Research*, 63:557–623, 2018.

Yongchao Chen, Rujul Gandhi, Yang Zhang, and Chuchu Fan. Nl2tl: Transforming natural languages to temporal logics using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15880–15903, 2023.

Tanel Tammet, Priit Järv, Martin Verrev, and Dirk Draheim. Experiments with llms for converting language to logic. In *International Conference on Neural-Symbolic Learning and Reasoning*, pages 305–314. Springer, 2024.

Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge university press, 2004.

Przemysław Andrzej Wałęga, B Cuenca Grau, Mark Kaminski, and Egor V Kostylev. DatalogMTL: Computational complexity and expressive power. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2019.

Michael Bauland, Martin Mundhenk, Thomas Schneider, Henning Schnoor, Ilka Schnoor, and Heribert Vollmer. The tractability of model-checking for ltl: The good, the bad, and the ugly fragments. *Electronic Notes in Theoretical Computer Science*, 231:277–292, 2009.

A Prasad Sistla and Edmund M Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM (JACM)*, 32(3):733–749, 1985.

Rajeev Goré and Florian Widmann. An optimal on-the-fly tableau-based decision procedure for pdl-satisfiability. In *International Conference on Automated Deduction*, pages 437–452. Springer, 2009.

Matteo Bertello, Nicola Gigante, Angelo Montanari, Mark Reynolds, et al. Leviathan: A new ltl satisfiability checking tool based on a one-pass tree-shaped tableau. In *IJCAI-International Joint Conference onArtificial Intelligence*, pages 950–956. AAAI Press, 2016.

Roberto Cavada, Alessandro Cimatti, Michele Dorigatti, Alberto Griggio, Alessandro Mariotti, Andrea Micheli, Sergio Mover, Marco Roveri, and Stefano Tonetta. The nuxmv symbolic model checker. In *Computer Aided Verification: 26th International Conference, CAV 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 18-22, 2014. Proceedings 26*, pages 334–342. Springer, 2014.

Jianwen Li, Yinbo Yao, Geguang Pu, Lijun Zhang, and Jifeng He. Aalta: an ltl satisfiability checker over infinite/finite traces. In *Proceedings of the 22nd ACM SIGSOFT international symposium on foundations of software engineering*, pages 731–734, 2014.

Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*, 2024.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45, 2024.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*, 2022.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1015. URL `https://aclanthology.org/P17-1015`.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL `https://aclanthology.org/2021.naacl-main.168`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR, 2023.

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. Naturalproofs: Mathematical theorem proving in natural language. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL `https://openreview.net/forum?id=Jvxa8adr3iY`.

Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. Naturalprover: Grounded mathematical proof generation with language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=rhdfTOiXBng`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=_VjQlMeSB_J`.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL `https://aclanthology.org/N19-1421`.

Hattie Zhou, Azade Nova, Hugo Larochelle, Aaron Courville, Behnam Neyshabur, and Hanie Sedghi. Teaching algorithmic reasoning via in-context learning. *arXiv preprint arXiv:2211.09066*, 2022.

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

Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-time Systems*, pages 255–299, 1990.

E Allen Emerson. Temporal and modal logic. In *Formal Models and Semantics*, pages 995–1072. Elsevier, 1990.

Philippe Schnoebelen. The complexity of temporal logic model checking. *Advances in modal logic*, 4(393-436):35, 2002.

Przemysław A Wałęga, Bernardo Cuenca Grau, Mark Kaminski, and Egor V Kostylev. Datalogmtl over the integer timeline. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 17, pages 768–777, 2020.

Przemysław A Wałęga, Michał Zawidzki, and Bernardo Cuenca Grau. Finitely materialisable datalog programs with metric temporal operators. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 18, pages 619–628, 2021.

Przemysław A Wałęga, Michał Zawidzki, Dingmin Wang, and Bernardo Cuenca Grau. Materialisation-based reasoning in datalogmtl with bounded intervals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6566–6574, 2023.

Kevin C Klement. Propositional logic. 2004.

Jon Barwise. An introduction to first-order logic. In *Studies in Logic and the Foundations of Mathematics*, volume 90, pages 5–46. Elsevier, 1977.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. Question answering over temporal knowledge graphs. *arXiv preprint arXiv:2106.01515*, 2021.

Mete Ismayilzada, Debjit Paul, Syrielle Montariol, Mor Geva, and Antoine Bosselut. Crow: Benchmarking commonsense reasoning in real-world tasks. *arXiv preprint arXiv:2310.15239*, 2023.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92, 2021.

Appendix

A Datasheets for Datasets

To help the community better understand the dataset, we present the datasheets of the `t-BEN` dataset, according to Gebru et al. [2021].

A.1 Motivation - Purpose

The dataset is used as a benchmark to test the LLM's reasoning ability on temporal logic. Temporal logic reasoning involves both logic reasoning and numerical reasoning, and the ability is useful in many downstream tasks. The benchmark specifically addressed the bias issue caused by data leakage by generating data randomly and automatically. Since it can be scaled up easily, it might also be used to fine tune a model to enhance its reasoning abilities.

Motivation - Creators / Funding Those information will be disclosed once the paper is accepted.

Composition - Instance All instances in the dataset are a temporal reasoning question written in DatalogMTL.

