Reinforcement Learning on Pre-Training Data

Siheng Li[1,3*†] **Kejiao Li**[1†] **Zenan Xu**[1†] **Guanhua Huang**[1] **Kun Li**[1,3] **Haoyuan Wu**[1,3]
Jiajia Wu[1] **Zihao Zheng**[1] **Chenchen Zhang**[1] **Kun Shi**[1] **Xue Gong**[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] **Bo Zhou**[1,‡] **Di Wang**[1]
[1]LLM Department, Tencent [2]HunYuan Infra [3]The Chinese University of Hong Kong
chayszhou@tencent.com

Abstract

Recent progress in large language models (LLMs) is largely driven by scaling training compute through either pre-training with next-token prediction (NTP) or post-training with reinforcement learning (RL). The former contributes to learning broad knowledge and skills from general data, while struggling with data inefficiency and catastrophic forgetting in continual learning settings. The latter incentivizes reasoning capabilities with strong generalization, but is constrained by limited data availability due to its reliance on human annotation. To alleviate these issues, we propose Reinforcement Learning on Pre-Training data (RLPT), which combines the advantages of learning from general data and RL. In particular, RLPT derives reward signals directly from general text data through a next-segment reasoning objective, rewarding the policy for correctly predicting next text segments conditioned on the prefix text. Experiments across multiple benchmarks and models demonstrate the effectiveness of RLPT. For example, RLPT yields substantial improvements in continual pre-training ($+4.6\%$) and provides a strong foundation for post-training ($+3.4\%$) on Qwen3-8B-Base.

1 Introduction

Large language models (LLMs) have achieved remarkable success across a wide range of domains, including human-aligned conversational assistants (Bai et al., 2022; Ouyang et al., 2022) and autonomous AI agents (Yao et al., 2022; Team et al., 2025; Liu et al., 2025). A central driver of this progress has been the scaling of training compute, achieved either by scaling pre-training with next-token prediction (NTP) (Brown et al., 2020; Touvron et al., 2023; Grattafiori et al., 2024; Yang et al., 2025) or by scaling reinforcement learning (RL) during post-training (Guo et al., 2025; xAI, 2025).

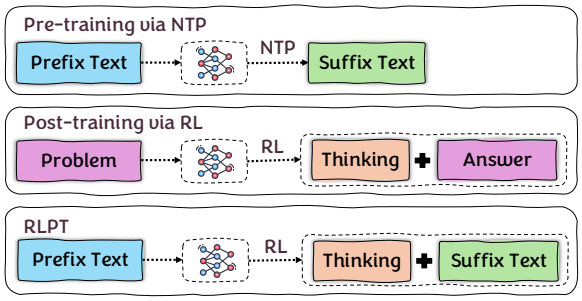

Figure 1: Relationship among pre-training via NTP, post-training via RL, and RLPT. Pre-training via NTP and RLPT learn from general text data, while post-training via RL and RLPT enable self-exploration of reasoning trajectories and optimization through RL.

Pre-training with NTP enables models to acquire extensive knowledge and skills from general data, but it suffers from data inefficiency[1] (Ruan et al., 2025; Ni et al., 2025) and exhibits catastrophic forgetting in continual learning settings (Luo et al., 2025). In contrast, scaling RL has emerged as a promising direction, enabling models to self-develop desirable behaviors while suppressing undesirable ones via pre-defined rewards. Recent studies suggest that RL favors generalization over memorization (Chu et al., 2025; Lai et al., 2025; Shenfeld et al., 2025) and incentivizes complex reasoning by allowing the model to think before prediction (Guo et al., 2025). However, most existing RL approaches are primarily confined to the post-training stage and rely on human supervision, such as reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) and reinforcement learning with verifiable rewards (RLVR) (Guo et al., 2025). This reliance limits the applicability of RL to general data, which may be critical for achieving general intelligence (Han et al., 2025).

This work proposes **R**einforcement **L**earning on **P**re-**T**raining data (**RLPT**), with the key insight of

[*]Work completed during an internship at Tencent.
[†]The first three authors contributed equally.
[‡]Project Lead.

[1]To reach intelligence comparable to that of humans, LLMs typically require an order of magnitude more data.

deriving reward signals directly from general text data. In particular, RLPT prompts the model to reason about and predict the next segment conditioned on the preceding context, for example, predicting the next sentence given a paragraph. The reward is then determined based on the consistency between the prediction and the ground-truth continuation. Intuitively, we treat a model trajectory as desirable if it leads to a continuation that is semantically equivalent to the ground-truth, and undesirable otherwise. As illustrated in Fig. 1, RLPT combines the advantages of learning from general data and RL, while addressing the limitations of previous approaches. In contrast to pre-training via NTP, which enforces next-token prediction via supervised learning, RLPT enables the model to self-develop reasonable behaviors and forget less in continual learning settings via RL (Shenfeld et al., 2025). Differing from post-training via RL, RLPT eliminates reliance on human supervision and scales to large-scale general data.

