Benchmarking Systematic Relational Reasoning with Large Language and Reasoning Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have been found to struggle with systematic reasoning. Even on tasks where they appear to perform well, their performance often depends on shortcuts, rather than on genuine reasoning abilities, leading them to collapse on out-of-distribution examples. Post-training strategies based on reinforcement learning and chain-of-thought prompting have recently been hailed as a step change. However, little is still known about the potential of the resulting "Large Reasoning Models" (LRMs) beyond problem solving in mathematics and programming, where finding genuine out-of-distribution problems can be difficult. In this paper, we focus on tasks that require systematic reasoning about relational compositions, especially for qualitative spatial and temporal reasoning. These tasks allow us to control the difficulty of problem instances, and measure in a precise way to what extent models can generalise. We find that that the considered LLMs and LRMs overall perform poorly overall, albeit better than random chance.

1 Introduction

Large Language Models (LLMs) have shown a remarkable generalization abilities, being able to learn from in-context demonstrations, and to generalize to unseen tasks in multi-task settings (Radford et al., 2019; Brown et al., 2020; Bubeck et al., 2023), with abilities in mathematics and programming that appear to go beyond the level of high-school students (Guo et al., 2024; Jimenez et al., 2024; OpenAI et al., 2025). Moreover, recent advances in post-training based on reinforcement learning have unlocked a further axis along which the ability of LLMs can be improved, for easily verifiable analytical problems (such as mathematics and programming) (Guo et al., 2025; Shao et al., 2024). The resulting models, called Large Reasoning Models (LRMs), are then encouraged to leverage chains-of-thought (CoT) or thinking tokens (Wei et al., 2022) to search though a solution space, which provably increases the complexity of problems that can be tackled (Feng et al., 2023), compared to standard LLM prompting.

Yet, a competing narrative is that current LLMs are not, in fact, general-purpose reasoners and rather rely on shallow pattern matching (Dziri et al., 2023; McCoy et al., 2024; Nguyen, 2024) and heuristics (Nikankin et al., 2024). There are recurring issues, even with the latest LLMs and LRMs, such as memorization of training data (Zhang et al., 2024b), the reversal curse (Berglund et al., 2024) and an over-reliance on co-occurrence statistics (Kang and Choi, 2023). This line of argument is further bolstered by the risk that popular static benchmarks, such as GSM8k and MMLU, may have been included in training corpora (Zhang et al., 2024b; Oren et al., 2024). The potential for dataset contamination is increasingly problematic, given the scaling laws for memorization (Carlini et al., 2023), and may explain why despite displaying erudite behaviour, current models still fail at seemingly basic tasks that are trivial for ordinary humans.

In this paper, we highlight the importance of using benchmarks that require Systematic Generalization (SG) for reliably evaluating the reasoning capabilities of LLMs and LRMs. SG is the ability of a model to solve test instances by composing knowledge that was learned from multiple training instances (Hupkes et al., 2020), where the test instances are systematically made larger than the training instances. This ensures that the test instances are new, while at the same time guaranteeing that the model has access to all the knowledge that is required for solving them. Composing atomic units into larger pieces for constructing a solution to an arbitrarily large problem is an essential ingredient for machines and humans to generalize from a limited amount of data (Lake et al., 2017). We specifically advocate the use of synthetic benchmarks, where the difficulty of problem instances can be controlled along different dimensions.

For the analysis in this paper, we leverage the Spatial Temporal and Reasoning (STaR) benchmark (Khalid and Schockaert, 2025). Its problem instances have a combinatorial structure, which makes it straightforward to generate large numbers of previously unseen cases, and in particular avoid issues of dataset contamination. The StaR benchmark has proven challenging for state-of-the-art neuro-symbolic reasoning methods (Minervini et al., 2020; Cheng et al., 2023; Lu et al., 2022), but has not yet been used for evaluating LLMs and LRMs. It poses an interesting challenge, because the disjunctive nature of the rules that govern the temporal and spatial reasoning problems means that the answer cannot be obtained by a single derivation (i.e. a single chain-of-thought) and essentially requires simulating the algebraic closure algorithm (Renz and Ligozat, 2005). Note, however, that these problems are computationally tractable (i.e. they can be solved in polynomial time) and should thus, in principle, be within the reach of LRMs. This is fundamentally different from evaluating LRMs on PSPACE-hard planning problems, where at best strong heuristic approximations can be expected (Valmeekam et al., 2024).

Our main finding is that many popular LLMs and LRMs struggle on STaR but do reason beyond random chance. We analyze the effects of increasing model size, fine-tuning and CoT test-time compute on reasoning performance.

2 Related Work

Spatial Reasoning The spatial reasoning capabilities of LLMs have already been studied from various angles. For instance, SPARTQA (Mirzaee et al., 2021), StepGame (Shi et al., 2022) and RoomSpace (Li et al., 2024b) are question answering datasets which require the model to infer the relative position of two objects based on a description of their position relative to other objects. Similarly, Yamada et al. (2024) test the ability of LLMs to follow natural language descriptions of trajectories in grid-like environments. STBench (Li et al., 2024c) evaluates LLMs on a suite of 13 tasks, most of which involve some form of spatial reasoning. However, rather than focusing on qualitative reasoning, these tasks essentially involve some form of geometric computation, e.g. determining if a given point belongs to some region or determining the regions through which a given trajectory passes. Cohn and Blackwell (2024a) evaluate whether LLMs can infer the composition of two RCC-8 relations, when given a description of their meaning, while Cohn and Blackwell (2024b) evaluate their commonsense understanding of cardinal directions (e.g. you are walking along the East shore of a lake, in which direction is the lake?). Wang et al. (2024) consider spatial reasoning in a multi-modal setting.

Several authors have also tried to improve LLM spatial reasoning. Li et al. (2024a) study the effectiveness of chain-of-thought (Wei et al., 2022) and tree-of-thoughts (Yao et al., 2023) prompting. They also show the effectiveness of using the LLM for semantic parsing and leaving the reasoning aspects themselves to a symbolic solver. A similar strategy is also pursued by Zhang et al. (2024a), who construct a graph representation of the input, and then either call a symbolic solver or rely on the LLM itself to reason about the extracted graph. Wu et al. (2024) improve chain-of-thought methods for spatial reasoning, by generating a visualization after each inference step. In multimodal settings, pre-training the model on synthetic data is a common strategy. Interestingly, Tang et al. (2024) found that by training the model on basic (visual) spatial reasoning capabilities (direction comprehension, distance estimation and localization), the model also performs better on out-of-distribution composite spatial reasoning tasks, such as finding the shortest path between two objects.

Systematic Generalization There is a plethora of work on measuring systematic generalization beyond relational reasoning, including SCAN (Lake and Baroni, 2018) for RNNs (Schuster and Paliwal, 1997), addition (Nye et al., 2021) and LEGO (Zhang et al., 2023), for transformers trained from scratch (Vaswani, 2017). This line of work clearly suggests that transformers struggle with SG. The most popular benchmark for systematic generalization in the context of relational reasoning is CLUTRR Sinha et al. (2019), which involves predicting family relations. Zhu et al. (2024) evaluated LLMs on this benchmark, showing that even modern LLMs with CoT prompting struggle with this task. The problems we consider in this paper are more challenging than those in CLUTRR, due to the need for combining multiple reasoning paths.

