Open-Reasoner-Zero: An Open Source Approach to Scaling Up Reinforcement Learning on the Base Model

Jingcheng Hu[1,2*], Yinmin Zhang[1], Qi Han[1], Daxin Jiang[1], Xiangyu Zhang[1], Heung-Yeung Shum[2]

[1]**StepFun**, [2]**Tsinghua University**

GitHub: `https://github.com/Open-Reasoner-Zero/Open-Reasoner-Zero`,
HuggingFace: `https://huggingface.co/Open-Reasoner-Zero`.

Abstract

We introduce Open-Reasoner-Zero, the first open source implementation of large-scale reasoning-oriented RL training focusing on scalability, simplicity and accessibility. Through extensive experiments, we demonstrate that a minimalist approach, vanilla PPO with GAE ($\lambda = 1$, $\gamma = 1$) and straightforward rule-based reward function, without any KL regularization, is sufficient to scale up both response length and benchmark performance on reasoning tasks, similar to the phenomenon observed in DeepSeek-R1-Zero. Notably, our implementation outperforms DeepSeek-R1-Zero-Qwen-32B on the GPQA Diamond benchmark, while only requiring 1/30 of the training steps. In the spirit of open source, we release our source code, parameter settings, training data, and model weights.

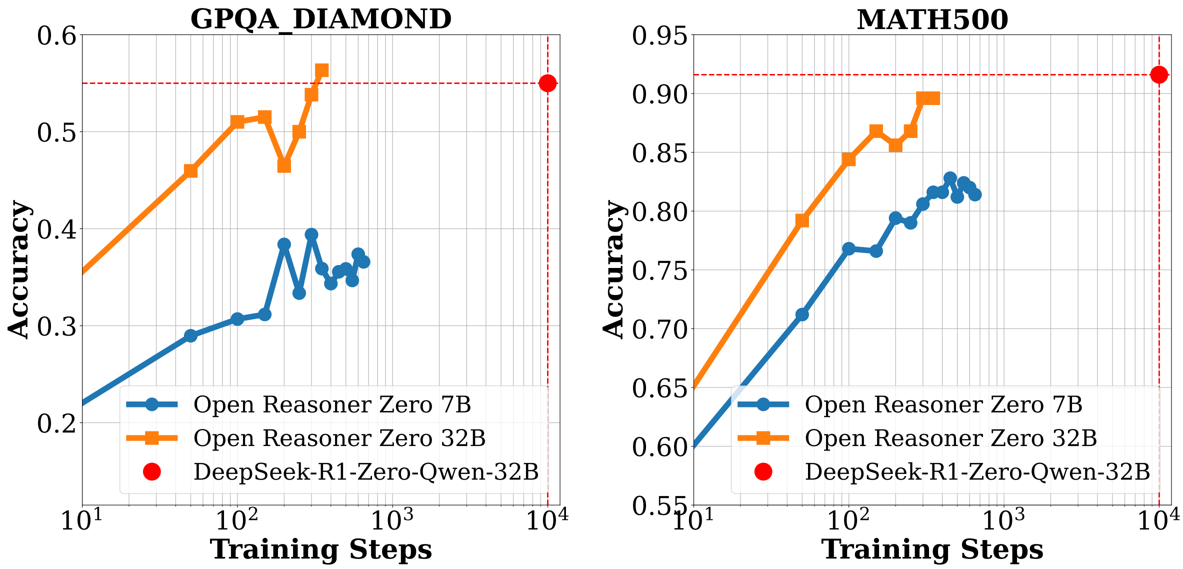

Figure 1: Evaluation performance of Open-Reasoner-Zero-{7B, 32B}. We report the average accuracy on the benchmarks for each question with 16 responses. Notably, Open-Reasoner-Zero-32B outperforms DeepSeek-R1-Zero-Qwen-32B on the GPQA Diamond benchmark while only requiring 1/30 of the training steps. We are continuing to scale up these RL settings until this preprint is released, as there is no sign of saturation yet.

*Work done during internship at StepFun.

Contents

1. Introduction

Large-scale reinforcement learning (RL) training of language models on reasoning tasks has emerged as a promising paradigm for mastering complex problem-solving skills. Recent breakthroughs, particularly OpenAI's o1 [1] and DeepSeek's R1-Zero [2], have demonstrated remarkable training time scaling phenomenon: as the training computation scales up, both the model's benchmark performance and response length consistently and steadily increase without any sign of saturation. Inspired by these advancements, we aim to explore this new scaling phenomenon by conducting large-scale RL training directly on base models, an approach we refer to as Reasoner-Zero training.

In this work, we introduce Open-Reasoner-Zero (ORZ), the first open-source implementation of large-scale reasoning-oriented RL training on large language models (LLMs) with our best practices, designed to be robust, scalable and simple-to-follow. Under Reasoner-Zero paradigm, LLMs are trained to master diverse reasoning skills under verifiable rewards, spanning arithmetic, logic, coding and common-sense reasoning (*e.g.*, scientific problems, numerical reasoning, natural language understanding and even creative writing). While DeepSeek's R1-Zero outlined their training pipeline briefly, we provide a comprehensive study of our training strategy, with in-depth insights into overcoming common challenges such as training instability, stagnating response length, benchmark performance plateaus, and reward design. Our goal is to democratize advanced RL training techniques accessible to the broader research community.

Our proposed Open-Reasoner-Zero-32B outperforms the DeepSeek-R1-Zero-Qwen-32B, with the same Qwen-32B base model, on GPQA Diamond benchmark, yet requires 1/30 iterations. We have conducted tens of thousands of iterations to explore our best practical setting. Through extensive ablation studies, we summarize some key findings and lessons learned from our exploration. Specifically, vanilla PPO using GAE ($\lambda = 1$ and $\gamma = 1$) and without any KL-related regularization, combined with a straightforward rule-based reward function, is sufficient to achieve steady scalability in both response length and benchmark performances across varying model sizes and training data scales. Open-Reasoner-Zero's stable scaling resonates well with the bitter lesson [3]: the most significant performance improvements stem from the scale of training data, model size, and training iterations, rather than the complexity of design choices. The most critical thing is how to design a simple and effective RL algorithm to scale up the training process.

We are excited to share this breakthrough of scale-up RL, along with the lessons learnt with the research community, enabling everyone to not only utilize the end results (*e.g.*, APIs or model weights), but to experience and participate in this transformative moment in AI development themselves. To help facilitate reproducibility and further research in large-scale RL directly training on LLMs, we are committed to release all of our training resources, including code, parameters, data, and model weights.

Our primary contributions are as follow:

1. We provide a fully open-source implementation of large-scale RL training directly on a base LLM, a strategy we refer to as Open-Reasoner-Zero.
2. We share empirical insights and lessons learned from frustrating failures and exciting breakthroughs during our scaling-up journey.
3. We release comprehensive training code, parameter settings, data, and model weights to the research community.

