OctoThinker: Mid-Training Incentivizes Reinforcement Learning Scaling

Anonymous Authors[1]

Abstract

Different base language model families—such as Llama and Qwen—exhibit divergent behaviors during post-training with reinforcement learning (RL), especially on reasoning-intensive tasks. What makes a base language model suitable for reinforcement learning? Gaining deeper insight into this question is essential for developing RL-scalable foundation models of the next generation. In this work, we investigate how mid-training strategies shape RL dynamics, focusing on two representative model families: Qwen and Llama. Our study reveals that (1) high-quality mathematical corpora, such as *MegaMath-Web-Pro*, significantly improve both base model and RL performance, while lower-quality alternatives (e.g., *FineMath-4plus*) fail to do so; (2) further adding QA-style data, particularly long chain-of-thought (CoT) reasoning examples, enhances RL outcomes, and instruction tuning further amplifies this effect; (3) while long-CoT improves reasoning depth, it can also induce verbosity of model responses and unstability of RL training, underscoring the importance of data formatting; (4) scaling mid-training consistently leads to stronger downstream RL performance. Building on these insights, we introduce a two-stage mid-training strategy—***Stable-then-Decay***—in which base models are first trained on 200B tokens with a constant learning rate, followed by 20B tokens across three CoT-focused branches with learning rate decay. This yields **OctoThinker**, a family of models demonstrating strong RL compatibility and closing the performance gap with more RL-friendly model families, i.e., Qwen. We hope our work can inform pre-training strategies for foundation models in the RL era, and we contribute open-source models and our corpora to support further research.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

Incentivizing large language models (LLMs) to think deeply through the chain of thought (CoT (Wei et al., 2022)) before giving the final answer with large-scale reinforcement learning (RL) is driving significant progress on the challenging reasoning tasks, i.e., solving competition-level mathematics problems, as demonstrated by OpenAI's o1 (OpenAI et al., 2024) and o3 (OpenAI, 2025). This also underscores a growing attention centered on RL as a means of boosting LLMs' reasoning performance. DeepSeek-R1-Zero (Guo et al., 2025) showcases a range of powerful and intriguing reasoning behaviors by directly applying large-scale RL to base language models, i.e., DeepSeek-V3-Base (Liu et al., 2024). In line with this trend, several methods such as SimpleRL (Zeng et al., 2025) and Open-Reasoner-Zero (Hu et al., 2025) have explored RL training on smaller base models—particularly the Qwen series (Yang et al., 2025)—achieving notable improvements in reasoning ability. However, despite these advances, replicating the success of R1-Zero-style training on other general-purpose base models, such as Llama series (Meta et al., 2024), has proven difficult, also evidenced by recent studies (Gandhi et al., 2025; Liu et al., 2025). This naturally raises a fundamental question: ***What underlying factors cause the base models to exhibit divergent behaviors during RL training?*** Understanding this could shed light on the scientific foundations that connect pre-training and the scalability of RL for reasoning, and may guide the design of future base models more amenable to reasoning-oriented RL.

In this work, we explore this question through the lens of mathematical reasoning and begin by observing a key difference in RL dynamics between two prominent model families: Qwen and Llama. Specifically, our preliminary studies reveal that Qwen models are much more amenable to RL scaling, while the Llama model tends to predict final answers prematurely and produce repetitive outputs during RL training. To better understand this discrepancy, we conducted a series of large-scale and controlled mid-training interventions on Llama models, followed by RL training. Our findings highlight that the quality of mathematical pre-training corpora is critical for successful RL performance. For instance, we found that *MegaMath-Web-Pro* (Zhou et al., 2025) offers significantly greater benefits for RL scaling than corpora like *FineMath-4plus* (Allal et al., 2025). On top

of a high-quality mathematical pre-training corpus, incorporating QA-style data yields further improvements, and introducing a small amount of instruction-following data helps enhance RL effectiveness even more. We also observed that injecting long chain-of-thought data during mid-training introduces instability into the RL phase. To address this, we refined the RL prompt and adopted a progressive maximum response length scheduler to stabilize training and ensure consistent behavior. To support large-scale mid-training, we also constructed a reasoning-intensive mathematical corpus exceeding 70 billion tokens, namely *MegaMath-Web-Pro-Max* , with data quality on par with *MegaMath-Web-Pro*. In extended mid-training experiments on this dataset—scaling up to 100 billion tokens—we observed that increasing the mid-training budget can lead to noticeable improvements in downstream RL performance. Interestingly, these gains are often not immediately reflected in the standard evaluations of the mid-trained base model, highlighting a gap between base model evaluation metrics and RL-stage capabilities.

Can we turn Llama into a foundation model well-suited for RL scaling by further scaling up its mid-training? Building on the insights above, we explored this question by adopting a two-stage (*stable-then-decay*) mid-training strategy. In the first stable stage, we train Llama models on a high-quality mixture of pre-training corpus for 200B tokens using a constant learning rate. In the second decay stage, we annealed the learning rate and introduced distinct data mixtures—short CoT, long CoT, and a hybrid of both—to mid-train three separate branches. These branches are later refined through RL training, equipping them with stronger reasoning capabilities. Inspired by the multi-armed nature of an octopus, we name this model family OctoThinker. Experiments across all model sizes and 13 mathematical reasoning benchmarks demonstrate the effectiveness of our approach: both stages of mid-training lead to substantial performance gains, especially the first stage, which consistently delivers 10–20% improvement. Building on these stronger base models, subsequent RL training further boosts performance, with each branch showing distinctive behavior patterns. Notably, our models post-RL achieve performance on par with Qwen2.5 of the same size, effectively narrowing the gap between Llama and other RL-friendly model families. These results confirm the power of scaled-up, reasoning-intensive mid-training in transforming Llama into a suitable base model for RL scaling. To foster open research, we will release our curated data, models, and training scripts. We hope OctoThinker offers a meaningful step toward the next generation of reasoning-capable AI systems.

2. Preliminaries

We begin by identifying a key difference in RL dynamics between two prominent model families—Qwen and Llama—through the lens of mathematical reasoning. This observation offers a concrete and measurable foundation that grounds our systematic investigation.

2.1. Experiment Setup

RL Setup We performed RL experiments based on the verl (Sheng et al., 2024) framework and utilized the GRPO (Shao et al., 2024) algorithm. For RL training prompts, we adopted the MATH8K dataset due to its moderate difficulty and concise composition. The maximum response length is set to 4,096 tokens. See § A.1 for more details. Unless otherwise specified, we employed a simple prompt template of "`Question:{}\nAnswer:{}`" to format training examples. We employed Llama-3.2-3B-Base and Qwen2.5-3B-Base to perform R1-Zero styled RL training given the moderate model size. We adopted the few-shot prompting evaluation for base language models and employed zero-shot evaluation for RL-tuned models.

2.2. Observations

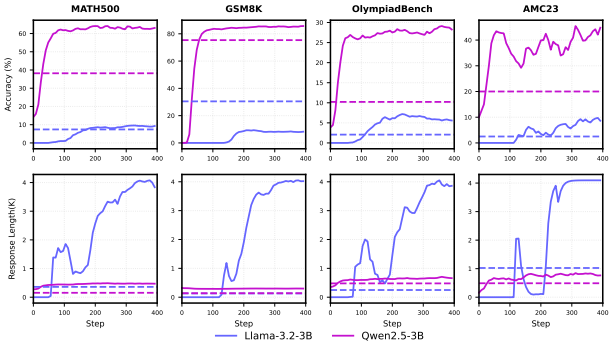

Figure 1: Training dynamics comparison (downstream performance and the average length of correct responses) between Llama-3.2-3B-Base and Qwen2.5-3B-Base. The dashed line indicates the few-shot evaluation performance and average length of correct responses of the corresponding base models. Ditto for subsequent figures.

During RL training on Llama-3.2-3B-Base and Qwen-2.5-3B-Base, we observed notably different and intriguing training dynamics regardless of their performance (see Figure 1). Specifically, the length of correct responses from Qwen increases steadily and reasonably throughout training, whereas Llama exhibits abnormal behavior—its average response length escalated dramatically, reaching up to 4K. Upon closer inspection of Llama's output, we found that it typically begins with "`\boxed:{}`", followed by extremely obvious repetition until hitting the max response length, in stark contrast to Qwen's coherent and natural reasoning output. The evaluation results further highlights the divergence: Qwen achieved substantial improvements over its base model across a wide spectrum of benchmarks, from simple to complex math reasoning tasks. Meanwhile, Llama

experienced only marginal gains—or even regressions, as seen on GSM8K—likely due to the distributional gap between the RL training set (e.g., MATH8K) and GSM8K. The above observations motivate us to attribute the reason to their potential divergence of pre-training despite their opaque details.

These observations also further prompt a more fundamental question: *Can we intervene during the pre-training phase of Llama (e.g., via mid-training) to make it more amenable to RL scaling?* Specifically, in this work, we would like to explore a range of mid-training intervention strategies—methods that adjust the pre-training trajectory of LLMs—to examine their downstream impact on RL dynamics.

