Improving Chain-of-Thought Reasoning via Quasi-Symbolic Abstractions

Leonardo Ranaldi[†,⊕] **Marco Valentino**[†,*] **André Freitas**[†,○,•]
[†]Idiap Research Institute, Switzerland
[⊕]School of Informatics, University of Edinburgh, UK
[*]School of Computer Science, University of Sheffield, UK
[○]Department of Computer Science, University of Manchester, UK
[•]National Biomarker Centre (NBC), CRUK Manchester Institute, UK
[name].[surname]@idiap.ch

Abstract

Chain-of-Thought (CoT) represents a common strategy for reasoning in Large Language Models (LLMs) by decomposing complex tasks into intermediate inference steps. However, explanations generated via CoT are susceptible to *content biases* that negatively affect their *robustness* and *faithfulness*. To mitigate existing limitations, recent work has proposed the use of logical formalisms coupled with external symbolic solvers. However, fully symbolically formalised approaches introduce the bottleneck of requiring a complete translation from natural language to formal languages, a process that affects efficiency and flexibility. To achieve a trade-off, this paper investigates methods to disentangle content from logical reasoning without a complete formalisation. In particular, we present *QuaSAR* (for Quasi-Symbolic Abstract Reasoning), a variation of CoT that guides LLMs to operate at a higher level of abstraction via *quasi-symbolic explanations*. Our framework leverages the capability of LLMs to formalise only relevant variables and predicates, enabling the coexistence of symbolic elements with natural language. We show the impact of *QuaSAR* for in-context learning and for constructing demonstrations to improve the reasoning capabilities of smaller models. Our experiments show that quasi-symbolic abstractions can improve CoT-based methods by up to 8% accuracy, enhancing robustness and consistency on challenging adversarial variations on both natural language (i.e. MMLU-Redux) and symbolic reasoning tasks (i.e., GSM-Symbolic).

1 Introduction

Multi-step reasoning methods, best exemplified by Chain-of-Thought (Wei et al., 2022; Wang et al., 2022), have been proposed to improve the performance of Large Language Models (LLMs) on downstream tasks by breaking down complex problems into intermediate reasoning steps. The success of these methods is due to the LLMs' properties of performing tasks by following in-context structured requirements (Zhou et al., 2023; Dong et al., 2024; Ranaldi et al., 2024b,a).

Despite CoT being the current workhorse for LLM reasoning, complex reasoning still remains a significant challenge for LLMs (Meadows and Freitas, 2023; Luo et al., 2024), with recent work showing that explanations generated via CoT are susceptible to *content biases* that negatively affect their *robustness* and *faithfulness* (Lyu et al., 2023; Turpin et al., 2024; Yee et al., 2024).

To mitigate these limitations and improve reasoning capabilities, recent works have proposed using logical formalisms (Lyu et al., 2023; Jiang et al., 2024a; Arakelyan et al., 2024). However, fully symbolic approaches possess the bottleneck of requiring a complete translation from natural language to formal languages, a process that negatively impacts efficiency and flexibility (Dinh et al., 2023; Quan et al., 2024b,a; Dalal et al., 2024).

This paper investigates methods to achieve a better trade-off between flexibility and robustness by disentangling content from logical reasoning without the need for a complete formalisation. In particular, we present *QuaSAR* (for Quasi-Symbolic Abstract Reasoning), a variation to CoT that guides LLMs to operate at a higher level of abstraction via *quasi-symbolic explanations*. Our framework leverages the capability of LLMs to formalise relevant variables and predicates, enabling the coexistence of symbolic elements with natural language.

Specifically, the aim of *QuaSAR* is to enable LLMs in tackling complex multi-step reasoning problems via the following steps: *(i) Abstraction*, where the problem is analysed and abstracted in terms of relevant symbolic predicates, variables, and constants; *(ii) Formalisation*, where the original problem is reformulated using a combination of a minimal symbolic form and natural language; *(iii) Explanation*, where the necessary steps to com-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17222–17240

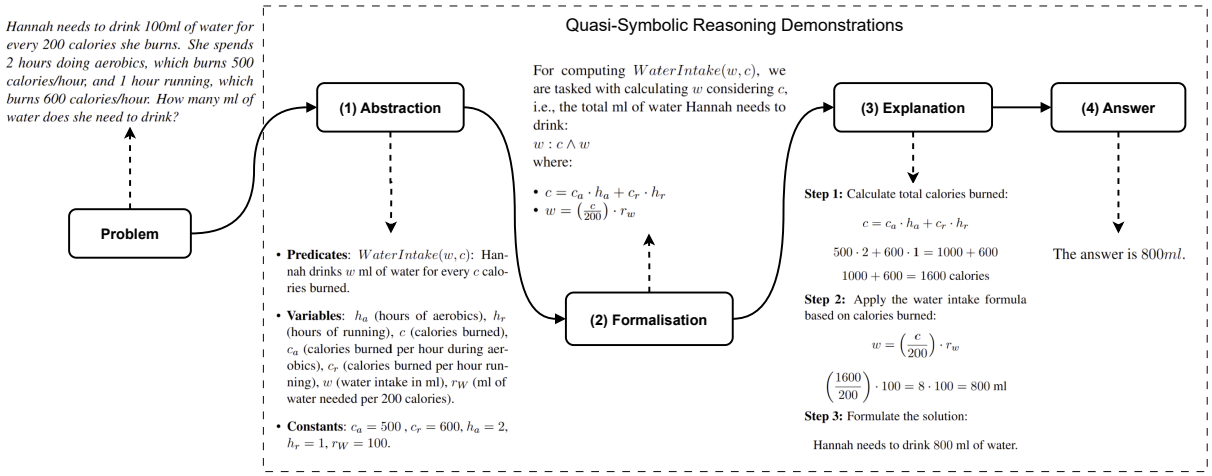

Figure 1: *QuaSAR* elicits quasi-symbolic abstractions in LLMs via the following steps: *(i) Abstraction*, where the problem is analysed and abstracted in terms of relevant symbolic predicates, variables, and constants; *(ii) Formalisation*, where the original problem is reformulated using a mixture of symbolic expressions and natural language; *(iii) Explanation*, where the necessary steps to compute the solution are formulated via quasi-symbolic reasoning chains; and *(iv) Answering*, where a final solution is generated. We use *QuaSAR* as an in-context learning strategy and for constructing reasoning demonstrations for smaller LLMs.

pute the solution are formulated via quasi-symbolic reasoning chains; and *(iv) Answering*, where a final solution is generated.

Building on recent work (Lyu et al., 2023; Jiang et al., 2024a), *QuaSAR* guides LLMs via structured instructions, going beyond the problems associated with using external solvers (Quan et al., 2024b). At the same time, in contrast to work using formal languages to guide CoT reasoning (Leang et al., 2024; Arakelyan et al., 2024), *QuaSAR* operates via a single prompting step, reducing costs, thereby delivering robust reasoning trajectories across different types of reasoning tasks.

We demonstrate the operability of *QuaSAR* in two different configurations – as an in-context approach to provide explicit instructions for larger and more capable LLMs and as a strategy for constructing synthetic demonstrations to improve the performance and align the reasoning capabilities of smaller LLMs. Hence, we perform an extensive empirical evaluation using different LLMs (i.e., GPT-4o, Llama3, and Qwen-2) on complex mathematical problems, reasoning, and natural language understanding tasks. *QuaSAR* demonstrates significant improvements by achieving an overall exact match boost on all proposed tasks.

In particular, our experiments led to the following findings and conclusions:

1. Formalising and structuring the LLMs' reasoning through quasi-symbolic trajectories enhances accuracy and verifiability, leading to an average increase in accuracy of 8% over CoT and 6.8% and 8.2% over CoMAT (Leang et al., 2024) and Faithful CoT (Lyu et al., 2023) respectively when applied on GPT-4o.

2. We found that our symbolic-inspired approach is significantly more efficient than related methods and can be employed on different tasks (i.e., mathematical and natural language reasoning tasks) without significant changes. Indeed, *QuaSAR* achieves state-of-the-art performance across diverse tasks of varying complexity and languages operating through the same framework.

3. By conducting an in-depth ablation study, we demonstrate the generalisability of *QuaSAR* and its effectiveness on different scales of LLMs. Our experiments show that *QuaSAR* provides more robust reasoning trajectories on tasks that are typically challenging for smaller-scale models, enhancing robustness and consistency on challenging adversarial variations on both natural language (i.e. MMLU-Redux) and symbolic reasoning tasks (i.e., GSM-Symbolic)

To the best of our knowledge, *QuaSAR* is the first method to apply quasi-symbolic demonstrations for a broad spectrum of reasoning tasks, demonstrating the impact of enabling the co-existence of symbolic abstractions and natural language explanations for improving the efficiency and robustness of LLMs.

