

# Style over Substance: Distilled Language Models Reason Via Stylistic Replication

Philip Lippmann

Delft University of Technology

Jie Yang

Delft University of Technology

## Abstract

Specialized reasoning language models (RLMs) have demonstrated that scaling test-time computation through detailed reasoning traces significantly enhances performance. Although these traces effectively facilitate knowledge distillation into smaller, instruction-tuned models, the precise nature of transferred reasoning remains unclear. In this study, we investigate to what extent distilled models internalize replicated stylistic patterns during reasoning. To this end, we systematically analyze reasoning traces, identifying structural and lexical patterns that characterize successful reasoning. We then introduce two new datasets – a dataset of emergent reasoning traces and a synthetic dataset explicitly constructed to replicate these stylistic patterns – to precisely examine their influence on distilled models’ reasoning capabilities. We find that models trained on the synthetic traces achieve comparable performance, indicating that distilled reasoning abilities rely significantly on surface-level patterns. Surprisingly, we observe an increase in performance even when the synthetic traces are altered to lead to the wrong answer. Our findings highlight how stylistic patterns can be leveraged to enhance LM reasoning across model families.

## 1 Introduction

Reasoning is fundamental to artificial intelligence, enabling systems to solve problems, make decisions, and explain outcomes. While traditional approaches to improving language model (LM) reasoning emphasize increased train-time compute (Kaplan et al., 2020; Hoffmann et al., 2022), recent research highlights that scaling test-time compute through self-refinement is similarly effective (Snell et al., 2024). This insight has inspired specialized reasoning-focused LMs (RLMs), such as o1 (OpenAI, 2024) and R1 (DeepSeek-AI et al., 2025), which generate detailed reasoning traces of their thought process during inference.

Reasoning traces have proven effective for distillation (Schmidhuber, 1992; Hinton et al., 2015), efficiently transferring sophisticated cognitive skills from RLMs to smaller, instruction-tuned models (Team, 2025). Yet, the precise nature of the reasoning knowledge transferred remains poorly understood (Allen-Zhu & Li, 2023). In particular, it is unclear whether distilled models genuinely internalize complex reasoning abilities or replicate superficial stylistic patterns from the original traces. This ambiguity echoes broader AI debates about whether language models genuinely understand content or simply engage in surface-level imitation (Bender & Koller, 2020; Bender et al., 2021; Shanahan, 2024; Mirzadeh et al., 2025). This uncertainty leads to a fundamental question: are we genuinely enhancing the reasoning capabilities of models, or merely teaching them to mimic domain-specific patterns that happen to improve performance on benchmarks? Motivated by this ambiguity, we aim to establish whether **style is key to improvements in reasoning**, where style is characterized primarily by structural attributes such as trace length, lexical coherence, and backtracking frequency, rather than comprehension itself.

To examine this, we first systematically analyze successful reasoning traces produced by state-of-the-art RLMs, identifying recurrent structural and lexical patterns. This analysis, guided by cognitive science frameworks that characterize critical stages in human problem-

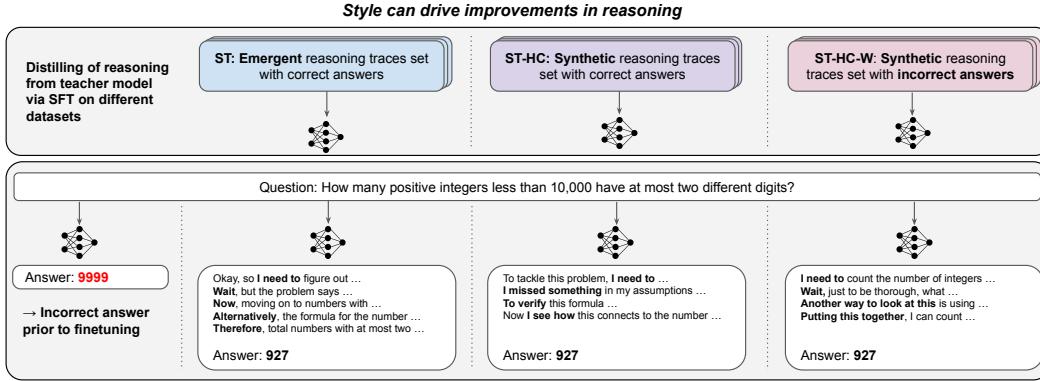


Figure 1: Reasoning trace style has noticeable influence on distilled model performance. We show how different types of reasoning traces – emergent traces with correct answers, synthetic traces with correct answers, and synthetic traces with incorrect answers – affect model performance after finetuning. All three approaches give the right answer due to improved reasoning capabilities compared to base models. Pivots highlighted in bold.

solving (Newell & Simon, 1972), reveals that effective reasoning traces consistently exhibit distinct *metacognitive behaviors*. These behaviors are often signaled by lexical pivots – markers such as “Wait” or “What if” – that prompt reconsideration of assumptions or integration of new insights.

Based on these findings, we introduce two complementary datasets explicitly designed to clarify the role of style in improving reasoning. The first dataset, SMOLTRACES (ST), comprises verified question-answer pairs with sophisticated reasoning traces generated by a state-of-the-art RLM, displaying naturally emergent reasoning behaviors. The second dataset, SMOLTRACES-HARDCODED (ST-HC), is synthetically constructed by embedding only the structural and lexical stylistic patterns identified earlier into reasoning traces generated by a standard LM without specialized reasoning capabilities. By comparing models trained on these datasets through supervised fine-tuning (SFT) (Brown et al., 2020), we evaluate how stylistic consistency influences reasoning performance. Our experiments demonstrate that even stylistically consistent synthetic traces from a weaker model achieve comparable downstream reasoning performance, underscoring the importance of style as a critical factor in model training.

