POST-REASONING: IMPROVING THE PERFORMANCE OF NON-THINKING MODELS AT NO COST

Anonymous authors
Paper under double-blind review

ABSTRACT

As the widespread adoption of Large Language Models (LLMs) accelerates, token consumption from intermediate reasoning traces increasingly contributes to inference latency and operational cost. Recent studies suggest that many real-world tasks require little to no explicit reasoning, with additional reasoning sometimes even degrading performance. In this work, we propose **Post-Reasoning**, a simple yet effective approach that improves instruction-tuned models by conditioning them to justify their answers after generating the final response. By design, it enables the final answer to be obtained without additional latency or token cost, while still improving performance through simple instruction augmentation. We evaluate Post-Reasoning across 117 model–benchmark settings spanning 13 open and proprietary models, 7 model families, and 9 diverse reasoning and knowledge-intensive benchmarks, including AMC, HMMT, GSM8K, GPQA, MMLU-Pro, and BIG-Bench Hard. Post-Reasoning improves performance in over 87.18% of evaluated settings, achieving a mean relative improvement of 16.30%. Furthermore, we propose supervised post-reason tuning, which further improves performance in over 91.11% of evaluated settings, and exceeds the prompt-based post-reasoning baseline by an average of 8.01%, demonstrating that post-reasoning can be effectively internalized through training. Ultimately, Post-Reasoning establishes a new performance ceiling for direct-answer capabilities.

1 INTRODUCTION

As Large Language Model (LLM) adoption accelerates, with over a billion people using AI in some capacity (Microsoft AI Economy Institute, 2025), token consumption has surged (Aubakirova et al., 2025), making efficient inference a critical operational challenge (OpenAI, 2025). Beyond the growing user base, token consumption is further amplified by reasoning traces (intermediate tokens) generated prior to producing final responses (Deloitte, 2026). Consequently, maximizing the number of queries served under a fixed token budget is emerging as a central optimization problem for both LLM-based system providers (e.g., Anthropic, OpenAI, Lovable) and consumers.

In reasoning-enabled models, a substantial portion of the token budget is consumed by reasoning traces rather than the final answer, leading to increased inference cost and latency per task (Sui et al., 2025; Chen et al., 2024; Luo et al., 2025b; Yeo et al., 2025; Han et al., 2024; Ma et al., 2025; Hao et al., 2024). Such models can consume on the order of $10\times$ more tokens without guaranteed performance gains (Srivastava et al., 2026). Real-world tasks for which LLMs are used lie on a spectrum of token consumption, where logic-intensive tasks such as advanced mathematics (e.g., MATH, AIME) and complex coding benchmarks (e.g., HumanEval, Codeforces) benefit from extended reasoning and occupy the higher end of the spectrum. In contrast, simpler benchmarks (e.g., GSM8K) and tasks such as summarization, factual question answering, and basic arithmetic often require little to no reasoning, with reduced reasoning sometimes matching or outperforming full reasoning (Li et al., 2025; Aggarwal et al., 2025), placing them toward the lower end; Appendix A provides a practical decision rule based on task complexity and the available latency and token budget.

In this work, we propose a novel approach, **Post-Reasoning**, that improves the performance of instruction-tuned models, i.e., models that directly generate responses without explicit reasoning traces and therefore lie on the extreme left of the token spectrum. Post-Reasoning conditions a model to generate additional tokens after producing the final answer and, unlike pre-reasoning approaches,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

does not incur additional latency or token cost before the answer is reached. Post-reasoning behavior can be induced in two ways: (1) through prompting, by instructing the model to first provide the answer and then justify it (e.g., "What is $10 + 2$? State the final answer immediately and justify your answer."), and (2) through supervised post-reason tuning, where the behavior is internalized into the model weights. Furthermore, the post-reasoning generation can optionally be omitted at inference time through stop-token-based decoding, allowing the final answer to be obtained without generating the additional justification tokens.

Experiments were conducted across 13 open-source and proprietary models spanning different scales (4B–70B) and model families (Llama, Gemma, Mistral, Qwen, Gemini, Claude, GPT). Across all evaluations Figure 1, Post-Reasoning improves model performance in 88.19% of settings, yielding average gains of 24.99% on competition mathematics (AMC 8/10/12), 23.11% on olympiad-style reasoning (HMMT), 1.13% on standard arithmetic reasoning (GSM8K), 10.14% on graduate-level scientific reasoning (GPQA Main), 4.45% on multi-domain knowledge reasoning (MMLU-Pro), and 3.16% on multi-step logical reasoning (BBH). Furthermore, supervised post-reason tuning on 10 open-source models demonstrates strong generalization, improving performance in 91.11% of evaluated settings and achieving significant gains across benchmarks (48.90% on AMC, 41.97% on HMMT, 2.80% on GSM8K, 12.65% on GPQA Main, 4.26% on MMLU-Pro, and 2.16% on BBH). Notably, post-reason tuned models outperform prompt-based Post-Reasoning in 74.44% of evaluated settings, suggesting that post-reasoning behavior can be effectively internalized through supervised training. In summary, our core contributions are as follows:

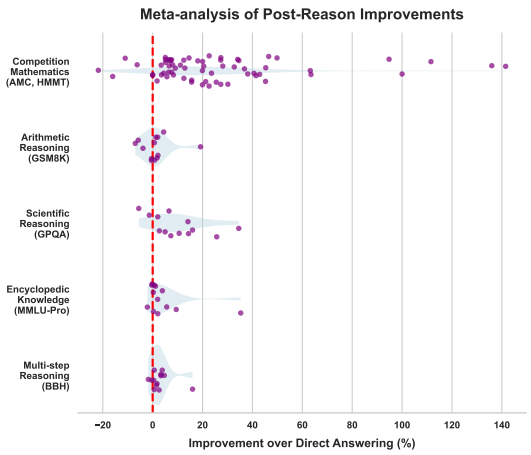


Figure 1: Meta-analysis of Post-Reasoning improvements across benchmarks. Each point denotes the relative gain over direct answering for a (model, task) pair. Gains are predominantly positive, with larger improvements on multi-step reasoning tasks (AMC, HMMT, MATH) and smaller or mixed gains on arithmetic and knowledge-intensive benchmarks.

- **Domain-generic Improvement on Instruct Models**: We introduce Post-Reasoning, a novel approach that improves the performance of instruction-tuned models without incurring additional inference latency or token cost for obtaining the final answer.
- **Post-Reason Prompting**: We demonstrate that conditioning a model to post-reason consistently improves direct-answer accuracy across both open and proprietary models. Across 13 models spanning 4B–70B parameters and multiple benchmark suites, post-reasoning improves performance in over 88.19% of evaluated settings, with gains exceeding 100% on several long-horizon reasoning tasks.
- **Supervised Post-Reason Tuning**: We propose a supervised post-reason tuning framework that optimizes models exclusively on post-answer justifications using masked loss optimization. Post-reason tuning further improves performance over prompt-based post-reasoning in over 75% of evaluated model–task combinations, while enabling strong cross-domain generalization beyond the training distribution.

Related Work. Chain-of-Thought (CoT) prompting (Wei et al., 2022) and subsequent reasoning-based approaches (Wang et al., 2023; Yao et al., 2023; Lightman et al., 2023; OpenAI, 2025; DeepSeek-AI, 2025) have demonstrated that generating intermediate reasoning traces can substantially improve performance on complex reasoning tasks. However, these gains often come at the cost of significantly increased inference-time token consumption, latency, and operational expense. Recent works therefore explore reasoning compression, pruning, and selective thinking strategies (Li et al., 2025; Aggarwal et al., 2025; Srivastava et al., 2026), showing that many tasks require little or no explicit reasoning and that excessive reasoning can sometimes degrade performance.

Our work is orthogonal to these approaches. Rather than reducing pre-answer reasoning traces, we investigate whether reasoning generated *after* the answer can still improve performance while avoiding additional latency before the final response. Furthermore, unlike prior reasoning SFT approaches that jointly train on reasoning traces and answers, our post-reason tuning framework supervises models only on post-answer justifications, enabling reasoning improvements without modifying the direct-answer generation process and yielding stronger cross-domain generalization beyond the training distribution.

2 METHODOLOGY

Let $\mathcal{V}$ denote the tokenizer vocabulary. Given an input prompt $x \in \mathcal{V}^D$, the model generates a response $y \in \mathcal{V}$ (for simplicity, treated as a single token). Let LM_θ denote a language model parameterized by θ, which maps an input sequence to a distribution over $\mathcal{V}$. In reasoning mode, the output y is conditioned not only on the input x, but also on two additional factors: (**F1**) an augmented instruction δ_t that prompts the model to reason (e.g., "What is $2+3+10$? *Think step by step*"), and (**F2**) auto-regressively generated reasoning tokens c_t (e.g., "$2+3=5$, $5+10=15$"), which are produced before the final answer. Formally,

$$c_t \sim \mathrm{LM}_\theta(\cdot \mid x, \delta_t), \quad y \sim \mathrm{LM}_\theta(\cdot \mid x, \delta_t, c_t). \tag{1}$$

Here, the second factor, i.e., c_t, is the primary contributor to increased latency and cost in obtaining the LLM response. In **Post-Reasoning**, we omit this factor from conditioning the answer and instead leverage the augmented instruction (F1) to help condition the response. To this end, we instruct the model to "state the final result first, then justify," denoted by δ_p:

$$y \sim \mathrm{LM}_\theta(\cdot \mid x, \delta_p), \quad c_p \sim \mathrm{LM}_\theta(\cdot \mid x, \delta_p, y). \tag{2}$$

Here, c_p denotes post-answer (post-justification) tokens. These tokens do not contribute to the response cost when generation is truncated immediately after producing the final answer y, for example by specifying a designated stop token (or end-of-answer marker) and terminating autoregressive decoding once this token is generated.[1]

2.1 PROMPT-BASED POST-REASONING

Similar to Chain-of-Thought prompting which asks model to first think then answer, we instruct models to generate justifications for their answers. The exact prompt templates and instruction formats are provided in Appendix B.6.

Post-Reason Prompt

```
<task-specific instruction> State the final answer
immediately and justify your answer.
```

2.2 SUPERVISED POST-REASON TUNING

To train the model for post-reasoning, we optimize the likelihood of post-reasoning tokens while masking the loss on the answer tokens. Specifically, given an input x, a target answer y, and post-reasoning tokens $c_p = (c_1, \ldots, c_T)$, the model is trained to predict the justification sequence conditioned on both the input and the answer:

$$\mathcal{L} = -\sum_{i=1}^{T} \log P_\theta(c_i \mid x, y, c_{<i}) \tag{3}$$

By restricting supervision to the post-reasoning tokens, the model is encouraged to generate coherent justifications conditioned on the answer, rather than overfitting to answer prediction itself.

[1] Optionally, generation can be continued beyond this point to produce c_p, providing interpretable justifications for the predicted outcome.

