Scale or Reason? A Compute-Equivalent Analysis of Reasoning Distillation

Nicolas Boizard[A,E], Hippolyte Gisserot-Boukhlef[B,E], Kevin El Haddad[A,D], Céline Hudelot[E] and Pierre Colombo[C]
[A]Diabolocom
[B]Artefact Research Center
[C]Cohere
[D]ISIA Lab, University of Mons
[E]MICS, CentraleSupélec, Université Paris-Saclay

Distilling reasoning traces from strong teacher models has become the standard recipe for building capable small language models. Yet reasoning traces are 5-20× longer than standard instruction fine-tuning (IFT) outputs, meaning every practitioner who chooses reasoning distillation implicitly forgoes training a larger IFT model on the same compute budget. Whether this trade-off is worthwhile remains unaddressed. We study it with a controlled experiment: a single teacher generates paired IFT and reasoning outputs for identical prompts by toggling only its reasoning mode, isolating supervision format as the sole variable. Training students at five scales (0.5B to 14B) and evaluating on 18 benchmarks, we find that at matched FLOPs, IFT lies on or near the Pareto frontier across the majority of configurations. Reasoning reaches the Pareto frontier only on open-ended tasks at 7B and above. Even there, a sequential curriculum mixing just 25-50% reasoning data with IFT captures most of the accuracy benefit at far lower compute cost.

Keywords: *reasoning distillation, instruction fine-tuning, compute scaling, language models*

1. Introduction

Reasoning distillation has become the dominant training recipe for capable small language models (Guo et al., 2025; Yang et al., 2025; Rastogi et al., 2025). Yet every reasoning trace is 5-20× longer than a standard IFT output (Figure 1), and this length gap has a direct cost: training on reasoning data consumes proportionally more compute. A practitioner with a fixed budget who chooses reasoning distillation implicitly gives up the option of training a larger IFT model instead, an alternative that standard scaling laws suggest is a highly efficient use of additional compute (Kaplan et al., 2020; Hoffmann et al., 2022). Whether reasoning justifies this cost has not been studied: DeepSeek-R1 (Guo et al., 2025), s1 (Muennighoff et al., 2025), and Snell et al. (2024) all evaluate at fixed model sizes, leaving the compute-allocation question entirely open. This leads directly to our question:

For a given compute budget, should a practitioner invest in reasoning distillation, or train a larger IFT model?

To isolate the effect of supervision format from all other variables, we use a single teacher that generates paired IFT and reasoning outputs for identical prompts by toggling its reasoning mode on and off. Both formats are thus distilled from the same teacher weights in a controlled setup, ensuring that supervision format is the sole variable. We train `Qwen2.5` students at five scales (0.5B to 14B), track FLOPs precisely across all configurations, and evaluate on 18 benchmarks spanning four task families. We replicate all key findings using a second teacher-student pair (`Nemotron-Super-49B`

Accepted the Conference on Empirical Methods in Natural Language Processing (EMNLP) 2026.
Corresponding author(s): nicolas.boizard@centralesupelec.fr

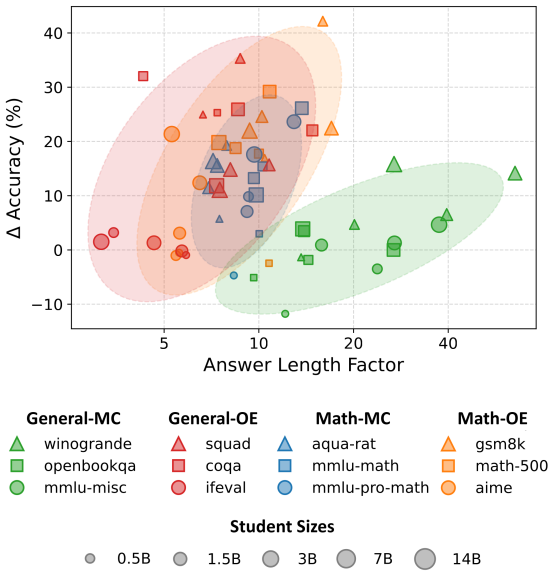


Figure 1 | Compute overhead of reasoning distillation. Each point is a benchmark-model-size pair. X-axis: extra-token factor when switching from IFT to reasoning supervision. Y-axis: accuracy gain (%). The overhead is large (5-20×) and heterogeneous across tasks.

with `Gemma-3`; Appendix G), confirming that results are not specific to a single architecture or teacher. This yields three findings:

- **Scaling IFT is compute-efficient by default.** At matched training and inference FLOPs, IFT models lie on or near the Pareto frontier across the majority of configurations. Reasoning only approaches the frontier on open-ended tasks and at larger scales (§3).
- **Format and capacity jointly determine the value of reasoning.** Output format, rather than knowledge domain, is the primary predictor: open-ended tasks consistently benefit from reasoning distillation, whereas multiple-choice tasks do not. Model capacity acts as a secondary prerequisite, as reasoning captures the Pareto frontier only at 7B parameters and above (§4).
- **A little reasoning goes a long way.** In the regime where reasoning is beneficial, a sequential curriculum using just 25 to 50% reasoning data captures most of the accuracy benefit at a fraction of the training cost. Furthermore, mixed training below the 50% threshold delivers enhanced accuracy while strictly maintaining efficient inference costs (§5).

Together, these findings provide a systematic answer to the compute allocation question in reasoning distillation.

2. Experimental Setup

2.1. Distillation

Controlled paired design. Our central methodological challenge is to compare IFT and reasoning supervision without confounding label quality with supervision format. We address this by using a single teacher, `Qwen3-235B-A22B` (Yang et al., 2025), to generate both formats for every prompt: IFT outputs are produced with reasoning mode disabled, reasoning outputs with it enabled. Student models in the IFT condition are trained on ⟨question, answer⟩ pairs; in the reasoning condition on ⟨question, trace, answer⟩ tuples. Because both formats come from the same teacher weights and the

Category		0.5B		1.5B		3B		7B		14B		235B
		IFT	Reasoning	IFT	Reasoning	IFT	Reasoning	IFT	Reasoning	IFT	Reasoning	Teacher
General-purpose Training												
General	OE	24.9	41.3 (▲ +16.4)	37.0	60.5 (▲ +23.5)	52.2	64.7 (▲ +12.5)	56.4	70.4 (▲ +14.0)	66.1	74.2 (▲ +8.1)	84.4
	MC	49.6	43.5 (▼ -6.1)	65.2	65.0 (▼ -0.2)	72.0	75.6 (▲ +3.6)	79.1	84.2 (▲ +5.2)	82.3	90.4 (▲ +8.1)	95.4
Math	OE	17.7	22.4 (▲ +4.7)	24.1	43.7 (▲ +19.6)	39.9	55.4 (▲ +15.5)	45.0	66.3 (▲ +21.3)	53.6	74.7 (▲ +21.1)	92.1
	MC	28.6	29.9 (▲ +1.3)	41.2	56.1 (▲ +14.8)	60.3	70.9 (▲ +10.6)	63.9	85.7 (▲ +21.8)	76.2	90.9 (▲ +14.7)	94.6
Math-centric Training												
General	OE	31.8	8.2 (▼ -23.6)	50.8	38.3 (▼ -12.5)	59.8	48.6 (▼ -11.2)	66.5	60.4 (▼ -6.1)	70.8	69.2 (▼ -1.6)	84.4
	MC	45.7	30.8 (▼ -14.9)	62.9	54.8 (▼ -8.1)	70.0	69.5 (▼ -0.5)	76.5	83.2 (▲ +6.6)	81.5	90.3 (▲ +8.7)	95.4
Math	OE	14.5	9.2 (▼ -5.3)	36.5	39.7 (▲ +3.2)	45.5	57.5 (▲ +12.0)	59.6	73.6 (▲ +14.0)	66.3	81.3 (▲ +14.9)	92.1
	MC	28.2	20.0 (▼ -8.2)	49.7	51.9 (▲ +2.2)	65.2	72.2 (▲ +7.0)	78.9	87.3 (▲ +8.4)	83.8	91.4 (▲ +7.6)	94.6

Table 1 | Average scores for IFT versus reasoning distillation by model size, corpus, and task format. Δ denotes absolute accuracy shifts from standard IFT. Accuracy breakdowns by dataset are detailed in Appendix J.

same prompts, supervision format is the sole independent variable.

Students. We train `Qwen2.5` base models (Qwen-Team, 2024) at five parameter scales: 0.5B, 1.5B, 3B, 7B, and 14B. Students are drawn from a different model family than the teacher to limit shared pretraining overlap. Training data covers two domains: a general-purpose corpus of 1.3M pairs from `Infinity-Instruct` (Li et al., 2025), and a reasoning-centric corpus of 300K pairs from the math subset of the `Llama-Nemotron-Post-Training-Dataset` (Bercovich et al., 2025).

Training. We study sequential training across reasoning proportions $\rho \in \{0\%, 25\%, 50\%, 75\%, 100\%\}$, where ρ is the percentage of reasoning instances in the dataset. At $\rho = 0\%$ the model sees only IFT supervision; at $\rho = 100\%$ only reasoning supervision. Intermediate values correspond to a curriculum where IFT examples come first, followed by reasoning examples. We also evaluate a mixed training baseline in which IFT and reasoning data are shuffled together within a single phase (§5). All models train for one epoch on 16 H100 GPUs with a fixed global batch size of 32; learning rates are selected by grid search (Appendix C).

2.2. Evaluation

Tasks. Our evaluation centers on 18 benchmarks structured along two dimensions: domain (general vs. math) and answer format (multiple-choice (MC) vs. open-ended (OE)). This taxonomy defines four task families: General-MC (`winogrande` (Keisuke et al., 2020), `openbookqa` (Mihaylov et al., 2018), `mmlu-misc` (Hendrycks et al., 2021)), General-OE (`squad` (Rajpurkar et al., 2016), `coqa` (Reddy et al., 2019), `ifeval` (Zhou et al., 2023)), Math-MC (`aqua-rat` (Ling et al., 2017), `mmlu-math` (Hendrycks et al., 2021), `mmlu-pro-math` (Wang et al., 2024)), and Math-OE (`gsm8k` (Cobbe et al., 2021), `math-500` (Lightman et al., 2024), `aime` (Zhang and Math-AI, 2025)). Additional tasks are introduced selectively in §4 to address specific experimental needs.

LLM-as-a-Judge. Answers are scored by a majority-vote jury of three judge models: `Llama-3.3-70B-Instruct` (Grattafiori et al., 2024), `Nemotron-Ultra-253B-v1` (Bercovich et al., 2025), and `GPT-OSS-120B` (Agarwal et al., 2025). Each judge follows a two-step protocol: first extracting

the final answer from the model output while ignoring any intermediate reasoning, then assessing its semantic equivalence to the reference. This ensures that scores reflect answer correctness, not trace quality, and prevents reasoning-trained models from receiving credit for verbose but wrong answers. Median absolute error across judges is 1.2 points; pairwise agreement is below 2 points on all benchmarks. Distilled models are evaluated zero-shot while base checkpoints rely on three-shot prompting. Full prompts and judge statistics are provided in Appendix B.

Compute cost. Because reasoning traces are 5-20× longer than IFT outputs, reasoning training and inference consume proportionally more compute for both forward and backward passes. We systematically track this overhead in FLOPs and report the formula used in Appendix D. This disparity is the foundation for the compute-matched comparisons in §3.

3. Is Reasoning Worth the Compute Cost?

We approach this question in three distinct steps: comparing the two supervision formats at equal model size, at matched training compute, and at matched inference compute.