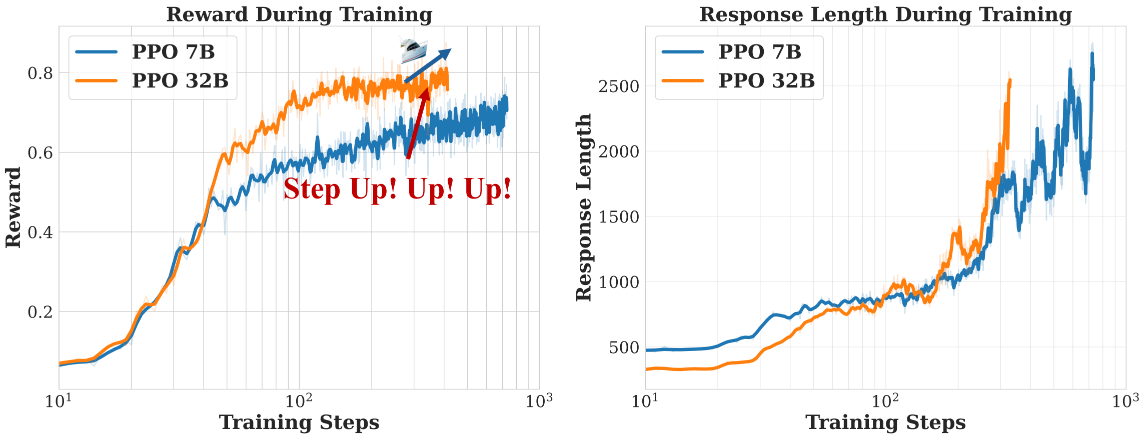

Figure 2: Train Time Scale up on Reward and Response Length of Open-Reasoner-Zero-7B, 32B. The training is still ongoing, and shows no sign of collapse.

2. Scale-up Reinforcement Learning from a Base Model

In this section, we describe the strategy and critical components for scale-up reasoning-oriented reinforcement learning (RL) directly from a base model. First, we introduce the basic yet critical settings for our scale-up RL training from a base model, including data curation, reward function, and detailed settings of the Proximal Policy Optimization (PPO) [4] algorithm. We then discuss key insights derived from our comprehensive ablation experiments that enable successful scale-up RL training.

2.1. Basic Settings

We conduct our experiments utilizing the Qwen2.5-{7B, 32B} as our base model [5], and directly starting the large-scale RL training without any fine-tuning (*e.g.*, distillation or SFT) [6, 7]. Building upon the Qwen2.5-{7B, 32B} base model, we scale up the standard PPO algorithm [4] for reasoning-oriented RL training, with careful consideration of scalability and robustness. Our training data comprises tens of thousands of carefully curated question and answer pairs consisting of STEM, Math, and Reasoning tasks, designed specifically for enhancing models' capability in diverse and complex problem-solving scenarios. Inspired by DeepSeek-R1 [2], we design our prompt template to elicit the model to utilize inference computation, gradually mastering the reasoning ability for complex tasks, as shown in Table 1. Furthermore, we develop an efficient and easy-to-use large-scale RL training framework based on OpenRLHF [8], by introducing a more flexible trainer, enabling GPU collocation generation, and training with offload and backload support. In the following sections, we provide detailed settings for our scale-up RL training from a base model.

2.1.1. Dataset

In this section, we introduce our carefully curated dataset, detailing its source description, cleaning process, and scaling insights for future directions. High-quality training data are crucial for scalable Reasoner-Zero training. We identify three key aspects in our data recipe: quantity, diversity, and quality. Following these key aspects, we curate our dataset through a comprehensive collection and cleaning process:

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. The reasoning process and answer are enclosed within <think> </think> and
<answer> </answer> tags, respectively, i.e., <think> reasoning process here </think>
<answer> answer here </answer>. User: You must put your answer inside <answer> </answer> tags, i.e.,
<answer> answer here </answer>. And your final answer will be extracted automatically by the \boxed{} tag.
{{prompt}}
Assistant: <think>

Table 1: Template for Open-Reasoner-Zero. prompt will be replaced with the specific reasoning question during training.

- We collect public data from various sources, including AIME (up to 2023), MATH, Numina-Math collection [9], Tulu3 MATH [10], and other open-source datasets. Based on source and problem difficulty, we retrieve AMC, AIME, Math, Olympiads, and AoPS forum components as our difficult level prompts to ensure appropriate difficulty levels.
- We synthesize additional reasoning tasks using programmatic approaches to augment the dataset.
- We exclude problems that are challenging to evaluate with our rule-based reward function, such as multiple-choice and proof-oriented problems, ensuring accurate and consistent reward computation during training.
- We implement a model-based filtering strategy based on heuristic evaluation of problem difficulty. Specifically, we use LLM to assess the pass rate of each problem, removing samples with either too high or zero pass rates.
- We apply N-gram and embedding similarity-based filtering to deduplicate samples and maintain data diversity.

The final curated data consists of approximately 57k samples spanning STEM, mathematics, and reasoning domains. This collection is specifically designed to enhance models' capabilities in complex problem-solving tasks, carefully balancing quantity, diversity, and quality. More detailed discussions are provided in the appendix. In the future, we aim to expand our dataset by collaborating with the research community to encourage researchers to voluntarily contribute additional data across various domains, from advanced mathematics and reasoning tasks to competitive programming and software engineering tasks.

2.1.2. Reward Function

Unlike DeepSeek-R1-Zero [2], our scale-up RL training employs a simple minimalist rule-based reward function that solely checks answer correctness, without any additional format rewards. Specifically, this reward function is designed to extract the content between '<answer>' and '</answer>' tags during training and compare it with the reference answer. To maintain clarity and simplicity in scale-up RL, we implement a binary reward scheme - awarding a reward of 1 for exact matches with the reference answer, and 0 for all other cases. To ensure rigorous and consitent assessment in evaluation, we adopt the widely-used Math-Verify[1] library and its usage as shown in Figure 3.

Surprisingly, we found that with our designed prompt, even unaligned base model can yield well-formatted responses in high probability. During early training stages, the base model can quickly learn and reinforce the correct format for reasoning and answering incentivized

[1] `https://github.com/huggingface/Math-Verify`

```
from math_verify import verify, parse
verify(parse(ground_truth), parse(model_output))
```

Figure 3: The code snippet for verifying the mathematical correctness of generated answers using the Math-Verify library.

by our simple rule-based reward function alone, as shown in Figure 4. More importantly, our preliminary experiments revealed that complicated reward functions were not only unnecessary, but could leave potential room for reward hacking.