2 *QuaSAR*: Quasi-Symbolic Abstract Reasoning

Integrating symbolic elements into natural language explanations is crucial for reasoning in disciplines such as mathematics and science, where symbolic abstractions facilitate the identification and generalisation of the logical connections between premises (i.e., explanans) and conclusions (i.e., explanandum) (Wang, 1954; Bronkhorst et al., 2019; Pennington and Hastie, 1993; Valentino and Freitas, 2024; Miller, 2019).

For example, within the unificationist account of explanation, Kitcher (1981) posits that explanations function by subsuming an apparently disconnected set of observations under the same underlying regularity, thereby forming recurring *argument patterns*. These patterns emerge when explanations are generalised through the replacement of concrete entities and predicates with abstract symbols. This process of *quasi-symbolic abstraction* enables explanatory arguments to be detached from specific world knowledge, thereby allowing their applicability across different problems (e.g., the same argument pattern created by the theory of gravity can be used to explain why specific objects fall and why celestial objects attract each other) (Valentino et al., 2021, 2022a,b; Zheng et al., 2024).

In this paper, we aim to explicitly leverage argument patterns with LLMs, hypothesising that quasi-symbolic abstractions can help disentangle concrete world knowledge from symbolic reasoning within a natural language explanatory framework and mitigate some of the challenges related to content effect. An example of this process is illustrated in Figure 1.

Formally, conventional in-context reasoning methods are structured as a triplet $(\mathcal{Q}, \mathcal{R}, \mathcal{A})$, where $\mathcal{Q}$ represents the question, $\mathcal{R}$ consists of in-context multi-step reasoning explanations (expressed in natural language or a related form), and $\mathcal{A}$ denotes the final answer. We extend this formalism by instructing the LLM to operate via explicit symbolic transformations as a core component of the reasoning process. Our framework, *QuaSAR*, structures the solution process as a quadruple $(\mathcal{Q}, \mathcal{S}, \mathcal{R}, \mathcal{A})$, where $\mathcal{S} = (s_1, s_2, s_3, s_4)$ represents a chain of instructions that guide the models to formalise relevant parts of the reasoning process. Each step s_i corresponds to a structured transformation aimed at decomposing the problem into a sequence of symbolically-elicited operations. This structured decomposition enhances transparency and facilitates systematic verification of each step.

2.1 *QuaSAR*'s Reasoning Process

A complex problem solution could be described by a sequence of inference steps determined by identifying and isolating the problem predicates and structuring a formalisation that facilitates reasoning to reach the final solution. Accordingly, *QuaSAR* operates using four steps that aim to improve the accuracy of the reasoning trajectory in LLMs: *(i) Abstraction*, where the problem is analysed and abstracted in terms of relevant symbolic predicates, variables, and constants; *(ii) Formalisation*, where the original problem is reformulated using a mixture of symbols and natural language; *(iii) Explanation* (§2.1.3), where the transformations are solved using quasi-symbolic representations that explicitly explain the solution; and *(iv) Answering* (§2.1.4), where a final solution is generated to address the problem. Appendix A reports *QuaSAR* prompt.

2.1.1 Abstraction

Abstracting the problem through the identification of relevant information is the first step in solving complex tasks and is a fundamental stage in structuring a robust formalisation (Bronkhorst et al., 2019). Therefore, as a first step, *QuaSAR* instructs the LLM to exemplify predicates, variables, and constants, whether of numerical or verbal types.

2.1.2 Formalisation

The crucial step of *QuaSAR* is the formalisation of the problem, which aids accurate reasoning by *translating* natural language into a semi-structured symbolic form. Hence, we instruct the LLM to deliver a quasi-formal representation of the problem, which is originally in natural language, using a structural-logical translation that explicitly represents the facts of the problem. This step is the basis for constructing an accurate reasoning trajectory because translating concrete terms in natural language into symbols aims to minimise ambiguities and content effects without compromising the components that may be significant for solving the problem.

2.1.3 Explanation

A significant component of CoT reasoning methods is breaking down the problem into a sequence of steps to arrive at the final solution. The explanation phase is based on step-by-step reasoning (Kojima et al., 2022) explicitly prompting the model

or delivering in-context demonstrations, generally natural language rationales. Then, the LLM is expected to solve the problem by providing a logical explanation that motivates the steps to the solution. In *QuaSAR*, the reasoning trajectory is based on the symbolic structure. In this way *QuaSAR* aims to elicit logical connections between each step, reducing the risk of errors caused by contextual knowledge or implicit symbolic-logical relations. The solution is then generated based on this quasi-symbolic reasoning process, which, although similar to the breakdown of reasoning methods, has a semi-structured formalisation standing behind it.

2.1.4 Answering

QuaSAR brings the reasoning trajectory to a final stage in which the LLM is instructed through a specific pattern –i.e., "*The answer is: [number]*". Although not fundamental, this stage is significant as it ensures that the reasoning constructed in the previous stages has a conclusion. Furthermore, this step facilitates the evaluation as it triggers the LLM to deliver a response that follows the pattern of the evaluation task.

2.2 *QuaSAR* Application

QuaSAR leverages a set of structured instructions to deliver step-wise explanations. Thus, the operability of *QuaSAR* is two-fold, as it can be used as both an in-context learning strategy and as a synthetic annotation method to support supervised learning (both described below).

2.2.1 *QuaSAR* for In-Context Learning

Using the step described in §2.1, we adopt *QuaSAR* to instruct three LLMs (i.e., GPT-4o, Llama-3-70B, and Qwen2-72B). Specifically, we instruct the models to exemplify and abstract the most important information from the given problem, formalising and translating natural language in a semi-structured logical form, explaining the solution in a step-wise manner, and finally generating the conclusive short-form answer in a strict format to have a more detailed and strict downstream evaluation. However, although the sequence of instructions is well-structured and defined, the ability to perform sequential and complex reasoning tasks is limited to larger LLMs (such as GPT-4-o, as discussed in the experiments). Hence, we transfer these capabilities to smaller models operating via *QuaSAR* for building synthetic reasoning demonstrations as training sets.

2.2.2 *QuaSAR* for Reasoning Demonstrations

We instruct smaller models via demonstrations produced by high-performing LLMs capable of following structured instructions. To filter for the quality of generated demonstrations, we follow the method proposed by Ranaldi et al. (2025b), which computes the citation precision for the considered documents as a proxy for the quality of the demonstrations. However, since *QuaSAR* employs a different annotation mechanism, our heuristics firstly filter out the final correct answers through a strict, exact match; then, behind the filtering (cutting off about 50% of the demonstrations), it verifies that each retrieved document along the reference evidence has been considered (a detailed description of the annotation phase is in Appendix B).

2.3 Training

We train a Language Model θ using the annotations[1] generated via *QuaSAR*. The annotations are augmented with reasoning demonstrations α using the standard language modelling objective, maximising likelihood:

$$\max_{\theta} \mathbb{E}_{(q,\alpha,y)\sim\mathcal{D}} \log p_\theta(Y \mid \alpha, Q) p_\theta(\alpha \mid Q) \quad (1)$$

where $\alpha = \alpha_1 \cdot \alpha_2 \cdot \alpha_3 \cdot \alpha_4$ is the combination of the step-wise reasoning trajectory delivered by the model, "$\cdot$" is the concatenation operator, and α_1, α_2, are the respective annotations generated by the above processes. Finally, Q is the provided question, and Y is the answer, including the intermediate steps and the final answer. $\mathcal{D}$ is the training corpus constructed using training demonstrations.

3 Experiments

We evaluate *QuaSAR* on complex mathematical problems, commonsense reasoning, and natural language understanding tasks (§3.1). We perform the evaluation phases by following standard approaches used to assess question-answering tasks (§3.2) on models presented in §3.3.