Intuition behind post-reasoning. In autoregressive next-token prediction, each generated token is conditioned on the preceding context. As discussed earlier, Chain-of-Thought (CoT) prompting introduces two forms of conditioning: $\mathbf{F1}$, the augmented instruction (e.g., "Think step by step"), and $\mathbf{F2}$, the generated reasoning tokens themselves(Wei et al., 2022). While prior work primarily attributes CoT improvements to the reasoning traces ($\mathbf{F2}$), the impact of the augmented instruction ($\mathbf{F1}$) remains underexplored. Post-Reasoning primarily leverages this instruction-level conditioning, suggesting that performance gains may arise not only from explicit reasoning traces, but also from how models are prompted (conditioned) to structure generation.

3 Experimental Setup

Models. To extensively study the impact of post-reasoning, we consider a range of open-source models across providers and sizes: 4B to 70B parameters: Qwen3.5 (4B, 9B, 27B) Yang et al. (2025), Llama-3.1 (8B) and Llama-3.3 (70B) Grattafiori et al. (2024), Gemma-3 (12B, 27B) Team et al. (2025), Ministral-3 (8B, 14B) Liu et al. (2026), and Mistral-Small (24B). Since the primary aim is to improve the performance of instruction-tuned models with no cost and latency trade-off, all the models chosen are either instruction-tuned or facilitate a turn-off thinking mode. Generations were constrained to a maximum of $2,048$ tokens for instruct and $4,096$ for Post-Reason models. We set temperature to 0.0, top-p to 0.8, and top-k to 20.

Benchmarks. We evaluate our approach on a diverse suite of benchmarks spanning multiple domains and levels of reasoning complexity. For standard mathematical reasoning, we use GSM8K (Cobbe et al., 2021). To assess performance on competition-level and structured problem-solving tasks, we evaluate on AMC (8/10/12) and Harvard-MIT Mathematics Tournament (HMMT) benchmarks (Ding et al., 2024). To further evaluate broader scientific, knowledge-intensive, and multi-step reasoning capabilities, we include GPQA (Rein et al., 2023), MMLU-Pro, and BIG-Bench Hard (BBH). Collectively, this benchmark suite spans a broad spectrum of task difficulty, ranging from standard arithmetic reasoning to complex scientific, logical, and long-horizon reasoning tasks.

Baseline. We compare post-reasoning against standard instruction-tuned models operating without reasoning. These baseline results are denoted as **Direct** in our experiments, while post-reason prompting results are denoted as **Post**. Notably, we do not directly compare against native pre-reasoning (CoT) models as the primary goal of Post-Reasoning is to improve the performance of standard instruction-tuned models without introducing additional inference latency or token cost. We nevertheless showed that Post-Reason complements native thinking on open-source models and low-effort thinking on proprietary models where token budget is strict in Appendix D; investigating this interaction is beyond the scope of the present study.

Table 1: AMC Competition Mathematics. Δ represents the relative percentage improvement. Direct denotes standard prompting. Post denotes the post reason-prompting.

	AMC 8			AMC 10			AMC 12		
Model	Direct	Post	Δ	Direct	Post	Δ	Direct	Post	Δ
Llama Family									
Llama-3.1 (8B)	7.09	**11.57**	+63.19%	7.42	**12.13**	+63.48%	4.43	**10.70**	+141.53%
Llama-3.3 (70B)	19.78	**28.73**	+45.25%	27.42	**33.03**	+20.46%	20.30	**26.94**	+32.71%
Gemma Family									
Gemma-3 (12B)	16.04	**17.16**	+6.98%	9.89	**13.26**	+34.07%	8.49	6.64	-21.79%
Gemma-3 (27B)	10.82	**15.30**	+41.40%	13.93	**17.08**	+22.61%	11.81	**15.13**	+28.11%
Mistral & Ministral									
Ministral-3 (8B)	13.06	**18.66**	+42.88%	10.11	**11.69**	+15.63%	9.23	7.75	-16.03%
Ministral-3 (14B)	17.54	**21.27**	+21.27%	12.36	**15.51**	+25.49%	7.01	**9.59**	+36.80%
Mistral-Small (24B)	7.09	**13.81**	+94.78%	12.13	**13.48**	+11.13%	6.27	**7.75**	+23.60%
Qwen Family									
Qwen3.5 (4B)	16.42	**20.90**	+27.28%	8.99	**10.79**	+20.02%	4.06	**5.17**	+27.34%
Qwen3.5 (9B)	18.43	**21.12**	+14.57%	12.36	**16.63**	+34.55%	8.12	**9.59**	+18.10%
Qwen3.5 (27B)	27.76	**30.67**	+10.48%	28.76	**32.36**	+12.52%	20.30	**20.66**	+1.77%

4 RESULTS AND DISCUSSIONS

Core Effect of Post-Reasoning. Across all 10 models from 4 families and sizes from 4B to 70B, the Post-Reason baseline outperforms the Direct Answer baseline in **85%** of evaluated cases, demonstrating that enforcing a post-reasoning sequence structure improves initial token generation without requiring any model updates.

4.1 MATHEMATICAL BENCHMARKS.

AMC Benchmarks. As shown in Table 1, post-reasoning yields consistent improvements across model families, with an average relative gain of **30.01%**. When stratified by scale, smaller models ($\leq$10B) achieve larger gains (**37.71%**) compared to mid-sized (10–30B, **31.65%**) and large models ($\geq$70B, **32.81%**), indicating that post-reasoning provides comparatively greater benefits for models with lower baseline accuracy under direct-answer prompting, while remaining effective across all scales.

The magnitude of improvement varies across models and tasks, ranging from modest gains to very large relative increases (e.g., up to **141.53%** on AMC 12 for Llama-3.1-8B), with occasional decreases (e.g., Gemma-3-12B and Ministral-3-8B on AMC 12). Despite this variability, improvements are observed across all AMC difficulty levels (AMC 8/10/12), indicating that post-reasoning acts as a general performance enhancement rather than a task-specific effect.

Competition Mathematics (HMMT). This trend extends to more challenging competition benchmarks, as shown in Table 2. On HMMT, post-reasoning improves performance in **18/20 cases (90%)**, with consistently large gains for several models (e.g., +100.00% for Llama-3.1-8B and +111.57% for Ministral-3-8B). These results reinforce that post-reasoning is particularly effective for long-horizon, multi-step reasoning tasks.

Standard Mathematics (GSM8K). In contrast, gains are more modest on GSM8K within the same table, with improvements observed in **6/10 models (60%)** and an average gain of **+1.17%**. This difference reflects task characteristics: AMC and HMMT problems require deeper, multi-step reasoning, whereas GSM8K involves shorter reasoning chains with fewer intermediate steps. Consequently, while post-reasoning remains beneficial, the available headroom is smaller when direct inference is already competitive.

4.2 GENERAL REASONING BENCHMARKS.

We further evaluate post-reasoning beyond mathematical tasks on a diverse set of reasoning and knowledge-intensive benchmarks, including GPQA, MMLU-Pro, and BIG-Bench Hard (Table 3). Across these benchmarks, post-reasoning improves performance in **24/30 cases (80%)**, demonstrating that its benefits extend across domains.

However, the magnitude of improvement is generally more modest compared to AMC and HMMT, with gains typically in the range of **1–10%**, and occasional larger improvements on specific models (e.g., +34.52% on GPQA for Qwen-3.5-9B). Improvements are more pronounced on GPQA, which requires deeper reasoning and domain-specific knowledge, whereas gains on MMLU-Pro and BIG-Bench Hard are smaller and more uniform. As observed earlier, gains vary across models and tasks, suggesting that the effectiveness of post-reasoning depends on both task complexity and model alignment. Altogether, these results indicate that the effectiveness of post-reasoning scales with the depth and complexity of reasoning required by the task, while remaining broadly beneficial across models and domains.

4.3 POST REASONING EFFECT ON PROPRIETARY MODELS

We further evaluate post-reasoning on proprietary frontier models, including Gemini-2.5-Flash, Claude-Haiku-4.5, and GPT-5.4-Mini, across the same set of benchmarks (Tables 4 to 6). Post-reasoning consistently improves performance across a majority of settings, demonstrating that its benefits are not limited to open-source models and generalize to commercially deployed systems.

On competition mathematics benchmarks (AMC and HMMT), post-reasoning yields consistent gains across all models, with improvements observed across nearly all tasks (e.g., up to **+22.65%** on AMC

Table 2: Mathematical Reasoning Benchmarks. Comparison of Direct and Post-Reason baselines across competition (HMMT) and standard (GSM8K) datasets. Δ denotes relative percentage improvement.

	HMMT Feb			HMMT Nov			GSM8K		
Model	Direct	Post	Δ	Direct	Post	Δ	Direct	Post	Δ
Llama Family									
Llama-3.1 (8B)	1.44	**2.88**	+100.00%	2.45	2.18	-11.02%	11.45	**13.65**	+19.21%
Llama-3.3 (70B)	6.73	**9.86**	+46.51%	7.36	**10.35**	+40.62%	34.27	**35.78**	+4.41%
Gemma Family									
Gemma-3 (12B)	3.61	**4.33**	+19.94%	5.18	**5.99**	+15.64%	21.30	**21.68**	+1.78%
Gemma-3 (27B)	5.77	**6.25**	+8.32%	5.72	**7.90**	+38.11%	31.92	31.77	-0.47%
Mistral & Ministral									
Ministral-3 (8B)	2.16	**4.57**	+111.57%	3.00	**4.36**	+45.33%	22.74	21.15	-6.99%
Ministral-3 (14B)	3.85	**4.33**	+12.47%	4.90	**5.18**	+5.71%	28.89	27.22	-5.78%
Mistral-Small (24B)	3.85	3.61	-6.23%	6.27	**7.08**	+12.92%	29.57	**29.95**	+1.29%
Qwen Family									
Qwen3.5 (4B)	3.12	**3.37**	+8.01%	2.72	**3.54**	+30.15%	19.33	**19.48**	+0.78%
Qwen3.5 (9B)	3.61	**4.33**	+19.94%	3.27	**4.90**	+49.85%	25.55	24.56	-3.87%
Qwen3.5 (27B)	8.41	8.41	+0.00%	10.63	**11.17**	+5.08%	46.70	**47.69**	+2.12%

Table 3: Evaluation on GPQA, MMLU-Pro, and BIG-Bench Hard. Post-reasoning yields consistent improvements across models, with variability depending on task characteristics. Δ denotes relative percentage improvement.

