Reinforcement Learning on Pre-Training Data

Siheng Li[1,3,*,†], Kejiao Li[1,†], Zenan Xu[1,†], Guanhua Huang[1], Evander Yang[1], Kun Li[1,3,*], Haoyuan Wu[1], Jiajia Wu[1], Zihao Zheng[1], Chenchen Zhang[1], Kun Shi[1], Kyrierl Deng[1], Qi Yi[1], Ruibin Xiong[1], Tingqiang Xu[1,*], Yuhao Jiang[1], Jianfeng Yan[1], Yuyuan Zeng[1], Guanghui Xu[1], Jinbao Xue[2], Zhijiang Xu[2], Zheng Fang[2], Shuai Li[2], Qibin Liu[2], Xiaoxue Li[2], Zhuoyu Li[2], Yangyu Tao[2], Fei Gao[2], Cheng Jiang[2], Bo Chao Wang[2], Kai Liu[2], Jianchen Zhu[2], Wai Lam[3], Wayyt Wang[1,‡], Bo Zhou[1,‡], Di Wang[1]

[1]LLM Department, Tencent **[2]HunYuan Infra Team**
[3]The Chinese University of Hong Kong
✉ chaysezhou@tencent.com

Abstract

The growing disparity between the exponential scaling of computational resources and the finite growth of high-quality text data now constrains conventional scaling approaches for large language models (LLMs). To address this challenge, we introduce Reinforcement Learning on Pre-Training data (RLPT), a new training-time scaling paradigm for optimizing LLMs. In contrast to prior approaches that scale training primarily through supervised learning, RLPT enables the policy to autonomously explore meaningful trajectories to learn from pre-training data and improve its capability through reinforcement learning (RL). While existing RL strategies such as reinforcement learning from human feedback (RLHF) and reinforcement learning with verifiable rewards (RLVR) rely on human annotation for reward construction, RLPT eliminates this dependency by deriving reward signals directly from pre-training data. Specifically, it adopts a next-segment reasoning objective, rewarding the policy for accurately predicting subsequent text segments conditioned on the preceding context. This formulation allows RL to be scaled on pre-training data, encouraging the exploration of richer trajectories across broader contexts and thereby fostering more generalizable reasoning skills. Extensive experiments on both general-domain and mathematical reasoning benchmarks across multiple models validate the effectiveness of RLPT. For example, when applied to Qwen3-4B-Base, RLPT yields absolute improvements of 3.0, 5.1, 8.1, 6.0, 6.6, and 5.3 on MMLU, MMLU-Pro, GPQA-Diamond, KOR-Bench, AIME24, and AIME25, respectively. The results further demonstrate favorable scaling behavior, suggesting strong potential for continued gains with more compute. In addition, RLPT provides a solid foundation, extending the reasoning boundaries of LLMs and enhancing RLVR performance.

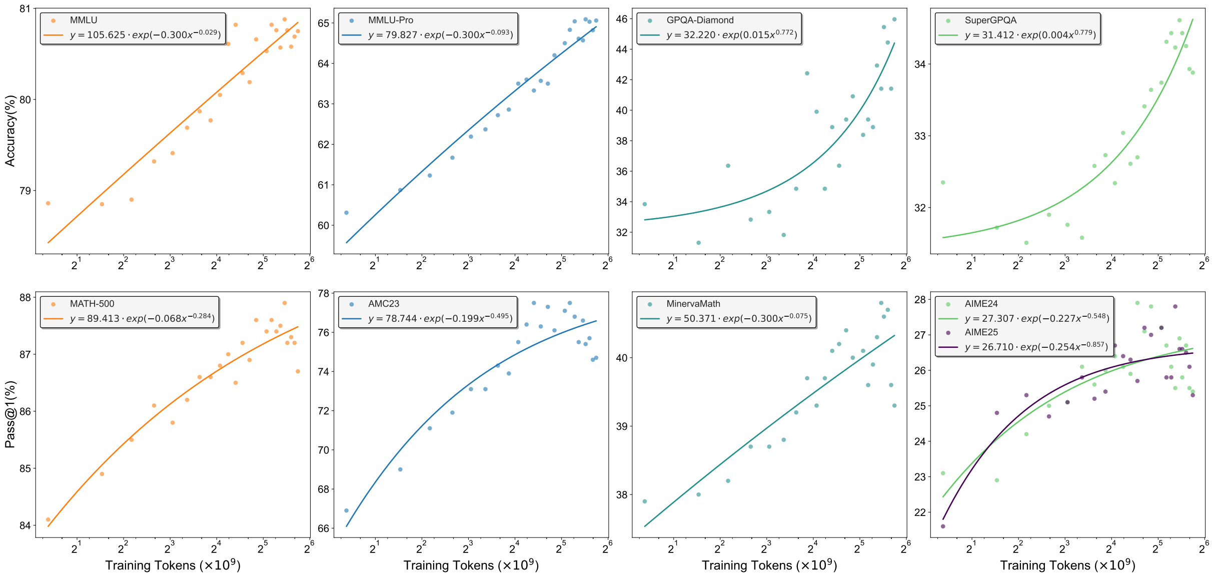

Figure 1: Scaling law of RLPT performance on various benchmarks with respect to training tokens.

