Unreasonable effectiveness of LLM reasoning: a doubly cautionary tale of temporal question-answering

Anonymous authors
Paper under double-blind review

Abstract

The remarkable success of Large Language Models in modeling both the syntax and the semantics of language has prompted a body of research into language-adjacent abilities, most notably commonsense reasoning. As LLMs' performance continues to advance on successive benchmarks, we turn to temporal reasoning, which lags somewhat behind other tasks due to its more complex logic. We start from previous work, where authors successfully induce (apparent) reasoning by breaking down the problem into a two-step procedure of temporal graph extraction and subsequent reasoning. Specifically, in the first step an LLM is prompted to parse a natural language description into a semi-structured timeline of events; and in the second step, it is given the extracted timeline and prompted to answer a temporal reasoning question. We conjecture that this procedure presents two separate opportunities for introducing errors and further hypothesise that a Neuro-symbolic approach should help in this matter. We follow the recent trend of using external executors in concert with LLMs to carry out exact reasoning and verification. We see the reasoning step of the original two-step procedure as a natural target for a symbolic solver and design a rule-based solution for Temporal Question-Answering, drawing on ideas from Allen's Interval Algebra. To our surprise, we find that our rule-based reasoner does *not* improve beyond the previously reported, purely neural solution. It appears that both our approach and the previous method operate at around the limits of achievable performance, imposed by the correctness of information extraction. Such a result seems to suggest that a non-symbolic LLM is capable of symbolic-level reasoning, although upon further investigation we discover that not to be the case. It is not that the neural solution makes no reasoning mistakes, but rather that the LLM manages to compensate for some of its erroneous replies by *short-cutting* to the correct answer in other questions; a.k.a. not reasoning but guessing. Although the effect is not pronounced performance-wise, we feel it is conceptually important: as we argue, production of correct answers is *not* a measure of reasoning.

1 Introduction

The unprecedented success of Large Language Models in modeling natural language now includes simulating convincingly human-like chain-of-thought verbalisations of reasoning (Wei et al., 2022b), tool use (Schick et al., 2024), and planning (Huang et al., 2022). Importantly, although trained in a primarily unsupervised paradigm, these architectures are capable of few-shot generalisation to new tasks (Brown et al., 2020), raising questions of emergent capabilities (Wei et al., 2022a). Further, in the short time since their introduction, LLMs continue to be refined: billions of parameters turn to tens of billions, additional strategies such as mixture-of-experts add to their strengths (Jiang et al., 2024), and data added to the training pool continues to fill the gaps at the edges of the distribution. Hand in hand with such improvements comes widespread interest in assessing LLMs, from conversational Turing-tests (Turing, 1950; Sejnowski, 2023) to a range of language comprehension and reasoning benchmarks (Srivastava et al., 2022; Mittal et al., 2024; Duan et al., 2024; Parmar et al., 2024). Research is now coming out at speeds that defy the ability of individual researchers to keep up with the field [1]. Thus, ironically to the subject matter, it would seem that an AI system capable

[1] As attested to by repositories such as e.g. `https://github.com/YiQi0318/LLMs_daily_arxiv/`

of natural language comprehension and reasoning might be what is required to understand the current state of LLMs. Indeed, efforts in that direction have already begun, with researchers proposing fully-automated scientific discovery approaches (Lu et al., 2024). It may be tempting to resort to such tools, particularly as the newest generation of models is marketed as capable of 'reasoning' (Chollet, 2024). However, while undeniably impressive in action, these models are underwritten by a purely neural paradigm, and as such are inherently limited. This fact is well understood among the proponents of Neuro-Symbolic integration (Besold et al., 2021), but also increasingly appreciated more widely (Schaeffer et al., 2024; Pfister & Jud, 2025; Stechly et al., 2025; Lee et al., 2024; Valmeekam et al., 2022; Shojaee et al., 2025). The key issue with the data-driven mode of automated decision-making is that outputs are an opaque blend of data, inductive priors, and chance. As a result, no rule is fixed, and there are no hard guarantees on getting the correct answers, or even ones that are self-consistent (Bao et al., 2024). Perhaps the most salient demonstration of that fickle nature comes from the recent PromptReport, which details an investigation into the 'black art' of prompt engineering (Schulhoff et al., 2024). Authors surface some unusual observations such as that an accidental duplication of part of the prompt can result in improved performance; or that automatic prompt optimisation yields as the most effective a 'word salad' jumble of tokens rather than coherent language. Unsurprisingly then, we and others (Giunchiglia et al., 2025) argue that systems without guarantees are inadmissible in safety-critical applications. Neuro-symbolic integration is not, however, straightforward to achieve. One common avenue to leverage rigorous rule-based systems is tool-augmented LLMs, with 'flavours' such as retrieval-augmented generation (RAG) or python-assisted LLMs (PAL), as well as more specialised ones such as LogicLM (Hu et al., 2022) or ToM-LM (Tang & Belle, 2024). The premise of this approach to integration is that the neural model invokes an external, verifiable tool to perform some or all of the reasoning. In essence, it amounts to acting as a parser between natural language and tool-specific syntax such as Python, Prover9 (McCune, 2005–2010), etc; potentially also relaying back the answers in natural language. For areas with ample training data, such as Python, this can work remarkably well for a majority of common / simpler problems. However, for highly specialised areas such as sound logical reasoning, success tends to be limited, particularly as problem complexity increases (Pan et al., 2023). Nevertheless, the tool-invoking / LLM-as-a-parser route provides an attractive alternative to relying on approaches such as chain-of-thought (CoT)(Wei et al., 2022b) and its variants, which *rationalise their guesses* rather than *reasoning to their answers* (Bao et al., 2024; Stechly et al., 2025). We are interested in this line of *hybrid NeSy* work, and specifically in applying it to temporal reasoning problems.