	GPQA			MMLU-Pro			BIG-Bench Hard		
Model	Direct	Post	Δ	Direct	Post	Δ	Direct	Post	Δ
Llama Family									
Llama-3.1 (8B)	29.66	**32.81**	+10.62%	27.47	**37.17**	+35.31%	42.84	**45.69**	+6.65%
Llama-3.3 (70B)	49.89	49.21	-1.36%	52.33	**53.37**	+1.99%	65.32	**66.52**	+1.84%
Gemma Family									
Gemma-3 (12B)	33.71	**34.61**	+2.67%	44.03	**44.17**	+0.32%	59.10	58.88	-0.37%
Gemma-3 (27B)	36.18	34.16	-5.58%	51.20	**51.33**	+0.25%	62.74	**63.83**	+1.74%
Mistral & Ministral									
Ministral-3 (8B)	36.18	**37.98**	+4.98%	48.20	48.03	-0.35%	54.86	**55.21**	+0.64%
Ministral-3 (14B)	37.08	**39.78**	+7.28%	54.23	**54.57**	+0.63%	56.98	**58.88**	+3.33%
Mistral-Small (24B)	37.75	**40.22**	+6.54%	53.33	**54.43**	+2.06%	59.88	**60.80**	+1.54%
Qwen Family									
Qwen3.5 (4B)	37.53	**42.92**	+14.36%	43.87	**45.57**	+3.88%	52.00	**54.42**	+4.65%
Qwen3.5 (9B)	43.60	**58.65**	+34.52%	50.60	**51.17**	+1.13%	56.07	**57.52**	+2.59%
Qwen3.5 (27B)	49.89	**62.70**	+25.68%	66.00	65.87	-0.20%	69.91	68.67	-1.77%

12 for Claude-Haiku-4.5 and **+27.31%** on HMMT Feb). These results reinforce earlier observations that post-reasoning is particularly effective for long-horizon, multi-step reasoning tasks.

On GSM8K, gains are smaller and less consistent, mirroring trends observed with open models, with improvements limited to **+2.02%** (Claude-Haiku-4.5) and **+0.70%** (GPT-5.4-Mini), and a slight decrease for Gemini-2.5-Flash.

On broader reasoning benchmarks (GPQA, MMLU-Pro, BIG-Bench Hard), post-reasoning improves performance in most cases, with moderate gains (e.g., **+15.99%** on GPQA for Gemini-2.5-Flash and **+15.98%** on BIG-Bench Hard for Claude-Haiku-4.5). Similar to earlier results, improvements are more pronounced on reasoning-intensive tasks (GPQA) compared to more knowledge-oriented benchmarks.

These results demonstrate that **post-reasoning generalizes across both open and proprietary models**, consistently improving performance, with the largest gains observed on tasks requiring deeper reasoning.

Table 4: Evaluation on proprietary models for AMC benchmarks. Post-reasoning consistently improves performance across models. Δ denotes relative percentage improvement.

	AMC 8			AMC 10			AMC 12		
Model	Direct	Post	Δ	Direct	Post	Δ	Direct	Post	Δ
Gemini-2.5-Flash	44.03	**47.39**	+7.63%	29.66	**32.36**	+9.10%	22.88	**24.35**	+6.42%
Claude-Haiku-4.5	35.07	**37.31**	+6.39%	27.19	**28.99**	+6.62%	19.56	**23.99**	+22.65%
GPT-5.4-Mini	31.72	**32.84**	+3.53%	21.80	**22.92**	+5.14%	15.50	**16.61**	+7.16%

Table 5: Proprietary models on mathematical benchmarks. Post-reasoning yields larger gains on complex competition problems (HMMT) compared to standard math tasks (GSM8K). Δ denotes relative percentage improvement.

	HMMT Feb			HMMT Nov			GSM8K		
Model	Direct	Post	Δ	Direct	Post	Δ	Direct	Post	Δ
Gemini-2.5-Flash	10.82	**11.30**	+4.44%	10.63	**11.44**	+7.62%	60.58	60.27	-0.51%
Claude-Haiku-4.5	13.22	**16.83**	+27.31%	15.80	**16.62**	+5.19%	48.90	**49.89**	+2.02%
GPT-5.4-Mini	6.49	6.49	0.00%	7.90	**8.17**	+3.42%	52.62	**52.99**	+0.70%

Table 6: Evaluation on proprietary models on GPQA, MMLU-Pro, and BIG-Bench Hard. Improvements are observed in most cases, with larger gains on reasoning-intensive tasks. Δ denotes relative percentage improvement.

	GPQA			MMLU-Pro			BIG-Bench Hard		
Model	Direct	Post	Δ	Direct	Post	Δ	Direct	Post	Δ
Gemini-2.5-Flash	49.21	**57.08**	+15.99%	72.07	**76.13**	+5.63%	73.46	**73.77**	+0.42%
Claude-Haiku-4.5	38.20	**43.60**	+14.14%	54.73	**59.90**	+9.45%	60.87	**70.60**	+15.98%
GPT-5.4-Mini	43.82	**44.72**	+2.05%	54.47	53.30	-2.15%	60.56	**62.88**	+3.83%

5 Supervised Post-Reason Tuning

Similar to inherently (pre-)reasoning models (Yang et al., 2025; OpenAI, 2025; DeepSeek-AI, 2025), where augmented reasoning behavior is internalized as part of the model, post-reasoning can also be learned as an intrinsic capability. To train the model, we use the supervised loss function described in Section 2.2 and explore three training strategies: (1) **Expert Distillation**, where the base model is trained on post-reasoning traces generated by a stronger expert model; (2) **Rephrased Distillation**, where the base model rewrites expert-generated reasoning traces in its own vocabulary; and (3) **Self-Distillation**, where the base model generates its own explanation conditioned on the input and final answer.

5.1 Post-Reasoning Train Samples

To prevent test set contamination, we construct a strictly filtered training corpus of approximately 3,500 integer-answer mathematical problems from the Numina dataset LI et al. (2024), removing any overlap with evaluation benchmarks such as GSM8K, AMC, and HMMT. To promote generalization rather than narrow overfitting, the dataset is curated to span multiple cognitive strata, ranging from standard algebraic reasoning (e.g., Orca Math Mitra et al. (2024)) to advanced multi-step logic (see Table 7). To ensure high-quality reasoning supervision, we prompt models to generate formal, proof-style justifications without prematurely revealing the final answer. Generated traces are filtered to remove instances that leak target assumptions or exhibit insufficient depth (e.g., fewer than 20 words), with invalid samples discarded and regenerated.

Table 7: Post-Reason train set, covering a range of reasoning depths from standard algebra to advanced multi-step logic.

Data Source	**Domain**	**Count**	**%**
Orca Math	Algebraic Logic	1,234	35.26
Synthetic Math	General Math	1,121	32.03
CN K-12	Intermediate	668	19.09
Olympiads	Multi-Step Logic	447	12.77
Synthetic AMC	Competition	30	0.86
Total		**3,500**	**100.00**

Table 8: Downstream accuracy ablation (% points) on Mistral-Small (24B). Self-distillation significantly outperforms expert and rephrased traces, particularly on complex reasoning tasks (GPQA).

Training Distribution	**MATH**	**GPQA**	**MMLU-Pro**	**BBH**
Expert Distillation	26.50	36.40	54.00	59.53
Rephrased Distillation	26.18	38.43	54.03	59.50
Self-Distillation	**27.21**	**45.17**	**54.53**	**59.94**

5.2 Training Setup

All models are fine-tuned using LoRA (Hu et al., 2021; Xu et al., 2023) with $r = 16$, $\alpha = 32$, and dropout 0.05. Training uses an effective batch size of 32 for 3 epochs, a learning rate of 2×10^{-5},

Table 9: Post-Reason training results on mathematical reasoning benchmarks. Each cell reports Post-Reason (PR) and Post-Reason SFT performance, with relative improvement shown in subscript.

Model	AMC 8	AMC 10	AMC 12	HMMT Feb	HMMT Nov	GSM8K
Llama Family						
Llama-3.1 (8B)	11.57 / **14.18**$_{(+22.56\%)}$	12.13 / **12.58**$_{(+3.71\%)}$	10.70 / **12.55**$_{(+17.29\%)}$	2.88 / **3.61**$_{(+25.35\%)}$	2.18 / **2.72**$_{(+24.77\%)}$	13.65 / 13.12$_{(-3.88\%)}$
Llama-3.3 (70B)	28.73 / 28.73$_{(0.00\%)}$	33.03 / **33.26**$_{(+0.70\%)}$	26.94 / **27.68**$_{(+2.75\%)}$	9.86 / **11.78**$_{(+19.47\%)}$	10.35 / 9.81$_{(-5.22\%)}$	35.78 / **36.01**$_{(+0.64\%)}$
Gemma Family						
Gemma-3 (12B)	17.16 / **24.63**$_{(+43.53\%)}$	13.26 / **13.71**$_{(+3.39\%)}$	6.64 / **9.23**$_{(+39.01\%)}$	4.33 / **4.81**$_{(+11.09\%)}$	5.99 / 5.72$_{(-4.51\%)}$	21.68 / 20.55$_{(-5.21\%)}$
Gemma-3 (27B)	15.30 / **18.28**$_{(+19.48\%)}$	17.08 / 16.63$_{(-2.63\%)}$	15.13 / 12.55$_{(-17.05\%)}$	6.25 / **6.97**$_{(+11.52\%)}$	7.90 / **8.17**$_{(+3.42\%)}$	31.77 / **31.92**$_{(+0.47\%)}$
Mistral & Ministral						
Ministral-3 (8B)	18.66 / 17.91$_{(-4.02\%)}$	11.69 / **12.81**$_{(+9.58\%)}$	7.75 / **11.44**$_{(+47.61\%)}$	4.57 / 3.37$_{(-26.26\%)}$	4.36 / **4.63**$_{(+6.19\%)}$	21.15 / **23.96**$_{(+13.29\%)}$
Ministral-3 (14B)	21.27 / **23.88**$_{(+12.27\%)}$	15.51 / **16.18**$_{(+4.32\%)}$	9.59 / **9.96**$_{(+3.86\%)}$	4.33 / **5.05**$_{(+16.63\%)}$	5.18 / **6.27**$_{(+21.04\%)}$	27.22 / **27.29**$_{(+0.26\%)}$
Mistral-Small (24B)	13.81 / **16.79**$_{(+21.58\%)}$	13.48 / **16.63**$_{(+23.37\%)}$	7.75 / **9.23**$_{(+19.10\%)}$	3.61 / **4.57**$_{(+26.59\%)}$	7.08 / **7.90**$_{(+11.58\%)}$	29.95 / **30.63**$_{(+2.27\%)}$
Qwen Family						
Qwen3.5 (4B)	20.90 / **22.39**$_{(+7.13\%)}$	10.79 / **12.36**$_{(+14.55\%)}$	5.17 / **6.27**$_{(+21.28\%)}$	3.37 / **5.29**$_{(+56.97\%)}$	3.54 / 3.54$_{(0.00\%)}$	19.48 / **20.24**$_{(+3.90\%)}$
Qwen3.5 (9B)	44.03 / 43.66$_{(-0.84\%)}$	16.63 / **16.85**$_{(+1.32\%)}$	9.59 / **11.44**$_{(+19.29\%)}$	4.33 / **4.57**$_{(+5.54\%)}$	4.90 / **6.81**$_{(+38.98\%)}$	24.56 / **26.08**$_{(+6.19\%)}$
Qwen3.5 (27B)	20.67 / **21.80**$_{(+5.47\%)}$	32.36 / **34.16**$_{(+5.56\%)}$	20.66 / **22.51**$_{(+8.95\%)}$	8.41 / **8.89**$_{(+5.71\%)}$	11.17 / **11.44**$_{(+2.42\%)}$	47.69 / 47.46$_{(-0.48\%)}$

and a cosine scheduler Loshchilov & Hutter (2017). The masked post-reasoning objective (Eq. 3) exhibits stable convergence (Figure 2).