We compare RLPT with NTP under both continual pre-training and post-training settings, where post-training models are initialized from the continual pre-training checkpoints. Experimental results across eight benchmarks demonstrate the effectiveness of RLPT. For example, on Qwen3-8B-Base, RLPT achieves average gains of 4.6% in continual pre-training (Tab. 1) and 3.4% in post-training performance (Tab. 2), indicating its potential for building strong foundation models. We also compare RLPT with concurrent work that applies RL to pre-training data, including RPT (Dong et al., 2025) and RLP (Hatamizadeh et al., 2025). While these approaches apply RL to pre-training data with token-level reward signals, RLPT adopts a segment-level reward and achieves superior performance (Tab. 3). Additional analyses show that RLPT incentivizes reasoning behavior (Tab. 5) and exhibits a mild form of test-time scaling, reflected by increased response length (Fig. 3). The main contributions are as follows:

- We propose RLPT, which applies RL directly to pre-training data to leverage the strengths of learning from general data and RL.
- Experiments across a diverse set of benchmarks and models demonstrate the effectiveness of RLPT in both continual pre-training and post-training settings.
- Further analysis provides insights into the training dynamics, model behavior, and methodological design of RLPT.

2 Preliminary

2.1 Pre-training via Next-token Prediction

Next-token prediction (NTP) forms the foundation of modern LLMs. Formally,

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

where x denotes a token sequence of length $|x|$. Pre-training with NTP enables LLMs to acquire broad knowledge and skills (Brown et al., 2020). However, it is data-inefficient (Ruan et al., 2025; Ni et al., 2025), as evidenced by the substantial amount of training data required to improve intelligence. Moreover, it suffers from catastrophic forgetting in continual learning settings across various model scales and domains (Luo et al., 2025). Recent studies further indicate that supervised learning with NTP tends to favor surface-level memorization, which limits generalization, whereas RL encourages generalization and mitigates forgetting (Chu et al., 2025; Shenfeld et al., 2025).

2.2 Post-training via Reinforcement Learning

RL has become an essential component for improving LLMs in the post-training stage. Formally,

$$\mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{q\sim D_q, o\sim\pi_\theta(\cdot|q)} r(o),$$

where $r(o)$ denotes the reward assigned to output o. In RLHF, rewards are provided by a neural reward model trained on human preferences to align model behavior with human values. In contrast, RLVR relies on rule-based reward functions to compare model outputs against reference answers. Recent studies indicate that applying RLVR to competitive problems incentivizes complex reasoning behaviors, often reflected as test-time scaling (Guo et al., 2025). Despite their effectiveness, both RLHF and RLVR are typically applied in the post-training stage and are constrained by limited data availability due to their reliance on human supervision.

3 Reinforcement Learning on Pre-Training Data

To unify the strengths of learning from large-scale pre-training data with the generalization and reasoning capabilities enabled by RL, we propose Reinforcement Learning on Pre-Training Data

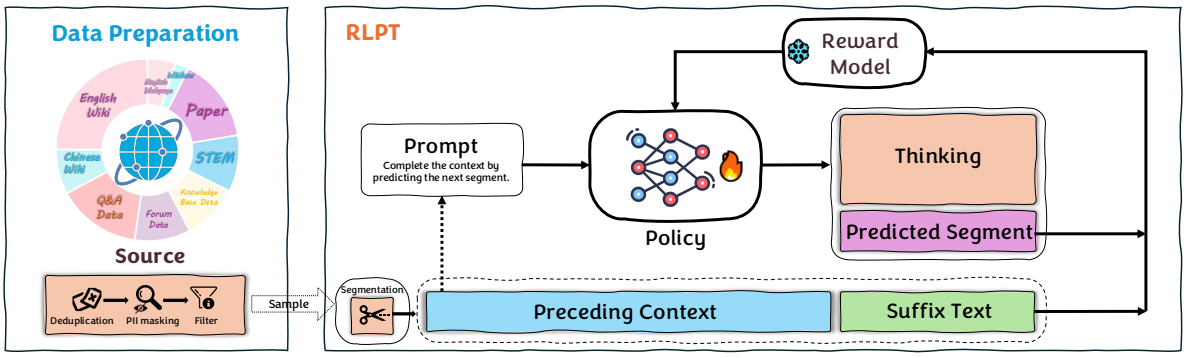

Figure 2: Overview of RLPT. A raw text is first segmented. RLPT then prompts the model to reason about and predict the next segment conditioned on the preceding context, and assigns rewards based on consistency between the prediction and the ground-truth continuation, as evaluated by a reward model.

(RLPT). The core idea is to derive reward signals directly from general text data through a novel next-segment reasoning objective. An overview of RLPT is shown in Fig. 2.

3.1 Data Preparation

We construct the training corpus for RLPT from a large-scale collection of publicly available web text, including Wikipedia, scientific writing, and general-domain discourse. To ensure data quality and regulatory compliance, we apply a multi-stage preprocessing pipeline that includes: (i) MinHash-based near-duplicate removal, (ii) detection and masking of personally identifiable information (PII), (iii) filtering of harmful and unsafe content, and (iv) contamination removal with respect to all development and evaluation benchmarks. After curation, we sample a subset of one billion (1B) tokens for all experiments in this work.