Although temporal logic is common in everyday life and abundant in natural language, LLMs still struggle with this task (Chen et al., 2021; Fatemi et al., 2024; Tan et al., 2023). The difficulty stems from the fact that reasoning about time requires the application of both logic and arithmetic, and also often commonsense / world knowledge. As Xiong et al. (2024) and others (Su et al., 2024; Tan et al., 2023) demonstrate, it is feasible to improve temporal reasoning within a purely neural paradigm. However, there remains room for improvement, both in terms of performance, but also conceptually and computationally. We aim to demonstrate the value of symbolic and rule-based approaches to address this gap by extending the study of Xiong et al. (2024). We identify the work as a promising starting point because the authors propose to break down the problem into two separate steps: temporal information extraction; and reasoning over the extracted *temporal graphs*. Such a formulation is a natural candidate for hybrid NeSy, as we expect LLMs to be better at extracting information than at reasoning; and, unlike natural language, temporal graphs are suitable inputs for a rule-based reasoner. We implement a custom solution to Temporal Question Answering in Python, inspired by Allen's Interval Algebra (Allen, 1983), which attains a near-perfect score on the ground truth temporal graphs. We then evaluate against the original (Xiong et al., 2024) on the same benchmarks (TimeQA (Chen et al., 2021), TempReason (Tan et al., 2023), TGQA (Xiong et al., 2024)), while retaining the LLM in the function of a parser. Surprisingly, we find no improvement over the reported bootstrapped CoT answers, yet a marked improvement over 'vanilla' CoT. Considering our results on the true temporal graphs, we suspect a form of *short-cutting* is taking place in the LLM. By short-cutting we refer here to the phenomenon of language models selectively attending / responding to the prompt tokens, which can result in ignoring explicit constraints, and further in paradoxical provision of correct answers even when fed false premises. To assess whether it is taking place in our case, we further examine the extracted temporal graphs as well as the individual answers of the LLM reasoner. Specifically, we flag instances where the responses are correct despite the temporal graph itself being wrong. We discover these cases to

constitute approximately 10% of the dataset, confirming our suspicions. Our results thus add to the growing body of research challenging the deceptive nomenclature and marketing of LLMs as capable of reasoning or thinking (Bao et al., 2024; Schaeffer et al., 2024; Stechly et al., 2025). Particularly problematic in this context is the predominant focus on correct performance, which, as we show, can be artificially inflated and deceptive. Arguably, Neuro-symbolic integration is harder than purely connectionist architectures in terms of engineering effort, as well as being inflexible to slight task variations. Further, it does not necessarily offer performance improvements, as in our case. We pose here that such difficulties should not be taken as a cautionary tale against NeSy; on the contrary, they confirm it to be at least a valuable verification tool, if not a target of aspiration of AI as a field.

2 Related work

A number of recent works address the problem of temporal reasoning in language models, noting the particular challenge posed by the logic of time; we focus here on the most directly related ones (Chen et al., 2021; Fatemi et al., 2024; Su et al., 2024; Tan et al., 2023; Xiong et al., 2024). First, Chen et al. (2021) introduce TimeQA, one of the earliest benchmarks designed specifically for *temporal*, or *time-sensitive* question answering (TQA), and one which we include in our evaluation. Since the study predates the latest advancements in LLMs, their solution is to train a custom question-answering encoder-decoder transformer architecture. Although as a purely task-dedicated model it is of less relevance to present work on 'generalist' LLMs, it provides useful context for our results.

Second, Tan et al. (2023) propose TempReason, a more comprehensive benchmark, explicitly aimed at advancing temporal reasoning capabilities of LLMs. Of particular relevance is the fact that they introduce a variant to the typically assumed open-book question answering, whereby rather than providing information to the model in natural language, they provide only the structured facts. Their solution involves pre-training, fine-tuning, and further time-sensitive reinforcement learning, the last one requiring a reference structured timeline of events. This significantly improves over baselines, particularly in the mode of evaluation with structured data in the prompt. However, it should be noted that for each type of temporal reasoning identified by the authors (time-time, time-event, and event-event relationship questions) a separate fine-tuning is applied.

The following study of Xiong et al. (2024), takes the idea of leveraging a structured timeline (Tan et al., 2023) further. The authors introduce TG-LLM, a two-step pipeline for TQA that formally splits the task of reasoning over natural language into, first, temporal information extraction; and second, reasoning. Since datasets featuring natural language and corresponding structured timelines are rare, Xiong et al. (2024) also develop a synthetic dataset, TGQA, which also features new temporal questions. To test the transferrability of their approach and synthetic data, they also use TempReason (Tan et al., 2023) and TimeQA (Chen et al., 2021) datasets. They fine-tune a medium-sized open-source LLM Llama (Touvron et al., 2023), separately for the first and second steps of the workflow. Additionally, for the 'reasoning' model the authors compare a 'vanilla' chain-of-thought (Wei et al., 2022b) against a bootstrapped version, achieving an improvement of 7-30%. As outlined already, this is the step we intend to replace with a rule-based reasoner, which we conjecture should further improve the overall results.

We take note also of two other recent benchmarks, even though we do not utilise them in our work. First of these, the Temporal Reasoning for Large Language Models (TRAM) (Wang & Zhao, 2023) dataset is a comprehensive assessment of temporal understanding and reasoning. The benchmark consists of over 520k samples and spans 38 tasks, which includes a variety of arithmetic, reasoning, factual and narrative / causal questions. Although extensive in the range of topics, it is limited in that the TRAM queries are cast as multiple-choice questions. The study provides also an evaluation of a number of models, noting the variability in performance across tasks and the remaining gap to human performance.

The second notable benchmark is the Test-of-Time (ToT) proposed by Fatemi et al. (2024). Unlike the majority of previous efforts, which tend to draw on existing databases and resources such as WikiData (Vrandečić & Krötzsch, 2014), ToT is an entirely synthetic dataset. Particularly interesting is its *semantic* portion, aimed at reasoning over timelines of events. Random graphs of various structures and sizes are generated programmatically and their nodes and edges are used to fill a text template of the form 'E1 was

the R1 of E2 from 1999 to 2003.. Entities and relations remain fully anonymous, distinguished by numericals. This truly isolates the task of reasoning about time from any linguistic correlations that may exist in the training data of LLMs. It is of particular interest to us, as such data structure supports a fully rule-based reasoner; we return to this in the Discussion.