As shown in Table 8, self-distillation consistently outperforms both expert and rephrased distillation, with the largest gains observed on GPQA. A likely reason is distributional alignment: in self-distillation, the model generates justifications in its own linguistic and reasoning style, making the resulting reasoning traces more compatible with its internal representations. In contrast, expert distillation introduces a distribution shift, as the model is trained to imitate reasoning traces produced by a different (often larger) model, which may not align with its own inductive biases. Rephrased distillation partially mitigates this mismatch, but still depends on external traces.

The advantage of self-distillation is particularly pronounced on complex reasoning tasks such as GPQA, where generating coherent, multi-step justifications is critical. Here, internally consistent reasoning appears more beneficial than imitating externally generated traces. On broader benchmarks (MMLU-Pro, BBH), the gains are smaller, suggesting that when tasks rely more on knowledge recall than deep reasoning, the choice of reasoning distribution has a reduced impact. Given its consistently strong performance and better alignment with the model's native reasoning distribution, we adopt self-distillation for all subsequent post-reason training experiments.

5.3 Results

Mathematical Reasoning. As shown in Table 9, supervised post-reason tuning further improves performance over prompt-based post-reasoning across a large majority of model–benchmark combinations. The gains are particularly pronounced on competition-style mathematical benchmarks such as AMC and HMMT, where multi-step and long-horizon reasoning are critical. In several cases, relative improvements exceed **20–40%**, especially for smaller and mid-sized models, indicating that post-reasoning behavior can be effectively internalized through supervised training.

Table 10: Post-Reason training results on cross-domain reasoning benchmarks.

Model	GPQA	MMLU-Pro	BBH
Llama			
L3.1 (8B)	32.81 / **34.16**$_{(+4.11)}$	37.17 / 36.80$_{(-1.00)}$	45.69 / 43.68$_{(-4.40)}$
L3.3 (70B)	49.21 / **50.11**$_{(+1.83)}$	53.37 / 53.13$_{(-0.45)}$	66.52 / 65.97$_{(-0.83)}$
Gemma			
G3 (12B)	34.61 / 34.16$_{(-1.30)}$	44.17 / **44.57**$_{(+0.91)}$	58.88 / **59.27**$_{(+0.66)}$
G3 (27B)	34.16 / **36.40**$_{(+6.56)}$	51.33 / 51.27$_{(-0.12)}$	63.83 / **65.47**$_{(+2.57)}$
Mistral			
M3 (8B)	37.98 / 35.51$_{(-6.50)}$	48.03 / 47.97$_{(-0.12)}$	55.21 / **55.64**$_{(+0.78)}$
M3 (14B)	39.78 / **40.22**$_{(+1.11)}$	54.57 / 53.60$_{(-1.78)}$	58.88 / **60.42**$_{(+2.62)}$
MS (24B)	40.22 / **45.17**$_{(+12.31)}$	54.43 / **54.53**$_{(+0.18)}$	60.80 / 59.94$_{(-1.41)}$
Qwen			
Q3.5 (4B)	42.92 / **43.37**$_{(+1.05)}$	45.57 / **45.67**$_{(+0.22)}$	54.42 / 53.99$_{(-0.79)}$
Q3.5 (9B)	58.65 / **60.90**$_{(+3.84)}$	51.17 / **51.50**$_{(+0.64)}$	57.52 / **58.47**$_{(+1.65)}$
Q3.5 (27B)	62.70 / **63.60**$_{(+1.44)}$	65.87 / 65.53$_{(-0.52)}$	68.67 / **68.78**$_{(+0.16)}$

The largest improvements are consistently observed on AMC and HMMT, suggesting that post-reason tuning strengthens structured reasoning capabilities beyond what is achieved through prompting alone. In contrast, gains on GSM8K are comparatively smaller and occasionally negative, likely because GSM8K relies on shorter reasoning chains where prompt-based post-reasoning is already sufficient, leaving limited room for additional improvement through training.

Cross-domain Generalization. Table 10 shows that the benefits of post-reason tuning extend beyond mathematical tasks. Despite being trained exclusively on mathematical reasoning data, the

tuned models exhibit consistent improvements over prompt-based post-reasoning across GPQA, MMLU-Pro, and BIG-Bench Hard, demonstrating effective cross-domain transfer.

Improvements are most notable on GPQA, where gains are consistently positive and occasionally substantial (e.g., up to **+12.31%**), reflecting its reliance on multi-step reasoning and uncertainty resolution. In contrast, gains on MMLU-Pro and BBH are smaller and more uniform, typically within **1–3%**, and occasionally negative. This aligns with the nature of these benchmarks, which rely more on knowledge retrieval and shallow reasoning compared to GPQA.

Across all benchmarks, the effectiveness of post-reason tuning scales with reasoning depth: larger gains are observed on tasks requiring long-horizon reasoning, while improvements are smaller but still present on knowledge-intensive tasks.

6 Post-Reasoning vs. Generic Instruction Augmentation

A natural question is whether the observed gains stem from generic instruction augmentation or if explicit reasoning is uniquely beneficial. As shown in Table 11, the cognitive objective of the post-generation task is the primary driver of performance gains.

Table 11: Instruction augmentations. Values denote average accuracy (%).

Benchmark	Post-Summary	Post-Confidence	Post-Reason
GPQA	32.59	34.53	**35.88**
MMLU-Pro	43.54	44.40	**45.26**
GSM8K	20.24	21.05	**21.76**

Specifically, the **Post-Summary** variant consistently underperforms because summarization merely restates the problem and the selected output. The **Post-Confidence** variant offers a stronger baseline by encouraging metacognitive reflection, yet it remains inferior to **Post-Reasoning**. By explicitly requiring the model to justify its output, **Post-Reasoning** imposes a causal constraint between the answer and subsequent tokens. This constraint encourages the model to internalize a coherent latent reasoning trajectory before answering. These results indicate that Post-Reasoning improves the model's latent reasoning process, rather than generating performance gains as a simple byproduct of an increased token count.

7 Limitations

While supervised post-reason tuning improves direct-answer capabilities, it has several limitations. First, performance on tasks requiring deep algorithmic search, such as HMMT, remains bounded, as complex logic trees may still benefit from explicit Chain-of-Thought (CoT) reasoning Wei et al. (2022); Snell et al. (2024). Second, gains on widely represented benchmarks such as GSM8K are comparatively small due to already strong direct-inference capabilities from extensive pretraining exposure. In some cases, additional reasoning can even introduce minor regressions, consistent with recent findings on reasoning redundancy in simple mathematical tasks Chen et al. (2024); Srivastava et al. (2026); Aggarwal et al. (2025). Finally, although self-distillation avoids human-annotated reasoning traces by relying only on question-answer pairs, its effectiveness depends on task complexity, with shallow factual tasks providing weaker supervision signals than multi-step reasoning problems.

8 Conclusion

We introduced Post-Reasoning, a paradigm that conditions models to justify answers after generation, eliminating the latency and token costs of intermediate reasoning. Evaluations across 13 models demonstrate this consistently enhances direct-answer accuracy on complex tasks. Furthermore, supervised post-reason tuning successfully embeds these capabilities into model weights, outperforming zero-shot baselines in 91.11% of cases. Ultimately, Post-Reasoning establishes a highly efficient performance ceiling for instruction-tuned models, motivating future research into latency-free inference.

AI USE STATEMENT

Generative AI tools were used to assist with grammar correction and stylistic refinement. AI tools were not used to generate datasets, implement the evaluated methods, execute experiments, or produce any reported measurements. All AI-assisted material was reviewed and verified by the authors, who take full responsibility for the final content, claims, and artifacts in this work.

REFERENCES

Pranjal Aggarwal, Seungone Kim, Jack Lanchantin, Sean Welleck, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. Optimalthinkingbench: Evaluating over and underthinking in llms, 2025. URL https://arxiv.org/abs/2508.13141.

Malika Aubakirova, Alex Atallah, Chris Clark, Justin Summerville, and Anjney Midha. State of AI: An empirical 100 trillion token study with OpenRouter. a16z (Andreessen Horowitz) and OpenRouter Inc., December 2025. URL https://openrouter.ai/state-of-ai. Accessed: 2026-05-07.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz 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.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Deloitte. Deloitte's enterprise ai infrastructure survey: A 2028 outlook. https://www.deloitte.com/us/en/insights/topics/technology-management/ai-infrastructure-survey.html, 2026. Accessed: 2026-05-07.

Mucong Ding, Chenghao Deng, Jocelyn Choo, Zichu Wu, Aakriti Agrawal, Avi Schwarzschild, Tianyi Zhou, Tom Goldstein, John Langford, Anima Anandkumar, et al. Easy2hard-bench: Standardized difficulty labels for profiling llm performance and generalization. *arXiv preprint arXiv:2409.18433*, 2024.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5060. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/de2ad3ed44ee4e675b3be42aa0b615d0-Abstract-Conference.html.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie

Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie

Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Huizhong Guo, Tianjun Wei, Dongxia Wang, Yingpeng Du, Ziyan Wang, Jie Zhang, and Zhu Sun. Think when needed: Model-aware reasoning routing for llm-based ranking, 2026. URL https://arxiv.org/abs/2601.18146.

Tingxu Han, Chunrong Fang, Shiyu Zhao, Shiqing Ma, Zhenyu Chen, and Zhenting Wang. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*, 2024.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Lingjie Jiang, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xingxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. Think only when you need with large hybrid-reasoning models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2505.14631.

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.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. Datasets: A community library for natural language processing, 2021. URL https://arxiv.org/abs/2109.02846.

Gengyang Li, Yifeng Gao, Yuming Li, and Yunfang Wu. Thinkless: A training-free inference-efficient method for reducing reasoning redundancy. *arXiv preprint arXiv:2505.15684*, 2025.

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. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

Guosheng Liang, Longguang Zhong, Ziyi Yang, and Xiaojun Quan. Thinkswitcher: When to think hard, when to think fast, 2025. URL https://arxiv.org/abs/2505.14183.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Clémence Lanfranchi, Corentin Barreau, Cyprien Courtot, Daniele Grattarola, Darius Dabert, Diego de las Casas, Elliot Chane-Sane, Faruk Ahmed, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet, Georgii Novikov, Guillaume Kunsch, Guillaume Lample, Guillaume Martin, Gunshi Gupta, Jan Ludziejewski, Jason Rute, Joachim Studnia, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Karmesh Yadav, Khyathi Chandu, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Maarten Buyl, Margaret Jennings, Marie Pellat, Mark Prins, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mia Chiquier, Michel Schimpf, Nathan Grinsztajn, Neha Gupta, Nikhil Raghuraman, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Patrick von Platen, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Quentin Torroba, Romain Sauvestre, Roman Soletskyi, Rupert Menneer, Sagar Vaze, Samuel Barry, Sanchit Gandhi, Siddhant Waghjale, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thiziri Nait Saada, Thomas Chabal, Thomas Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Bewley, Tom Edwards, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vincent Maladière, Virgile Richard, Wassim Bouaziz, Wen-Ding Li, William Marshall, Xinghui Li, Xinyu Yang, Yassine El Ouahidi, Yihan Wang, Yunhao Tang, and Zaccharie Ramzi. Ministral 3, 2026. URL https://arxiv.org/abs/2601.08584.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017. URL https://arxiv.org/abs/1608.03983.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. Adacot: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning, 2025. URL https://arxiv.org/abs/2505.11896.