3.1. At Fixed Size, Reasoning Helps

To establish an initial baseline, Table 1 presents the raw performance of both supervision paradigms across all model sizes and the two relevant training corpora: the general-purpose one derived from `Infinity-Instruct` and the math-centric one derived from `Llama-Nemotron-Post-Training-Dataset`.

General-purpose training. Reasoning supervision consistently outperforms IFT when trained on the general-purpose corpus, with the largest gains on open-ended and mathematical tasks. On Math-OE, a 3B reasoning model already surpasses the 14B IFT baseline outright (55.4 vs. 53.7); on General-OE, it comes within 1.4 points of it (64.7 vs. 66.1). On both MC families, the 14B reasoning model narrows the performance gap with the teacher to within 4-5 points, despite being roughly 17× smaller. The pattern holds across nearly all conditions with one notable exception: on General-MC, IFT leads at 0.5B, and the two formats remain effectively tied at 1.5B. However, from 3B onward, reasoning re-establishes a definitive lead.

Math-centric training. When trained on the math-centric corpus, reasoning supervision again outperforms IFT from 1.5B onward on math tasks. More strikingly, from 3B upward, math-centric reasoning models surpass even general-purpose reasoning on both Math-OE (57.5 vs. 55.4) and Math-MC (72.2 vs. 70.9), despite training on just 300K samples compared to 1.3M. However, this specialization comes at a cost: on general-domain tasks, math-centric reasoning lags well behind general-purpose reasoning at smaller scales, and the gap closes only partially by 14B (General-MC: 90.3 vs. 90.4; General-OE: 69.2 vs. 74.2). Conversely, standard IFT models trained on the exact same math-centric corpus do not exhibit this penalty, remaining competitive on general-domain tasks across all tested scales. This demonstrates that the general-task degradation is tied to the reasoning format rather than the math-centric data itself. This penalty is particularly severe at smaller scales, demonstrating that distilling general capabilities from a reasoning-intensive domain (e.g., mathematical problem solving or code generation, as commonly used in RL pipelines) strictly requires larger parameter counts. Consequently, we report the remainder of our experiments with the general-purpose training corpus.

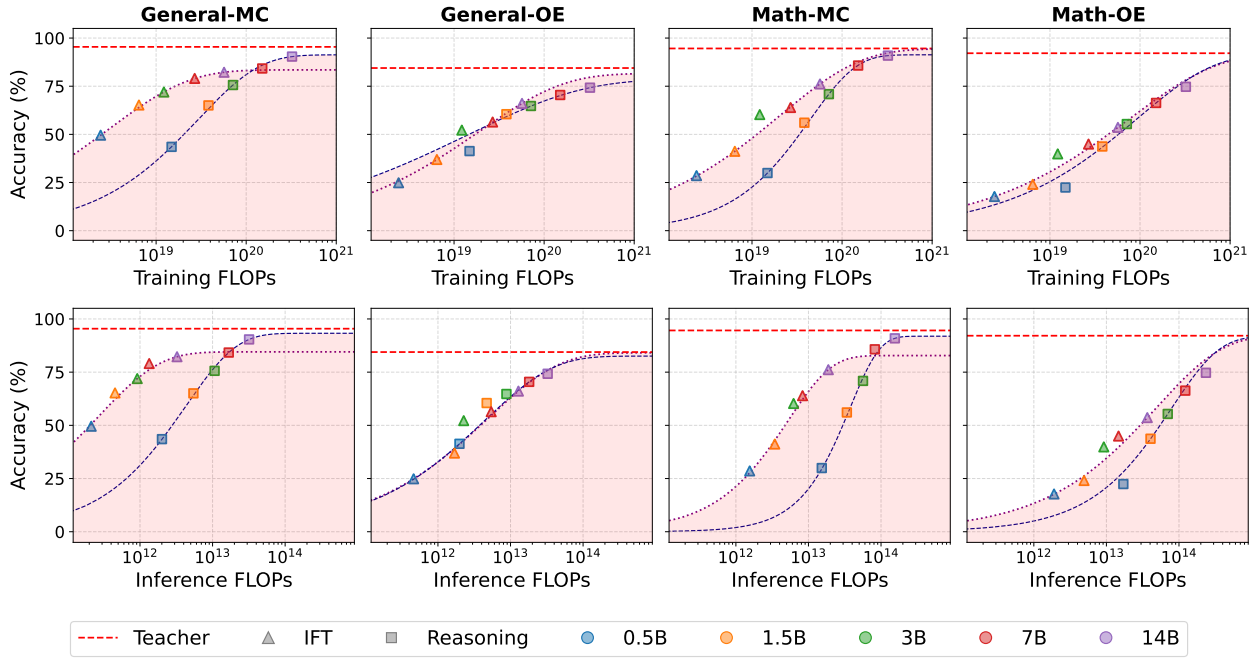


Figure 2 | Accuracy as a function of training FLOPs (top) and inference FLOPs (bottom) for IFT and reasoning models on the general-purpose corpus. Each point corresponds to one model size; curves are saturating growth fits to the observed data (see Appendix D for details).

The limits of parameter-matched comparisons. Both supervision formats improve with model size, but at different rates. IFT improves steadily across all four task families, exhibiting consistent gains at each size step; yet even at 14B, it remains well below teacher accuracy, with the largest gap appearing on open-ended tasks. Reasoning scales more steeply and outperforms IFT in 70% of scale-corpus configurations (28 out of 40 pairs), though not uniformly. While these results make a strong case for reasoning at a fixed parameter count, this comparison carries a fundamental limitation: equal model size does not imply equal compute. Because a 3B reasoning model consumes substantially more FLOPs than a 3B IFT model, this disparity changes the efficiency picture entirely.

3.2. At Fixed FLOPs, IFT Dominates

In this section, we shift our focus to compute efficiency by plotting accuracy against training and inference FLOPs for IFT and reasoning distillation on the general-purpose corpus.

Training efficiency. Contrary to the fixed capacity comparison, when compute is held constant rather than model size, the efficiency picture reverses entirely. Figure 2 reveals that IFT models define the Pareto frontier of performance as a function of the training budget, with no reasoning configuration matching the accuracy of a compute-equivalent IFT model at any tested scale. This establishes that scaling model size is a more compute-efficient strategy for maximizing performance across the majority of our evaluations. For example, on general multiple-choice tasks, despite being twice the size, a 7B IFT model required 5× less training compute to achieve similar accuracy compared to a 3B reasoning-distilled model. While the cost overhead is smaller on open-ended tasks, a gap remains; for instance, on open-ended math tasks, a 7B IFT model achieves performance similar to a 1.5B reasoning model while being 1.6× less costly to train. However, we observe that the performance gap at matched training FLOPs continuously narrows with model scale, suggesting this disparity may

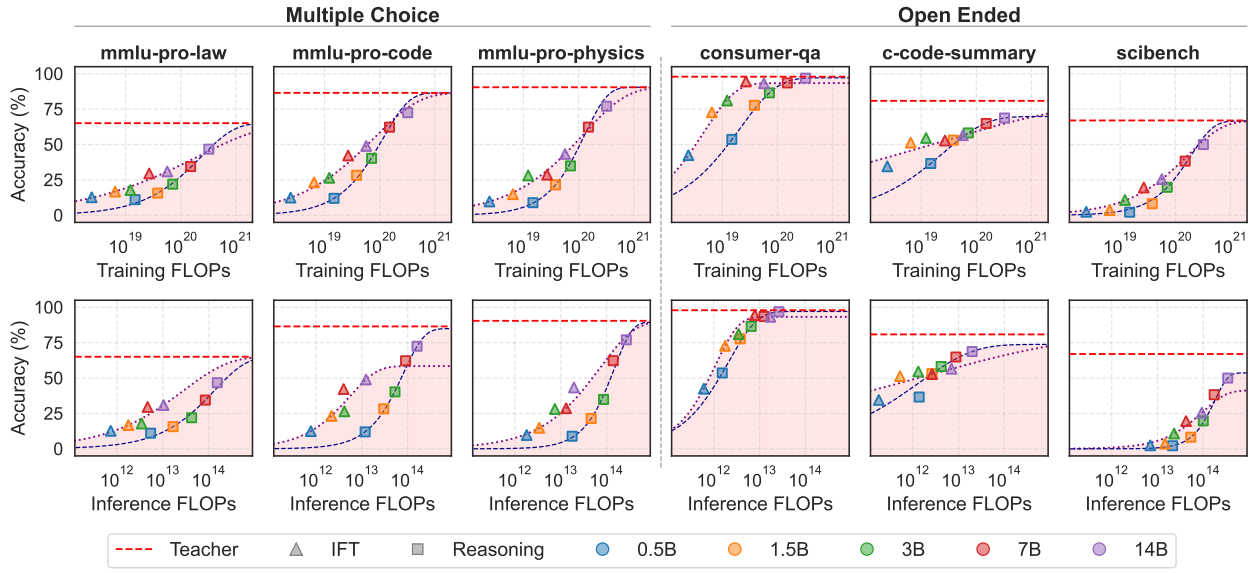


Figure 3 | Accuracy as a function of training FLOPs (top) and inference FLOPs (bottom) for IFT and reasoning models on reasoning-intensive tasks.

close beyond 14B parameters.

Inference efficiency. Shifting focus from training to deployment, the pattern holds (Figure 2). Because reasoning models generate outputs 7-15× longer than standard IFT models of the same size, their inference cost is substantially higher at every scale. On OE tasks, reasoning models still generate outputs approximately 7× longer than their compute-equivalent IFT counterparts, yet they approach or even join the Pareto frontier. On MC tasks, however, this overhead grows to ~10-15×, causing reasoning models to fall well behind IFT efficiency across most evaluated scales.

4. What Drives Reasoning Efficiency?

§3 demonstrates that while standard IFT generally dominates at fixed compute, systematic exceptions emerge for open-ended tasks at larger parameter scales. This section investigates why output format and model capacity dictate when reasoning distillation justifies its computational overhead.

4.1. Format Predicts Value; Domain Does Not

Reasoning models incur significantly higher overhead on multiple-choice tasks than on open-ended tasks across diverse domains, questioning whether distillation efficiency is driven by domain complexity or output format.

Isolating format from domain. To disentangle these factors further, we evaluate three additional reasoning-intensive domains featuring one open-ended and one multiple-choice benchmark each: law (`mmlu_pro_law` (Wang et al., 2024), `consumer_qa` (Guha et al., 2023)), code (`mmlu_pro_code`, `C_Code_Summarization` (Liu et al., 2020)), and physics (`mmlu_pro_physics`, `scibench` (Wang et al., 2023)). Holding the knowledge domain constant while varying task format provides a precise test. If domain dictates efficiency, reasoning-intensive tasks will uniformly favor reasoning models.

Conversely, if output format dominates, open-ended and multiple-choice versions of the same domain will diverge, with one capturing or resting near the Pareto frontier alongside IFT models, and the other falling behind.

Output format dictates efficiency for reasoning-intensive tasks. As initially illustrated in Figure 2 and further complemented by Figure 3, the two task formats exhibit a clear divergence: open-ended benchmarks place reasoning models on or near the Pareto frontier for both training and inference efficiency, while a significant gap remains for multiple-choice evaluations. For example, on `C_Code_Summarization`, a 7B reasoning model matches an IFT model at identical inference cost with a training overhead of only 1.3×. Conversely, on multiple-choice benchmarks like `mmlu_pro_code`, a 10× inference compute gap exists between a 7B IFT model and a 3B reasoning model achieving similar performance, although this narrows with scale. These results confirm that while knowledge domain exerts marginal influence (math and reasoning-intensive multiple-choice tasks benefit slightly more than general ones), output format strictly dictates efficiency. For open-ended setups, this Pareto equivalence transforms the architectural decision into a practical exchange between model capacity and generation length, allowing practitioners to deploy smaller reasoning models for memory constraints, or larger IFT models to minimize latency.

