RLP: Reinforcement as a Pretraining Objective

Ali Hatamizadeh[†1], **Syeda Nahida Akter**[†2*], **Shrimai Prabhumoye**[†1,3], **Jan Kautz**[1], **Mostofa Patwary**[1], **Mohammad Shoeybi**[1], **Bryan Catanzaro**[1], **Yejin Choi**[1,4]
NVIDIA[1], Carnegie Mellon University[2], Boston University[3], Stanford University[4]
ahatamizadeh@nvidia.com, sprabhumoye@nvidia.com

Abstract

The dominant paradigm for training large reasoning models starts with pre-training using next-token prediction loss on vast amounts of data. Reinforcement learning, while powerful in scaling reasoning, is introduced only as the very last phase of post-training, preceded by supervised fine-tuning. While dominant, is this an optimal way of training? In this paper, we present RLP, an information-driven reinforcement pretraining objective, that brings the core spirit of reinforcement learning—exploration—to the last phase of pretraining. The key idea is to treat *chain-of-thought* as an exploratory action, with rewards computed based on the *information gain* it provides for predicting future tokens. This training objective essentially encourages the model to think for itself before predicting what comes next, thus teaching an independent thinking behavior earlier in the pretraining. More concretely, the reward signal measures the increase in log-likelihood of the next token when conditioning on both context and a sampled reasoning chain, compared to conditioning on context alone. This approach yields a verifier-free dense reward signal, allowing for efficient training for the full document stream during pretraining. Specifically, RLP reframes reinforcement learning for reasoning as a pretraining objective on ordinary text, bridging the gap between next-token prediction and the emergence of useful chain-of-thought reasoning. Pretraining with RLP on QWEN3-1.7B-BASE lifts the overall average across an eight-benchmark math-and-science suite by 19%. With identical post-training, the gains compound, with the largest improvements on reasoning-heavy tasks such as AIME25 and MMLU-Pro. Applying RLP to the hybrid NEMOTRON-NANO-12B-V2 increases the overall average from 42.81% to 61.32% and raises the average on scientific reasoning by 23%, demonstrating scalability across architectures and model sizes. Code: https://github.com/NVlabs/RLP

1. Introduction

Large Language Models (LLMs) pretrained with next-token prediction loss have demonstrated broad utility, but this objective does not explicitly encourage long-range reasoning or integration with world knowledge. Consequently, state-of-the-art models (Guo et al., 2025; Yang et al., 2025) rely on post-training objectives such as supervised fine-tuning (SFT) and reinforcement learning with human or verified feedback (RLHF, RLAIF, RLVR) (Ouyang et al., 2022; Lambert et al., 2024) to induce complex reasoning abilities. In contrast, human comprehension is not a linear token-by-token process, but rather a parallel integration of input with prior knowledge (Baumgaertner et al., 2002; Hagoort et al., 2004; Metzner et al., 2015). Current pretraining lacks such mechanisms, limiting the model's ability to reason and ground language in world knowledge during learning.

To fill this gap, we propose **R**einforcement **L**earning **P**re-training (RLP) which treats Chain-of-Thought (CoT) generation as an explicit action taken before predicting each next token. As shown in Fig.1, the model first samples an internal thought, then predicts the observed token from the same context augmented with that thought. The training signal is the increase in log-likelihood of the observed token when the thought is present compared to a no-think baseline. This yields a verifier-free and dense reward that assigns position-wise credit

† Equal contribution
* Work done during internship at NVIDIA

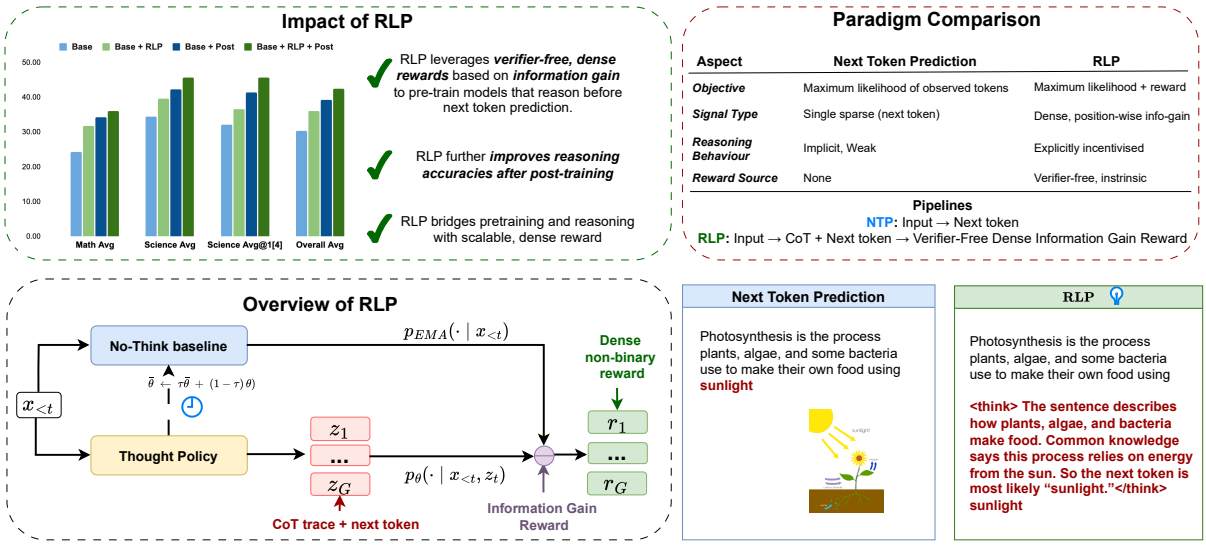


Figure 1: **Visualization of the RLP framework.** A chain-of-thought is sampled *before* next-token prediction. Rewards are computed by contrasting the predictor conditioned on the CoT with a *No-think* EMA baseline, yielding a verifier-free, dense signal. We list the advantages of RLP over the traditional pretraining objective (top right) and show the impact after end-to-end training (top left).

wherever thinking improves prediction. Because the signal is defined for ordinary text with teacher forcing, RLP reframes reinforcement learning for reasoning as reinforcement pretraining on the same streams used for maximum likelihood.

Unlike post-training with verifiable rewards, which requires task-specific checkers or curated solutions, RLP is verifier-free: the signal is computed directly from log-evidence under the model and a baseline, allowing uniform application to domain agnostic web-scale text. Compared to reinforcement pretraining via prefix-matching rewards (RPT) (Dong et al., 2025), which uses sparse binary reward and often relies on proxy-model filtering of "easy" tokens, RLP provides a continuous improvement signal at every position and trains on the full documents. This eliminates the need to preselect high-entropy tokens or couple training to a separate small model. Prior RPT demonstrations also depend on distilled checkpoints with strong prior reasoning ability, which clouds whether the method helps base models. RLP is designed to shape thinking in base models by rewarding only those thoughts that measurably help next-token prediction.

This work makes the following key contributions: We introduce **RLP, a verifier-free information-gain objective** that augments next-token prediction by rewarding thoughts in proportion to their predictive utility. We develop a **practical and stable training algorithm** that interleaves reinforcement updates with standard likelihood training via group-relative advantages, a clipped surrogate for thought tokens, and a slowly updated Exponential Moving Average (EMA) baseline. We provide **theoretical guarantees** linking expected reward to reductions in cross-entropy and to a computable lower bound, ensuring both interpretability and tractability. We conduct comprehensive experiments showing that RLP outperforms strong baselines, remains robust after strong post-training, generalizes across diverse corpora, and scales effectively to larger model sizes and hybrid architectures—establishing it as a broadly applicable reinforcement pretraining objective.