3.2 Next-Segment Reasoning

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

$$\mathcal{D}_s = (s_{<i}, s_i, s_{>i}) \mid i = 2, \ldots, n-1,$$

where $s_{<i} = [s_1, s_2, \ldots, s_{i-1}]$ denotes the context, s_i is the target segment, and $s_{>i}$ is its subsequent segments. Based on this formulation, we propose Next-Segment Reasoning (NSR), which trains the policy to predict the target segment s_i conditioned on the preceding context $s_{<i}$. This objective aligns naturally with the autoregressive generation process of modern LLMs while operating at a higher semantic granularity. During training, NSR is implemented by designing appropriate prompts and extracting the predicted segment between special tags in the model output. The prompt used for the NSR task is illustrated below.

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

You should first provide your step-by-step reasoning.

Then, provide only the predicted next sentence enclosed
within <answer> and </answer> tags.

### Context
{prompt}

### Your Output Format
Step-by-step reasoning about what the next sentence
should be.

<answer>
The predicted next sentence.
</answer>
```

The reward is defined as the consistency between the predicted segment and the ground-truth continuation, as evaluated by a generative reward model (GRM). Specifically, the GRM is prompted to assess whether the prediction conveys equivalent semantic content to the ground-truth while allowing for linguistic variation. In practice, we observe that directly comparing the predicted segment $\hat{s}_i$ with the ground-truth next segment s_i can be overly strict, since the prediction might span multiple subsequent segments. To address this issue, we provide the GRM with multiple subsequent segments $s_{\geq i}$ as the reference and instruct it to verify whether the predicted segment constitutes a valid prefix of the reference content, following Dong et al. (2025). The prompt used for the GRM is shown below.

```
Annotation Guidelines for Comparing Prefix Semantic
Equivalence

## Task
Given a Predicted text and a Reference text, determine
whether the Predicted text is a prefix (initial segment)
 of the Reference, and whether it expresses exactly the
same semantic content as that corresponding prefix.

The Predicted text should serve as a paraphrased version
 of the beginning of the Reference, preserving all
specific semantics rather than simplifying or
summarizing the content. The expression may vary, but
the meaning must remain fully consistent.

<<<Reference>>>
{reference}
<<<End Reference>>>

<<<Predicted>>>
{predicted}
<<<End Predicted>>>

## Scoring Rules
If the Predicted text semantically equals the
corresponding prefix of the Reference, assign a score 1.

If the Predicted text does not semantically equal the
corresponding prefix of the Reference, assign a score 0.

Focus primarily on semantic equivalence, not on exact
wording.