4.2. Model Capacity Determines Whether Traces Help or Hurt

While output format dictates where reasoning distillation is computationally viable, model capacity acts as a strict secondary prerequisite.

Reasoning dominates from 7B upward. Beyond this capacity threshold, reasoning models reliably capture both the training and inference Pareto frontiers. On open-ended tasks (Figure 2, Figure 3), reasoning models match or exceed IFT accuracy while maintaining optimal compute efficiency. On multiple-choice tasks, reasoning models at this scale surpass IFT in absolute accuracy, though they incur higher inference costs that deviate from established scaling trends.

Incorrect traces are longer, not deeper. The relative difference in inference cost between consecutive reasoning model scales decreases steadily with model size (Figure 2, Figure 3, and highlighted in Appendix I). Plotting the generation length of correct versus incorrect reasoning traces against model size and accuracy (Figure 4) reveals a consistent relationship: across all scales and task families, incorrect answers yield traces 1.5 to 2× longer than correct ones. Rather than reflecting deeper processing, longer traces correlate directly with failure. Qualitative inspection reveals smaller models frequently explore dead ends or restate problems without making tangible progress. As capacity increases, two phenomena emerge:

1. The length ratio between failures and correct answers shrinks (from roughly 2× at 0.5B to 1.5× at 14B), indicating larger models allocate generation effort more efficiently.
2. Because accuracy inherently increases with scale, the average output length naturally gravitates toward the shorter, correct trace length.

Conversely, average generation length for smaller models is skewed by long, incorrect traces, severely degrading the accuracy to inference cost ratio. This makes small reasoning models computationally expensive and practically ineffective. While this strong correlation makes trace length a natural confidence signal (similar to multi sampling strategies utilizing answer length as an uncertainty measure (Hassid et al., 2025)), attempting a simple early stopping strategy based on length thresholds

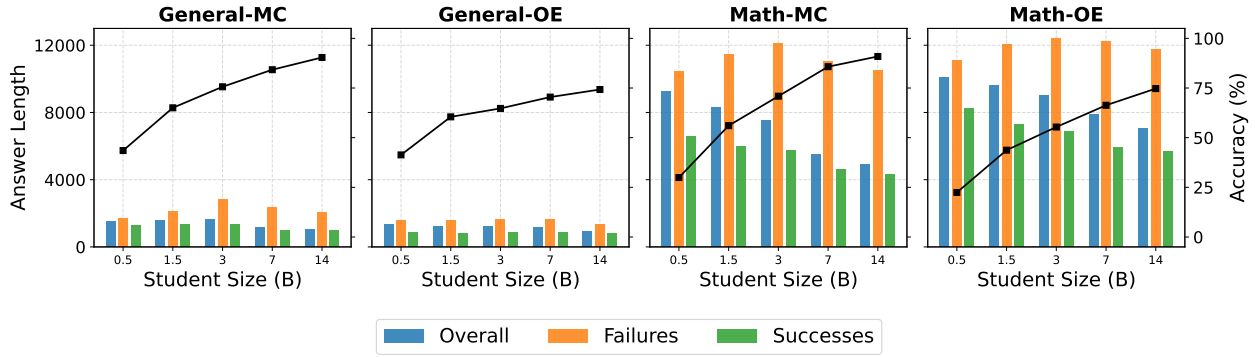


Figure 4 | Answer length analysis across student sizes and correctness in reasoning models.

proved unsuccessful. The reduction in inference FLOPs is entirely offset by a corresponding drop in accuracy without shifting the Pareto frontier, an outcome we consequently report in Appendix H.

Format and capacity jointly define the beneficial regime. Together, output format and model capacity define when reasoning distillation earns its compute cost. open-ended formats make reasoning a viable alternative to IFT: both reach comparable accuracy on the efficiency frontier, reducing the choice to a deployment constraint. Scaling beyond 7B pushes this further: reasoning overtakes IFT on the frontier, becoming the dominant strategy for open-ended tasks.

5. Leveraging IFT and Reasoning with Hybrid Data Mixtures

§3 demonstrates that reasoning distillation boosts performance but carries a steep compute premium, with training and inference costs running 5 to 20× higher than standard IFT. §4 traced this overhead to smaller models generating inefficiently: producing long traces that correlate with failure rather than reasoning depth, driving up costs without proportionate accuracy gains. This section investigates whether this computational burden can be reduced. We study two hybrid strategies blending IFT- and reasoning-style data in varying proportions: *sequential training* (IFT followed by reasoning) and *mixed training* (both data types combined and shuffled in a single training pass).

Sequential training recovers most reasoning gains with minimal reasoning data. Sequential training addresses a direct question: how much reasoning data is required? Figure 5 demonstrates that performance gains emerge at just a 25% reasoning ratio (meaning 75% IFT data followed by 25% reasoning traces) and remain broadly consistent across task families and model sizes, with the notable exception of the smallest models (0.5B and 1.5B) evaluated on general multiple-choice tasks. At every intermediate ratio (25%, 50%, and 75%), sequential models successfully rest on the Pareto frontier for open-ended tasks. This matches the efficiency of full reasoning distillation at a mere fraction of its training cost.

Mixed training enables reasoning improvements without increasing inference cost. Mixed training pursues a distinctly different objective. Rather than strictly maximizing overall accuracy, it investigates whether a model can acquire reasoning capabilities while still retaining the concise output style typical of standard IFT. Figure 6 demonstrates that this is achievable below a specific data threshold. At reasoning proportions between 25 and 50%, models improve substantially over

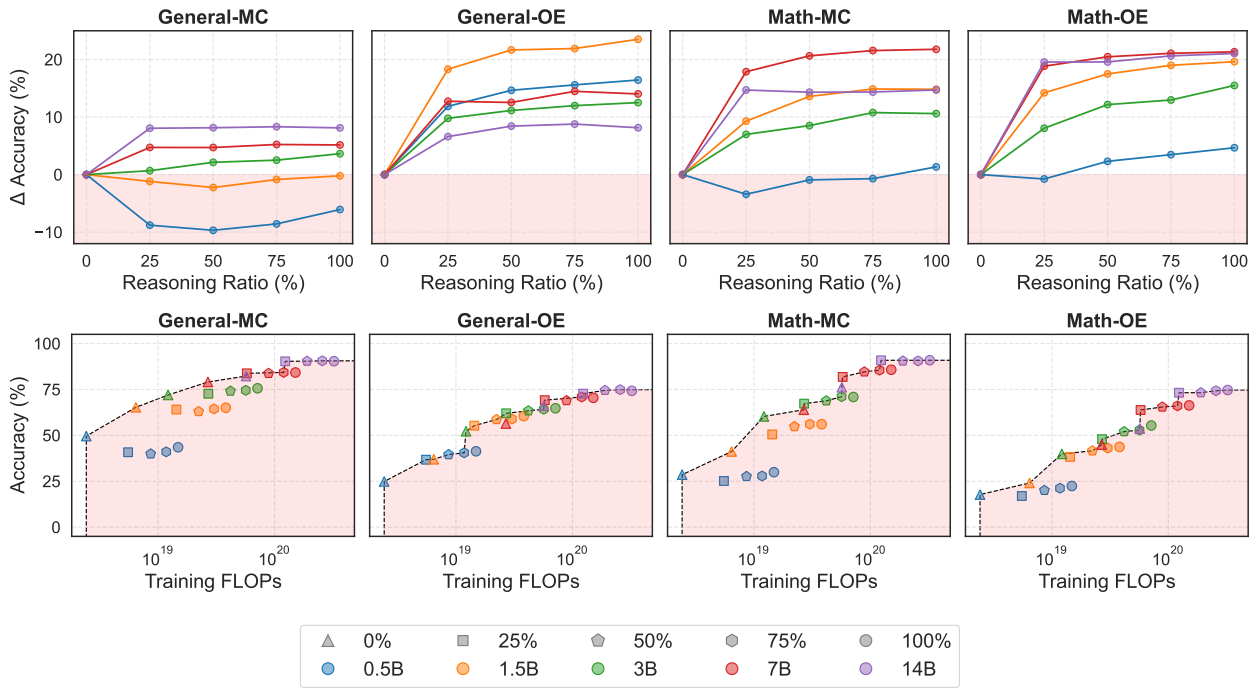


Figure 5 | Impact of the reasoning proportion on downstream performance in the sequential training scenario (standard IFT first, then reasoning). A 25% reasoning proportion captures most gains on Math_OE at a fraction of the compute. Results show the accuracy gap relative to the IFT baseline (0%).

the IFT baseline on mathematical tasks while maintaining short outputs. This allows them to gain accuracy without incurring the typical inference compute penalty.

Mixed training exhibits a phase transition once reasoning data becomes predominant, collapsing into purely reasoning-driven behavior. While mixed training remains beneficial at lower proportions of reasoning data, a clear mode switch emerges beyond the 50% threshold. Models begin to default entirely to reasoning style generation at inference time, which completely erases any inference cost advantage. Output lengths converge toward full reasoning distillation, while accuracy remains slightly behind that of sequential training at the 75% mark. Extending the benefits of mixed training beyond this point would likely require an explicit mechanism to control exactly when reasoning is invoked (Byrd and Srivastava, 2022; AlKhuzaey et al., 2024), a promising direction we leave for future work.

Sequential and mixed training serve complementary objectives. Sequential training reduces the training compute required to approach full reasoning performance: utilizing just 25% reasoning data recovers most accuracy gains at a fraction of the cost. Mixed training reduces inference cost: below the 50% threshold, models improve over standard IFT while preserving output brevity. Together, these findings establish that committing fully to reasoning distillation is unnecessary. When training compute is the primary constraint, sequential mixing efficiently captures most performance gains; when inference cost is the constraint, mixed training below the 50% threshold offers a practical middle ground.

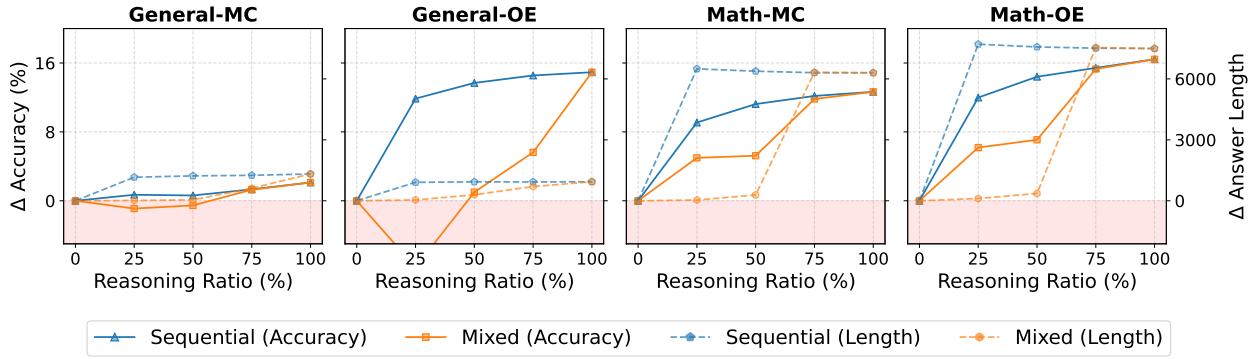


Figure 6 | Sequential vs mixed training across varying reasoning proportions. Solid lines indicate the accuracy gap relative to the IFT baseline, while dashed lines represent average answer length in tokens. All values are averaged across model sizes.