Our empirical validation is comprehensive, assessing the efficacy of RLP along four key axes. First, we evaluate its performance relative to traditional next-token prediction baselines. On the QWEN3-1.7B-BASE model, RLP outperforms continuous pretraining by $+17\%$ and RPT by nearly $+4\%$. We show the advantage persists even when the baseline uses $35\times$ **more data** to match FLOPs, confirming the gains arise from methodology rather than compute. Second, we demonstrate the robustness of these improvements, showing they are not transient. As shown in Fig.1, when subjected to an identical, strong post-training regimen, the foundational advantages of RLP **compound**, allowing our final model to surpass its conventionally trained counterparts by a significant 7–8% margin. Third, unlike methods requiring narrow, curated datasets, RLP successfully extracts a powerful reasoning signal from diverse, general-purpose web corpora–establishing its **versatility across data domains** (Table 4). Finally, we confirm its scalability and architecture-agnostic power.

When applied to a 12B hybrid Mamba-Transformer (NEMOTRON-NANO-12B-V2), RLP achieves a staggering **35% relative improvement** over a heavily trained baseline while using just 0.125% of the data—a testament to its remarkable data efficiency and broad applicability across LLM families and sizes.

2. Methodology

We introduce RLP, a pretraining-time procedure that explicitly induces reasoning. As illustrated in Fig. 1, RLP inserts a short Chain-of-Thought (CoT) *before* next-token prediction and measures how much that thought improves the model's log-probability of the observed token relative to a no-think baseline. This improvement, which is a log-likelihood ratio, is a verifier-free, dense reward available at every position in ordinary text corpora. By valuing thoughts in proportion to their predictive benefit, RLP turns reinforcement *pretraining* into learning to think on the same data used for standard next-token training.

Parameterization and roles.

We separate the components for clarity:

- **Thought policy / predictor** $\pi_\theta(c_t \mid x_{<t})$ and $p_\theta(x_t \mid x_{<t}, c_t)$ share *exactly the same* network and parameters θ. The network first samples a CoT c_t and then, conditioned on the concatenated prefix $(x_{<t}, c_t)$, scores the next token x_t.
- **No-think baseline** $\bar{p}_\phi(x_t \mid x_{<t})$ (parameters ϕ) is an EMA teacher of the current network used to score the same token without any CoT channel.

Thus, there is a single model that both *generates* the thought and *predicts* the next token given that thought; the EMA teacher provides the no-think counterfactual.

Classical next-token objective.

Given a text sequence $\boldsymbol{x} = (x_0, \ldots, x_T)$ and position t, the standard next-token objective for a predictor q_η is

$$\mathcal{L}_{\text{NTP}}(\eta) \;:=\; \mathop{\mathbb{E}}_{(x_{<t}, x_t)\sim\mathcal{D}}\big[\log q_\eta(x_t \mid x_{<t})\big]. \tag{1}$$

For distributions p and q on the next token, we define Cross-entropy (CE) as

$$\mathrm{CE}(p, q) \;\stackrel{\text{def}}{=}\; \mathop{\mathbb{E}}_{x\sim p}\big[-\log q(x)\big]. \tag{2}$$

Using $p^*(\cdot \mid x_{<t})$ for the data distribution over x_t, maximizing equation 1 is equivalent to minimizing $\mathbb{E}_{x_{<t}\sim\mathcal{D}}\big[\mathrm{CE}\big(p^*,\, q_\eta(\cdot \mid x_{<t})\big)\big]$. *We include equation 1 only for context as our training* ***does not*** *include a standard NTP loss term.* Instead, RLP optimizes an information-gain objective defined below and updates parameters *only through the tokens of the sampled thoughts*.

2.1. Reasoning as an action

RLP augments next-token prediction with a sampled thought. At each position t, the policy draws a latent CoT random variable

$$z_t \sim \pi_\theta(\cdot \mid x_{<t}),$$

and we write c_t for its realization. The network then predicts x_t with the *reasoned* scorer $p_\theta(\cdot \mid x_{<t}, c_t)$. As a no-think counterfactual we use $\bar{p}_\phi(\cdot \mid x_{<t})$, the EMA teacher queried on the same context without providing the CoT.

EMA teacher instantiation and schedule.

We instantiate the EMA teacher to match the current model on the *first* batch ($\phi \leftarrow \theta$), and thereafter update it *after* each optimizer step via

$$\phi \;\leftarrow\; \tau\,\phi + (1-\tau)\,\theta, \qquad \tau = 0.999.$$

This choice makes $\bar{p}_\phi$ a *moving counterfactual* that is (i) *current* enough to provide informative comparisons and (ii) *intentionally lagged* to mitigate reward hacking. If the baseline were frozen, the counterfactual would drift too far from the evolving model; if it tracked the model without lag, the log-likelihood ratio would collapse toward zero and invite degenerate strategies. The post-update averaging yields a one-step-lagged, smoothed teacher that stabilizes training.

2.2. Information-gain reward

With teacher forcing on the next token, define the reasoned and baseline log-evidence

$$S_{\text{pred}}(c_t) \;:=\; \log p_\theta\big(x_t \,\big|\, x_{<t}, c_t\big), \tag{3}$$

$$S_{\text{EMA}} \;:=\; \log \bar{p}_\phi\big(x_t \,\big|\, x_{<t}\big). \tag{4}$$

The *information-gain* reward is the log-likelihood ratio

$$r(c_t) \;:=\; S_{\text{pred}}(c_t) \;-\; S_{\text{EMA}}, \tag{5}$$

which compares the reasoned scorer with a no-think baseline on the observed next token. Rewards are computed under teacher forcing for each t. When updating the policy, we *treat* $r(c_t)$ *as a constant with respect to* θ (no backpropagation through p_θ or $\bar{p}_\phi$); see §2.4.

2.3. Expected improvement identity

Proposition 1 (CE reduction)**.** *For any fixed* $(x_{<t}, c_t)$*,*

$$\mathop{\mathbb{E}}_{x_t \sim p^*}[r(c_t)] \;=\; \mathrm{CE}\big(p^*, \bar{p}_\phi(\cdot \mid x_{<t})\big) \;-\; \mathrm{CE}\big(p^*, p_\theta(\cdot \mid x_{<t}, c_t)\big).$$

where $p^*(\cdot \mid x_{<t})$ is the data distribution over x_t. Maximizing the expected reward therefore maximizes the predictive usefulness of the thought for the next token.