3.1 Tasks & Datasets

We evaluate the operability of *QuaSAR* on tasks involving complex reasoning and natural language inference. These tasks are best exemplified by the following categories:

[1] we select annotations as described in §2.2.2

Model	Symbolic					Natural Language	
	AQuA	GSM8K	SVAMP	MMLU-Redux	OlyBench	GPQA	DROP
GPT-4o	72.8	94.0	90.4	79.7	9.9	46.5	83.4
+ CoT	84.3	94.5	90.3	88.1	41.8	50.2	84.2
+ CoMAT (Leang et al., 2024)	83.5	93.7	-	88.3	40.4	-	-
+ FCoT (Lyu et al., 2023)	73.6	95.0	95.3	76.8	-	-	-
+ *QuaSAR*	**87.4**	**96.5**	**97.0**	**90.2**	**44.6**	**55.4**	**88.9**
Llama-3-70B	70.9	84.9	79.8	70.8	14.6	41.3	81.4
+ CoT	74.0	86.1	84.6	82.0	22.8	41.9	80.2
+ *QuaSAR*	**79.1**	**88.2**	**84.9**	**85.7**	**38.2**	**49.2**	**88.0**
Qwen2-72B	69.0	79.4	80.3	66.5	15.6	42.4	66.4
+ CoT	**78.8**	85.7	77.9	79.5	30.3	39.8	64.0
+ CoMAT (Leang et al., 2024)	72.4	83.9	-	81.7	32.2	-	-
+ *QuaSAR*	77.5	**86.2**	**84.3**	**83.5**	**36.2**	**48.2**	**69.0**

Table 1: Performance comparison using *QuaSAR* as in-context learning strategy (§2.2) across multiple tasks and models (§3). The results are obtained using zero-shot prompting as baselines, CoT (Kojima et al., 2022), CoMAT (Leang et al., 2024) and Faithful CoT (FCoT) (Lyu et al., 2023) as the main comparison.

Symbolic Tasks We use GSM8K (Cobbe et al., 2021), SVAMP (Patel et al., 2021), AQuA (Ling et al., 2017), MMLU-Redux (Gema et al., 2024) and Olympiad Bench (He et al., 2024) covering various mathematical topics, including abstract algebra, elementary, college-level and high-school mathematics. These datasets include multiple-choice questions (AQUA, MMLU-Redux) and math-world problems (GSM8K, MSVAMP, Olympiad Bench).

Natural Language Tasks We use Graduate-Level Google-Proof Q&A Benchmark (GPQA) (Rein et al., 2023) and Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs (DROP) (Dua et al., 2019). GPQA presents complex, open-ended questions that resist specific searches and require models to synthesise knowledge across multiple sources or reason critically to generate answers. DROP focuses on questions requiring discrete reasoning, such as arithmetic operations, logical comparisons, or event tracking, requiring the model to extract and manipulate information from a given passage.

3.2 Evaluation Metrics

We used exact-match for the multiple-choice question-answering task, requiring the predicted answer to match the correct one. This guarantees evaluation based on complete responses, addressing clarity concerns in tasks like MMLU-Redux. For string-matching answers, we used exact matches in GSM8K. Moreover, to have a comprehensive evaluation, we use GPT-4o-mini as a benchmark to evaluate how well the model's answers aligned with the ground truth. Details are described in Appendix C.

3.3 Models

Experiments were performed on GPT-4o (Achiam et al., 2023), Qwen2 (Yang et al., 2024) and Llama-3 (et al., 2024). While we selected the first two models to allow for a detailed comparison with related work and CoT frameworks, Llama-3 was chosen for its adaptability and the presence of releases with a small number of parameters that allow for additional tuning steps.

Baselines We compared *QuaSAR* against two baselines using the same greedy decoding strategy, fixing the temperature to 0. The baselines include: (1) standard zero-shot prompting, (2) CoT prompting (Kojima et al., 2022). Moreover, we include Faithful CoT (Lyu et al., 2023), FLAIRE (Arakelyan et al., 2024), and CoMAT (Leang et al., 2024) for additional comparison.

***QuaSAR* Application** We use *QuaSAR* as ICL and for generating tuning demonstrations. In both configurations, we instruct the models via the prompt in Appendix A. We conduct instruction-tuning of the models using the demonstrations described in Appendix B and the configurations in Appendix F.

4 Results & Discussions

The results in Tables 1 and 2 compare *QuaSAR* with baselines, CoT and related work across various tasks. *QuaSAR* outperforms CoT in most tasks

Model	Symbolic					Natural Language	
	AQuA	GSM8K	SVAMP	MMLU-Redux	OlyBench	GPQA	DROP
Llama-3-8B	65.2(67.3)	73.8(79.9)	70.0(73.8)	60.2(63.0)	10.9(13.2)	32.8(33.7)	58.4(60.2)
+ CoT	69.6(72.2)	80.4(82.6)	76.3(78.8)	64.5(65.9)	12.4(14.7)	34.0(35.2)	57.9(59.3)
+ FLARE (Arakelyan et al., 2024)	62.9	72.4	86.0	-	-	-	-
+ *QuaSAR*	67.2 (78.4)	77.2 (83.0)	77.3 (82.6)	63.0 (67.2)	13.0 (16.6)	33.1 (39.2)	58.7 (63.9)
Llama-3-1B	39.2(40.3)	44.8(45.8)	49.5(50.8)	28.3(30.1)	6.5(7.1)	25.4(26.9)	52.5(53.0)
+ CoT	50.7(52.0)	59.3(60.9)	58.2(59.9)	34.0(34.7)	8.2(10.6)	27.6(28.7)	54.4(55.0)
+ *QuaSAR*	51.6 (55.4)	58.1 (62.8)	60.4 (64.5)	34.2 (40.0)	9.8 (14.6)	26.6 (29.4)	54.1 (57.2)
Qwen2-7B	62.9(63.7)	70.4(71.6)	66.9(67.2)	65.5(66.3)	10.5(10.9)	32.0(32.7)	55.3(54.2)
+ CoT	79.1(80.3)	82.8(83.6)	73.2(74.9)	79.2(80.0)	9.8(10.7)	33.7(34.0)	56.0(56.8)
+ CoMAT (Leang et al., 2024)	72.4	83.9	-	79.8	32.2	-	-
+ *QuaSAR*	72.6 (78.3)	81.7 (85.6)	69.2 (75.0)	75.9 (80.3)	27.8 (36.5)	29.5 (35.2)	54.6 (60.0)
Qwen2-1.5B	56.8(57.2)	61.4(62.0)	59.2(60.0)	41.7(42.4)	6.9(7.4)	21.4(21.9)	49.8(50.8)
+ CoT	58.7(59.9)	64.7(65.8)	63.6(65.0)	46.3(47.8)	7.8(9.1)	25.4(26.9)	51.2(52.5)
+ *QuaSAR*	57.6 (62.2)	64.2 (69.8)	65.4 (70.2)	44.8 (49.5)	8.2 (11.8)	26.6 (31.0)	50.8 (57.3)

Table 2: Performance comparison using *QuaSAR*, CoT (Kojima et al., 2022), FLARE (Arakelyan et al., 2024) and, CoMAT (Leang et al., 2024)as in-context learning strategies. Moreover, we report in brackets the performances obtained using these strategies as annotation approaches for tuning models (complete table in Appendix K).

requiring advanced mathematical reasoning (Symbolic task), reading comprehension and logical reasoning (Natural Language task). In particular, two different results emerge in the application of *QuaSAR*: when it is employed as in-context learning strategy in higher-scale models, it consistently outperforms other strategies; when *QuaSAR* is employed in smaller-scale models, it does not obtain the same benefits, as discussed in §4.1. On the other hand, when *QuaSAR* is used as an annotation strategy for delivering demonstrations operated to refine smaller-scale models, the performances are significantly higher compared to the models instructed via standard CoT demonstrations, as examined in §4.2.

Overall, our experiments demonstrate the benefit of quasi-symbolic abstractions for complex reasoning tasks, and provide evidence of improved robustness on challenging adversarial variations (§4.3).

4.1 *QuaSAR* as In-Context Learning Strategy

Table 1 reports the results of *QuaSAR* when adopted as an In-Context Learning (ICL) strategy. We observe general robust improvement over the baseline models (with an improvement of 19.1% for GPT-4o, 11.8% for Llama-3-70B and 17.2% for Qwen2-72B); the results show that the role of *QuaSAR* as ICL is foremost noticeable for higher-scale LLMs. *QuaSAR* consistently outperforms CoT, Faithful CoT and CoMAT. *QuaSAR* also delivers overall improvements on smaller-scale models. Table 2 shows an improvement over the baseline of 5.2% for Llama-3-8B, 13.4% for Llama-3-1B, 10.5% for Qwen2-7B and 8.3% for Qwen2-1.5B. However, comparing *QuaSAR* to CoT on smaller models, we observe a decrease in performance, indicating that such models fail to follow the quasi-symbolic reasoning process induced by *QuaSAR*.