Rule-based Reasoning with LLMs Sun et al. (2024) studied the ability of LLMs to apply a given rule, when provided as part of the prompt. In con-

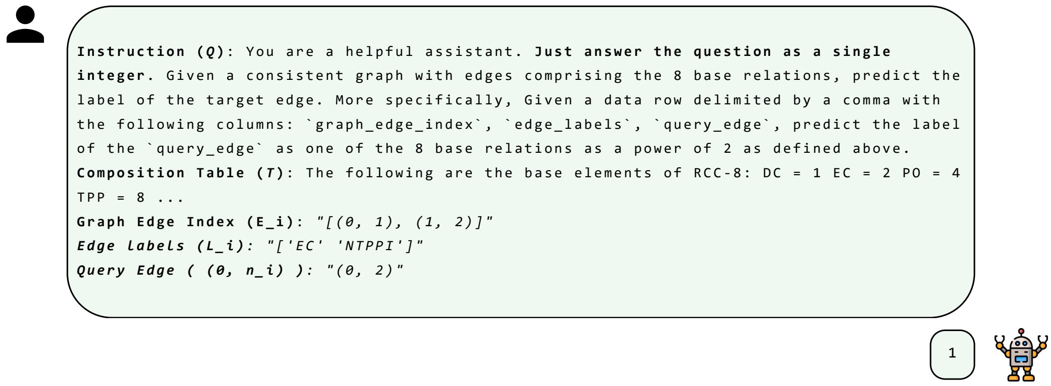


Figure 1: An illustration of the input representation to the language model which is prompted to respond (modulo thinking tokens) with a single label for the query edge.

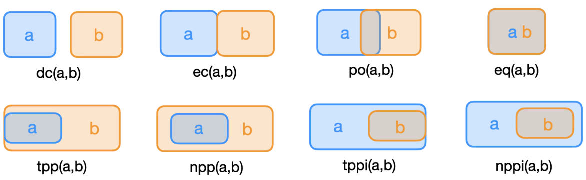


Figure 2: Illustration of the RCC-8 relations.

trast to our experiments in this paper, they only evaluated the application of a single rule, some of which were complex (e.g. encoding the composition of a path of several relations). They found chain-of-thought prompting to be largely ineffective, which appears to be related to the fact that multi-hop reasoning was not required for their benchmark. They also found evidence that models rely on prior knowledge about the considered domains (e.g. the composition of family relations).

3 The STaR Problem

In each problem instance of STaR, we are given a set of facts $\mathcal{F}$, referring to a set of binary relations $\mathcal{R}$ and a set of entities $\mathcal{E}$. The set of relations is fixed across problem instances, but the entities are not. Each of the facts is an *atom* of the form $r(a, b)$, with $r \in \mathcal{R}$ and $a, b \in \mathcal{E}$. The problems we consider essentially require models to learn a set of rules $\mathcal{K}$, which they can then use to decide whether a given atom $r(a, b)$ can be inferred from the set of facts $\mathcal{F}$. To be successful, models must be capable of composing the learned rules in a systematic way. In particular, most problem instances require multiple rule applications to be chained, and the number of such inference steps may be larger for test examples than for training examples.

Disjunctive Rules Most reasoning benchmarks focus on Horn rules of the following form ($k \geq 3$):

$$r(X_1, X_k) \leftarrow \bigwedge_{i=1}^{k-1} r_i(X_i, X_{i+1}) \qquad (1)$$

where X_i are entity variables. Given a set $\mathcal{K}$ of such rules, the main reasoning task of interest is typically to decide whether some hypothesis $r_\ell(e, f)$ can be inferred from a set of facts $\mathcal{F}$ using the rules in $\mathcal{K}$. This can be decided by repeatedly selecting facts $r_1(e_1, e_2), ..., r_{k-1}(e_{k-1}, e_k)$ that match the body of a rule of the form (1) in $\mathcal{F}$ and adding the conclusion $r(e_1, e_k)$ of that rule to $\mathcal{F}$. This iterative derivation of facts is well-aligned with the style of reasoning that is enabled by chain-of-thought prompting, which can partially explain the success of such strategies for tasks that require simple logical reasoning. However, in many domains, Horn rules are not sufficient for capturing the required knowledge. A more general approach is to focus on disjunctive rules of the following form:

$$\bigvee_{i=1}^{m} s_l(X_1, X_k s) \leftarrow \bigwedge_{i=1}^{k-1} r_i(X_i, X_{i+1}) \qquad (2)$$

Given such a rule and the facts $r_1(e_1, e_2), ..., r_{k-1}(e_{k-1}, e_k)$, then all we can infer is that one of $s_1(e_1, e_k), ..., s_m(e_1, e_k)$ must be true. When reasoning with disjunctive rules, we are typically also given a set of constraints, such as:

$$\bot \leftarrow r_1(X,Y) \wedge r_2(X,Y)$$

encoding that at most one of the facts $r_1(e,f), r_2(e,f)$ can be true for any entities e, f. Reasoning with disjunctive rules is provably more expressive, but computationally also more expensive: while reasoning with Horn rules is possible in polynomial time, reasoning with disjunctive rules and constraints is an NP-complete problem. However, there are important special cases where reasoning with disjunctive rules is still possible in polynomial time. This is the case, in particular, for many of the calculi that have been proposed for qualitative reasoning about time and space, such as the Interval Algebra (IA (Allen, 1983)) and the Region Connection Calculus (RCC8 (Randell et al., 1992)).[1]

StaR Benchmark STaR (Khalid and Schockaert, 2025) consists of spatial and temporal reasoning problems. The spatial reasoning problems involve reasoning in RCC-8 (Randell et al., 1992). This calculus is defined using 8 relations, illustrated in Fig. 2. The entities in this case represent spatial regions. For instance, the fact $\mathsf{ec}(a,b)$ specifies that the region a is adjacent (i.e. Externally Connected) to the region b. Reasoning in RCC-8 is based on two types of knowledge. First, we have the knowledge that the relations are Jointly Exhaustive and Pairwise Disjoint (JEPD), meaning that there is exactly one of the eight relations that holds between any two regions. Second, we have knowledge about the composition of the eight relations. For instance, knowing that $\mathsf{ec}(a,b)$ and $\mathsf{po}(b,c)$ hold, the relations that may hold between a and c are dc, ec, po, tpp and ntpp. This knowledge can be encoded using a disjunctive rule. It is typically summarized in a so-called composition table, which encodes the compositions of all relations in a compact format. The temporal instances in StaR involve reasoning in IA (Allen, 1983). The overall structure of these reasoning problems is similar as in RCC-8, but here there is a set of 13 JEPD relations. The entities in this case represent time intervals, and we have relations such as $\mathsf{m}(e,f)$, encoding that the end point of e coincides with the starting point of f.

[1]When the $\mathcal{F}$ is allowed to contain disjunctions of facts, then reasoning with these calculi is NP-complete. However, since we only focus on the case where $\mathcal{F}$ is a set of facts, reasoning for our purposes is tractable in these calculi.

	Model	Param.	Quantization			Reasoning
			A	B	C	
Small	Qwen-2.5	7B	×	×	✓	N/A
	Qwen-2.5 **(R)**	7B	×	×	✓	✓
	Llama-3	8B	×	×	✓	N/A
	Gemma-2	9B	×	×	✓	N/A
Medium	Phi-4	14B	×	×	✓	N/A
	Qwen-2.5	14B	×	×	✓	N/A
	Qwen-2.5 **(R)**	14B	×	×	✓	✓
	Gemma-2	27B	×	×	✓	N/A
Large	Llama-3.3	70B	✓	✓	N/A	N/A
	Qwen-2.5	72B	✓	✓	N/A	N/A
	o3-mini	?	N/A	N/A	N/A	✓

Table 1: Model configurations for experimental settings in 4. All the quantizations are four-bit. **(R)** denotes the R1 distilled models (Guo et al., 2024).