[*] Work completed during an internship at Tencent.

[†] The first three authors contributed equally to this work.

[‡] Project Lead.

1 Introduction

Large language models (LLMs) have achieved remarkable success across diverse domains, including human-aligned conversational assistants (Bai et al., 2022; Ouyang et al., 2022) and autonomous AI agents (Team et al., 2025a). A central driver of this progress has been the scaling of computational resources during training, realized through the simultaneous expansion of both data and model parameters. For instance, training corpora have grown from billions of tokens in BERT (Devlin et al., 2019) to trillions in Llama (Touvron et al., 2023; Grattafiori et al., 2024), while model sizes have scaled from millions of parameters in BERT (Devlin et al., 2019) to the trillion-parameter level in Kimi K2 (Team et al., 2025a). However, parameter scaling requires increasingly demanding infrastructure and results in prohibitive inference costs, whereas data scaling is constrained by the scarcity of high-quality web corpora (Villalobos et al., 2024; Muennighoff et al., 2023; Ruan et al., 2025).

In this paper, we propose a new scaling paradigm RLPT[1] to optimize LLMs through reinforcement learning (RL) on pre-training data. In contrast to prior scaling approaches that primarily rely on supervised learning, RLPT allocates training compute to enable the policy to autonomously explore meaningful reasoning trajectories to learn from pre-training data and improve its overall capabilities through reinforcement learning (RL). This paradigm offers two main advantages. First, it enables reasoning for learning: rather than directly learning token by token, the model generates intermediate reasoning content that can uncover the latent thought process underlying data construction, augment the original data, and support more data-efficient learning (Ruan et al., 2025). Second, RL leverages self-explored trajectories for training, maintains proximity to the original policy distribution, and thereby fosters stronger generalization capabilities (Chu et al., 2025; Lai et al., 2025; Shenfeld et al., 2025). However, directly scaling RL also introduces new challenges, since existing frameworks such as reinforcement learning from human feedback (RLHF) (Bai et al., 2022) and reinforcement learning with verifiable rewards (RLVR) (Guo et al., 2025) still rely heavily on human annotation, which constrains their scalability on pre-training data.

To address this challenge, we propose a novel next-segment reasoning objective that can obtaining meaningful self-supervised reward from unlabeled internat data. To be more specifically, the model is first required to predict the subsequent segment of text, and then the reward signal is derived by evaluating the semantic consistency between the predicted segment and the real segment using a generative reward model. Based on different prediction segment configurations, we propose two tasks with distinct effect. The first requires the model to predict a complete subsequent sentence given the preceding context, which we term the Autoregressive Segment Reasoning (ASR) task. The second involves a context with masked tokens in the middle, where the model must leverage both preceding and following context to infer a continuous span of masked tokens, which we designate as the Middle Segment Reasoning (MSR) task. During training, we interleave ASR and MSR task to simultaneously optimize the model's autoregressive generation capabilities as well as the in-context understanding abilities.

We evaluate RLPT across both general-domain and mathematical reasoning tasks using multiple models. Experimental results demonstrate that RLPT delivers consistent and substantial improvements in both settings. For example, when applied to Qwen3-4B-Base, RLPT achieves absolute gains of 3.0, 5.1, 8.1, and 6.0 on MMLU (Hendrycks et al., 2021a), MMLU-Pro (Wang et al., 2024), GPQA-Diamond (Rein et al., 2024), and KOR-Bench (Ma et al., 2024), respectively, together with improvements of 6.6 and 5.3 in Pass@1 on AIME24 and AIME25 (MAA, a). Comparable gains are also observed on Llama3.2-3B-Base and Qwen3-8B-Base, with detailed results provided in Sec. 4.3. Beyond standalone performance, RLPT also strengthens the reasoning capability of LLMs. When serving as the foundation for RLVR, it yields additional improvements of 2.3 and 1.3 in Pass@1, and 3.7 and 2.0 in Pass@8, on AIME24 and AIME25 with Qwen3-4B-Base, respectively. We further analyze the scaling behavior of RLPT, showing that downstream performance empirically follows a scaling law with training compute (Fig. 1), highlighting its potential for continued progress with increased compute. In addition to quantitative results, qualitative analysis of reasoning trajectories reveals diverse reasoning strategies, providing insight into the effectiveness of RLPT. Finally, we distill practical design lessons from RLPT to inform future research in this direction.