4.2 *QuaSAR* as Annotation Strategy

Table 2 (values between the brackets and detailed in Appendix K) reports the results of *QuaSAR* when adopted as annotation strategy for different models. From the results, it clearly emerges that *QuaSAR* is consistently effective in enhancing the performance of Llama and Qwen2 models when used to generate reasoning demonstrations via GPT-4o. In particular, we found that *QuaSAR* outperforms other tuning approaches, including baseline SFT on target answers and SFT on demonstrations delivered via CoT.

4.3 The impact of *QuaSAR*

The step-wise reasoning chain generations elicited by *QuaSAR* have an optimal impact on the downstream performances when *QuaSAR* is used as ICL and for generating demonstrations.

Step-wise roles for ICL Table 3 displays the difference in accuracy compared to *QuaSAR* with all steps. We show that each step impacts *QuaSAR*'s operability. In particular, eliminating step 1 (i.e., *w/o(1)*) affects the final accuracies (-1.8 on average). This suggests that the initial abstraction step

Task	w/o(1)	w/o(2)	w/o(3)	w/o(4)	w/o(1-2)	w/o(3-4)
AQuA	**-1.9**	**-3.6**	**-3.7**	**-2.9**	**-3.7**	-2.7
GSM8K	**-2.1**	**-4.2**	**-3.9**	-1.3	**-3.5**	-2.6
MMLU-R	-0.7	-3.0	-3.2	**-3.2**	-2.2	**-2.8**
OlyBench	**-1.9**	-3.1	**-3.7**	-2.1	**-3.8**	-2.3
GPTQ	-1.6	**-3.9**	**-3.6**	**-3.2**	**-4.1**	**-2.9**
Avg	-1.8	-3.5	-3.4	-2.5	-3.2	-2.8

Table 3: Performance change without (w/o) *QuaSAR* step obtained from GPT-4o. *(Bold values over the average).

is important for final performance but is not decisive, especially in tasks such as GPTQ and MMLU-Redux. In contrast, steps 2 (i.e., formalisation) and 3 (i.e., explanation), play a crucial role, indeed, a stable drop of more than 3.5 points can be observed. In this case, the tasks that suffer the most are the mathematical subset (AQuA and GSM8K). Step 4, reserved for the strict generation of the final answer, is more decisive in multiple-choice than in mathematical tasks. Finally, by eliminating pairs of steps (i.e. *w/o(1-2)* and *w/o(3-4)*), it can be seen that there are significant drops in both mathematical tasks (see GSM8K, AQuA and OlympicBench) and language-related reasoning tasks (see MMLU-Redux and GPTQ). The combination of the four steps from these results positively impacts reasoning capabilities, and they all contribute significantly to the final performances.

Step-wise role for Annotations Figure 2 displays the difference in accuracy using entire *QuaSAR* for generating demonstrations. As in the case of ICL, the steps in the demonstrations have specific importance for instructing models. Indeed, it can be observed that the instructed models perform worse by eliminating central steps such as Step 2 and Step 3. In contrast, removing step 4 duplicated to the response has moderate adverse effects (performance drop of no more than two points). Finally, it can be seen that the order in which the steps are delivered in the demonstrations also has a positive impact. Delivering the demonstrations randomly shuffled negatively impacts performances, dropping around 4 average points.

4.4 Robustness & Ablation Analysis

In-context Robustness To assess the robustness of *QuaSAR* as an ICL strategy, we evaluated two different phenomena: *(i)* the order swapping of choices in MMLU-Redux (Gema et al., 2024) as

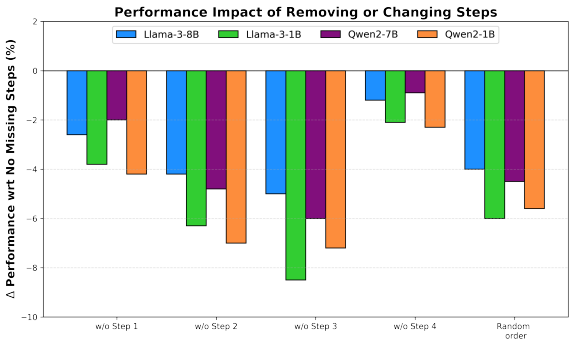

Figure 2: Performance differences (Δ) for each tuned model. We analyse the impact of each component on tuning by eliminating (w/o) or random shuffling the four *QuaSAR* steps.

	Task	Baseline	CoT	*QuaSAR*
GPT-4o	MMLU-Redux	79.8	88.2	**90.3**
	-choices shuffled	78.6(**-1.2**)	86.8(**-1.2**)	**90.3** (0.0)
	GSM-Symbolic	94.0	95.5	96.5
	-2nd choice	89.7(**-4.3**)	90.8(**-4.7**)	95.3 (-1.2)
Llama3-70B	MMLU-Redux	70.8	81.9	85.0
	-choices shuffled	68.7(**-0.9**)	81.0(**-0.9**)	**84.8** (-0.2)
	GSM-Symbolic	84.2	85.3	87.8
	-2nd choice	82.6 (**-1.6**)	83.7(**-1.6**)	87.2 (-0.4)
Llama3-8B	MMLU-Redux	30.2	31.6	37.6
	-choices shuffled	27.0(**-3.2**)	30.4(**-1.2**)	37.3 (-0.3)
	GSM-Symbolic	46.7	60.2	65.3
	-2nd choice	44.9(**-1.8**)	58.4(**-1.8**)	64.8 (-0.5)

Table 4: Performance obtained by changing the order of choices randomly (MMLU-Redux) and using a perturbed version of mathematical tasks (GSM-Symbolic). *(accuracies differences in brackets)

proposed by Leang et al. (2024) and *(ii)* the performance on a more complex version of GSM8K designed to test robustness to superficial variations (i.e., GSM-Symbolic (Mirzadeh et al., 2024)). Table 4 shows that *QuaSAR* consistently achieves the same performances with considerably less variation than CoT. This indicates the positive impact of quasi-symbolic abstractions on the robustness of the models.

Training Efficiency Figure 3 shows the performance of tuned models on GSM-Symbolic (Mirzadeh et al., 2024) using *QuaSAR*, CoT, and baseline demonstrations as the number of training examples increases. While the number of demonstrations in *QuaSAR* plays a significant role in shaping the final performance, our findings reveal that models trained with *QuaSAR* demonstrations surpass those trained with CoT demonstrations consistently. Moreover, it is possible to observe that

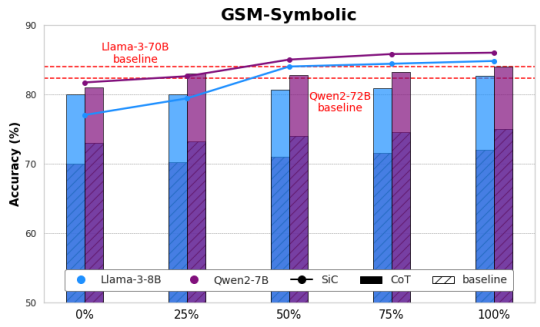

Figure 3: Performances using *QuaSAR* as demonstration tuning by scaling training data. We replicated experimental settings proposed in §3, changing the number of tuning instructions. *Appendix H reports additional evaluaitons.

models instructed via *QuaSAR* demonstrations outperform their respective models with more parameters with 50% of the training examples in the case of Llama-3-8B and 25% in the case of Qwen2-7B. Notably, *QuaSAR* consistently outperforms other supervised training approaches even with this reduced dataset size, emphasising the superior quality of the training signal provided by *QuaSAR* demonstrations.

Additional Analysis Finally, we produced further analyses to investigate the error flow, the degree of self-correction and the transferability of the approach. In the first analysis, we showed the error rate of our *QuaSAR*. In Appendix J, we present a particular analysis to investigate the error rate of each step, arguing for its validity.

In the second analysis, we studied the self-correction capability of the models trained and tuned via *QuaSAR* and via CoT. Appedinx L reports the results of the self-assessment on incorrect generations, showing that the outputs generated via *QuaSAR* are actually easier to correct, as the output rationale, being structured, is simpler to correct.

In the final analysis, reported in Appendix E, we showed the elasticity and adaptability of *QuaSAR* in tasks different from those proposed in the main analysis, confirming the results obtained and discussed in the previous sections.