2.1.3. RL Algorithm

We adopt the Proximal Policy Optimization (PPO) algorithm [4] as the RL algorithm for our scale-up training, unlike GRPO used in DeepSeek-R1-Zero. Specifically, for each question q (*i.e.,* prompt), the model generates a group of responses $\{o_1, o_2, ..., o_n\}$ and receives corresponding rewards $\{r_1, r_2, ..., r_n\}$ based on the rule-based reward function, where n represents the number of sampled trajectories (*i.e.,* rollout size per prompt). For each response o_i at time step t (*i.e.,* token t), let s_t denote the state at time t which comprises the question and all previously generated tokens, and a_t denote the token generated at that step. We compute the advantage estimation $\hat{A}_t$ for each token using Generalized Advantage Estimation (GAE) [11]. Generally, GAE provides a trade-off between bias and variance in the advantage estimation by combining multiple n-step advantage estimates through an exponentially weighted average controlled by the parameter λ. The advantage is computed as $\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + ... + (\gamma\lambda)^{T-t-1}\delta_{T-1}$, where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ is the TD (temporal difference) residual and γ is the discount factor that determines how much future rewards are valued relative to immediate rewards. The PPO algorithm updates the policy model parameters θ to maximize the expected reward and value model parameters ϕ to minimize the value loss by optimizing the following objective function:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{t,s_t,a_t\sim\pi_{\theta_{\text{old}}}}[\min(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_t, \text{clip}(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon)\hat{A}_t)], \tag{1}$$

$$\mathcal{J}_{\text{value}}(\phi) = \frac{1}{2}\mathbb{E}_{t,s_t,a_t\sim\pi_{\theta_{\text{old}}}}[(V_\phi(s_t) - R_t)^2], \tag{2}$$

where ϵ is the clipping parameter, π_θ is the current policy, $\pi_{\theta_{\text{old}}}$ is the old policy before the update, V_ϕ is the value function, and $R_t = \sum_{k=0}^{T-t}\gamma^k r_{t+k}$ is the discounted return. We instantiate the PPO algorithm with carefully tuned hyperparameters: GAE parameter $\lambda = 1.0$, discount factor $\gamma = 1.0$, and clipping parameter $\epsilon = 0.2$.

2.2. Key Findings

In this study, we explore best practices for reasoning-oriented RL training with an emphasis on stability and scalability. We conduct extensive experiments across the design space of Reasoner-Zero training. Here are the key findings from our experiments:

- **RL Algorithm Key Implementations**: Our empirical studies demonstrate that vanilla PPO provides a remarkably stable and robust training process across different model scales and training duration without requiring additional modifications. Through extensive experiments, we identified that the GAE parameters play a critical role in PPO for reasoning tasks. Specifically, setting $\lambda = 1.0$ and $\gamma = 1.0$, while typically considered suboptimal in traditional RL scenarios, achieves the ideal balance for scale-up RL training.

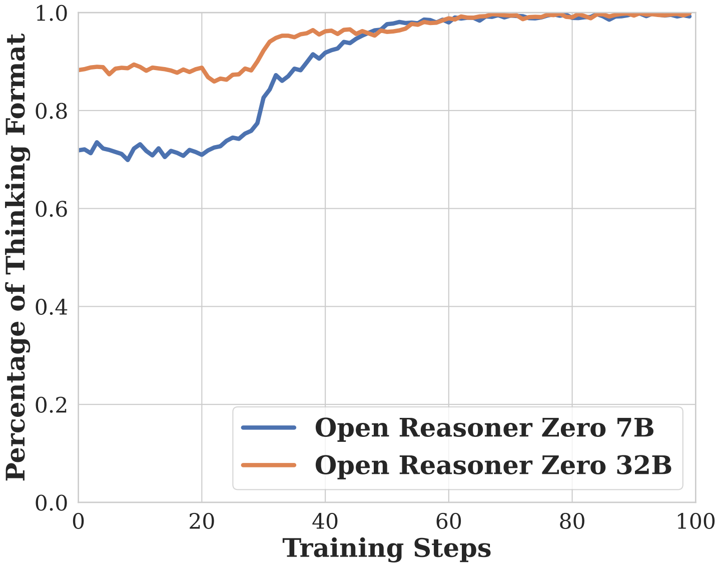

Figure 4: Percentage of responses following the reasoning format. Results demonstrate rapid adoption of structured reasoning patterns even by the base model using only a simple rule-based reward function. Our findings suggest that complicated reward functions are unnecessary for training Reasoner-Zero models.

- **Minimal Reward Function Design**: We show that a simple rule-based reward function is not only sufficient but optimal, as minimal design leaves no room for potential reward hacking. Notably, even unaligned base models quickly adpots to desired format, suggesting this is a straightforward task without requiring complex reward engineering.
- **Loss Function**: We achieve stable training without relying on any KL-based regularization techniques (*e.g.*, KL shaped rewards and loss), different from the de facto RLHF community [12] and Reasoner model [13, 2]. This also offers promising potential for further large-scaling RL.
- **Scale up Training Data**: We identify that scaling up data quantity and diversity is crucial for Reasoner-Zero training. While training on limited academic datasets like MATH leads to quick performance plateaus, our curated large-scale diverse dataset enables continuous scaling without signs of saturation on both training and test sets.

3. Experiments

In this section, we present comprehensive experimental results and analysis of our Open-Reasoner-Zero models. We begin by the training setup and hyperparameters, followed by an in-depth analysis of traning results, and ablation studies. We then investigate the preliminary results on two fronts: leveraging our trained reasoner model for distillation, and employing the Open-Reasoner-Zero training pipeline on the distilled model to further enhance its resasoning capabilities, following an approach similar to DeepSeek-R1 [2]. Finally, we discuss the evaluation results and provide a detailed analysis of the training process.

3.1. Training Details and Hyperparameters

We initialize both our policy and critic networks with Qwen-2.5 base models (7B and 32B variants), where value head is random initialized from $\mathcal{U}(-\sqrt{5}, \sqrt{5})$ with no bias term. The