Haotian Luo, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao, Dacheng Tao, and Li Shen. Adar1: From long-cot to hybrid-cot via bi-level adaptive reasoning optimization, 2025a. URL https://arxiv.org/abs/2504.21659.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025b.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*, 2025.

Microsoft AI Economy Institute. Global ai adoption 2025, 2025. URL https://www.microsoft.com/en-us/corporate-responsibility/topics/ai-economy-institute/reports/global-ai-adoption-2025/.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-math: Unlocking the potential of slms in grade school math, 2024. URL https://arxiv.org/abs/2402.14830.

OpenAI. The state of enterprise ai: 2025 report. https://cdn.openai.com/pdf/7ef17d82-96bf-4dd1-9df2-228f7f377a29/the-state-of-enterprise-ai_2025-report.pdf, 2025. Accessed: 2026-05-07.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

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, 2023. URL https://arxiv.org/abs/2311.12022.

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.

Gaurav Srivastava, Aafiya Hussain, Sriram Srinivasan, and Xuan Wang. Do llms overthink basic math reasoning? benchmarking the accuracy-efficiency tradeoff in language models, 2026. URL https://arxiv.org/abs/2507.04023.

Jiayuan Su, Fulin Lin, Zhaopeng Feng, Han Zheng, Teng Wang, Zhenyu Xiao, Xinlong Zhao, Zuozhu Liu, Lu Cheng, and Hongwei Wang. Cp-router: An uncertainty-aware router between llm and lrm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33065–33073, 2026. doi: 10.1609/aaai.v40i39.40589. URL https://ojs.aaai.org/index.php/AAAI/article/view/40589.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Yu-Chien Tang, Jun-Chen Hung, Wen-Chih Peng, and An-Zi Yen. Vdar-router: Adaptive llms routing via verbalized query difficulty analysis retrieval, 2026. URL https://arxiv.org/abs/2607.18098.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin,

Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020. URL https://arxiv.org/abs/1910.03771.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment, 2023. URL https://arxiv.org/abs/2312.12148.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

Zishun Yu, Tengyu Xu, Di Jin, Karthik Abinav Sankararaman, Yun He, Wenxuan Zhou, Zhouhao Zeng, Eryk Helenowski, Chen Zhu, Sinong Wang, Hao Ma, and Han Fang. Think smarter not harder: Adaptive reasoning with inference aware optimization, 2025. URL https://arxiv.org/abs/2501.17974.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3716–3730. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.emnlp-main.184. URL https://aclanthology.org/2025.emnlp-main.184/.

Xiaoyun Zhang, Jingqing Ruan, Xing Ma, Yawen Zhu, Haodong Zhao, Hao Li, Jiansong Chen, Ke Zeng, and Xunliang Cai. When to continue thinking: Adaptive thinking mode switching for efficient reasoning, 2025b. URL https://arxiv.org/abs/2505.15400.

APPENDIX CONTENTS

Appendix A: When to Use Thinking & How to Decide? 18
Appendix B: Detailed Experimental Setup 19
Appendix C: Supervised Fine-Tuning (SFT) Setup 23
Appendix D: Native Latent Thinking Prompting Evaluations 25
Appendix E: Answer Latency and Parsing Reliability 28
Appendix F: Post-Task Ablation 29
Appendix G: Extended Optimization Dynamics 30
Appendix H: Training-Corpus Dependency 32

A When to Use Thinking & How to Decide?

A.1 How to switch between thinking and non-thinking?

In practice, one of the most promising approaches is to employ a **router model** that predicts whether a query requires reasoning and routes it to a non-thinking or thinking model accordingly (e.g., as shown in Table 3 of Aggarwal et al., 2025). A router can be obtained through training-free methods (Su et al., 2026; Tang et al., 2026) or trained to predict whether a query will benefit from additional reasoning. For instance, the approach in Guo et al. (2026) trains such a router by comparing the performance and token cost of Think and Non-Think modes and learning the expected utility gain of reasoning. During inference, reasoning is invoked only when the predicted gain exceeds a predefined threshold or budget.

A.2 Which tasks?

OptimalThinkingBench (Aggarwal et al., 2025) provides a broad characterization of which tasks benefit from substantial reasoning versus minimal reasoning. Additional reasoning is generally beneficial for tasks involving algorithms, graph reasoning, arithmetic, geometry, formal logic, and strategic games. In contrast, simple factual retrieval, general knowledge questions, and other straightforward queries often derive little or no benefit from thinking and may even experience degraded performance due to unnecessary reasoning. OptimalThinkingBench comprises 72 domains in which thinking provides little or no performance benefit. Determining the appropriate amount of thinking for a given task has therefore emerged as an increasingly important research direction (Fang et al., 2025; Jiang et al., 2025; Lou et al., 2025; Zhang et al., 2025b; Yu et al., 2025; Liang et al., 2025; Luo et al., 2025a; Zhang et al., 2025a; Chen et al., 2024).

Overall, although thinking models are essential for reasoning-intensive tasks, **non-thinking models remain equally important**, making model routing a practical solution for dynamically selecting between thinking and non-thinking models.

B Detailed Experimental Setup

To ensure full transparency and reproducibility, this section provides comprehensive details regarding the hardware, software, decoding configurations, models, and prompt templates utilized in our prompting and supervised post-reason tuning experiments, including both localized open-source deployments and proprietary API integrations.

B.1 Hardware Infrastructure

All baseline inference evaluations for open-source weights were conducted on a localized high-performance compute cluster equipped with NVIDIA H200 Tensor Core GPUs, each featuring 141GB of VRAM. Due to the high memory capacity of the H200 accelerators, inference for models up to 32B parameters was efficiently executed on single-GPU nodes. The only exception was `Llama-3.3-70B-Instruct`, which required tensor parallelism distributed across 2x H200 GPUs to accommodate the massive model weights and extended KV-cache. Conversely, evaluations involving proprietary closed-source models were conducted via their respective official APIs, bypassing these local hardware dependencies.

B.2 Software Stack and Libraries

The experimental pipeline was built upon a robust Python software stack, structurally organized to follow the chronological progression of our methodology:

- **Prompting (Inference & Evaluation):** All localized open-source model generation was accelerated using the `vllm` engine (Kwon et al., 2023). Proprietary model evaluations were orchestrated asynchronously utilizing the `openai` Python SDK and `asyncio`, with concurrency tracked via `tqdm`. Downstream evaluation, regular expression answer parsing, and result aggregation were executed using `pandas`, `json`, and `re`.
- **Supervised Post-Reason Tuning (Distillation Data Generation):** The synthetic corpora required for Target-Conditioned Self-Distillation were generated, filtered, and concatenated utilizing the `openai` Python SDK, `asyncio`, and the Hugging Face `datasets` library (Lhoest et al., 2021).
- **Supervised Post-Reason Tuning (Model Training):** Model weights, tokenization, and training loops were implemented using PyTorch (Paszke et al., 2019) and the Hugging Face `transformers` library (Wolf et al., 2020). Parameter-efficient fine-tuning via Low-Rank Adaptation (LoRA) was configured using `peft`.

B.3 Inference Engine and API Configuration

To maintain consistent high-throughput generation across the diverse open-source model suite, we deployed the vLLM engine Kwon et al. (2023). A dedicated `tmux` orchestration pipeline was used to spin up independent vLLM servers mapped to specific network ports for asynchronous client requests.

To optimize the KV-cache and prevent Out-of-Memory (OOM) errors during varying contextual loads, the `-max-model-len` argument was dynamically scaled:

- **Standard Instruct Evaluation:** Capped at $16,384$ tokens.
- **Native Thinking Evaluation:** Expanded to $32,768$ tokens to accommodate the extensive, unbounded latent reasoning traces generated by models like GPT-OSS, Qwen3 and Qwen3.5.

For Qwen-family models leveraging latent planning, the `-reasoning-parser qwen3` flag was explicitly passed. All local models utilized `-enable-prefix-caching` to accelerate the 3-shot prompt processing, and GPU memory utilization was strictly bounded at 0.90.

For the closed-source proprietary models (Gemini-2.5-Flash, Claude-Haiku-4.5, and GPT-5.4-Mini), request orchestration was managed via customized asynchronous Python pipelines interfacing with their official REST APIs. To ensure strict parity with the standard instruct baselines and isolate

the effect of structural prompting, all native latent reasoning features (e.g., Anthropic's `thinking` blocks, OpenAI's `reasoning_effort`, and Gemini's `thinking_budget`) were explicitly disabled during generation.

B.4 Model Deployment Registry

Our evaluation pipeline utilized a diverse suite of models distributed across the experimental stages. In the **Prompting Evaluation (Inference-Time Prompting Efficacy)**, all 14 distinct open-source models were evaluated to establish rigorous zero-shot baselines. This included the standard instruct models as well as the GPT-OSS 20B, Qwen3, and Qwen3.5 families, which were additionally evaluated on their thinking capabilities. Furthermore, a **Prompting Extension** incorporated proprietary models to provide state-of-the-art closed-source baselines.

In the **Supervised Post-Reason Tuning Evaluation**, the focus shifted to evaluating the architecture weights updated via the SFT pipeline. This stage was conducted specifically on the standard instruct architectures (Llama, Gemma, Ministral, Mistral) and the Qwen3.5 family. Notably, the Qwen3 and GPT-OSS models were utilized exclusively during prompt-based evaluations for native thinking comparisons and were not subjected to the supervised post-reason tuning pipeline. Table 12 details the complete registry alongside their specific deployment protocols.

Table 12: Complete Model Deployment Registry (Prompting & SFT)

Model Family	Model Identifier	Experiment	GPU Config	Port
Gemma 3	`gemma-3-12b-it`	Prompting, SFT	1x H200	8010
	`gemma-3-27b-it`	Prompting, SFT	1x H200	8011
Llama 3	`Llama-3.1-8B-Instruct`	Prompting, SFT	1x H200	8018
	`Llama-3.3-70B-Instruct`	Prompting, SFT	2x H200	8009
Ministral	`Ministral-3-8B-Instruct-2512`	Prompting, SFT	1x H200	8015
	`Ministral-3-14B-Instruct-2512`	Prompting, SFT	1x H200	8016
Mistral	`Mistral-Small-24B-Instruct-2501`	Prompting, SFT	1x H200	8017
Qwen 3	`Qwen3-8B`	Prompting	1x H200	8012
	`Qwen3-14B`	Prompting	1x H200	8013
	`Qwen3-32B`	Prompting	1x H200	8014
Qwen 3.5	`Qwen35-4B`	Prompting, SFT	1x H200	8019
	`Qwen35-9B`	Prompting, SFT	1x H200	8020
	`Qwen35-27B`	Prompting, SFT	1x H200	8021
GPT-OSS	`GPT-OSS-20B`	Prompting	1x H200	8008
Proprietary API Models (Prompting Extension)				
Gemini	`gemini-2.5-flash`	Prompting	API	-
Claude	`claude-haiku-4-5`	Prompting	API	-
GPT	`gpt-5.4-mini`	Prompting	API	-

B.5 Generation Hyperparameters

Hyperparameters were standardized based on the architectural capabilities of the models and the requirements of the active prompting strategy. For standard instruct baselines in both prompting and supervised post-reason tuning evaluations, generation was constrained to $2,048$ tokens for direct extraction and $4,096$ tokens for post-reason extraction on local models. All evaluations used temperature 0.0. For the proprietary API models, maximum output tokens were uniformly expanded to $8,192$ to ensure responses were not prematurely truncated during extensive post-reasoning generations.