Each problem instance is formulated as a directed labelled graph $\mathcal{G}$, where the vertices represent entities and the edges are labelled with a relation from $\mathcal{R}$, where $\mathcal{R}$ is either the set of RCC-8 relations or the set of IA relations. The goal is to infer the relation between two designated entities: a head entity h and a tail entity t. The problem instances are constructed such that there is a unique relation that can be inferred. To find this relation, however, information from multiple paths between h and t may need to be combined. Each of these paths makes it possible to infer a conclusion of the form $r_1(h,t) \vee ... \vee r_m(h,t)$. In other words, each path allows us to eliminate certain relationships as candidate answers, but we may need to combine several paths to eliminate all-but-one of the relations and thus obtain the answer. The dataset is constructed with two levers of complexity: b, the number of simple paths between the head and tail entity, and k, the length of each simple path. In accordance with the focus on SG, the training data is comprised of relatively small problem instances, with $k \in \{2,3,4\}$ and $b \in \{1,2,3\}$. The test data contains instances with $k \in \{2,\ldots,10\}$ and $b \in \{1,2,3,4\}$.

4 Experimental Setup

Input representation In principle, the only contextual information needed to solve an instance of STaR is the composition table. Khalid and Schockaert (2025) considered to what extent neuro-symbolic models were able to learn (and then systematically apply) this composition table from the training data provided. Here, we focus on a simpler setting, where we provide the composition table as part of the prompt. Our main focus is thus on whether LLMs and LRMs are able to follow the instructions and apply the composition rules in a systematic way. This allows us to evaluate models in a zero-shot fashion, or with a small number of in-context demonstrations (as well as evaluating fine-tuned models which should in principle be able to learn the composition table). We specify the composition table using a compact integer encoding (using powers of two; see the appendix for an example of the full prompt). The graph that defines a given problem instance is similarly encoded using integer labels. The model is furthermore instructed to provide the answer using the same integer encoding. This is illustrated in Fig. 1.

Evaluation setup To evaluate the models, for each combination of (k, b), we use a uniform subsample of the full set of test problem instances for RCC-8 and IA. For RCC-8, each of the eight relations appears equally frequently as gold labels, meaning that the performance of naive baselines such as random guessing is at $1/8 = 0.125$. Similarly, the performance of naive baselines on IA is at $1/13 \approx 0.076$. We evaluate 2 types of models, LLMs (instruction tuned) and LRMs, on 3 distinct settings: (A) **Zero-shot**, (B) **Few-shot** and (C) **Fine-tuned**. Settings (A) and (B) evaluate the model's in-context learning and instruction following abilities. For the few-shot experiments, we provide 5 in-context demonstrations of the desired input/output pairs. For the experiments with fine-tuned models, we leverage the entire training set comprising 57600 and 93400 instances for RCC-8 and IA respectively. For testing, for settings (A) and (B) we use 500 test sample instances for RCC-8 and 100 for IA, for each combination of k and b. We use 50 samples per (k, b) configuration for setting (C) and the reasoning experiment. We use the following models: Llama-3 and Llama-3.3 (Grattafiori et al., 2024), Qwen (Qwen et al., 2025), Phi-4 (Abdin et al., 2024), Gemma-2 (Team et al., 2024) o3-mini (OpenAI et al., 2025). The setup is summarized in Table 1. Further implementation details and data statistics are provided in App. B.

5 Results

We now present an overview of the results, focusing first on the non-reasoning models in Section 5.1 (i.e. the standard LLMs), and then on the reasoning models in Section 5.2.

5.1 Non-reasoning models

The results for RCC-8 are summarized in Figure 3 and for IA in Figure 4. Broadly, both of these are similar. We therefore focus on RCC-8 below.

For the zero-shot experiments, all models perform close to random guessing for all but the simplest problem instances. Somewhat better results are observed only when $b \leq 2$ and $k \leq 4$. Qwen2.5-72B overall emerges as the strongest model. Its results remain clearly above random chance (although still very weak) for $b = 1$ and $k \geq 5$. For lower values of k, gemma-2-9b and gemma-2-27b are the next best-peforming models. Interestingly, the much larger Llama-3.3-70B model performs poorly for low values of k, but performs the best for $b = 2, k = 10$, and similar to Qwen2.5-72B for $b = 1, k = 10$.

The results for the few-shot experiments are similar, with non-trivial performance only achieved for $k \leq 3$. Qwen2.5-72B performs consistently better than in the zero-shot case. For Llama-3.3-70B we also see clear improvements for $b \in \{1, 2\}$ (and to some extent also $b = 4$). The most interesting changes can be seen for $b = 1$, where some of the smaller models now perform notably better, especially Qwen2.5-14B, gemma-2-9b and phi-4.

Finally, the results for the fine-tuned models are much better. We can see a noticeable performance gap, with the Qwen models and gemma-2-27b clearly outperforming the others. Interestingly, this is also the case for the smallest Qwen model (Qwen2.5-7B). In contrast, the two Llama models clearly underperform, with even Llama-3.3-70B only beating the much smaller phi-4 model on the hardest problem instances. It is surprising to see that the performance for $b = 2$, $b = 3$ and $b = 4$ is similar, despite the latter setting being much harder. We will come back to this point in Section 6. In short, however, this is due to the fact that these models have learned to reliably predict some of the simplest relations. For instance, eq can only be predicted if there is a path between the head and tail entities that only consists of eq relations. For some of the other relations, there are similar insights that can be leveraged. The ability of these

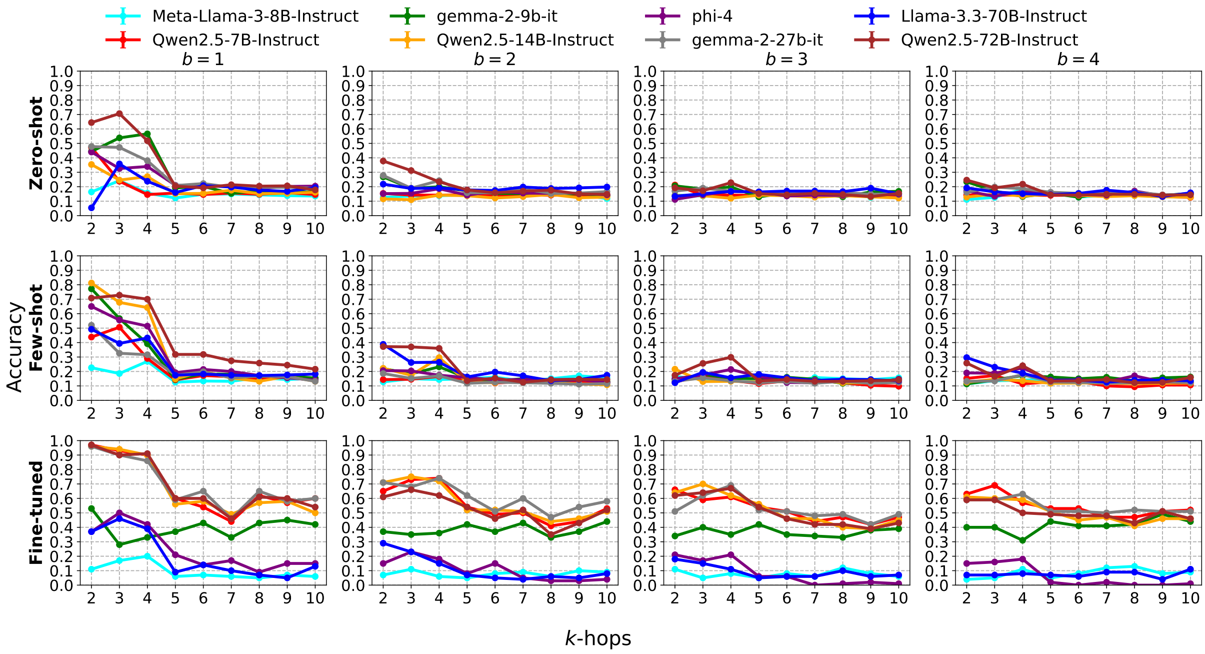


Figure 3: The results for the non-reasoning models on RCC-8 for the 3 settings (accuracy).