Type of Sample	# of Positive Samples	# of Negative Samples
SingleAtom	500	500
MultiAtoms	300	300
Rational	500	500
MixedOperators (with 2 operators)	1739	1739
MixedOperators (with 3 operators)	145	145
MixedOperators (with 3 operators)	126	126
MultiRules (with 2 rules)	250	250
MultiRules (with 4 rules)	250	250
MultiRules (with 6 rules)	150	150
Recursive	500	500

Table 2: The number of samples of different categories in our dataset

Composition - Size Depending on the complexity of the reasoning problems, we divided the dataset into six sub dataset, the number of instances are listed in Table 2.

For MultiAtoms, we don't specify the number of operators it has in the rule nor evaluate them separately, while in general it follows the following distribution presented in Table 3.

Note that the dataset doesn't contain all possible instances. There are infinite number of possible instances.

Composition - Instance Details Each instance contains a data field, which is a set of the known variables, a set of rules, a single query and a boolean value indicating that if the query is true. They are represented in JSON format.

Composition - Label Yes, the label is presented for each instace in the dataset.

Type of Sample	# of Positive Samples	# of Negative Samples
MultiAtoms (with 2 atoms in the rule)	109	115
MultiAtoms (with 3 atoms in the rule)	79	79
MultiAtoms (with 4 atoms in the rule)	61	64
MultiAtoms (with 5 atoms in the rule)	51	42
Total	300	300

Table 3: The distribution of the number of atoms in our MultiAtoms subset of our dataset

Composition - Missing Information No, all information is completed.

Composition - Relationships All instances are independent in our dataset.

Composition - Splits There isn't a recommended data split for our dataset.

Composition - Errors No, there isn't any error in our dataset. All instances are verified to be correct.

Composition - Self-contained Yes, the dataset is self-contained, no external resource is required.

Composition - Confidentiality No, all data is considered as public.

Collection The dataset is generated automatically without input from the real world. The generation algorithm is presented in Section 4

Processing We used The Metric Temporal Reasoner (MeTeoR) to verify all generated instances.

Use The dataset is intended to be used as a metric to evaluate the general LLM's temporal reasoning ability.

Distribution The dataset will be publicly available on HuggingFace with no restrictions on re-distribution.

B Prompts Used in the basic evaluation

For all evaluations, we prepend a system message to introduce the syntax of DatalogMTL language as below:

You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b.

Zero-shot For zero-shot evaluations, as well as all DeepSeek evaluations, the system prompt we uses is the above general introduction plus the statement: *You should not give any explanation and you should only output "true" or "false"*. We are using the statement *Now we have some temporal data and some rules, data: {data} rule: {rule}, Is {inquiry} true or not?* as the user prompt to evaluate LLM's reasoning ability.

Here is an example of the complete prompt we constructed to do zero-shot evaluation.

System Prompt	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b. You should not give any explanation and you should only output "true" or "false"
User Prompt	Now we have some temporal data and some rules, data: B@[3,10] rule: A:-Diamondplus[6,10]B Is A@[1,4] true or not?
LLM's output	false
Expected Answer	true

Few-shot For few-shot evaluations, just like the zero-shot case, the system prompt we uses is the above general introduction plus the statement: *You should not give any explanation and you should only output "true" or "false"*. However, in the user prompt, we are integrating some examples using the following syntax:

To help you better understand the task, I will provide two examples. Example 1: data: {pos data} rule: {pos rule} in this case you should output "true" for {pos inquiry}. Example 2: data: {neg data} rule: {neg rule} in this case you should output "false" for {neg inquiry}. Now we have some temporal data and some rules, data: {data} rule: {rule} Is {inquiry} true or not?"

{pos data}, *{pos rule}* and *{pos inquiry}* are from a positive sample, *{neg data}*, *{neg rule}* and *{neg inquiry}* are from a negative sample. They are samples not in the testing set, but has the same type as the testing samples.

Here is an example of the complete prompt we constructed to do few-shot evaluation.

System Prompt	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b. You should not give any explanation and you should only output "true" or "false"
User Prompt	To help you better understand the task, I will provide two examples. Example 1: data: B@[5,7] rule: A:-Boxminus[10,12]B in this case you should output "true" for A@[17,17] Example 2: data: B@[1,9] rule: A:-Diamondplus[3,3]B in this case you should output "false" for A@[-25,-6] Now we have some temporal data and some rules, data: B@[3,10] rule: A:-Diamondplus[6,10]B Is A@[1,4] true or not?"
LLM's output	false
Expected Answer	true

Zero-shot-CoT For zero-shot-cot evaluations, the system prompt we uses is the above general introduction without the additional the statement that we used in zero-shot or few-shot to ask LLM's to answer directly. We are going to put some directions to let LLM think step by step in the user prompt below:

Now we have some temporal data and some rules, data: {data} rule: {rule} Is {inquiry} true or not? Do not answer directly, think step by step.

After that, we will put the LLM's response in the context and then asks LLM again about the final answer by *Based on your previous response, without any explanation state your answers to the question. You should only output "true" or "false".*

Here is an example of the complete prompt we constructed to do zero-shot evaluation.