Further experiments contextualize these findings through targeted ablation studies, which examine the relative importance of trace correctness versus style. Remarkably, we find that stylistically consistent reasoning traces – even those explicitly designed to lead to incorrect conclusions – still substantially enhance downstream reasoning performance over the base model, as shown in figure 1. These results underscore that stylistic consistency significantly influences LM reasoning capabilities, providing an explanation for the effectiveness of reasoning distillation from RLMs to regular LMs.

In summary, our work makes two major contributions: (1) we demonstrate that distilled reasoning improvements rely heavily on stylistic patterns present in reasoning traces; and (2) we identify specific structural and lexical features indicative of successful reasoning. Collectively, these contributions deepen our understanding of how language models perform reasoning tasks, and the datasets we release can serve as valuable resources for future research into synthetic data generation and fine-tuning methodologies that explicitly target the relationship between reasoning trace style and substance.

## 2 Background

**Chain-of-Thought** Early approaches to elicit reasoning from LMs, such as Chain-of-Thought (CoT) (Wei et al., 2023), demonstrate that intermediate reasoning steps are key for LMs to

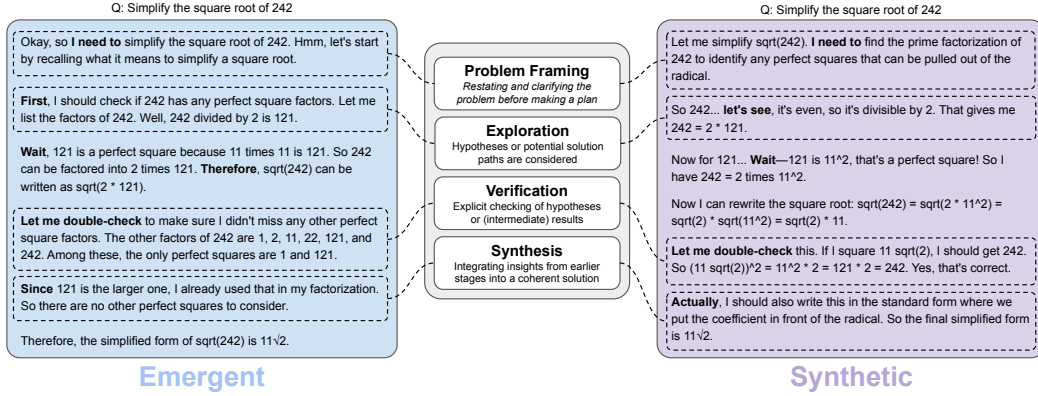


Figure 2: Comparison of emergent and synthetic reasoning traces for solving the same problem. The left side shows a reasoning trace generated by an RLM, while the right side displays a synthetic trace created using our hard-coded template with predefined pivots. Both approaches follow similar cognitive stages (center): problem framing, exploration, verification, and synthesis. The dashed boxes highlight examples of each stage, demonstrating that synthetic traces can effectively replicate the style of emergent reasoning.

improve their problem-solving abilities (Zhang et al., 2022). While CoT improves performance on reasoning tasks, it primarily focuses on generating a linear sequence of steps towards a solution. On the other hand, RLM reasoning traces (sometimes called “long CoT”) – which are the focus of this paper – differ from CoT as they do not just try to build toward the solution linearly, but instead actively backtrack, verify, and explore different lines of thinking.

**Reasoning traces** Reasoning traces are semi-structured textual representations that capture a model’s thought process while working toward the solution of a problem during inference. Here, additional tokens are generated before the final answer is given to reason about the problem. These traces typically include explicit markers of metacognition, such as planning statements, hypothesis testing, and self-correction. For an example of a full reasoning trace, see appendix A. A key characteristic of effective reasoning traces is the presence of *pivots*, points where the model explicitly moves between different categories of metacognition. We delve into the specific types of pivots in section 3.1.

**Finetuning on reasoning traces** Recent research has demonstrated that finetuning language models on reasoning traces significantly enhances their reasoning capabilities (Huang et al., 2024a; Team, 2025; DeepSeek-AI et al., 2025). In this approach, detailed reasoning traces generated by RLMs are used as training data to transfer structured reasoning behaviors into smaller, instruction-tuned models via SFT (Min et al., 2024). Models finetuned on these traces consistently outperform those trained only on final answers or simpler step-by-step solutions, suggesting that the explicit structure and content of reasoning traces play a critical role in improving model performance (Xu et al., 2025; Labs, 2025). Despite these successes, the precise mechanisms underlying the effectiveness of reasoning trace distillation remain unclear, particularly regarding the balance between structural stylistic cues and the cognitive complexity of the reasoning itself.

### 3 Hard-coding reasoning traces to approximate emergent ones

In examining whether distilled models internalize genuine reasoning capabilities or primarily benefit from structural and lexical patterns, we investigate the relationship between *style* and *substance* in reasoning traces. Here, style encompasses the structural and lexical features while substance refers to factual correctness and semantic content. To examine stylistic influence, we develop a methodology that replicates the structural patterns of successful reasoning while varying content, allowing us to assess style’s contribution to

reasoning distillation. We hypothesize that emergent reasoning behaviors can be effectively approximated by encoding the metacognitive pivots characteristic of RLM traces into synthetic reasoning traces. We define *emergent traces* as those naturally produced by RLMs after training via reinforcement learning, while *synthetic traces* refer to our approximations that incorporate the stylistic elements we identify as key, but originate from standard LMs without specialized reasoning capabilities. This approach reduces reliance on costly RLM inference<sup>1</sup> while enabling us to control stylistic components that potentially drive performance improvements in distilled models.

### 3.1 Reasoning trace structure and pivot types

**Trace analysis** Effective reasoning traces exhibit systematic structural patterns reflective of human-like problem-solving strategies. Cognitive science literature characterizes human reasoning as a structured process comprising distinct stages: *problem framing*, *hypothesis exploration*, *verification*, and *synthesis* (Newell & Simon, 1972). Guided by this framework, we systematically analyze 17K successful reasoning traces produced by a state-of-the-art RLM (see appendix B for details) and find that effective RLM-generated reasoning traces consistently align with these cognitive stages. Specifically, reasoning traces begin with explicit *problem framing*, restating and clarifying key aspects of the problem leading to a plan, followed by an *exploration* stage in which hypotheses or potential solution paths are considered. This is complemented by a *verification* stage, characterized by explicit checking of hypotheses or intermediate results, culminating in a final *synthesis*, integrating insights from earlier stages into a coherent solution.