## Output
Only output the score on a single line. Do not provide
any explanatory text or additional content.

Output format:
Score: 0 or Score: 1
```

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

$$r(o, s_{\geq i}) = \begin{cases} 1 & \text{if GRM}(\hat{s}_i, s_{\geq i}) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The training objective of RLPT is defined as

$$\mathcal{J}_{\text{RLPT}}(\theta) = \mathbb{E}_{(s_{<i}, s_{\geq i}) \sim \mathcal{D}_s, o \sim \pi_\theta} r(o, s_{\geq i}).$$

This objective is optimized with GRPO (Shao et al., 2024). During training, trajectories that lead to the correct next segment are reinforced, while others are suppressed, fostering RL on general data.

3.3 Training

Cold-Start. We employ a cold-start stage for RLPT to establish basic reasoning and instruction-following capabilities through supervised fine-tuning on instruction-following data.

Next-Segment Reasoning. For NSR, we use sentences as the default segment unit. We conduct preliminary studies on using LLMs to extract steps but observe no clear improvements. Sentence segmentation is performed using the NLTK toolkit (Bird, 2006). Starting from the 1B pre-training corpus, this process yields 23M sentences. After filtering out sentences shorter than 16 tokens, longer than 256 tokens, or with preceding context exceeding 2048 tokens, we retain 9M sentences. Each sentence, together with its preceding context, serves as an RL training example.

4 Experiments

4.1 Experimental Setup

We compare RLPT with two lines of work. The first line is mainstream next-token prediction training, referred to as NTP (Sec. §4.4). The second line consists of concurrent work that also applies RL to general text data, including RPT (Dong et al., 2025) and RLP (Hatamizadeh et al., 2025) (Sec. §4.5). For comparisons with NTP, we evaluate performance under both continual pre-training and post-training settings. In the continual pre-training setting, models are trained on 1B tokens of pre-training data (Sec. §3.1). In the post-training setting, models are initialized from the continual pre-training checkpoint and further trained on the MATH training dataset (Hendrycks et al., 2021b) using RLVR, to assess the impact of different continual pre-training methods on post-training performance. For comparisons with concurrent work, we focus on continual pre-training performance.

4.2 Implementation

Experiments are conducted on Qwen3-1.7B-Base and Qwen3-8B-Base, as well as Llama3.2-3B-Instruct. For the cold-start stage, we train on 100K internal instruction-following examples for two epochs using a batch size of 1024 and a learning rate of 1×10^{-5} with a cosine scheduler; results are reported in Appx. A.1. For next-segment reasoning, we adopt a batch size of 1024, 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 optimize using GRPO. Training is performed for 512 steps, covering 6% of the 1B pre-training data. We employ `gpt-oss-120b` (OpenAI, 2025), with 5.1B active parameters and low reasoning effort, as the GRM. For NTP, models are also initialized from the cold-start checkpoint and trained on the same pre-training data for one epoch using a learning rate of 1×10^{-6} with a cosine scheduler. For post-training via RLVR, we adopt the same hyperparameters as RLPT, except that models are trained on MATH for two epochs with a batch size of 128. The reproduction of RPT follows the original pa-

Model	MMLU	MMLU-Pro	GPQA	AIME25	MATH500	GSM8K	AMC23	Minerva	Average
Qwen3-1.7B-Base	41.97	21.73	23.36	3.75	46.88	58.89	25.94	11.21	29.21
+ NTP	58.26	37.49	18.43	3.75	61.08	75.55	28.75	24.31	38.45
+ **RLPT**	**61.28**	**41.68**	**26.01**	**7.08**	**66.45**	**77.20**	**35.00**	**26.06**	**42.59**
Qwen3-8B-Base	48.93	42.11	30.56	6.67	64.63	79.37	45.00	25.74	42.87
+ NTP	75.25	60.54	35.73	17.50	80.95	87.56	58.75	38.88	56.89
+ **RLPT**	**78.81**	**65.08**	**42.55**	**19.58**	**85.75**	**90.17**	**68.75**	**41.08**	**61.47**
Llama-3.2-3B-Instruct	61.93	36.79	**26.26**	**1.67**	42.05	68.16	22.19	14.20	34.16
+ NTP	57.46	33.55	17.55	1.25	42.00	67.71	18.13	12.91	31.32
+ **RLPT**	**62.36**	**38.50**	20.83	0.83	**48.50**	**72.33**	**25.94**	**18.06**	**35.92**

Table 1: Continual pre-training performance, with the best result highlighted in bold for each model scale.

Foundation	MMLU	MMLU-Pro	GPQA	AIME25	MATH500	GSM8K	AMC23	Minerva	Average
Qwen3-1.7B-Base	58.08	36.49	**25.13**	5.00	65.75	**81.25**	36.88	27.62	42.02
+ NTP	60.19	40.24	21.72	**7.50**	66.88	80.73	**42.50**	26.98	43.34
+ **RLPT**	**61.98**	**43.01**	24.87	5.00	**69.88**	80.42	40.63	**28.49**	**44.28**
Qwen3-8B-Base	73.75	56.72	36.11	16.25	81.05	**92.17**	66.88	38.42	57.67
+ NTP	76.05	63.27	40.53	20.00	85.73	91.81	65.63	43.34	60.79
+ **RLPT**	**79.48**	**67.59**	**46.84**	**21.25**	**88.95**	92.10	**73.44**	**43.47**	**64.14**
Llama-3.2-3B-Instruct	61.63	34.15	15.78	0.00	48.40	**79.95**	27.81	19.39	35.89
+ NTP	60.44	37.41	**24.24**	1.25	48.40	72.87	28.44	18.29	36.42
+ **RLPT**	**62.24**	**39.67**	23.99	**2.50**	**53.10**	74.72	**28.44**	**20.59**	**38.16**