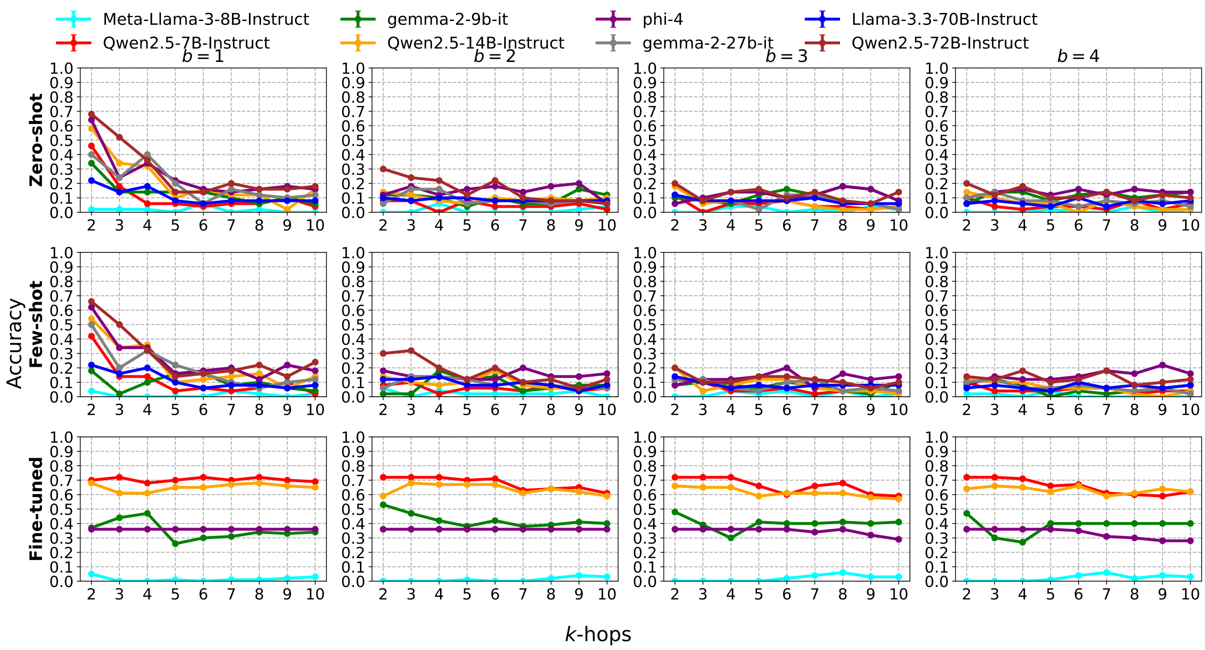


Figure 4: The results for the non-reasoning models on IA for the 3 settings (accuracy).

models to discover the underlying principles, and reliably apply them to out-of-distribution settings is remarkable. At the same time, however, it is clear that they are not capable of principled reasoning, as their performance on the hardest relations remains poor. The performance of all the considered models, even the best-performing fine-tuned models, remains far below that of state-of-the-art neuro-symbolic methods (Khalid and Schockaert, 2025), which achieve near-perfect results on these problem instances, despite having to learn the composition table from training examples.

5.2 Reasoning models

For the reasoning models, we focus on the zero-shot evaluation setting. The results are summarized in Table 2. Note that we only include results for a sample of all (k, b) configurations due to the much higher cost that is involved in using these models.

Compared to the non-reasoning models without

	Conf.	o3-mini		Qwen 7B		Qwen 14B	
	(k, b)	Acc	F1	Acc	F1	Acc.	F1
RCC-8	(9, 3)	0.30	0.24	0.12	0.07	0.06	0.05
	(9, 2)	0.48	0.38	0.06	0.02	0.26	0.23
	(9, 1)	0.90	0.85	0.08	0.07	0.20	0.15
	(8, 4)	0.44	0.35	0.10	0.08	0.16	0.12
	(8, 3)	0.56	0.52	0.12	0.11	0.14	0.10
	(5, 2)	0.68	0.63	0.12	0.07	0.18	0.15
IA	(9, 3)	0.30	0.29	0.04	0.03	0.10	0.10
	(9, 2)	0.44	0.42	0.06	0.04	0.22	0.18
	(9, 1)	0.78	0.74	0.20	0.15	0.14	0.09
	(8, 4)	0.36	0.30	0.04	0.06	0.12	0.07
	(8, 3)	0.34	0.36	0.04	0.03	0.14	0.07
	(5, 2)	0.56	0.52	0.04	0.03	0.04	0.03

Table 2: Zero-shot (setting (A)) results for the reasoning models on the STaR benchmark. The Qwen models are distilled R1 models which were run locally. The accuracies and macro F1 scores are reported for a sample of test configurations due to API resource constraints.

	Label	Pr.	Re.	F1.	Count
RCC-8	`DC`	0.14	0.31	0.20	13
	`EC`	0.43	0.25	0.32	12
	`PD`	0.14	0.18	0.16	11
	`TPP`	**1.00**	0.09	0.17	11
	`NTPP`	0.00	0.00	0.00	**17**
	`TPPI`	0.72	**1.00**	0.84	13
	`NTPPI`	0.68	**1.00**	0.81	13
	`EQ`	**1.00**	**1.00**	**1.00**	10
IA	`=`	0.14	0.83	0.24	6
	`<`	0.00	0.00	0.00	4
	`>`	0.00	0.00	0.00	9
	`d`	**1.00**	0.10	0.18	10
	`di`	0.00	0.00	0.00	9
	`o`	**1.00**	0.57	0.73	7
	`oi`	**1.00**	**1.00**	**1.00**	5
	`m`	**1.00**	**1.00**	**1.00**	9
	`mi`	**1.00**	0.67	0.80	6
	`s`	**1.00**	**1.00**	**1.00**	9
	`si`	**1.00**	**1.00**	**1.00**	8
	`f`	**1.00**	0.83	0.91	6
	`fi`	**1.00**	**1.00**	**1.00**	12

Table 3: Fine-grained breakdown of classification scores for the $k = 9, b = 2$ dataset configuration for the fine-tuned Qwen2.5-14B LLM. We sample 50 points randomly from each STaR dataset.

fine-tuning, the performance of o3-mini (OpenAI et al., 2025) is remarkably strong. The setting with $b = 1$ is intuitively well-aligned with the chain-of-thought process. Accordingly, we can see that the model performs well for $b = 1$, even with $k = 9$, achieving an accuracy of 0.9, which is substantially higher than what any of the fine-tuned models has achieved. However, for $b \geq 2$ the results quickly deteriorate. Interestingly, this behavior is qualitatively different from that of the fine-tuned models. Where the fine-tuned models have learned to identify easy-to-predict relations, o3-mini seems capable of interpreting the composition table and systematically applying it to a single reasoning path (although not with perfect accuracy, even for $b = 1$). For $b \geq 2$, the disjunctive nature of the reasoning problem proves problematic, suggesting that the model is limited in its capacity to generalize to unseen reasoning tasks.

For the distilled Deepseek-R1 models (Guo et al., 2025), the results are below random chance for all settings where $b \geq 2$. For $k = 9$ and $b = 1$, the results are above random chance (except for Qwen 7B on RCC-8), but not meaningfully better than the non-reasoning models in the zero-shot setting.

