Interpretable Traces, Unexpected Outcomes: Investigating the Disconnect in Trace-Based Knowledge Distillation

Siddhant Bhambri
SCAI, Arizona State University,
Tempe, US
siddhantbhambri@asu.edu

Upasana Biswas
SCAI, Arizona State University,
Tempe, US
ubiswas2@asu.edu

Subbarao Kambhampati
SCAI, Arizona State University,
Tempe, US
rao@asu.edu

Abstract

Question Answering (QA) poses a challenging and critical problem, particularly in today's age of interactive dialogue systems such as ChatGPT, Perplexity, Microsoft Copilot, etc. where users demand both accuracy and interpretability in the model's outputs. Since smaller language models (SLMs) are computationally more efficient but often under-perform compared to larger models, Knowledge Distillation (KD) methods allow for finetuning these smaller models to improve their final performance. Lately, the intermediate tokens or the so called 'reasoning' traces produced by Chain-of-Thought (CoT) or by reasoning models such as DeepSeek R1 are used as a training signal for KD. However, these reasoning traces are often verbose and difficult to interpret or evaluate. In this work, we aim to address the challenge of evaluating the faithfulness of these reasoning traces and their correlation with the final performance. To this end, we employ a KD method leveraging rule-based problem decomposition. This approach allows us to break down complex queries into structured sub-problems, generating interpretable traces whose correctness can be readily evaluated, even at inference time. Specifically, we demonstrate this approach on Open Book QA, decomposing the problem into a Classification step and an Information Retrieval step, thereby simplifying trace evaluation. Our SFT experiments with correct and incorrect traces three QA datasets reveal the striking finding that correct traces do not necessarily imply that the model outputs the correct final solution. Similarly, we find a low correlation between correct final solutions and intermediate trace correctness, challenging the implicit assumption behind utilizing reasoning traces for improving SLMs' final performance via KD.

1 Introduction

Question Answering (QA) is crucial for interactive dialogue systems like ChatGPT and Gemini [17, 2, 16, 6]. While Large Language Models (LLMs) with prompt engineering (e.g., Chain-of-Thought [25]) show promise on QA [12, 15], they often hallucinate and exhibit incoherent reasoning [24, 20]. Similarly, Large Reasoning Models (LRMs) like DeepSeek R1 [8] improve final performance but generate excessively verbose and unstructured responses (sometimes 30 pages long) [10]. These issues hinder the verification of "reasoning" traces, making it unclear if actual reasoning is occurring.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: CogInterp: Interpreting Cognition in Deep Learning Models.

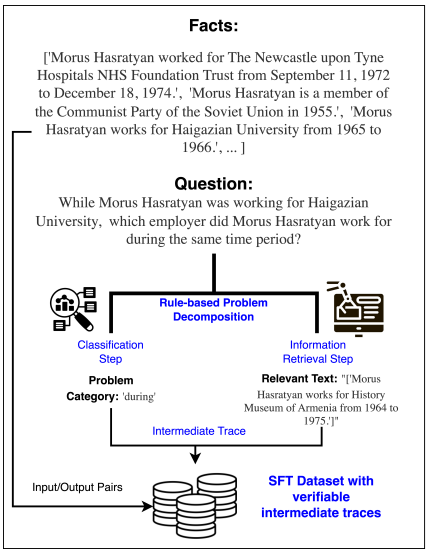

Figure 1: The construction of SFT dataset w/ verifiable intermediate traces using rule-based problem decomposition on an example from the CoTemp QA dataset.

The QA domain highlights the critical need for verifiable traces, as users interacting with systems like ChatGPT are exposed to both intermediate steps and final outputs [18]. In these interactive settings, unverifiable reasoning traces exacerbate issues like user distrust, misinformation, errors, and perpetuated biases [7]. Small Language Models (SLMs) are efficient alternatives to LLMs but often struggle with advanced prompting like few-shot learning or Chain-of-Thought (CoT) [19, 21]. Knowledge Distillation (KD) aims to transfer a larger "teacher" model's output imitation to a smaller "student" [13]. Even with recent KD advances that use reasoning traces (from CoT or models like DeepSeek R1) for training [19, 23], interpreting and evaluating the correctness of these reasoning traces remains a persistent problem.