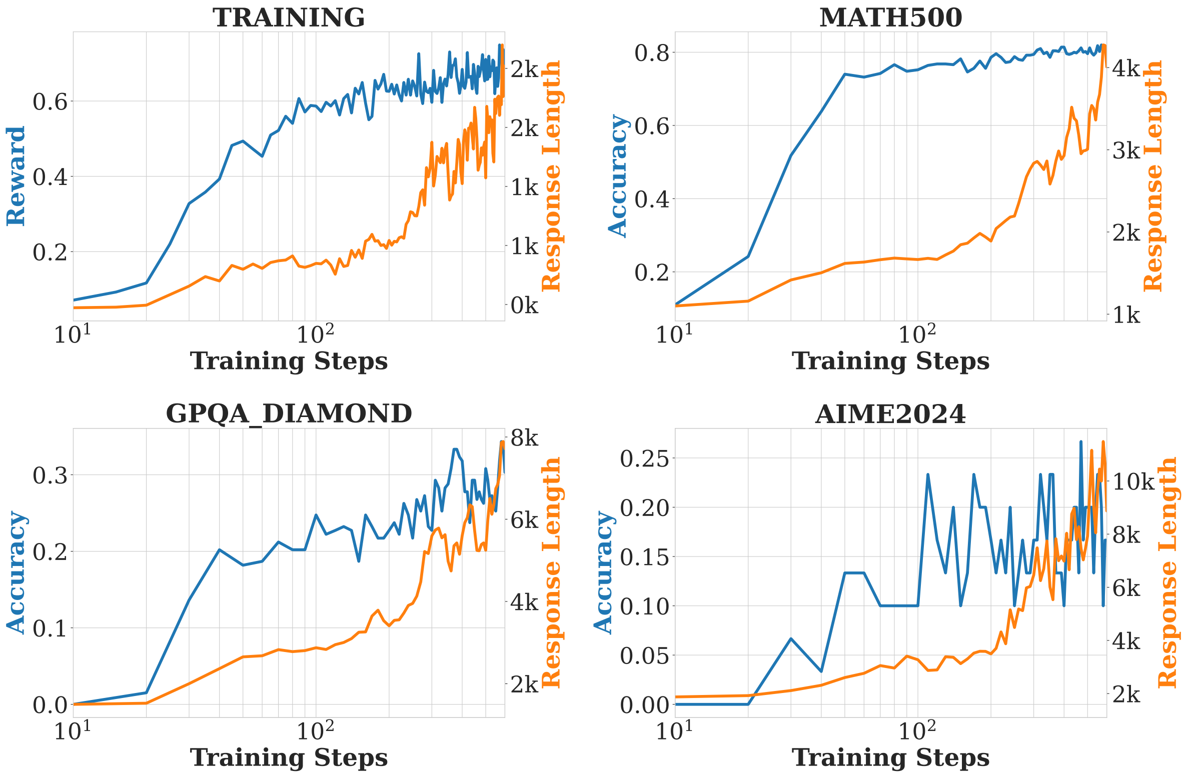

Figure 5: Comparison of training and evaluation reward and average response length for the Open-Reasoner-Zero 7B model. All of benchmarks experience a sudden increase in reward and response length at a certain point, a phenomenon like emergent behavior.

policy and critic do not share weights during training. For both policy and critic networks, we employ AdamW optimizer with $\beta = [0.9, 0.95]$ without weight decay. The learning rates are set to 1×10^{-6} and 5×10^{-6} for the policy and critic networks, respectively. The learning rate scheduler are both constant learning rate with linear warm-up of 50 optimizer steps. We employ sample packing during training.

Each generation step contains 128 unique prompts sampled from the dataset, and generating 64 responses per prompt with temperature and top-p both set to 1.0. To maintain training stability, we implement strict on-policy optimization for the policy network, where each generation corresponds to exactly one optimization step. The critic network, being less sensitive to off-policy updates, processes the experiences in 12 mini-batches, effectively performing 12 optimization steps per iteration. We apply batch level advantage normalization in the training.

Notably, our training process operates stably without any KL-related regularization terms or entropy bonuses, demonstrating that vanilla PPO can achieve stable training without these commonly used stabilization techniques.

To comprehensively evaluate our models' reasoning capabilities, we conduct experiments on diverse benchmarks spanning mathematical reasoning, coding, and general problem solving. These include GPQA DIAMOND [14], AIME2024, AIME2025 [15], MATH500 [16], and LIVECODEBENCH [17] datasets. For each benchmark, we report the average accuracy across 16 samples per question as our primary evaluation metric. Moreover, we also assess the models' general capabilities through evaluations on MMLU [18] and MMLU_PRO [19] benchmarks to provide a comprehensive understanding of their performance across diverse tasks.

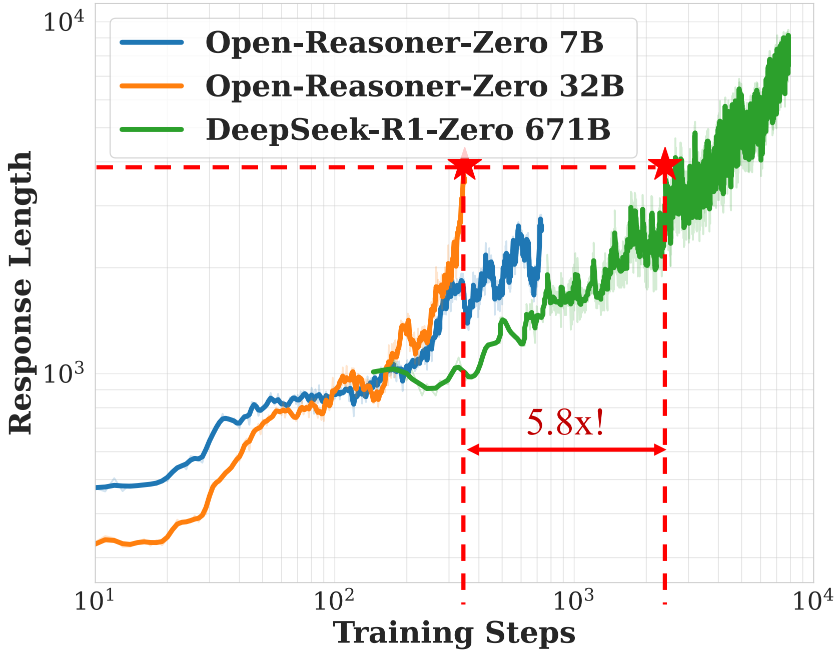

Figure 6: Training response length v.s training steps of our experiments. Our models demonstrate continuous improvements in response length throughout training, similar to DeepSeek-R1-Zero (671B MoE). Notably, our Open-Reasoner-Zero-32B model achieves comparable response lengths to DeepSeek-R1-Zero (671B MoE) with 5.8x fewer training steps. DeepSeek R1 Zero's response length data is estimated from Figure 3 in their paper.

3.2. Training Results

In this section, we present the key findings from our experimental training results. We evaluate the training process from multiple perspectives, including reward in training sets, average response lengths, and generation quality metrics. These metrics provide a holistic view of model performance and learning dynamics.

Training Curves. Figure 2 shows the training reward and average response length curves of our experiments for both Open-Reasoner-Zero 7B and 32B, while Figure 5 shows the reward/accuracy and average response length curves of our experienments for Open-Reasoner-Zero 7B on training and evaluation sets. The training reward curve and response length curve represent the average reward of the generated responses and the average length of the generated responses at each generation step, respectively. We observe consistent improvements in these metrics throughout training across both models and all benchmarks, with notable observations: Open-Reasoner-Zero exhibits an intriguing "step moment" phenomenon, where response metrics suddenly increase during training, revealing emergent reasoning capabilities.