Our contributions can be summarized in three aspects:

- We propose RLPT, a method that scales RL on pre-training data. To remove the reliance on human annotation, we design a next-segment reasoning objective, consisting of ASR and MSR tasks, which reward LLMs for correctly predicting the ground-truth next segment given the preceding context.
- Extensive experiments on general-domain and mathematical reasoning tasks across multiple models show that RLPT substantially improves performance and exhibits a favorable scaling trend, empirically establishing a scaling law in benchmark performance as compute increases, indicating strong potential for continued gains.
- Results further demonstrate that RLPT provides a strong foundation for subsequent RLVR, extending the reasoning boundaries of LLMs and boosting performance on mathematical reasoning benchmarks.

[1] RLPT stands for Reinforcement Learning on Pre-Training data.

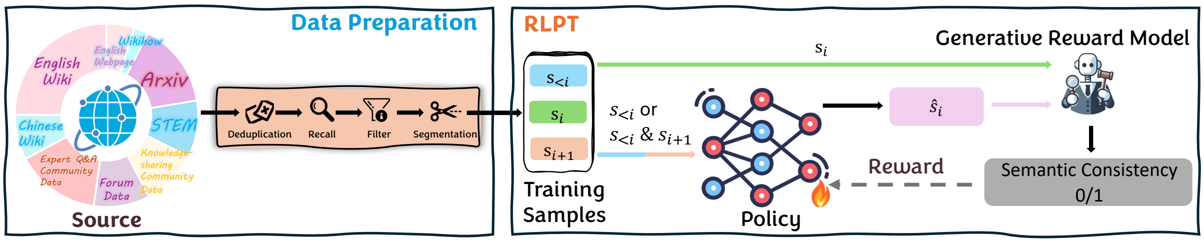

Figure 2: Overview of RLPT. Raw data from internet corpora is processed into training samples of the form $(s_{<i}, s_i, s_{i+1})$. During the reinforcement pre-training stage, the policy LLM predicts $\hat{s}_i$ conditioned on $s_{<i}$ (ASR) or on $(s_{<i}, s_{i+1})$ (MSR). The prediction is then compared with s_i to compute the reward.

2 Preliminary

In this section, we briefly review reinforcement learning (RL) and the supervised learning paradigm of next-token prediction in training large language models (LLMs), as these serve as the foundation of RLPT.

2.1 Reinforcement Learning in Large Language Models

Reinforcement learning (RL) has become an essential component for improving LLMs. Formally,

$$\mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}[q \sim D_q, o \sim \pi_\theta(\cdot \mid q)][r(o)], \tag{1}$$

where $r(o)$ denotes the reward assigned to output o. In reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Bai et al., 2022), rewards are provided by a neural reward model trained on human-annotated preference pairs. More recently, reinforcement learning with verifiable rewards (RLVR) employs rule-based functions that compare model outputs against reference answers (Guo et al., 2025; Zeng et al., 2025). Optimizing Eq. 1 encourages the model to reinforce behaviors associated with higher rewards while suppressing those linked to lower rewards. In practice, this objective is typically optimized using policy gradient algorithms such as PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024). Despite their effectiveness, both RLHF and RLVR face scalability challenges due to their reliance on human annotations.

2.2 Next-Token Prediction

Next-token prediction (NTP) is the fundamental training objective of modern LLMs. Formally,

$$\mathcal{J}_{\text{NTP}}(\theta) = \mathbb{E}[x \sim \mathcal{D}_x] - \frac{1}{|x|} \sum_{i=1}^{|x|} \log \pi_\theta(x_i \mid x_{<i}), \tag{2}$$

where x is a token sequence and $|x|$ its length. Pre-training and post-training based on NTP constitute the mainstream optimization paradigm for LLMs, yielding remarkable success across diverse applications. Nevertheless, recent studies suggest that supervised fine-tuning (SFT) under the NTP paradigm often promotes surface-level memorization rather than fostering the deeper generalization capabilities achievable with RL (Chu et al., 2025; Lai et al., 2025; Shenfeld et al., 2025).

3 Reinforcement Learning on Pre-Training Data

To address the limitations of scalability and generalization, we propose Reinforcement Learning on Pre-Training data (RLPT). In this framework, next-segment reasoning serves as the reinforcement learning (RL) objective, where the subsequent segment in unlabeled text acts as the ground truth. This self-supervised objective removes the reliance on human annotation and enables RL to scale directly on large pre-training corpora. An overview of RLPT is shown in Fig. 2.

3.1 Data Preparation

We construct a corpus for RLPT by aggregating web text from diverse sources such as Wikipedia, arXiv, and threaded conversation data. To ensure data quality and compliance, we apply a multi-stage preprocessing pipeline consisting of: (i) MinHash-based near-deduplication, (ii) detection and masking of personally identifiable information (PII), and (iii) contamination removal with respect to all development and evaluation sets. Given the inherent noise in web corpora, we further implement a rigorous filtering procedure that integrates both rule-based and model-based methods. The rule-based stage eliminates content that is clearly unsuitable for language model training, whereas the model-based stage employs an instruction-tuned language model to perform more fine-grained quality assessments. Furthermore, we curated high-quality QA data from the annealing dataset Team et al. (2025b) for mathematical reasoning tasks to enhance the model's reasoning ability.