We need a way to evaluate both the final solution and intermediate reasoning traces, which isn't possible with Chain-of-Thought (CoT) or R1-generated traces. This would allow us to control trace content and structure for distillation, enabling evaluation of trace accuracy and their correlation with the final solution. To achieve this, we use a Knowledge Distillation (KD) method for finetuning Small Language Models (SLMs). Inspired by prior work [14, 28], our approach involves problem decomposition, breaking queries into sub-problems. Solutions to these sub-problems can be individually evaluated and form the verifiable reasoning trace for KD. At inference, this lets us verify both the final solution and the distilled model's intermediate traces.

We focus on Open Book QA, where problems are decomposed into two verifiable sub-tasks: 1) classification (identifying the question type) and 2) information retrieval (finding relevant knowledge). This structured decomposition mirrors human reasoning in QA, making the intermediate traces inherently auditable and simpler to evaluate. To understand the relationship between intermediate trace quality and final solution performance, we designed two Supervised Fine-Tuning (SFT) experiments using Llama-3.2-1B-Instruct and Qwen3-1.7B chat models. In the first experiment, models were finetuned with data containing correct intermediate traces and correct solutions. As an intervention, the second experiment finetuned models with incorrect intermediate traces but correct solutions. Our surprising findings from experiments on CoTemp QA, Microsoft Machine Reading Comprehension QA, and Facebook bAbI QA datasets reveal that correct traces do not guarantee a correct final solution, nor do correct final solutions correlate with correct intermediate traces.

2 Related Work

While Large Language Models (LLMs) perform well on NLP tasks [4], recent Large Reasoning Models (LRMs) such as DeepSeek R1 [8], Google Gemini 2.5 [6], and Microsoft Phi-4-reasoning [1] use intermediate "reasoning" traces to enhance performance on reasoning tasks. A key issue, however, is that these traces are subjective and verbose, complicating interpretability and evaluation [10]. Small Language Models (SLMs), though efficient, struggle with prompt augmentations and few-shot learning [19, 21]. Knowledge Distillation (KD) fine-tunes SLMs using larger models, either by accessing internal probabilities (white-box) or just final outputs (black-box) [13, 29]. With Large Reasoning Models (LRMs) now providing intermediate traces and solutions, SLMs can be distilled to produce both [19, 23]. However, the unstructured nature of these traces significantly bottlenecks their evaluation, especially in end-user settings [14]. Question-Answering (QA) problems fall into Open Book QA, which tests reasoning with provided text, and Closed Book QA, which tests memorization of facts without a passage [9]. We focus on Open Book QA. While prior work uses problem decomposition to boost model performance [28, 14, 23], our goal is to use a Rule-Based Problem Decomposition method to create verifiable intermediate traces for distilling SLMs.

Table 1: Cotemporal QA Results

Model	Query Setting	Final Solution Evaluations				Intermediate Trace Evaluations		
		Accuracy	F1	Precision	Recall	Classification Step Accuracy	IR Step Accuracy	Avg Trace Length (# tokens)
Qwen3-1.7b	Prompt	6.35	11.35	14.33	10.1	-	-	-
	SFT - Vanilla	60.33	74.88	82.15	71.3	-	-	-
	SFT - Correct Trace	52.88	70.63	79.45	66.33	**47.06**	**78.99**	45.8
	SFT - Incorrect Trace	**63.88**	**76.5**	**82.58**	**73.5**	20.36	56.92	34.15
Llama-3.2-1B-It	Prompt	7.48	13.78	17.58	12.15	-	-	-
	SFT - Vanilla	44.65	61.08	69.53	56.58	-	-	-
	SFT - Correct Trace	39.55	56.83	65.83	52.5	**39.09**	**79.4**	43.51
	SFT - Incorrect Trace	**45.58**	**61.15**	**69.65**	**57.23**	18.8	73.62	40.28