For the auxiliary thinking evaluations during prompt-based testing, the evaluated open-source models (GPT-OSS, Qwen3, Qwen3.5) utilized native reasoning tags. Consequently, their token limits were expanded significantly to $16,384$ tokens. Table 13 provides the exact decoding parameters used across the different prompting strategies.

Table 13: Exact Generation Hyperparameters by Architecture and Strategy

Model Class	Strategy	Max Tokens	Temp.	Top-p	Top-k	Pres. Pen.	Rep. Pen.
Llama, Ministral, Mistral, Gemma (Prompting & SFT)	Direct Answer	2,048	0.0	0.8	20	-	-
	Post-Reason	4,096	0.0	0.8	20	-	-
GPT-OSS, Qwen 3 (Prompting Only)	Direct Answer	2,048	0.0	0.8	20	-	-
	Post-Reason	4,096	0.0	0.8	20	-	-
	Thinking Direct	16,384	0.0	0.95	20	-	-
	Thinking Post	16,384	0.0	0.95	20	-	-
Qwen 3.5 (Prompting & SFT)	Direct Answer	2,048	0.0	0.8	20	1.5	1.0
	Post-Reason	4,096	0.0	0.8	20	1.5	1.0
	Thinking Direct	16,384	0.0	0.95	20	1.5	1.0
	Thinking Post	16,384	0.0	0.95	20	1.5	1.0
Proprietary APIs (Prompting Extension)	Direct	2,048	0.0	-	-	-	-
	Post-Reason	4,096	0.0	-	-	-	-
	Thinking Direct	16,384	0.0	-	-	-	-
	Thinking Post	16,384	0.0	-	-	-	-

B.6 Prompting Frameworks

All evaluations used **3-shot prompting**. For each benchmark, the `<task-specific instruction>` comprises the benchmark system instruction, the 3-shot examples, and the question. We then append the Direct or Post-Reason suffix to this complete task-specific instruction. The Thinking Direct suffix is identical to the Direct suffix, and Thinking Post is identical to the Post-Reason suffix. The two prompts therefore have the following form:

Direct/ Thinking Direct Prompt

```
<task-specific instruction> State the final answer
immediately.
```

Post-Reason/ Thinking Post Prompt

```
<task-specific instruction> State the final answer immediately
and justify your answer.
```

As outlined in our methodology, these strategies were deployed in prompt-based evaluation to establish distinct comparative baselines:

- **Standard Prompting (All Models):** Evaluated using the **Direct Answer** and **Post-Reason** system instructions, completely disabling internal thinking to test standard generative formatting.
- **Native Thinking (Select Models):** Restricted to the GPT-OSS 20B, Qwen3, and Qwen3.5 model families for open-source models. For closed-source models, it is restricted to Gemini-2.5-Flash, Claude-Haiku-4.5, and GPT-5.4-Mini, with thinking level set to low. These were evaluated using the **Thinking Direct** and **Thinking Post** instructions, which contain identical textual constraints to the standard prompts but with thinking to invoke the internal `<think>` reasoning block prior to generating visible text.

Tables 14 detail the exact system prompts utilized for each respective benchmark in our suite.

Table 14: Generic benchmark system instructions shared by Direct, Post-Reason, Thinking Direct, and Thinking Post.

Benchmark	**System instruction**
AMC 8	Solve the competition mathematics problem. Return the final answer as an integer.
AMC 10	Solve the competition mathematics problem. Return the final answer as an integer.
AMC 12	Solve the competition mathematics problem. Return the final answer as an integer.
HMMT February	Solve the competition mathematics problem. Return the final answer as an integer.
HMMT November	Solve the competition mathematics problem. Return the final answer as an integer.
GSM8K	Solve the grade-school mathematics problem. Return the final answer as a number.
GPQA	Answer the graduate-level science multiple-choice question. Return the letter of the correct option.
MMLU-Pro	Answer the multiple-choice question. Return one option letter from A through J.
BBH	Solve the task and return the answer in the format required by the question.

C SUPERVISED FINE-TUNING (SFT) SETUP

To explicitly embed the post-reasoning cognitive pathway into the model weights, we utilized a highly constrained Supervised Fine-Tuning (SFT) pipeline. Rather than training the model on the full conversational sequence, we implemented a **Targeted Loss Masking** algorithm to prevent the model from overfitting to the instruction prompts or memorizing the answers.

During the tokenization phase, the system prompt, the user query, and the immediate statement of the final numerical answer were explicitly masked by assigning them a label index of `-100`. Consequently, the autoregressive cross-entropy loss was computed *exclusively* over the generative tokens corresponding to the detailed logical trajectory (the `Explanation:` block). This forced the parameter updates to optimize solely for the structural and logical integrity of the latent reasoning, entirely decoupling the reasoning process from the prompt structure.

C.1 COMPUTATIONAL COST AND RESOURCE ALLOCATION

To provide full transparency regarding the computational footprint and environmental impact of our study, we highly optimized the Supervised Fine-Tuning (SFT) pipeline to minimize both hardware requirements and human oversight.

Due to the memory efficiency of LoRA combined with DeepSpeed ZeRO-3 and Flash Attention 2, the complete fine-tuning process was successfully executed using only **1x NVIDIA H200 (141GB) GPU per model**. The sole exception to this single-node constraint was the `Llama-3.3-70B` architecture, which required distribution across **2x NVIDIA H200 GPUs** to safely accommodate the expanded optimizer states and gradient checkpoints without triggering Out-of-Memory (OOM) failures. Furthermore, the streamlined data processing and training orchestration required approximately **2 hours of dedicated setup and monitoring per model**, demonstrating the practical viability of our targeted masking approach for rapid model alignment.

C.2 HYPERPARAMETER CONFIGURATION

All fine-tuning was executed using Low-Rank Adaptation (LoRA) Hu et al. (2021) to efficiently update the attention and multi-layer perceptron (MLP) blocks. To maintain high memory efficiency across our H200 cluster, we utilized gradient checkpointing, Flash Attention 2 Dao (2023), and bfloat16 precision. The exact training and LoRA parameters utilized across all models are detailed in Table 15.

Table 15: Exact SFT and LoRA Hyperparameters

Hyperparameter	**Value**
LoRA Configuration	
Rank (r)	16
Alpha (α)	32
Dropout	0.05
Target Modules	`q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj`
Training Arguments	
Learning Rate	2×10^{-5}
LR Scheduler	Cosine
Warmup Ratio	0.1
Epochs	3
Per-Device Batch Size	2
Gradient Accumulation Steps	16
Max Sequence Length	4,096
Optimizer	AdamW (`adamw_torch`)
Precision	bfloat16
Gradient Checkpointing	Enabled

C.3 Answer-Token Masking Ablation

We compared the masked Post-Reason objective against an unmasked objective that applies loss to both the answer and justification. The data, LoRA configuration, and seeds were identical; the loss mask over the answer was the only difference. Table 16 reports the existing three-seed results. Masking the answer produced higher observed accuracy in all nine evaluated settings, showing that direct supervision of the answer tokens was not required for these gains.

Table 16: Post-Reason SFT accuracy under masked and unmasked objectives. Values are mean accuracy over three seeds. Δ is Masked minus Unmasked in accuracy points.

	AMC 10			GPQA			MMLU-Pro		
Model	Unmasked	Masked	Δ	Unmasked	Masked	Δ	Unmasked	Masked	Δ
Qwen-3.5 Family									
Qwen-3.5 (4B)	9.75	10.86	$+1.11^*$	40.90	42.87	$+1.97^*$	44.90	45.70	$+0.80^*$
Ministral-3 Family									
Ministral-3 (14B)	14.51	16.01	$+1.50^*$	35.51	39.68	$+4.17^*$	51.43	54.91	$+3.48^*$
Llama-3.3 Family									
Llama-3.3 (70B)	32.01	33.15	$+1.14^*$	48.09	49.36	$+1.27^*$	51.67	53.31	$+1.64^*$

$^*p < 0.05$, one-sided Welch t-test, $n = 3$ seeds per arm.

D NATIVE LATENT THINKING PROMPTING EVALUATIONS

While the core experiments established the regularizing effects of post-reasoning constraints on standard generative models, modern architectures are increasingly equipped with native latent reasoning mechanisms (e.g., <think> tokens). To provide a comprehensive analysis of inference-time prompting, we conducted an auxiliary evaluation to determine how structural formatting interacts with these built-in cognitive blocks.

D.1 EXPERIMENTAL DESIGN AND SCOPE

This evaluation was strictly bounded to model families in our registry natively trained to support latent reasoning traces: the GPT-OSS (20B), Qwen3 (8B, 14B, 32B), and Qwen3.5 (4B, 9B, 27B) architectures. For closed-source models, the evaluation additionally included Gemini-2.5-Flash, Claude-Haiku-4.5, and GPT-5.4-Mini with thinking level set to low.

We established two comparative baselines analogous to our standard instruct evaluations, but modified to explicitly invoke the reasoning blocks prior to generating visible text:

- **Thinking Direct:** The model utilizes its <think> block to explore the problem space, followed immediately by generating the final extracted answer.
- **Thinking Post:** The model utilizes its <think> block, outputs the final extracted answer, and is then forced to generate a visible, step-by-step post-reasoning justification.

Our objective was to observe whether the structural constraint of generating a post-reasoning explanation provides any supplementary regularization when the model has already been afforded a dedicated, unconstrained latent thinking phase.

D.2 THINKING BASELINE RESULTS AND ANALYSIS

The addition of native thinking mechanisms allocate more compute and explore logical trajectories prior to outputting the initial answer token, the relative impact of formatting the subsequent text is shifted. Figure 2 illustrates the overall training loss curve for the Post-Reason SFT framework.

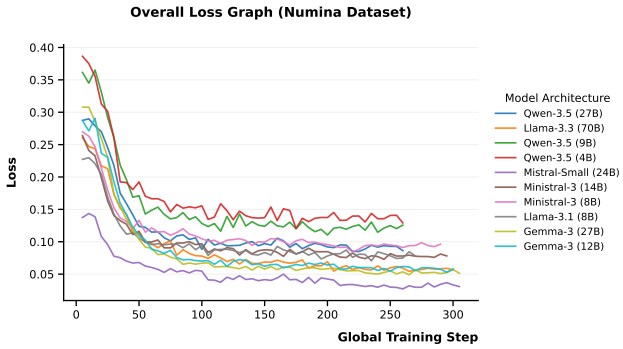


Figure 2: Training loss curve for the Post-Reason SFT framework, demonstrating stable gradient descent over the masked rationale.