3.2 Next-Segment Reasoning

Given a text t from the pre-training data, we divide it into a sequence of contiguous segments $t = [s_1, s_2, \dots, s_n]$, where each s_i corresponds to a semantically coherent unit such as a phrase, a complete sentence, or a reasoning step. We then construct a dataset

$$\mathcal{D}s = (s_{<i}, , s_i, s_{i+1}) \mid i = 2, \dots, n-1, \tag{3}$$

where $s_{<i} = [s_1, s_2, \dots, s_{i-1}]$ denotes the context, s_i is the target segment, and s_{i+1} is its subsequent segment. Based on this formulation, we introduce two segment-level training objectives that capture richer semantics than token-level prediction. Inspired by next-token prediction (NTP), we propose Autoregressive Segment Reasoning (ASR), which trains the policy to predict s_i from $s_{<i}$, aligning with the autoregressive generation process of modern LLMs. To further enable the model to leverage broader contextual information, we introduce Middle Segment Reasoning (MSR), which trains the policy to predict s_i from both $s_{<i}$ and s_{i+1}. This resembles masked language modeling (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020) and is particularly useful for tasks such as code completion. During training, we interleave ASR and MSR by designing different prompts and extracting the predicted segment between special tags in the output. The prompt for the ASR task is illustrated below.

```
Complete the text provided under ### Context by predicting the next most probable sentence.

Please reason step by step to determine the best possible continuation, and then enclose your final
answer within <|startofprediction|> and <|endofprediction|> tags.

### Context

{context}
```

Similarly, the prompt for the MSR task is presented as follows

```
## Text Material ##:
{prompt}

<MASK>

{next_step}

## Task ##:
Fill in the <MASK> section of the material with appropriate sentences or a solution step.

Carefully reason step by step to determine the most suitable completion.
Finally, provide your best prediction for the <MASK> section.
Enclose your final answer for the <MASK> part within <|startofprediction|> and <|endofprediction|>.
```

The reward is defined as the semantic consistency between the predicted and reference segments, evaluated by a generative reward model G_{rm}. This model assesses whether the two segments convey equivalent content while allowing for linguistic variation. In practice, we find that directly comparing the predicted segment with the ground-truth next segment is overly strict, since the model may generate outputs that span multiple subsequent segments. To address this issue, we provide G_{rm} with several subsequent segments as reference and instruct it to verify whether the predicted segment is a valid prefix of the reference content. The prompt for G_{rm} is shown below.

```
## Task
Given a Predicted sentence and a Reference paragraph, determine whether the Predicted text is a prefix (
initial segment) of the Reference paragraph, and whether it expresses exactly the same semantic content
as the corresponding prefix of the Reference.
The Predicted text does not need to match the prefix of the Reference word-for-word, but it must convey
the same meaning.

Reference:
{reference}

Predicted:
{predicted}

## Scoring Rules

If the Predicted text semantically matches the prefix of the Reference, assign a score of 1.
If the Predicted text does not semantically match the prefix of the Reference, assign a score of 0.
When making your judgment, focus primarily on semantic equivalence, not on exact wording.

Only output the score on a single line; do not provide any explanatory text or additional content.
Output format (choose one):

Score: 0
or
Score: 1
```

Given a predicted segment $\hat{s}_i$ extracted from the model output o, the reward is specified as

$$r(o, s_i) = \begin{cases} 1 & \text{if } G_{rm}(\hat{s}_i, s_i) = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The training objective of RLPT is defined as

$$\begin{aligned} \mathcal{J}_{\text{SRPT}}(\theta) = {} & \mathbb{E}_{ASR}[(s_{<i}, s_i) \sim \mathcal{D}_s,\, o \sim \pi_\theta(\cdot \mid s_{<i})][r(o, s_i)] \\ & + \lambda\, \mathbb{E}_{MSR}[(s_{<i}, s_i, s_{i+1}) \sim \mathcal{D}_s,\, o \sim \pi_\theta(\cdot \mid s_{<i}, s_{i+1})][r(o, s_i)], \end{aligned} \tag{5}$$

where $\lambda \in (0, 1)$ is a hyperparameter that balances the contributions of ASR and MSR terms, and may be adjusted depending on the requirements of specific downstream applications.

3.3 Training Details

Cold-Start. RLPT can be applied to a base model after next-token pre-training, but it requires a minimum level of instruction-following ability to initiate next-segment reasoning. To satisfy this requirement, we introduce a cold-start phase consisting of supervised fine-tuning on instruction-following data.