Proposition 2 (Lower bound via marginalization over thoughts)**.** *Let* $\pi_\theta(z_t \mid x_{<t})$ *be the distribution over CoTs and define the collapsed predictor*

$$\tilde{p}_\theta(x \mid x_{<t}) \;=\; \mathop{\mathbb{E}}_{z_t \sim \pi_\theta(\cdot \mid x_{<t})}\big[p_\theta(x \mid x_{<t}, z_t)\big].$$

Then for any realized x_t*,*

$$\mathop{\mathbb{E}}_{c_t \sim \pi_\theta}\big[S_{pred}(c_t)\big] \;\le\; \log \tilde{p}_\theta(x_t \mid x_{<t}), \quad \textit{and} \quad J(\theta) \;=\; \mathbb{E}[r(c_t)] \;\le\; \mathbb{E}\left[\log \frac{\tilde{p}_\theta(x_t \mid x_{<t})}{\bar{p}_\phi(x_t \mid x_{<t})}\right].$$

The CoT-conditioned objective is thus a computable lower bound on the improvement one would obtain after marginalizing thoughts. Refer to §8.1 of the appendix for the proofs of the propositions.

2.4. RLP objective and optimization

RLP optimizes the thought policy to produce thoughts that *increase* predictive evidence. Our training *does not* include the standard next-token loss in equation 1. Instead, we optimize only the information-gain objective

$$\max_\theta \; J(\theta) \;=\; \mathop{\mathbb{E}}_{x_{<t} \sim \mathcal{D}} \; \mathop{\mathbb{E}}_{c_t \sim \pi_\theta(\cdot \mid x_{<t})}\big[\, r(c_t) \,\big], \tag{6}$$

or, equivalently, we *minimize* the negative information-gain loss $\mathcal{L}_{\text{IG}}(\theta) = -J(\theta)$. Gradients are applied only to the *thought tokens*; $r(c_t)$ is treated as a constant (no backpropagation through p_θ or $\bar{p}_\phi$) .

Algorithm 1 RLP for next-token prediction with information gain

Inputs: dataset $\mathcal{D}$, group size $G \geq 2$, clipping $(\epsilon_\ell, \epsilon_h)$, EMA decay $\tau \in (0,1)$, learning rate η.
Model: a single network with parameters θ used both as (i) thought policy π_θ and (ii) reasoned predictor p_θ; EMA baseline $\bar{p}_\phi$.
Initialization: mark ϕ as uninitialized.
while training **do**
  Set the behavior snapshot $\theta_{\text{old}} \leftarrow \theta$. ▷ used for the current sampling pass
  Sample minibatch $\{(x_{<t}^{(b)}, x_t^{(b)})\}_{b=1}^{B} \sim \mathcal{D}$.
  For each b, sample G thoughts $c_t^{(b,i)} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x_{<t}^{(b)})$ with $|c_t^{(b,i)}| \geq 1$.
  if ϕ is uninitialized **then**
    $\phi \leftarrow \theta$ ▷ lazy init of EMA teacher
  Compute baseline log-evidence (teacher forcing, no grad) $S_{\text{EMA}}^{(b)}$ as per equation 3.
  Compute reasoned log-evidence $S_{\text{pred}}^{(b,i)}$ and rewards $r^{(b,i)}$ as per equation 3 and equation 5.
  Group baseline $\bar{r}^{(b)}$ and $A^{(b,i)}$ (inclusive mean with correction; sg is stop-grad) as per equation 7.
  Per-token importance ratios and clipped surrogate for $\ell_u^{(b,i)}$ with prefix $\text{prefix}_u^{(b,i)}$:
  $\rho_u^{(b,i)} = \exp\Big(\log \pi_\theta(\ell_u^{(b,i)} \mid \text{prefix}_u^{(b,i)}) - \log \pi_{\theta_{\text{old}}}(\ell_u^{(b,i)} \mid \text{prefix}_u^{(b,i)})\Big)$.
  $L_{\text{clip}}^{(b,i)} = -\frac{1}{|c_t^{(b,i)}|} \sum_u \min\Big(\rho_u^{(b,i)}\ \text{sg}\big(A^{(b,i)}\big),\ \text{clip}(\rho_u^{(b,i)}; 1-\epsilon_\ell, 1+\epsilon_h)\ \text{sg}\big(A^{(b,i)}\big)\Big)$.
  Policy update on thought tokens:
  $\mathcal{L}(\theta) = \frac{1}{BG} \sum_{b=1}^{B} \sum_{i=1}^{G} L_{\text{clip}}^{(b,i)}, \quad \theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$.
  EMA update of baseline: $\phi \leftarrow \tau\, \phi + (1-\tau)\, \theta$.
Output: trained policy/predictor (shared θ) and EMA baseline ϕ.

Group-relative baseline (inclusive mean with correction).

To reduce variance, for each context we sample $G \geq 2$ thoughts $\{c_t^{(i)}\}_{i=1}^{G}$ and use a corrected inclusive mean baseline. Let

$$\bar{r} \;=\; \frac{1}{G} \sum_{j=1}^{G} r\big(c_t^{(j)}\big).$$

We define the advantages

$$A^{(i)} \;:=\; \frac{G}{G-1}\Big(r\big(c_t^{(i)}\big) - \bar{r}\Big), \qquad \text{with no gradient propagated through } \bar{r}. \tag{7}$$

This multiplicative factor removes the $\big(1 - \frac{1}{G}\big)$ shrinkage inherent to the inclusive mean, yielding an unbiased estimator with low variance.

Per-token importance ratios and clipped surrogate.

We update the log-probability of the *thought* tokens with a clipped surrogate. Let $\ell_u^{(i)}$ be the u-th token in $c_t^{(i)}$ and $\text{prefix}_u^{(i)} = (x_{<t}, \ell_{1:u-1}^{(i)})$. With behavior parameters θ_{old} used to sample the thoughts, define the per-token importance ratio

$$\rho_u^{(i)} = \exp\Big(\log \pi_\theta(\ell_u^{(i)} \mid \text{prefix}_u^{(i)}) - \log \pi_{\theta_{\text{old}}}(\ell_u^{(i)} \mid \text{prefix}_u^{(i)})\Big).$$

We write $\text{clip}(\rho; 1-\epsilon_\ell, 1+\epsilon_h)$ for elementwise clipping and denote stop-gradient by $\text{sg}(\cdot)$. The surrogate loss is

$$\mathcal{L}_{\text{clip}}(\theta) \;=\; -\,\mathbb{E}\left[\frac{1}{|c_t^{(i)}|} \sum_u \min\Big(\rho_u^{(i)}\ \text{sg}(A^{(i)}),\ \text{clip}(\rho_u^{(i)}; 1-\epsilon_\ell, 1+\epsilon_h)\ \text{sg}(A^{(i)})\Big)\right]. \tag{8}$$

Benchmark	$\mathcal{M}_{\text{base}}$	$\mathcal{M}_{\text{CPT}}$	$\mathcal{M}_{\text{RLP}}$	$\mathcal{M}_{\text{base}}$ +Post	$\mathcal{M}_{\text{CPT}}$ +Post	$\mathcal{M}_{\text{RLP}}$ +Post
AIME25	2.25	3.96	**5.02**	5.32	5.89	**7.05**
MATH500	48.45	57.52	**58.48**	61.92	62.70	**64.30**
GSM8K	54.16	72.85	**74.48**	78.22	78.70	**80.50**
AMC23	25.94	31.25	**31.25**	35.00	34.38	**36.50**
Minerva	15.30	19.03	**21.19**	25.30	26.10	**27.80**
MMLU	50.08	41.95	**56.14**	58.36	59.00	**61.50**
MMLU@1[4]	44.85	40.00	**52.18**	56.00	58.53	**61.00**
MMLU-Pro	28.17	27.81	**34.62**	37.85	39.92	**42.40**
MMLU-Pro@1[4]	23.95	24.61	**30.80**	36.53	38.49	**41.30**
GPQA	25.25	26.26	**28.28**	30.93	29.27	**33.33**
GPQA@1[4]	27.52	24.75	**27.02**	31.52	30.01	**34.97**
Math Avg	24.35	30.77	**31.74**	34.29	34.63	**36.03**
Science Avg	34.50	32.01	**39.68**	42.38	42.73	**45.74**
Science Avg@1[4]	32.11	29.79	**36.67**	41.35	42.34	**45.76**
Overall	30.32	30.85	**36.03**	39.34	39.90	**42.51**

Table 1: Quantitative benchmarks for Qwen3-1.7B-Base, showing the impact of RLP. Shaded columns indicate RLP variants; "Post" indicates SFT + RLVR post-training.