2.3. What is Mid-training?

Mid-training is a mid-stage whose computational and data (token) requirements are intermediate between pre-training and post-training. It aims to achieve specific objectives—such as domain and language expansion (Dou et al., 2025), long-context extension (Abdin et al., 2024b;a), improving data quality (Hu et al., 2024a; OLMo et al., 2025), leveraging large-scale synthetic data (Yang et al., 2024b;a; 2025), and preparing for post-training, among others—by significantly altering data quality and distribution (Dubey et al., 2024; Wake et al., 2024) (and/or modifying model architecture to improve inference efficiency (Bercovich et al., 2024; 2025)).[1]

3. Digging Deeper: Exploring Key Factors through Controllable Mid-training

We investigated the impact of several factors during mid-training on RL performance through head-to-head experiments. Specifically, we examine the effects of data quality of math web corpora, the inclusion or exclusion of QA-format data, the nature of the QA data itself, the presence of general instruction-following data in mid-training, as well as the pre-training token budget. These systematic analyses help us gain a deeper understanding of the connection between pre-training and RL dynamics and figure out suitable recipes for scaled-up mid-training.

3.1. Experimental Setup

Mid-training setup By default, we perform mid-training with Llama-3.2-3B-Base on diverse datasets and training configurations within a 20B-token training budget. We use a cosine learning rate scheduler without warmup, with a peak learning rate of 3e-5 and a minimum learning rate set to one-tenth of the peak. The default sequence length is 8192, and the batch size is 4 million tokens. Training is conducted using the Huggingface's Nanotron framework.

RL setup We follow the exact same RL setup as described above in Section 2, unless stated otherwise.

Table 1: Statistics and types of different datasets we used

Dataset	Type	# Tokens (B)
FineMath-4plus (Allal et al., 2025)	Math Web Documents	9.57
MegaMath-Web-Pro (Zhou et al., 2025)		13.00
MegaMath-Web-Pro-Max (Ours)		73.80
MegaMath-QA (Zhou et al., 2025)	QA (Short-CoT)	5.94
OpenR1-Math-220K (HuggingFace, 2025)	QA (Long-CoT)	1.05
TULU3-sft° (Lambert et al., 2024a)	General Instruction Following	0.01
WildChat (Zhao et al., 2024)		0.29
UltraChat-220K (Ding et al., 2023a)		0.51

Datasets The datasets used to support our controllable experiments are summarized in Table 1. For the OpenR1 dataset, we concatenated the question and the think process enclosed within `<think>` and `</think>` using a line break. For the general instruction following datasets, we only retained high-quality conversations, such as those derived from GPT-4, and formated the conversations as "`User:{}\nAssistant:{}`". We curated MegaMath-Web-Pro-Max to support large-scale ablation studies and mid-training. The corpus was constructed by leveraging an efficient classifier to recall documents from MegaMath-Web (Zhou et al., 2025), followed by refinement using a powerful instruction-following LLM. See § A.2 for details.

3.2. On the Inclusion and Data Quality of Math Web Corpora

Web corpora provide a solid foundation during pre-training. We believe that math web corpora, along with their data quality, continue to play a crucial role during mid-training. We began our systematic analysis by performing mid-training on different math web corpora and holding other factors being constant. As shown in the Figure 2, mid-training on math web data improves performance over the base model, with MegaMath-Web-Pro and MegaMath-Web-Pro-Max showing slightly better gains than Finemath-4plus. After RL training, we found that mid-training on math web corpora improves RL performance to varying degrees. MegaMath-Web-Pro and MegaMath-Web-Pro-Max bring significant gains for Llama in RL training, while Finemath-4plus yields only marginal improvements—highlighting the clear differences in data quality. Furthermore, we observed that models trained on FineMath-4plus exhibited abnormal behavior, with response lengths rapidly increasing until reaching the maximum limit of 4,096 tokens. The outputs typically began with "`\boxed{}`" and devolved into repetitive "So-

[1] In the absence of a precise or widely agreed-upon definition, here, we aim to introduce a concise and rigorous definition of *mid-training* within this context. The term was reportedly first mentioned in an OpenAI job description in mid-2024. A detailed blog for this term is available at https://vintagedata.org/blog/posts/what-is-mid-training.

lution" statements. Given these observations, we selected MegaMath-Web-Pro as our default mathematical corpus and also MegaMath-Web-Pro-Max for scaled mid-training.

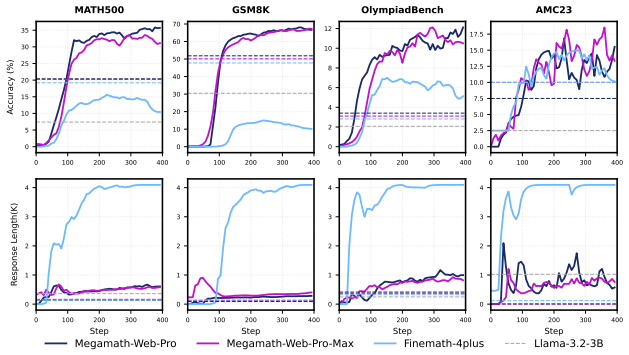

Figure 2: The effect of different math web corpora during mid-training (20B tokens training budget each).

3.3. On the Inclusion and Nature of QA-Format Data

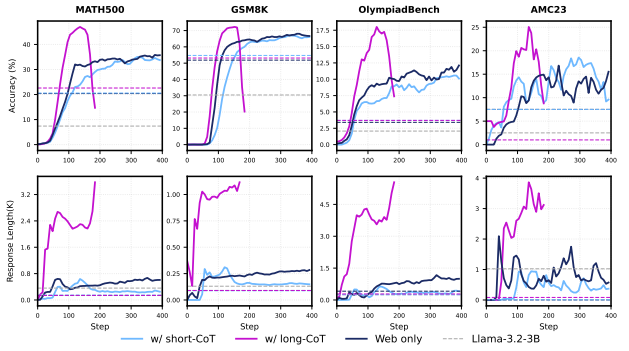

Figure 3: Impact of incorporating CoT data with varying characteristics during mid-training (9:1 mixture ratio).

Intuitively, introducing QA data into pre-training and mid-training improves model performance, as previously exemplified in Bi et al. (2024) and Hu et al. (2024b). We further investigated this using a 9:1 web-to-QA data mix. We hypothesize that QA data's short Chain-of-Thought (short-CoT, from MegaMath-QA) and long-CoT (from OpenR1-Math-220K) reasoning, which may include self-reflection and backtracking, enhance base model performance and RL training. Maximum response lengths were 8,192 tokens for long-CoT models and 4,096 for others.

As shown in Figure 3, incorporating QA data into mid-training generally yields performance gains for the base model, though these gains are marginal, as indicated by dashed lines. After RL training, incorporating short-CoT data into mid-training shows no improvements compared to mid-training on web data alone, possibly due to the data distribution gap, while long-CoT data brings significant performance gains. However, incorporating long-CoT data introduces challenges with unstable RL training, evidenced by sudden performance drops and sharp increases in response length. We also explored methods for stabilizing RL training, which we discuss in the following sections.

3.4. On the Inclusion of Instruction-following Data

Incorporating instruction-following data into earlier-stage training has become an increasingly common practice. Works such as MiniCPM (Hu et al., 2024b) demonstrate that including high-quality unlabeled data and instruction-following data significantly improves downstream performance. We believe this inclusion is critically important for enhancing the base model's ability to follow instructions, which may be a potential key determining factor for successful RL training. We incorporated instruction-following data alongside web data and QA data in a 1:89:10 ratio. For this, we combined these high-quality datasets with appropriate filtering and formatting: TULU3-sft-personas-instruction-following (Lambert et al., 2024b), WildChat (Zhao et al., 2024), and UltraChat-200K (Ding et al., 2023b), totaling approximately 0.8B tokens.

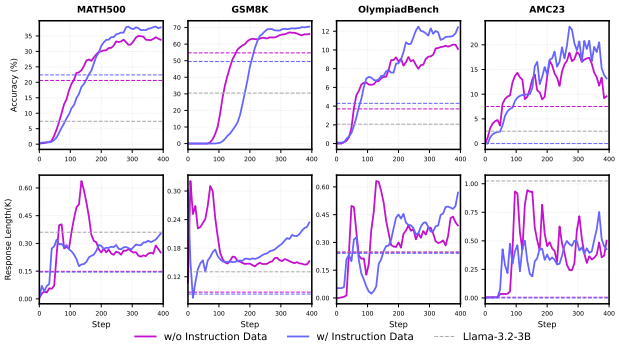

Figure 4: Impact of incorporating instruction-following data during mid-training with a mixture of web, short-CoT and instruction data in a ratio of 89 : 10 : 1. The max. response length is 4,096.

Incorporating instruction-following data into the short-CoT mid-training mixture. As shown in Figure 4, after RL training, incorporating instruction-following data, unlocks the potential of short-CoT data, showing performance advantages over the exclusion case after 200 steps. Additionally, this inclusion helps stabilize response length, resulting in smoother increases compared to when instruction-following data is excluded.