6 Analysis

In Section 5, we already saw that the behavior of the fine-tuned LLMs, on the one hand, and o3-mini, on the other hand, was qualitatively different. To further analyze this, Table 3 shows a breakdown of the results per relation type, for one of the best-performing fine-tuned models (Qwen2.5-14B). Table 4 shows the same breakdown for o3-mini. In both tables, we focus on the case where $k = 9$ and $b = 2$. Focusing on Table 3 first, for RCC-8 we can see that the fine-tuned Qwen2.5-14B model achieves perfect results on `eq`, which can be explained by the fact that this relation can only be predicted if there is a chain of `eq` relations between the head and tail entity. For `ntppi` and `tppi`, the model was able to exploit a similar insight. The performance on the other relations, however, is much worse, although still better than random chance (except for `ntpp`). For IA, we can see a similar pattern. Some of the relations are easier to predict, with the model achieving perfect results on several relations: `oi`, `m`, `s`, `si` and `fi`. However, for other relations, the results are very poor. This again shows that the model was able to learn some “tricks” that allow it to reliably predict some of the easier relations, even on out-of-distribution settings, while at the same time failing to apply the rules from the composition table in a systematic way.

The results for o3-mini in Table 4 paint a dramatically different picture. First, note that o3-mini does not achieve perfect results on any of the relations. This shows that it was not able to leverage domain-specific insights (such as the idea that `eq` can only be predicted if there is a chain of `eq`-relations). On the other hand, the model achieves

	Label	Pr.	Re.	F1.	Count
RCC-8	DC	0.69	0.90	**0.78**	10
	EC	0.50	**1.00**	0.67	3
	PD	0.43	0.27	0.33	11
	TPP	0.33	0.44	0.38	9
	NTPP	**1.00**	0.20	0.33	5
	TPPI	0.00	0.00	0.00	2
	NTPPI	0.50	0.25	0.33	4
	EQ	0.75	0.50	0.60	6
IA	=	0.50	0.17	0.25	6
	<	0.10	**1.00**	0.18	1
	>	0.83	**1.00**	**0.91**	5
	d	0.50	0.60	0.55	5
	di	0.67	0.50	0.57	4
	o	0.00	0.00	0.00	2
	oi	0.75	**1.00**	0.86	3
	m	**1.00**	0.50	0.67	4
	mi	**1.00**	0.33	0.50	3
	s	**1.00**	0.20	0.33	5
	si	**1.00**	0.25	0.40	4
	f	0.33	0.50	0.40	2
	fi	**1.00**	0.17	0.29	6

Table 4: Fine-grained breakdown of classification scores for the $k = 9, b = 2$ dataset configuration for the o3-mini LRM. We sample 50 points randomly from each STaR dataset.

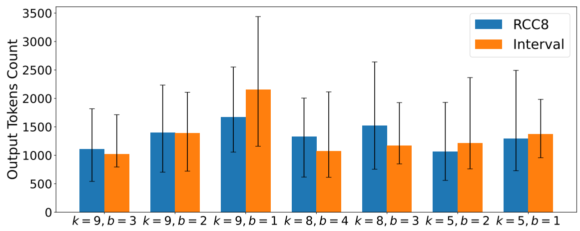


Figure 5: The median number of output tokens with the interquartile range for the Qwen 7B reasoning model for the same dataset splits as in Table 2. The number of maximum tokens was set to 8192.

non-trivial results for almost all the relations. This suggests that the correct predictions are due to the ability of the model to follow the instructions from the composition table in a somewhat systematic, albeit error-prone way.

CoT analysis Reasoning models can adapt the number of output tokens, i.e. the amount of test-time compute, based on the difficulty of a given problem instance. To analyze this aspect, Figure 5 shows the number of output tokens that were generated by the Qwen 7B reasoning model. Note that we cannot do this analysis for o3-mini as the intermediate reasoning process is hidden for this model. Counterintuitively, the analysis in Figure 5 reveals that the number of output tokens goes down, as the number of paths b increases, for all the considered values of k. This seems to suggest that the model is aware of its limitations on these problem instances, giving up the reasoning process more quickly. In contrast, we can see that considerably more output tokens were used for $k = 9, b = 1$ than for $k = 5, b = 1$, which further supports our hypothesis that problem instances with $b = 1$ are more natural for chain-of-thought based reasoning problems.

7 Conclusions

We have studied the performance of recent LLMs and so-called Large Reasoning Models (LRMs) on a challenging benchmark involving qualitative spatial and temporal reasoning problems. This analysis allows us test the abilities of models on a different style of reasoning problems than those that are typically considered, and crucially, than those that are used for training LRMs. The considered problems involve composing relations, using rules that are specified in a composition table. A particular challenge arises because multiple "reasoning paths" need to be combined to arrive at the final answer, which is harder to capture using a chain-of-thought process.

Several insights arise from our analysis. While LLMs perform poorly in zero-shot and few-shot settings, fine-tuned LLMs achieved notably better results. However, further analysis shows that this is because fine-tuned models achieve near-perfect results on some of the easier test instances, i.e. relations that can be predicted by relying on simple rules, rather than a systematic application of the rules from the composition table. In particular, these models still perform poorly on problem instances that require systematic reasoning. As far as LRMs are concerned, o3-mini performs much better than LLMs in zero-shot and few-shot settings, but does not overall improve on the performance of fine-tuned LLMs. Interestingly, the behavior of the fine-tuned LLMs and o3-mini is qualitatively different. Indeed, o3-mini seems to rely more on an error-prone, but systematic application of the rules from the composition table, achieving strong results for problems involving only a single reasoning path. However, when multiple reasoning paths need to be combined, its performance deteriorates quickly. These results suggest that LRMs, despite their clearly improved reasoning abilities, are still limited in terms of generalizing to previously unseen reasoning tasks.

Limitations

The state-of-the-art in reasoning models is still quickly changing, and any conclusions that can be drawn from current models, such as o3-mini, may quickly become obsolete as newer models are released. A key question, which remains unanswered, is whether reasoning models can be designed that generalize to previously unseen reasoning tasks. Furthermore, while we have advocated the use of temporal and spatial reasoning, further analysis is needed to test the reasoning abilities of current models on a broader range of problems, and to better understand their failure modes more generally. In terms of the considered models, we have focused our analysis on open-source models that can be run locally (with the exception of o3-mini), and quantization was used to make this possible. It is possible that fine-tuning larger models may lead to better results. Finally, only a limited set of (k, b) configurations was used to evaluate the reasoning models due to compute constraints.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

James F Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2024. The reversal curse: LLMs trained on “a is b” fail to learn “b is a”. In *The Twelfth International Conference on Learning Representations*.

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

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Kewei Cheng, Nesreen K. Ahmed, and Yizhou Sun. 2023. Neural compositional rule learning for knowledge graph reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Anthony G. Cohn and Robert E. Blackwell. 2024a. Can large language models reason about the region connection calculus? *CoRR*, abs/2411.19589.

Anthony G. Cohn and Robert E. Blackwell. 2024b. Evaluating the ability of large language models to reason about cardinal directions (short paper). In *16th International Conference on Spatial Information Theory, COSIT 2024, September 17-20, 2024, Québec City, Canada*, volume 315 of *LIPIcs*, pages 28:1–28:9. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2021. 8-bit optimizers via block-wise quantization. *arXiv preprint arXiv:2110.02861*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2023. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and et al Alex Vaughan. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, et al. 2024. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. 2024. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*.

Cheongwoong Kang and Jaesik Choi. 2023. Impact of co-occurrence on factual knowledge of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7721–7735, Singapore. Association for Computational Linguistics.

Irtaza Khalid and Steven Schockaert. 2025. Systematic relational reasoning with epistemic graph neural networks. In *The Thirteenth International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *ICLR*.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253.

Fangjun Li, David C. Hogg, and Anthony G. Cohn. 2024a. Advancing spatial reasoning in large language models: An in-depth evaluation and enhancement using the stepgame benchmark. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18500–18507. AAAI Press.

Fangjun Li, David C. Hogg, and Anthony G. Cohn. 2024b. Reframing spatial reasoning evaluation in language models: A real-world simulation benchmark for qualitative reasoning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 6342–6349. ijcai.org.