System Prompt	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b.
User Prompt	Now we have some temporal data and some rules, data: B@[3,10] rule: A:-Diamondplus[6,10]B Is A@[1,4] true or not? Do not answer directly, think step by step.
LLM's output	First, let's understand the rule: A:-Diamondplus[6,10]B. This rule means that if B is true at some point between t+6 and t+10, then A is true at time t. Now, let's look at the data: B@[3,10]. This means that B is true at some point between time 3 and time 10. We are asked to determine if A@[1,4] is true or not. This means we are looking to see if A is true at some point between time 1 and time 4. Given the rule, for A to be true at time 1, B would need to be true at some point between time 7 (1+6) and 11 (1+10). For A to be true at time 4, B would need to be true at some point between time 10 (4+6) and 14 (4+10). However, B is only true between time 3 and time 10. Therefore, B is not true at any point between time 7 and 11 or between time 10 and 14. So, A@[1,4] is not true.
User Prompt	Based on your previous response, without any explanation state your answers to the question. You should only output "true" or "false"
LLM's output	false
Expected Answer	true

C Prompts Used in the natural language evaluation

For the evaluations using natural language, we no longer need to introduce the syntax of DatalogMTL language. Therefore there is no general system prompt.

Zero-shot For zero-shot setting, we change the system prompt to a simple sentence to ensure that LLMs output in a desired format *You should not give any explanation and you should only output "true" or "false"*. We are using the statement *Now we have some temporal data and some rules, data: {data} rule: {rule}, Is {inquiry} true or not?* as the user prompt to evaluate LLM's reasoning ability. {data}, {rule} and {inquiry} are all replaced by their verbalized representation.

Here is an example of the complete prompt we constructed to do zero-shot evaluation.

System Prompt	You should not give any explanation and you should only output "true" or "false"
User Prompt	Now we have some temporal data and some rules, data: A holds From 10.000 to 10.000 rule: B holds in each time such that A will hold sometime between 4.000 and 15.000 hours in the future Is B holds From -5.000 to 1.000 true or not?
LLM's output	false
Expected Answer	true

Few-shot For few-shot evaluations, just like the zero-shot case, the system prompt we uses is the same: *You should not give any explanation and you should only output "true" or "false"*. However, in the user prompt, we are integrating some examples using the following syntax:

To help you better understand the task, I will provide two examples. Example 1: data: {pos data} rule: {pos rule} in this case you should output "true" for {pos inquiry}. Example 2: data: {neg data} rule: {neg rule} in this case you should output "false" for {neg inquiry}. Now we have some temporal data and some rules, data: {data} rule: {rule}

{pos data}, *{pos rule}* and *{pos inquiry}* are verbalized representations from a positive sample, *{neg data}*, *{neg rule}* and *{neg inquiry}* are verbalized representations from a negative sample. They are samples not in the testing set, but has the same type as the testing samples.

Here is an example of the complete prompt we constructed to do few-shot evaluation.

System Prompt	You should not give any explanation and you should only output "true" or "false"
User Prompt	To help you better understand the task, I will provide two examples. Example 1: data: B holds From 5.000 to 7.000 rule: A holds in each time such that B did hold continuously for 10.000 and 12.000 hours before in this case you should output "true" for A holds From 17.000 to 17.000 Example 2: data: B holds From 1.000 to 9.000 rule: A holds in each time such that B will hold sometime between 3.000 and 3.000 hours in the future in this case you should output "false" for A holds From -25.000 to -6.000 Now we have some temporal data and some rules, data: B holds From 3.000 to 10.000 rule: A holds in each time such that B will hold sometime between 6.000 and 10.000 hours in the future Is A holds From 1.000 to 4.000 true or not?
LLM's output	true
Expected Answer	true

Zero-shot-CoT For verbalized zero-shot-cot evaluations, we no longer use system message since we don't need to introduce the synatx nor ask it to answer in a specific format. We organize our user prompts in the following format:

Now we have some temporal data and some rules, data: {data} rule: {rule} Is {inquiry} true or not? Do not answer directly, think step by step.

where {data}, {rule} and {inquiry} refers to the verbalized rules.

After that, we will put the LLM's response in the context and then asks LLM again about the final answer by *Based on your previous response, without any explanation state your answers to the question. You should only output "true" or "false".*

Here is an example of the complete verbalized prompt we constructed to do zero-shot evaluation.

System Prompt	
User Prompt	Now we have some temporal data and some rules, data: B holds From 3.000 to 10.000 rule: A holds in each time such that B will hold sometime between 6.000 and 10.000 hours in the future Is A holds From 1.000 to 4.000 true or not? Do not answer directly, think step by step.
LLM's output	To determine if A holds from 1.000 to 4.000 based on the given rule and data, let's break it down step by step: 1. **Understand the Data:** - B holds from 3.000 to 10.000. 2. **Understand the Rule:** - A holds at each time such that B will hold sometime between 6.000 and 10.000 hours in the future. 3. **Analyze the Time Intervals:** - We need to check if B holds in the interval [6.000, 10.000] hours in the future from any time within [1.000, 4.000]. 4. **Check Each Time Point in [1.000, 4.000]:** - For time 1.000: - 6.000 to 10.000 hours in the future from 1.000 is [7.000, 11.000]. - B holds from 3.000 to 10.000, which overlaps with [7.000, 10.000]. - Therefore, B holds sometime in [7.000, 11.000]. - For time 2.000: - 6.000 to 10.000 hours in the future from 2.000 is [8.000, 12.000]. - B holds from 3.000 to 10.000, which overlaps with [8.000, 10.000]. - Therefore, B holds sometime in [8.000, 12.000]. - For time 3.000: - 6.000 to 10.000 hours in the future from 3.000 is [9.000, 13.000]. - B holds from 3.000 to 10.000, which overlaps with [9.000, 10.000]. - Therefore, B holds sometime in [9.000, 13.000]. - For time 4.000: - 6.000 to 10.000 hours in the future from 4.000 is [10.000, 14.000]. - B holds from 3.000 to 10.000, which overlaps with [10.000, 10.000]. - Therefore, B holds sometime in [10.000, 14.000]. 5. **Conclusion:** - For each time point in the interval [1.000, 4.000], B holds sometime in the future interval [6.000, 10.000] hours from that time point. - Therefore, according to the rule, A holds from 1.000 to 4.000. So, based on the given rule and data, A holds from 1.000 to 4.000.
User Prompt	Based on your previous response, without any explanation state your answers to the question. You should only output "true" or "false"
LLM's output	true
Expected Answer	true