3 Methods

As illustrated in Figure 1, the original study leveraged a large language model both for extracting the TG from text, as well as reasoning over it. We propose to compare a symbolic reasoner to the LLM, when provided *inferred* temporal graphs.

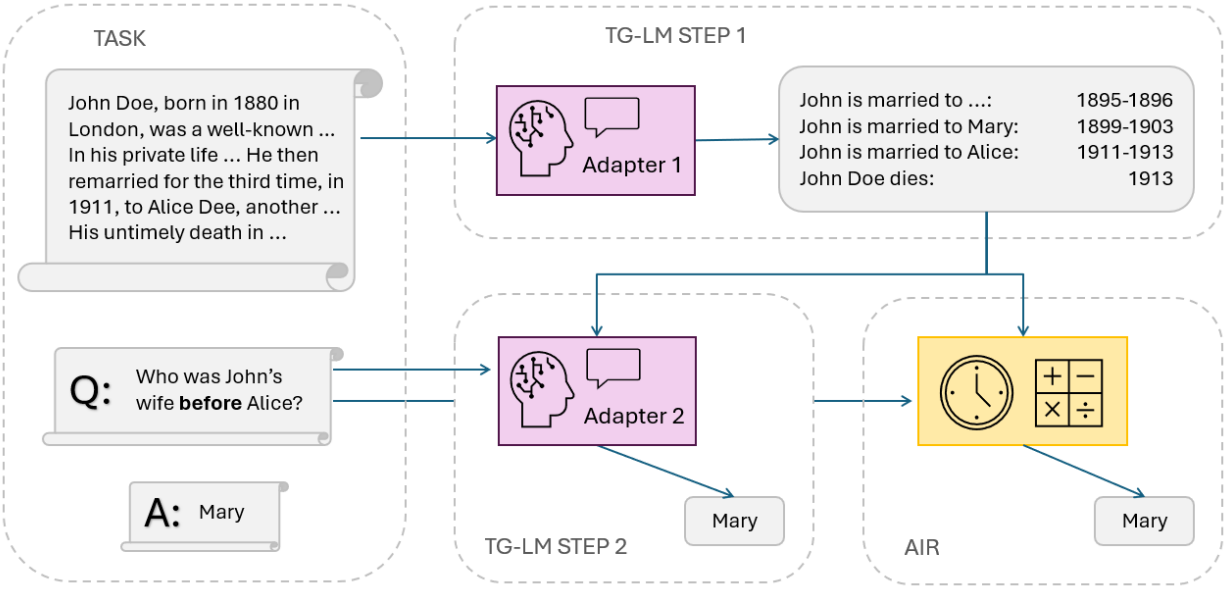

Figure 1: Left, the task formulation, a triplet of: narrative description of events; question requiring the understanding of time; and the correct answer. Top right, the first step of the 2-step pipeline of (Xiong et al., 2024): a fine-tuned LLM extracts the temporal graph from natural language. Bottom middle, the second step of the 2-step pipeline: a separate fine-tuned LLM answers questions based on structured text, rather than full narrative. Bottom right, our proposed alternative second step: a Python rule-based reasoner answers questions based on structured text and the rules of Allen's Interval Algebra.

3.1 Data

Since the aim is to extend previous work, we use the same datasets as the original TG-LLM paper (Xiong et al., 2024): TimeQA (Chen et al., 2021); TempReason (Tan et al., 2023); and the newly introduced TGQA (Xiong et al., 2024). These are all temporal question-answering datasets, intended for evaluating and / or fine-tuning LLMs, and thus each sample naturally consists of a minimum of: a short piece of narrative text; a question requiring the understanding of time / temporal logic; and the correct answer. However, since Xiong et al. (2024) propose to leverage the concept of *temporal graphs* to guide reasoning, they (and therefore we) also require access to the correct, structured timeline of events. Note, we use *temporal graphs* as a shorthand for the structured timeline of events following Xiong et al. (2024), although none of the datasets adhere to the typically assumed (subject, relation, object, start, end) syntax. Both the TempReason and the TGQA dataset contain the ground truth temporal graphs, owing to the construction of these benchmarks. For TimeQA, Xiong et al. (2024) separately extract the temporal graphs, employing a mix of rule-based and LLM-based processing. Following a semi-automatic verification step confirming their robustness, these are

then treated as the ground truth[2]. We evaluate performances on the *test* split of each. For context we briefly describe each dataset below.

TimeQA (Chen et al., 2021) draws on WikiData (Vrandečić & Krötzsch, 2014) for accurate time-sensitive facts and Wikipedia for their corresponding text. It is constructed in a semi-automated manner that combines data mining and crowd-sourced verification. Questions concern four temporal relations: 'in'; 'between'; 'before'; and 'after'. Importantly, the dataset supports two levels of difficulty, *easy* vs. *hard*, distinguished by the level of explicitness. Specifically, the question might pertain to a date or date range explicitly mentioned in the text, or it might require an answer that is implied by other dates, without being included *verbatim* in the story. This means that effectively the *easy* portion of TimeQA is more a test of temporal understanding than temporal reasoning.