**Pivot types** Crucially, effective reasoning traces are rarely linear; instead, they frequently revisit previous stages to correct errors, validate assumptions, or explore alternative strategies. This non-linear metacognitive behavior is operationalized through *pivots* – lexical markers signaling explicit shifts between reasoning stages. Our analysis identifies four primary pivot categories, each aligning closely with a corresponding reasoning stage: (1) *Realization pivots*, such as “Wait” or “Oh,” signal recognition of errors or oversights during the exploration stage; (2) *Verification pivots*, initiated by phrases such as “Let me check,” explicitly validate intermediate hypotheses; (3) *Exploration pivots*, introduced with phrases such as “What if” or “Another approach,” prompt the consideration of alternative solution paths; and (4) *Integration pivots*, typically signaled by expressions such as “Now I see how,” synthesize previously explored ideas into a coherent final solution. We visualize what these stages look like in practice for emergent and synthetic traces in figure 2

Our analysis reveals that successful reasoning traces commonly employ multiple pivot types (96.1% contain at least three pivot categories, additional information per type given in appendix B), whereas unsuccessful traces frequently lack such pivots or exhibit limited diversity. This underscores the critical importance of structured, metacognitive transitions in effective reasoning. Motivated by these insights, we explicitly encode these structural and lexical patterns into a reasoning template to facilitate the generation of synthetic reasoning traces that capture RLM-like reasoning behaviors. We show the associated prompt structure in figure 3, which we subsequently use to guide reasoning trace generation (section 3.2), enabling us to systematically control and isolate stylistic reasoning elements in order to examine their impact on reasoning performance. First, the pivot categories are explicitly defined, followed by the general stages of problem solving that they correspond to. To generate the synthetic data, in addition to this prompt, the teacher LM is given the question and instructions on how to format its answer.

### 3.2 Reasoning trace data generation

**Collect seed data** Initially we curate seed data consisting of questions and their correct answers, ensuring that the accuracy of the eventual synthetic data can be verified. While previous works on RLM distillation often focus exclusively on math (Huang et al., 2024a), we

<sup>1</sup>At time of writing, the difference in API costs per token between flagship LMs and RLMs approach an order of magnitude from the same provider: input/output pricing of \$2.50/\$10.00 for GPT-4o compared to \$15.00/\$60.00 for o1 <https://openai.com/api/pricing/> [Accessed: 2025-02-21]



Figure 3: The prompt used to guide GPT-4o for generating the synthetic ST-HC dataset. This prompt explicitly defines the four key pivot types (Realization, Verification, Exploration, Integration) and mandates adherence to the four reasoning stages (Problem Framing, Exploration, Verification, Synthesis) derived from our analysis of emergent RLM traces (section 3.1). Its goal is to enforce specific stylistic patterns, including non-linear thinking and explicit self-correction, characteristic of effective reasoning.

aim to cover a wider range of additional domains that benefit from reasoning, such as coding, science, and logic. Specifically, for questions from several scientific domains, we select OlympicArena (Huang et al., 2024b). For logic and coding, we select AGIEval (Zhong et al., 2023) and LiveCodeBench v4 (Jain et al., 2024), respectively. We select NuminaMATH (LI et al., 2024), where we randomly select a subset of 20,000 samples, and OmniMath (Gao et al., 2024) as sources of quantitative reasoning problems for our seed data, resulting a total of 31,586 question-answer pairs.

**Generate synthetic traces** We use the seed data to generate high-quality synthetic reasoning traces via state-of-the-art RLMs and LMs. For ST, we choose R1 as, at the time of writing, it is the best performing RLM that provides its full reasoning traces as part of its response. For ST-HC, we choose GPT-4o as our teacher model – using the prompt structure specified in



figure 3. We perform up to five rollouts per seed sample, discarding incorrect responses, stopping if the model provides the correct answer. The rollouts are done in a zero-shot manner, i.e. we provide only the question to the model and do not keep previous attempts as context.

**Filtering synthetic samples** First, we filter out samples that are of short length (less than 50 tokens for the entire trace) to prioritize sample quality, as shorter samples typically contain few pivots and are therefore not as impactful in training. In an effort to provide a fairer comparison, we align the datasets to the same number of samples by downsizing the larger dataset. As the final ST-HC dataset contained fewer samples ( $N=18,242$ ), we downsample the larger ST dataset to match this size, randomly removing questions not present in ST-HC and ensuring both datasets used for finetuning contain an equal number of samples. This results in a final 18K samples for both ST and ST-HC, each in the form of a triple (question, reasoning trace, answer). We provide a more detailed overview of the resulting datasets in appendix C.

### 3.3 Experimental details

**Model finetuning** We finetune a range of already instruction-tuned base LMs using our contributed reasoning datasets. For this, we select recent models of different families and sizes, namely: Llama 3.2 3B (Meta, 2024), Ministral 8B (Jiang et al., 2023), and Qwen2.5 32B (Qwen et al., 2025). We choose these models as they are all high performing for their parameter count, come with open-source weights, and have permissive licenses. The learning rate used during SFT varies for each LM in line with the model’s parameter count. For 3B models we use a peak learning rate of  $6 \times 10^{-5}$ , for 8B models we use  $4 \times 10^{-5}$ , and for 32B models we use  $1 \times 10^{-5}$ . For all models we use an effective batch size of 16. All models are trained for five epochs using a linear warmup for the first 10% of steps followed by cosine annealing. We use the AdamW optimizer (Loshchilov & Hutter, 2019) with  $\beta_1 = 0.9$  and  $\beta_2 = 0.95$  and a weight decay of  $1 \times 10^{-4}$ . Training is performed on a system comprising 8 Nvidia H100 GPUs using bfloat16 precision.