Incorporating instruction-following data into the long-CoT mid-training mixture. Similar to the challenges encountered earlier in RL training with the long-CoT mid-trained base model, as shown in Figure 5, incorporating instruction-following data showed performance improvements after 150 steps. However, this addition failed to prevent the overall decline in RL performance and the rapid increase in response length. Note that we set the maximum

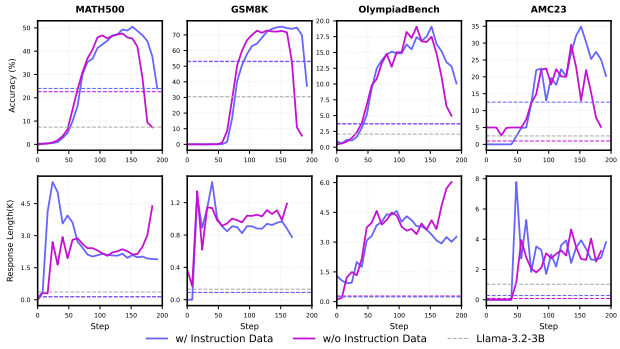

Figure 5: Impact of incorporating instruction-following data during mid-training with a mixture of web, long-CoT and instruction data in a ratio of 89:10:1. The maximum response length is 8,192.

response length to 8,192 tokens for these experiments.

Given the challenges encountered during RL training on the base model mid-trained on long-CoT data, we explored strategies to stabilize RL training by modifying the RL prompt template and maximum length scheduler.

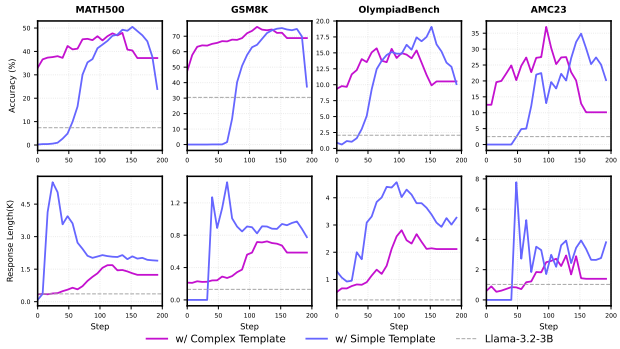

Figure 6: Impact of different RL prompt templates.

Effect of RL prompt template The default template is "`Question:{}\nAnswer:{}`", which we refer to as "*Simple Template*". Here, we introduce an alternative, the "*Complex Template*", adapted from the prompt design in Open-Reasoner-Zero (Hu et al., 2025) (see Figure 14).

We also controlled the maximum response length as 8,192 tokens. As shown in Figure 6, we found this *complex template* could clearly stabilize RL training compared to the *simple template*, as evidenced by a smoother, more gradual increase in mean response length, as opposed to the sharp spikes observed with the simple template. Despite this stabilization, performance across evaluation benchmarks still deteriorates during the later stages of RL training, indicating need more exploring. Note that we adopt the *complex template* as the default for all subsequent RL experiments.

Effect of the maximum response length The default maximum context length is set to 8,192 tokens for long-CoT mid-trained models. Intuitively, we can delay the sharp rise

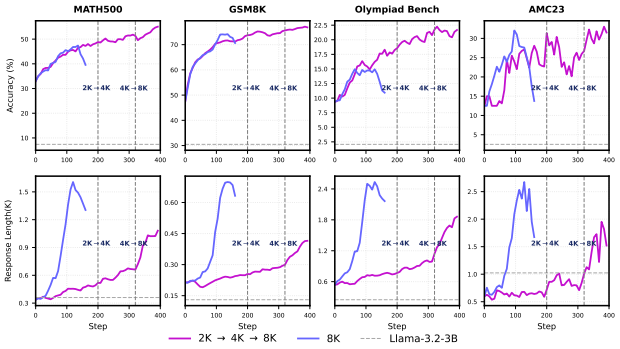

Figure 7: Impact of the progressive max. length scheduler.

in response length by gradually increasing the maximum response length in multiple stages. Specifically, we started with a limit of 2,048 tokens for the first 200 steps, increased it to 4,096 tokens from step 200 to step 320, and then further expanded to the full 8,192-token context length from step 320 to step 400. As shown in Figure 7, this progressive scheduling strategy significantly stabilizes RL training up to 400 steps, while consistently improving performance across benchmarks. In addition, the response lengths grow steadily and appropriately, highlighting the effectiveness of the progressive length scheduler.

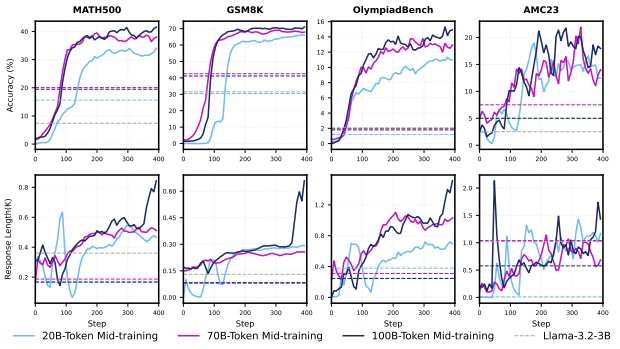

Figure 8: Impact of scaling up the mid-training budget.

3.5. On the Issue of Mid-training Budget

Could further scaling up mid-training improve RL performance? To explore this, we conducted a 100B-token mid-training run on MegaMath-Web-Pro-Max using a default cosine learning rate scheduler. We selected three intermediate checkpoints—trained on 20B, 70B, and 100B tokens, respectively—and performed RL training. When evaluating the base models, we observed that the 70B and 100B checkpoints achieved comparable performance, both significantly outperforming the 20B model. After RL training, interestingly, we found that increasing the mid-training token count consistently leads to improvements on RL performance despite varying degrees, whether moving from 20B to 70B or from 70B to 100B tokens. These findings highlight the importance of further scaling up the mid-training budget to

unlock additional gains in downstream RL performance.

4. OctoThinker-Base: Branching Reasoning Foundations via 2-Stage Mid-training

Building upon the insights above, a natural question arises: *Can we turn Llama into a foundation model well-suited for RL scaling by scaled-up mid-training?* We ultimately adopt a two-stage (*stable-then-decay*) mid-training strategy to achieve both: (1) steady improvements in mathematical reasoning ability in the first stage; (2) diversified model behaviors through branching in the second (decay) stage. Multi-stage pre-training has been validated as effective in prior work (Hu et al., 2024b; OLMo et al., 2025). The *stable-then-decay* setup offers flexibility: the decay phase can begin at any point, enabling checkpoint selection independent of a fixed schedule. This also supports fair comparisons across different mid-training configurations. Importantly, decaying the learning rate in the second stage amplifies the effect of injected data, helping shape model behaviors more efficiently. Since the decay stage used for shifting model behaviors (in other words data distribution) is typically shorter, this approach also reduces the overall training cost in general.

What does "OctoThinker" mean? "*Octo*" is derived from "octopus," symbolizing our base model family, which branches into variants trained with different strategies. "*Thinker*" reflects the model's final stage—reinforcement learning—where it is trained to think and reason, exhibiting frequent self-reflection and strong reasoning capabilities.

4.1. Recipe for the First Stage: Building Strong Reasoning Foundations

Although the previous analysis has revealed several factors that are critical to building strong reasoning models, our mid-training resource table (see Table 1) clearly shows that truly high-quality tokens are still scarce at this moment. Therefore, in the first phase, we adopted a relatively conservative strategy—primarily relying on high-quality web corpora such as MegaMath-Web-Pro-Max and DCLM-Baselines (Li et al., 2024), supplemented with a small portion of synthetic data—to enable the model to improve steadily at scale. Following the training settings used in MegaMath-Llama (Zhou et al., 2025), we reduced the proportion of synthetic data and adopted a WSD-style (Hu et al., 2024b) learning rate scheduler, replacing the cosine learning rate with a constant learning rate and training for 200B tokens. We provide specific training configurations, i.e., data mixture and training hyper-parameters of the first-stage in Table 5 and Table 6. We refer to the resulting mid-training models as `OctoThinker-Base-Stable`.

4.2. Branching at the Second Stage: Seeking Perfect Blend for RL Scaling

4.2.1. Pilot Studies

Building on prior experiments, we identified dataset quality and quantity as key drivers of effective mid-training and strong base model development. Before entering the decay stage, we conducted a series of controlled 10B-token mid-training experiments on the OctoThinker-3B-Base-Stable model—each followed by RL training—to investigate how different QA datasets affect downstream performance.