TempReason (Tan et al., 2023) also leverages the knowledge base of WikiData (Vrandečić & Krötzsch, 2014), and similarly selects time-sensitive facts for generating the benchmark. However, as it is a more contemporary study aware of previous efforts, its approach departs from TimeQA and results in a somewhat different set of questions. Although there is some overlap in terms of the main subject of temporally-evolving facts, there is less than 1% overlap in terms of (subject, answer) pairs, indicating that the benchmarks are complementary. Importantly, the creation includes a convenient step of selecting sets of facts connected by the subject, which means the temporal graph is already provided with the data. This benchmark supports 3 variants of temporal reasoning questions[3], named *L1*, *L2* and *L3*. In contrast to TimeQA, these subtypes are not explicitly linked to difficulty, but rather to the type of task. Namely, the variants differ by asking about, respectively: a time-time relation; a time-event relation; and, an event-event relation. Following Xiong et al. (2024) we only use the *L2* and *L3* subtypes.

Finally, TGQA (Xiong et al., 2024) is a synthetic dataset created by sub-sampling the Yago11k knowledge graph and then generating corresponding narratives in natural language, using GPT-3.5. Random sub-graphs of up to 5 nodes are selected, and, to avoid potential issues of data memorisation in pre-trained LLMs, anonymised. To ensure alignment between the produced story and the source temporal graph, an additional semi-automated validation step is introduced. An LLM is queried for each individual temporal fact; should it fail to produce the correct response, the sample is then inspected manually. Questions are generated by populating templates from the ground-truth temporal graph. The graph-first rather than story-first construction of TGQA allows for a slightly wider variety of question types than either TimeQA or TempReason. Specifically, Xiong et al. (2024) distinguish 7 categories of queries: deciding which out of 2 events occurred first; deciding which out of several events occurred n-th; providing the duration of an event; providing the duration of gap between events; identifying the immediately preceding / following event; answering with the time of occurrence; deciding overlap; and deciding simultaneity of two events.

3.2 Rule-based Temporal Question-Answering

We develop a minimal rule-based question-answering code base in Python, leveraging the relations from Allen's Interval Algebra (AIA)(Allen, 1983). AIA formalises the commonsense temporal logic around the notion of intervals and their possible relations. Allen (1983) posits 13 such relations (see Table 1), which together with the transitivity table supports development of algorithms for reasoning about events in time. We don't implement the full language of Allen's algebra simply because we don't require it for our current purposes, though we see it as a fruitful direction for future work.

The core element in our processing pipeline is a simple Python *Interval* class [4] that implements AIA relations as operations that can be performed between the two instances of the class. Specifically, each instance requires start and end times to be provided on initialisation; this allows each of the AIA relations to be an invokable method returning Boolean truth values as to whether the relationship holds. For example, to understand whether event X has occurred before event Y, the dedicated method checks whether the end time of X was before the start time of Y (first row of Table 1).

[2]For convenience we download all datasets from a repository provided by Xiong et al. (2024), rather than from their respective original sources.

[3]TempReason also contains questions that do not test reasoning, but e.g. knowledge of temporal facts.

[4]Implementation adapted from the publicly available https://github.com/bartonip/pyintervals/tree/master

Table 1: The 13 relations of Allen's Interval Algebra, composed of identity and 6 asymmetric relations and their inverse relation.

Relation	Notation	Inverse	Illustration	Implementation
X before Y	$<$	$>$	XXX YYY	X.end < Y.start
X equal Y	$=$	$=$	XXX YYY	X.start == Y.start & X.end == Y.end
X meets Y	m	mi	XXX YYY	X.end == Y.start
X overlaps Y	o	oi	XXX YYY	X.start < Y.start & X.end > Y.start
X during Y	d	di	XXX YYYYY	X.start > Y.start & X.end < Y.end
X starts Y	s	si	XXX YYYYY	X.start == Y.start & X.end < Y.end
X finishes Y	f	fi	XXX YYYYY	X.start > Y.start & X.end == Y.end

To perform verifiable reasoning, the supporting code handles parsing of the plain-text temporal graphs into Python objects, as well as question-answering. First, each TG is encoded as an unordered collection of instances of the *Interval* class. Simple string parsing extracts the start and end times from the structured text, as well as the names / descriptions of the events as labels. Second, since different questions of temporal logic will require the evaluation of different AIA relations (or their combinations), separate sub-routines handle answering each possible question type. Heuristic rules then match questions to appropriate function streams. For an illustrative example, see Figure 2: since the keyword 'before' occurs in the text of the query, the subroutine that checks for either 'before' or 'meets' will be selected. All intervals will be then compared to the target 'Alice' one, and those evaluating to *True* selected. From these, the one chronologically last will be selected. We note that we make a number of simplifying assumptions in developing our code base; e.g. in this example we ignore the possibility that events may overlap. We refer to our rule-based TQA solution as Allen's Interval Reasoner (AIR).

We note that the code base does not provide a universal temporal reasoner since neither of the datasets provide instances of true knowledge graphs, but rather forms of structured text. Thus, we merely develop a set of utilities that cover the question types and the temporal graph formats present in the three benchmarks (TimeQA, TempReason, TGQA); many more are possible. Furthermore, the set of utilities is developed on the **ground truth** temporal graphs (TG-true), a setting where parsing and QA are straightforward due to their well-structured nature. However, with the aim being to apply the AIR system on *inferred* graphs (TG-pred), we need to consider the fact that LLM-extracted TGs may contain formatting errors. Although, following Xiong et al. (2024) we will be using an instruction to 'Return json only', it is well known that LLMs may over-generate, repeat tokens, etc. We therefore also develop utilities to correct all such mistakes. Admittedly, this is not a general or easily scalable solution for a NeSy system; the process requires substantial time, and only works for the model and task context of the study. The present study was meant as a proof-of-concept of an AIA-LLM system, although it also (inadvertently) turned into a cautionary tale of deceptive