**Baselines** To contextualize our findings, we evaluate (1) the base instruction-tuned model, (2) the model after SFT on ST, and (4) the model after SFT on ST-HC. Additionally, to gauge the impact of the particular style replication we propose versus distilling from regular CoT, we add another baseline, where we instruct the generating LLM to think step-by-step (SBS), following Kojima et al. (2023). We then use the resulting CoT for distillation – similarly to how we use the emergent (ST) and synthetic (ST-HC) reasoning traces. This comprehensive evaluation allows us to isolate the impact of our hard-coded reasoning approach across different model sizes and compare it against the reasoning capabilities of the model itself. For completeness, we evaluate the teacher models used to generate our datasets as well.

**Evaluation** To evaluate the reasoning capabilities of all models, we select challenging, widely used benchmarks that test reasoning capabilities. These include MATH500 (Lightman et al., 2023), AIME2024, and GPQA (Rein et al., 2024); covering math and a range of scientific domains. Specifically, MATH500 and AIME2024 feature 500 and 30 competition math problems, respectively. GPQA consists of 198 questions from a range of scientific fields such as Biology and Chemistry. We focus only on the hardest (“Diamond”) subset of this particular dataset.

## 4 Results and discussion

### 4.1 Hard-coded reasoning results

The performance across all combinations of models and datasets evaluated in our study is presented in table 1. Our results clearly indicate that models finetuned with structured reasoning traces, whether emergent (ST) or synthetic (ST-HC), consistently and significantly outperform their baseline instruction-tuned counterparts across all benchmarks. Notably, even the smaller 3B and 8B parameter models exhibit substantial performance improvements. For instance, the Llama 3.2 3B model gains over 31 absolute percentage points on

Model	Variant	Params	MATH500	AIME2024	GPQA (D)
Llama 3.2	Base	3B	36.4	6.7	26.3
	SBS	3B	45.8	10.0	28.3
	ST	3B	68.4	23.3	31.3
	ST-HC	3B	64.2	16.7	29.3
Ministral	Base	8B	52.8	10.0	28.8
	SBS	8B	60.6	16.7	31.3
	ST	8B	78.2	33.3	38.9
	ST-HC	8B	77.0	33.3	34.8
Qwen2.5	Base	32B	76.8	16.7	49.0
	SBS	32B	78.2	20.0	49.5
	ST	32B	89.0	53.3	56.1
	ST-HC	32B	83.4	46.7	53.0
Teacher Models	R1	671B	96.8	76.7	71.7
	GPT-4o	-	75.4	13.3	53.0
	GPT-4o HC	-	81.2	16.7	55.1

**Table 1.** Performance comparison of language models finetuned on reasoning traces. We compare base models against versions finetuned on SBS (step-by-step thinking GPT-4o), ST (emergent traces from R1), and ST-HC (synthetic traces from GPT-4o with the prompt in figure 3). This evaluates the impact of reasoning trace style on downstream performance across model scales and benchmarks. GPT-4o HC refers to the base LM prompted with our structure from figure 3. All results are pass@1 accuracy as a percentage.

MATH500 when finetuned on ST, and the Ministral 8B model sees its AIME2024 score more than triple with either ST or ST-HC finetuning. These findings demonstrate that smaller models can markedly benefit from structured reasoning finetuning, challenging previous suggestions of minimal improvements for models of this scale (Team, 2025). Comparing the two fine-tuning approaches, we observe that models trained on synthetic traces achieve performance that approaches, and in some cases matches those trained on emergent RLM traces. This demonstrates that replicating the style of reasoning, even using a less capable teacher model guided by our prompt, is highly effective for distilling reasoning capabilities. Finally, the results show that while generating synthetic data with a SBS prompt is beneficial compared to the base model, there is a significant performance gap between SBS and our ST and ST-HC methods. This demonstrates that the performance improvements are indeed substantially driven by the specific stylistic patterns we identified, rather than solely by distilling the generating LLM’s general reasoning abilities.

When analyzing the thinking process during evaluation, we observe a clear correlation between successful reasoning and longer reasoning traces, as illustrated in figure 4. Models finetuned on either emergent or synthetic reasoning traces consistently produce substantially longer outputs compared to their respective base models across all evaluation benchmarks. This suggests that adopting and replicating a structured, elaborate reasoning style – whether learned from emergent traces or via synthetic ones – is a key mechanism driving the enhanced downstream reasoning capabilities we observed. Notably, figure 4 shows that finetuning on ST yields slightly longer reasoning traces than ST-HC, which correlates with their relative performance to one another – corroborating that a larger number of tokens spent thinking typically correlates with improved reasoning (Muennighoff et al., 2025).

## 4.2 Ablative study

To further isolate the impact of reasoning style versus factual correctness, we conduct an ablation study using two modified datasets. First, we create ST-HC-W by adapting the synthetic ST-HC traces to retain their stylistic structure while leading to incorrect final answers generated by GPT-4o-mini (details given in appendix D). Second, we create ST-NT, which contains only the question-answer pairs from the original ST data, removing the reasoning traces entirely. The results, presented in table 2, reveal several key insights. Models finetuned on ST-HC-W consistently outperform the base instruction-tuned models