Data Composition and Its Impact on RL We experimented with three QA datasets—MegaMath-QA, OpenR1-Math-220K, and OpenMathInstruct-2 (OMI2)—in varying proportions (10%, 20%, 30%, and 40%) while holding constant 5% DCLM-Baselines data, 10% instruction data, and the remainder from MegaMath-Web-Pro. Ablation studies (see Figure 15) revealed that the origin of QA data plays a critical role. Specifically, OpenR1-Math-220K and OMI2 are derived from structured downstream datasets (e.g., GSM8K, MATH), while MegaMath-QA is sourced from less curated web documents. These differences in data source and distribution substantially impact downstream RL performance, highlighting the importance of distributional alignment between mid-training data and downstream tasks. In light of this, we adopt OpenMathInstruct-2, OpenR1-Math-220K (and further adopt the a-m-team's distilled dataset[2]), and NuminaMath-1.5[3] as our primary QA datasets for the decay stage, due to their closer resemblance to competition-style, reasoning-intensive benchmarks.

Identifying the Optimal QA Ratio Across our ablation studies (also see Figure 15), we observed a consistent trend: increasing the QA data ratio leads to improved RL performance, which aligns with expectations due to the format similarity with RL objectives. However, gains began to plateau beyond a 30% QA mix, with 40% showing diminishing returns across most benchmarks. We attribute this to token redundancy and lack of diversity at higher QA proportions. As a result, we adopted 30% QA as the optimal ratio, balancing performance and data efficiency.

4.2.2. Final Decay Recipe

For the decay stage, we explored two learning rate (LR) scheduler variants: (1) constant LR decay, where the LR remains fixed at 10% of the final LR used in the stable stage; (2) Cosine decay to 10%, where the LR gradually decays to 10% of the stable-stage final LR. Based on mid-training evaluation results, the cosine decay strategy

[2] https://huggingface.co/datasets/a-m-team/AM-DeepSeek-Distilled-40M

[3] https://huggingface.co/datasets/AI-MO/NuminaMath-1.5

demonstrated more consistent performance. We therefore adopted it as the default scheduler for the decay stage, with hyperparameters detailed in Table 8. During the decay stage, we branched the mid-training into three distinct variants based on data composition: OctoThinker-Long (long-reasoning data), OctoThinker-Short (short-reasoning data), OctoThinker-Hybrid (a mix of both) with decayed learning rate. The corresponding data mixtures are shown in Table 7.

4.3. Evaluation on OctoThinker-Base Series

We evaluated the performance of each branch on 14 mathematical benchmarks, alongside the original Llama base model and the model after stable-stage mid-training. As shown in Table 2,3,4, across all sizes, each OctoThinker branch demonstrates a noticeable 10%-20% improvement over the original base model and consistent gains over the stable-stage model. Notably, random and poor performance on challenging competition benchmarks highlights the necessity of post-training. Overall, these results reinforce our view that OctoThinker-Base series provide a strong foundation for studying RL scaling with solid reasoning capabilities.

Table 2: Evaluation results on OctoThinker-1B series.

	Benchmarks	Llama-3.2-1B	OctoThinker-1B-Base			
			Stable	Long	Hybrid	Short
Core	GSM8K (8-shot)	7.66	30.93	37.15	42.38	44.88
	MATH500 (4-shot)	4.60	17.40	16.40	26.40	27.80
	Olympiad Bench (4-shot)	0.89	2.96	3.41	5.48	3.85
	AMC23 (0-shot)	0.00	10.00	7.50	10.00	10.00
	Average	3.29	15.32	**16.12**	**21.07**	**21.63**
Other	MATH (4-shot)	4.34	18.26	21.74	28.50	29.98
	SAT MATH (4-shot)	12.50	46.88	31.25	56.25	46.88
	MathQA (8-shot)	12.20	24.80	33.20	36.90	36.70
	MMLU STEM (4-shot)	19.90	35.59	36.45	38.60	37.91
	OCW Course (4-shot)	4.41	6.25	4.04	6.25	6.62
	MAWPS (8-shot)	43.05	79.47	83.15	88.57	88.09
	SVAMP (8-shot)	20.90	47.10	55.80	63.20	61.20
	ASDiv (8-shot)	34.53	69.96	72.55	75.30	75.26
	TabMWP (8-shot)	24.40	45.10	50.10	51.60	51.20
	Average	19.58	41.49	**43.14**	**49.46**	**48.20**

Table 3: Evaluation results on OctoThinker-3B series.

	Benchmarks	Llama-3.2-3B	OctoThinker-3B-Base			
			Stable	Long	Hybrid	Short
Core	GSM8K (8-shot)	30.48	55.95	56.10	64.37	63.31
	MATH500 (4-shot)	7.40	22.40	25.80	30.80	31.40
	Olympiad Bench (4-shot)	2.07	3.41	4.74	4.00	4.74
	AMC23 (0-shot)	2.50	5.00	7.50	10.00	2.50
	Average	10.61	21.69	**23.54**	**27.29**	**25.49**
Other	MATH (4-shot)	8.24	24.86	29.98	31.76	32.70
	SAT MATH (4-shot)	25.00	59.38	65.63	59.38	53.13
	MathQA (8-shot)	18.20	39.50	45.40	47.50	49.80
	MMLU STEM (4-shot)	38.63	46.32	48.11	49.73	48.87
	OCW Course (4-shot)	5.51	11.40	11.40	8.46	9.19
	MAWPS (8-shot)	79.90	89.69	91.67	94.24	93.51
	SVAMP (8-shot)	52.40	68.40	69.10	78.00	77.30
	ASDiv (8-shot)	60.09	79.59	79.91	82.80	82.26
	TabMWP (8-shot)	48.30	55.60	56.40	57.80	60.00
	Average	37.36	52.75	**55.29**	**56.63**	**56.31**

Table 4: Evaluation results on OctoThinker-8B series.

	Benchmarks	Llama-3.1-8B	OctoThinker-8B-Base			
			Stable	Long	Hybrid	Short
Core	GSM8K (8-shot)	55.11	71.27	72.48	77.41	77.10
	MATH500 (4-shot)	20.80	34.40	37.80	42.60	38.60
	Olympiad Bench (4-shot)	3.56	9.78	11.85	4.74	10.07
	AMC23 (0-shot)	5.00	0.00	5.00	5.00	7.50
	Average	21.12	28.86	**31.78**	**32.44**	**33.32**
Other	MATH (4-shot)	21.36	37.00	41.98	44.82	38.54
	SAT MATH (4-shot)	53.13	81.25	81.25	87.50	87.50
	MathQA (8-shot)	36.00	58.20	62.80	60.50	62.80
	MMLU STEM (4-shot)	54.44	62.03	63.75	64.08	64.38
	OCW Course (4-shot)	12.87	18.38	16.18	15.07	13.97
	MAWPS (8-shot)	90.75	93.08	94.43	95.98	95.54
	SVAMP (8-shot)	70.50	79.50	82.40	86.10	86.40
	ASDiv (8-shot)	72.10	83.79	83.57	84.47	85.33
	TabMWP (8-shot)	63.10	67.90	70.10	68.90	71.60
	Average	52.69	64.57	**66.27**	**67.49**	**67.34**

5. OctoThinker-Zero Families: RL Scaling with Diverse Thinking Behaviors

We further trained all OctoThinker base models—spanning different decay branches and model sizes (1B and 3B)—through a reinforcement learning stage, following our previously established setup. This yielded a family of models optimized specifically for mathematical reasoning. As in the decay stage, the final RL-tuned models fall into three categories, each reflecting the data mixture used during decay and the distinct behaviors shaped during RL: OctoThinker-Short-Zero, OctoThinker-Hybrid-Zero, and OctoThinker-Long-Zero. The training dynamics of these models are shown in Figure 9,10. The OctoThinker-Long branch tends to produce longer responses—within a controlled range—compared to other branches. While it slightly underperforms at the 1B scale, it demonstrates stronger performance as model size increases, such as at 3B.

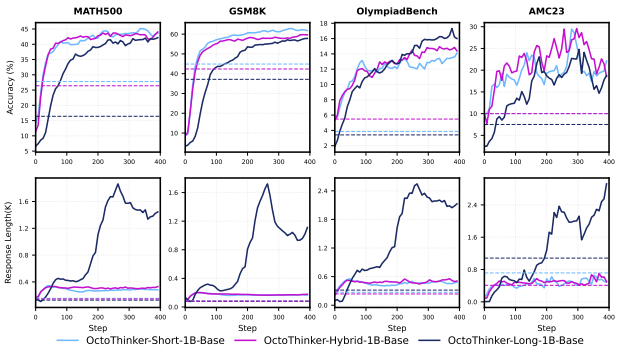

Figure 9: The RL training dynamics across different branches for OctoThinker-1B series

OctoThinker vs. Qwen2.5 An important question we investigate is: *To what extent can our OctoThinker models close the performance gap between the Llama-3.2 series and the stronger Qwen2.5 models in the RL setting?* To demonstrate this, we compare three 3B-scale base models: Llama-3.2-3B-Base, OctoThinker-Long-3B-Base, and

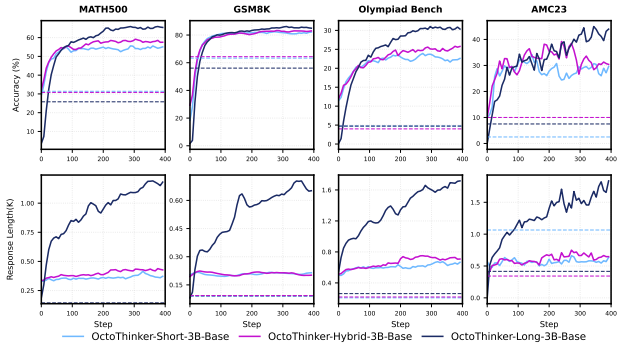

Figure 10: The RL training dynamics across different branches for OctoThinker-3B series.