D Bad Case Analysis

D.1 Cases that are incorrect in symbolized representation but are correct in verbalized representation

	Symbolic	Verbalized

System Prompt	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b.	(No introduction to the syntax is required for verbalized representations)
User Prompt	Now we have some temporal data and some rules, data: B@[3,10] rule: A:-Diamondplus[6,10]B Is A@[1,4] true or not? Do not answer directly, think step by step.	Now we have some temporal data and some rules, data: B holds From 3.000 to 10.000 rule: A holds in each time such that B will hold sometime between 6.000 and 10.000 hours in the future Is A holds From 1.000 to 4.000 true or not? Do not answer directly, think step by step.

LLM's output	First, let's understand the rule: A:-Diamondplus[6,10]B. This rule means that if B is true at some point between t+6 and t+10, then A is true at time t. Now, let's look at the data: B@[3,10]. This means that B is true at some point between time 3 and time 10. We are asked to determine if A@[1,4] is true or not. This means we are looking to see if A is true at some point between time 1 and time 4. Given the rule, for A to be true at time 1, B would need to be true at some point between time 7 (1+6) and 11 (1+10). For A to be true at time 4, B would need to be true at some point between time 10 (4+6) and 14 (4+10). However, B is only true between time 3 and time 10. Therefore, B is not true at any point between time 7 and 11 or between time 10 and 14. So, A@[1,4] is not true.	To determine if A holds from 1.000 to 4.000 based on the given rule and data, let's break it down step by step: 1. **Understand the Data:** - B holds from 3.000 to 10.000. 2. **Understand the Rule:** - A holds at each time such that B will hold sometime between 6.000 and 10.000 hours in the future. 3. **Analyze the Time Intervals:** - We need to check if B holds in the interval [6.000, 10.000] hours in the future from any time within [1.000, 4.000]. 4. **Check Each Time Point in [1.000, 4.000]:** - For time 1.000: - 6.000 to 10.000 hours in the future from 1.000 is [7.000, 11.000]. - B holds from 3.000 to 10.000, which overlaps with [7.000, 10.000]. - Therefore, B holds sometime in [7.000, 11.000]. - For time 2.000: - 6.000 to 10.000 hours in the future from 2.000 is [8.000, 12.000]. - B holds from 3.000 to 10.000, which overlaps with [8.000, 10.000]. - Therefore, B holds sometime in [8.000, 12.000]. - For time 3.000: - 6.000 to 10.000 hours in the future from 3.000 is [9.000, 13.000]. - B holds from 3.000 to 10.000, which overlaps with [9.000, 10.000]. - Therefore, B holds sometime in [9.000, 13.000]. - For time 4.000: - 6.000 to 10.000 hours in the future from 4.000 is [10.000, 14.000]. - B holds from 3.000 to 10.000, which overlaps with [10.000, 10.000]. - Therefore, B holds sometime in [10.000, 14.000]. 5. **Conclusion:** - For each time point in the interval [1.000, 4.000], B holds sometime in the future interval [6.000, 10.000] hours from that time point. - Therefore, according to the rule, A holds from 1.000 to 4.000. So, based on the given rule and data, A holds from 1.000 to 4.000.
User Prompt	Based on your previous response, without any explanation state your answers to the question. You should only output "true" or "false"	
LLM's output	false	true
Expected Answer	true	

D.2 Cases that are incorrect in zero-shot by are correct when chain of thought is applied

Example 1:

	Without CoT	With CoT
System Prompt	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b. You should not give any explanation and you should only output "true" or "false"	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b.
User Prompt	Now we have some temporal data and some rules, data: B@[6,9] rule: A:-Diamondminus[6,15]B Is A@[12,18] true or not?	Now we have some temporal data and some rules, data: B@[6,9] rule: A:-Diamondminus[6,15]B Is A@[12,18] true or not? Do not answer directly, think step by step.
LLM's output		First, let's understand the rule: A:-Diamondminus[6,15]B. This rule means that if B is true at some point between t-15 and t-6, then A is true at time t. Now, let's consider the fact A@[12,18]. This means we are checking if A is true at some point between time 12 and 18. To check this, we need to see if B is true at some point between (12-15) and (12-6), i.e., between time -3 and 6, or between (18-15) and (18-6), i.e., between time 3 and 12. From the given data, we know that B is true between time 6 and 9. This falls within the required time range for A to be true. Therefore, A@[12,18] is true.

User Prompt		Based on your previous response, without any explanation state your answers to the question. You should only output "true" or "false"
LLM's output	false	true
Expected Answer	true	

Example 2:

	Without CoT	With CoT
System Prompt	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b. You should not give any explanation and you should only output "true" or "false"	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b.
User Prompt	Now we have some temporal data and some rules, data: A@[8,13] rule: B:-Diamondminus[10,12]A Is B@[21,24] true or not?	Now we have some temporal data and some rules, data: A@[8,13] rule: B:-Diamondminus[10,12]A Is B@[21,24] true or not? Do not answer directly, think step by step.