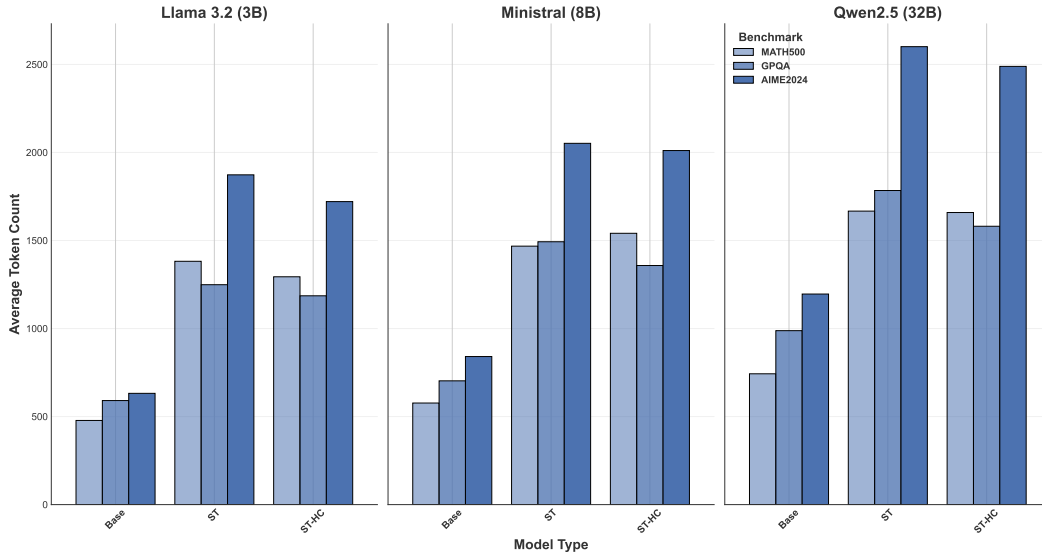


Figure 4: Average token count by model family and training dataset across evaluations.

across all evaluation benchmarks. This demonstrates that learning the stylistic patterns of reasoning enhances problem-solving capabilities even when the training data’s final conclusion is incorrect. However, ST-HC-W models perform noticeably worse than those trained on the stylistically similar but factually correct ST-HC dataset, confirming the value of accurate data. Unsurprisingly, models trained on ST-NT show only modest gains over the base models, as they do not learn to generate additional thinking tokens in the style of an RLM during inference.

These ablations confirm that while factual correctness is important for optimal performance, the stylistic patterns inherent in reasoning traces play a critical role in enhancing the reasoning abilities of distilled models. Crucially, we do not claim that style alone improves reasoning as the body of the traces still contains correct reasoning up until the answer, but rather emphasize that style is essential for improving reasoning capabilities. Thus, unlike traditional distillation methods relying predominantly on sample correctness, our results suggest that explicitly transferring a specific output structure is important.

## 5 Related work

**Language model reasoning** Language model reasoning has received increasing interest in recent years (Cobbe et al., 2021). Initially, LMs were conditioned on reasoning examples during pretraining, post-training or in-context to improve their reasoning capabilities (Zhang et al., 2022; Wang et al., 2023; Wei et al., 2023; Li et al., 2024), but test-time scaling (Snell et al., 2024; Muennighoff et al., 2025) has introduced a new paradigm for improving LM reasoning. Further, DeepSeek-AI et al. (2025) explore training RLMs with reinforcement learning, finding that they mimic human reasoning processes like self-reflection and verification. The resulting RLMs provide reasoning traces as training data to enhance LM reasoning ability. Our work demonstrates that the structural and lexical patterns in these traces play a significant role in improving reasoning performance, offering insights into what is actually transferred during reasoning distillation.

**Generalization in language models** Generalization in language models implies the ability to tackle unseen problems rather than simply reciting training data (Kang et al., 2024). LMs struggle with problems that differ from their training distribution, often closely following observed patterns down to individual terms (Razeghi et al., 2022). Recent research shows LMs rely heavily on memorized patterns rather than developing generalizable reasoning capabilities (Schwarzschild et al., 2024). This pattern-matching behavior is especially evident



Model	Variant	Parameters	MATH500	AIME2024	GPQA Diamond
Llama 3.2	ST-HC-W	3B	48.2	10.0	28.4
	ST-NT	3B	40.6	6.7	26.9
Ministral	ST-HC-W	8B	62.8	20.0	29.9
	ST-NT	8B	56.2	13.3	30.0
Qwen2.5	ST-HC-W	32B	80.2	26.7	51.3
	ST-NT	32B	78.8	20.0	49.8

**Table 2.** Ablation study results evaluating the distinct contributions of reasoning trace style and answer correctness. We compare model performance after finetuning on: (1) ST-HC-W, featuring stylistically consistent synthetic traces from ST-HC but deliberately leading to *incorrect* answers, and (2) ST-NT, containing only the question-answer pairs from ST *without* reasoning traces. All results are pass@1 accuracy as a percentage.

in mathematical reasoning, where LMs show variance across different instantiations of the same question and declining performance when only numerical values change (Mirzadeh et al., 2025). Data contamination has also been identified as a source of apparent but false generalization (Jiang et al., 2024). Our work extends this discussion by investigating how stylistic patterns in reasoning traces influence model performance, revealing that structural elements of reasoning may be as important as factual content for enhancing problem-solving capabilities.

**Reasoning distillation** Distillation has long been used to improve various aspects of machine learning models (Schmidhuber, 1992; Hinton et al., 2015; Sanh et al., 2020). For reasoning tasks with verifiable solutions, researchers have implemented rejection sampling methodologies that extract and validate advanced models’ reasoning processes (Zelikman et al., 2022). More recently, significant performance improvements have been achieved through SFT on synthetic datasets generated by superior LMs (Gunasekar et al., 2023). With the emergence of RLms, reasoning performance of instruction-tuned LMs can be substantially enhanced by finetuning on high-quality reasoning traces (DeepSeek-AI et al., 2025). Multiple efforts have demonstrated success in distilling RLms via SFT on reasoning traces that contain step-by-step thinking (Min et al., 2024; Huang et al., 2024a; Team, 2025; Labs, 2025; Xu et al., 2025). Our work attempts to decompose what is actually being transferred during this distillation process, revealing that stylistic elements of reasoning traces contribute significantly to performance gains independent of their factual correctness.