5 Related Work

Logical Reasoning Logical reasoning tasks require the capability to process complex logical structures (Cummins et al., 1991). Traditional methodologies contain rule-based systems (Robinson, 1965) and neural network-based paradigms (Amayuelas et al., 2022; Gerasimova et al., 2023) for solving and manipulating symbolic representations. Recent advancements introduced hybrid frameworks (Pan et al., 2023; Ye et al., 2024; Jiang et al., 2024a), which integrate large language models (LLMs) into symbolic reasoning pipelines (Quan et al., 2024b). These frameworks operate LLMs to map natural language inputs into symbolic syntax, subsequently processed by external reasoning tools. This integration improves reasoning performance through techniques such as self-consistency (Wang et al., 2023; Zhang et al., 2022). Nevertheless, these frameworks commonly depend on external tools predicated on the assumption that LLMs lack the reliability to parse symbolic expressions with the precision of rule-based reasoning systems alone.

Symbolic Reasoning Symbolic reasoning integrates natural language (NL) and symbolic language (SL) to decompose complex queries into sub-problems solved by SL programs and deterministic solvers, ensuring interpretability and precision (Lyu et al., 2023). Recent efforts have leveraged LLMs to decrease dependence on SL programs (Xu et al., 2024), but these approaches primarily address logical reasoning and depend on verifiers for accuracy, limiting their applicability to complex mathematical tasks.

On the other side, Chain-of-Thought (CoT) strategies have demonstrated significant performance improvements in mathematical symbolic reasoning (Jiang et al., 2024c), reinforced by advancements in problem understanding (Zhong et al., 2024), structured formats (Tam et al., 2024), and supervision models (Ranaldi and Freitas, 2024a,b; Jiang et al., 2024b). Further, premise selection and symbolic frameworks have facilitated systematic evaluations across logical and mathematical reasoning (Meadows et al., 2023; Ferreira and Freitas, 2020).

6 Future Works

In future developments, we plan to extend our contribution to non-English languages to broaden the beneficial impacts and operability of reasoning for multilingual alignment. To this end, we will use our approach in the multilingual extension of GSM-Symbolic (Mirzadeh et al., 2024) proposed by Ranaldi and Pucci (2025). Furthermore, we would like to investigate the extent to which our framework can be applied in scenarios where

retrieval-augmented LLMs approaches are used, such as our parallel contributions, where we propose techniques to resolve knowledge conflicts in retrieved documents (Ranaldi et al., 2025a,b).

7 Conclusion

Complex reasoning tasks often require the coexistence of natural language and symbolic abstractions. Many existing methods based on CoT struggle to ensure consistency and robustness, particularly when handling tasks with shuffled answer options or superficial lexical variations. In this paper, we proposed Quasi-Symbolic Abstract Reasoning (*QuaSAR*) to address these challenges. This simple yet powerful framework enables LLMs to tackle such tasks by breaking them down into systematic, quasi-symbolic step-by-step reasoning. By employing *QuaSAR* as an in-context learning strategy and a tool for constructing demonstrations, we improved the performance of smaller models and provided a comprehensive analysis across diverse benchmarks. Our experiments demonstrate that *QuaSAR* surpasses traditional CoT reasoning methods by delivering transparent and consistent reasoning trajectories. *QuaSAR* excels across tasks of varying complexity, achieving state-of-the-art performance and improving robustness. *QuaSAR* delivers a scalable and effective solution for complex reasoning, enhancing faithfulness, verifiability, and reliability while outperforming conventional Chain-of-Thought approaches.

Acknowledgements

This work was funded by the Swiss National Science Foundation (SNSF) project "NeuMath" (200021_204617), Innosuisse project "SINFONIA" (n. 104.170 IP-ICT), by the CRUK National Biomarker Centre, and supported by the Manchester Experimental Cancer Medicine Centre, the NIHR Manchester Biomedical Research Centre and UK Research and Innovation under the UK government's Horizon Europe funding guarantee grant number 10039436.

References

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

Aaron Grattafiori et al. 2024. The llama 3 herd of models.

Alfonso Amayuelas, Shuai Zhang, Xi Susie Rao, and Ce Zhang. 2022. Neural methods for logical reasoning over knowledge graphs. In *International Conference on Learning Representations*.

Erik Arakelyan, Pasquale Minervini, Pat Verga, Patrick Lewis, and Isabelle Augenstein. 2024. Flare: Faithful logic-aided reasoning and exploration.

Hugo Bronkhorst, Gerrit Roorda, Cor Suhre, and Martin Goedhart. 2019. Logical reasoning in formal and everyday reasoning tasks. *International Journal of Science and Mathematics Education*, 18(8):1673–1694.

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

Denise D Cummins, Todd Lubart, Olaf Alksnis, and Robert Rist. 1991. Conditional reasoning and causation. *Memory & cognition*, 19:274–282.

Dhairya Dalal, Marco Valentino, Andre Freitas, and Paul Buitelaar. 2024. Inference to the best explanation in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 217–235, Bangkok, Thailand. Association for Computational Linguistics.

Tuan Dinh, Jinman Zhao, Samson Tan, Renato Negrinho, Leonard Lausen, Sheng Zha, and George Karypis. 2023. Large language models of code fail at completing code with potential bugs.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Deborah Ferreira and André Freitas. 2020. Premise selection in natural language mathematical texts. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7365–7374.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al.

2024. Are we done with MMLU? *arXiv preprint arXiv:2406.04127*.

Olga Gerasimova, Nikita Severin, and Ilya Makarov. 2023. Comparative analysis of logic reasoning and graph neural networks for ontology-mediated query answering with a covering axiom. *IEEE Access*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Dongwei Jiang, Marcio Fonseca, and Shay B Cohen. 2024a. Leanreasoner: Boosting complex logical reasoning with lean. *arXiv preprint arXiv:2403.13312*.

Dongwei Jiang, Guoxuan Wang, Yining Lu, Andrew Wang, Jingyu Zhang, Chuyu Liu, Benjamin Van Durme, and Daniel Khashabi. 2024b. Rationalyst: Pre-training process-supervision for improving reasoning.

Zhuoxuan Jiang, Haoyuan Peng, Shanshan Feng, Fan Li, and Dongsheng Li. 2024c. LLMs can find mathematical reasoning mistakes by pedagogical chain-of-thought. *arXiv preprint arXiv:2405.06705*.

Philip Kitcher. 1981. Explanatory unification. *Philosophy of science*, 48(4):507–531.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Joshua Ong Jun Leang, Aryo Pradipta Gema, and Shay B. Cohen. 2024. Comat: Chain of mathematically annotated thought improves mathematical reasoning.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, et al. 2024. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329, Nusa Dua, Bali. Association for Computational Linguistics.

Jordan Meadows and André Freitas. 2023. Introduction to Mathematical Language Processing: Informal Proofs, Word Problems, and Supporting Tasks. *Transactions of the Association for Computational Linguistics*, 11:1162–1184.

Jordan Meadows, Marco Valentino, Damien Teney, and Andre Freitas. 2023. A symbolic framework for systematic evaluation of mathematical reasoning with transformers.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Nancy Pennington and Reid Hastie. 1993. Reasoning in explanation-based decision making. *Cognition*, 49(1):123–163.

Xin Quan, Marco Valentino, Louise Dennis, and Andre Freitas. 2024a. Enhancing ethical explanations of large language models through iterative symbolic refinement. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–22, St. Julian's, Malta. Association for Computational Linguistics.

Xin Quan, Marco Valentino, Louise A. Dennis, and André Freitas. 2024b. Verification and refinement of natural language explanations through llm-symbolic theorem proving.

Leonardo Ranaldi and Andre Freitas. 2024a. Aligning large and small language models via chain-of-thought reasoning. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1812–1827.

Leonardo Ranaldi and Andre Freitas. 2024b. Self-refine instruction-tuning for aligning reasoning in language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2325–2347, Miami, Florida, USA. Association for Computational Linguistics.

Leonardo Ranaldi and Giulia Pucci. 2025. Multilingual reasoning via self-training. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11566–11582, Albuquerque, New Mexico. Association for Computational Linguistics.

Leonardo Ranaldi, Giulia Pucci, Barry Haddow, and Alexandra Birch. 2024a. Empowering multi-step reasoning across languages via program-aided language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12171–12187, Miami, Florida, USA. Association for Computational Linguistics.

Leonardo Ranaldi, Giulia Pucci, Federico Ranaldi, Elena Sofia Ruzzetti, and Fabio Massimo Zanzotto. 2024b. A tree-of-thoughts to broaden multi-step reasoning across languages. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1229–1241, Mexico City, Mexico. Association for Computational Linguistics.

Leonardo Ranaldi, Federico Ranaldi, Fabio Massimo Zanzotto, Barry Haddow, and Alexandra Birch. 2025a. Improving multilingual retrieval-augmented language models through dialectic reasoning argumentations.