6. Related Work

Instruction tuning and reasoning. Instruction Fine-Tuning (IFT) has been the standard recipe for aligning LLMs with human instructions (Wei et al., 2021; Ouyang et al., 2022; Chung et al., 2024). Chain-of-Thought (CoT) extended this paradigm by supervising intermediate reasoning steps, yielding strong gains on arithmetic, symbolic, and commonsense reasoning benchmarks (Rajani et al., 2019; Nye et al., 2021; Cobbe et al., 2021; Wei et al., 2022; Kojima et al., 2022).

Knowledge distillation. Knowledge Distillation (KD) transfers capabilities from strong teachers to smaller students (Bucilua et al., 2006; Hinton et al., 2015). Beyond representation-based KD, text-based distillation has become central for reasoning: large teacher models generate either IFT- or reasoning-style traces that guide student learning (Kim and Rush, 2016; Zhou and Chiam, 2023; Hsieh et al., 2023; He et al., 2023). This approach reduces the cost of expensive RL while preserving performance (Guo et al., 2025; Yang et al., 2025; Rastogi et al., 2025). Despite its widespread adoption, the structural conditions under which reasoning distillation is actually more compute-efficient than standard IFT remain underexplored.

Compute-optimal scaling. The relationship between model capacity, dataset size, and training compute has been extensively formalized to guide the efficient pretraining of large language models (Kaplan et al., 2020; Hoffmann et al., 2022). Recently, this framework has expanded to encompass inference time scaling, revealing that allocating additional compute during generation can systematically enhance reasoning capabilities (Snell et al., 2024; Brown et al., 2024; Muennighoff et al., 2025). Recent research extends this analysis to API-based models, evaluating performance as a function of token cost (Chen et al., 2026) and finding analogous cost heterogeneity.

7. Conclusion

In this paper, we investigate whether the compute overhead of reasoning distillation is justified compared to scaling standard IFT, as a function of training and inference budgets. Surprisingly, while reasoning models boost raw performance, standard IFT lies on or near the Pareto frontier for the vast majority of configurations. Reasoning reaches the frontier only when two conditions hold simultaneously: the tasks require an open-ended output format rather than a specific knowledge domain, and the models contain at least 7B parameters. Below this capacity threshold, answer length

is dictated by failure, which heavily inflates inference costs. Additionally, we demonstrate that hybrid approaches can significantly reduce these compute costs. A sequential curriculum using just 25 to 50% reasoning data recovers most of the accuracy benefit at a fraction of the training cost, while mixed training below a 50% corpus mix threshold improves accuracy while strictly preserving standard IFT inference efficiency.

Limitations

This study focuses on models up to 14B parameters, covering the typical regime where distillation is utilized to enhance small model performance. While this reflects a practically motivated experimental setup, and our results establish that reasoning models capture the Pareto frontier at 7B (a finding further confirmed at the 14B scale), additional scaling experiments are necessary to verify whether the identified conditions remain stable beyond this capacity. Additionally, this work specifically examines the distillation paradigm. At the model scales we consider, reinforcement learning typically underperforms distillation and incurs substantially higher training costs, suggesting that our core observations may remain consistent in that setting. Nevertheless, a dedicated study using reinforcement learning remains a promising avenue for future work to confirm this hypothesis.

Ethics Statement

A central motivation of this work is to better contextualize the increasing emphasis on raw accuracy in modern language model development. Recent advances have led practitioners to adopt increasingly complex training and inference strategies, often involving substantially higher computational costs, in pursuit of incremental performance gains. While such gains are valuable, they are frequently reported without a clear account of their associated training and inference efficiency trade-offs. In this paper, we aim to provide a more balanced perspective by explicitly evaluating reasoning distillation and instruction fine-tuning under matched compute budgets. By quantifying both training and inference FLOPs alongside accuracy, we hope to encourage a more cost-aware evaluation of model design choices. Our findings suggest that simpler and more efficient approaches, such as IFT or carefully designed hybrid curricula, can often match or approach the performance of more expensive reasoning-based training regimes. We do not argue against the use of compute-intensive methods when they are justified by downstream needs. However, we emphasize the importance of aligning model complexity with practical constraints and deployment requirements. We hope this work contributes to a broader shift toward more targeted model development, where improvements in accuracy are systematically weighed against their computational and environmental costs. We additionally release all code and paired training datasets (IFT and reasoning outputs for the exact same inputs) to enable reproducibility and future controlled studies on reasoning.

Acknowledgments

We sincerely thank the Jean Zay supercomputer (GENCI-IDRIS-CNRS) through grant AD010617149. Furthermore, we gratefully acknowledge the support of the French government through the France 2030 program as part of the ArGiMi project.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Samah AlKhuzaey, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2024. Text-based question difficulty prediction: A systematic review of automatic approaches. *International Journal of Artificial Intelligence in Education*, 34(3):862–914.

Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, and 1 others. 2025. Llama-nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949*.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.

Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Matthew Byrd and Shashank Srivastava. 2022. Predicting difficulty and discrimination of natural language questions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 119–130.

Lingjiao Chen, Chi Zhang, Yeye He, Ion Stoica, Matei Zaharia, and James Zou. 2026. The price reversal phenomenon: When cheaper reasoning models end up costing more. *arXiv preprint arXiv:2603.23971*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

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. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, and 1 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. 2025. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *arXiv preprint arXiv:2505.17813*.

Nan He, Hanyu Lai, Chenyang Zhao, Zirui Cheng, Junting Pan, Ruoyu Qin, Ruofan Lu, Rui Lu, Yunchen Zhang, Gangming Zhao, and 1 others. 2023. Teacherlm: Teaching to fish rather than giving the fish, language modeling likewise. *arXiv preprint arXiv:2310.19019*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Sakaguchi Keisuke, Le Bras Ronan, Bhagavatula Chandra, and Choi Yejin. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327.

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

Jijie Li, Li Du, Hanyu Zhao, Bo-wen Zhang, Liangdong Wang, Boyan Gao, Guang Liu, and Yonghua Lin. 2025. Infinity instruct: Scaling instruction selection and synthesis to enhance language models. *arXiv preprint arXiv:2506.11116*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 158–167.

Shangqing Liu, Yu Chen, Xiaofei Xie, Jingkai Siow, and Yang Liu. 2020. Retrieval-augmented generation for code summarization via hybrid gnn. *arXiv preprint arXiv:2006.05405*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2381–2391.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20286–20332.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, and 1 others. 2021. Show your work: Scratchpads for intermediate computation with language models.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Qwen-Team. 2024. Qwen2.5: A party of foundation models.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4932–4942.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 2383–2392.

Abhinav Rastogi, Albert Q Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep Barmentlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, and 1 others. 2025. Magistral. *arXiv preprint arXiv:2506.10910*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Yikang Shen, Matthew Stallone, Mayank Mishra, Gaoyuan Zhang, Shawn Tan, Aditya Prasad, Adriana Meza Soria, David D Cox, and Rameswar Panda. 2024. Power scheduler: A batch size and token number agnostic learning rate scheduler. *arXiv preprint arXiv:2408.13359*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2023. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang bit Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yifan Zhang and Team Math-AI. 2025. American invitational mathematics examination (aime) 2025.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Tianxun Zhou and Keng-Hwee Chiam. 2023. Synthetic data generation method for data-free knowledge distillation in regression neural networks. *Expert Systems with Applications*, 227:120327.

A. Distillation Setup Positioning

While frontier laboratories increasingly rely on reinforcement learning (RL) to train their largest models, recent technical reports from DeepSeek-R1 (Guo et al., 2025), Qwen3 (Yang et al., 2025), and Magistral (Rastogi et al., 2025) consistently identify distillation as the standard approach for training smaller models within the same family. These works show that small- and mid-scale models trained solely with RL are both less performant and less sample-efficient than models distilled from stronger teachers.

In particular, DeepSeek reports that a model distilled from DeepSeek-R1 substantially outperforms its RL-only counterpart across benchmarks (Guo et al., 2025, Appendix F.1), achieving an average score of 71.55 compared to 58.45. Similarly, Mistral finds that a Mistral-Small-24B model distilled from Magistral-Medium reasoning traces attains a higher average score than an RL-only counterpart (60.71 vs. 58.91; Rastogi et al., 2025, Table 3). These results establish distillation not merely as a pragmatic choice for controlled experiments, but as a necessary ingredient for strong performance at smaller scales.

B. Prompting Details

This section provides examples of paired IFT-reasoning data (Table 3, Table 4), as well as the detailed prompts used for answer generation (Table 5) and evaluation by LLM judges (Table 6).

C. Training Hyperparameters

All training runs are performed for a single epoch on an identical number of samples with a global batch size of 32 items distributed across 16 H100 GPUs. This setup ensures that the number of samples per forward and backward pass remains strictly constant between standard IFT and reasoning training. The learning rate follows a Warmup-Stable-Decay (WSD) schedule (Shen et al., 2024) (150-step linear warmup, constant plateau, and 300-step linear decay to 10% of the peak value), using the `AdamW_fused` optimizer (Loshchilov and Hutter, 2017). Peak learning rates are selected via grid search over $\{2\times10^{-5}, 1\times10^{-5}, 7\times10^{-6}, 5\times10^{-6}, 3\times10^{-6}, 2\times10^{-6}, 1\times10^{-6}\}$. Table 2 lists the peak learning rates for all models and data formats.

Model	Reasoning	IFT
`Qwen2.5-0.5B`	2e-5	1e-5
`Qwen2.5-1.5B`	1e-5	7e-6
`Qwen2.5-3B`	7e-6	5e-6
`Qwen2.5-7B`	5e-6	3e-6
`Qwen2.5-14B`	3e-6	1e-6
`Gemma3-1B`	2e-5	1e-5
`Gemma3-4B`	5e-6	3e-6
`Gemma3-12B`	2e-6	1e-6

Table 2 | Peak learning rates selected for each student model and training data format.

D. FLOPs Computation

We compute training and inference FLOPs following Hoffmann et al. (2022).

$$\begin{aligned}\text{FLOPs}_{\text{forward}} = &\underbrace{2\,l\,V\,d_{\text{model}}}_{\text{embeddings}} + \underbrace{\left(6\,l\,d_{\text{model}}^2 + 2\,l^2\,d_{\text{model}} + 3\,l^2\,h + 2\,l^2\,d_{\text{model}} + 2\,l\,d_{\text{model}}^2\right)\cdot N_l}_{\text{attention}} \\ &+ \underbrace{4\,l\,d_{\text{model}}\,d_{\text{ff}}\,N_l}_{\text{feed-forward}} + \underbrace{2\,l\,d_{\text{model}}\,V}_{\text{output logits}}\end{aligned} \tag{1}$$

$$\text{FLOPs}_{\text{training step}} = 3 \cdot \text{FLOPs}_{\text{forward}} \tag{2}$$

$$\text{FLOPs}_{\text{training}} = \sum_{i=1}^{N_s} \text{FLOPs}_{\text{training step}}(i) \tag{3}$$

$$\begin{aligned}\text{FLOPs}_{\text{inference}} = &\underbrace{2\,l_p\,d_{\text{model}}\,V}_{\text{embeddings}} + \underbrace{\left(6\,l_p\,d_{\text{model}}^2 + 2\,l_p^2\,d_{\text{model}} + 3\,l_p^2\,h + 2\,l_p^2\,d_{\text{model}} + 2\,l_p\,d_{\text{model}}^2\right)\cdot N_l}_{\text{attention}} \\ &+ \underbrace{4\,l_p\,d_{\text{model}}\,d_{\text{ff}}\,N_l}_{\text{feed-forward}} + \underbrace{2\,d_{\text{model}}\,V}_{\text{output logits}}\end{aligned} \tag{4}$$

$$\begin{aligned}\text{FLOPs}_{\text{inference w/ cache}} = &\underbrace{2\,d_{\text{model}}\,V}_{\text{embedding}} + \underbrace{\left(6\,d_{\text{model}}^2 + 2\,l_p\,d_{\text{model}} + 3\,l_p\,h + 2\,l_p\,d_{\text{model}} + 2\,d_{\text{model}}^2\right)\cdot N_l}_{\text{attention}} \\ &+ \underbrace{4\,d_{\text{model}}\,d_{\text{ff}}\,N_l}_{\text{feed-forward}} + \underbrace{2\,d_{\text{model}}\,V}_{\text{logits}}\end{aligned} \tag{5}$$

$$\text{FLOPs}_{\text{multi-token inference}} = \text{FLOPs}_{\text{inference}}(l_p) + \sum_{i=1}^{l_g} \text{FLOPs}_{\text{inference w/ cache}}(l_p + i) \tag{6}$$

D.1. Notations

The following notation is used throughout the FLOPs computation formulas:

- V : vocabulary size
- d_{model} : hidden dimension of the model
- d_{ff} : dimension of feed-forward layers
- h : number of attention heads
- N_l : number of transformer layers
- l : sequence length
- l_p : prompt length
- l_g : generation length
- N_s : number of training samples

D.2. Training FLOPs

Equations 1 to 3 present the FLOPs computations associated with model training as a function of the model architecture and sequence length.