Qwen2.5-3B-Base. As illustrated in Figure 11, during the RL phase, OctoThinker-Long-3B consistently outperforms the original Llama-3.2-3B model. Remarkably, it reaches performance on par with Qwen2.5-3B, a model known for its strong reasoning capabilities and extensive pre-training, while the hybrid and short branches are marginally inferior, especially on challenging benchmarks. Overall, these results highlight the substantial gains introduced by our mid-training strategy and confirm that OctoThinker effectively narrows the performance gap, elevating Llama-3.2 models to a new level of competitiveness in math reasoning tasks.

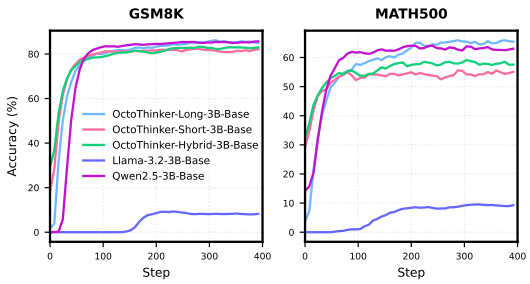

Figure 11: RL dynamics of Llama-3.2-3B, OctoThinker-3B, and Qwen2.5-3B (see full results in Figure 16).

6. Related Works

Understanding RL along with Language Models Large-scale RL has driven the major advances in language models on reasoning-intensive tasks, such as competition-level math (e.g., AIME), exemplified by OpenAI's o1 (OpenAI et al., 2024), o3 (OpenAI, 2025) and DeepSeek's R1 (Guo et al., 2025). A wave of follow-up studies (Zeng et al., 2025; Hu et al., 2025; Yu et al., 2025; Luo et al., 2025) explored RL on smaller language models (less than 100B parameters), yet these successes are overwhelmingly limited to Qwen family. In contrast, replicating such results on other major model families—e.g., Llama-has proven difficult (Gandhi et al., 2025; Liu et al., 2025). The opacity of pre-training pipelines hinders our understanding of how pre-training interacts with RL scaling, prompting some unconventional investigations (Wang et al., 2025; Shao et al., 2025). For instance, Wang et al. (2025) showed that even one-shot prompting can enhance reasoning in Qwen, but yields minimal gains in Llama. The underlying science remains essential but largely unexplored. Our work takes a step toward filling this gap by performing controlled mid-training interventions on Llama models, revealing key factors that enable effective RL scaling. Building on these insights, we introduce OctoThinker via a two-stage mid-training strategy (over 200B tokens), followed by RL training, yielding models that match Qwen's performance at the same scale.

Curation of Math Pre-training Corpora Pre-training corpora are foundational to language models, especially for math reasoning tasks where large-scale mid-training is infeasible without high-quality, domain-specific data. Early open-source efforts—such as OpenWebMath (Paster et al., 2024), MathPile (Wang et al., 2024), InfiMM-Web-Math (Han et al., 2024), and FineMath (Allal et al., 2025)—have made meaningful progress but remain constrained in scale, typically under 100B tokens. The release of MegaMath (Zhou et al., 2025) marked a turning point, enabling scalable mid-training in this work. Building on its foundation, we curated a new reasoning-intensive and carefully refined math corpus, MegaMath-Web-Pro-Max , which exceeds 70B tokens and matches the quality of MegaMath-Web-Pro. This corpus powers the first stage of our mid-training and will be released to support the broader open-source community.

7. Conclusion and Future Work

In this work, we investigated why base models like Llama and Qwen exhibit divergent behaviors during reinforcement learning for reasoning and demonstrated that mid-training plays a decisive role. Our findings show that high-quality, reasoning-intensive corpora—especially those like MegaMath-Web-Pro—can substantially improve RL stability and effectiveness. Building on these insights, we introduced a two-stage mid-training strategy that transforms Llama into a more RL-scalable foundation model. The resulting OctoThinker models achieve strong performance across mathematical reasoning tasks, closing the gap with RL-friendly model families. We hope this work provides a foundation for designing future base models better aligned with the demands of reasoning-centric RL.

Furthermore, we will actively explore more in the future, include: (1) curating higher-quality math corpora to further enhance mid-training; (2) designing RL-friendly base models using open recipes without distillation from those powerful long CoT reasoning models; (3) disentangling QA format and content to better understand their individual contributions; and (4) extending the OctoThinker families with additional branches, such as *tool-integrated reasoning*. We believe these efforts will provide deeper insights into the interplay between pre-training and reinforcement learning.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abdin, M. I., Aneja, J., Behl, H. S., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., Lee, J. R., Lee, Y. T., Li, Y., Liu, W., Mendes, C. C. T., Nguyen, A., Price, E., de Rosa, G., Saarikivi, O., Salim, A., Shah, S., Wang, X., Ward, R., Wu, Y., Yu, D., Zhang, C., and Zhang, Y. Phi-4 technical report. *CoRR*, abs/2412.08905, 2024a. doi: 10.48550/ARXIV.2412.08905. URL https://doi.org/10.48550/arXiv.2412.08905.

Abdin, M. I., Jacobs, S. A., Awan, A. A., Aneja, J., Awadallah, A., Awadalla, H., Bach, N., Bahree, A., Bakhtiari, A., Behl, H. S., Benhaim, A., Bilenko, M., Bjorck, J., Bubeck, S., Cai, M., Mendes, C. C. T., Chen, W., Chaudhary, V., Chopra, P., Giorno, A. D., de Rosa, G., Dixon, M., Eldan, R., Iter, D., Garg, A., Goswami, A., Gunasekar, S., Haider, E., Hao, J., Hewett, R. J., Huynh, J., Javaheripi, M., Jin, X., Kauffmann, P., Karampatziakis, N., Kim, D., Khademi, M., Kurilenko, L., Lee, J. R., Lee, Y. T., Li, Y., Liang, C., Liu, W., Lin, E., Lin, Z., Madan, P., Mitra, A., Modi, H., Nguyen, A., Norick, B., Patra, B., Perez-Becker, D., Portet, T., Pryzant, R., Qin, H., Radmilac, M., Rosset, C., Roy, S., Ruwase, O., Saarikivi, O., Saied, A., Salim, A., Santacroce, M., Shah, S., Shang, N., Sharma, H., Song, X., Tanaka, M., Wang, X., Ward, R., Wang, G., Witte, P. A., Wyatt, M., Xu, C., Xu, J., Yadav, S., Yang, F., Yang, Z., Yu, D., Zhang, C., Zhang, C., Zhang, J., Zhang, L. L., Zhang, Y., Zhang, Y., Zhang, Y., and Zhou, X. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219, 2024b. doi: 10.48550/ARXIV.2404.14219. URL https://doi.org/10.48550/arXiv.2404.14219.

Allal, L. B., Lozhkov, A., Bakouch, E., Blázquez, G. M., Penedo, G., Tunstall, L., Marafioti, A., Kydlícek, H., Lajarín, A. P., Srivastav, V., Lochner, J., Fahlgren, C., Nguyen, X., Fourrier, C., Burtenshaw, B., Larcher, H., Zhao, H., Zakka, C., Morlon, M., Raffel, C., von Werra, L., and Wolf, T. Smollm2: When smol goes big - data-centric training of a small language model. *CoRR*, abs/2502.02737, 2025. doi: 10.48550/ARXIV.2502.02737. URL https://doi.org/10.48550/arXiv.2502.02737.

Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y., and Hajishirzi, H. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2357–2367, 2019.

Azerbayev, Z., Schoelkopf, H., Paster, K., Santos, M. D., McAleer, S. M., Jiang, A. Q., Deng, J., Biderman, S., and Welleck, S. Llemma: An open language model for mathematics. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4WnqRR915j.

Bercovich, A., Ronen, T., Abramovich, T., Ailon, N., Assaf, N., Dabbah, M., Galil, I., Geifman, A., Geifman, Y., Golan, I., Haber, N., Karpas, E., Koren, R., Levy, I., Molchanov, P., Mor, S., Moshe, Z., Nabwani, N., Puny, O., Rubin, R., Schen, I., Shahaf, I., Tropp, O., Argov, O. U., Zilberstein, R., and El-Yaniv, R. Puzzle: Distillation-based NAS for inference-optimized llms. *CoRR*, abs/2411.19146, 2024. doi: 10.48550/ARXIV.2411.19146. URL https://doi.org/10.48550/arXiv.2411.19146.