Next-Segment Reasoning. In this work, we define a segment unit as a sentence by default. We also conducted preliminary studies with alternative segmentation units, such as employing LLMs to extract integrated atomic steps from text, but these approaches did not yield clear improvements over sentence-level segmentation. Therefore, we adopt sentence segmentation as the default setting in our experiments and leave the exploration of other strategies for future work. For sentence segmentation, we use the NLTK toolkit (Bird, 2006), filtering out sentences that are too short. Each remaining sentence is then treated as a target for RL under the next-segment reasoning objective.

4 Experiments

4.1 Experimental Setup

Experiments are conducted on Llama3 models (Grattafiori et al., 2024) and Qwen3 models (Yang et al., 2025). In the cold-start supervised fine-tuning (SFT) stage, we use a batch size of 1024, a learning rate of 2×10^{-5} with a cosine scheduler, and train for 3 epochs. For next-segment reasoning, we adopt a batch size of 512, a maximum response length of 8192, and a constant learning rate of 1×10^{-6}. For each prompt, we sample 8 outputs with a temperature of 1.0, and optimization is performed using on-policy GRPO (Shao et al., 2024) without KL regularization. In the mathematical reasoning domain, we further conduct RLVR experiments, evaluating its performance when built on RLPT, with RLVR configured using the same hyperparameters as the next-segment reasoning task.

4.2 Evaluation Metric

We evaluate model performance on both general-domain and mathematical reasoning tasks. For the general domain, we use benchmarks including MMLU (Hendrycks et al., 2021a), MMLU-Pro (Wang et al., 2024), GPQA-Diamond (Rein et al., 2024), SuperGPQA (Du et al., 2025), KOR-Bench (Ma et al., 2024), and OlympiadBench (He et al., 2024), reporting accuracy as the evaluation metric. For mathematical reasoning, we evaluate on MATH-500 (Hendrycks et al., 2021b), AMC23 (MAA, b), Minerva Math (Lewkowycz et al., 2022), and AIME (MAA, a), using the Pass@k metric, which measures the probability that at least one correct solution appears among k independent attempts. We adopt the unbiased estimator of Pass@k (Chen et al., 2021):

$$\text{Pass@}k = \mathbb{E}_{x \sim \mathcal{D}} \left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}} \right], \tag{6}$$

where n is the number of sampled responses per prompt and c is the number of correct responses. We sample $n = 64$ responses with temperature 0.6 and top-p 0.95, and report Pass@1 and Pass@8. The maximum generation length is set to 32,768 tokens. Correctness in mathematical reasoning is evaluated using Math-Verify[2].

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

4.3 Experimental Results

Training	MMLU	MMLU-Pro	GPQA-Diamond	SuperGPQA	KOR-Bench	OlympiadBench
			Llama-3.2-3B-Base			
Base	4.2	21.3	3.5	7.7	3.1	1.7
+ Cold-Start	59.4	34.7	16.7	15.8	39.1	14.4
+ RLPT	59.4	**36.2**	**28.3**	**19.2**	**39.4**	**15.9**
			Qwen3-4B-Base			
Base	30.6	16.0	17.7	25.4	3.7	35.0
+ Cold-Start	77.8	59.7	31.3	32.3	50.7	51.7
+ RLPT	**80.8**	**64.8**	**39.4**	**34.3**	**56.7**	**52.7**
			Qwen3-8B-Base			
Base	58.9	47.0	27.8	28.5	40.6	38.8
+ Cold-Start	81.6	64.9	45.5	37.8	55.1	57.6
+ RLPT	**83.0**	**68.3**	**47.5**	**40.1**	**55.7**	**59.7**

Table 1: performance on general-domain tasks across different models, with the best results highlighted.

Training	Pass@1					Pass@8				
	MATH	**AMC23**	**Minerva**	**AIME24**	**AIME25**	**MATH**	**AMC23**	**Minerva**	**AIME24**	**AIME25**
Base	39.8	24.7	17.7	7.3	4.5	79.9	65.6	41.0	24.3	21.6
+ Cold-Start	83.6	65.9	38.2	20.6	21.9	95.0	91.8	54.1	40.3	39.5
+ RLPT	87.4	77.1	40.1	27.2	27.2	95.3	92.1	54.8	45.3	40.9
+ RLVR	89.1	76.3	41.6	27.6	27.7	96.7	**94.3**	55.8	49.8	41.6
+ RLPT+ RLVR	**90.6**	**79.7**	**42.1**	**29.9**	**29.0**	**96.8**	93.5	**56.8**	**53.5**	**43.6**

Table 2: Performance on mathematical reasoning benchmarks based on the Qwen3-4B-Base model with 64 samples per prompt, the best performance are highlighted.