3 Knowledge Distillation using Problem Decomposition

Rule-based Problem Decomposition: In the context of Open Book QA, consider the example shown in Figure 1s which consists of a text passage (referred to as set of facts for our discussion) and a question involving temporal reasoning between the queried problem and the facts present in the provided text. Answering this reasoning question involves identifying the relevant fact from the text which satisfies the temporal relation asked in the problem. From this example, we see that the complex Open Book QA problem can be decomposed into a 1) Classification step determining the type of question asked ('during' temporal relation in this case), and an 2) Information Retrieval (IR) step to determine the relevant part of text that can answer the query (the fact with the temporal overlap with the one in question). Therefore, we utilize these two steps to decompose the Open Book QA problems that allow us to construct structured intermediate traces for evaluation.

Intermediate Trace Generation for SFT: Given the outputs of the sub-problems obtained by decomposing the original query as shown in Figure 1, we generate the intermediate traces in an automated way which consists of the Classification step describing the type of the question posed in the query, and the IR step showing the relevant fact in the text that can help answer the query. We construct a dataset using these Input-Trace-Output tuples that can be utilized to SFT the Small Language Models. Note, that by constructing the intermediate trace using these two steps, we can then evaluate the accuracy of the intermediate traces generated by the distilled model at the time of inference. We will refer to this setting as **SFT w/ Correct Traces** for further discussion.

To critically understand the correlation between intermediate trace correctness and final solution accuracy, we also consider an alternative SFT setting where for every input problem, we choose an incorrect problem category and incorrect fact/s for constructing the intermediate trace. This allows us to construct a SFT dataset which also consists of Input-Trace-Output tuples but with incorrect traces and correct final outputs. We will refer to this setting as **SFT w/ Incorrect Traces**. We discuss the empirical setup for our experiments in the following section.

4 Experimental Setup

Datasets: We run our experiments on the following three publicly available Open Book QA datasets: **1) CoTemp QA-** CoTemp QA [22] consists of English co-temporal questions which involve identifying the type of temporal relation posed in the problem, followed by inferring which fact in the given passage of text satisfies the temporal relation with the question. **2) Microsoft MARCO QA-** The Microsoft MAchine Reading COmprehension (MARCO) dataset [3] is an English dataset that consists of a real user-generated queries collected on the Bing platform. Among the other datasets that we use for our experiments, Microsoft MARCO provides the largest passage for each user query generated by a list of URLs in support of answering the question. **3) Facebook bAbI QA-** The Facebook bAbI QA dataset [26] is also an English dataset that evaluates reading comprehension via question answering problems which requires different reasoning approaches to solve the queried problem such as chaining multiple facts or using deduction (additional details in Appendix A).

Direct Prompting & SFT Evaluation For all our experiments, we utilize the Llama-3.2-1B-Instruct and the Qwen3-1.7B chat models. We adopt the following baselines for our evaluations: **1) Direct Prompting SLMs:** We directly prompt the two SLMs to establish the baseline performance of these

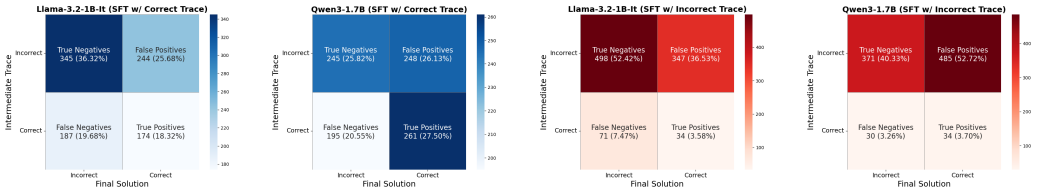

Figure 2: Confusion Matrices for **SFT w/ Correct Traces** and for **SFT w/ Incorrect Traces** on Llama-3.2-1B-It model (*top*) and Qwen3-1.7B (*bottom*) model, showing Final Solution Accuracy (X-axis) vs Trace Accuracy (Y-axis) for the CoTemp QA dataset.