Response Length Scale-up vs DeepSeek-R1-Zero. As shown in Figure 6, we observe a consistent increase in response length throughout training with no signs of saturation, mirroring the behavior seen in DeepSeek-R1-Zero. Notably, while both model size and training steps contribute to response length improvements, our Open-Reasoner-Zero-32B model achieves comparable response lengths to DeepSeek-R1-Zero (671B MoE) in just 1/5.8 of the training steps. This remarkable training efficiency demonstrates the effectiveness of our minimalist approach

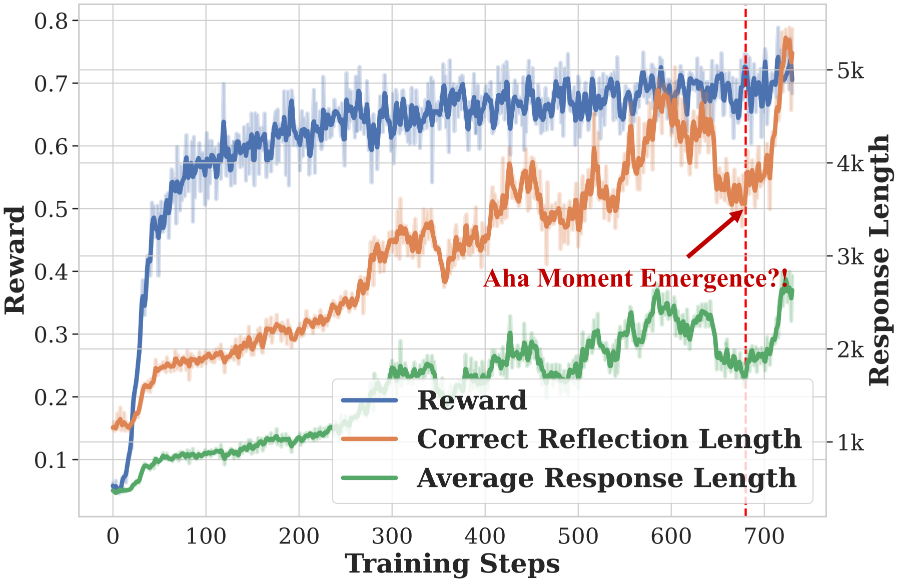

Figure 7: Reflection patterns in generated responses. The Average Correct Reflection Length consistently exceeds the Average Response Length throughout the training process. A particularly noteworthy phenomenon emerges around step 680, where we observe a simultaneous acceleration in three metrics: Reward in training set, Average Correct Reflection Length, and Average Response Length.

to large-scale RL training.

Quality Analysis. Here we provide some qualitative analysis of the generated responses from our Open-Reasoner-Zero models. To analyze the model's reflection capabilities and observe the Aha moment like DeepSeek-R1-Zero, we identify five representative reflection patterns (‘"wait,"‘, ‘"recheck"‘, ‘"retry"‘, ‘"alternatively,"‘, and ‘"however,"‘), following a methodology similar to [20]. We count the number of responses containing any of these patterns as ’reflection responses’, and identify the average correct reflection length (the length of responses containing reflection patterns that achieve correct answers). As shown in Figure 7, the average correct reflection length consistently exceeds the average response length throughout the training process, indicating that responses containing reflection patterns utilize more "thinking time" to achieve correct answers, similar to the test-time scale described in OpenAI o1. A particularly noteworthy phenomenon emerges around step 680, where we observe a simultaneous acceleration in three metrics: the reward, average correct reflection length, and average response length. Through manual inspection of model outputs before and after step 680, we observed qualitatively more pronounced reflection patterns in the latter responses. This emergent behavior warrants further investigation, and we are currently conducting detailed analyses to understand the underlying mechanisms of this phenomenon. For comprehensive quantitative and qualitative analyses, please refer to our detailed documentation available at Notion[2].

3.3. Ablation Study

We present ablation studies over key training strategies and hyperparameters that enable successful scaling of RL training directly from a base model. More comprehensive ablation

[2] Notion: Comprehensive quantitative and qualitative analyses.

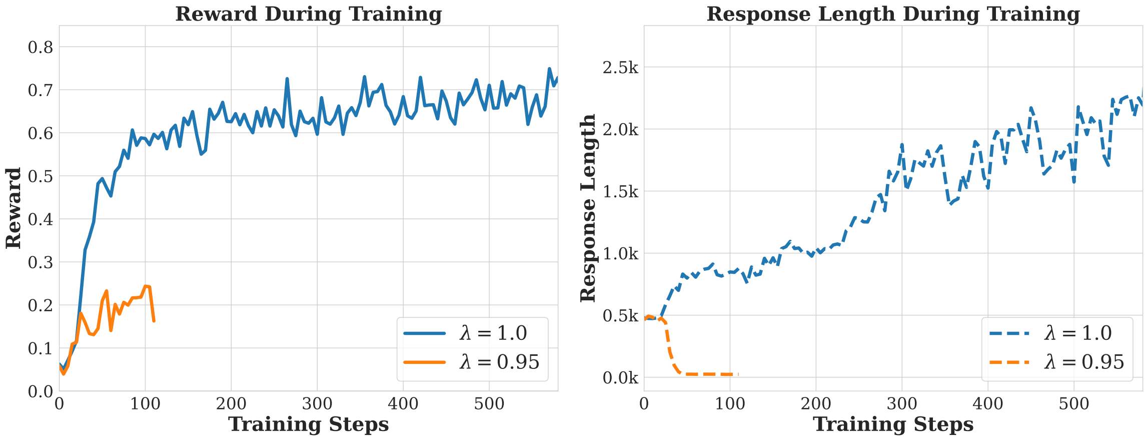

Figure 8: Comparison of different GAE λ values. GAE $\lambda = 1.0$ shows better stability and performance compared to $\lambda = 0.95$ for both training reward and response length.

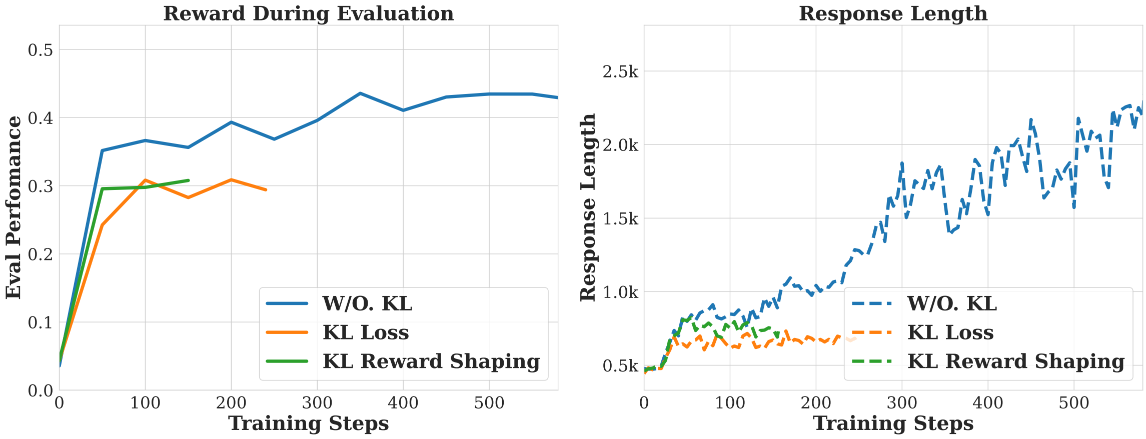

Figure 9: Comparisons to applying KL-related regularizations. Notably, training without KL constraints demonstrates superior average benchmark performance and length scaling property, compared to models trained with KL Loss and KL Penalty. Performance is evaluated on MATH500, AIME2024, and GPQA DIAMOND benchmarks using pass@1 metric.

studies are available in appendix.