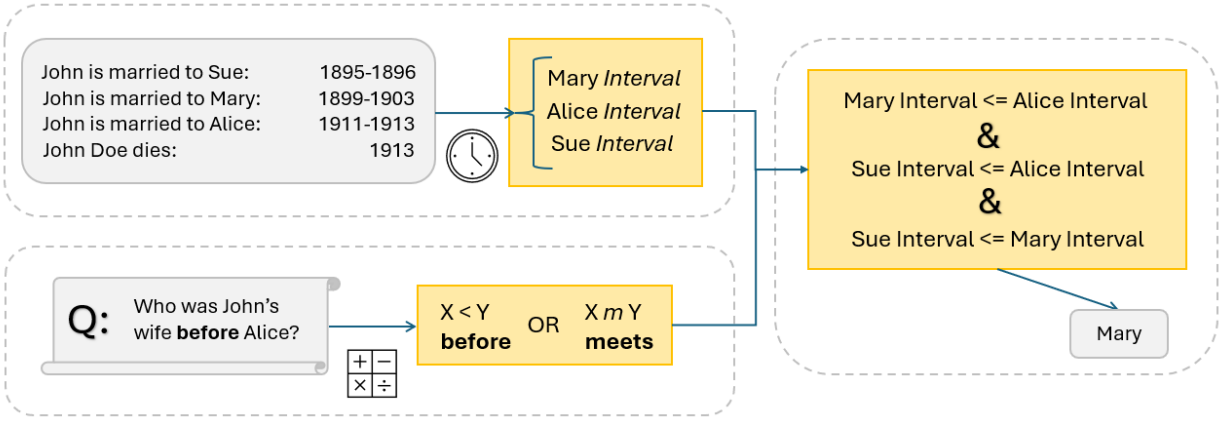

Figure 2: Top left: parsing the structured text to obtain an unordered collection of objects of the *Interval* class. Bottom left: matching question types to appropriate AIA relation-based question-answering strategy. Right: query is answered by processing the collection of intervals with the selected utility.

effectiveness of LLMs when evaluated on performance only. We leave a more general extension to future work, as outlined in the Discussion.

3.3 Inferred temporal graphs

Since the datasets provided by Xiong et al. (2024) do not contain the *inferred* temporal graphs, only the ground truth ones, we need to reproduce a portion of the study. Specifically, we need to train appropriate low rank adapters (LoRA) (Hu et al., 2022) for temporal graph extraction, and then run these in inference mode on the held-out *test* data. In their original approach, Xiong et al. (2024) train several adapters, to evaluate transfer from TGQA vs. training on each benchmark separately. Since we are primarily interested in a real-world setting, where curated training data is not available, we choose to train on the synthetically-generated TGQA. We use the code provided (Xiong et al., 2024) and similarly use the open-source Llama 2 13B model (Touvron et al., 2023) due to limited resources. We still have separate adapters for each dataset, to cater to their varying TG formats, as each benchmark specifies the graph in a different manner (e.g. providing dates first vs. last in the sentence). Specifically, we use the same input stories and true temporal graphs for supervised fine tuning, only vary the formatting of output. This is primarily to align with Xiong et al. (2024), who required the first adapter to output data in the format expected by the second (reasoning) adapter. To produce inferred temporal graphs, we apply the fine-tuned adapters, following Xiong et al. (2024) in all settings to the best of our knowledge. That means we include an in-context example at test time and use the 'easy' inference mode.

3.4 Evaluation

To evaluate AIR-LLM responses we use exact match (EM) only, as our rule-based reasoner returns nodes of the TG rather than free text (unlike Xiong et al. (2024), who additionally employ token-level F1 score and perplexity-based accuracy). We relax, however, the notion of EM, since the QA datasets (being semi-automatically generated) may ignore a degree of ambiguity in the data while providing a single 'correct' answer. In contrast, our rule-based mechanism returns all semantically / logically valid responses. To illustrate, and continuing with the example of 'What happened before..?', several events might be preceding the target and might furthermore end at the same time, making the answer ambiguous without further clarifications (e.g. taking whichever started later vs whichever lasted longer). To provision for such scenarios, we simply require that the correct answer is included in the AIR-identified set. We evaluate our rule-based reasoner on the ground truth temporal graphs and reach near-perfect performance, as expected.

We also develop a simple scoring function to evaluate the inferred temporal graphs against the ground-truth ones (TG-pred-score). This serves as a sanity check for the evaluation of results on the *inferred* temporal graphs. Our question answering AIR system respects temporal logic by design, and the results from ground truth temporal graphs assure us to the correctness of parsing and routing utilities. However, with the LLM outperforming our symbolic solver, we want to additionally verify that AIR is correct to the degree the graphs themselves are. Further, since we suspect short-cutting, we can use the scoring function as a means of testing our hypothesis. The scoring function grants 0.5 point for each correct start and end of a temporal interval in the graph, and averages the total over the length of the TG. For example, if the timeline consists of 4 events and the LLM extracts the timing of only 2 correctly, the score will be 0.5; it will also be 0.5 if all events' start times are correct, but end times incorrect. We also have a variant that penalises over-generation, since additional nodes in a temporal graph may impact on the correctness of answers. For brevity we omit that from discussion, as it does not impact materially on the results.

3.5 Additional experiments

Observing surprisingly poor outcomes against the reported LLM-based reasoning, and wanting to dissect the reasoning discrepancies in detail, we also decide to re-run the second step of the TG-LLM pipeline (Xiong et al., 2024). We again fine-tune a LoRA adapter (Hu et al., 2022) for Llama 13B (Touvron et al., 2023) on the train split of TGQA, with the ground truth temporal graphs as input, and reasoning answers as output. We do not use CoT bootstrapping nor graph augmentation for computational reasons, but otherwise reproduce Xiong et al. (2024); that is, we use only the 'base' / 'vanilla' Chain-of-Thought (Wei et al., 2022b) and provide additional information in the prompt. With adapter trained, we run inference on the *test* split of TGQA only, again due to computational and researcher time considerations. We run the adapter both on the *inferred*, as well as *ground truth* temporal graphs, to illustrate the benefits of a symbolic approach.