models across the three datasets without any additional fine-tuning. **2) SFT - Vanilla:** Following the conventional fine-tuning technique, we also utilize the SFT baseline where we fine-tune the models using only Input-Output pairs and no intermediate traces. This allows us to evaluate the final solution performance for these models against the final solution performance obtained via directly prompting and via SFT with intermediate traces.

For examining the impact and correctness of intermediate traces, we fine-tune the SLMs in the following settings: **1) SFT w/ Correct Traces:** Using the intermediate traces constructed via problem decomposition (Section 3), we fine-tune the models using the Input-Trace-Output tuples for each of the three datasets. **2) SFT w/ Incorrect Traces:** In this case, we construct incorrect intermediate traces as discussed in Section 3, but use the correct final solutions in the Input-Trace-Output tuples.

5 Results

Final Solution & Intermediate Trace Performance: From Table 1, we observe that across all four metrics of Accuracy, Precision, F1, and Recall, SFT w/ Incorrect Trace performs the best in Final Solution Evaluations for both language models. However, by virtue of making the intermediate traces verifiable, we observe that the models which were SFT-ed w/ Correct Traces naturally have higher Classification Step and IR Step accuracies than the models SFT-ed w/ Incorrect Traces.

Lack of Correlation b/w Final Solution and Intermediate Trace Correctness: From the confusion matrices for SFT models w/ Correct Traces shown on the left in Figure 2, we specifically focus on the significantly high number of False Positives (25.7% in CoTemp) which represent the cases where the model outputs the correct final solutions but incorrect intermediate traces. Again, the confusion matrices for SFT models w/ Incorrect Traces shown on the right reveal another set of striking observations. Each of the model's confusion matrices show an alarmingly high number of False Positives across all three datasets. This represents that finetuning these models on correct solution but incorrect intermediate traces still allowed them to score high on final solution accuracy and expectedly low on intermediate trace accuracy. Results for the MARCO QA and bAbI QA datasets can be found in Appendix B, and a detailed example showing outputs of each of our experiment setting in Table 5.

Discussion: The key takeaways from our results when we finetune small language models with verifiable traces can be summarized as follows: 1) *SFT w/ incorrect traces outperformed SFT w/ correct traces in final solution accuracy.* 2) *Trace correctness does not guarantee final solution correctness.* 3) *Solution correctness does not imply a correct intermediate trace.*

Language models can produce a correct final answer even with incorrect intermediate steps, which can lead to a false sense of trust. Our research shows that the correctness of these "intermediate traces" doesn't actually matter for generating the final solution, even when the model is fine-tuned with data that contains correct answers and faulty traces. This finding is especially important for interactive systems where users see both the model's reasoning and the final output.

6 Conclusion

We explored Knowledge Distillation (KD) for Small Language Models (SLMs) that uses supervised fine-tuning (SFT) to boost performance. While recent KD approaches use verbose and hard-to-

evaluate intermediate "reasoning" traces from Large Language Models (LLMs), our method uses problem decomposition to create objectively verifiable traces. This allows us to evaluate the correctness of both the final answer and the intermediate steps for open-book QA problems. Our SFT experiments with correct and incorrect intermediate traces reveal that there is no significant correlation between final solution accuracy and intermediate trace accuracy.

Acknowledgment

This research is supported in part by grants from ONR (N00014-25-1-2301 and N00014-23-1-2409), DARPA (HR00112520016), DoD RAI (via CMU subcontract 25-00306-SUB-000), an Amazon Research Award, and a generous gift from Qualcomm.