2.5. Reward properties and guarantees

Does thinking actually help?

The reward $r(c_t)$ is positive exactly when the model that used the sampled thought assigns higher probability to the observed next token than the EMA baseline that did not think. In expectation over the data distribution, this equals the reduction in cross-entropy between the reasoned scorer and the no-think baseline (Prop. 1).

Positionwise credit at every step.

Since the task is next-token prediction, the reward is computed independently at each position t as

$$r(c_t) \;=\; \log p_\theta(x_t \mid x_{<t}, c_t) \;-\; \log \bar{p}_\phi(x_t \mid x_{<t}).$$

Credit is attached exactly where the thought changes predictive probability, yielding one scalar per token and removing the need for a learned value function or any external verifier.

Putting it all together.

Algorithm 1 composes the above pieces into a single training loop. Specifically, multiple thoughts are sampled per position and information-gain rewards are computed against a moving EMA counterfactual. Group-relative advantages are formed and the shared network is updated *only* on the thought tokens via the clipped surrogate in equation 8. In this case, the improvements originate from learning to generate thoughts that systematically raise predictive evidence.

3. Experimental Setup

We experiment with QWEN3-1.7B-BASE (Yang et al., 2025) and then scale our experiments to a larger NEMOTRON-NANO-12B-V2 (Nano, 2025) model.[1]

RLP.

We apply RLP on a diverse set of datasets across two settings: (i) *SFT-style reasoning corpora*, including a math-centric set (OmniMath (Gao et al., 2024)) and mixed math + general-reasoning sets (OpenThoughts (Guha et al., 2025), Nemotron-Crossthink (Akter et al., 2025)); and (ii) *general-purpose pretraining corpora*, covering academic papers (ACAD), math textbooks (Math-Text), and open-ended web pages QA pairs from

[1]Details about hyper-parameters for each of the below phases can be found in §9.

Common Crawl (Web-Crawl)(Nano, 2025). We train with RLP for 1B tokens using general pretraining corpora ($\mathcal{D}_{\text{PT}}$) to evaluate its effect in an end-to-end LLM pretraining pipeline. We denote this model as $\mathcal{M}_{\text{RLP}}$.

Continuous Pretraining.

To ensure compute equivalent comparison with $\mathcal{M}_{\text{RLP}}$, we do continuous pretraining on the base model denoted by $\mathcal{M}_{\text{base}}$ with the same tokens used in RLP. We denote this model as $\mathcal{M}_{\text{CPT}}$. This serves as an additional baseline for our experiments.

Post-Training.

All models undergo a SFT stage on OpenThoughts data (Guha et al., 2025). To further enhance, we apply Reinforcement Learning with Verifier Rewards (RLVR) using MATH dataset (Hendrycks et al., 2021b). This two-stage post-training pipeline provides an evaluation framework to verify that gains from RLP persist under strong alignment, while also revealing how much additional improvement can be achieved through subsequent post-training. For consistency, all models are trained with identical SFT and RLVR receipes, ensuring that any observed differences in downstream accuracies can be attributed to the pretraining condition ($\mathcal{M}_{\text{base}}$ vs $\mathcal{M}_{\text{CPT}}$ vs $\mathcal{M}_{\text{RLP}}$).

3.1. Evaluation Metrics

We conduct a thorough benchmark assessment using a series of tasks using NeMo-Skills[2].

Math Reasoning (MATH AVG). We consider four diverse math benchmarks : GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021c), Minerva Math (Lewkowycz et al., 2022), AMC23. We report Pass@1 average of 8 runs for these.

Science Reasoning (SCIENCE AVG). For conceptual science and specialized knowledge, we evaluate on MMLU (Hendrycks et al., 2021a), MMLU-Pro (Wang et al., 2024), and the graduate-level STEM benchmark GPQA-Diamond (Rein et al., 2024). For science benchmarks, we report the average greedy and Pass@1 scores from 4 runs (SCIENCE AVG@1[4]).

4. Results

Table 1 reports the performance of QWEN3-1.7B-BASE under different pretraining and post-training objectives. First, RLP consistently outperforms both the $\mathcal{M}_{\text{base}}$ and $\mathcal{M}_{\text{CPT}}$ across nearly all benchmarks, with especially strong gains on reasoning-heavy tasks such as AIME25 and MMLU-Pro. We see that $\mathcal{M}_{\text{RLP}}$ is relatively on average 19% and 17% better than $\mathcal{M}_{\text{base}}$ and $\mathcal{M}_{\text{CPT}}$ respectively. This highlights the effectiveness of dense, verifier-free reinforcement signals for instilling reasoning capabilities during pretraining. Second, the benefits of RLP persist even after strong post-training (SFT + RLVR). While all models improve after post-training, $\mathcal{M}_{\text{RLP}}$ achieves the highest scores with the overall average substantially higher than both $\mathcal{M}_{\text{base}}$ by 8% and $\mathcal{M}_{\text{CPT}}$ by 7% relatively. This indicates that RLP establishes robust reasoning foundations that are not washed out by downstream alignment but instead compound with post-training. We observe particularly large gains in science domains, with $\mathcal{M}_{\text{RLP}}$ +Post achieving $+3$ points over $\mathcal{M}_{\text{CPT}}$ +Post. This trend suggests that RLP is not limited to mathematical reasoning but also generalizes effectively to other domains. The ability to strengthen performance in science benchmarks highlights that RLP fosters a broader class of multi-step explanation-driven reasoning skills, moving beyond domain-specific improvements and pointing toward a more versatile foundation for reasoning in LLMs. Overall, the results demonstrate that RLP not only induces reasoning ability during pretraining but

Model	Math Avg	Science Avg	Science Avg@1[4]	Avg
$\mathcal{M}_{\text{base}}$	61.38	34.51	32.54	42.81
$\mathcal{M}_{\text{RLP}}$	**65.33**	**57.26**	**61.37**	**61.32**

Table 2: Comparison of $\mathcal{M}_{\text{base}}$ and RLP for the NEMOTRON-NANO-12B-V2 model.

[2] https://github.com/NVIDIA/NeMo-Skills

also synergizes with post-training, leading to models with stronger and more durable reasoning abilities than those trained with next-token prediction or continuous pretraining.