General Domain. The performance on general-domain tasks is summarized in Tab. 1, where RLPT delivers substantial and consistent gains across all benchmarks and models. In particular, when applied to Qwen3-4B-Base, it achieves absolute improvements of 3.0, 5.1, 8.1, 2.0, and 6.0 on MMLU, MMLU-Pro, GPQA-Diamond, SuperGPQA, and KOR-Bench, respectively. With Qwen3-8B-Base, the improvements are 1.4, 3.4, 2.0, 2.3, and 2.1 on MMLU, MMLU-Pro, GPQA-Diamond, SuperGPQA, and OlympiadBench, respectively. Furthermore, results on Llama-3.2-3B-Base confirm the generalizability of RLPT across different model families, with absolute improvements of 1.5, 11.6, and 3.4 on MMLU-Pro, GPQA-Diamond, and SuperGPQA, respectively. Since these benchmarks span diverse domains including STEM, law, economics, and health, the results demonstrate that RLPT effectively leverages the extensive knowledge contained in large-scale pre-training corpora.

Mathematical Reasoning. As shown in Tab. 2, RLPT yields substantial gains in mathematical reasoning, improving performance on both Pass@1 and Pass@8. On the challenging AIME24 and AIME25 benchmarks, RLPT achieves absolute improvements of 6.6 and 5.3 on Pass@1, and 5.0 and 1.4 on Pass@8, respectively. These improvements indicate that RLPT is effective in unlocking the reasoning boundary, thereby providing a strong basis for subsequent RLVR training. Indeed, when RLPT is used as the initialization for RLVR, it further boosts performance, with absolute gains of 2.3 (AIME24) and 1.3 (AIME25) on Pass@1, and 3.7 (AIME24) and 2.0 (AIME25) on Pass@8. This demonstrates that RLPT enhances both exploitation and exploration, which are typically considered competing objectives.

4.4 Analysis

Scaling Properties. As shown in Fig. 1, the performance of RLPT on various benchmarks follows a power-law decay with respect to the number of training tokens, suggesting potential for further gains through scaling compute. We also report the scaling trend when RLPT serves as the foundation for RLVR in Fig. 3. In this setting, RLPT provides a strong initialization, yielding consistent improvements throughout training. Notably, both Pass@1 and Pass@8 improve, indicating that the gains from RLPT do not come at the expense of exploration capability, which

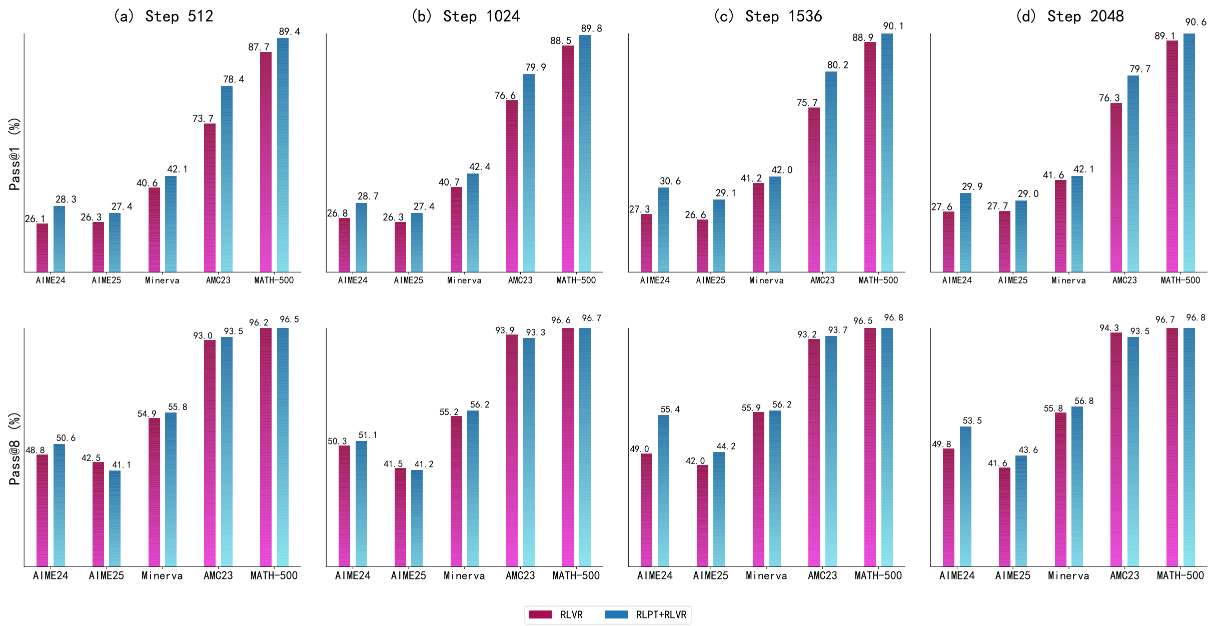

Figure 3: Comparative scaling properties of RLVR and RLPT + RLVR.