Wenbin Li, Di Yao, Ruibo Zhao, Wenjie Chen, Zijie Xu, Chengxue Luo, Chang Gong, Quanliang Jing, Haining Tan, and Jingping Bi. 2024c. Stbench: Assessing the ability of large language models in spatio-temporal analysis. *CoRR*, abs/2406.19065.

Shengyao Lu, Bang Liu, Keith G. Mills, Shangling Jui, and Di Niu. 2022. R5: rule discovery with reinforced and recurrent relational reasoning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D Hardy, and Thomas L Griffiths. 2024. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121.

Pasquale Minervini, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktäschel. 2020. Learning reasoning strategies in end-to-end differentiable proving. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6938–6949. PMLR.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. 2021. SPARTQA: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4582–4598, Online. Association for Computational Linguistics.

Timothy Nguyen. 2024. Understanding transformers via n-gram statistics. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. 2024. Arithmetic without algorithms: Language models solve math with a bag of heuristics. *CoRR*, abs/2410.21272.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *Preprint*, arXiv:2112.00114.

OpenAI, :, Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, Jerry Tworek, Lorenz Kuhn, Lukasz Kaiser, Mark Chen, Max Schwarzer, Mostafa Rohaninejad, Nat McAleese, o3 contributors, Oleg Mürk, Rhythm Garg, Rui Shu, Szymon Sidor, Vineet Kosaraju, and Wenda Zhou. 2025. Competitive programming with large reasoning models. *Preprint*, arXiv:2502.06807.

Yonatan Oren, Nicole Meister, Niladri S. Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. 2024. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

David A. Randell, Zhan Cui, and Anthony G. Cohn. 1992. A spatial logic based on regions and connection. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92). Cambridge, MA, USA, October 25-29, 1992*, pages 165–176. Morgan Kaufmann.

Jochen Renz and Gérard Ligozat. 2005. Weak composition for qualitative spatial and temporal reasoning. In *Principles and Practice of Constraint Programming - CP 2005, 11th International Conference, CP 2005, Sitges, Spain, October 1-5, 2005, Proceedings*, volume 3709 of *Lecture Notes in Computer Science*, pages 534–548. Springer.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Zhengxiang Shi, Qiang Zhang, and Aldo Lipani. 2022. Stepgame: A new benchmark for robust multi-hop spatial reasoning in texts. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11321–11329. AAAI Press.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4505–4514. Association for Computational Linguistics.

Wangtao Sun, Chenxiang Zhang, Xueyou Zhang, Ziyang Huang, Haotian Xu, Pei Chen, Shizhu He, Jun Zhao, and Kang Liu. 2024. Beyond instruction following: Evaluating rule following of large language models. *CoRR*, abs/2407.08440.

Yihong Tang, Ao Qu, Zhaokai Wang, Dingyi Zhuang, Zhaofeng Wu, Wei Ma, Shenhao Wang, Yunhan Zheng, Zhan Zhao, and Jinhua Zhao. 2024. Sparkle: Mastering basic spatial capabilities in vision language models elicits generalization to composite spatial reasoning. *CoRR*, abs/2410.16162.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, and et al Ravin Kumar. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Karthik Valmeekam, Kaya Stechly, and Subbarao Kambhampati. 2024. Llms still can't plan; can lrms? A preliminary evaluation of openai's o1 on planbench. *CoRR*, abs/2409.13373.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, and Neel Joshi. 2024. Is A picture worth A thousand words? delving into spatial reasoning for vision language models. *CoRR*, abs/2406.14852.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Thomas Wolf. 2020. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024. Visualization-of-thought elicits spatial reasoning in large language models. *CoRR*, abs/2404.03622.

Yutaro Yamada, Yihan Bao, Andrew Kyle Lampinen, Jungo Kasai, and Ilker Yildirim. 2024. Evaluating spatial understanding of large language models. *Trans. Mach. Learn. Res.*, 2024.

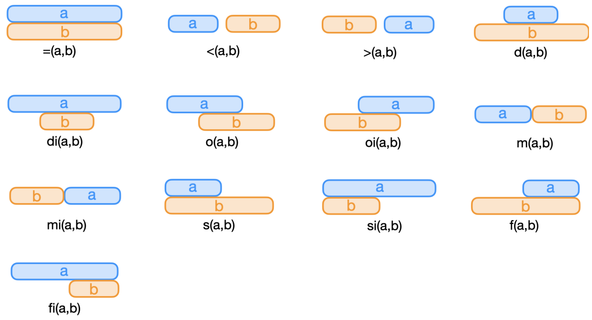


Figure 6: Illustration of the IA relations.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Ge Zhang, Mohammad Ali Alomrani, Hongjian Gu, Jiaming Zhou, Yaochen Hu, Bin Wang, Qun Liu, Mark Coates, Yingxue Zhang, and Jianye Hao. 2024a. Path-of-thoughts: Extracting and following paths for robust relational reasoning with large language models. *CoRR*, abs/2412.17963.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, William Song, Tiffany Zhao, Pranav Vishnu Raja, Charlotte Zhuang, Dylan Z Slack, Qin Lyu, Sean M. Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. 2024b. A careful examination of large language model performance on grade school arithmetic. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yi Zhang, Arturs Backurs, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, and Tal Wagner. 2023. Unveiling transformers with lego: a synthetic reasoning task. *Preprint*, arXiv:2206.04301.

Zhaocheng Zhu, Yuan Xue, Xinyun Chen, Denny Zhou, Jian Tang, Dale Schuurmans, and Hanjun Dai. 2024. Large language models can learn rules. *Preprint*, arXiv:2310.07064.

A Details on RCC-8 and IA

Figure 6 provides an illustration of the 13 relations of the interval algebra. The composition tables for RCC-8 and IA are shown respectively in Tables 5 and 6. To illustrate how reasoning with these calculi works, suppose we are given the following facts:

$$\mathsf{ec}(a,b) \quad \mathsf{ntpp}(b,c) \quad \mathsf{po}(a,d) \quad \mathsf{ec}(d,c)$$

Using the composition table, from $\mathsf{ec}(a,b)$ and $\mathsf{ntpp}(b,c)$, we know that the following must hold:

$$\mathsf{po}(a,c) \vee \mathsf{tpp}(a,c) \vee \mathsf{ntpp}(a,c)$$

Similarly, from $\mathsf{po}(a,d)$ and $\mathsf{ec}(d,c)$, we know that the following must hold:

$$\mathsf{dc}(a,c) \vee \mathsf{ec}(a,c) \vee \mathsf{po}(a,c) \vee \mathsf{tppi}(a,c) \vee \mathsf{ntppi}(a,c)$$

Since it is not possible for more than one relation to hold between a and c, the only possibility is that $\mathsf{po}(a,c)$ holds.

In general, sound and complete reasoning in RCC-8 and IA is possible by using the algebraic closure algorithm (for the case where the initial information does not contain any disjunctions). This algorithm amounts to maintaining, for each pair of entities, a set of possible relations. These sets are iteratively refined by applying composition rules, until convergence. The algorithm runs in cubic time. The problem instances in the StaR benchmark are simpler than general RCC-8 and IA problems. For these instances, it always suffices to consider the paths between the designated entities h and t. Each path gives rise to a set of candidate relations, and the final answer is obtained by taking the intersection of these sets. The complexity of reasoning is thus linear in the number of entities. This ensures that the considered models should, in principle, be powerful enough to solve the problem instances, even for larger problems, and without needing an excessive number of output tokens for the LRMs.