D.3. Inference FLOPs

Equations 4 to 6 detail the FLOPs required for model inference as a function of the model architecture, input sequence length, and output sequence length.

E. Details on Interpolation Curves

To predict the impact of further model scaling, we fit a saturating growth function to the observed data: $f(x) = \alpha + \beta(1 - \exp(-\gamma x^{\delta}))$, where x denotes FLOPs and $f(x)$ gives interpolated accuracy. Parameters are constrained so that $\alpha, \beta > 0$, $\alpha + \beta$ does not exceed teacher accuracy, $\gamma > 0$, and $0 < \delta \leq 1$. The function is fitted by minimizing mean absolute error.

F. Extended Task Evaluation

Main results for law, code, and physics domains are presented in §4 (Figure 3). Below we provide additional analyses: reasoning ratio sensitivity and inference Pareto analysis for these domains (Figure 7, Figure 8).

G. Alternative Teacher-Student Configurations

We extend our study using `Nemotron-Super-49B-v1.5` as teacher with Gemma 3 students (1B, 4B, 12B), training on 200K general-domain pairs (Figure 9, Figure 10, Figure 11).

H. Generation Early-Stopping

Leveraging the observation that incorrect reasoning traces are typically longer, we designed a simple early stopping strategy. However, as shown in Figure 12, this approach fails to improve the Pareto frontier, as any reduction in inference cost comes directly at the expense of accuracy. Conversely, certain mathematical tasks actively benefit from extended generation lengths in the reasoning setting (Figure 13), demonstrating that reasoning models are capable of extrapolating well beyond their training time trace lengths.

I. Inference Cost Scaling Trends

We fit log-linear curves to inference FLOPs as a function of model size (Figure 14). For General-OE, Math-MC, and Math-OE, scaling exponents are closely aligned between IFT and reasoning, consistent with reasoning answers shortening slightly faster as model size increases. For General-MC, reasoning models display larger scaling coefficients, reflecting higher cost for marginal gains.

J. Task-Level Results

Figures 15 to 21 show task-level breakdowns of the aggregated results presented in the main body.

Question	Answer
Lotto 6/49 is one of three national lottery games in Canada. Launched on June 12, 1982, Lotto 6/49 was the first nationwide Canadian lottery game to allow players to choose their own numbers. Previous national games, such as the Olympic Lottery, Loto Canada and Superloto used pre-printed numbers on tickets. Lotto 6/49 led to the gradual phase-out of that type of lottery game in Canada. Based on the paragraph above can we conclude that Canada's Lottery 6/49 was very popular in the 70s?	**IFT:** No, we cannot conclude that "Canada's Lottery 6/49 was very popular in the 70s" based on the paragraph. The text states that Lotto 6/49 was launched on June 12, 1982, which is after the 1970s, so the statement is incorrect. --- **Reasoning:** `<think>` Okay, let's see. The user is asking if we can conclude that Canada's Lotto 6/49 was very popular in the 70s based on the given paragraph. First, I need to check the dates mentioned. ... The answer should state that the conclusion is incorrect because the game started in 1982, which is after the 70s.`</think>` No, we cannot conclude that "Canada's Lottery 6/49 was very popular in the 70s" based on the paragraph. The text explicitly states that Lotto 6/49 was **launched on June 12, 1982**, which is in the early 1980s, not the 1970s. The paragraph also notes that earlier national lottery games (e.g., Olympic Lottery, Loto Canada, Superloto) used pre-printed numbers and were phased out after Lotto 6/49's introduction. There is no information provided about the popularity of any lottery games in the 1970s, nor was Lotto 6/49 even in existence during that decade. The conclusion is therefore unsupported.
I know that the answer to the question What are isolators found in microwave antennas made of? is in A necessary condition for the aforementioned reciprocity property is that the materials in the antenna and transmission medium are linear and reciprocal. Most materials used in antennas meet these conditions, but some microwave antennas use high-tech components such as isolators and circulators, made of nonreciprocal materials such as ferrite. These can be used to give the antenna a different behavior on receiving than it has on transmitting, which can be useful in applications like radar. Can you tell me what it is?	**IFT:** The isolators found in microwave antennas are made of **nonreciprocal materials such as ferrite**. --- **Reasoning:** `<think>` Okay, let's see. The user is asking about what isolators found in microwave antennas are made of. They provided a passage that should contain the answer. First, I need to read through the passage carefully. The key part here is where it mentions isolators and circulators. The sentence says, "made of nonreciprocal materials such as ferrite." So the answer should be ferrite. Wait, but let me make sure. The passage starts by talking about reciprocity in antennas, but then introduces isolators and circulators as exceptions. It says these components are made of nonreciprocal materials. The example given is ferrite. So yes, the answer is definitely ferrite. I should check if there are any other materials mentioned, but the text only lists ferrite as an example. The user might be looking for that specific term. No other materials are named here. So the answer should be ferrite.`</think>` Answer: The isolators found in microwave antennas are made of **ferrite**, a nonreciprocal material.

Table 3 | Examples of generated data from `Infinity-Instruct`, showing pairs of IFT and reasoning answers. Reasoning traces are shortened with ... due to their length.

Question	Answer
Solve the following math problem. Make sure to put the answer (and only the answer) inside `\boxed{}`. Evaluate the limit $$\lim_{n\to\infty}\left(\frac{n-10}{n+1}\right)^{3n+1}$$	**IFT:** We are asked to evaluate the limit: $\lim_{n\to\infty}\left(\frac{n-10}{n+1}\right)^{3n+1}$. Step 1: Simplify the base expression $\frac{n-10}{n+1}=\frac{n\left(1-\frac{10}{n}\right)}{n\left(1+\frac{1}{n}\right)}=\frac{1-\frac{10}{n}}{1+\frac{1}{n}}$. … $(3+\frac{1}{n})=-33$. Now exponentiate: $L=e^{-33}$. Final Answer: `\boxed{e^{-33}}` **Total characters:** 1,417 --- **Reasoning:** `<think>` Okay, so I need to evaluate the limit as $n\to\infty$ of $\left(\frac{n-10}{n+1}\right)^{3n+1}$. Hmm, let's see. I remember that limits involving expressions raised to variables can sometimes be solved using the exponential function and … Thus: $\ln L\approx\lim_{n\to\infty}(3n+1)\cdot\left(-\frac{11}{n+1}\right)=-11\cdot\lim_{n\to\infty}\frac{3n+1}{n+1}=-11\cdot 3=-33$. Exponentiating both sides: $L=e^{-33}$. Final Answer: `\boxed{e^{-33}}` **Total characters:** 11,225
Solve the following math problem. Make sure to put the answer (and only the answer) inside `\boxed{}`. Find the number of integer solutions to the equation $xyz=2003$.	**IFT:** We are asked to find the number of integer solutions to the equation $xyz=2003$. Step 1: Prime Factorization. First, factor 2003: $2003\div 17=117.823$ (not divisible by 17). … Try dividing by small primes up to $\sqrt{2003}\approx 44.76$. Eventually, we find: `\boxed{12}`. **Total characters:** 2,806 --- **Reasoning:** `<think>` Okay, so I need to find the number of integer solutions to the equation $xyz=2003$. Hmm, let me start by understanding what exactly the problem is asking for. We need all ordered triples of integers (x,y,z) such that when you multiply them together, you get 2003. … Thus, only the above 12 combinations are valid. Final Answer: `\boxed{12}`. **Total characters:** 26,747

Table 4 | Examples of generated data from `Llama-Nemotron-Post-Training-Dataset`, showing pairs of IFT and reasoning answers with character counts.

Benchmark	Instruction
gsm8k	Solve the following math problem. Make sure to put the answer (and only answer) inside \boxed{}.
math_500	Solve the following math problem. Make sure to put the answer (and only answer) inside \boxed{}.
aime	Solve the following math problem. Make sure to put the answer (and only answer) inside \boxed{}.
mmlu_math	Solve the following math problem. Make sure to put the answer (and only answer) inside \boxed{}.
mmlu_pro_math	Solve the following math problem. Make sure to put the answer (and only answer) inside \boxed{}.
aqua_rat	Solve the following math problem. Make sure to put the answer (and only answer) inside \boxed{}.
winogrande	Given a sentence with a blank (_) and two possible options, choose the option that correctly fills the blank so that the sentence makes the most logical sense. Make sure to put the answer (and only answer) inside \boxed{}.
openbookqa	Select the option that best completes the scenario based on everyday reasoning about cause and effect. Make sure to put the answer (and only answer) inside \boxed{}.
squad	Read the passage and answer the question by selecting the text span from the passage that best answers it. Make sure to put the answer (and only answer) inside \boxed{}.
mmlu_misc	Answer the following multiple-choice question by selecting the option that best fits the correct knowledge. Make sure to put the answer (and only answer) inside \boxed{}.
coqa	Read the passage and answer the question by selecting the text span from the passage that best answers it. Make sure to put the answer (and only answer) inside \boxed{}.
ifeval	Answer the following instruction.

Table 5 | Instruction prompts used for answer generation across evaluation benchmarks.

Benchmark	Instruction
Default	You will be given a Question, a User Answer (only its ending is shown due to length), and a Ground Truth. Your task is not to answer the question, but to say if the user answer is equivalent in meaning to the ground truth. First, extract the final result from both the User Answer and the Ground Truth Answer, based on the Question. Then, compare the two final results and determine whether they convey the same meaning. If they are equivalent, respond with \boxed{yes}. If they are not equivalent, or if the User Answer does not contain a valid answer, respond with \boxed{no}. Question: {question} User Answer: {answer} Ground Truth: {truth}
`ifeval`	You will be given an Instruction and a User Answer (only its ending is shown due to length). Your task is not to answer the Instruction, but to determine whether the User Answer follows all the formal requirements stated in the Instruction. If the User Answer contains a thinking process, you should ignore it and only focus on the final answer. First, identify every explicit requirement in the Instruction (e.g., no commas, maximum word count, required word occurrences, formatting rules). Then, compare the User Answer against these requirements. If all requirements are satisfied, respond with \boxed{yes}. If any requirement is violated, respond with \boxed{no}. Question: {question} User Answer: {answer}

Table 6 | Instruction prompts used for LLM-based answer assessment. Judge models follow a two-step protocol: first extracting the final answer while ignoring intermediate reasoning, then assessing semantic equivalence against the reference. We validated reliability through human verification and inter-judge agreement. Across all benchmarks, the median absolute error is 1.2, with pairwise comparisons confirming alignment: 0.9 (Nemotron vs. Llama), 1.0 (Nemotron vs. GPT-OSS), 1.8 (Llama vs. GPT-OSS).