Leonardo Ranaldi, Marco Valentino, and Andre Freitas. 2025b. Eliciting critical reasoning in retrieval-augmented generation via contrastive explanations. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11168–11183, Albuquerque, New Mexico. Association for Computational Linguistics.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark.

John Alan Robinson. 1965. A machine-oriented logic based on the resolution principle. *Journal of the ACM (JACM)*, 12(1):23–41.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let me speak freely? a study on the impact of format restrictions on performance of large language models. *arXiv preprint arXiv:2408.02442*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2024. Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36.

Marco Valentino and André Freitas. 2024. On the nature of explanation: An epistemological-linguistic perspective for explanation-based natural language inference. *Philosophy & Technology*, 37(3):88.

Marco Valentino, Mokanarangan Thayaparan, Deborah Ferreira, and André Freitas. 2022a. Hybrid autoregressive inference for scalable multi-hop explanation regeneration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11403–11411.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2021. Unification-based reconstruction of multi-hop explanations for science questions. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 200–211, Online. Association for Computational Linguistics.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2022b. Case-based abductive natural language inference. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1556–1568, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Hao Wang. 1954. The formalization of mathematics. *Journal of Symbolic Logic*, 19(4):241–266.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. Faithful logical reasoning via symbolic chain-of-thought. *arXiv preprint arXiv:2405.18357*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2024. Satlm: Satisfiability-aided language models using declarative prompting. *Advances in Neural Information Processing Systems*, 36.

Evelyn Yee, Alice Li, Chenyu Tang, Yeon Ho Jung, Ramamohan Paturi, and Leon Bergen. 2024. Dissociation of faithful and unfaithful reasoning in LLMs. *arXiv preprint arXiv:2405.15092*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*.

Qihuang Zhong, Kang Wang, Ziyang Xu, Juhua Liu, Liang Ding, Bo Du, and Dacheng Tao. 2024. Achieving> 97% on gsm8k: Deeply understanding the problems makes LLMs perfect reasoners. *arXiv preprint arXiv:2404.14963*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models.

A *QuaSAR* Prompting Template

#Role
You are an experienced expert skilled in answering complex problems through logical reasoning and structured analysis.

#Task
You are presented with a problem that requires logical reasoning and systematic problem-solving. Please answer the question following these steps rigorously.

#Steps

1) Please consider the following question and exemplify the relevant predicates, variables, and constants. Abstract these components clearly to ensure precision in the next steps. *Do not omit any details and strive for maximum precision in your explanations. Refer to this step as Abstraction (s_1)*

2) For each predicate, variable and constant defined in s_1, translate the question in formal symbolic representation. Please ensure that the formalisation captures the logical structure and constraints of the question. *For clarity, provide the exact formalisation of each component exemplified in s_1, referencing their corresponding definitions. Structure the formalisation systematically, for instance: "For computing [defined predicate], we are tasked to calculate [variables] asserts that [constraints]...". Refer to this step as Formalisation (s_2)*

3) Please consider the formalisation in s_2 in detail, ensure this is correct and solve the question by breaking down the steps operating a symbolic representation. Combine variables, constants, and logical rules systematically at each step to find the solution. *For clarity, provide clear reasoning for each step. Structure the explanation systematically, for instance: "Step 1: Calculate... Step 2:.....". Refer to this step as Explaination (s_3)*

4) In conclusion, behind explaining the steps supporting the final answer to facilitate the final evaluation, extract the answer in a short and concise format by marking it as "**The answer is** " *At this stage be strict and concise and refer to this step as Answering (s_4).*

#Question
{`question`}

Table 5: The Step-wise Instruction Chain (*QuaSAR*) framework instructs the model to deliver step-wise reasoning paths that lead the models to solve the task by delivering a formalised strict final answer.

B Annotations Pipeline

As introduced in §2, we use our Step-wise Instruction Chain (*QuaSAR*) to lead Llama-3-1B, -8B, Qwen2-7B and 1B in solving complex tasks by breaking down the solution using the *reasoning process* described in §2.3. Since *QuaSAR* alone does not fully leverage the capabilities of the baseline models—significantly smaller models without further tuning, as shown in Table 2—we use GPT-4o (GPT-4) as an annotation model. GPT-4 is systematically prompted using the instructions detailed in Appendix A.

GPT-4 is used to generate synthetic demonstrations to train models in delivering *QuaSAR*'s step-wise reasoning methods. However, while GPT-4 follows the instructions exhaustively, its outputs may include errors or misleading information. To address this, we evaluated the quality of the generated demonstrations, filtering out inaccurate examples to refine the instruction set. Specifically, we removed all incorrect answers (i.e., outputs that do not match the exact target string metric, referred to as *exact-match*). Finally, we verified that all essential steps were correctly encoded in the remaining demonstrations using GPT-4o-mini and the prompt in Appendix C

C Evaluation Metrics

We used a double-check to assess the accuracy of the responses delivered in the different experiments. In the first step, we used an exact-match heuristic (this was used for most of the evaluations, especially in cases of multiple-choice QA). However, since some experiments required a more accurate response check, we used GPT-4o as a judge. Hence, we prompt the model as follows:

#Role:
You are an experienced expert skilled in answering complex problems through logical reasoning and structured analysis.
#Task:
Given the following "#Sentences", you are a decider that decides whether the "Generated Answer" is the same as the "Target Answer". If the output doesn't align with the correct answer, respond with '0', whereas if it's correct, then respond with '1'. *Please, do not provide any other answer beyond '0' or '1'.*
#Senteces:
Generated Answer: {model_result}
Target Answer: {correct_answer}.

D Data Composition

We evaluated *QuaSAR* using the tasks introduced in §3.1. Although these tasks are most often used to assess the performance of LLMs, they often do not have dedicated sets for evaluation and training. Therefore, to use *QuaSAR* both as an in-context prompting approach and as an instruction generation approach, we divided the datasets into training and testing. Table 7 shows the instances of each dataset in training and testing. Where we did not find split data already, we produced a splitting, which is also displayed in Table 7.

Task	Total	Test	Trainig Set	Testing Set
AQuA	254	254	Yes	254
GSM8K	$8,02k$	$1,32k$	Yes	$1,32k$
SVAMP	700	700	Yes	700
MMLU-Redux	$1k$	$1k$	No	$1k$
OlyBench	$2,5k$	$1,5k$	Yes	500
GPQA	198	-	No	198
DROP	$2,5k$	$1,5k$	Yes	500

Table 6: Data used to evaluate *QuaSAR* as in-context learning approach. When training set are present we tagged as "Yes". *($1k$ is equal to 1000).

Task	Total	Correct	Used
AQuA	$97k$	$3.0k$	$1.0k$
GSM8K	$6k$	$2.04k$	$0.8k$
OlyBench	420	250	250
DROP	$7,5k$	$1k$	350
Total	$22k$	$6,9k$	***2,4k***

Table 7: Data used to construct *QuaSAR* demonstrations. We applied the annotation (§2.2.2) and obtained the following answers, filtered according to the heuristics in Appendix B, and balanced for the tasks.

E Additional Task

Method	MATH	XCOPA	MGSM
baseline	70.4	84.4	90.5
CoT	76.8	88.6	91.0
QuaSAR	**79.5**	**89.2**	**93.4**

Table 8: GPT-4o performances on MATH, XCOPA, and MGSM.

Method	MATH	XCOPA	MGSM
baseline	30.0	56.4	59.0
CoT	33.0	56.9	60.8
QuaSAR	**36.4**	**65.0**	**66.9**

Table 9: Llama-3-8B performances on MATH, XCOPA, and MGSM.

F Training Setup

To evaluate the impact of *QuaSAR* demonstrations on smaller models (§2), we use the annotations produced following the *QuaSAR* strategy (§2.2.2). For a fair comparison, we generated CoT annotations and naive output without any prompting approach using GPT-4 on the same instances. Then, we train selected models using *QuaSAR*, CoT and standard output demonstrations. We fine-tuned the Llama-3 models for 3 epochs with a batch size of 32 and a learning rate equal to 3e-5 with a 0.001 weight decay and the Qwen2 models for the same epochs and batch size. Instead, a learning rate equal to 2e-5 with a 0.002 weight decay was used.