Table 2: Post-training performance of models initialized from the foundation models listed in the table, under the same post-training procedure. The best result at each model scale is highlighted in bold.

per. Specifically, low-entropy tokens are filtered based on Qwen3-1.7B-Base using a threshold of 1.0, and the model is trained with the same settings as RLPT, initialized from the cold-start checkpoint.

4.3 Evaluation

We evaluate model performance on a suite of benchmarks, including MMLU (Hendrycks et al., 2021a), MMLU-Pro (Wang et al., 2024), GPQA-Diamond (Rein et al., 2024), GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021b), AMC23 (MAA, b), Minerva Math (Lewkowycz et al., 2022), and AIME25 (MAA, a). We adopt the Pass@k metric, which measures the probability that at least one correct solution appears among k independent samples (Chen et al., 2021):

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

For all models, we sample $n = 4$ responses for MMLU, MMLU-Pro, and GPQA-Diamond, and $n = 8$ for other benchmarks, using temperature 0.7, top-p 0.8, top-k 20, and a maximum length of 8192, and report Pass@1 by default.

4.4 Comparison with NTP

Continual Pre-training. We report the continual pre-training results in Tab. 1. As shown, RLPT outperforms NTP across multiple benchmarks, achieving average benchmark performance improvements of 4.14, 4.58, and 4.60 on Qwen3-1.7B-Base, Qwen3-8B-Base, and Llama-3.2-3B-Instruct, respectively. These results indicate the effectiveness of RLPT as a continual pre-training method. Notably, NTP can even degrade performance, as evidenced by the performance drop on Llama-3.2-3B-Instruct, despite using the same training data as RLPT. We attribute this distinction to properties of RL that favor generalization while mitigating catastrophic forgetting (Shenfeld et al., 2025).

Post-training. We report the post-training results in Tab. 2. As shown, RLPT achieves better perfor-

Method	MMLU	MMLU-Pro	GPQA	AIME25	MATH500	GSM8K	AMC23	Minerva	Average
RPT† (Dong et al., 2025)	55.09	35.40	20.33	**7.50**	62.63	**77.65**	28.75	24.54	38.99
RLP‡ (Hatamizadeh et al., 2025)	52.18	30.80	**27.02**	5.02	58.48	74.48	31.25	21.19	37.55
RLPT	**61.28**	**41.68**	26.01	7.08	**66.45**	77.20	**35.00**	**26.06**	**42.59**

Table 3: Continual pre-training results on Qwen3-1.7B-Base across RPT, RLP, and RLPT. †: results reproduced by ourselves under the identical training setting; ‡: results reported from the original paper, trained with the same amount of data. All methods are evaluated under the same evaluation setting.

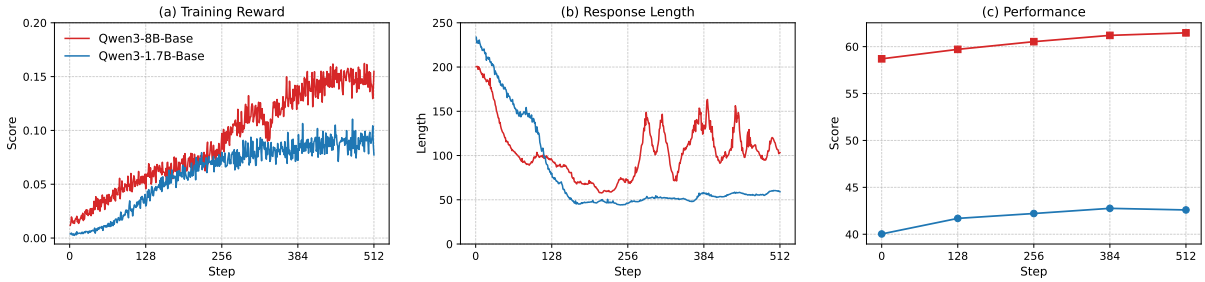

Figure 3: Training dynamics of RLPT. (a) Training reward during RL optimization. (b) Response length over training steps. (c) Average benchmark performance measured by Pass@1.

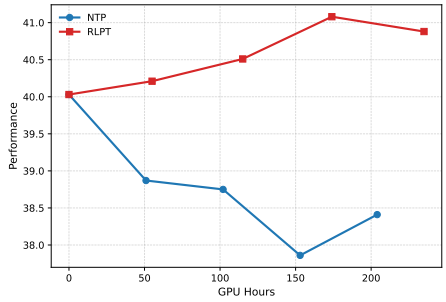

Figure 4: Performance of RLPT and NTP under matched training compute. Results report the averaged benchmark performance measured by Pass@1.

mance than the baselines, with average improvements of 0.94, 3.35, and 1.74 on Qwen3-1.7B-Base, Qwen3-8B-Base, and Llama-3.2-3B-Instruct, respectively. The gains are more pronounced for larger models, such as Qwen3-8B-Base, indicating the potential of RLPT for improving stronger foundation models. These results suggest that RLPT more effectively learns from pre-training data and provides a stronger foundation for post-training.

4.5 Comparison with Concurrent Work

We compare RLPT with concurrent work, including RPT and RLP. Both methods employ RL to train the model to reason before next-token prediction, whereas RLPT targets next-segment prediction. As shown in Tab. 3, RLPT outperforms these approaches. We attribute this advantage to our segment-level objective, which is more training-efficient since each segment comprises multiple tokens, and more reasoning-intensive, as predicting the next segment generally requires deeper reasoning than predicting the next token.

4.6 Analysis

Training Dynamics. We present the training dynamics in Fig. 3. The training reward increases steadily throughout optimization (Fig. 3a), accompanied by consistent improvements in benchmark performance (Fig. 3c). More interestingly, we observe a mild form of test-time scaling behavior on Qwen3-8B-Base, where the response length gradually increases as training progresses (Fig. 3b), although the effect is weaker than that reported in RLVR (Guo et al., 2025). We attribute this difference to the fact that pre-training data is generally less reasoning-intensive than the competitive problems used in RLVR.

Training Compute. RLPT requires substantially more training compute than NTP. Specifically, one epoch of NTP on Qwen3-1.7B-Base consumes 51 GPU hours, whereas one epoch of RLPT requires 26,126 GPU hours, corresponding to an approximately 512× increase in compute cost. This overhead constitutes a fundamental limitation of RLPT. To assess the value of RLPT, we further compare the performance of NTP and RLPT under matched GPU-hour budgets. As shown in Fig. 4, RLPT still

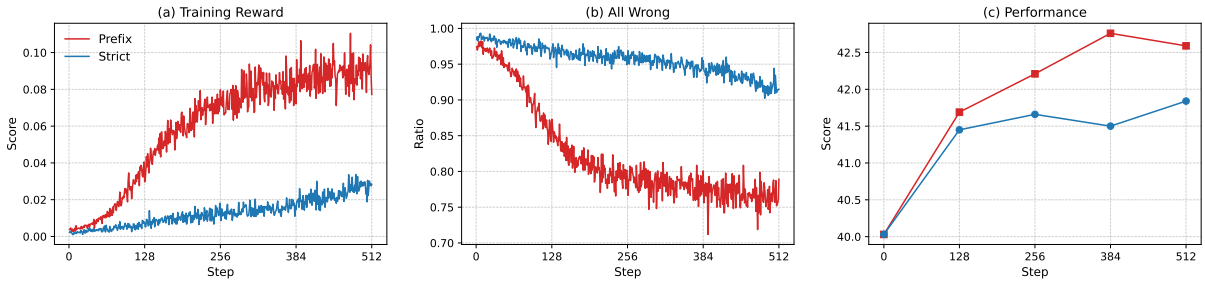