B Implementation Details

B.1 Compute resources

All relevant hyperparameters were tuned using grid search, as detailed below. All experiments were conducted using RTX 4090 and RTX 6000 Ada NVIDIA GPUs. For the small models, the results for all (k,b) configurations, for the zero-shot, few-shot and fine-tuned settings, can be obtained in around 6-8 hours per model. For the large 70B models at 4-bit quantization, with a smaller sample size of 50 instances per (k,b) configuration, a single full run (i.e. 24 (k,b) configurations) takes around 1 day. We use the unsloth library (Daniel Han and team, 2023) for fine-tuning all models with 4-bit quantization and the transformers library for downloading the weights and running all the open-source models locally (Wolf, 2020).

	dc	ec	po	tpp	ntpp	tppi	ntppi
dc	$\mathcal{R}_8$	dc, ec, po, tpp, ntpp	dc, ec, po, tpp, ntpp	dc, ec, po, tpp, ntpp	dc, ec, po, tpp, ntpp	dc	dc
ec	dc, ec, po, tppi, ntppi	dc, ec, po, tpp, tppi, eq	dc, ec, po, tpp, ntpp	ec, po, tpp, ntpp	po, tpp, ntpp	dc, ec	dc
po	dc, ec, po, tppi, ntppi	dc, ec, po, tppi, ntppi	$\mathcal{R}_8$	po, tpp, ntpp	po, tpp, ntpp	dc, ec, po, tppi, ntppi	dc, ec, po, tppi, ntppi
tpp	dc	dc, ec	dc, ec, po, tpp, ntpp	tpp, ntpp	ntpp	dc, ec, po, tpp, tppi, eq	dc, ec, po, tppi, ntppi
ntpp	dc	dc	dc, ec, po, tpp, ntpp	ntpp	ntpp	dc, ec, po, tpp, ntpp	$\mathcal{R}_8$
tppi	dc, ec, po, tppi, ntppi	ec, po, tppi, ntppi	po, tppi, ntppi	po, eq, tpp, tppi	po, tpp, ntpp	tppi, ntppi	ntppi
ntppi	dc, ec, po, tppi, ntppi	po, tppi, ntppi	po, tppi, ntppi	po, tppi, ntppi	po, tppi, tpp, ntpp, ntppi, eq	ntppi	ntppi

Table 5: RCC-8 composition table (Randell et al., 1992), excluding the trivial composition with eq. We write $\mathcal{R}_8$ for the trivial case, where the composition consists of all eight relations.

B.2 Hyper Parameters

We use the 8-bit quantized AdamW optimizer (Dettmers et al., 2021; Kingma and Ba, 2017) for fine-tuning the models. We use the same fine-tuning strategy and hyperparameters for all the models that are trained locally. For inference, the maximum output tokens for the non-reasoning models is set to 256. For fine-tuning we use a learning rate of 2×10^{-4} with a maximum step size of 60 and weight decay with a linear scheduler for all the models. We use gradient accumulation with steps 4 and only fine-tune for 1 epoch since further training did not meaningfully improve the validation loss. To maximize GPU memory utilization with respect to model size, we make use of Flash attention (Dao et al., 2022) and quantized low rank adaptors (Dettmers et al., 2024). The adaptors are applied as Q, K, V, O, Gate, Up and Down projectors with hidden dimension size of 128 for all small and medium models and 64 for large models (the latter only because 128 could not fit in memory on the RTX 6000 Ada).

For the reasoning Qwen models in Table 2, we set the maximum output tokens to 8192, and for o3-mini this is set to 15000.

B.3 Data Statistics

The dataset statistics for the STaR benchmark for the training and test sets are summarized in the Table 7. These are respectively subsampled for the experimental evaluations in the main text. All random sampling is done with a global seed of 0 for reproducibility. Some example graphs generated via this procedure for the RCC-8 dataset are displayed in Figure 10.

B.4 Prompts

The prompts used for non-fine tuning experiments for RCC-8 are shown in Fig. 7 with mutatis mutandis changes for IA and for IA for the instruction-tuning setting in Fig. 8 with similar changes for RCC-8. We experimented with textual graph labels as opposed to integers in the prompt and the requested output format but found the accuracy and the adherence of the small models to be extremely poor in this setting with very low accuracies.

C Additional Analysis

Conducting a fine-grained classification level analysis of o3-mini for the instances where it thought for longer than 15000 tokens and responded with nothing over all the reasoning datasets is shown in figure 9. We find that o3-mini took unexpectedly longer for the trivial relations such as $=$, and for and fi for IA and po for RCC-8.

Figure 7: The given prompt is for the inference RCC-8 dataset, while the Interval prompt for inference has a similar structure but different base elements and composition table.

Input B.1: RCC8 Inference Prompt

```
System: You are a helpful assistant. Just answer the question as a single integer.
```

```
User: You are a qualitative spatial and temporal reasoning expert specializing in
RCC-8

The following are the base elements of RCC-8:

    DC = 1
    EC = 2
    PO = 4
    TPP = 8
    NTPP = 16
    TPPI = 32
    NTPPI = 64
    EQ = 128

The following is the composition table of RCC-8 as a JSON dictionary:
{(1, 1): [], (1, 2): [1, 2, 4, 8, 16], ..., (128, 64): [64], (128, 128):
[128]}

Now the question is: Given a consistent graph with edges comprising the 8
base relations, predict the label of the target edge. More specifically,
Given a data row delimited by a comma with the following columns:
`graph_edge_index`, `edge_labels`, `query_edge`, predict the label of the
`query_edge` as one of the 8 base relations as a power of 2 as defined above.
```

```
(The optional few-shot examples:
Example 1:
[(0, 1), (1, 2)], ['EQ', 'NTPPI'], (0, 2)
64

...

Example 5:
[(0, 1), (1, 2), (2, 3)], ['EQ', 'EQ', 'EC'], (0, 3)
2
Examples end here.
)
```

```
[(0, 1), (1, 4), (0, 2), (2, 4), (0, 3), (3, 4)],
['EQ', 'NTPPI', 'EQ', 'NTPPI', 'TPPI', 'NTPPI'], (0, 4)
```

Figure 8: The given prompt is for the finetuining interval dataset, while the RCC-8 prompt for finetuning has a similar structure but different base elements and composition table.

Input B.2: Interval Finetuning Prompt

```
Below is an instruction that describes a task, paired with an input that provides
further context. Write a response that appropriately completes the request.

### Instruction:
You are a qualitative spatial and temporal reasoning expert specializing in
Interval Algebra.

The following are the base elements of Interval Algebra:

    '=': 1
    '<': 2
    '>': 4
    'd': 8
    'di': 16
    'o': 32
    'oi': 64
    'm': 128
    'mi': 256
    's': 512
    'si': 1028
    'f': 2048
    'fi': 4096

The following is the composition table of RCC-8 as a JSON dictionary:
(eq, eq): [eq], (eq, lt): [lt],, ..., (fi, gt): [gt, oi, di, mi, si]}

Now the question is: Given a consistent graph with edges comprising the 8
base relations, predict the label of the target edge. More specifically,
Given a data row delimited by a comma with the following columns:
`graph_edge_index`, `edge_labels`, `query_edge`, predict the label of the
`query_edge` as one of the 8 base relations as a power of 2 as defined above.


### Input:
[(0, 1), (1, 4), (0, 2), (2, 4), (0, 3), (3, 4)],
['m', '>', 'di', 'fi', 'di', 'oi'], (0, 4)

### Response:
16
```