Scaling Model Size and Architecture:

We further scale RLP to NEMOTRON-NANO-12B-V2 (Nano, 2025) ($\mathcal{M}_{\text{base}}$), a hybrid Mamba-Transformer language model of 12B parameter size. In this comparison we take an intermediate checkpoint of NEMOTRON-NANO-12B-V2 trained till 19.8 trillion tokens and apply RLP for 250 million tokens only. $\mathcal{M}_{\text{base}}$ on the other hand is trained for 20 trillion tokens. Table 2 demonstrates that the benefits of RLP persist and even amplify when scaling to larger model sizes and generalizes to different model architectures. On the NEMOTRON-NANO-12B-V2 model, $\mathcal{M}_{\text{RLP}}$ substantially outperforms $\mathcal{M}_{\text{base}}$ across all domains, and particularly $\mathcal{M}_{\text{RLP}}$ is relatively 35% on average better than $\mathcal{M}_{\text{base}}$ inspite of being trained on approx. 200 billion less tokens. While math performance improves moderately, the most striking gains emerge in science reasoning, where Science Avg jumps an absolute 23%. These results highlight that RLP yields not only stronger math performance but also robust cross-domain reasoning capabilities.

RPT Comparison

Following the experimental setup in RPT with Omni-MATH dataset, we trained both methods for one epoch under matched data and compute budgets before evaluating on our benchmark suite. As summarized in Table Table 3, RLP achieves uniformly higher aggregates: *Math Avg* improves from 47.50 to 49.62 (+2.12; +4.5% relative), *Science Avg* from 35.88 to 37.07 (+1.19; +3.3%), and *Overall Avg* from 41.69 to 43.35 (+1.66; +4.0%). Methodologically, RPT applies reinforcement only to tokens pre-selected by an auxiliary assistant via entropy filtering and optimizes a sparse, binary next-token correctness signal that ignores the CoT content, limiting where the signal can be applied. In contrast, RLP evaluates each sampled CoT by the information gain it provides for the observed next token and updates at all positions without an auxiliary filter which yields consistently better averages under the matched setting above. Crucially, this dense, per-token information-gain reward supplies richer credit assignment than RPT's sparse binary signal and, in our matched experiments, empirically yields better performance.

Model	Math Avg	Science Avg	Avg
$\mathcal{M}_{\text{base}}$	35.96	32.11	34.03
$\mathcal{M}_{\text{RPT}}$	47.50	35.88	41.69
$\mathcal{M}_{\text{RLP}}$	**49.62**	**37.07**	**43.35**

Table 3: **RLP outperforms RPT across all averages.** QWEN3-1.7B-BASE was trained with both RPT and RLP for one epoch with matched data and compute.

5. Ablations

Does RLP provide generalizable improvements across diverse corpora?

A key advantage of RLP is its scalability to large, diverse corpora, unlike RLVR, which relies on small, curated reasoning datasets and raises concerns about generalizability. Prior work (Chen et al., 2025; Setlur et al., 2025) highlights the need for complex reasoning corpora to sustain RL improvements, but such datasets are costly to curate and impractical at pretraining scale. For these ablations, we apply RLP to QWEN3-1.7B-BASE for 200 steps—utilizing 170M input tokens—holding the rest of the setup fixed.

As illustrated in Table 4, RLP delivers consistent gains across all corpus families, eliminating concerns that RL based pretraining only benefits curated reasoning data. Relative to $\mathcal{M}_{\text{base}}$ average improves by 7-9% with strongest gains on Nemotron-Crossthink (SFT-style) and Web-Crawl (general-purpose corpora). Unlike prior work (Akter et al., 2025), where RL gains were limited to math and weakened under mixed data, RLP achieves simultaneous improvements across all benchmarks, demonstrating genuine cross-domain transfer. Even on purely non-reasoning general corpora such as web-crawl, RLP extracts a reasoning signal that scales with data diversity (Appendix 11). Table 4 illustrates that unlike prior work (Liu et al., 2025b; Zhou et al., 2025), RLP can be applied to any data format like academic papers, textbooks, web-crawl as well as SFT style data. Overall, RLP is scalable, domain-agnostic pre-training augmentation that enhances both reasoning and accuracy.

Model	Dataset	Type	Math Avg	Science Avg	Science Avg@1[4]	Avg
$\mathcal{M}_{\text{base}}$	-	-	35.96	34.50	32.11	34.19
$\mathcal{M}_{\text{CPT}}$	Nemotron-Crossthink [170M]	Equal Input Token	37.11	35.76	32.15	35.01
	Nemotron-Crossthink [6B]	Equal FLOPs	43.90	37.74	32.47	38.04
	$\mathcal{D}_{\text{PT}}$[1B]	PT Data Mix	45.34	32.14	29.33	35.60
$\mathcal{M}_{\text{RLP}}$	OmniMath [170M]	SFT	46.48	40.27	37.54	41.43
	OpenThoughts [170M]		47.64	40.84	35.88	41.45
	Nemotron-Crossthink [170M]		49.76	42.54	37.78	**43.36**
	ACAD [170M]	General	47.68	40.59	36.87	41.71
	Math-Text [170M]		48.07	40.46	36.32	41.62
	Web-Crawl [170M]		48.87	40.75	36.77	**42.13**
	$\mathcal{D}_{\text{PT}}$[1B]	PT Data Mix	46.35	39.68	36.67	40.90

Table 4: **RLP across diverse corpora.** RLP trained on six SFT-style and general-purpose datasets yields consistent gains, indicating transferable reasoning from mixed/open-ended data.

Does the improvement sustain under compute equivalent baselines?

A critical question is whether RLP's gains stem from its unique RL-based pretraining or simply higher compute. Standard next-token pretraining quantifies compute by input tokens, but RLP adds rollout costs not captured by this metric. For fair comparison, we evaluate against $\mathcal{M}_{\text{CPT}}$ baselines under: (a) equal Input Tokens Seen and (b) equal total Compute FLOPs. RLP is fixed to $T_{inp} = 170$M tokens; the token-matched $\mathcal{M}_{\text{CPT}}$ [170M] continues pretraining on 170M tokens (Input Token), while the FLOP-matched budget corresponds to 6B tokens for CPT ($\mathcal{M}_{\text{CPT}}$ [6B])(see Appendix 10).

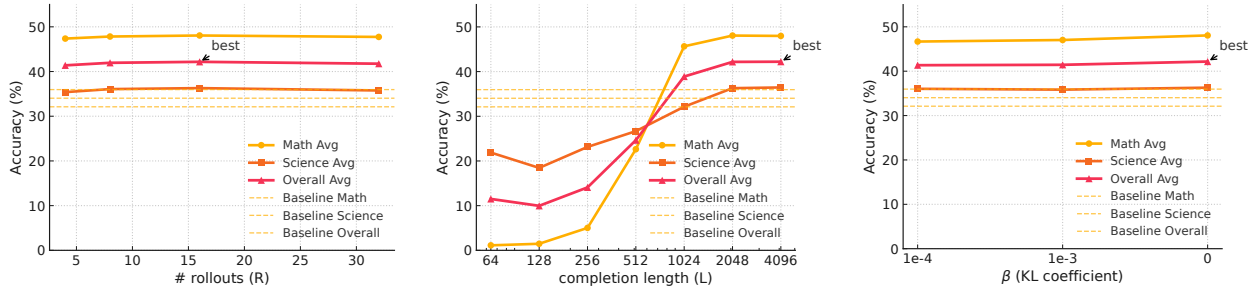
In Table 4, $\mathcal{M}_{\text{RLP}}$ outperforms $\mathcal{M}_{\text{CPT}}$ trained on the same 170M tokens and maintains a clear advantage even against a compute-matched $\mathcal{M}_{\text{CPT}}$ exposed to 6B tokens (35× more data). Despite this disparity, RLP achieves a 5.3% gain on average (compare $\mathcal{M}_{\text{CPT}}$ Nemotron-Crossthink [6B] vs $\mathcal{M}_{\text{RLP}}$ Nemotron-Crossthink [170M]), with consistent improvements across math and science benchmarks. These results show that RLP's gains stem not from more efficient use of compute, not larger budgets, validating the effectiveness of our approach.