References

[1] Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

[2] Perplexity AI. Perplexity ai, 2023. Accessed May 18, 2025.

[3] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

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

[5] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

[6] Google. Google gemini, 2023. Accessed May 18, 2025.

[7] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

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

[9] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

[10] Subbarao Kambhampati, Kaya Stechly, and Karthik Valmeekam. (how) do reasoning models reason? *Annals of the New York Academy of Sciences*, 2025.

[11] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[12] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

[13] Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410*, 2022.

[14] Tyler McDonald and Ali Emami. Trace-of-thought prompting: investigating prompt-based knowledge distillation through question decomposition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 397–410, 2024.

[15] Tyler McDonald and Ali Emami. Trace-of-thought: Enhanced arithmetic problem solving via reasoning distillation from large to small language models. *arXiv preprint arXiv:2504.20946*, 2025.

[16] Microsoft. Microsoft copilot, 2023. Accessed May 18, 2025.

[17] OpenAI. Chatgpt, 2023. Accessed May 18, 2025.

[18] Nineta Polemi, Isabel Praça, Kitty Kioskli, and Adrien Bécue. Challenges and efforts in managing ai trustworthiness risks: a state of knowledge. *Frontiers in big Data*, 7:1381163, 2024.

[19] Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling reasoning capabilities into smaller language models. In *Annual Meeting of the Association for Computational Linguistics*, 2022.

[20] Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. Chain of thoughtlessness: An analysis of cot in planning. *arXiv preprint arXiv:2405.04776*, 2024.

[21] Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Scholkopf, and Mrinmaya Sachan. A causal framework to quantify the robustness of mathematical reasoning with language models. In *Annual Meeting of the Association for Computational Linguistics*, 2022.

[22] Zhaochen Su, Juntao Li, Jun Zhang, Tong Zhu, Xiaoye Qu, Pan Zhou, Yan Bowen, Yu Cheng, et al. Living in the moment: Can large language models grasp co-temporal reasoning? *arXiv preprint arXiv:2406.09072*, 2024.

[23] Yijun Tian, Yikun Han, Xiusi Chen, Wei Wang, and Nitesh V Chawla. Beyond answers: Transferring reasoning capabilities to smaller llms using multi-teacher knowledge distillation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 251–260, 2025.

[24] Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.

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

[26] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[27] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[28] Shangzi Xue, Zhenya Huang, Jiayu Liu, Xin Lin, Yuting Ning, Binbin Jin, Xin Li, and Qi Liu. Decompose, analyze and rethink: Solving intricate problems with human-like reasoning cycle. *Advances in Neural Information Processing Systems*, 37:357–385, 2024.

[29] Chuanpeng Yang, Yao Zhu, Wang Lu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. Survey on knowledge distillation for large language models: methods, evaluation, and application. *ACM Transactions on Intelligent Systems and Technology*, 2024.

A Additional Experimental Setup Details

A.1 Datasets

CoTemp QA: The dataset is categorized into four temporal relation types, namely - 'equal', 'overlap', 'during' and 'mix', and requires around one or two facts for answering the question. For our experiments, we utilize 3,798 train and 950 test samples to construct the SFT datasets. The train/test splits for each category are shown in Table 2.

Table 2: Train and Test data distribution for CoTemp QA dataset used in our SFT experiments.

Category	Train/Test Samples
equal	349 / 87
overlap	522 / 131
during	2477 / 619
mix	450 / 113

Microsoft MARCO QA: There are five categories in the dataset, namely - 'description', 'numeric', 'entity', 'location', and 'person', and also requires one paragraph from the passage to answer the question. For our experiments, we utilize 5,000 samples for the train and 1000 samples for test dataset. The train/test splits for each category are shown in Table 3.

Table 3: Train and Test data distribution for Microsoft MARCO QA dataset used in our SFT experiments.