Figure 5: Comparison between Strict Reward and Prefix Reward in RLPT on Qwen3-1.7B-Base. (a) Training reward during RL optimization. (b) Fraction of prompts for which all 8 sampled rollouts receive zero reward. (c) Average benchmark performance measured by Pass@1.

outperforms NTP. This difference can be attributed to the catastrophic forgetting exhibited by NTP during continual pre-training (Luo et al., 2025). In contrast, RLPT mitigates forgetting through RL-based learning and continues to benefit from training data, as reflected by its performance gains.

Model	Reward Model	Average
Qwen3-1.7B-Base	–	29.21
+ NTP	–	38.45
+ RLPT	Qwen3-1.7B	41.08
+ RLPT	Qwen2.5-1.5B-Instruct	41.49
+ RLPT	gpt-oss-120b (A5.1B)	**41.69**

Table 4: Average benchmark performance of RLPT with different generative reward models. All reward models yield consistent performance gains.

Reward Modeling. Our initial approach adopts a strict reward that requires the predicted sentence to convey exactly the same semantic content as the ground-truth sentence. In practice, however, this formulation proves to be overly rigid. Natural language semantics are highly diverse, and there often exist multiple plausible next sentences that may convey more or less information than the ground-truth sentence, making exact semantic equivalence difficult to satisfy. To mitigate this limitation, we adopt a relaxed prefix reward (Sec. §3.2) following Dong et al. (2025). This reward assigns a score of 1 when the predicted sentence conveys the same semantic information as the prefix of the ground-truth continuation, i.e., the text following the preceding context. As shown in Fig. 5, the prefix reward leads to higher training rewards, fewer all-wrong cases, and improved benchmark performance. A limitation of RLPT is its reliance on a relatively strong reward model. By default, we use `gpt-oss-120b` with 5.1B active parameters. To assess this dependence, we additionally evaluate RLPT using weaker reward models, including Qwen3-1.7B and Qwen2.5-1.5B-Instruct, with all reward model settings trained for 128 steps. As shown in Tab. 4, RLPT achieves performance gains across all reward models, suggesting that it does not critically depend on strong reward models.

Case Study. We present a case study of RLPT in Tab. 5. As shown, the model summarizes the context, hypothesizes the next continuation, and reasons step by step before producing the predicted next sentence within the designated tags. This example illustrates how RLPT trains the model to acquire reasonable behaviors from general data, which may support more generalizable reasoning capabilities (Han et al., 2025).

5 Related Work

Training Paradigms. The progress of language models has been fundamentally driven by scaling training compute, either through large-scale pre-training based on next-token prediction (Brown et al., 2020; Touvron et al., 2023; Liu et al., 2024; Yang et al., 2025), or more recently through post-training with RL (Guo et al., 2025; xAI, 2025; Olmo et al., 2025). These two lines of research each exhibit distinct strengths and limitations. Pre-training excels at acquiring extensive knowledge and general skills, but often suffers from data inefficiency (Ruan et al., 2025). In contrast, RL-based post-training effectively incentivizes reasoning behaviors and generalization capability (Chu et al., 2025; Lai et al., 2025; Shenfeld et al., 2025), but is constrained by the limited scale of post-training data due to its reliance on human annotation. In contrast to both paradigms, RLPT applies RL directly to general pre-training data, thereby combining their complementary advantages. An earlier

Prompt	Complete the text provided under ### Context by predicting the next most probable sentence. You should first provide your step-by-step reasoning. Then, provide only the predicted next sentence enclosed within <answer> and </answer> tags. ### Context ## Solutions for exercises "C" in September, 2001 In this page, the sketch of the solutions and final results are published. To achieve the maximum score, more detailed solutions are needed. **C. 635.** In an arithmetic class, pupils are asked how many legs a hen, six dogs, and seven palpigradis have in total. Alex says 46, Ben says 52, Cecilia says 66, Dora says 78, and Edith says 82. Who is right? **Solution.** If the number of legs of a palpigradi is x, then the total number of legs is $2+6\cdot4+7x = 7(x+3)+5$. Thus, the remainder of the number of legs divided by 7 is 5. ### Your Output Format Step-by-step reasoning about what the next sentence should be. <answer> The predicted next sentence. </answer>
GT	82 is the only number out of 46, 52, 66, 78, and 82 that leaves a remainder of 5 when divided by 7, therefore only Edith can be right.