Is RLP comparable to CPT with high-quality reasoning data?

High-quality reasoning corpora have shown to substantially boost base model reasoning ability when used in continuous pretraining (CPT) or mid-training (Wang et al., 2025; Gandhi et al., 2025). This raises the important question of whether CPT can match or even surpass RLP under such favorable conditions. To investigate this, we conduct CPT on both reasoning-centric, Nemotron-Crossthink and general pretraining ($\mathcal{D}_{\text{PT}}$) datasets, each using 170M tokens. Our results in Table 4 show that even with high quality reasoning data, RLP consistently outperforms CPT by a significant margin. Specifically, $\mathcal{M}_{\text{RLP}}$ outperforms $\mathcal{M}_{\text{CPT}}$, showing an average gain of 8% on Nemotron-Crossthink and 5% on pre-training data mix ($\mathcal{D}_{\text{PT}}$) on 1B tokens. These results highlight two key insights. First, while CPT benefits from reasoning-dense corpora, it remains sensitive to domain skew—evident in the weak science accuracy on $\mathcal{D}_{\text{PT}}$—whereas RLP generalizes more evenly across disciplines. Second, the consistent margin by which RLP outperforms CPT, even in the presence of high quality reasoning data, underscores that the gains of RLP are not merely due to data quality but stem from the algorithmic design itself. This reinforces the conclusion that RLP provides a generalizable mechanism for leveraging reasoning data during pretraining, complementing rather than being overshadowed by high-quality corpus selection.

Ablations on rollout count, completion length, and KL weight.

Fig. 2 visualizes the trends across three settings: (a) rollouts, (b) completion length, and (c) KL. Please look into §10 for more detailed numbers and per-task breakdowns. More rollouts help up to $G = 16$ (*Overall* 42.17%); $G = 4$ and 8 already reach 41.38% and 41.95%, while $G = 32$ decreases slightly to 41.75% (Fig. 2a). Increasing completion length gives the largest gains. Specifically, *overall* rises from 11.50% at 64 to 42.17% at 2048, with *Math/Science* moving from $1.12\%/21.88\%$ to $48.06\%/36.29\%$ (Fig. 2b). Extending to 4096 yields 42.21% at roughly twice the thought budget, so we default to 2048. Furthermore, a KL anchor does not help. Specifically,



(a) Number of rollouts (b) completion length (c) KL coefficient β

Figure 2: **Ablations on Qwen3-1.7B.** Curves report Math/Science/Overall averages. Dashed lines mark the base model.

$\beta = 10^{-4}$ and 10^{-3} give 41.35% and 41.44%, compared to 42.17% at $\beta = 0$, and it also increases memory and step time (Fig. 2c). We therefore use $G = 16$, completion length 2048, and $\beta = 0$ in later experiments.

6. Related Work

Next-Token Prediction.

Next-token prediction is the standard pretraining objective for LLMs: predict the next word from prior context (Shannon, 1951; Bengio et al., 2003). Scaling it with Transformers (Vaswani et al., 2017) enabled landmark and state-of-the-art systems (Radford et al., 2018; Brown et al., 2020; Smith et al., 2022; Bi et al., 2024; Nano, 2025; Yang et al., 2025). Anticipating tokens across corpora induces syntactic, semantic, and pragmatic structure that transfers broadly. Alternatives include masked language modeling (Devlin et al., 2019) and span corruption (Raffel et al., 2020), but next-token prediction remains dominant for its alignment with left-to-right generation and strong downstream accuracy across tasks. In this work. we add a verifier-free dense reward during pretraining that leverages reasoning before prediction.

Verifier-Free Rewards in Post-Training.

Recent work explores verifier-free rewards. Yuan et al. (2024) uses iterative DPO where, after SFT, the model judges its own candidates to create preference pairs. Liu et al. (2025b) trains with incentive RL on SFT corpora. Zhao et al. (2025) proposes RL from an internal feedback while using the model's confidence as reward. RLP, in contrast, is a GRPO-style pretraining objective. It operates on any text data including web-crawl, academic papers and SFT datasets and optimizes continuation quality beyond next-token prediction. Because these methods target post-training policies, direct comparisons are not well-posed.

7. Conclusion

We introduce RLP, a reinforcement pretraining objective that rewards chain-of-thought by its information gain for next-token prediction. Unlike traditional approaches that defer RL to post-training, RLP instills reasoning during pretraining, yielding gains that persist and compound after alignment. Experiments across datasets, domains, and architectures show that RLP consistently outperforms compute-matched baselines and scales efficiently to large hybrid models, establishing reinforcement pretraining as a principled and general alternative to likelihood-only training.

References

Syeda Nahida Akter, Shrimai Prabhumoye, Matvei Novikov, Seungju Han, Ying Lin, Evelina Bakhturina, Eric Nyberg, Yejin Choi, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-crossthink: Scaling self-learning beyond math reasoning, 2025. URL https://arxiv.org/abs/2504.13941. 6, 8

Annette Baumgaertner, Cornelius Weiller, and Christian Büchel. Event-related fmri reveals cortical sites involved in contextual sentence integration. *Neuroimage*, 16(3):736–745, 2002. 1

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003. 10

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024. 10

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. 10

Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron: Advancing math and code reasoning through reinforcement learning, 2025. URL https://arxiv.org/abs/2505.16400. 8

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168. 7

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. 10

Qingxiu Dong, Li Dong, Yao Tang, Tianzhu Ye, Yutao Sun, Zhifang Sui, and Furu Wei. Reinforcement pre-training. *arXiv preprint arXiv:2506.08007*, 2025. 2

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025. URL https://arxiv.org/abs/2503.01307. 9

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-math: A universal olympiad level mathematic benchmark for large language models, 2024. URL https://arxiv.org/abs/2410.07985. 6

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models, 2025. URL https://arxiv.org/abs/2506.04178. 6, 7, 16

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. 1

Peter Hagoort, Lea Hald, Marcel Bastiaansen, and Karl Magnus Petersson. Integration of word meaning and world knowledge in language comprehension. *science*, 304(5669):438–441, 2004. 1

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021a. URL https://arxiv.org/abs/2009.03300. 7

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. *arXiv preprint arXiv:2103.03874*, 2021b. 7, 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. *NeurIPS*, 2021c. 7

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, 2025. URL https://arxiv.org/abs/2503.24290. 17

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1. 16

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024. 1

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. 7

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models, 2025a. URL https://arxiv.org/abs/2505.24864. 17

Wei Liu, Siya Qi, Xinyu Wang, Chen Qian, Yali Du, and Yulan He. Nover: Incentive training for language models via verifier-free reinforcement learning. *arXiv preprint arXiv:2505.16022*, 2025b. 8, 10

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7. 16

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li 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. 17