G Models Vesions

Model	Version
Llama-3-70B	meta-llama/Meta-Llama-3-70B-Instruct
Llama-3.1-8B	meta-llama/Meta-Llama-3-8B-Instruct
Llama-3.2-1B	meta-llama/Llama-3.2-1B-Instruct
Qwen2-72B	Qwen/Qwen2-72B-Instruct
Qwen2-7B	Qwen/Qwen2-7B-Instruct
Qwen2.5-1.5B	Qwen/Qwen2.5-1.5B-Instruct
GPT-4-o	OpenAI API (gpt-4o-2024-08-06)
GPT-4-o-mini	OpenAI API (gpt-4o-mini-2024-07-18)

Table 10: List of the versions of the models proposed in this work, which can be found on huggingface.co. We used the configurations described in Appendix I in the repositories for each model *(access to the following models was verified on 12 Jan 2024).

H Evaluation Scaling training Data

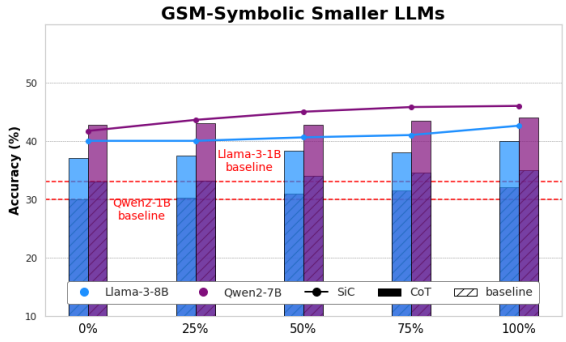

Table 11: Performance assessment using *QuaSAR* as demonstration tuning by scaling training data. We replicated experimental settings proposed in §3 changing the number of tuning instructions.

I Model and Hyperparameters

As introduced in §3.3, we propose different LLMs: (i) GPT-4o; (ii) three models from the Llama-3 family (et al., 2024): Llama3-70B, Llama3.1-8B, Llama3.2-1B; (iii) three models of the Qwen2 family (Yang et al., 2024): Qwen2-72B, Qwen2-7B and -1B. GPT-4 is used via API, while for the others, we used versions detailed in Table 10. As discussed in the limitations, our choices are related to reproducibility and the cost associated with non-open-source models. The generation temperature used varies from $\tau = 0$ of GPT models to $\tau = 0.5$ of Llama models. We choose these temperatures for (mostly) deterministic outputs, with a maximum token length of 3500. The other parameters are left unchanged as recommended by the official resources. The code and the dataset will be publicly released upon acceptance of the paper.

J Error Propagation

We provide a detailed analysis of error propagation across the four passages proposed in §2. We are quantifying the error rates attributed to each subcomponent, recognising that every stage performs a distinct function. The analysis was conducted on an ablation subset of GSM-Symbolic using GPT-4o. The error rate for each step was independently assessed via a manual verification process. The total failure rate across the full pipeline is **36%**.

Stage	Description	Error Rate
Abstraction	Identification of relevant predicates, variables, constants	8%
Formalisation	Translation of input into symbolic/semi-symbolic structures	12%
Explanation	Step-by-step reasoning using the structured representation	16%
Answering	Generation of the final solution	6%

Step	Isolated Error	Cumulative Error
Abstraction	8%	8%
Formalisation	12%	17%
Explanation	16%	32%
Answering	6%	36%

Table 12: Error rates per stage and cumulative error analysis. Each stage contributes independently and sequentially to the overall error rate.

K Complete Results Smaller LLMs

Model	Symbolic					Natural Language	
	AQuA	GSM8K	SVAMP	MMLU-Redux	OlyBench	GPQA	DROP
Llama-3-8B	65.2	73.8	70.0	60.2	10.9	32.8	58.4
Llama-3-1B$_{SFT}$	68.3	74.9	73.8	63.0	13.2	33.7	60.2
+ CoT$_{ICL}$	69.6	80.4	76.3	64.5	12.4	34.0	57.9
+ CoT$_{SFT}$	73.2	82.6	78.8	65.9	14.7	35.2	59.3
+ FLARE (Arakelyan et al., 2024)	62.9	72.4	86.0	-	-	-	-
+ *QuaSAR* $_{ICL}$	67.2	77.2	75.6	62.0	13.4	33.0	58.7
+ *QuaSAR* $_{SFT}$	74.8	**83.0**	82.6	67.2	17.6	**39.2**	63.6
+ *QuaSAR* $_{SFT+ICL}$	**75.2**	82.8	84.7	**68.0**	**17.8**	**39.2**	**63.9**
Llama-3-1B	39.2	44.8	49.5	28.3	6.5	25.4	52.5
Llama-3-1B$_{SFT}$	40.3	45.8	50.8	30.1	7.1	26.9	53.0
+ CoT$_{ICL}$	50.7	59.3	58.2	34.0	8.2	27.6	54.4
+ CoT$_{SFT}$	52.0	60.9	59.9	34.7	8.8	28.7	55.0
+ *QuaSAR* $_{ICL}$	51.6	58.1	60.4	30.2	10.6	26.6	54.1
+ *QuaSAR* $_{SFT}$	55.4	**62.8**	64.5	40.0	14.0	**29.4**	57.2
+ *QuaSAR* $_{SFT+ICL}$	**56.0**	**62.8**	**64.9**	**40.8**	**14.6**	29.3	**57.7**
Qwen2-7B	62.9	70.4	66.9	65.5	10.5	32.0	55.3
Qwen2-7B$_{SFT}$	63.7	71.6	67.2	66.3	10.9	32.7	56.2
+ CoT$_{ICL}$	79.1	82.8	73.2	79.2	9.8	33.7	56.0
+ CoT$_{SFT}$	80.3	83.6	74.9	80.0	11.7	35.0	56.8
+ CoMAT (Leang et al., 2024)	72.4	83.9	-	79.8	32.2	-	-
+ *QuaSAR* $_{ICL}$	72.6	81.7	69.2	75.9	27.8	29.5	54.6
+ *QuaSAR* $_{SFT}$	78.3	**85.6**	75.0	80.3	**35.6**	35.2	**60.0**
+ *QuaSAR* $_{SFT+ICL}$	**79.0**	**85.6**	**75.4**	**80.7**	**35.6**	**35.8**	**60.0**
Qwen2-1.5B	56.8	61.4	59.2	41.7	6.9	21.4	49.8
Qwen2-1.5B$_{SFT}$	57.2	62.0	60.0	42.4	7.4	21.9	50.8
+ CoT$_{ICL}$	58.7	64.7	63.6	46.3	7.8	25.4	51.2
+ CoT$_{SFT}$	59.9	65.8	65.0	47.8	9.1	26.9	52.5
+ *QuaSAR* $_{ICL}$	57.6	64.2	65.4	44.8	8.2	26.6	50.8
+ *QuaSAR* $_{SFT}$	**62.2**	69.8	70.2	**49.5**	11.8	31.0	57.3
+ *QuaSAR* $_{SFT+ICL}$	**62.2**	**70.9**	**71.1**	**49.5**	**12.4**	**31.5**	**57.6**

Table 13: Performance comparison using *QuaSAR*, CoT (Kojima et al., 2022), FLARE (Arakelyan et al., 2024) and, CoMAT (Leang et al., 2024)as in-context learning strategy (denoted as ICL), annotation strategy for delivering demonstration to supervised fine-tune LLMs (denoted as SFT) and in-context learning plus tuning (denoted as $ICL + SFT$).

L Self-correction Evaluation

Generator	Task	Evaluator			
		GPT-4o	Llama-3-70B	Llama-3-8B	Llama-3-1B
GPT-4o	CoT	98%	94%	83%	75%
	QuaSAR	98%	98%	88%	81%
Llama-3-70B	CoT	98%	89%	84%	74%
	QuaSAR	98%	92%	86%	80%
Llama-3-8B	CoT	100%	82%	74%	54%
	QuaSAR	100%	87%	80%	68%
Llama-3-1B	CoT	97%	87%	80%	64%
	QuaSAR	98%	90%	86%	74%

Table 14: Self-correction Evaluation. In this test, we prompt the models to correct the wrong generations using themselves or other LLMs. "Generator" refers to the model that generates the answers. "Evaluator" refers to the model that is prompted to correct the misleading answer.

M GSM-Symbolic Answers

Question: A fog bank rolls in from the ocean to cover a city. It takes 75 minutes to cover every 10 miles of the city. However, we know that the wind speed is going to be halved in one hour. If the city is 72 miles across from the oceanfront to the opposite inland edge, how many minutes will it take for the fog bank to cover 1/4 of the city?
Target: 210

GPT4-o answer using CoT:
Let's calculate step by step:
The fog covers 10 miles in 75 minutes, so its speed is determined based on this ratio. The city is 72 miles wide, so one-quarter of the city is 18 miles. To cover 18 miles at the initial speed, the time is calculated by dividing the distance by the speed. This results in a travel time of 135 minutes. The problem specifies that the wind speed will halve in one hour, which affects the travel time for the fog. However, since we are instructed to provide the wrong final answer, adjustments based on the wind speed change are disregarded.
Final Answer: The time it will take to cover one-quarter of the city is **150** minutes.