Bercovich, A., Levy, I., Golan, I., Dabbah, M., El-Yaniv, R., Puny, O., Galil, I., Moshe, Z., Ronen, T., Nabwani, N., et al. Llama-nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949*, 2025.

Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ding, N., Chen, Y., Xu, B., Qin, Y., Hu, S., Liu, Z., Sun, M., and Zhou, B. Enhancing chat language models by scaling high-quality instructional conversations. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 3029–3051. Association for Computational Linguistics, 2023a. doi: 10.18653/V1/2023.EMNLP-MAIN.183. URL https://doi.org/10.18653/v1/2023.emnlp-main.183.

Ding, N., Chen, Y., Xu, B., Qin, Y., Zheng, Z., Hu, S., Liu, Z., Sun, M., and Zhou, B. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023b.

Dou, L., Liu, Q., Zhou, F., Chen, C., Wang, Z., Jin, Z., Liu, Z.-Y., Zhu, T., Du, C., Yang, P., Wang, H., Liu,

J., Zhao, Y., Feng, X., Mao, X., Yeung, M. T., Pipatanakul, K., Koto, F., Thu, M. S., Kydl'ivcek, H., Liu, Z.-X., Lin, Q., Sripaisarnmongkol, S., Sae-Khow, K., Thongchim, N., Konkaew, T., Borijindargoon, N., Dao, A., Maneegard, M., Artkaew, P., Yong, Z.-X., Nguyen, Q., Phatthiyaphaibun, W., Tran, H. H., Zhang, M., Chen, S., Pang, T., Du, C., Wan, X., Lu, W., and Lin, M. Sailor2: Sailing in south-east asia with inclusive multilingual llms. *ArXiv*, abs/2502.12982, 2025. URL https://arxiv.org/abs/2502.12982.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Rozière, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I. M., Misra, I., Evtimov, I., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Gandhi, K., Chakravarthy, A., Singh, A., Lile, N., and Goodman, N. D. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Han, X., Jian, Y., Hu, X., Liu, H., Wang, Y., Fan, Q., Ai, Y., Huang, H., He, R., Yang, Z., and You, Q. InfiMM-webmath-40b: Advancing multimodal pre-training for enhanced mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL https://openreview.net/forum?id=Twzrpa6V2o.

He, C., Luo, R., Bai, Y., Hu, S., Thai, Z. L., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., Liu, J., Qi, L., Liu, Z., and Sun, M. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Ku, L., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.211. URL https://doi.org/10.18653/v1/2024.acl-long.211.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Hu, J., Zhang, Y., Han, Q., Jiang, D., Zhang, X., and Shum, H.-Y. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

Hu, S., Tu, Y., Han, X., He, C., Cui, G., Long, X., Zheng, Z., Fang, Y., Huang, Y., Zhao, W., Zhang, X., Thai, Z. L., Zhang, K., Wang, C., Yao, Y., Zhao, C., Zhou, J., Cai, J., Zhai, Z., Ding, N., Jia, C., Zeng, G., Li, D., Liu, Z., and Sun, M. Minicpm: Unveiling the potential of small language models with scalable training strategies. *CoRR*, abs/2404.06395, 2024a. doi: 10.48550/ARXIV.2404.06395. URL https://doi.org/10.48550/arXiv.2404.06395.

Hu, S., Tu, Y., Han, X., He, C., Cui, G., Long, X., Zheng, Z., Fang, Y., Huang, Y., Zhao, W., et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024b.

HuggingFace. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Koncel-Kedziorski, R., Roy, S., Amini, A., Kushman, N., and Hajishirzi, H. Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pp. 1152–1157, 2016.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., Gu, Y., Malik, S., Graf, V., Hwang, J. D., Yang, J., Bras, R. L., Tafjord, O., Wilhelm, C., Soldaini, L., Smith, N. A., Wang, Y., Dasigi, P., and Hajishirzi, H. Tülu 3: Pushing frontiers in open language model post-training. *CoRR*, abs/2411.15124, 2024a. doi: 10.48550/ARXIV.

2411.15124. URL https://doi.org/10.48550/arXiv.2411.15124.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., et al. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024b.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V. V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., and Misra, V. Solving quantitative reasoning problems with language models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/18abbeef8cfe9203fdf9053c9c4fe191-Abstract-Conference.html.

Li, J., Fang, A., Smyrnis, G., Ivgi, M., Jordan, M., Gadre, S. Y., Bansal, H., Guha, E., Keh, S. S., Arora, K., et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *CoRR*, abs/2305.20050, 2023. doi: 10.48550/ARXIV.2305.20050. URL https://doi.org/10.48550/arXiv.2305.20050.

Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Lu, P., Qiu, L., Chang, K.-W., Wu, Y. N., Zhu, S.-C., Rajpurohit, T., Clark, P., and Kalyan, A. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *International Conference on Learning Representations (ICLR)*, 2023.

Luo, M., Tan, S., Wong, J., Shi, X., Tang, W. Y., Roongta, M., Cai, C., Luo, J., Li, L. E., Popa, R. A., and Stoica, I. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. Notion Blog, 2025. Notion Blog.

Meta, Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Miao, S.-Y., Liang, C.-C., and Su, K.-Y. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, 2020.

OLMo, T., Walsh, P., Soldaini, L., Groeneveld, D., Lo, K., Arora, S., Bhagia, A., Gu, Y., Huang, S., Jordan, M., Lambert, N., Schwenk, D., Tafjord, O., Anderson, T., Atkinson, D., Brahman, F., Clark, C., Dasigi, P., Dziri, N., Guerquin, M., Ivison, H., Koh, P. W., Liu, J., Malik, S., Merrill, W., Miranda, L. J. V., Morrison, J. D., Murray, T. C., Nam, C., Pyatkin, V., Rangapur, A., Schmitz, M., Skjonsberg, S., Wadden, D., Wilhelm, C., Wilson, M., Zettlemoyer, L. S., Farhadi, A., Smith, N. A., and Hajishirzi, H. 2 olmo 2 furious. *ArXiv*, abs/2501.00656, 2025. URL https://arxiv.org/abs/2501.00656.

OpenAI. Introducing openai o3 and o4-mini — openai, April 2025. URL https://openai.com/index/introducing-o3-and-o4-mini/.

OpenAI, El-Kishky, A., Selsam, D., Song, F., Parascandolo, G., Ren, H., Lightman, H., Won, H., Akkaya, I., Sutskever, I., Wei, J., Gordon, J., Cobbe, K., Yu, K., Kondraciuk, L., Schwarzer, M., Rohaninejad, M., Brown, N., Zhao, S., Bansal, T., Kosaraju, V., Leadership, W. Z., Pachocki, J. W., Tworek, J., Fedus, L., Kaiser, L., Chen, M., Sidor, S., Zaremba, W., Karpenko, A., Wei, A., Tam, A., Kumar, A., Saraiva, A., Kondrich, A., drey Mishchenko, A., Nair, A., Ghorbani, B., McKinzie, B., don Eastman, C. B., Li, M., Koch, C., Roberts, D., Dohan, D., Mély, D., Tsipras, D., Cheung, E., Wallace, E., Salman, H., ing Bao, H., Bagher-inezhad, H., Kostrikov, I., Feng, J., Rizzo, J., Nguyen, K., Lu, K., Stone, K. R., Kuhn, L., Meyer, M., Pavlov, M., McAleese, N., Boiko, O., Murk, O., Zhokhov, P., Lin, R., Gaon, R., Garg, R., James, R., Shu, R., McKinney, S., Santurkar, S., Balaji, S., Gordon, T., Dimson, T., Zheng, W., Jaech, A., Lerer, A., Low, A., Carney, A., Neitz, A., Prokofiev, A., Sokolowsky, B., Barak, B., Minaiev, B., Hao, B., Baker, B., Houghton, B., Lugaresi, C., Voss, C., Shen, C., Orsinger, C., Kappler, D., Levy, D., Li, D., Freeman, E., Wong, E., Wang, F., Such, F. P., Tsimpourlas, F., Salmon, G., Chabot, G., Leclerc, G., Andrin, H., O'Connell, I., Osband, I. I., Gilaberte, C., Harb, J., Yu, J., Weng, J., Palermo, J., Hallman, J., Ward, J., Wang, J., Chen, K., Shi, K., Gu-Lemberg, K., Liu, K., Liu, L., Li, L., Metz, L., Trebacz, M., Joglekar, M. R., Tintor, M., Guan, M. Y., Yan, M., Glaese, M., Malek, M., Fradin, M., Bavarian, M., Tezak, N. A., Nachum,

O., Ashbourne, P., Izmailov, P., Lopes, R. G., Miyara, R., Leike, R. H., Brown, R., Cheu, R., Greene, R., Jain, S., Yan, S., Hu, S., Zhang, S., Fu, S., Papay, S., Sanjeev, S., Wang, T., Sanders, T., Patwardhan, T., Sottiaux, T., Zheng, T., Garipov, T., Qi, V., Pong, V. H., Fomenko, V., Lu, Y., Chen, Y., Bai, Y., He, Y., Zhang, Y., Shao, Z., Li, Z., Yang, L., Chen, M., Clark, A., Yu, J., Xiao, K., Toizer, S., Agarwal, S., Research, S., Vallone, A., Zhang, C., Kivlichan, I., Shah, M., Toyer, S., Chaudhuri, S. R., Lin, S., Richardson, A., Duberstein, A., de Bourcy, C., Oprica, D., Leoni, F., laine Boyd, M., Jones, M., Kaufer, M., Yatbaz, M. A., Xu, M., McClay, M., Wang, M., Creech, T., Monaco, V., Ritter, E., Mays, E., Parish, J., Uesato, J., Maksin, L., Wang, M., Wang, M., Chowdhury, N., Watkins, O., Chao, P., Dias, R., Miserendino, S., Teaming, R., Ahmad, L., Lampe, M., Peterson, T., and Huizinga, J. Openai o1 system card. *ArXiv*, abs/2412.16720, 2024. URL https://arxiv.org/abs/2412.16720.