LLM's output		First, let's understand the rule: B:-Diamondminus[10,12]A. This rule means that if A is true at some point between t-12 and t-10, then B is true at time t. Now, let's look at the data: A@[8,13]. This means that A is true continuously from time 8 to time 13. Now, we need to check if B@[21,24] is true or not. For B to be true at any time between 21 and 24, A needs to be true at some point between (t-12) and (t-10). Let's check for each time t in [21,24]: - For t=21, A needs to be true at some point between 21-12=9 and 21-10=11. A is true at these times according to the data. - For t=22, A needs to be true at some point between 22-12=10 and 22-10=12. A is true at these times according to the data. - For t=23, A needs to be true at some point between 23-12=11 and 23-10=13. A is true at these times according to the data. - For t=24, A needs to be true at some point between 24-12=12 and 24-10=14. A is true at time 12, but not at time 14. However, the rule requires A to be true at some point between these times, not at all times, so this condition is satisfied. Therefore, B@[21,24] is true.
User Prompt		Based on your previous response, without any explanation state your answers to the question. You should only output "true" or "false"
LLM's output	false	true
Expected Answer	true	

D.3 SingleAtom Error Cases

False Negative Example:

System Prompt	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows:If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a.If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a.If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b.If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b.

User Prompt	Now we have some temporal data and some rules, data: A@[7,15] rule: B:-Boxminus[3,5]A Is B@[8,30] true or not? Do not answer directly, think step by step.
LLM's output	To determine whether (B@[8,30]) is true given the data and the rule, we need to follow these steps: 1. **Understand the Rule**: The rule is (B leftarrow Boxminus[3,5]A). This means that (B) is true at time (t) if (A) is true continuously in the interval ([t-5, t-3]). 2. **Check the Interval for (B@[8,30])**: We need to check if (B) is true for every time point in the interval ([8, 30]). For each time (t) in this interval, (A) must be true continuously in the interval ([t-5, t-3]). 3. **Analyze the Data**: The data given is (A@[7,15]), which means (A) is true continuously from time 7 to time 15. 4. **Check Each Time Point in ([8, 30])**: - For (t = 8): The interval ([t-5, t-3]) is ([3, 5]). (A) is not true in this interval because (A) is only true from 7 to 15. - For (t = 9): The interval ([t-5, t-3]) is ([4, 6]). (A) is not true in this interval. - For (t = 10): The interval ([t-5, t-3]) is ([5, 7]). (A) is not true continuously in this interval (it starts at 7). - For (t = 11): The interval ([t-5, t-3]) is ([6, 8]). (A) is true continuously in this interval. - For (t = 12): The interval ([t-5, t-3]) is ([7, 9]). (A) is true continuously in this interval. - For (t = 13): The interval ([t-5, t-3]) is ([8, 10]). (A) is true continuously in this interval. - For (t = 14): The interval ([t-5, t-3]) is ([9, 11]). (A) is true continuously in this interval. - For (t = 15): The interval ([t-5, t-3]) is ([10, 12]). (A) is true continuously in this interval. - For (t = 16): The interval ([t-5, t-3]) is ([11, 13]). (A) is true continuously in this interval. - For (t = 17): The interval ([t-5, t-3]) is ([12, 14]). (A) is true continuously in this interval. - For (t = 18): The interval ([t-5, t-3]) is ([13, 15]). (A) is true continuously in this interval. - For (t = 19): The interval ([t-5, t-3]) is ([14, 16]). (A) is not true in this interval (it ends at 15). - For (t = 20) to (t = 30): The intervals ([t-5, t-3]) will be ([15, 17]) to ([25, 27]). (A) is not true in any of these intervals. 5. **Conclusion**: (B) is not true for all time points in the interval ([8, 30]). Specifically, (B) is not true at (t = 8, 9, 10, 19,) and any (t) from 20 to 30. Therefore, (B@[8,30]) is not true.
User Prompt	Based on your previous response, without any explanation state your answers to the question. You should only output "true" or "false"
LLM's output	false
Expected Answer	true

False Positive Example:

System Prompt	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows:If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a.If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a.If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b.If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b.
User Prompt	Now we have some temporal data and some rules, data: A@[0,3] rule: B:-Diamondminus[2,4]A Is B@[-34,8] true or not? Do not answer directly, think step by step.
LLM's output	To determine whether (B@[-34,8]) is true given the data and the rule, we need to follow these steps: 1. **Understand the Rule**: The rule is (B leftarrow Diamondminus[2,4]A). This means that (B) is true at time (t) if (A) is true at some time between (t-4) and (t-2). 2. **Determine the Time Interval for (B@[-34,8])**: We need to check if (B) is true at any time within the interval ([-34, 8]). 3. **Apply the Rule to Each Time Point in the Interval**: - For each time (t) in the interval ([-34, 8]), we need to check if (A) is true at some time between (t-4) and (t-2). 4. **Check the Data**: The data given is (A@[0,3]), which means (A) is true from time 0 to time 3. 5. **Evaluate the Rule for Each Time Point**: - For (t = -34): We need (A) to be true between ([-38, -36]). Since (A) is only true between ([0, 3]), (A) is not true in this interval. - For (t = -33): We need (A) to be true between ([-37, -35]). Again, (A) is not true in this interval. - Continue this process up to (t = 8): - For (t = 0): We need (A) to be true between ([-4, -2]). (A) is not true in this interval. - For (t = 1): We need (A) to be true between ([-3, -1]). (A) is not true in this interval. - For (t = 2): We need (A) to be true between ([-2, 0]). (A) is not true in this interval. - For (t = 3): We need (A) to be true between ([-1, 1]). (A) is true at time 0, which is within this interval. - For (t = 4): We need (A) to be true between ([0, 2]). (A) is true in this interval. - For (t = 5): We need (A) to be true between ([1, 3]). (A) is true in this interval. - For (t = 6): We need (A) to be true between ([2, 4]). (A) is true at time 2 and 3, which is within this interval. - For (t = 7): We need (A) to be true between ([3, 5]). (A) is true at time 3, which is within this interval. - For (t = 8): We need (A) to be true between ([4, 6]). (A) is not true in this interval. 6. **Conclusion**: Since (A) is true at some time between (t-4) and (t-2) for (t = 3, 4, 5, 6,) and (7), (B) is true at these times. Therefore, (B) is true at some points within the interval ([-34, 8]). Thus, (B@[-34,8]) is true.
User Prompt	Based on your previous response, without any explanation state your answers to the question. You should only output "true" or "false"
LLM's output	true
Expected Answer	false

613 ### D.4 Comparison between CoT prompt and DeepSeek reasoning

614 Example 1:

Table 4: A detailed context comparison between GPT-4o with CoT and DeepSeek-R1-Distill-Qwen-32B

	Qwen2.5-32B-Instruct with CoT	DeepSeek-R1-Distill-Qwen-32B
System Prompt	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b.	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b. You should not give any explanation and you should only output "true" or "false"
User Prompt	Now we have some temporal data and some rules, data: A@[9,12] rule: B:-Diamondminus[3,13]A Is B@[25,25] true or not? Do not answer directly, think step by step.	Now we have some temporal data and some rules, data: A@[9,12] rule: B:-Diamondminus[3,13]A Is B@[25,25] true or not?

LLM's output	To determine if B@[25,25] is true given the rule and the data, we need to follow these steps: 1. **Understand the Rule**: The rule is B:-Diamondminus[3,13]A. This means that B is true at a time t if A is true at some point between t-13 and t-3. 2. **Identify the Time Interval for B**: We are checking if B@[25,25] is true. This means we are looking at the time t = 25. 3. **Determine the Relevant Time Interval for A**: According to the rule, for B to be true at t = 25, A must be true at some point between 25-13 and 25-3, which is between 12 and 22. 4. **Check the Data for A**: The data provided is A@[9,12]. This means A is true from time 9 to time 12. 5. **Compare the Intervals**: The interval for A (9 to 12) does not overlap with the interval required by the rule for B (12 to 22). Specifically, A is only true up to time 12, and the rule requires A to be true at some point between 12 and 22. 6. **Conclusion**: Since A is not true at any point between 12 and 22, B@[25,25] cannot be true based on the given rule and data. Therefore, B@[25,25] is not true.	Okay, so I need to figure out if B is true at time 25 based on the given data and rule. Let me start by understanding the problem step by step. First, the data provided is A@[9,12]. This means that A is true from time 9 up to and including time 12. So, A is active during the interval [9,12]. Next, the rule given is B:-Diamondminus[3,13]A. I need to parse this rule correctly. In DatalogMTL, the Diamondminus operator is used, which has a specific meaning. From the semantics provided, Diamondminus[a,b]A being true at time t means that A must be true at some time between t - b and t - a. So, breaking down the rule: B is derived if there exists some time within the interval [t - 13, t - 3] where A is true. Therefore, for B to hold at time t, A must be true at least once in that interval. Now, the question is whether B is true at time 25, specifically at the point [25,25]. So, t is 25. Let's compute the interval for A based on the rule. The interval for A would be from t - b to t - a, which is 25 - 13 to 25 - 3. Calculating that: 25 - 13 = 12 25 - 3 = 22 So, the interval is [12,22]. This means that for B to be true at 25, A must be true at some point between 12 and 22. Looking back at the data, A is true from 9 to 12. So, A is true at 12, which is the end of its interval. Now, does 12 fall within [12,22]? Yes, it does. Therefore, A is true at 12, which is within the required interval for B to hold at 25. Wait, but I should double-check. The interval for A is [9,12], so A is true at 12. The rule requires A to be true at some point between 12 and 22. Since 12 is included in both intervals, A is true at 12, which satisfies the condition. Therefore, B should be true at time 25.
User Prompt	Based on your previous response, without any explanation state your answers to the question. You should only output "true" or "false"	
LLM's output	false	true

Expected Answer	true

Example 2:

Table 5: A detailed context comparison between GPT-4o with CoT and DeepSeek-R1-Distill-Qwen-32B

	Qwen2.5-32B-Instruct with CoT	DeepSeek-R1-Distill-Qwen-32B
System Prompt	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b.	You are given a dataset and a temporal rule, and your task is to judge whether the given fact is entailed by the dataset and the rule. The rules are expressed as DatalogMTL, a knowledge representation language that extends Datalog with operators from metric temporal logic (MTL). The semantics of four MTL operators are given as follows: If Diamondminus[a,b]A is true at the time t, it requires that A needs to be true at some time between t-b and t-a. If Boxminus[a,b]A is true at the time t, it requires that A needs to be true continuously between t-b and t-a. If Diamondplus[a,b]A is true at the time t, it requires that A needs to be true at some point between t+a and t+b. If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b. You should not give any explanation and you should only output "true" or "false"
User Prompt	Now we have some temporal data and some rules, data: A@[6,6] rule: A:-Diamondminus[2,2]A Is A@[80,80] true or not? Do not answer directly, think step by step.	Now we have some temporal data and some rules, data: A@[6,6] rule: A:-Diamondminus[2,2]A Is A@[80,80] true or not?