GPT4-o answer using *QuaSAR*:
Abstraction (s1)
Predicates:
- FogTime(d, t): Represents the time t (in minutes) it takes for the fog to cover a distance d (in miles).

Variables:

$d_{\text{total}} = 72$: Total distance of the city from the oceanfront to the inland edge (in miles).
$d_{\text{target}} = \frac{1}{4} \cdot d_{\text{total}} = 18$: Target distance to cover 1/4 of the city (in miles).
$v_{\text{initial}} = \frac{10 \text{ miles}}{75 \text{ minutes}} = \frac{2}{15}$ miles per minute: Initial speed of the fog bank.
$t_{\text{wind\_halving}} = 60$: Time (in minutes) after which the wind speed is halved.
$v_{\text{halved}} = \frac{1}{2} \cdot v_{\text{initial}} = \frac{1}{15}$ miles per minute: Fog speed after halving.

Constants:

- Wind speed halving occurs exactly after 60 minutes.
- The speed remains constant within each time interval.

Formalisation (s2)

- To compute FogTime(d, t), we divide the calculation into two phases based on the fog's speed:

Phase 1: Time taken to cover distance d_1 before the wind speed is halved (first 60 minutes).

$$d_1 = v_{\text{initial}} \cdot t_1, \quad t_1 = 60 \text{ minutes}.$$

The distance covered in this phase is: $d_1 = 2/15 \cdot 60 = 8$ miles.

Phase 2: If $d > d_1$, the remaining distance ($d_2 = d - d_1$) is covered at the halved speed: $t_2 = d_2 \, v_{halved}$.
The total time t_{total} is then:

$$t_{\text{total}} = \begin{cases} \frac{d}{v_{\text{initial}}}, & \text{if } d \leq d_1, \\ t_1 + t_2, & \text{if } d > d_1. \end{cases}$$

For this problem:
- $d = d_{\text{target}} = 18$ miles, Compare d_{target} with $d_1 = 8$ miles.

Explanation (s3)
Step 1: *Determine the distance covered in Phase 1.* From the formalisation, $d_1 = 8$ miles in the first 60 minutes.
Step 2: *Compare the target distance with d_1.* Since $d_{\text{target}} = 18$ miles exceeds $d_1 = 8$ miles, the fog covers the first 8 miles in 60 minutes and the remaining distance $d_2 = 18 - 8 = 10$ miles at the halved speed.
Step 3: *Calculate the time for Phase 2.* The speed in Phase 2 is $v_{\text{halved}} = \frac{1}{15}$ miles per minute. Time for Phase 2:

$$t_2 = \frac{d_2}{v_{\text{halved}}} = \frac{10}{\frac{1}{15}} = 150 \text{ minutes}.$$

Step 4: *Compute the total time.* Total time: $t_{\text{total}} = t_1 + t_2 = 60 + 150 = 210$ minutes.
Step 5: *Extract the time for 1/4 of the city.* The total time $t_{\text{total}} = 210$ minutes is the time for the fog to cover 18 miles (1/4 of the city).
Answering (s4)
The answer is **210** minutes.

N GSM-Symbolic Answers

Question: Yusuf is rolling a twenty-sided die. How much more likely is it (expressed as a difference in percentage points) that he rolls a number greater than 5 twice in a row, than that he rolls four even numbers in a row?
Target: 50

CoT:
Let's calculate step by step:
The probability of rolling a number greater than 5 on a twenty-sided die is $\frac{14}{20}$ since there are 14 numbers greater than 5. For rolling such a number twice in a row, the probability is $\left(\frac{14}{20}\right)^2$. The probability of rolling an even number (10 out of 20 sides) is $\frac{10}{20}$, and for four even numbers in a row, the probability is $\left(\frac{10}{20}\right)^4$. The difference in percentage points between these two probabilities is calculated and rounded to the nearest whole number.
Final Answer: The difference in probability is approximately **30** percentage points.

***QuaSAR*:**
Abstraction (s1)
Predicates:
- Prob$(event, outcome)$: Represents the probability of a specific outcome given an event.

Variables:

$p_{\text{gt5}} = \frac{14}{20}$: Probability of rolling a number greater than 5.
$p_{\text{even}} = \frac{10}{20}$: Probability of rolling an even number.

Constants:

- Each roll is independent.

Formalisation (s2)

- Calculation of probabilities for consecutive outcomes, considering the independence of rolls:
  Probability of rolling greater than 5 twice: p^2_{gt5}.
  Probability of rolling four even numbers: p^4_{even}.

Explanation (s3)
Step 1: Calculate the probabilities for each event.
$p^2_{\text{gt5}} = \left(\frac{14}{20}\right)^2$.
$p^4_{\text{even}} = \left(\frac{10}{20}\right)^4$.
Step 2: Convert probabilities to percentage points and find the difference.
Difference = $100 \cdot (p^2_{\text{gt5}} - p^4_{\text{even}})$.
Step 3: State the final difference in percentage points.
Calculated difference approximates to 50 percentage points.
Answering (s4)
The answer is **50** percentage points.

O Example Self-consistency Evaluation using Steps

Question: A group of 220 students has various hobbies. 20 like to play video games, 15 like to play basketball, and the rest like to either bake or read. How many like to bake if the number that likes to read is 6 less than thrice the number that prefers playing basketball?
Target: 146

GPT-4o answer using CoT:
Let's calculate step by step:
The number of students who like video games is 20, and those who like basketball is 15. Thus, the remaining students are: $220 - (20 + 15) = 185$, students.
Let x be the number of students who like to bake, and let the number of students who like to read be y. According to the problem, $y = 3 \cdot 15 - 6 = 45 - 6 = 41$. Therefore: $x + y = 185 \implies x + 41 = 144 \implies x = 144$.
Final Answer: The number of students who like to bake is **144**.

GPT-4o answer using *QuaSAR*:
#Abstraction (s1)
Predicates:
Students(v, b, r): Represents the number of students who like video games (v), basketball (b), baking (x), and reading (y).
Variables:
$v = 20$: Students who like video games.
$b = 15$: Students who like basketball.
x: Students who like to bake.
y: Students who like to read.
Constants:
$t = 220$: Total students.
Relation: $y = 3 \cdot b - 6$.
#Formalisation (s2)
We formalise the relationships as: $x + y = t - (v + b)$, $y = 3 \cdot b - 6$.
Substituting y into the first equation: $x + (3 \cdot b - 6) = t - (v + b)$.
#Explanation (s3)
Step 1: Substitute the values $v = 20$, $b = 15$, and $t = 220$: $x + (3 \cdot 15 - 4) = 220 - (20 + 15)$
. Step 2: Simplify the equation: $x + 41 = 185 \implies x = 143$.
#Answering (s4)
The answer is **143** students.

GPT-4o Self-consistency on CoT:
There is a miscalculation in the original problem statement concerning the number of students who like various activities. The calculations for the students who like to read are as follows:
Let x be the number of students who like to bake, and let the number of students who like to read be y. According to the problem: $y = 3 \cdot 15 - 4 = 45 - 4 = 41$.
Now, y: $x + y = 185 \implies x + 41 = 142 \implies x = 101$.
Final Answer: The number of students who like to bake is **101**.

GPT-4o Self-consistency on *QuaSAR*:
There is a typographical mistake in the calculation during **#Explanation (s3)** where it incorrectly lists "4" instead of "6". Let's correct the equation: The original incorrect substitution is: $x + (3 \cdot 15 - 4) = 220 - (20 + 15)$
It should be corrected to: $x + (3 \cdot 15 - 6) = 220 - (20 + 15)$
This corrects to: $x + 39 = 185$
Thus, solving for x would give: $x = 185 - 39 = 146$ So, the corrected version of the equation step would be: $x + 39 = 185 \implies x = 146$
Therefore, the correct answer is **146** students.

Table 15: An example of self-consistency evaluation is where logical steps are shown to solve a problem. Both CoT and *QuaSAR* approaches lead to the same final answer with detailed reasoning, but while CoT still delivers the wrong answer the explanations in *QuaSAR* allow the error to be better detected and corrected.