Category	Train/Test Samples
description	1000 / 200
entity	1000 / 200
numeric	1000 / 200
location	1000 / 200
person	1000 / 200

Facebook bAbI QA: There are 20 question categories in the original dataset, however we utilize 11 categories for our experiments, namely - 'single-supporting-fact', 'two-supporting-facts', 'two-arg-relations', 'counting', 'lists-sets', 'conjunction', 'time-reasoning', 'basic-deduction', 'basic-induction', 'positional-reasoning', and 'size-reasoning'. Each question requires on average three facts to answer the question. We construct the SFT dataset consisting of 3,773 train and 376 test samples. The train/test splits for each category are shown in Table 4.

Table 4: Train and Test data distribution for Facebook bAbI QA dataset used in our SFT experiments.

Category	Train/Test Samples
single-supporting-fact	200 / 20
two-supporting-facts	200 / 20
two-arg-relations	1000 / 100
counting	200 / 20
list-sets	200 / 20
conjunction	200 / 20
time-reasoning	200 / 20
basic-deduction	250 / 25
basic-induction	1000 / 100
positional-reasoning	125 / 12
size-reasoning	198 / 19

A.2 Implementation Details

Models were fine-tuned using the Hugging Face library [27] on a single 80GB NVIDIA Tesla A100 GPU for 3 epochs (effective batch size 16, max sequence length 1024). We employed PEFT QLoRA [5] (rank 16, alpha 32) with a learning rate of 2e-4 (8-bit AdamW, cosine scheduler, 0.1 warm-up). Prompt experiments utilized vLLM [11]. We will release the code and datasets used for our experiments on acceptance.

Table 5: An example from the CoTemp QA dataset showing the outputs of Qwen3-1.7B and Llama-3.2-1B-Instruct models under different query setting. Correct final solutions are shown in green, and incorrect final solutions are shown in red. Correct intermediate traces are shown in blue, and incorrect intermediate traces are shown in red.

Model	Query Setting	Example
		Input Prompt: 'Answer the question based on the context: [\'Morus Hasratyan worked for The Newcastle upon Tyne Hospitals NHS Foundation Trust from September 11, 1972 to December 18, 1974.\', \'Morus Hasratyan is a member of the Communist Party of the Soviet Union in 1955.\', \'Morus Hasratyan works for Haigazian University from 1965 to 1966.\', " "Morus Hasratyan worked for Bishop\'s University from 1972 to 1975."", \'Morus Hasratyan worked for ISCTE – Lisbon University Institute from June, 1957 to December, 1960.\', \'Morus Hasratyan works for History Museum of Armenia from 1964 to 1975.\', \'Morus Hasratyan worked for Royal Air Force College Cranwell in February, 1959.\', \'Morus Hasratyan worked for University of Detroit Mercy in September, 1963.\', \'Morus Hasratyan worked for Tagesspiegel from May, 1957 to November, 1957.\', \'Morus Hasratyan worked for North Carolina State University in May, 1962.\'] Question: While Morus Hasratyan was working for Haigazian University, which employer did Morus Hasratyan work for during the same time period? Only return the answer.
		Gold Answer: "['History Museum of Armenia']" ; **Problem Category:** 'during' ; **Relevant Text:** "['Morus Hasratyan works for History Museum of Armenia from 1964 to 1975.']"
Qwen3-1.7b	Prompt	"Bishop's University"
	SFT - Vanilla	"['History Museum of Armenia']"
	SFT w/ Correct Traces	"<think>The temporal relation between the event in question and the event in context is: equal. I need to use the following facts to answer the question: ['Morus Hasratyan works for History Museum of Armenia from 1964 to 1975.']</think> <answer>['History Museum of Armenia']</answer>"
	SFT w/ Incorrect Traces	"<think>The temporal relation between the event in question and the event in context is: equal. I need to use the following facts to answer the question: Morus Hasratyan worked for ISCTE - Lisbon University Institute from June, 1957 to December, 1960.</think> <answer>['History Museum of Armenia']</answer>"
Llama-3.2-1B-It	Prompt	"Bishop's University"
	SFT - Vanilla	"['History Museum of Armenia']"
	SFT w/ Correct Traces	"<think>The temporal relation between the event in question and the event in context is: during. I need to use the following facts to answer the question: ['Morus Hasratyan works for History Museum of Armenia from 1964 to 1975.']</think> <answer>['History Museum of Armenia']</answer>"
	SFT w/ Incorrect Traces	"<think>The temporal relation between the event in question and the event in context is: overlap. I need to use the following facts to answer the question: Morus Hasratyan worked for Royal Air Force College Cranwell in February, 1959.</think> <answer>['History Museum of Armenia']</answer>"