## 6 Conclusion

Our work challenges existing assumptions by demonstrating that improvements in reasoning capabilities in distilled language models can be driven by stylistic replication. By analyzing reasoning traces through the lens of established principles from cognitive science on how humans solve problems, we identified structural patterns and key lexical markers (*pivots*) characteristic of effective LM reasoning. We validated these findings by synthetically replicating the stylistic patterns found in emergent reasoning traces using a weaker model, achieving comparable downstream reasoning performance after distillation. Ablation studies confirmed that even reasoning traces with incorrect solutions enhance downstream performance, highlighting the effectiveness of stylistic mimicry in distilled models. Collectively, our results contribute to a deeper understanding of reasoning in language models and offer practical insights into efficient finetuning strategies and data generation practices.

## References

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=Uuf2q9TfXGA>.

- Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5185–5198, 2020. URL <https://aclanthology.org/2020.acl-main.463/>.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21*, pp. 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL <https://doi.org/10.1145/3442188.3445922>.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL <https://arxiv.org/abs/2005.14165>.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukas Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-math: A universal olympiad level mathematic benchmark for large language models, 2024. URL <https://arxiv.org/abs/2410.07985>.

- Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. URL <https://arxiv.org/abs/2306.11644>.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL <https://arxiv.org/abs/1503.02531>.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL <https://arxiv.org/abs/2203.15556>.
- Xin Huang, Rui Xu, Jiawen Gu, Yuxi Jiang, Lingfeng Zhao, Zhenguo Jiang, Wanjuan Yin, Daxin Jiang, Xipeng Huang, and Xipeng Qiu. O1: The journey of replication. *arXiv preprint arXiv:2406.04692*, 2024a. URL <https://arxiv.org/abs/2406.04692>.
- Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, Yikai Zhang, Yuqing Yang, Ting Wu, Binjie Wang, Shichao Sun, Yang Xiao, Yiyuan Li, Fan Zhou, Steffi Chern, Yiwei Qin, Yan Ma, Jiadi Su, Yixiu Liu, Yuxiang Zheng, Shaoting Zhang, Dahua Lin, Yu Qiao, and Pengfei Liu. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai, 2024b. URL <https://arxiv.org/abs/2406.12753>.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. Mistral 7b, 2023. URL <https://arxiv.org/abs/2310.06825>.
- Minhao Jiang, Ken Ziyu Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. Investigating data contamination for pre-training language models, 2024. URL <https://arxiv.org/abs/2401.06059>.
- Katie Kang, Amrith Setlur, Dibya Ghosh, Jacob Steinhardt, Claire Tomlin, Sergey Levine, and Aviral Kumar. What do learning dynamics reveal about generalization in llm reasoning?, 2024. URL <https://arxiv.org/abs/2411.07681>.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL <https://arxiv.org/abs/2001.08361>.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL <https://arxiv.org/abs/2205.11916>.
- Bespoke Labs. Bespoke-stratos: The unreasonable effectiveness of reasoning distillation, 2025. URL <https://www.bespokelabs.ai/blog/bespoke-stratos-the-unreasonable-effectiveness-of-reasoning-distillation>. Accessed: 2025-01-22.
- Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. Common 7b language models already possess strong math capabilities, 2024. URL <https://arxiv.org/abs/2403.04706>.

- Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath, 2024. URL [https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina\\_dataset.pdf](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf).
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step, 2023. URL <https://arxiv.org/abs/2305.20050>.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL <https://arxiv.org/abs/1711.05101>.
- Meta. Llama 3.2 lightweight models (1b/3b), 2024. URL <https://www.llama.com/docs/model-cards-and-prompt-formats/llama3.2/>. Accessed: 2025-01-22.
- Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems, 2024. URL <https://arxiv.org/abs/2412.09413>.
- Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=AjXkRZIVjB>.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL <https://arxiv.org/abs/2501.19393>.
- Allen Newell and Herbert Alexander Simon. *Human problem solving*. Prentice-Hall, 1972.
- OpenAI. Learning to reason with llms, 2024. URL <https://openai.com/index/learning-to-reason-with-llms/>.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.
- Yasaman Razeghi, Robert L. Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot reasoning, 2022. URL <https://arxiv.org/abs/2202.07206>.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=Ti67584b98>.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL <https://arxiv.org/abs/1910.01108>.
- Jürgen Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 03 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.2.234. URL <https://doi.org/10.1162/neco.1992.4.2.234>.
- Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zack Lipton, and Zico Kolter. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint*, 2024.

- Murray Shanahan. Talking about large language models. *Commun. ACM*, 67(2):68–79, January 2024. ISSN 0001-0782. doi: 10.1145/3624724. URL <https://doi.org/10.1145/3624724>.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL <https://arxiv.org/abs/2408.03314>.
- NovaSky Team. Sky-t1: Train your own o1 preview model within \$450. <https://novasky-ai.github.io/posts/sky-t1>, 2025. Accessed: 2025-01-09.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL <https://arxiv.org/abs/2203.11171>.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL <https://arxiv.org/abs/2201.11903>.
- Haotian Xu, Xing Wu, Weinong Wang, Zhongzhi Li, Da Zheng, Boyuan Chen, Yi Hu, Shijia Kang, Jiaming Ji, Yingying Zhang, Zhijiang Guo, Yaodong Yang, Muhan Zhang, and Debing Zhang. Redstar: Does scaling long-cot data unlock better slow-reasoning systems?, 2025. URL <https://arxiv.org/abs/2501.11284>.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STar: Bootstrapping reasoning with reasoning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL [https://openreview.net/forum?id=\\_3ELRdg2sgI](https://openreview.net/forum?id=_3ELRdg2sgI).
- Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models, 2022. URL <https://arxiv.org/abs/2210.03493>.
- Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models, 2023. URL <https://arxiv.org/abs/2304.06364>.



## A Emergent Reasoning Trace Example

We give an example of a reasoning trace in figure 5. Here, a RLM answers a mathematical puzzle question, initially producing an extensive reasoning trace with frequent backtracking, followed by giving the final correct answer.