	<	>	d	di	o	oi	m	mi	s	si	f	fi
<	<		<, o, m, d, s	<	<	<, o, m, d, s	<	<, o, m, d, s	<	<	<, o, m, d, s	<
>		>	>, oi, mi, d, f	>	>, oi, mi, d, f	>	>, oi, mi, d, f	>	>, oi, mi, d, f	>	>	>
d	<	>	d		<, o, m, d, s	>, oi, mi, d, f	<	>	d	>, oi, mi, d, f	d	<, o, m, d, s
di	<, o, m, di, fi	>, oi, di, mi, si	o, oi, d, s, f, di, si, fi, =	di	o, di, fi	oi, di, si	o, di, fi	oi, di, si	o, di, fi	di	oi, di, si	di
o	<	>, oi, di, mi, si	o, d, s	<, o, m, di, fi	<, o, m	o, oi, d, s, f, di, si, fi, =	<	oi, di, si	o	o, di, fi	o, d, s	<, o, m
oi	<, o, m, di, fi	>	oi, d, f	>, oi, mi, di, si	o, oi, d, di, s, si, f, fi, =	>, oi, mi	o, di, fi	>	oi, d, f	oi, >, mi	oi	oi, di, si
m	<	>, oi, di, mi, si	o, d, s	<	<	o, d, s	<	f, fi, =	m	m	d, s, o	<
mi	<, o, m, di, fi	>	oi, d, f	>	oi, d, f	>	s, si, =	>	d, f, oi	>	mi	mi
s	<	>	d	<, o, m, di, fi	<, o, m	oi, d, f	<	mi	s	s, si, =	d	<, m, o
si	<, o, m, di, fi	>	oi, d, f	di	o, di, fi	oi	o, di, fi	mi	s, si, =	si	oi	di
f	<	>	d	>, oi, mi, di, si	o, d, s	>, oi, mi	m	>	d	>, oi, mi	f	f, fi, =
fi	<	>, oi, di, mi, si	o, d, s	di	o	oi, di, si	m	si, oi, di	o	di	f, fi, =	fi

Table 6: Allen's interval algebra composition table (Allen, 1983), excluding the trivial composition with $=$.

Table 7: Data statistics of the STaR reasoning datasets. These are respectively subsampled for the experimental valuations in the main text.

Dataset	Training regime	No. of relations	# Train	# Test per config.	Test regime
RCC-8	$b \in \{1,2,3\}, k \in \{2,3\}$	8	57,600	6,400	$b \in \{1,2,4\}, k \in \{2,\dots,10\}$
IA	$b \in \{1,2,3\}, k \in \{2,3\}$	13	93,400	9,300	$b \in \{1,2,4\}, k \in \{2,\dots,10\}$

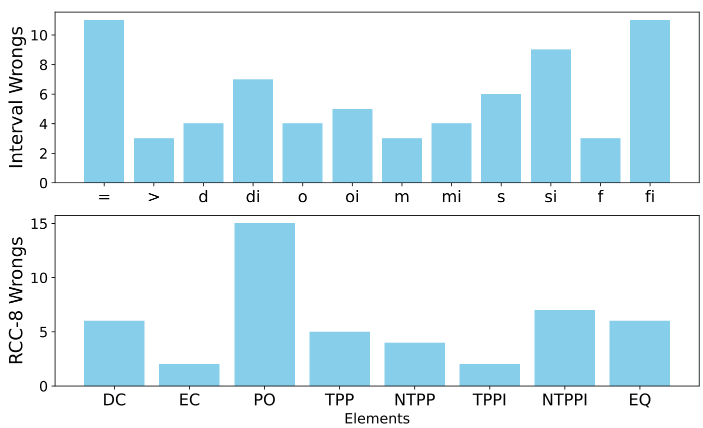
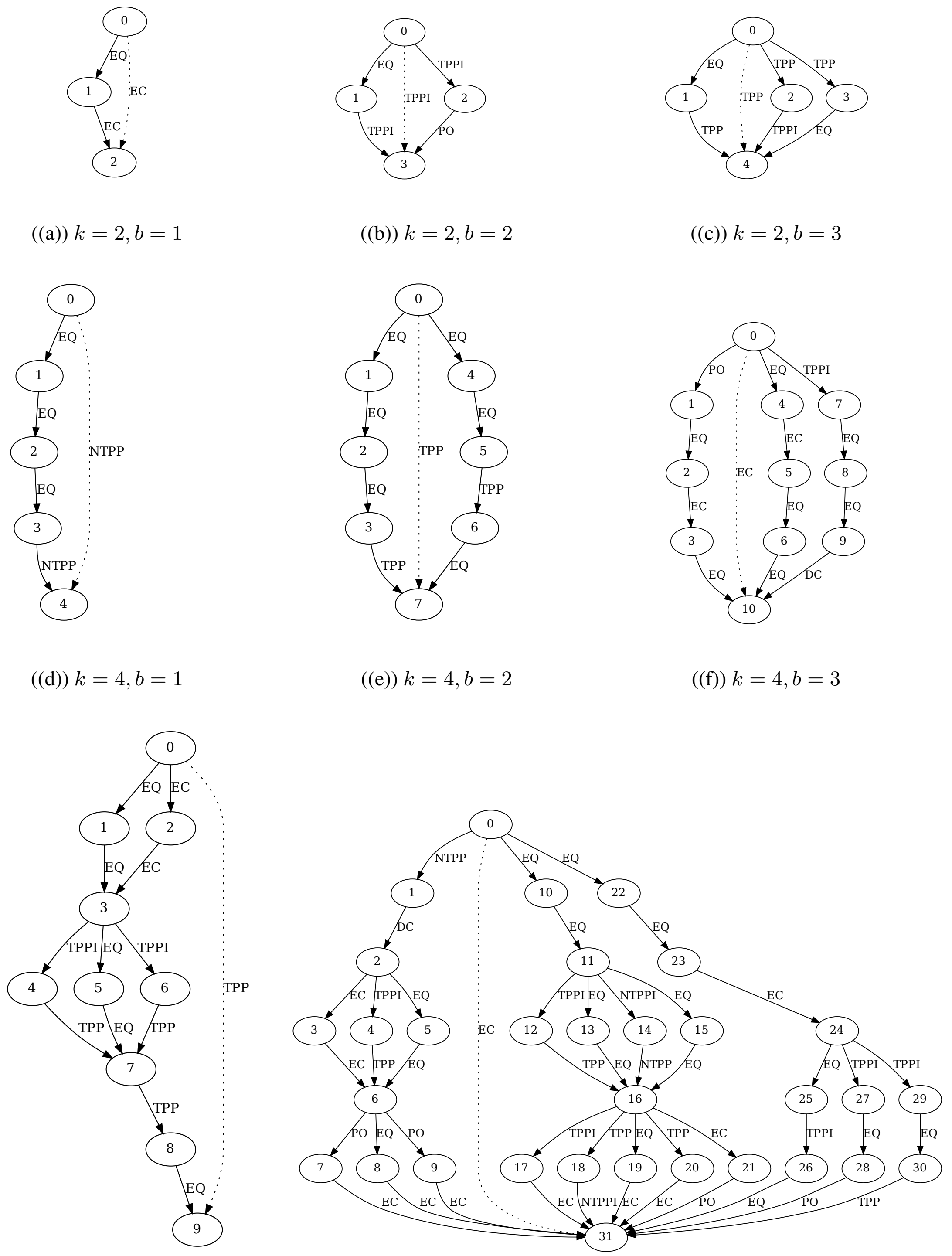


Figure 9: Non-responses from o3 where it took longer than the maximum allotted number of tokens. Certain classes are overrrepresented and for IA coincide with those that can easily predicted by leveraging heuristics based on dataset construction constraints.



((a)) $k = 2, b = 1$ ((b)) $k = 2, b = 2$ ((c)) $k = 2, b = 3$

((d)) $k = 4, b = 1$ ((e)) $k = 4, b = 2$ ((f)) $k = 4, b = 3$

((g)) $k = 6, b = 1$ ((h)) $k = 6, b = 3$

Figure 10: Some graph instances for the RCC-8 dataset generated using the procedure described in (Khalid and Schockaert, 2025). The target edge label between the source node and the tail node that needs to be predicted by the model is indicated by the dotted line.