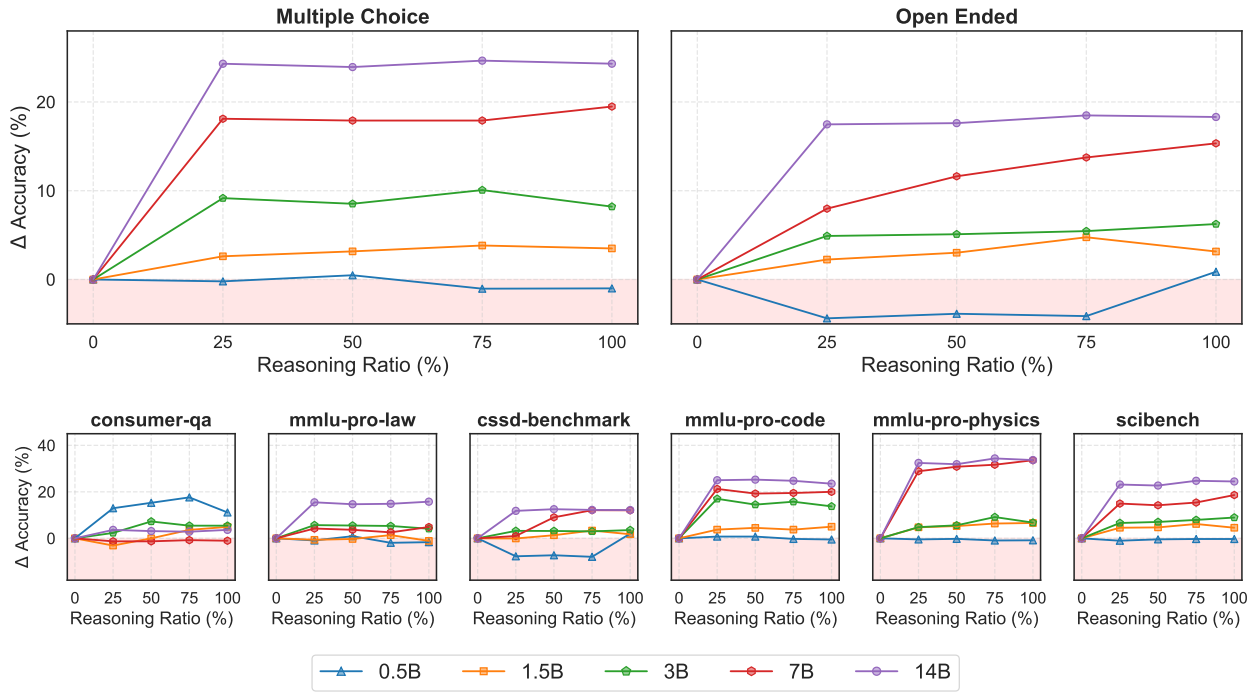


Figure 7 | Impact of reasoning ratio on law, code, and physics tasks in the sequential training scenario.

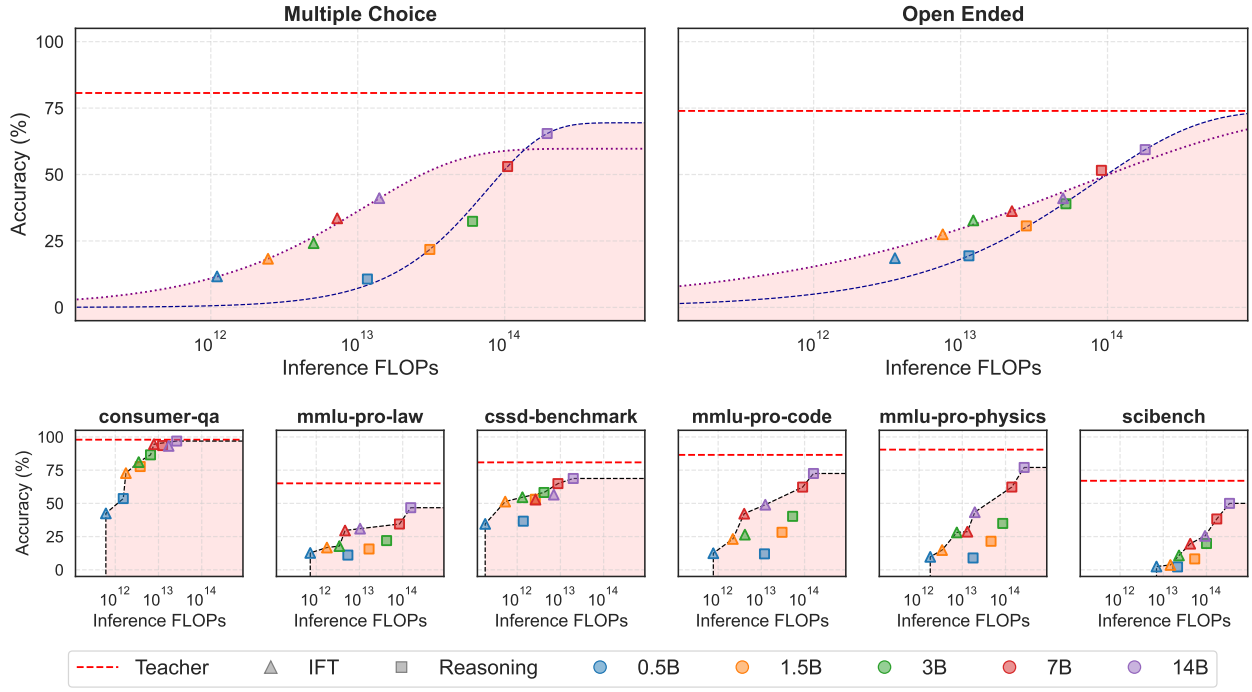


Figure 8 | Accuracy vs. inference FLOPs for law, code, and physics tasks.

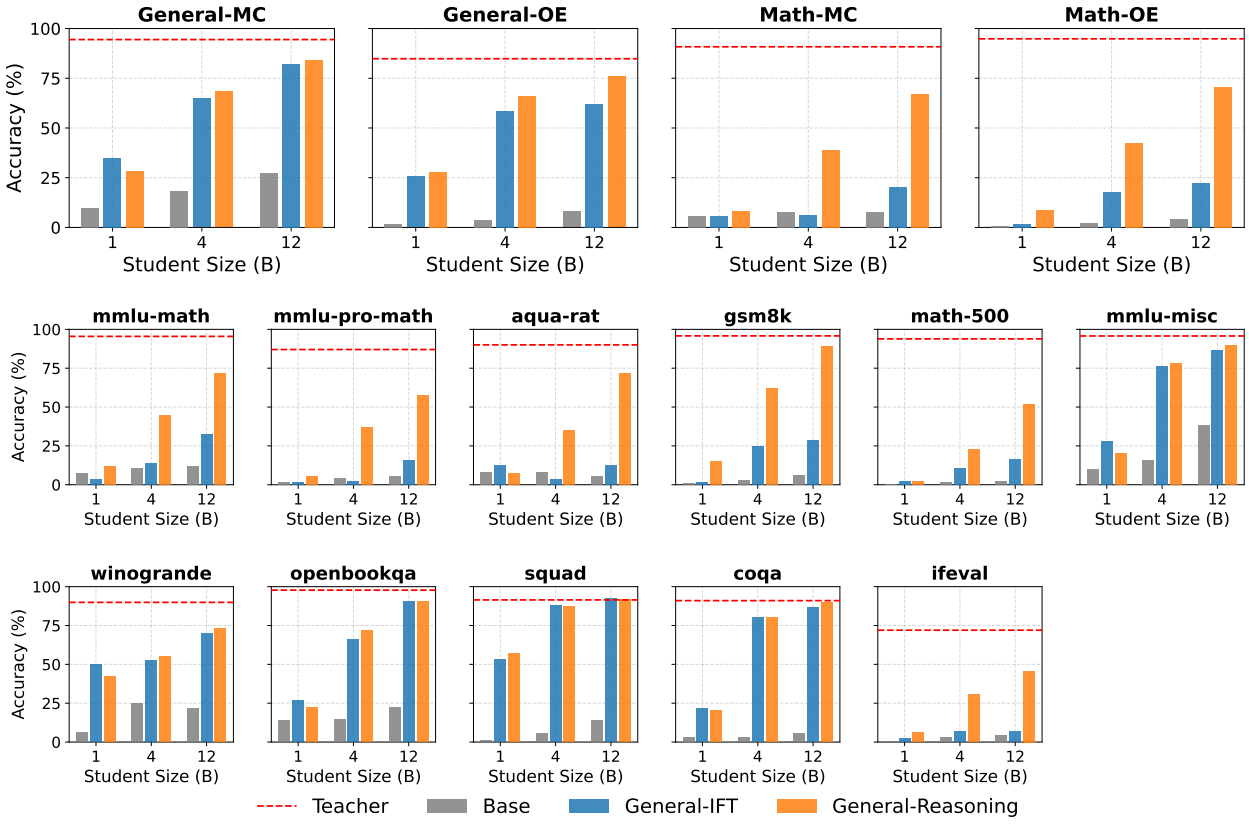


Figure 9 | Performance of Nemotron-Gemma teacher-student configuration.

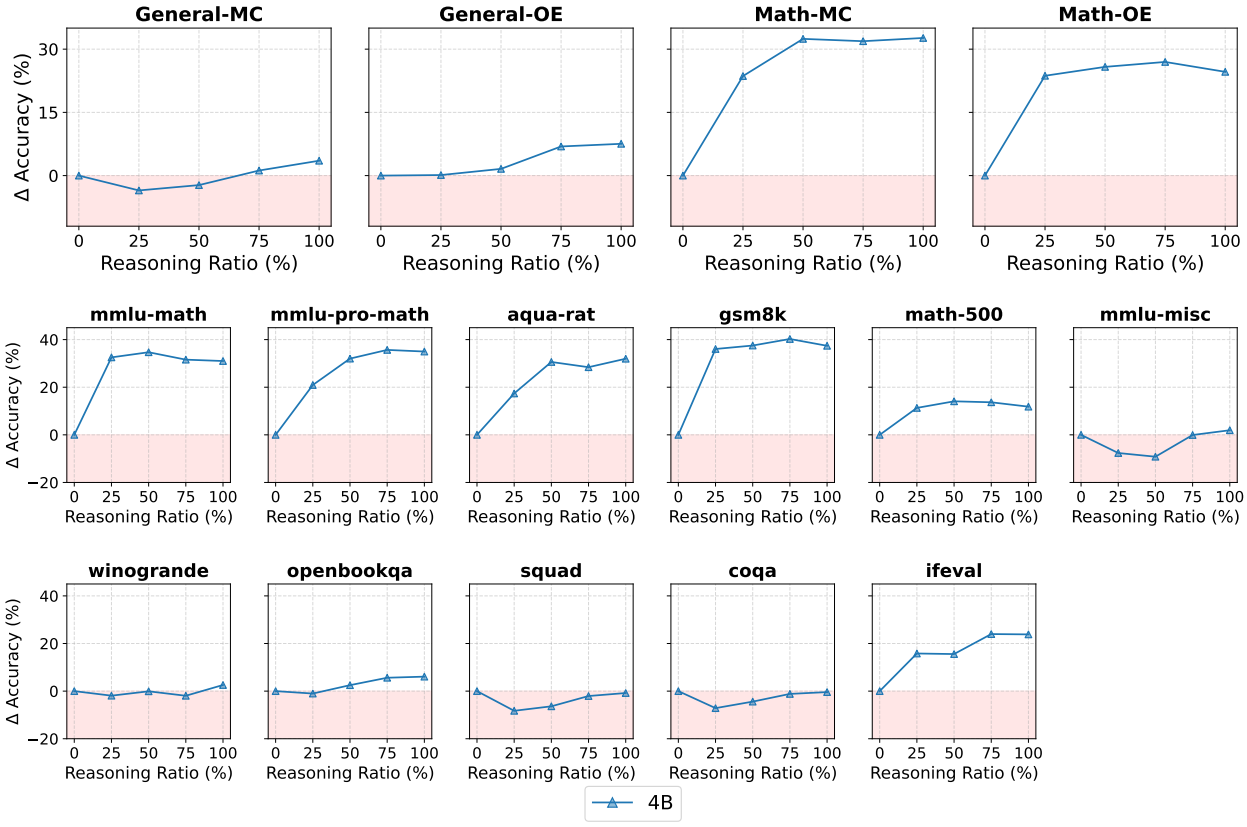


Figure 10 | Impact of reasoning ratio for the Nemotron-Gemma configuration at 4B.