**Reasoning Trace Example**

**Question:** A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

---

**Reasoning:** Okay, let's see. The problem says a juggler can juggle 16 balls. Half of them are golf balls, and half of the golf balls are blue. I need to find out how many blue golf balls there are. Alright, let me break this down step by step.

First, the total number of balls is 16. Half of those are golf balls. So, half of 16 would be... 16 divided by 2, which is 8. So, there are 8 golf balls. Got that down. Now, out of these 8 golf balls, half are blue. Hmm, so again, half of 8 would be 8 divided by 2. Let me calculate that. 8 divided by 2 is 4. So, does that mean there are 4 blue golf balls? That seems right, but let me double-check to make sure I didn't skip any steps or misunderstand the question.

Wait, the problem says half of the balls are golf balls. So total balls: 16. Half are golf balls: 8. Then half of the golf balls are blue. So half of 8 is indeed 4. So, 4 blue golf balls. Let me just visualize this. If there are 16 balls in total, and half (so 8) are golf balls, and of those 8 golf balls, half are blue and half are another color. So splitting 8 into two equal parts gives 4 and 4. So yes, 4 blue golf balls. That makes sense.

Is there another way to approach this problem? Maybe by using fractions. Let's see. Total balls = 16. Half are golf balls: so  $1/2$  of 16. Then half of those are blue: so  $1/2$  of  $1/2$ , which is  $1/4$ . So  $1/4$  of the total balls are blue golf balls.  $1/4$  of 16 is 4. Yep, same answer. So whether I do it step by step or use fractions, I still get 4. That seems consistent.

Alternatively, if I were to write equations, let's denote total balls as  $T = 16$ . Number of golf balls  $G = T / 2$ . Then number of blue golf balls  $B = G / 2$ . Substituting,  $B = (16 / 2) / 2 = 8 / 2 = 4$ . Same result. So all methods lead to 4.

I think that's solid. Let me just check if I interpreted "half of the golf balls are blue" correctly. The problem didn't mention anything about the other half, but since half of the golf balls are blue, the other half could be any other color, but that's irrelevant to the question. The question only asks for the number of blue golf balls, so 4 is the answer.

Wait, just to be thorough, the problem doesn't have any trick wording, right? Like, when it says half of the balls are golf balls, that's straightforward. Then half of those golf balls are blue. So no, no tricky parts. If it had said half of the blue balls are golf balls, that would be different, but it's the other way around. Half of the golf balls are blue. So straightforward.

Another check: total golf balls are 8. If 4 are blue, then 4 are not blue. That adds up to 8. Then the other 8 balls (non-golf balls) are something else entirely. But since the question is only about blue golf balls, we don't need to consider those.

I think that's all. All angles checked. So, 4 blue golf balls. Confident that's correct.

**Final Answer:** The number of blue golf balls is 4.

Figure 5: A detailed example of mathematical problem-solving from R1 with a full reasoning trace.

## B Emergent Trace Analysis

This appendix provides quantitative details from the analysis performed on 16,710 successful emergent reasoning traces generated by R1 model, taken from Labs (2025). This analysis

informs the discussion of pivot types and reasoning stages in Section 3.1. On average, these traces are substantial in length, comprising 2,713 tokens. Structurally, they contain an average of 141.7 paragraphs. Metacognitive activity is frequent, with an average of 93.5 identified pivots occurring within each reasoning trace.

### B.1 Details of Categorizing the Reasoning Trace Contents

To analyze reasoning traces, we developed a systematic approach for identifying pivots and reasoning stages using regular expression pattern matching. For each of the four pivot categories (Realization, Verification, Exploration, and Integration), we created comprehensive regex patterns capturing lexical markers that signal metacognitive transitions – for instance, phrases such as “Actually” for Realization pivots or “Therefore” for Integration pivots. Similarly, we defined patterns for the four reasoning stages (Problem Framing, Exploration, Verification, and Synthesis) based on characteristic expressions and structural elements. We experimented with using an LM in the form of GPT-4o mini for this task but found it to be no more performant. This framework enabled automated extraction and quantification of reasoning elements across all 16,710 traces. While matching to regular expressions is not a perfect way to extract this information, we found it to perform well considering the very standardized nature of reasoning traces. We iteratively refined our patterns over multiple rounds after manual inspection of reasoning traces by the authors, ensuring balanced detection across all categories. Each trace was analyzed for both the frequency of pivot occurrences and the presence of reasoning stages, allowing us to quantify both the metacognitive transitions and the structural patterns that characterize effective reasoning.

### B.2 Pivot Analysis

**Pivot Diversity** A key indicator of complex reasoning is the variety of metacognitive shifts employed. The analyzed traces show an average diversity of 3.51 distinct pivot types per trace. Furthermore, a very high majority, 96.1% of the traces, contains at least three different pivot categories. This high percentage strongly confirms the observation that successful traces typically involve multiple forms of reflection and correction during the reasoning process.

**Pivot Type Frequencies** The frequency and prevalence of each specific pivot type across the dataset are summarized in Table 3. Notably, Integration and Realization pivots remain extremely common, appearing in nearly all analyzed traces. Exploration pivots are the least prevalent, present in 87.0% of traces, indicating consideration of alternative paths or hypotheses does not occur as frequently.

Pivot Type	Avg. Occurrences per Trace	% Traces Present
Realization	18.96	98.6%
Exploration	16.11	87.0%
Verification	1.37	89.6%
Integration	67.64	100.0%

**Table 3.** Frequency and prevalence of identified pivot types within the analyzed emergent reasoning traces (N=16,710).

### B.3 Reasoning Stage Analysis

The analysis also quantifies the presence of segments corresponding to the four cognitive reasoning stages discussed in section 3.1. The difference compared to the pivot types analysis is that a single reasoning stage can contain multiple pivots, even of separate types. In this case, we ascribe the trace to the reasoning stage found at the beginning of the trace. The average occurrences and prevalence of these stages are detailed in table 4. Synthesis stages, often comprising multiple steps or integration points, are universally present. Problem

Framing, Verification, and Exploration stages are all highly prevalent, appearing in the vast majority of traces, consistent with the high frequency of their corresponding pivot types.