Paul Metzner, Titus von der Malsburg, Shravan Vasishth, and Frank Rösler. Brain responses to world knowledge violations: A comparison of stimulus-and fixation-triggered event-related potentials and neural oscillations. *Journal of Cognitive Neuroscience*, 27(5):1017–1028, 2015. 1

NVIDIA Nemotron Nano. Efficient hybrid mamba-transformer reasoning model. *arXiv preprint arXiv:2508.14444*, 2025. 6, 7, 8, 10

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. 1

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 10

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. 10

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. URL https://openreview.net/forum?id=Ti67584b98. 7

Amrith Setlur, Matthew Y. R. Yang, Charlie Snell, Jeremy Greer, Ian Wu, Virginia Smith, Max Simchowitz, and Aviral Kumar. e3: Learning to explore enables extrapolation of test-time compute for llms, 2025. URL https://arxiv.org/abs/2506.09026. 8

Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951. 10

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019. 16

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022. 10

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 10

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, 2024. URL https://arxiv.org/abs/2406.01574. 7

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. Octothinker: Revisiting mid-training in the era of rl scaling. https://tinyurl.com/OctoThinker, 2025. Notion Blog. 9

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. 1, 6, 10

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 3, 2024. 10

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025. 10

Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. Reinforcing general reasoning without verifiers. *arXiv preprint arXiv:2505.21493*, 2025. 8

8. Appendix

8.1. Proofs

In this section, we provide the proofs supporting the methodology in §2. We first prove the tokenwise cross-entropy (CE) reduction identity (Prop. 1), then the lower bound via marginalization over thoughts (Prop. 2). Finally, we state and prove Prop. 3, which formalizes the positionwise-credit claim described in §2.5: under teacher forcing, averaging the expected tokenwise information-gain rewards across positions recovers the expected per-token sequence-level CE improvement.

For convenience, we recall the key definitions from the main text: the reasoned and baseline log-evidence $S_{\text{pred}}(c_t) = \log p_\theta(x_t \mid x_{<t}, c_t)$ and $S_{\text{EMA}} = \log \bar{p}_\phi(x_t \mid x_{<t})$ (equation 3); the information-gain reward $r(c_t) = S_{\text{pred}}(c_t) - S_{\text{EMA}}$ (equation 5); and the cross-entropy $\mathrm{CE}(p, q) \overset{\text{def}}{=} \mathbb{E}_{x\sim p}\big[-\log q(x)\big]$ (equation 2).

8.2. Proof of Proposition 1 (Expected improvement identity)

Proof of Proposition 1. Fix the context $x_{<t}$ and a realized thought c_t, and let $p_t^*(x) := p^*(x \mid x_{<t})$ denote the data distribution over x_t at this position. By the reward definition equation 5 together with equation 3,

$$r(c_t) = \log p_\theta\big(x_t \mid x_{<t}, c_t\big) - \log \bar{p}_\phi\big(x_t \mid x_{<t}\big).$$

Taking expectation with respect to $x_t \sim p_t^*$ and using linearity of expectation,

$$\mathop{\mathbb{E}}_{x_t\sim p_t^*}\big[r(c_t)\big] = \mathop{\mathbb{E}}_{x_t\sim p_t^*}\Big[\log p_\theta(x_t \mid x_{<t}, c_t)\Big] \;-\; \mathop{\mathbb{E}}_{x_t\sim p_t^*}\Big[\log \bar{p}_\phi(x_t \mid x_{<t})\Big].$$

By the definition of cross-entropy equation 2, $\mathrm{CE}(p, q) = \mathbb{E}_{x\sim p}[-\log q(x)]$, so each expectation of a log-likelihood equals the negative cross-entropy:

$$\mathop{\mathbb{E}}_{x_t\sim p_t^*}\Big[\log p_\theta(x_t \mid x_{<t}, c_t)\Big] = -\,\mathrm{CE}\big(p^*,\, p_\theta(\cdot \mid x_{<t}, c_t)\big), \qquad \mathop{\mathbb{E}}_{x_t\sim p_t^*}\Big[\log \bar{p}_\phi(x_t \mid x_{<t})\Big] = -\,\mathrm{CE}\big(p^*,\, \bar{p}_\phi(\cdot \mid x_{<t})\big).$$

Substituting into the previous display yields

$$\mathop{\mathbb{E}}_{x_t\sim p_t^*}\big[r(c_t)\big] = \mathrm{CE}\big(p^*, \bar{p}_\phi(\cdot \mid x_{<t})\big) \;-\; \mathrm{CE}\big(p^*, p_\theta(\cdot \mid x_{<t}, c_t)\big),$$

which is the desired identity. □

8.3. Proof of Proposition 2 (Lower bound via marginalization over thoughts)

Proof of Proposition 2. Fix $(x_{<t}, x_t)$ and recall $S_{\text{pred}}(c_t) = \log p_\theta(x_t \mid x_{<t}, c_t)$ and $\tilde{p}_\theta(x \mid x_{<t}) = \mathbb{E}_{z_t\sim\pi_\theta(\cdot\mid x_{<t})}\big[p_\theta(x \mid x_{<t}, z_t)\big]$.

(i) Jensen bound.

Conditioning on $(x_{<t}, x_t)$ and taking expectation over $c_t \sim \pi_\theta(\cdot \mid x_{<t})$,

$$\mathbb{E}_{c_t\sim\pi_\theta}\big[S_{\text{pred}}(c_t)\big] = \mathbb{E}_{c_t}\Big[\log p_\theta(x_t \mid x_{<t}, c_t)\Big] \;\le\; \log \mathbb{E}_{c_t}\Big[p_\theta(x_t \mid x_{<t}, c_t)\Big] = \log \tilde{p}_\theta(x_t \mid x_{<t}),$$

where the inequality is Jensen's inequality applied to the concave function $\log(\cdot)$. This proves (i) pointwise for the realized x_t.

(ii) Bound on $J(\theta)$.

By definition of the reward in equation 5 and teacher forcing (see equation 3),

$$J(\theta) \;=\; \mathbb{E}\Big[\mathbb{E}_{c_t\sim\pi_\theta}\big[S_{\text{pred}}(c_t)\big] \;-\; S_{\text{EMA}}\Big] \;\le\; \mathbb{E}\Big[\log \tilde{p}_\theta(x_t \mid x_{<t}) \;-\; \log \bar{p}_\phi(x_t \mid x_{<t})\Big] \;=\; \mathbb{E}\left[\log \frac{\tilde{p}_\theta(x_t \mid x_{<t})}{\bar{p}_\phi(x_t \mid x_{<t})}\right],$$

where the inequality uses part (i) and the outer expectation is over $(x_{<t}, x_t) \sim \mathcal{D}$. This proves (ii).

Tightness.

Equality in (i) (and hence in (ii)) holds precisely when $p_\theta(x_t \mid x_{<t}, c_t)$ is almost surely constant in c_t under $\pi_\theta(\cdot \mid x_{<t})$ (e.g., when the predictor ignores the thought or when the thought policy is degenerate).

□

8.4. Tokenwise–to–sequence connection under teacher forcing (positionwise credit)

This subsection formalizes the claim in §2.5 that summing positionwise CE improvements recovers the sequence-level (per-token) improvement. The following proposition is *new to the appendix* and not required elsewhere; it clarifies how tokenwise rewards aggregate at the sequence level under teacher forcing.