Response	Step-by-step reasoning: • The previous sentence is about the mathematical formulation of the problem by letting x represent the number of legs of a palpigradi, and deriving the total number of legs as an expression in terms of x and showing that the total number of legs $\equiv 5 \pmod 7$. • The natural continuation of this line of reasoning would be: since the answer to the question should be congruent to 5 modulo 7 (i.e., leave a remainder of 5 when divided by 7), we can quickly check which of the given answers satisfies this condition. • Without doing the actual division, we can actually compute each answer modulo 7: – 46 for Alex: 46 ÷ 7 = 6 remainder 4 $\rightarrow 46 \equiv 4 \bmod 7 \neq 5$ – 52 for Ben: 52 ÷ 7 = 7 remainder 3 $\rightarrow 52 \equiv 3 \bmod 7 \neq 5$ – 66 for Cecilia: 66 ÷ 7 = 9 remainder 3 $\rightarrow 66 \equiv 3 \bmod 7 \neq 5$ – 78 for Dora: 78 ÷ 7 = 11 remainder 1 $\rightarrow 78 \equiv 1 \bmod 7 \neq 5$ – 82 for Edith: 82 ÷ 7 = 11 remainder 5 $\rightarrow 82 \equiv 5 \bmod 7$ • Hence, only Edith's answer is congruent to 5 modulo 7, meaning it is the correct answer. So the predicted next sentence should confirm Edith's answer as the correct one based on this modulo 7 check. <answer> Only Edith's answer (82) satisfies this condition, as $82 \equiv 5 \pmod 7$, so she is right. </answer>

Table 5: Case study of RLPT. The response is generated by a Qwen-8B-Base model trained with RLPT.

work, Quiet-STaR (Zelikman et al., 2024), shares a similar motivation, but relies on short reasoning tokens, for example 16 tokens, and requires architectural modifications to support optimization. The most closely related approaches to our work are RPT (Dong et al., 2025) and RLP (Hatamizadeh et al., 2025), both of which apply RL to general text data with a next-token reasoning objective, while RLPT targets at next-segment reasoning.

Reinforcement Learning in LLMs. RL has become a central approach for improving LLMs. Early applications focused on aligning model outputs with human feedback, i.e., RLHF (Bai et al., 2022; Ouyang et al., 2022; Mu et al., 2024), typically using neural reward models trained on human-annotated preference pairs. More recently, RL has been used to incentivize reasoning behaviors by leveraging rule-based reward functions that evaluate outputs against verifiable answers, i.e., RLVR (Guo et al., 2025; Team et al., 2025; Liu et al., 2025). Despite these advances, both directions depend on human annotation, which limits the application of RL to the post-training stage. In contrast, RLPT derives reward signals directly from pre-training data, removing the need for human annotation and enabling RL to scale effectively over large-scale general data.

6 Conclusion

This work introduces RLPT, which applies RL to pre-training data to leverage the advantages of both learning from general data and RL. By deriving reward signals from raw text via a next-segment reasoning objective, RLPT teaches models to develop reasonable trajectories for predicting the next segment conditioned on the preceding context. Experimental results show improvements across multiple benchmarks under both continual pre-training and post-training settings, underscoring the potential of training LLMs with RL on general data. We hope this work will inspire further research on more general RL-based training, advancing LLM training

from an "AlphaGo" stage toward an "AlphaZero" regime (Silver et al., 2017, 2018; Han et al., 2025), with reduced reliance on human demonstration and greater dependence on self-exploration.

Limitations

We acknowledge several limitations of this work, many of which stem from the relatively under-explored nature of reinforcement learning on large-scale pre-training data.

Data. Pre-training corpora exhibit highly imbalanced data quality: a substantial portion of the data is noisy or provides limited reasoning value, while only a small fraction demonstrates strong reasoning characteristics, which are the primary targets of RLPT. As a result, reinforcement learning on such data can be inefficient, since the model must sample multiple outputs per prompt and reason about the next segment even when the underlying data is weakly informative. Future work could explore data selection or filtering strategies to identify and extract the most valuable, particularly reasoning-intensive, subsets from large-scale pre-training corpora before applying reinforcement learning.

Method. RLPT relies on a generative reward model, which incurs additional computational overhead during training. Moreover, as with many learned reward functions, this design may be susceptible to reward hacking, potentially constraining the scalability and robustness of the approach.

Experiments. Due to computational constraints, our experiments are limited to models of up to 8B parameters. Extending the investigation to larger models, and in particular to mixture-of-experts architectures (Du et al., 2022; Dai et al., 2024), remains an important direction for future work.

Acknowledgments

This research/paper was partially supported by the Center for Perceptual and Interactive Intelligence (CPII) Ltd. under the Innovation and Technology Commission's InnoHK scheme.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, and 1 others. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*.

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