Thinking Process of RLPT

Alright, let's tackle this step by step. I'm going to predict the next most probable sentence after the given context...
Understanding the Context: the Fourier transform of $\chi_{[-a,a]}$ is defined, the integral is restricted to $[-a,a]$, the exponential is split into cosine and sine parts... The cosine integral gives $\frac{2\sin(a\xi)}{\xi}$, while the sine integral gives 0...
Identifying the Next Step: the next move would be to combine these results... Since the sine term vanishes, the final answer should come only from the cosine contribution...
Verifying the Flow: after evaluating both parts, the solution normally concludes by combining them into the closed-form Fourier transform...
Considering Alternative Path: could the solution instead highlight the behavior at $\xi = 0$ or present a piecewise definition... Alternatively, the solution might emphasize this special case, yet it is usually handled by noting the limit, so the general form is sufficient...
Looking back, one might also check the special case $\xi = 0$, but the limit already gives $2a$, so the concise expression is consistent...
So the natural continuation is: <|startofprediction|>Thus, combining these results, the Fourier transform of $\chi_{[-a,a]}$ is given by $F(\xi) = \frac{2\sin(a\xi)}{\xi}$. <|endofprediction|>

Table 3: Example of reasoning patterns of RLPT. Task-related keywords are highlighted in blue, while reasoning-related keywords are highlighted in red.

remains a common challenge for current RLVR methods (Yue et al., 2025; Walder & Karkhanis, 2025; Chen et al., 2025; Wu et al., 2025).

Thinking Patterns. To demonstrate the benefits of RLPT, we provide in Tab. 3 an illustrative example of its reasoning process. In this case, the model approaches the next-segment reasoning task through a structured sequence: it first abstracts the preceding context to capture the overarching flow, then determines the subsequent step, formulates a candidate continuation, verifies its plausibility, explores alternative possibilities, performs backtracking when appropriate, and ultimately produces the final answer. This structured trajectory aligns with the multi-step reasoning strategies exhibited by LLMs in complex problem-solving (Guo et al., 2025; Jaech et al., 2024), which helps explain the effectiveness of RLPT.

Reward Modeling. In developing RLPT, we iteratively refined our reward modeling approach after encountering several challenges with our initial formulation. Our initial approach adopted a strict reward that required the predicted segment to convey exactly the same semantic content as the ground-truth segment. This constraint proved too rigid, leading to numerous false positives. We observed that the model often generated outputs that

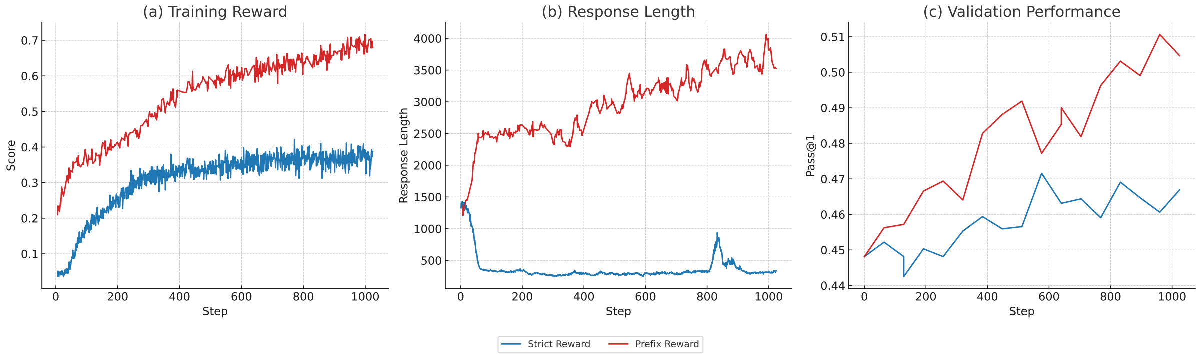

Figure 4: Comparison between Strict Reward and Prefix Reward: (a) Training Reward, (b) Response Length, (c) Validation Performance (Pass@1).

encompassed multiple ground-truth segments, largely due to the uneven distribution of information across sentence-based segmentation: some sentences contained only a single formula, while others might captured the complete solution to a subproblem. Such discrepancies disrupted the training process and yielded only limited improvements in downstream performance, as illustrated in Fig. 4. To address this issue, we introduce a relaxed prefix reward, which assigns a score of 1 as long as the predicted segment forms a valid prefix of the ground-truth completion. This adjustment addresses segments with varying information content and provides a more stable training signal. It also enables the model to generate longer responses, which in turn results in improved performance on downstream mathematical reasoning tasks, as shown in Fig. 4.