GAE Analysis. We compare different GAE λ combinations. From the experimental results, we find that GAE λ=1.0 performs best in terms of training stability and final performance. Specifically, in the training reward, the GAE λ=1.0 curve rises quickly in the early stage and remains stable, finally converging to about 0.8; while the GAE λ=0.95 curve rises slowly and fluctuates. In the Response Length, the GAE Lambda=1.0 curve maintains a reasonable level during the training process; while the GAE λ=0.95 curve shows an unstable trend, leading to PPO learning instability. These results indicate that GAE λ=1.0 can better balance the training stability and generation quality. Moreover, discount factor (γ) set to 1.0 also has a significant impact on the scale-up RL training. Less than 1.0 will result in penalty for long-term reward, leading to a decrease in response length decrease and struggling to improve the final performance.

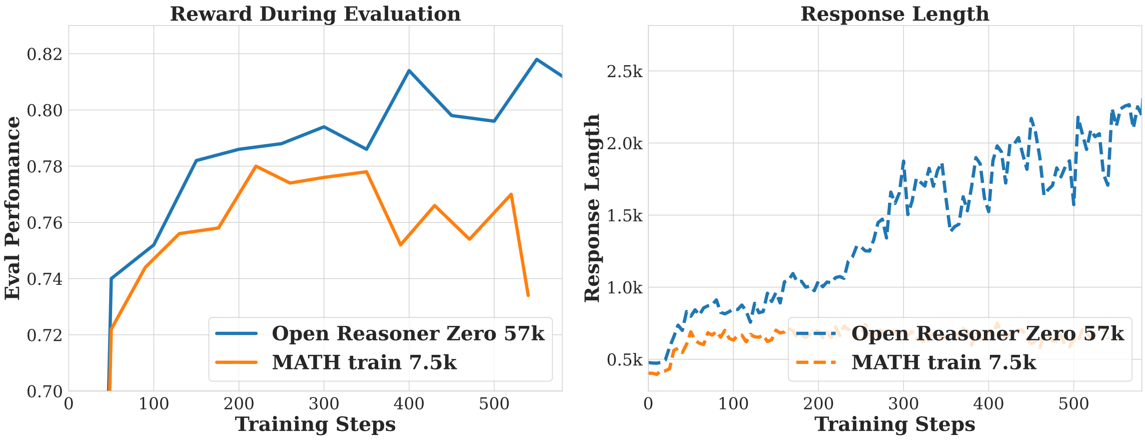

Figure 10: Data scale ablation study. Training data from math train 7.5k to Open-Reasoner-Zero 57k, we observe a consistent increase in both training reward and response length for training and evaluation set, indicating that data scale plays a crucial role in training performance. Performance is evaluated on MATH500 benchmark using pass@1 metric.

KL Constrains Analysis. We evaluate different combinations of KL Loss and KL Penalty for the Open-Reasoner-Zero 7B model, analyzing their impact on evaluation metrics and response length of the training set. This analysis is particularly important since reward shaping can introduce additional effects on training rewards. Our experimental results demonstrate that removing both KL Loss and KL Penalty yields optimal training stability and final performance. Both KL Loss and KL Penalty mechanisms not only slow down the training process but also consume computational resources that could be better utilized for reward optimization. Furthermore, eliminating these components reduces hyperparameter tuning burden and implementation complexity, which is crucial for scaling up RL training effectively.

Data Scale. We compare different data scales for training, ranging from 7.5k to 30k samples. As shown in Figure 10, larger data scales consistently lead to better performance in both training reward and response length for both training and evaluation sets. This result suggests that data scale plays a crucial role in training performance, and increasing the training data scale can effectively improve the model's reasoning capabilities. More comprehensive ablation studies including data quantity, quality, and diversity are available in the appendix.

3.4. Evaluation Results

In this section, we list our main experimental results. In our 32B experiments, Open-Reasoner-Zero demonstrates significant improvements in both training efficiency and model performance, as shown in Figure 1 11. The model achieves superior response length and accuracy across all benchmarks, notably outperforming DeepSeek-R1-Zero-Qwen2.5-32B on the GPQA DIAMOND benchmark while requiring only 1/30 of the training steps.

As shown in Figure 5, Open-Reasoner-Zero 7B model demonstrates interesting learning dynamics across different benchmarks. The accuracy generally shows a steady increase during training, while the response length exhibits more dramatic growth patterns. Notably, we observe an interesting emergent phenomenon during evaluation where both the reward and response length exhibit sudden, step-function-like increases at a certain point, which we refer to as the "step moment". This suggests that the models learn to master more detailed and comprehensive

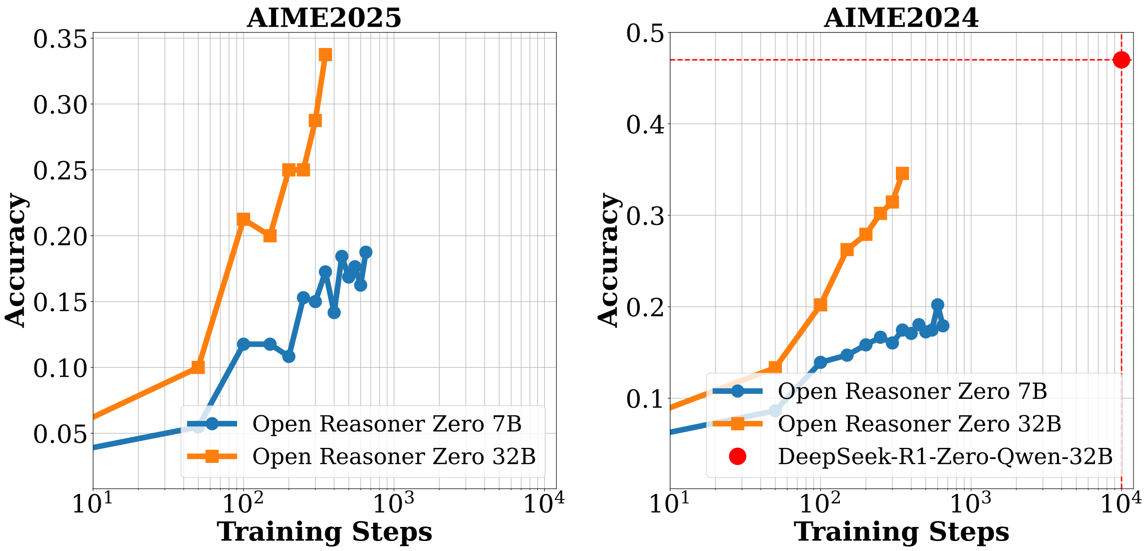

Figure 11: Evaluation performance of Open-Reasoner-Zero-{7B, 32B}. We report the average accuracy on the benchmark dataset for each question with 16 responses. We are continuing to scale up these RL settings until this preprint is released, as there is no sign of saturation yet.