Tables 17 through 22 detail the performance of the **Thinking Direct** (denoted in tables as Think Dir) and **Thinking Post** (denoted in tables as Think Post) strategies across our benchmark suite.

These results reveal a nuanced interplay between latent reasoning and output formatting. For smaller architectures (e.g., Qwen-3 8B), the addition of a visible post-reasoning requirement yields massive relative improvements, such as a $+290.43\%$ gain on AMC 8, which suggests that their internal <think> blocks are not yet sufficiently robust to independently secure the correct logical trajectory without the anchor of subsequent generative constraints. Conversely, highly capable models (e.g., Qwen-3.5 27B) exhibit diminishing returns or slight regressions under formatting, indicating that their latent planning is already optimal and forcing additional output structure can occasionally disrupt their inherent reasoning pathways.

Table 17: Open-source models: AMC Competition Mathematics. Δ represents the relative percentage improvement.

	AMC 8			AMC 10			AMC 12		
Model	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ
GPT-OSS Family									
GPT-OSS (20B)	86.19	90.30	+4.77%	84.27	87.42	+3.74%	79.34	83.03	+4.65%
Qwen-3 Family									
Qwen-3 (8B)	23.51	91.79	+290.43%	75.96	88.54	+16.56%	60.52	76.01	+25.59%
Qwen-3 (14B)	92.54	94.78	+2.42%	91.01	90.56	-0.49%	83.03	79.70	-4.01%
Qwen-3 (32B)	85.82	86.19	+0.43%	87.19	81.57	-6.45%	72.69	71.22	-2.02%
Qwen-3.5 Family									
Qwen-3.5 (4B)	80.97	94.03	+16.13%	74.83	89.89	+20.13%	50.92	86.35	+69.58%
Qwen-3.5 (9B)	74.63	93.28	+24.99%	79.10	90.11	+13.92%	46.49	77.49	+66.68%
Qwen-3.5 (27B)	92.16	90.67	-1.62%	93.26	88.54	-5.06%	82.29	78.23	-4.93%

Table 18: Open-source models: Standard and Competition Mathematics. Δ represents the relative percentage improvement.

	GSM8K			HMMT Feb			HMMT Nov		
Model	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ
GPT-OSS Family									
GPT-OSS (20B)	95.53	95.83	+0.31%	50.72	54.33	+7.12%	62.67	64.31	+2.62%
Qwen-3 Family									
Qwen-3 (8B)	95.00	95.83	+0.87%	28.85	43.99	+52.48%	27.52	59.13	+114.86%
Qwen-3 (14B)	95.75	96.36	+0.64%	50.72	50.24	-0.95%	64.31	64.31	0.00%
Qwen-3 (32B)	94.47	97.04	+2.72%	18.51	46.15	+149.32%	32.97	59.13	+79.34%
Qwen-3.5 Family									
Qwen-3.5 (4B)	95.83	96.36	+0.55%	11.06	48.08	+334.72%	31.06	59.67	+92.11%
Qwen-3.5 (9B)	97.04	97.12	+0.08%	12.50	38.70	+209.60%	29.97	57.22	+90.92%
Qwen-3.5 (27B)	97.42	97.42	+0.00%	35.34	25.72	-27.22%	54.22	41.96	-22.61%

Table 19: Open-source models: General Reasoning Benchmarks. Δ represents the relative percentage improvement.

	GPQA			MMLU-Pro			BIG-Bench Hard		
Model	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ
GPT-OSS Family									
GPT-OSS (20B)	60.00	60.90	+1.50%	72.90	74.47	+2.15%	88.60	92.20	+4.06%
Qwen-3 Family									
Qwen-3 (8B)	51.91	53.71	+3.47%	75.33	75.33	+0.00%	89.42	89.91	+0.55%
Qwen-3 (14B)	52.58	55.96	+6.43%	77.97	78.20	+0.29%	89.91	91.49	+1.76%
Qwen-3 (32B)	57.98	59.55	+2.71%	80.17	80.83	+0.82%	91.61	92.27	+0.72%
Qwen-3.5 Family									
Qwen-3.5 (4B)	66.07	68.54	+3.74%	78.50	79.23	+0.93%	90.68	92.84	+2.38%
Qwen-3.5 (9B)	72.81	73.93	+1.54%	82.10	82.27	+0.21%	91.78	93.81	+2.21%
Qwen-3.5 (27B)	77.98	81.12	+4.03%	87.07	86.50	-0.65%	95.61	95.53	-0.08%

D.3 CLOSED-SOURCE MODELS

We evaluated Gemini-2.5-Flash, Claude-Haiku-4.5, and GPT-5.4-Mini under four conditions. Direct and Post-Reason used native thinking disabled. Thinking Direct used exactly the Direct prompt with thinking level set to low, while Thinking Post used exactly the Post-Reason prompt with thinking level set to low. All four conditions used the same system instruction, three-shot prefix, question, model setting, and temperature-zero decoding.

Tables 20–22 report accuracy and relative percentage improvements. Post-Reason improved all 27 low-effort model–benchmark settings, with a mean improvement of 7.90% and a median improvement of 4.95%. The observed improvements range from 1.66% to 55.44%.

Table 20: Closed-source models: AMC Competition Mathematics. Δ represents the relative percentage improvement.

	AMC 8			**AMC 10**			**AMC 12**		
Model	Direct	Post-Reason	Δ	Direct	Post-Reason	Δ	Direct	Post-Reason	Δ
Claude-Haiku-4.5	35.07	37.31	+6.39%	27.19	28.99	+6.62%	19.56	23.99	+22.65%
GPT-5.4-Mini	31.72	32.84	+3.53%	21.80	22.92	+5.14%	15.50	16.61	+7.16%
Gemini-2.5-Flash	44.03	47.39	+7.63%	29.66	32.36	+9.10%	22.88	24.35	+6.42%
	AMC 8			**AMC 10**			**AMC 12**		
Model	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ
Claude-Haiku-4.5	82.84	89.18	+7.65%	72.36	79.33	+9.63%	57.93	67.16	+15.93%
GPT-5.4-Mini	87.03	89.55	+2.90%	80.40	84.49	+5.09%	79.34	81.55	+2.79%
Gemini-2.5-Flash	76.87	78.36	+1.94%	59.27	61.57	+3.88%	43.76	46.13	+5.42%

Table 21: Closed-source models: Standard and Competition Mathematics. Δ represents the relative percentage improvement.

	GSM8K			**HMMT Feb**			**HMMT Nov**		
Model	Direct	Post-Reason	Δ	Direct	Post-Reason	Δ	Direct	Post-Reason	Δ
Claude-Haiku-4.5	48.90	49.89	+2.02%	13.22	16.83	+27.31%	15.80	16.62	+5.19%
GPT-5.4-Mini	52.62	52.99	+0.70%	6.49	6.49	0.00%	7.90	8.17	+3.42%
Gemini-2.5-Flash	60.58	60.27	-0.51%	10.82	11.30	+4.44%	10.63	11.44	+7.62%
	GSM8K			**HMMT Feb**			**HMMT Nov**		
Model	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ
Claude-Haiku-4.5	95.04	97.74	+2.84%	28.61	44.47	+55.44%	50.95	53.95	+5.89%
GPT-5.4-Mini	91.30	95.30	+4.38%	50.96	54.33	+6.61%	61.85	65.40	+5.74%
Gemini-2.5-Flash	90.91	93.63	+2.99%	13.94	15.62	+12.05%	18.80	20.16	+7.23%

Table 22: Closed-source models: General Reasoning Benchmarks. Δ represents the relative percentage improvement.

	GPQA			**MMLU-Pro**			**BBH**		
Model	Direct	Post-Reason	Δ	Direct	Post-Reason	Δ	Direct	Post-Reason	Δ
Claude-Haiku-4.5	38.20	43.60	+14.14%	54.73	59.90	+9.45%	60.87	70.60	+15.98%
GPT-5.4-Mini	43.82	44.72	+2.05%	54.47	53.30	-2.15%	60.56	62.88	+3.83%
Gemini-2.5-Flash	49.21	57.08	+15.99%	72.07	76.13	+5.63%	73.46	73.77	+0.42%
	GPQA			**MMLU-Pro**			**BBH**		
Model	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ	Think Dir	Think Post	Δ
Claude-Haiku-4.5	57.53	58.88	+2.35%	78.07	81.90	+4.91%	85.38	93.80	+9.86%
GPT-5.4-Mini	65.09	68.31	+4.95%	78.13	81.87	+4.79%	87.69	91.06	+3.84%
Gemini-2.5-Flash	57.30	68.31	+19.21%	82.33	83.70	+1.66%	90.37	93.26	+3.20%

E Answer Latency and Parsing Reliability

We measured wall-clock time from request submission to the final answer token under Direct and deployment-time Post-Reason. For Post-Reason, generation stops when the answer field is complete, before generating the requested justification. Measurements used single requests on an idle H200 node; each entry in Table 23 is the range of per-benchmark medians across nine benchmarks. Post-Reason remained within 0.02–0.07 seconds of Direct across the three evaluated models.

Table 23: Wall-clock time to the final answer token. Post denotes answer-stopped Post-Reason.

Strategy	**Llama-3.1 (8B)**	**Llama-3.3 (70B)**	**Qwen-3.5 (9B)**
Direct	0.02–0.05 s	0.07–0.17 s	0.04–0.08 s
Post	0.05 s	0.15–0.24 s	0.06–0.09 s

Table 24: Single-stream decoding throughput and median time to first token (TTFT). Ranges show the minimum and maximum across nine benchmarks.

Model	**Decode throughput (tokens/s)**	**Median TTFT**
Llama-3.1 (8B)	176 (156–181)	26 ms
Llama-3.3 (70B)	45 (42–46)	77 ms
Qwen-3.5 (9B)	169 (156–171)	40 ms

Across the 90 evaluated model–benchmark cells, the answer preceded the justification and was parsed successfully on at least 99.6% of examples. Median tokens to answer were 10.5 for Direct and 10.9 for Post-Reason. The larger generation limit for Post-Reason applies only when the post-answer justification is retained.

F PROMPTING - EXTENDED EXPERIMENTAL DETAILS: POST-TASK ABLATION

In Table 11, we explored the efficacy of different post-generation tasks to isolate the effect of logical justification. To isolate whether the content generated after the answer matters, we compare three answer-first continuation tasks: a justification, a summary, and a confidence statement. This ablation was conducted across three representative models from our suite: Llama-3.1 (8B), Gemma-3 (12B), and Mistral-Small (24B). To ensure a fair comparison, the same questions, 3-shot examples, answer representation, and decoding configuration are used throughout. Only the requested post-answer content changes.

Table 25 gives the common, benchmark-independent suffixes.

Table 25: Unified suffixes for the post-task ablation.