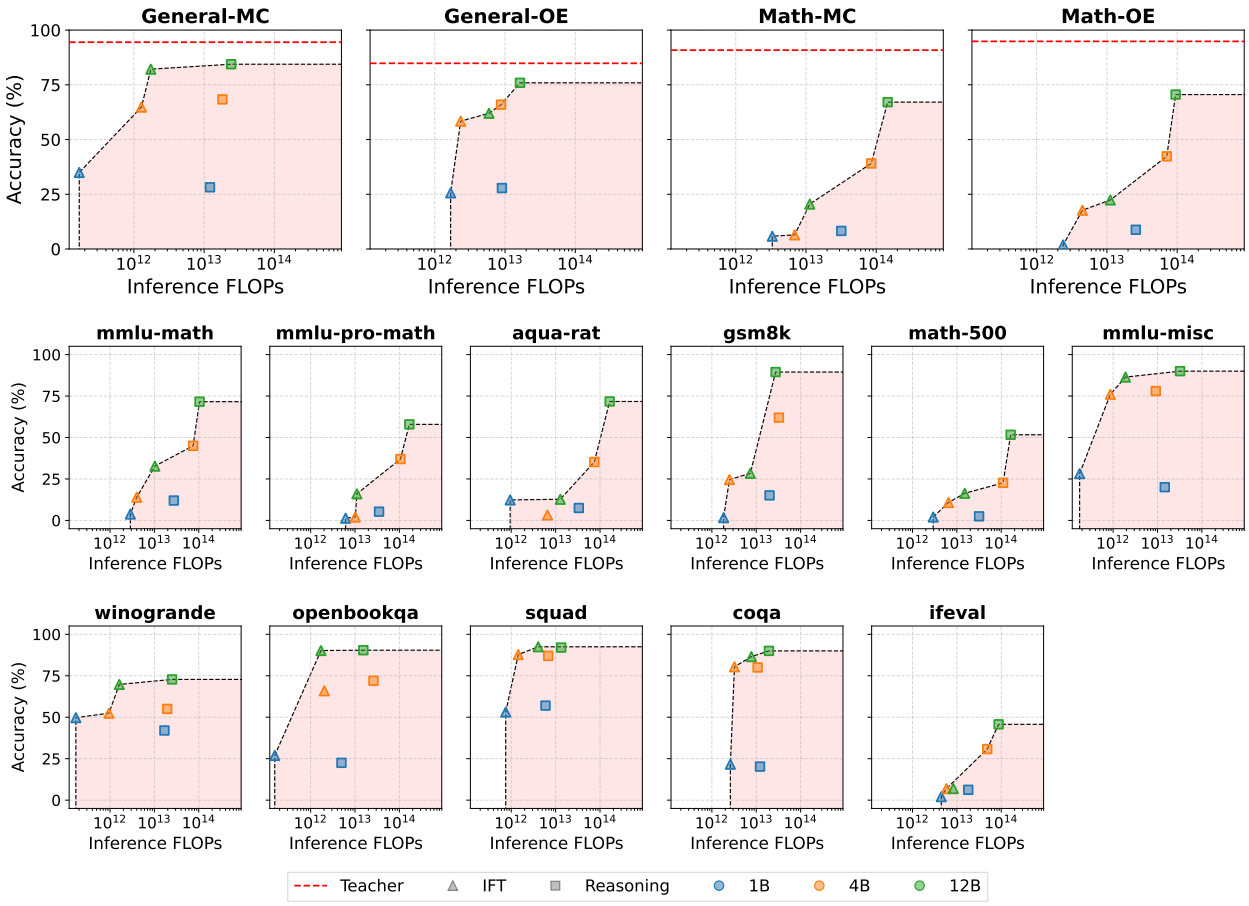


Figure 11 | Accuracy vs. inference FLOPs for the Nemotron-Gemma configuration.

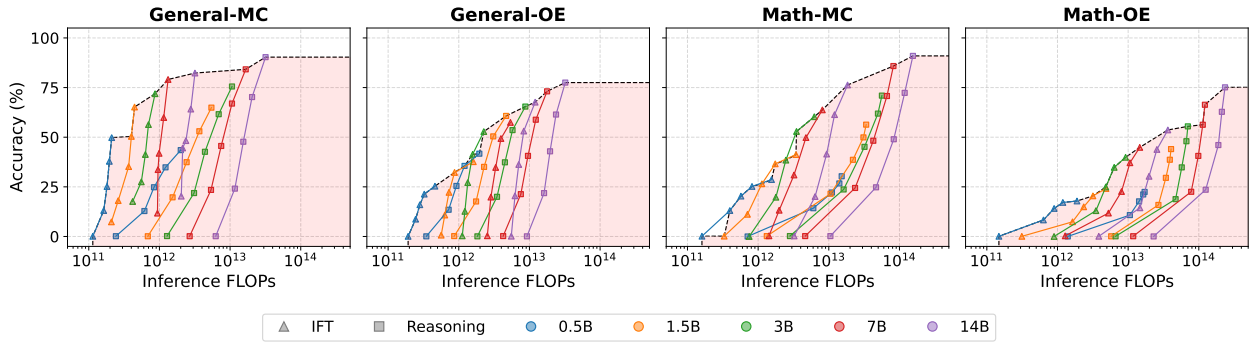


Figure 12 | Inference-cost impact of generation early-stopping. Each model is evaluated at five maximum-length thresholds (0th, 25th, 50th, 75th, 100th answer length percentiles).

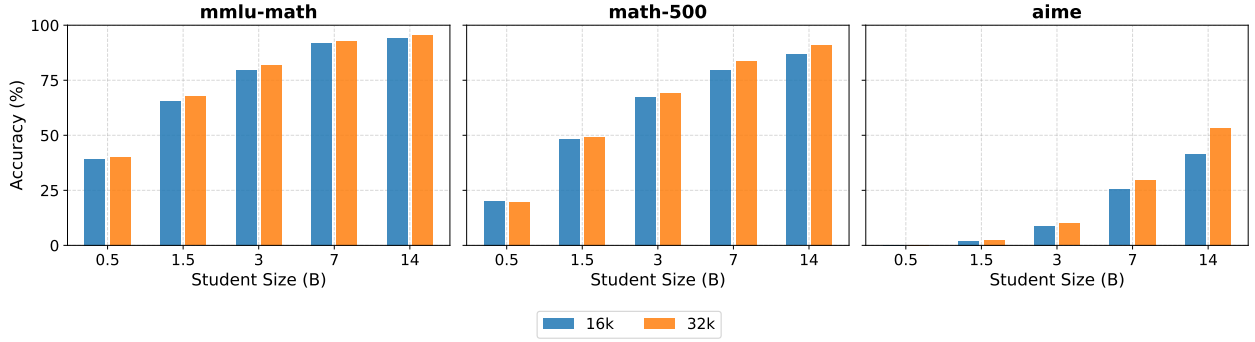


Figure 13 | Impact of increasing maximum generation length (16K to 32K tokens) on math tasks.

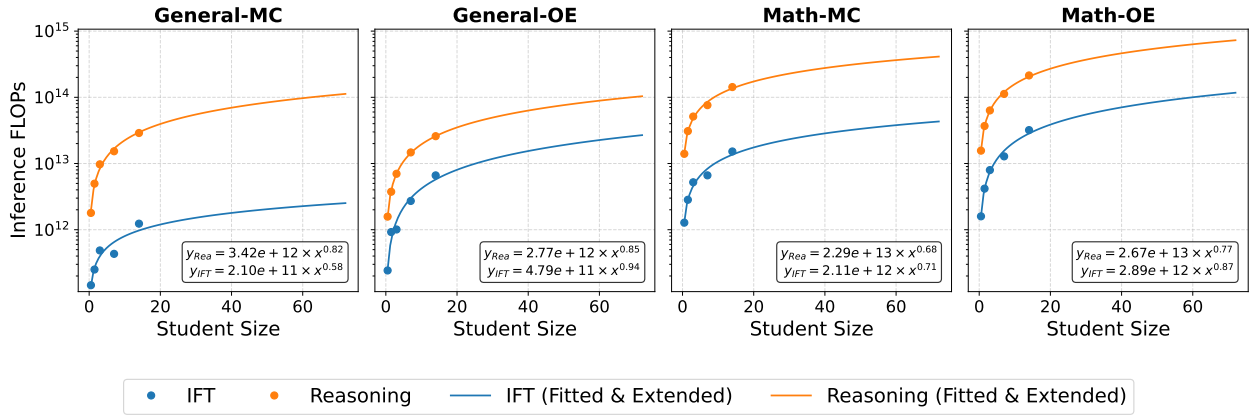


Figure 14 | Inference FLOPs vs. student model size. Power-law fits show scaling coefficients for IFT and reasoning.

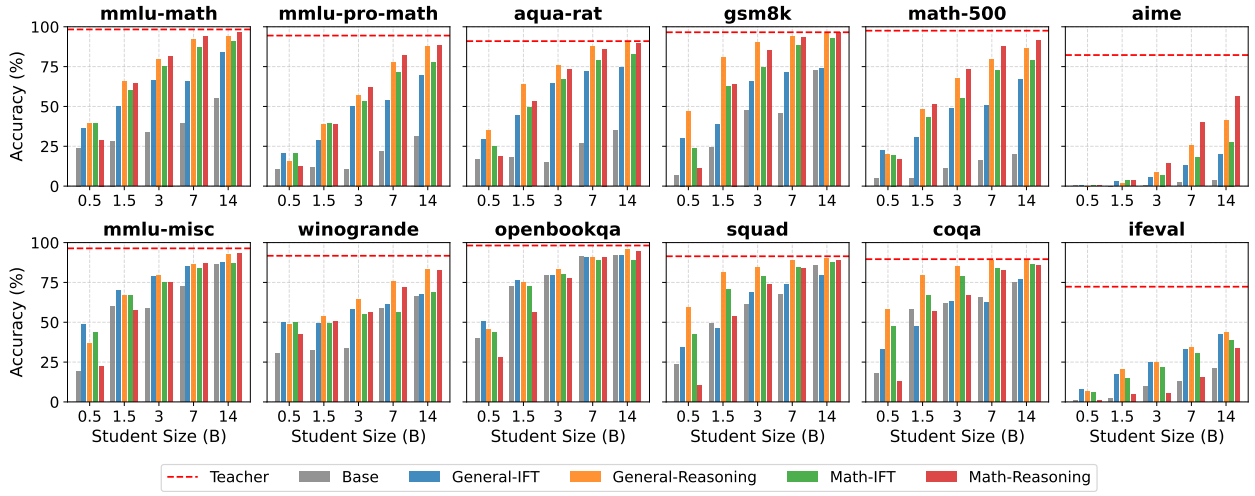


Figure 15 | Task-level downstream performance of mono-phasic models.