Model	**MMLU**	**MMLU_PRO**
Qwen2.5-32B Base	83.3	55.1
Qwen2.5-32B Instruct	83.2	69.2
Open-Reasoner-Zero-32B	**85.1**	**72.7**

Table 2: Generalization performance of Open-Reasoner-Zero models on MMLU and MMLU_PRO benchmarks. Through solely scaling up RL training on reasoning-oriented tasks, Open-Reasoner-Zero achieves superior performance on both benchmarks, surpassing Qwen2.5 Instruct without any additional instruction tuning. This demonstrates the remarkable effectiveness of our training pipeline in enhancing model generalization capabilities.

reasoning capacities as training progresses. This pattern is particularly pronounced in GPQA DIAMOND and AIME2024, where response lengths increase substantially in the later training steps.

We then present the generalization capabilities of our models on knowledge and instruction following benchmarks, MMLU_PRO and IFEval. As shown in Table 2, Open-Reasoner-Zero 32B models demonstrate strong generalization capabilities significantly outperforming Qwen2.5 Instruct 32B on MMLU, MMLU_PRO through pure scale-up RL training on reasoning-oriented tasks, without any additional instruction tuning.

4. Conclusion and Discussions

In this work, we present Open-Reasoner-Zero (ORZ), the first open-source implementation of large-scale reasoning-oriented RL training, focusing on scalability, simplicity, and accessibility. Through extensive experiments, our best practice demonstrates that vanilla PPO with GAE ($\lambda = 1, \gamma = 1$) and straightforward rule-based reward function, without any KL regularization, is sufficient to scale up in both response length and benchmark performance on reasoning tasks with surprising generalization capabilities, achieving competitive results compared to

DeepSeek-R1-Zero pipeline. We provide comprehensive analysis of the key components and settings required for successful large-scale RL training, along with critical insights into scaling up PPO. By releasing our complete training resources, we aim to enable broader participation in this pivotal moment of AI development. We believe we are at an early stage of this new scaling trend, and we are excited to share our findings and experiences with the community.

Recall a bitter lesson from the past: the only thing that matters in the long run is what scales up effectively with increased computation and data. This fundamental insight continues to guide our research direction. In the future, we plan to further explore the following directions for continuously scaling up reasoning-oriented RL:

- **Data Scaling:** We will investigate how to effectively scale up by increasing the quantity, quality and diversity of training data. By open sourcing our own training dataset, we hope to encourage the research community to contribute and share more training data.
- **Model Scaling:** We will explore how to scale up model architectures to improve reasoning abilities. We will investigate how multimodal models can enable richer reasoning across different modalities, and how extended sequence lengths can allow for more complex multi-step reasoning.
- **Test Time Scaling:** We will explore how to scale up test time computation. We will investigate how multi-turn interactions can enhance contextual reasoning abilities, how value model can assess reasoning trajectories, and how multi-agent scenarios can lead to more sophisticated reasoning strategies.
- **Scenario Scaling:** We will explore how to scale up the complexity of reasoning for general scenarios. Our focus will be on generalizing reasoning capabilities to increasingly diverse tasks spanning creative writing, scientific discovery, and social interaction domains.

5. Acknowledgements

This work was supported by computing resources and infrastructure provided by StepFun. We are grateful for our colleagues from StepFun and Tsinghua University for their valuable feedback and contributions.

References

[1] OpenAI. Learning to reason with llms. `https://openai.com/index/learning-to-reason-with-llms/`, 2025.

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

[3] Richard Sutton. The bitter lesson. *Incomplete Ideas (blog)*, 13(1):38, 2019.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[5] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

[6] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. `https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2`, 2025. Notion Blog.

[7] Chengqi Lyu, Songyang Gao, Yuzhe Gu, Wenwei Zhang, Jianfei Gao, Kuikun Liu, Ziyi Wang, Shuaibin Li, Qian Zhao, Haian Huang, Weihan Cao, Jiangning Liu, Hongwei Liu, Junnan Liu, Songyang Zhang, Dahua Lin, and Kai Chen. Exploring the limit of outcome reward for learning mathematical reasoning, 2025.

[8] Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Weixun Wang, Dehao Zhang, and Yu Cao. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024.

[9] Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. `[https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf)`, 2024.

[10] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025.

[11] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

[13] Kimi Team. Kimi k1.5: Scaling reinforcement learning with llms. 2025.

[14] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[15] Mislav Balunović, Jasper Dekoninck, and Martin Vechev Ivo Petrov, Nikola Jovanović. Matharena: Evaluating llms on uncontaminated math competitions, February 2025.

[16] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021*, 2021.

[17] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

[18] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

[19] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems, NeurIPS 2024*, 2024.

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

[21] Loubna Ben Allal, Lewis Tunstall, Anton Lozhkov, Elie Bakouch, Guilherme Penedo, and Gabriel Martín Blázquez Hynek Kydlicek. Open r1: Evaluating llms on uncontaminated math competitions, February 2025.

[22] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[23] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[24] Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *arXiv preprint arXiv:2412.09413*, 2024.

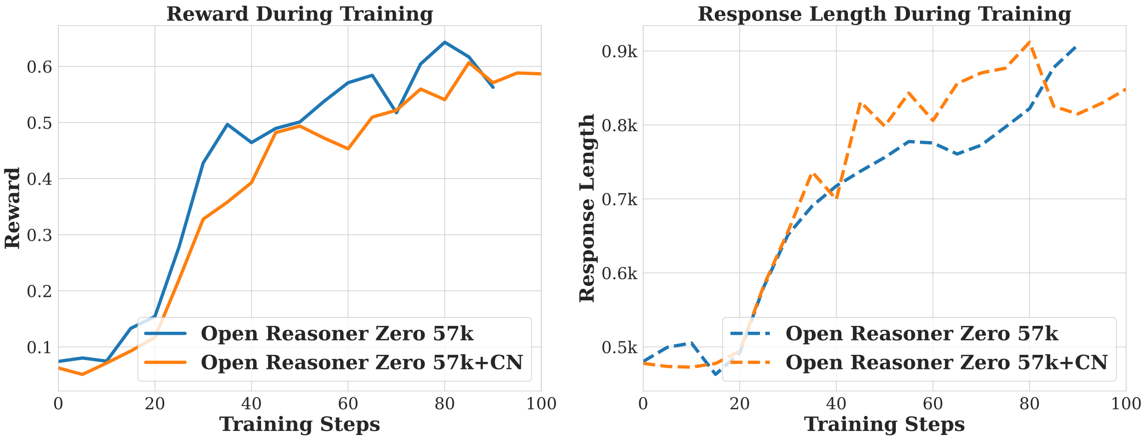

Figure 12: Data Curation Ablation Study. CN represents Chinese data and EN represents English data. Our results demonstrate that the English-only dataset yields superior training stability and final model performance.