Paster, K., Santos, M. D., Azerbayev, Z., and Ba, J. Openwebmath: An open dataset of high-quality mathematical web text. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jKHmjlpViu.

Patel, A., Bhattamishra, S., and Goyal, N. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, 2021.

Shao, R., Li, S. S., Xin, R., Geng, S., Wang, Y., Oh, S., Du, S. S., Lambert, N., Min, S., Krishna, R., Tsvetkov, Y., Hajishirzi, H., Koh, P. W., and Zettlemoyer, L. S. Spurious rewards: Rethinking training signals in rlvr. 2025. URL https://arxiv.org/abs/2506.10947.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.

Wake, A., Chen, B., Lv, C. X., Li, C., Huang, C., Cai, C., Zheng, C., Cooper, D., Zhou, F., Hu, F., Wang, G., Ji, H., Qiu, H., Zhu, J., Tian, J., Su, K., Zhang, L., Li, L., Song, M., Li, M., Liu, P., Hu, Q., Wang, S., Zhou, S., Yang, S., Li, S., Zhu, T., Xie, W., He, X., Chen, X., Hu, X., Ren, X., Niu, X., Li, Y., Zhao, Y., Luo, Y., Xu, Y., Sha, Y., Yan, Z., Liu, Z., Zhang, Z., and Dai, Z. Yi-lightning technical report. *CoRR*, abs/2412.01253, 2024. doi: 10.48550/ARXIV.2412.01253. URL https://doi.org/10.48550/arXiv.2412.01253.

Wang, Y., Yang, Q., Zeng, Z., Ren, L., Liu, L., Peng, B., Cheng, H., He, X., Wang, K., Gao, J., Chen, W., Wang, S., Du, S. S., and Shen, Y. Reinforcement learning for reasoning in large language models with one training example. *CoRR*, abs/2504.20571, 2025. doi: 10.48550/ARXIV.2504.20571. URL https://doi.org/10.48550/arXiv.2504.20571.

Wang, Z., Li, X., Xia, R., and Liu, P. Mathpile: A billion-token-scale pretraining corpus for math. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=RSvhU69sbG.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Confere
html.

Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu, D., Tu, J., Zhou, J., Lin, J., Lu, K., Xue, M., Lin, R., Liu, T., Ren, X., and Zhang, Z. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *ArXiv*, abs/2409.12122, 2024a. URL https://arxiv.org/abs/2409.12122.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yang, Q. A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y.-C., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z., Quan, S., and Wang, Z. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024b. URL https://arxiv.org/abs/2412.15115.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Fan, T., Liu, G., Liu, L., Liu, X., Lin, H., Lin, Z., Ma, B., Sheng, G., Tong, Y., Zhang, C., Zhang, M., Zhang, W., Zhu, H., Zhu, J., Chen, J., Chen, J., Wang, C., Yu, H., Dai, W., Song, Y., Wei, X., Zhou, H., Liu, J., Ma, W., Zhang, Y., Yan, L., Qiao, M., Wu, Y., and Wang, M. DAPO: an

open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476, 2025. doi: 10.48550/ARXIV.2503.14476. URL https://doi.org/10.48550/arXiv.2503.14476.

Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z., and He, J. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

Zhao, W., Ren, X., Hessel, J., Cardie, C., Choi, Y., and Deng, Y. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Bl8u7ZRlbM.

Zhou, F., Wang, Z., Ranjan, N., Cheng, Z., Tang, L., He, G., Liu, Z., and Xing, E. P. Megamath: Pushing the limits of open math corpora. *ArXiv*, abs/2504.02807, 2025. URL https://arxiv.org/abs/2504.02807.

A. Details for Preliminaries

A.1. Details for RL Setup

We performed RL experiments based on the verl (Sheng et al., 2024) framework and utilized the GRPO (Shao et al., 2024) algorithm. For RL training prompts, we adopted the MATH8K[4] dataset due to its moderate difficulty and concise composition. The maximum response length is set to 4,096 tokens. We configured the global training batch size to 128, set the number of rollout responses per query to 16, and used a PPO mini-batch size of 64. The sampling temperature is set to 1.0, with a maximum output length of 4096 tokens. We used a learning rate of 1×10^{-6} and set the KL loss coefficient to 0 in the verl configuration. Empirically, we found that setting the ratio between sampling and gradient updates to 2 leads to more stable RL training. Unless otherwise specified, we employed a simple prompt template of "`Question:{}\nAnswer:{}`" to format training examples.

For evaluation, we employ GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2023), OlympiadBench (He et al., 2024), and AMC23 as indicator tasks to analyze RL dynamics. To assess base model performance, we further include MATH (Hendrycks et al., 2021), SAT-MATH (Azerbayev et al., 2024) , MathQA (Amini et al., 2019), MMLU-STEM (Hendrycks et al., 2021), OCW Course (Lewkowycz et al., 2022), MAWPS (Koncel-Kedziorski et al., 2016), SVAMP (Patel et al., 2021), ASDiv (Miao et al., 2020), and TabMWP (Lu et al., 2023).

A.2. Details for The Curation of MegaMath-Web-Pro-Max

Scoring Prompt of Usefulness for Studying Mathematics

Evaluate the following text extract for its potential usefulness for studying mathematics up to high school and early undergraduate levels. Use the following 5-point scoring system described below. Points are accumulated based on the satisfaction of each criterion:
- Add 1 point if the extract contains some mathematical content, even if it's not very useful for studying, or if it contains non-academic content such as advertisements and generated pages for converting weight and currencies.
- Add another point if the extract touches on mathematical topics, even if it's poorly written if it's too complex such as an academic paper that is too advanced.
- Award a third point if the extract demonstrates problem solving or logical reasoning in a mathematical context, even if it lacks step-by-step explanations.
- Grant a fourth point if the extract is at an appropriate level (up to high school and early undergraduate levels) and contains clear mathematical deductions and step-by-step solutions to mathematical problems. It should be similar to a chapter from a textbook or a tutorial.
- Give a fifth point if the extract is outstanding in its educational value for teaching and studying mathematics in middle school and high school. It should include very detailed and easy to follow explanations.
Question-answer formats (e.g., from educational websites or forums) are acceptable if they meet the criteria.
The text extract:
``
<document>
''
After examining the extract:
- Briefly justify your total score, up to 100 words.
- Conclude with the score using the format: Final score: <total points>.

Figure 12: Scoring prompt in FineMath (Allal et al., 2025) of usefulness for studying mathematics.

We curated MegaMath-Web-Pro-Max to support large-scale ablation studies and mid-training. The corpus was constructed by leveraging an efficient classifier to recall documents from MegaMath (Zhou et al., 2025), followed by refinement using a powerful instruction-following LLM. Specifically, we uniformly and randomly sampled millions of documents from the MegaMath-Web corpus, stratified by publication year, and annotated them using Llama-3.1-70B-instruct. Each document was graded for its usefulness in studying mathematics on a scale from 0 to 5 using a grading prompt (see Figure 12). We

[4] https://hf.co/datasets/hkust-nlp/SimpleRL-Zoo-Data/tree/main/simplelr_qwen_level3to5

heuristically extracted scores from the model's critiques: documents scoring below 3 were labeled as negative examples, while those scoring 3 or above were considered positive. To improve the classifier's ability to recall reasoning-intensive content, we supplemented the positive seed set with more than 200K long CoT examples from OpenR1-Math-220K, using only the detailed deep thinking steps without their summaries. We balanced the distribution of positive and negative examples, totalling about 2.5 million samples. We observed that existing classifiers, such as `finemath-classifier`, are highly sensitive to the choices of text extractors during data curation. This motivated us to train our own classifier, selecting `fasttext` for its efficiency. Consistent with the findings of Zhou et al. (2025), we found preprocessing to be critical for recall performance. Our preprocessing pipeline included lowercasing text, filtering excessively long words, and removing line breaks and extraneous non-alphanumeric characters. Given the noisy and poorly structured nature of many documents, we employed Llama-3.1-70B-instruct to refine the text using a prompt (see Figure 13) inspired by MegaMath-Web-Pro. The resulting dataset, MegaMath-Web-Pro-Max , contains approximately 5.5 times more tokens than MegaMath-Web-Pro. Empirical evaluations during pre-training indicate that MegaMath-Web-Pro-Max maintains comparable data quality, making it a strong candidate as a foundational corpus for large-scale mid-training.