LLM's output	To determine whether A@[80,80] is true given the rule A:-Diamondminus[2,2]A and the data A@[6,6], we need to follow these steps: 1. **Understand the Rule**: The rule A:-Diamondminus[2,2]A means that A is true at time t if A is true at some point between t-2 and t+2. 2. **Analyze the Data**: The data provided is A@[6,6], which means A is true at time 6. 3. **Apply the Rule to the Data**: According to the rule, A@[t] is true if A is true at some point between t-2 and t+2. We need to check if A@[6,6] can satisfy the rule for any time t. 4. **Check the Time Interval for A@[80,80]**: We need to see if A@[80,80] can be true based on the rule. For A@[80,80] to be true, A must be true at some point between 80-2 and 80+2, i.e., between 78 and 82. 5. **Compare the Given Data with the Required Interval**: The data A@[6,6] indicates that A is true at time 6. However, this does not fall within the interval [78, 82]. 6. **Conclusion**: Since A@[6,6] does not fall within the interval [78, 82], A@[80,80] cannot be true based on the given rule and data. Therefore, A@[80,80] is not true.	Okay, so I'm trying to figure out whether A@[80,80] is true based on the given data and rule. Let me break this down step by step. First, the data provided is A@[6,6]. That means at time 6, A is true. Now, the rule is A:-Diamondminus[2,2]A. I need to understand what this rule means. The Diamondminus operator in MTL is defined such that Diamondminus[a,b]A is true at time t if A is true at some time between t-b and t-a. So, for the rule A:-Diamondminus[2,2]A, it means that A is true at time t if there exists some time between t-2 and t-2 (since a and b are both 2) where A is true. Wait, that seems a bit confusing. Let me clarify. The operator Diamondminus[2,2]A at time t requires that A is true at some point between t-2 and t-2. But t-2 and t-2 is just a single point in time, t-2. So, effectively, this rule is saying that A is true at time t if A was true at time t-2. So, applying this rule, if A is true at time t-2, then A is true at time t. Now, looking at the data, A is true at time 6. So, according to the rule, A should also be true at time 6 + 2 = 8. Then, applying the rule again, A at 8 would imply A at 10, and so on. This creates a chain where A is true at every even time starting from 6: 6, 8, 10, 12, etc. But the question is about A@[80,80]. Is 80 an even number? Yes, 80 is even. So, following the chain, A should be true at 80 because it's 6 + 2*37 = 80. Therefore, A is true at 80. Wait, but let me double-check. The rule is A:-Diamondminus[2,2]A, which as I understood, means A at t if A was at t-2. So starting from 6, A is true at 8, then 10, and so on. So 80 is 6 + 2*37, which is correct. So yes, A should be true at 80. I think that's the correct reasoning. So the answer should be true.
User Prompt	Based on your previous response, without any explanation state your answers to the question. You should only output "true" or "false"	
LLM's output	false	true
Expected Answer	true	

E Can LLM Understand Symbolic Representation?

We ask LLM to translate generated natural language representations of logic rules to symbolic representations and them compare the translated symbolic representations with the ground truth symbolic representation to verify if LLM has the ability to understand symbolic representations. Specifically, we passed the same prompt that used in our symbolic evaluations, "The rules are expressed as DatalogMTL, a......If Boxplus[a,b]A is true at the time t, it requires that A needs to be true continuously between t+a and t+b.", into LLMs, along with few examples telling LLM the output format, then ask LLM to translate verbalized samples into symbolic ones. If the translated symbolic rule from the verbalized rule is exactly the same as the original symbolic rule, then we consider LLM has the ability to understand both the symbolic rule and the verbalized rule.

We passed 50 samples selected from **MultiRules** subset, which is considered the most challenging, into the LLMs. LLM accurately translated 96% of testing samples from verbalized representations to symbolic representations.

In addition, we noticed that larger LLMs with strong reasoning abilities, such as DeepSeek-R1, performs pertty good on some cases, further proving that the semantics is understood.

Considering all those points, We believe that LLM can understand the symbolic representation.

F Detailed Benchmark Construction Pseudo Code

Our dataset generation algorithm is driven by generating rules. In a high level view, it generate rules one by one in a same context, while the generation process for each rule contains the context check, ensuring the generated rules are non-trivial.

Algorithm 1: Generate

Parameters : f: The set of features Enabled
Parameters : N: The number of rules
Parameters : V: A boolean flag to control if the program should generate a positive sample or a negative sample
Output: A problem instance I containing a set of rules, a set of data, a query and a boolean value representing whether the query is valid or not.