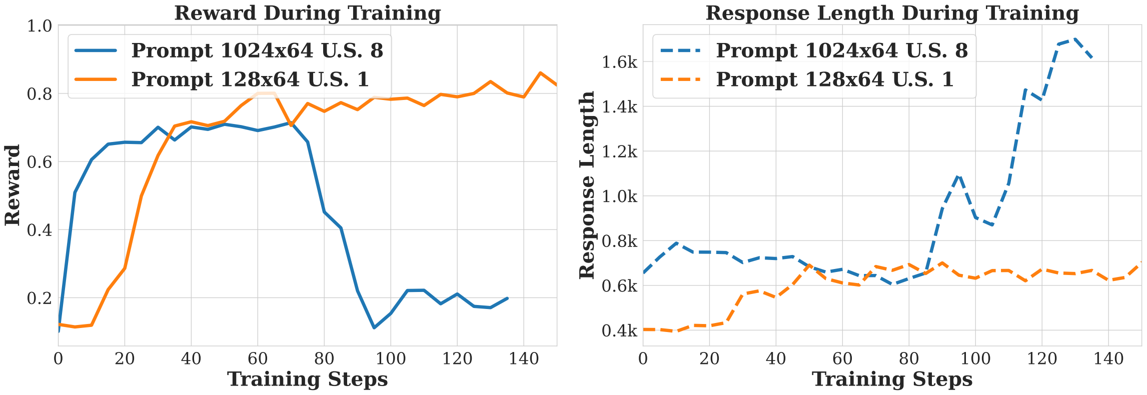

Figure 13: Comparison of different Prompt, Rollout, Batch Size combinations. U.S. represents Update steps of model parameters in each generation steps. On policy update for each sample collection performance better than off policy update on both training reward and response length.

A. More Ablation Studies

In this section, we present additional ablation studies conducted during our exploration of scaling up RL training. Notably, our ablation experiments were conducted during our efforts to scale up RL training, with some experiment employing different basic training strategies to explore various aspects of the training process.

More Ablations over Data Curation. Based on our analysis of data quality issues, we conduct comprehensive ablation studies to evaluate how different data curation strategies affect model training stability and performance. Motivated by OpenR1's finding [21] that SFT performance degradation on Chinese subsets was due to simpler question patterns, we experiment with two data curation approaches: using English-only data versus using both English and Chinese data. Our results demonstrate that the English-only dataset yields superior training stability and final model performance. While most of our experiments utilize the full dataset including Chinese

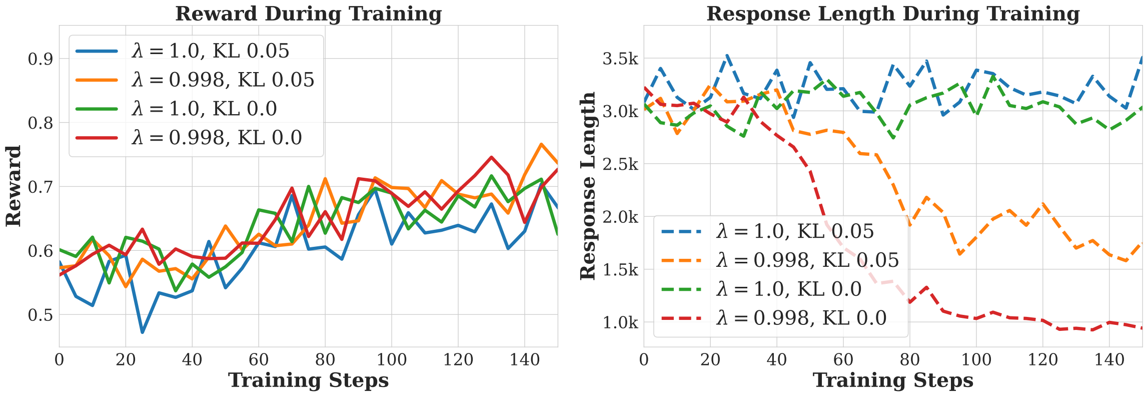

Figure 14: Comparison of different KL Loss, KL Penalty, and GAE λ values.

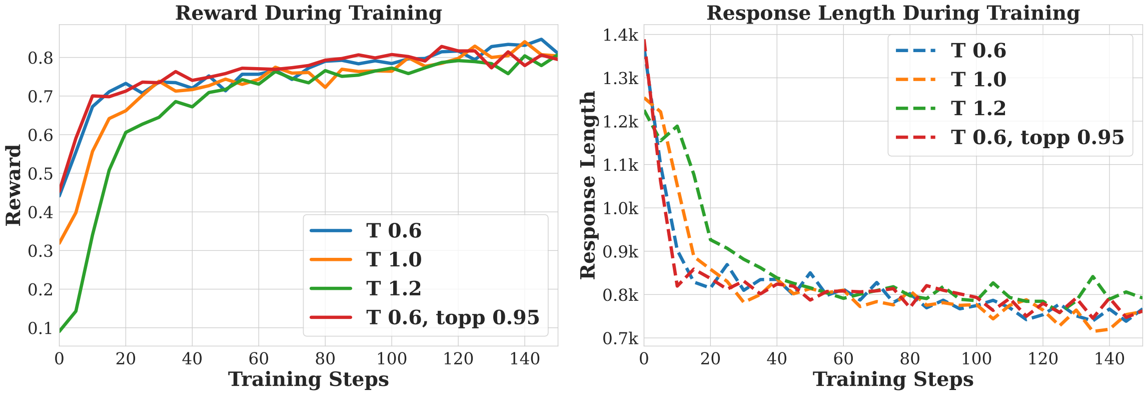

Figure 15: Comparison of different sampling strategies. T represents temperature and topp represents top-p sampling.

content, we make the final Open-Reasoner-Zero 57k dataset publicly available as it provides broader applicability across diverse tasks.

Sampling Strategy. We compare different sampling strategies, including temperature T=0.6, 1.0, 1.2 and T=0.6 topp=0.95, with different model initialization, training dataset and training hyperparameters compared to our main settings. Specifically, we use Qwen2.5-Math-7B[22] as initialization here and use MATH train set as training data. As for the training hyperparameters, here we adopt 1024 unique prompts and 8 responses for each prompt in each generation. We process the experiences into at most 8 mini-batches for both policy and critic training. From the experimental results, we find that most basic sampling strategy works well compared to changing temperature or topp. Considering the scalability of training recipe, we finally opt for the most basic sampling strategy that T and topp both equal to 1.0.

More Ablations over KL Loss & KL Penalty & GAE λ Analysis. We analyze the GAE Lambda and KL Loss for the LLaMA3.1-Instruct-SFT model [23]. This model is trained on STILL-2 data [24]. From the experimental results, we find that the combination of GAE λ=1.0 and no KL Loss performs best in terms of training stability and final performance. As shown in the figure, this configuration shows the most stable performance in both training reward and response length. Additionally, our early experiments also found that introducing KL Penalty (similar to reward shaping in RLHF) significantly affected the reasoning ability of the model. Based on these findings and for scalability, we finally chose the training strategy of not using KL constrains.