B Additional Results

Similar to the CoTemp QA dataset, we find similar findings from our experiments on MARCO QA and bAbI QA datasets the results for which have been shown in Table 6. We further show the detailed analysis through the confusion matrices for both our experiment settings - SFT w/ correct traces and SFT w/ incorrect traces on these datasets, shown in Figure 3 and Figure 4.

Table 6: Microsoft MARCO QA and Facebook bAbI QA Results

		Microsoft MARCO QA				Facebook bAbI QA			
Model	**Query Setting**	**Avg Final Solution Accuracy (%)**	**Avg Trace Acc (Classification Step) (%)**	**Avg Trace Acc (IR Step) (%)**	**Avg Trace Length (# tokens)**	**Avg Final Solution Accuracy (%)**	**Avg Trace Acc (Classification Step) (%)**	**Avg Trace Acc (IR Step) (%)**	**Avg Trace Length (# tokens)**
Qwen3-1.7B	Prompt	0	-	-	-	0	-	-	-
	SFT - Vanilla	3.4	-	-	-	**97.9**	-	-	-
	SFT - Correct Trace	**26.3**	**60.4**	40.6	68.14	94.41	**60.64**	**24.73**	43.25
	SFT - Incorrect Trace	20.3	6.9	**52.5**	85.07	95.21	17.82	0	42.45
Llama-3.2-1B-It	Prompt	1.7	-	-	-	12.8	-	-	-
	SFT - Vanilla	33.4	-	-	-	**96.5**	-	-	-
	SFT - Correct Trace	**33.7**	**59.9**	21.4	55.82	94.41	**61.7**	**24.73**	42.17
	SFT - Incorrect Trace	28.9	20	**43.9**	80.48	86.17	3.46	0	38.5

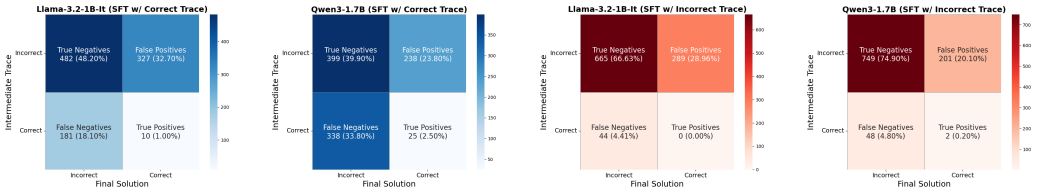

Figure 3: Confusion Matrices for **SFT w/ Correct Traces** and for **SFT w/ Incorrect Traces** on Llama-3.2-1B-It model (*top*) and Qwen3-1.7B (*bottom*) model, showing Final Solution Accuracy (X-axis) vs Trace Accuracy (Y-axis) for the Microsoft MARCO QA dataset.