Reasoning Stage	Avg. Occurrences per Trace	% Traces Present
Problem Framing	3.13	79.2%
Exploration	6.81	87.0%
Verification	3.34	89.6%
Synthesis	84.08	100.0%

**Table 4.** Frequency and prevalence of identified reasoning stages within the analyzed emergent reasoning traces (N=16,710).

#### B.4 Key Insights from Analysis

The quantitative analysis reinforces the qualitative observations presented in the main paper. Successful emergent reasoning traces consistently utilize a diverse range of pivot types, averaging 3.51 distinct types per trace, with an overwhelming 96.1% using three or more, indicating frequent and varied metacognitive adjustments. Furthermore, the presence and frequency of identified reasoning stage segments, detailed in Table 4, generally align with the cognitive science framework encompassing problem framing, exploration, verification, and synthesis. The high prevalence of all four stage types underscores the iterative nature of the observed reasoning process.

Data Source	Number of Samples
OlympicArena (Huang et al., 2024b)	4,250
AGIEval (Zhong et al., 2023)	2,385
LiveCodeBench v4 (Jain et al., 2024)	713
NuminaMATH (LI et al., 2024)	20,000
OmniMath (Gao et al., 2024)	4,238
<b>Total</b>	<b>31,586</b>

**Table 5.** Composition of the seed data pool used for generating reasoning traces. The number of samples from NuminaMATH reflects a randomly selected subset.

## C Dataset Statistics

This section provides detailed statistics about the datasets created and utilized in our study, namely the seed data pool and the derived reasoning trace datasets SMOLTRACES (ST) and SMOLTRACES-HARDCODED (ST-HC).

### C.1 Seed Data Compilation

The foundation for generating our reasoning trace datasets is a curated collection of question-answer pairs sourced from diverse benchmarks spanning mathematics, science, logic, and coding. As detailed in Section 3.2, we selected problems from OlympicArena (Huang et al., 2024b), AGIEval (Zhong et al., 2023), LiveCodeBench v4 (Jain et al., 2024), NuminaMATH (LI et al., 2024), and OmniMath (Gao et al., 2024). This process resulted in a final seed dataset comprising 31,586 unique question-answer pairs. Table 5 presents a breakdown of the sources contributing to this seed data pool. To ensure the integrity of our downstream evaluations, we performed decontamination on this initial pool by removing any questions overlapping with our chosen evaluation benchmarks (MATH500, AIME2024, GPQA Diamond).

Statistic	Dataset Values
Trace Origin	ST: Emergent (R1) ST-HC: Synthetic (GPT-4o + Hard-coded Prompt, Fig. 3)
Seed Questions Source	ST: Shared Pool (Table 5, N=31,586) ST-HC: Shared Pool (Table 5, N=31,586)
Final Number of Samples	ST: 25,802 (adjusted to N=18,242 after) ST-HC: 18,242
Filtering Applied	ST: Correct Answer (up to 5 attempts), Min. Length (50 tokens) ST-HC: Correct Answer (up to 5 attempts), Min. Length (50 tokens)
Avg. Trace Length (Tokens)	ST: 2,521 ST-HC: 2,101
Avg. Pivots per Trace	ST: 93.4 ST-HC: 89.1

**Table 6.** Summary statistics for the final generated reasoning trace datasets used in fine-tuning. Values reflect the approximately 18,250 filtered and balanced samples in each dataset (ST and ST-HC). Average length and pivot counts are illustrative; actual values depend on the final composition.

## C.2 Generated Reasoning Trace Datasets

Using the curated seed data, we generated two parallel datasets featuring detailed reasoning traces, as described in Section 3.2. Table 6 provides a summary comparing key statistics of the final, balanced ST and ST-HC datasets. Average trace length and pivot counts reflect measurements across these final samples.

**SMOLTRACES (ST):** This dataset contains emergent reasoning traces generated by the state-of-the-art RLM R1 (DeepSeek-AI et al., 2025). For each seed question, we prompted R1 in a zero-shot manner up to five times, retaining the first trace that yielded the correct final answer.

**SMOLTRACES-HARDCODED (ST-HC):** This dataset comprises synthetic reasoning traces generated using GPT-4o, guided by the structured prompt detailed in figure 3 (Figure 3). This prompt enforces the inclusion of specific structural elements and lexical pivots identified in our analysis (Section 3.1). Similar to ST, generation involved up to five zero-shot attempts per seed question, keeping the first correct trace.

**Balancing:** To ensure a fair comparison in finetuning experiments, we balanced the datasets by size. As the ST-HC dataset contained fewer samples after filtering, we downsample the larger ST dataset by randomly removing samples that are not present in the smaller dataset until both datasets have an equal number of samples. Therefore, the final versions of both ST and ST-HC used for finetuning contain an equal number of samples (N=18,242).

## D Synthetic Traces with Wrong Answer Details

This section details the construction process for the ST-HC-W dataset, used in our ablation study (Section 4.2) to investigate the impact of reasoning trace style independent of final answer correctness. The goal is to create a dataset that retains the stylistic and structural characteristics of the synthetic ST-HC traces but deliberately leads to an incorrect final answer. First, we use the ground truth answer associated with the seed question for each sample in ST-HC. Then, we prompt GPT-4o-mini, instructing it to provide a different, incorrect answer that is similar in format (e.g. “9.11” instead of “9.9”). We compare the answer generated by GPT-4o-mini against the known correct answer and if the generated answer matches the correct one, we repeat the procedure.

The resulting ST-HC-W dataset mirrors the size of ST-HC, containing 18,242 samples. Each sample includes the original question, a reasoning trace stylistically similar to ST-HC, but which concludes with an incorrect final answer. This construction allows for better isolation of the effect of learning stylistic reasoning patterns during finetuning.