$G \leftarrow EmptyGraph()$;
while *i in 1.....N* **do**
  do
    $G \leftarrow GenerateGraph(G)$;
    while *n in $G.nodes$* **do**
      Assign node with random values
    end
    $G \leftarrow GenerateRules(G)$
  while *New Info can be Inferred from I*;
end
$Rules, Data \leftarrow Extract\ Rules\ associted\ with\ G$;
$DeltaNew \leftarrow Facts\ Inferred\ From\ Graph\ G$;
$QueryEntity, Interval \leftarrow$ Randomly Select From $DeltaNew$;
if *V* **then**
  $QueryInterval \leftarrow$ A random sub-interval from $Interval$;
else
  $QueryInterval \leftarrow$ A random sub-interval that is not in $Interval$;
end
return *Rules, Data, QueryEntity, QueryInterval, V*

The graph generation algorithm 2 will generate a graph where nodes in the graph represents predicates such as A, B and C. We are going to attach details information about predicates and rules into the corresponding nodes and edges of the graph, but at this time we only need the structure of the graph, i.e. nodes and edges don't have special information attached.

Algorithm 2: Graph Generation

Input: G: The existing graph
Parameters : f: The set of features Enabled
Output: G: The generated graph (including the old information in the existing graph)
Output: $List[V]$: The list of new nodes, representing predicates, in the new graph
Output: V_o: The output node which depends on the some other nodes (in case that recursive is not enabled in f) in $List[V]$

$NewNode \leftarrow []$
Determine the lowest possible number of nodes to add l and the highest number of possible nodes to add r based on f.
$N \leftarrow random(l, r)$;
while *i in 1.....N* **do**
  $p \leftarrow A\ randomly\ assigned\ predicate$;
  $G.AddNode(p)$;
  $NewNode.Push(p)$;
end
$OutNode \leftarrow RandomSelect(NewNode)$;
while *p in NewNode* **do**
  if *"recursive" not in f and p == $OutNode$* **then**
    continue;
  end
  $G.AddEdge(p, OutNode)$
end
return *G, NewNode, OutNode*

After the structure of the graph is generated, we are going to attach rule information to each edge of the graph using the Rule Generation algorithm 3. Since we are doing Graph Generation and Rule Generation alternately, in the rule generation we only care about edges that don't already has a rule, we will skip the edges that already has a rule associated with that.

Algorithm 3: Rule Generation

Input: G: The existing graph
Parameters : f: The set of features Enabled
Output: G: The generated graph (including the old information in the existing graph)

$SelectedOp \leftarrow Set()$
$SelectedOp.add(RandomSelect(Boxminus, Boxplus, Diamondplusm, Diamondminus))$
if *"mixed_operators" in f* **then**
  Randomly select and add more operators to SelectedOp;
end
while *$Edge$ in G* **do**
  $u, v, a \leftarrow G$;
  if *No rule is associated with $Edge$* **then**
    $Op \leftarrow$ Randomly select an operator from $SelectedOp$;
    $Interval \leftarrow$ Randomly create an interval;
    Create an item literal with Op and $Interval$ and associated that with $Edge$;
  end
end
return G

G Computational Resource Requirement

For LLama-3-8B and Qwen2.5-32B, we used two NVIDIA H100 80GB HBM3 GPUs, and hosted using vLLM. Zero-shot and few-shot inference usually take less than 10 mintues, and chain-of-thought usually takes less than 1 hour.

For Distilled DeepSeek Models, we used two NVIDIA H100 80GB HBM3 GPUs, and inference usually takes less than 1 hour.

For DeepSeek R1, we used the cloud inference platform Fireworks AI[5], and the full evaluation takes less than $10 USD.

For GPT-4o, we used the cloud inference platform OpenAI [6]. The full evaluation takes less than $100 USD.

H Limitation

Our experiments were constrained by the speed, computational resources, and financial costs associated with utilizing GPT-4o and DeepSeek-R1. For instance, although our generator allows for the creation of temporal data and rules with arbitrary sizes, we obtained results across multiple temporal reasoning datasets of varying complexities on a relatively small scale due to the financial costs associated with GPT-4o and DeepSeek-R1 API calls.

Another limitation of this preliminary exploration into testing the temporal reasoning abilities of LLMs is that we present experimental results from only three prompting settings, despite the availability of more advanced prompting strategies. Additionally, while our results demonstrate that DeepSeek-R1 and its distilled models significantly outperform the other evaluated models, we do not establish the underlying factors contributing to this superiority. Our human analysis of certain error cases provides limited insights, and we do not propose an effective method for enhancing LLMs' ability to handle temporal logic reasoning problems.

[5]Fireworks AI Platform can be accessed at `https://fireworks.ai/`

[6]The OpenAI Platform can be accessed at `https://platform.openai.com/`

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: Our contributions are clearly stated in the abstract.

   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: We have discussed the limitation in H.

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: [NA]

   Justification: The paper doesn't include theoretical results.

   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 uploaded our full generated dataset and reproduce-able code.

   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: We uploaded our full generated dataset and reproduce-able code.

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: Those information is clearly stated in Section 5.

   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: [No]

   Justification: All experiments are deterministic, there are no training involved in this paper.

   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: The required compute resources are provided in G.

   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: We have reviewed and have confirmed that our work aligned with the NeurIPS Code of Ethics.

   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: [NA]

    Justification: Our work has no negative societal impacts.

    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: [NA]

    Justification: Our work doesn't pose such risks

    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: Open-source models and proprietary models are used during the evaluation process. They are properly cited in our 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: New datasets and our code are well documented in the paper.

    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: [NA]

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

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: [NA]

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

    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: [NA]

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

    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.