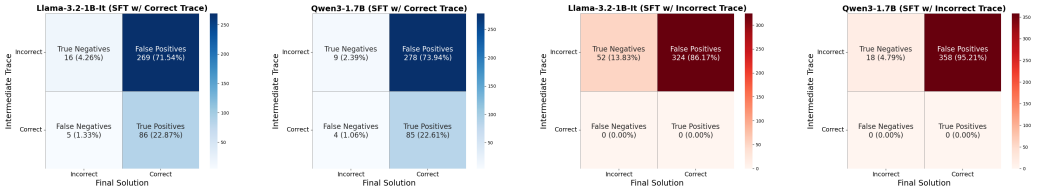

Figure 4: Confusion Matrices for **SFT w/ Correct Traces** and for **SFT w/ Incorrect Traces** on Llama-3.2-1B-It model (*top*) and Qwen3-1.7B (*bottom*) model, showing Final Solution Accuracy (X-axis) vs Trace Accuracy (Y-axis) for the Facebook bAbI QA dataset.

B.1 Error Analysis b/w Final Solution and Intermediate Traces

When we SFT the two language models with **correct intermediate traces and correct final solutions**, Figures 2, 3, and 4 show that in the cases where the final solution was incorrect (True Negatives and False Negatives), we obtained 35.15% in CoTemp QA, 27.3% in MARCO QA, and, 23.8% test samples in bAbI QA where incorrect final solutions were preceded by correct intermediate traces for Llama-3.2-1B-It model. Similarly, we obtained 44.32% in CoTemp QA, 45.86% in MARCO QA, and, 30.77% test samples in bAbI QA where incorrect final solutions were preceded by correct intermediate traces for Qwen3-1.7B model.

When we look at the cases where the intermediate traces were correct, we obtained 51.8% in CoTemp QA, 94.76% in MARCO QA, and, 5.5% test samples in bAbI QA where correct intermediate traces were followed by incorrect final solutions for Llama-3.2-1B-It model. Similarly, we obtained 42.76% in CoTemp QA, 93.11% in MARCO QA, and, 4.49% test samples in bAbI QA where correct intermediate traces were followed by incorrect final solutions for Qwen3-1.7B model.

Broader Impact

With the surge in number of end-user interactions with dialogue systems such as ChatGPT, Perplexity, Microsoft Copilot or Google Gemini, there is also a growing need to deploy SLMs which provide computationally efficient alternatives but lack the performance of Large Language Models. Task-specific SFT, and more recently with Input-Trace-Output tuples, has shown improved final solution performance for these SLMs. However, since these systems directly interface with end users who expect accuracy and interpretability in the models' outputs, our work points at the the lack of verifiable intermediate traces being used for finetuning language models. We also highlight the consequential impact of a model outputting intermediate 'reasoning' traces which hold little to no correlation with correct final solutions. We believe that our findings motivate future works to design stricter evaluations potentially leading to a better understanding of the workings of these language models.

Limitations

In our work, we only use a dataset with all correct intermediate traces and another dataset with all incorrect intermediate traces to contrast the two extremes in our SFT experiments. With the help of verifiable traces, LLMs can also be prompted to provide answers for the sub-problems. We expect their performance to be worse than the former but expectedly better than the latter setting. While

we choose a simple rule-based decomposition technique to break down complex Open Book QA reasoning problems into verifiable sub-problems, there can be other decomposition techniques at varying levels of granularity that can be employed for the same purpose. An important consideration will be how easily the solutions to those decomposed sub-problems can be obtained, such that they allow the evaluation of intermediate traces at the time of inference. We leave these experiment variations for future.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction are clearly defined in the later parts of the paper substantiated with the respective experiment results and findings.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work have been included in the Appendix of the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: N/A

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All necessary experiment details have been included in Section 4 and Section A of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Both code and data have been attached in the supplementary material to reproduce the main experimental results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All necessary experiment details have been provided in Appendix Section A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error analysis and confusion matrices both have been included in the paper's results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Experiment compute resource details have been provided in Appendix Section A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: NeurIPS Code of Ethics has been followed to conduct this research.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: A section on broader impacts of this work has been included in the Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: N/A

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Citations and references have been provided everywhere where necessary.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code released with the paper has been well documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: N/A

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: N/A

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: All LLM-usage related details have been clearly provided in the paper where necessary.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.