4 Results

4.1 Reasoning quality is limited to information quality

We develop the rule-based utilities for temporal question answering (AIR) around the notion of Allen's Interval Algebra. As mentioned already, we limit ourselves to the question types and TG formats present across TimeQA, TempReason and TGQA datasets. We achieve ≥ 0.99 accuracy when applying AIR on the ground truth TG's, as shown in Figure 3 (denoted as AIR (TG true)). However, when applied to the *inferred* temporal graphs, our reasoner drops sharply in performance. The only dataset where scores remain above 0.5 is TGQA; the one used for fine-tuning the story-to-graph adapter. On the remaining TempReason and TimeQA benchmarks the performance is very poor, ranging between 0.2 and 0.43 (AIR-LLM in Figure 3). Importantly, and to our surprise, these scores are all between 0.1 and as much as 0.25 **below** the reported, purely neural solutions (we re-report these as TG-LLM (full) and GPT-4 in Figure 3, from the original Xiong et al. (2024)). When compared to the best scores from the original benchmark papers (Chen et al., 2021; Tan et al., 2023), only on the more difficult TempReason AIR-LLM is outperforming the baseline. Seeing as we achieve near-perfect performance on true TGs, we therefore suspect that the issue lies with the *inferred* temporal graphs.

Since the inferred temporal graphs are at risk of formatting issues, we develop additional utilities to correct such minor mistakes. Although it is difficult to assure correctness of our approach without manual inspection of each sample, we discover in the process that a substantial proportion of TimeQA and TempReason graphs are malformed beyond repair, as shown in Figure 4 A. We add heuristic checks for such cases and leverage our knowledge of true TGs to ensure that the parsing of LLM answers is not at fault. Finally, to isolate TG quality from reasoning quality, we also report our scoring of temporal graphs against the ground truth (denoted as TG-pred-score in Fig. 3). These broadly align with AIR-LLM accuracy, although tend to be slightly lower. That is primarily because each story tends to have more than a single question associated with it. It is possible to answer a number of such questions correctly, even if most of a given graph is incorrect, so long as the question pertains to the correct portion of the TG or the error does not impact on the temporal

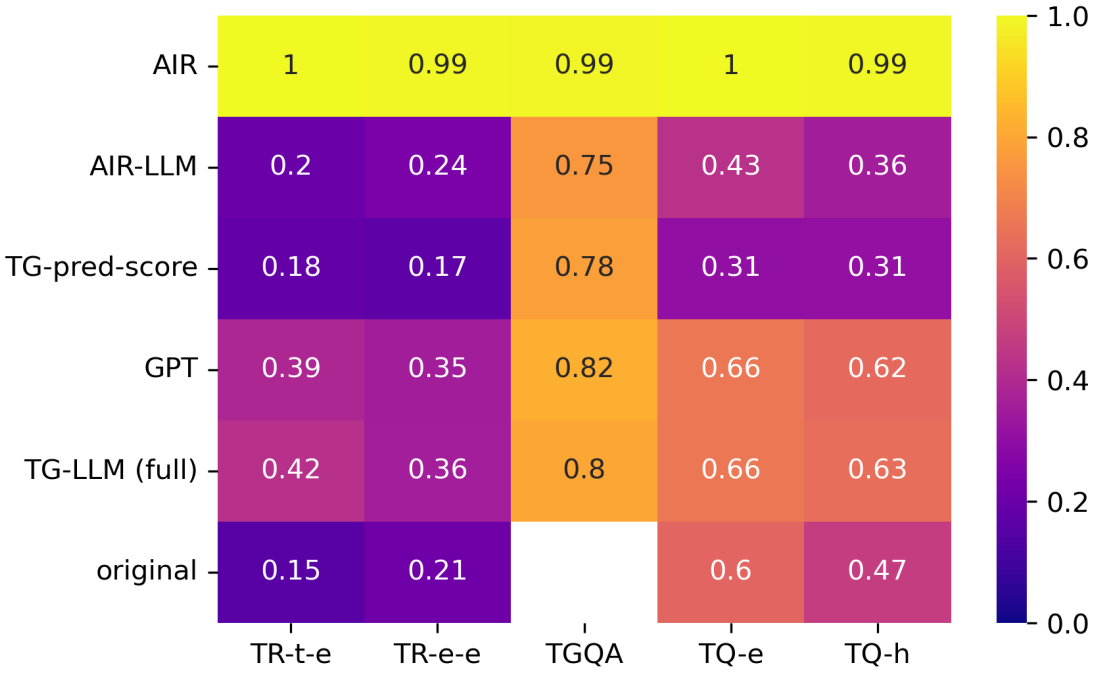

Figure 3: Temporal question answering and graph quality results. Rows: 'AIR' and 'AIR-LLM' refer to our solution on TG-true and TG-pred respectively; 'GPT-4', 'TG-LLM (full)' are re-reported from Xiong et al. (2024); 'original' from Tan et al. (2023) and Chen et al. (2021) for TempReason and TimeQA, respectively. Columns: 'TR-t-e' stands for TempReason L2, a.k.a. time-event relation; 'TR-e-e' refers to TempReason L3, a.k.a. event-event relations; 'TGQA' is the synthetic dataset of Xiong et al. (2024); 'TQ-e' and 'TQ-h' are the easy and hard subset of TimeQA.

function queried. For example, if all the intervals in a graph are offset by a set amount of time, neither their chronological order nor duration would be affected.

Thus, we are able to answer practically all questions when provided true graphs, and we also know the inferred graphs to be lacking. We are therefore led to conclude that the performance gap between our AIR-LLM and the reference TG-LLM of Xiong et al. (2024) may at least in part (if not in entirety) be due to short-cutting. To verify our suspicions, however, we require access to the LLM reasoning answers. Hence, we perform an additional experiment and re-run also the second step of the TG-LLM pipeline.

4.2 Unreasonably good LLM reasoning

We select the TGQA dataset for the in-depth case study and re-run the second portion of the TG-LLM pipeline (Xiong et al., 2024) without CoT bootstraping or TG augmentation. That is, we fine-tune a *reasoning* adapter on true temporal graphs (from the *train* split), and then apply it to the *test* split TGs *inferred* by the first adapter. For completeness, we also run the reasoning adapter on true graphs from the *test* split, to isolate TG-LLMs reasoning performance from graph extraction performance.