Condition	Appended suffix
Post-Reason	`State the final answer immediately and justify your answer.`
Post-Summary	`State the final answer immediately and then provide a concise summary.`
Post-Confidence	`State the final answer immediately and then report your confidence with a brief explanation.`

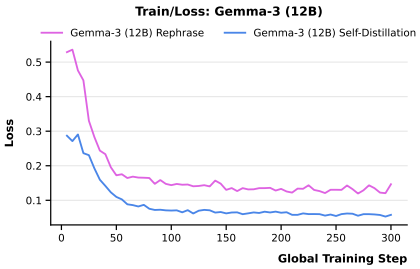
The corresponding results are reported in Table 11 in the main paper. Under a comparable answer-first format and continuation budget, Post-Summary underperforms Post-Confidence, which in turn underperforms Post-Reason. This pattern supports the interpretation that the requested justification contributes more than continuation length alone.

G SUPERVISED POST-REASON TUNING - EXTENDED OPTIMIZATION DYNAMICS

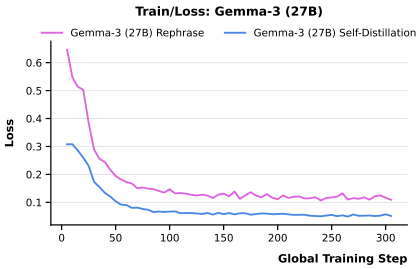
To demonstrate that the convergence benefits of self-distillation are not architectural anomalies, we provide the paired training loss curves (Self-Distillation vs. Standard Rephrased Distillation) across the complete suite of fine-tuned models.

To ensure a rigorous comparison, all paired training runs were executed utilizing identical hyperparameters (detailed in Appendix C), hardware configurations, and random seeds. As observed in Figure 3, target-conditioned self-distillation yields strictly superior optimization dynamics characterized by three distinct phenomena:

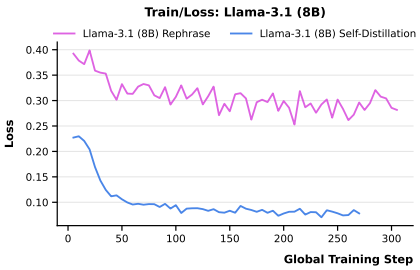
- **Accelerated Initial Convergence:** Across all architectures, the self-distilled objective achieves a lower loss basin significantly faster during the first epoch.
- **Gradient Stability:** The standard rephrased distillation curves exhibit higher variance and periodic spiking, whereas the targeted masking approach produces a distinctly smoother descent trajectory.
- **Scale Invariance:** The smoothing effect and lower-bound convergence hold true regardless of the base model's parameter count (spanning 4B to 70B) or lineage (Gemma, Llama, Ministral, Qwen).



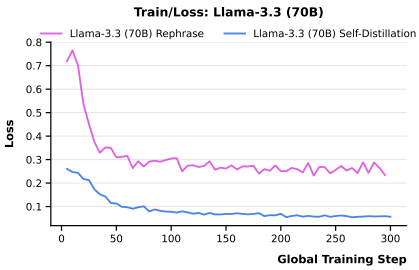
(a) Gemma-3 (12B)



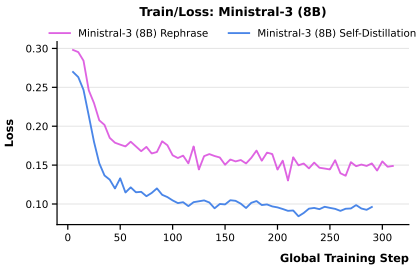
(b) Gemma-3 (27B)



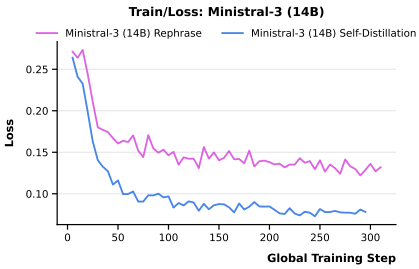
(c) Llama-3.1 (8B)



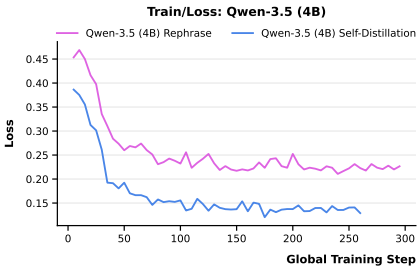
(d) Llama-3.3 (70B)



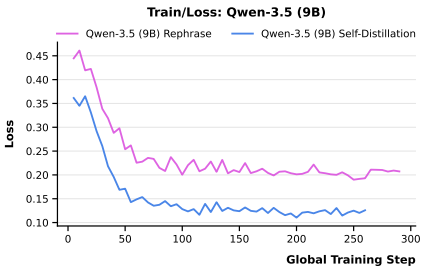
(e) Ministral-3 (8B)



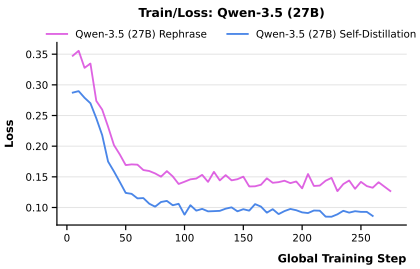
(f) Ministral-3 (14B)



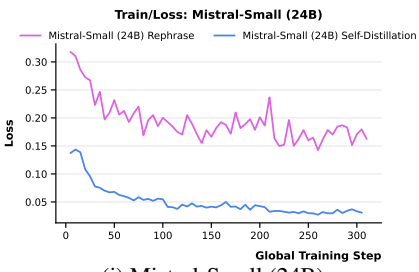
(g) Qwen-3.5 (4B)



(h) Qwen-3.5 (9B)



(i) Qwen-3.5 (27B)



(j) Mistral-Small (24B)

Figure 3: Extended loss convergence comparisons for all 10 models. Solid blue lines represent the proposed Self-Distillation, while magenta lines represent the baseline Rephrased Distillation. Across all scales and architectures, the proposed method yields lower variance and faster convergence.

H SUPERVISED POST-REASON TUNING - ABLATION STUDY: TRAINING CORPUS DEPENDENCY

To isolate the source of the optimization benefits observed in our primary experiments, we conducted a data-centric ablation study. We held the Target-Conditioned Self-Distillation methodology constant but replaced the training corpus. We compared models fine-tuned on the Numina dataset (which is heavily biased toward complex, multi-step mathematical reasoning) against models fine-tuned on the Massive Multitask Language Understanding (MMLU) dataset (which is predominantly multiple-choice factual recall).

Our objective was to determine whether the convergence advantages of our method are universally applicable or intrinsically tied to the structural depth of the training data.

H.1 OPTIMIZATION DYNAMICS: REASONING VS. FACTUAL RECALL

Target-conditioned self-distillation is designed to force the model to refine its intermediate chain of thought by masking the final answer. In mathematical problem-solving (Numina), these intermediate steps are critical algorithmic derivations. In factual tasks (MMLU), the intermediate steps are often shallow justifications that bottleneck on the base model's parametric memory.

This theoretical distinction is clearly visible in the training dynamics. When applied to the Numina dataset (as shown previously in Appendix G), our method achieves a significantly lower loss basin compared to standard distillation. However, when applied to the MMLU dataset (Figure 4), this advantage vanishes. The loss trajectories on the factual dataset remain entangled with baseline expectations, demonstrating that the masking mechanism requires complex reasoning pathways to optimize effectively.

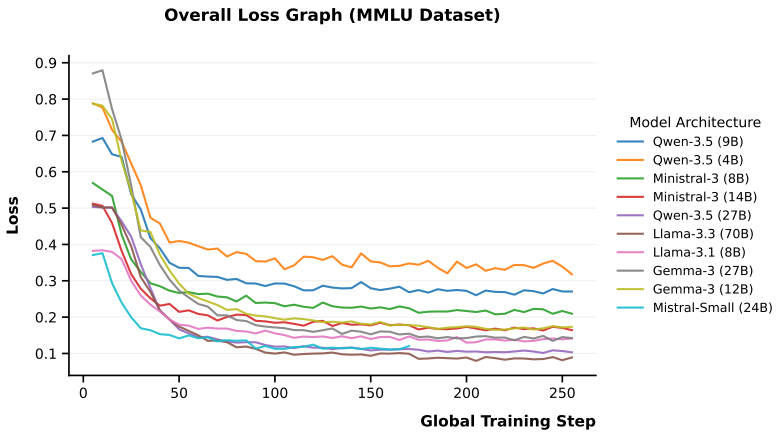


Figure 4: Aggregate training loss convergence across all models on the MMLU dataset. When the training corpus is swapped from reasoning-heavy data (Numina) to factual data (MMLU), the optimization benefits of Target-Conditioned Self-Distillation are neutralized.

H.2 DOWNSTREAM BENCHMARK PERFORMANCE

The neutralization of optimization benefits on the MMLU training set correlates directly with downstream performance. To quantify this, we evaluated the suite of MMLU-trained self-distilled models across our standard benchmark battery.

As detailed in Table 26, utilizing the Numina training corpus yielded better performance compared to the MMLU corpus across all three complex reasoning benchmarks—BBH, GPQA, and MMLU-Pro. This ablation provides conclusive evidence that Target-Conditioned Self-Distillation thrives when applied to reasoning-dense datasets. The deep algorithmic trajectories inherent in Numina provide the necessary structural complexity to effectively guide the self-distillation objective and shape the

Table 26: Downstream performance comparison between MMLU SFT and Numina SFT. Values represent accuracy in percentage points. Δ represents the relative percentage improvement of Numina SFT over MMLU SFT.

	BBH			GPQA			MMLU-Pro		
Model	MMLU	Numina	Δ	MMLU	Numina	Δ	MMLU	Numina	Δ
Llama Family									
Llama-3.1 (8B)	41.28	53.67	+30.01%	32.81	34.16	+4.11%	35.67	36.80	+3.17%
Llama-3.3 (70B)	64.74	62.20	-3.92%	50.11	50.11	+0.00%	52.83	53.13	+0.57%
Gemma Family									
Gemma-3 (12B)	58.88	59.27	+0.66%	35.28	34.16	-3.17%	43.37	44.57	+2.77%
Gemma-3 (27B)	65.09	65.47	+0.58%	35.73	36.40	+1.88%	50.77	51.27	+0.98%
Mistral & Ministral									
Ministral-3 (8B)	55.63	55.64	+0.02%	37.98	35.51	-6.50%	47.07	47.97	+1.91%
Ministral-3 (14B)	59.55	60.42	+1.46%	38.88	40.22	+3.45%	53.33	53.60	+0.51%
Mistral-Small (24B)	59.55	59.94	+0.65%	42.47	45.17	+6.36%	54.13	54.53	+0.74%
Qwen-3.5 Family									
Qwen3.5 (4B)	53.16	53.99	+1.56%	43.60	43.37	-0.53%	45.23	45.67	+0.97%
Qwen3.5 (9B)	57.64	58.47	+1.44%	60.00	60.90	+1.50%	51.20	51.50	+0.59%
Qwen3.5 (27B)	68.24	68.78	+0.79%	61.57	63.60	+3.30%	65.50	65.53	+0.05%

loss landscape, proving the technique acts as a powerful reasoning multiplier rather than a simple knowledge injector.