5 Related Work

Scaling Paradigms. The progress of language models has been fundamentally driven by scaling compute, which can be broadly divided into training-time scaling and test-time scaling. Training-time scaling primarily relies on next-token prediction, increasing computational cost by enlarging model size or expanding pre-training data to reduce prediction loss (Radford et al., 2019; Brown et al., 2020; Kaplan et al., 2020; Hoffmann et al., 2022). In contrast, test-time scaling allocates more compute during inference by generating extended chains of reasoning before producing the final answer (Brown et al., 2024; Jaech et al., 2024; Muennighoff et al., 2025; Guo et al., 2025). RLPT belongs to the training-time scaling paradigm but differs from prior approaches that emphasize supervised learning. Instead, it employs reinforcement learning (RL), allocating compute for the model to self-explore and learn from large-scale pre-training corpora. RL provides two notable advantages. First, it enables the model to uncover the latent reasoning underlying data, which can be regarded as a compressed form of deliberative thinking reflected in scientific papers or textbooks (Ruan et al., 2025). Second, recent research suggests that RL supports better generalization compared with supervised learning (Chu et al., 2025; Lai et al., 2025; Shenfeld et al., 2025). The most relevant approaches are RPT (Dong et al., 2025) and Quiet-STaR (Zelikman et al., 2024), both of which apply RL on unlabeled data for training-time scaling. However, RLPT differs by focusing on next-segment prediction rather than next-token prediction.

Reinforcement Learning in LLMs. RL has become a central paradigm for LLMs. Early applications mainly focused on aligning model outputs with human values (Bai et al., 2022; Ouyang et al., 2022; Mu et al., 2024), typically through reward models trained on human-annotated preference pairs. More recently, RL has been used to strengthen reasoning abilities by leveraging rule-based reward functions that evaluate outputs against reference answers (Guo et al., 2025; Zhu et al., 2025). Despite these advances, both directions ultimately depend on human-provided or verifiable supervision, which limits scalability. In contrast, RLPT introduces the next-segment reasoning objective, where the subsequent segment in natural text serves as the reference. This design removes the need for human annotation and enables RL to scale effectively on large-scale pre-training data.

6 Conclusion

This work introduces RLPT, a new training-time scaling paradigm that applies reinforcement learning to pre-training data. At its core, RLPT adopts a self-supervised next-segment reasoning objective, which removes the need for human annotations and enables RL training on large unlabeled corpora. Extensive experiments demonstrate the effectiveness of RLPT, yielding substantial gains in both general-domain and mathematical reasoning tasks. Moreover, the performance exhibits favorable scaling properties with respect to training compute, suggesting strong potential for further gains.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 interactive presentation sessions*, pp. 69–72, 2006.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*, 2025.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 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. 4171–4186, 2019.

Qingxiu Dong, Li Dong, Yao Tang, Tianzhu Ye, Yutao Sun, Zhifang Sui, and Furu Wei. Reinforcement pre-training. *arXiv preprint arXiv:2506.08007*, 2025.

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

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

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.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

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*, 2021a.

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021b.

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

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

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

Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Min Xie, Qingfu Zhang, et al. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training. *arXiv preprint arXiv:2507.05386*, 2025.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Kaijing Ma, Xinrun Du, Yunran Wang, Haoran Zhang, Zhoufutu Wen, Xingwei Qu, Jian Yang, Jiaheng Liu, Minghao Liu, Xiang Yue, et al. Kor-bench: Benchmarking language models on knowledge-orthogonal reasoning tasks. *arXiv preprint arXiv:2410.06526*, 2024.

MAA. American invitational mathematics examination (AIME). Mathematics Competition Series, n.d.a. URL https://maa.org/math-competitions/aime.

MAA. American mathematics competitions (AMC 10/12). Mathematics Competition Series, n.d.b. URL https://maa.org/math-competitions/amc.

Tong Mu, Alec Helyar, Johannes Heidecke, Joshua Achiam, Andrea Vallone, Ian Kivlichan, Molly Lin, Alex Beutel, John Schulman, and Lilian Weng. Rule based rewards for language model safety. *Advances in Neural Information Processing Systems*, 37:108877–108901, 2024.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376, 2023.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Yangjun Ruan, Neil Band, Chris J Maddison, and Tatsunori Hashimoto. Reasoning to learn from latent thoughts. *arXiv preprint arXiv:2503.18866*, 2025.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025a.

Tencent Hunyuan Team, Ao Liu, Botong Zhou, Can Xu, Chayse Zhou, ChenChen Zhang, Chengcheng Xu, Chenhao Wang, Decheng Wu, Dengpeng Wu, et al. Hunyuan-turbos: Advancing large language models through mamba-transformer synergy and adaptive chain-of-thought. *arXiv preprint arXiv:2505.15431*, 2025b.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: Will we run out of data? limits of LLM scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*, 2024.

Christian Walder and Deep Karkhanis. Pass@ k policy optimization: Solving harder reinforcement learning problems. *arXiv preprint arXiv:2505.15201*, 2025.

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

Fang Wu, Weihao Xuan, Ximing Lu, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

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

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Eric Zelikman, Georges Raif Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah Goodman. Quiet-STar: Language models can teach themselves to think before speaking. In *First Conference on Language Modeling*, 2024.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*, 2025.