Two things are apparent from the results shown in Figure 4 B. First, without CoT bootstrapping or TG augmentation of the main results of Xiong et al. (2024), the performance of the neural reasoner no longer surpasses our symbolic AIA-based solution, dropping from 0.82 to 0.62. Second, even when provided the **ground truth** temporal graphs, LLMs performance is lower than the reported TG-LLM on **inferred** TGs. Both of these observations are in line with our short-cutting hypothesis. The fact that removing output / data augmentation methods degrades performance is no surprise: LLMs are not *reasoning* towards the

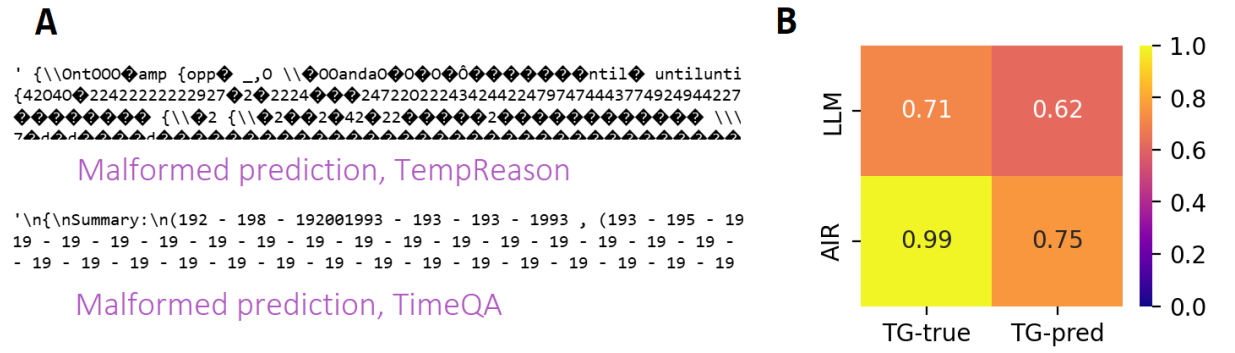

Figure 4: A. Two examples of mal-formed predictions, from TimeQA and TempReason benchmarks; since the adapter was trained on TGQA we expect the most glaring mistakes to be made on the other benchmarks. B. Dissecting the rule-based vs. LLM-based reasoning on true vs. predicted temporal graphs.

answer, but rather *predicting*[5], hence their responses are liable to change with the data or manner of posing the question (Schulhoff et al., 2024). The second observation drives the point home: the input data *has all the information one needs to reason to the answer*; the only way one could get any of these wrong (beyond the odd ambiguity of the question), much less 30%, is if one were guessing.

There is another interesting observation to be made: the symbolic reasoner improves on the neural one by 30% on true graphs, but only 13% on inferred ones. This could, of course, be due to chance or some form of nearing performance plateau by the LLM, though we rather suspect it to be another symptom of short-cutting. We thus investigate finally whether the 0.62 performance is *genuine*. Specifically, we check whether any of the correct LLM answers coincide with *incorrect* AIR-LLM answers. We proceed to find that for approx. 10% of the dataset the LLM short-cuts: it answers correctly, despite the inferred temporal graph being faulty and implying a wrong answer.

5 Discussion

We set out to demonstrate the benefit of employing a hybrid NeSy approach for the task of temporal reasoning in natural language, an area where the complementary strengths of neural and symbolic approaches are particularly salient. Since the logic of time is somewhat more complex than e.g. commonsense reasoning, and specialised data less abundant, it is an area still ripe for improvements for LLMs. At the same time, temporal reasoning in a verifiable manner is relatively straightforward with appropriately structured input. We develop such a rule-based reasoner for time-sensitive question-answering as a natural complement of the TG-LLM framework (Xiong et al., 2024). Specifically, we intend to employ an LLM as a natural language parser only, leaving reasoning to the rule-based solution. We demonstrate reliability on structured input, but observe surprisingly disappointing results on LLM-inferred structures, with performance below that of fully-neural (Xiong et al., 2024). The discrepancy leads us to further experiments and examination of error attribution in both pipelines, since the surprising dominance of LLMs over a verifiable reasoner resembles the known pattern of short-cutting. We find that without the modifications of CoT bootstrapping and augmentation, TG-LLM performs worse than our reasoner. Further, TG-LLM indeed shortcuts to the correct answer in 10% of the examined cases. The second point in particular is worth highlighting, and it resonates with a range of recent works that challenge the misleading nomenclature of 'reasoning' and 'thinking' that surrounds LLMs (Bao et al., 2024; Pfister & Jud, 2025; Stechly et al., 2025; Lee et al., 2024; Valmeekam et al., 2022; Shojaee et al., 2025). We highlight here specifically Bao et al. (2024) and Stechly et al. (2025), who both investigate the relationship of chain-of-thought tokens to the final answer. Although we are rather concerned with prompt content than steps of 'reasoning', we feel the overarching take-away is similar: final answer depends only *statistically* on the input and intermediate information, unlike in

[5]In other words, providing 'educated guesses'.

verifiable reasoning. We argue that the recent successes of these models in purported 'reasoning' stems from the deceptive result-oriented mindset, where correct answers are equated with evidence of cognition. As the structural limitations of LLMs are becoming increasingly widely acknowledged, we maintain that rule-based and symbolic computation is a valuable complement that can help redress these shortcomings.

5.1 Limitations and future work

Three limitations of the present work, as well as availability of convenient benchmark of Fatemi et al. (2024) lead us to scope a natural extension of the study. A first limitation is that our attempted proof-of-concept AIR-LLM offers no performance improvement over the purely neural solution. Secondly, manual implementation of question parsing and other utilities for specific benchmarks is time-costly. And finally, as a result of manual, benchmark-focused development, our AIR reasoner is not a universal one, thus won't be immediately transferrable to other problems / datasets.