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, and 1 others. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International conference on machine learning*, pages 5547–5569. PMLR.

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

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

Seungwook Han, Jyothish Pari, Samuel J. Gershman, and Pulkit Agrawal. 2025. Position: General intelligence requires reward-based pretraining. In *Forty-second International Conference on Machine Learning Position Paper Track*.

Ali Hatamizadeh, Syeda Nahida Akter, Shrimai Prabhumoye, Jan Kautz, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. 2025. Rlp: Reinforcement as a pretraining objective. *arXiv preprint arXiv:2510.01265*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Min Xie, Qingfu Zhang, and 1 others. 2025. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training. *arXiv preprint arXiv:2507.05386*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*.

MAA. n.d.a. American invitational mathematics examination (AIME). Mathematics Competition Series.

MAA. n.d.b. American mathematics competitions (AMC 10/12). Mathematics Competition Series.

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

Jinjie Ni, Qian Liu, Longxu Dou, Chao Du, Zili Wang, Hang Yan, Tianyu Pang, and Michael Qizhe Shieh. 2025. Diffusion language models are super data learners. *arXiv preprint arXiv:2511.03276*.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, and 1 others. 2025. Olmo 3. *arXiv preprint arXiv:2512.13961*.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

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

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

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

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

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, and 1 others. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, and 1 others. 2018. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

xAI. 2025. Grok 4 model card.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

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

A Appendix

A.1 Cold-start Results

We report the performance after the cold-start stage in Tab. 6. As shown, cold-start improves the performance of Qwen3-1.7B-Base and Qwen3-8B-Base, while slightly degrading performance on Llama-3.2-3B-Instruct, which has already undergone extensive post-training (Grattafiori et al., 2024). Notably, RLPT clearly outperforms both the original models and their cold-start counterparts, indicating that it effectively learns from pre-training data through the proposed next-segment reasoning objective. We further report post-training performance initialized from different foundation models in Tab. 7. As shown, RLPT consistently outperforms the baseline models, highlighting its potential for building stronger foundation models.

A.2 Information About Use Of Ai Assistants

In this work, AI assistants are used solely for writing support, such as improving grammar and readability. All AI-assisted edits have been carefully reviewed by the authors. The AI assistants did not contribute to any scientific content, experimental design, analysis, or conclusions of this work.

Model	MMLU	MMLU-Pro	GPQA	AIME25	MATH500	GSM8K	AMC23	Minerva	Average
Qwen3-1.7B-Base	41.97	21.73	23.36	3.75	46.88	58.89	25.94	11.21	29.21
+ Cold-start	59.39	38.70	19.19	8.33	62.85	77.02	30.31	24.45	40.03
+ **RLPT**	**61.28**	**41.68**	**26.01**	**7.08**	**66.45**	**77.20**	**35.00**	**26.06**	**42.59**
Qwen3-8B-Base	48.93	42.11	30.56	6.67	64.63	79.37	45.00	25.74	42.87
+ Cold-start	75.50	62.39	39.27	18.33	83.95	88.95	61.56	39.66	58.70
+ **RLPT**	**78.81**	**65.08**	**42.55**	**19.58**	**85.75**	**90.17**	**68.75**	**41.08**	**61.47**
Llama-3.2-3B-Instruct	61.93	36.79	**26.26**	**1.67**	42.05	68.16	22.19	14.20	34.16
+ Cold-start	59.02	35.25	17.93	0.00	45.33	69.77	25.00	14.98	33.41
+ **RLPT**	**62.36**	**38.50**	20.83	0.83	**48.50**	**72.33**	**25.94**	**18.06**	**35.92**

Table 6: Continual pre-training performance, with the best result highlighted in bold for each model scale.

Foundation	MMLU	MMLU-Pro	GPQA	AIME25	MATH500	GSM8K	AMC23	Minerva	Average
Qwen3-1.7B-Base	58.08	36.49	**25.13**	5.00	65.75	**81.25**	36.88	27.62	42.02
+ Cold-start	60.25	40.29	20.58	**5.42**	66.80	79.96	36.25	25.78	41.92
+ **RLPT**	**61.98**	**43.01**	24.87	5.00	**69.88**	80.42	**40.63**	**28.49**	**44.28**
Qwen3-8B-Base	73.75	56.72	36.11	16.25	81.05	**92.17**	66.88	38.42	57.67
+ Cold-start	75.67	64.31	42.93	24.17	86.63	92.11	65.94	43.43	61.90
+ **RLPT**	**79.48**	**67.59**	**46.84**	**21.25**	**88.95**	92.10	**73.44**	**43.47**	**64.14**
Llama-3.2-3B-Instruct	59.77	35.86	17.68	0.83	51.95	73.12	28.13	18.93	35.78
+ Cold-start	61.63	34.15	15.78	0.00	48.40	**79.95**	27.81	19.39	35.89
+ **RLPT**	**62.24**	**39.67**	**23.99**	**2.50**	**53.10**	74.72	**28.44**	**20.59**	**38.16**

Table 7: Post-training performance of models initialized from the foundation models listed in the table, under the same post-training procedure. The best result at each model scale is highlighted in bold.