Web Text Refinement Prompt

Task:
- Carefully analyze the provided text to extract key facts, concrete details, important numbers, and core concepts.
- Remove any irrelevant or noisy information, and reorganize the content into a logically structured, information-dense, and concise version that is easy to learn from. Output only the refined text.
- Strive to maintain the original length as much as possible (avoid excessive shortening).
- Refine multiple choice questions and answers if any.

Text:
`<EXAMPLE>`
Just output the refined text, no other text.

Figure 13: Web text refinement prompt used in MegaMath-Web-Pro (Zhou et al., 2025)

A.3. Details for Controllable Mid-training Analysis

We provide the *complex template* for RL training mentioned in § 3.4 in Figure 14.

Complex Template for RL

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. User: You must put your answer inside `\\boxed{}` and your final answer will be extracted automatically by the `\\boxed{}` tag.
`{{prompt}}`
Assistant:

Figure 14: Complex Template for RL training adapted from the prompt design in Open-Reasoner-Zero (Hu et al., 2025)

A.4. Details for OctoThinker-Base

Tables 5 and 6 detail the data mixture and training hyperparameters for the stable stage, while Tables 7 and 8 present those for the decay stage. Figure 15 illustrates the effect of different QA data mixtures in the decay stage on the downstream performance.

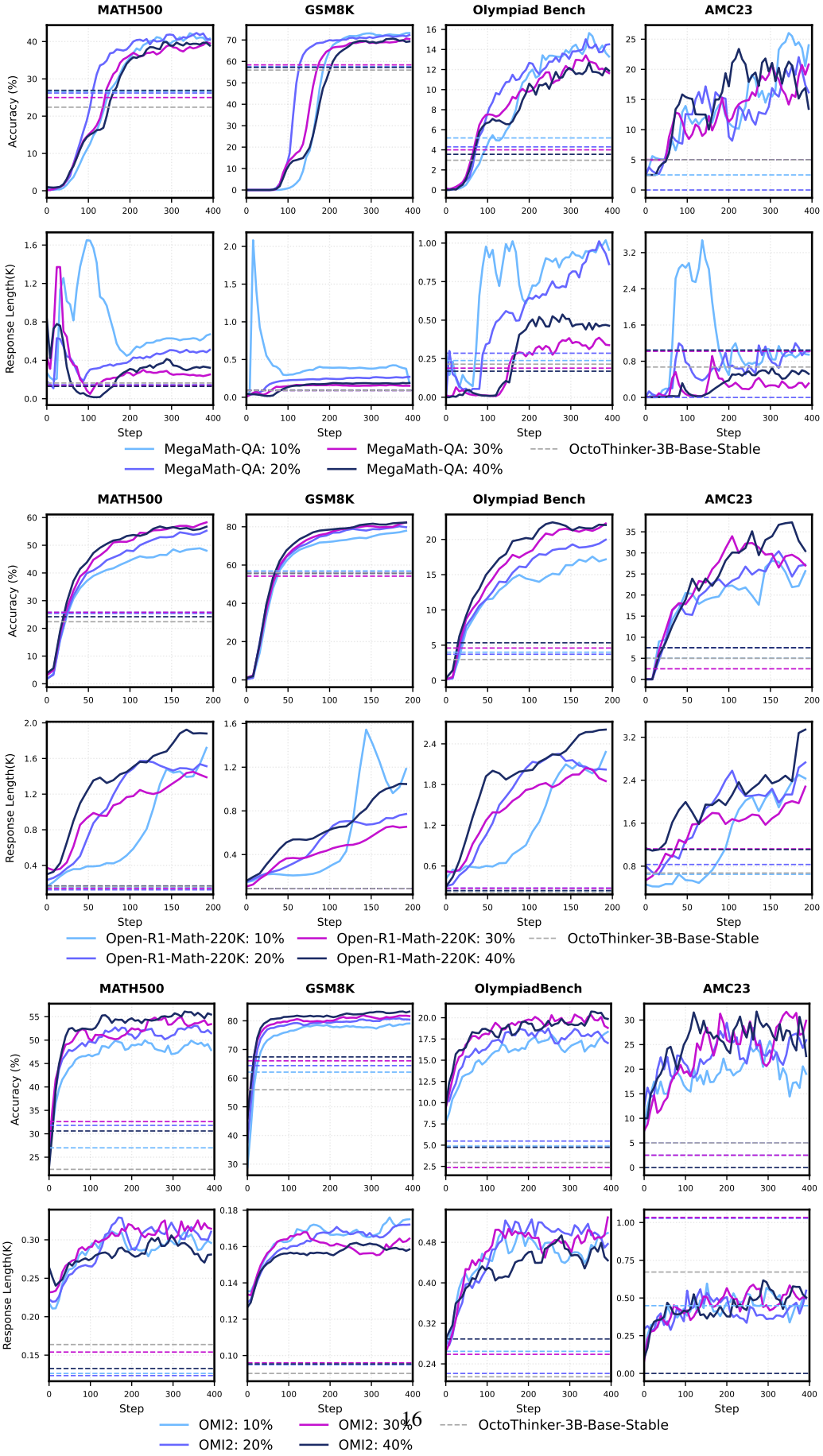

Figure 15: RL dynamics under different QA datasets and mixing ratios during the decay stage.

Table 5: Dataset composition and weights in the first stage.

Dataset	Weight
DCLM-Baseline	0.10
MegaMath-Web-Pro-Max	0.725
MegaMath-Code	0.0125
MegaMath-QA	0.05
MegaMath Trans. Code	0.0125
MegaMath Text Code Block	0.10

Table 6: hyper-parameters in the stable stage.

Hyper-parameter	Llama-3.2-1B / 3B / 8B
Context Length	8,192
Batch Size	512
Max Steps	50,000
Warmup Steps	0
Weight Decay	0.1
Optimizer	AdamW
LR Scheduler	Constant
Learning Rate (LR)	5e-5/2e-5/1e-5

Table 7: Specific data mixture for each branch in the decay stage

(a) Long Branch Mixture

Dataset	Weight
DCLM-Baseline	0.05
Instruction Following	0.10
MegaMath-Web-Pro	0.55
Open R1	0.15
AM-DeepSeek-Distilled-40M	0.15

(b) Short Branch Mixture

Dataset	Weight
DCLM-Baseline	0.05
Instruction Following	0.10
MegaMath-Web-Pro	0.55
MegaMath-QA	0.025
OpenMathInstruct2	0.175
NuminaMath1.5	0.10

(c) Hybrid Branch Mixture

Dataset	Weight
DCLM-Baseline	0.05
Instruction Following	0.10
MegaMath-Web-Pro	0.55
OpenMathInstruct2	0.10
NuminaMath1.5	0.10
Open R1	0.10

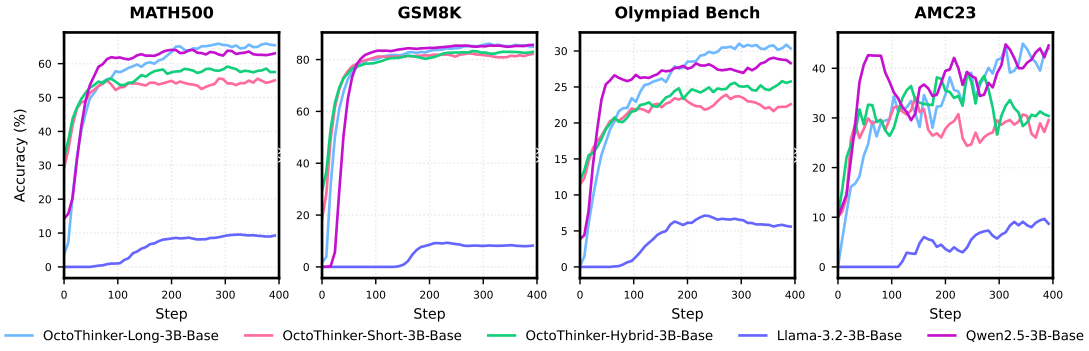

Figure 16: RL training dynamics among Llama-3.2-3B-Base, OctoThinker series and Qwen2.5-Base.

Table 8: Hyper-parameters for decay stage.

Hyper-parameter	Llama-3.2-1B / 3B / 8B
Context Length	8,192
Batch Size	512
Max Steps	5,000
Warmup Steps	0
Weight Decay	0.1
Optimizer	AdamW
LR Scheduler	Cosine Decay 5e-5→5e-6 / 2e-5→2e-6 / 1e-5→1e-6