The first of these limitations is straightforward to ameliorate; we simply need high-quality temporal graphs. Large Language Models are far better as parsers than reasoners; in fact, as already demonstrated by Xiong et al. (2024) on the TimeQA dataset, extracting reliable temporal graphs is possible, albeit with a little more effort and control than fine-tuning on a synthetic dataset. To address the second and third limitations, we intend to examine the possibility of leveraging PAL, in concert with synthetic data exactly like that of Fatemi et al. (2024). Specifically, LLMs would be tasked with writing executable Python programs to parse and answer questions using AIA logic and the Interval Python class, in a similar vein to Pan et al. (2023). Preliminary tests carried out suggest it to be a promising avenue. We expect that when data admits controllable graph complexity and length, a Neuro-symbolic approach will provide clear benefits over LLMs.

References

James F Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11): 832–843, 1983.

Guangsheng Bao, Hongbo Zhang, Linyi Yang, Cunxiang Wang, and Yue Zhang. LLMs with Chain-of-Thought Are Non-Causal Reasoners. *CoRR*, abs/2402.16048, 2024. URL `https://doi.org/10.48550/arXiv.2402.16048`.

Tarek R Besold, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luis C Lamb, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, et al. Neural-symbolic learning and reasoning: A survey and interpretation 1. In *Neuro-symbolic artificial intelligence: The state of the art*, pp. 1–51. IOS press, 2021.

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.

Wenhu Chen, Xinyi Wang, and William Yang Wang. A Dataset for Answering Time-Sensitive Questions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL `https://openreview.net/forum?id=9-LSfSU74n-`.

Francois Chollet. OpenAI breakthrough high score on ARC-AGI-Pub, 2024. URL `https://arcprize.org/blog/oai-o3-pub-breakthrough`.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. GTBench: Uncovering the Strategic Reasoning Capabilities of LLMs via Game-Theoretic Evaluations. *Advances in Neural Information Processing Systems*, 37:28219–28253, 2024.

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.

Eleonora Giunchiglia, Fergus Imrie, Mihaela van der Schaar, and Thomas Lukasiewicz. Machine learning with requirements: A manifesto. *Neurosymbolic Artificial Intelligence*, 1:NAI–240767, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-Rank Adaptation of Large Language Models. *ICLR*, 1(2):3, 2022.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents. In *International conference on machine learning*, pp. 9118–9147. PMLR, 2022.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of Experts. *arXiv preprint arXiv:2401.04088*, 2024.

Seungpil Lee, Woochang Sim, Donghyeon Shin, Wongyu Seo, Jiwon Park, Seokki Lee, Sanha Hwang, Sejin Kim, and Sundong Kim. Reasoning Abilities of Large Language Models: In-depth Analysis on the Abstraction and Reasoning Corpus. *ACM Transactions on Intelligent Systems and Technology*, 2024.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards Fully Automated Open-ended Scientific Discovery. *arXiv preprint arXiv:2408.06292*, 2024.

W. McCune. Prover9 and mace4. `http://www.cs.unm.edu/~mccune/prover9/`, 2005–2010.

Chinmay Mittal, Krishna Kartik, Parag Singla, et al. FCoReBench: Can Large Language Models Solve Challenging First-Order Combinatorial Reasoning Problems? *arXiv preprint arXiv:2402.02611*, 2024.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3806–3824, 2023.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. Towards Systematic Evaluation of Logical Reasoning Ability of Large Language Models. *CoRR*, abs/2404.15522, 2024. URL `https://doi.org/10.48550/arXiv.2404.15522`.

Rolf Pfister and Hansueli Jud. Understanding and Benchmarking Artificial Intelligence: OpenAI's o3 Is Not AGI. *arXiv preprint arXiv:2501.07458*, 2025.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are Emergent Abilities of Large Language Models a Mirage? *Advances in Neural Information Processing Systems*, 36, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models can Teach Themselves to Use Tools. *Advances in Neural Information Processing Systems*, 36, 2024.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. The Prompt Report: A Systematic Survey of Prompting Techniques. *arXiv preprint arXiv:2406.06608*, 2024.

Terrence J Sejnowski. Large Language Models and the Reverse Turing Test. *Neural computation*, 35(3): 309–342, 2023.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity. *arXiv preprint arXiv:2506.06941*, 2025.

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. *arXiv preprint arXiv:2206.04615*, 2022.

Kaya Stechly, Karthik Valmeekam, Atharva Gundawar, Vardhan Palod, and Subbarao Kambhampati. Beyond Semantics: The Unreasonable Effectiveness of Reasonless Intermediate Tokens. *arXiv preprint arXiv:2505.13775*, 2025.

Zhaochen Su, Jun Zhang, Tong Zhu, Xiaoye Qu, Juntao Li, Min Zhang, and Yu Cheng. Timo: Towards Better Temporal Reasoning for Language Models. *arXiv preprint arXiv:2406.14192*, 2024.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. Towards Benchmarking and Improving the Temporal Reasoning Capability of Large Language Models. *arXiv preprint arXiv:2306.08952*, 2023.

Weizhi Tang and Vaishak Belle. ToM-LM: Delegating Theory of Mind Reasoning to External Symbolic Executors in Large Language Models. In *International Conference on Neural-Symbolic Learning and Reasoning*, pp. 245–257. Springer, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023.

Alan Mathison Turing. Mind. *Mind*, 59(236):433–460, 1950.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large Language Models Still Can't Plan (A Benchmark for LLMs on Planning and Reasoning about Change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

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

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*, 2022a. ISSN 2835-8856. URL https://openreview.net/forum?id=yzkSU5zdwD. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in neural information processing systems*, 35:24824–24837, 2022b.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. Large language models can learn temporal reasoning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10452–10470, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.563. URL https://aclanthology.org/2024.acl-long.563/.