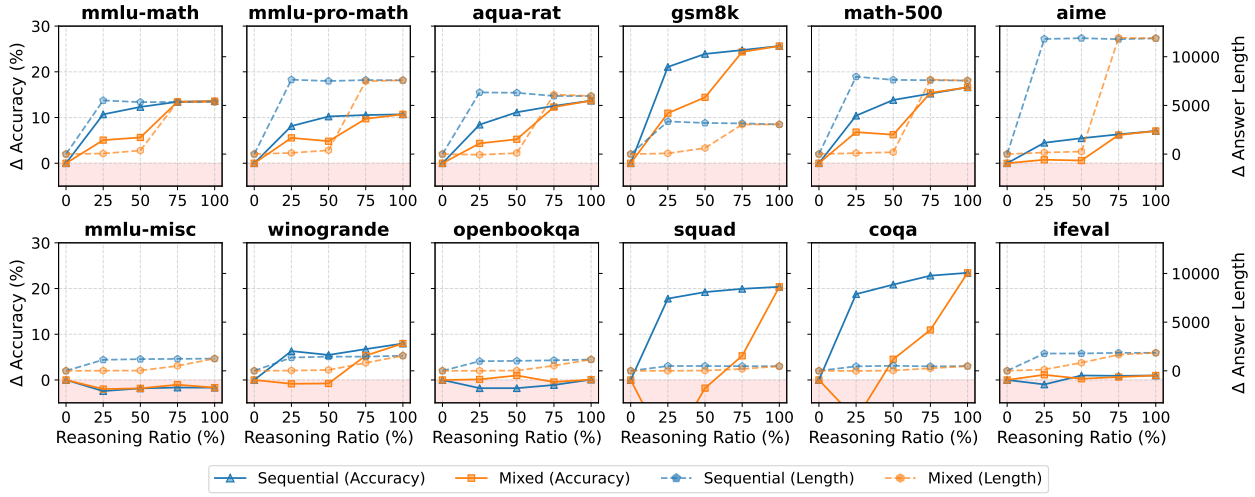


Figure 16 | Task-level comparison of sequential and mixed training.

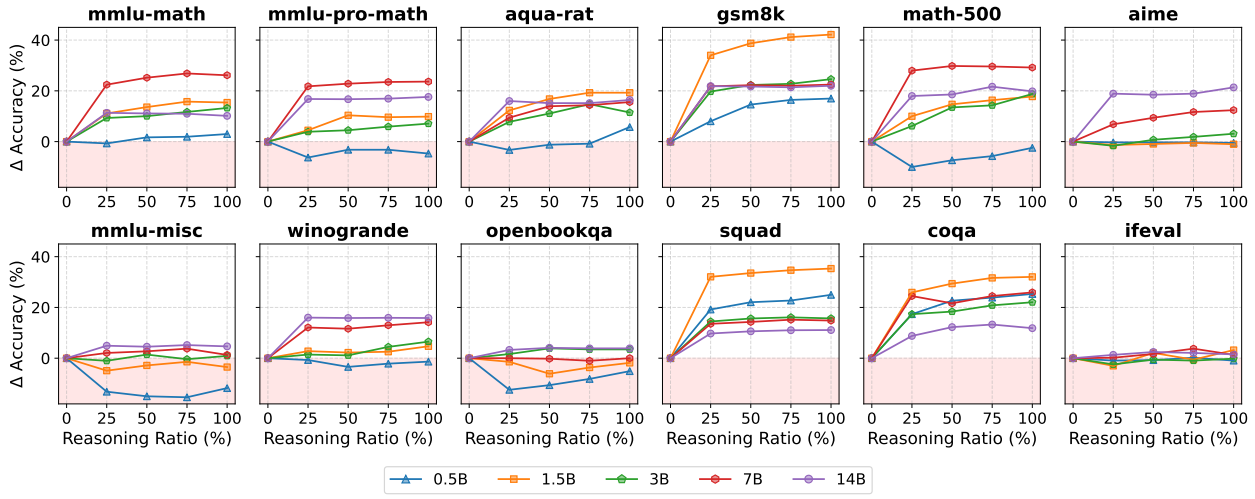


Figure 17 | Task-level impact of the reasoning ratio.

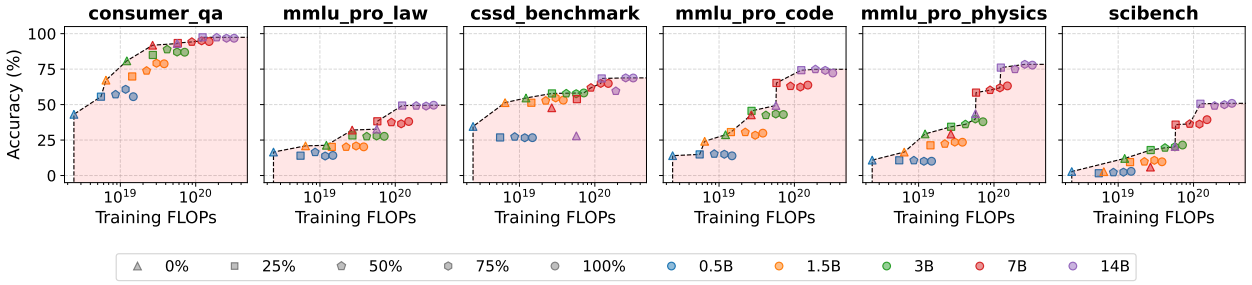


Figure 18 | Task-level accuracy vs. training FLOPs.

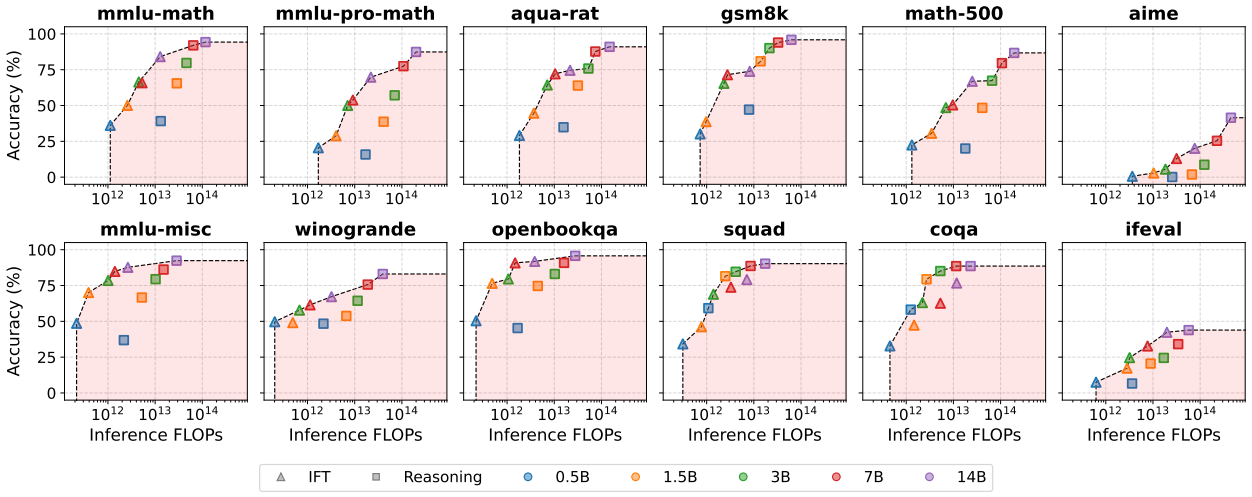


Figure 19 | Task-level accuracy vs. inference FLOPs.

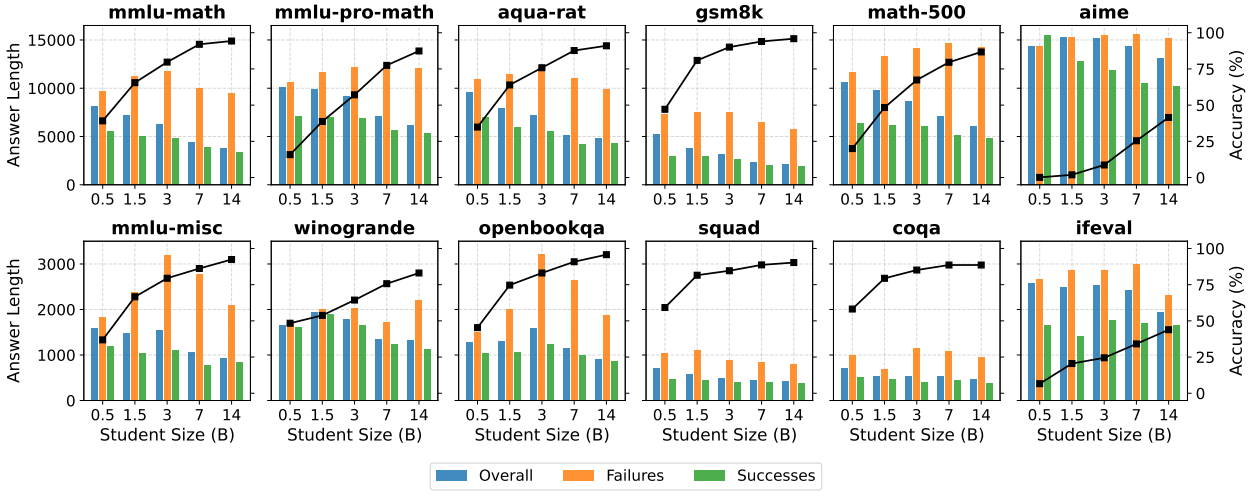


Figure 20 | Task-level answer length analysis.

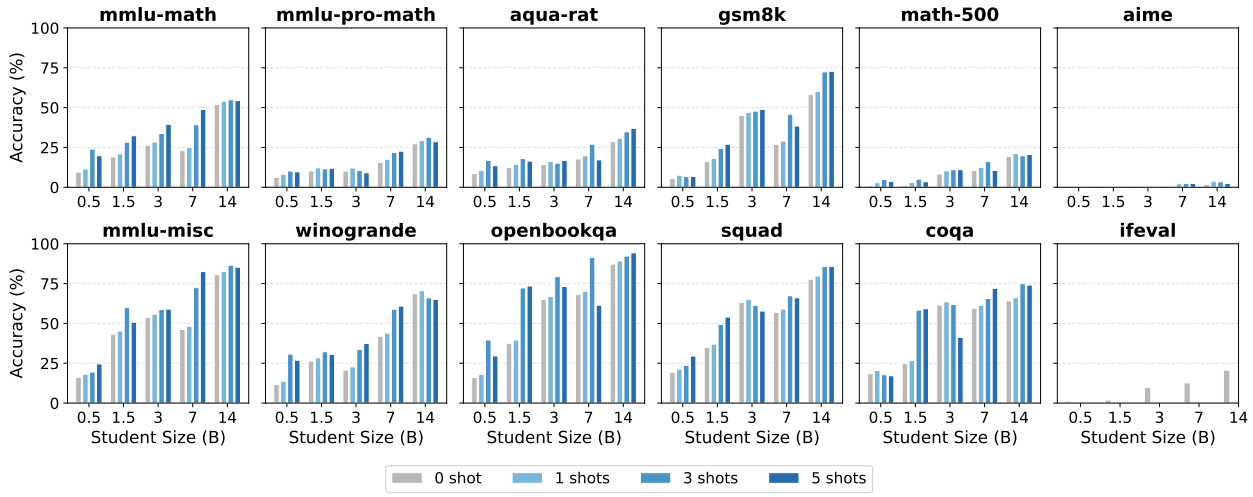


Figure 21 | Base student performance across 0-shot and few-shot settings.