Proposition 3 (Tokenwise–to–sequence connection under teacher forcing)**.** *Let $\boldsymbol{x} = (x_1, \ldots, x_T)$ be drawn from the data distribution $p^*(\boldsymbol{x})$ and fix a policy $\pi_\theta(c_t \mid x_{<t})$, the reasoned scorer $p_\theta(\cdot \mid x_{<t}, c_t)$, and the no-think baseline $\bar{p}_\phi(\cdot \mid x_{<t})$. Define the sequence-level (per-token) cross-entropy for the baseline and the (stochastic) reasoned scorer by*

$$\mathrm{CE}_{\mathrm{seq}}\big(p^*, \bar{p}_\phi\big) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[-\frac{1}{T}\sum_{t=1}^{T}\log \bar{p}_\phi(x_t \mid x_{<t})\right],$$

$$\mathrm{CE}_{\mathrm{seq}}\big(p^*, p_\theta[\pi_\theta]\big) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[-\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{c_t\sim\pi_\theta(\cdot\mid x_{<t})}\big[\log p_\theta(x_t \mid x_{<t}, c_t)\big]\right].$$

Then the average over positions of the expected *tokenwise information-gain rewards equals the per-token sequence-level CE improvement of the reasoned scorer against the baseline:*

$$\mathbb{E}_{\boldsymbol{x}}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{c_t\sim\pi_\theta(\cdot\mid x_{<t})}\,\mathbb{E}_{x_t\sim p^*(\cdot\mid x_{<t})}\big[r(c_t)\big]\right] \;=\; \mathrm{CE}_{\mathrm{seq}}\big(p^*, \bar{p}_\phi\big) \;-\; \mathrm{CE}_{\mathrm{seq}}\big(p^*, p_\theta[\pi_\theta]\big)\,.$$

Proof. **(i) Conditional independence under teacher forcing.** At position t, teacher forcing samples the target token from the data channel while the thought is sampled from the policy given the same prefix:

$$x_t \sim p^*(\cdot \mid x_{<t}), \qquad c_t \sim \pi_\theta(\cdot \mid x_{<t}).$$

Hence

$$p(c_t, x_t \mid x_{<t}) = \pi_\theta(c_t \mid x_{<t})\, p^*(x_t \mid x_{<t}), \quad \text{i.e.} \quad c_t \perp x_t \mid x_{<t}.$$

This implies $\mathbb{E}_{x_t\sim p^*(\cdot\mid x_{<t}, c_t)}[\cdot] = \mathbb{E}_{x_t\sim p^*(\cdot\mid x_{<t})}[\cdot]$.

(ii) Positionwise CE reduction. By Proposition 1, for any fixed $(x_{<t}, c_t)$,

$$\mathbb{E}_{x_t\sim p^*(\cdot\mid x_{<t})}\big[r(c_t)\big] = \mathrm{CE}\big(p^*, \bar{p}_\phi(\cdot \mid x_{<t})\big) - \mathrm{CE}\big(p^*, p_\theta(\cdot \mid x_{<t}, c_t)\big).$$

Taking expectation over $c_t \sim \pi_\theta(\cdot \mid x_{<t})$ and using linearity of expectation gives

$$\mathbb{E}_{c_t}\mathbb{E}_{x_t}\big[r(c_t)\big] = \mathrm{CE}\big(p^*, \bar{p}_\phi(\cdot \mid x_{<t})\big) - \mathbb{E}_{c_t}\,\mathrm{CE}\big(p^*, p_\theta(\cdot \mid x_{<t}, c_t)\big).$$

(iii) Sum over positions. Average the identity in (ii) over $t = 1, \ldots, T$ and over $\boldsymbol{x} \sim \mathcal{D}$:

$$\mathbb{E}_{\boldsymbol{x}}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{c_t}\mathbb{E}_{x_t}\big[r(c_t)\big]\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\frac{1}{T}\sum_{t=1}^{T}\mathrm{CE}\big(p^*, \bar{p}_\phi(\cdot \mid x_{<t})\big)\right] - \mathbb{E}_{\boldsymbol{x}}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{c_t}\,\mathrm{CE}\big(p^*, p_\theta(\cdot \mid x_{<t}, c_t)\big)\right].$$

By the definition of cross-entropy in equation 2 and the chain rule for likelihoods,

$$\mathbb{E}_{x_t\sim p^*(\cdot\mid x_{<t})}\big[-\log \bar{p}_\phi(x_t \mid x_{<t})\big] = \mathrm{CE}\big(p^*, \bar{p}_\phi(\cdot \mid x_{<t})\big),$$

and similarly for the reasoned scorer inside the c_t-expectation. Therefore the two sums on the right are exactly $\mathrm{CE}_{\mathrm{seq}}\big(p^*, \bar{p}_\phi\big)$ and $\mathrm{CE}_{\mathrm{seq}}\big(p^*, p_\theta[\pi_\theta]\big)$ as defined above, yielding the claimed equality.

□

9. Experimental Setup

RLP:

We employ RLP on both base and intermediate checkpoints using diverse datasets. To facilitate this, we use Hugging Face (2025) as the RL training backbone and deploy training using 32 H100 80GB SXM5 GPUs for 170M to 10B tokens. We train the base models with key settings including a constant learning rate of $1e^{-6}$, a batch size of 512 and a maximum context length of 2048 tokens. Each generation step contains 512 unique prompts sampled from the dataset, and performing 16 rollouts with temperature 0.7. We set KL coefficient to 0 across all runs.

Continuous Pre-training:

We continuously pretrain the $\mathcal{M}_{\text{base}}$ model using both general pretraining and specialized post-training corpus to draw comparison between pretraining and RLP training objective. For this experimentation, we use Megatron-LM (Shoeybi et al., 2019) as the pretraining backbone and continuously train on 32 H100 80GB SXM5 GPUs for 170M to 10B tokens depending on the data size and comparison requirement. During training, we use the AdamW optimizer (Loshchilov & Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.95$ and weight decay of 0.1. We use a 2-way tensor and pipeline parallelism to train the model. We set the maximum value of learning rate to $1e^{-6}$, minimum to $1e^{-7}$, and use a batch size of 6M tokens with a 8192 context length.

Post-Training:

For supervised fine-tuning (SFT), we use the OpenThoughts3 dataset (Guha et al., 2025). We filtered examples that did not include a final answer. With this filtering scheme, the total number of samples for SFT post-training is $45,6024$. For RLVR, we used the The Mathematics Aptitude Test of Heuristics (MATH) dataset (Hendrycks et al., 2021b) with $7,500$ examples. This dataset includes problems from various subjects such as algebra, geometry, number theory and precalculus. We trained models in all RLVR experiments for 1 epoch with a global batch size of 1024 and used cosine annealing and an initial learning rate of $1e^{-6}$.

10. Extended ablation details

Table S.1 reports per-task accuracies for each setting, and Fig. 2 provides the corresponding curves for (a) rollout count, (b) completion length, and (c) KL coefficient. Unless stated, each sweep holds the other two dimensions at the best configuration (16 rollouts, completion length 2048, $\beta = 0$).

Rollout count. Increasing G improves accuracy up to $G = 16$, where *Overall* reaches 42.17% (from 34.03%, $+8.14$ points). The largest taskwise lifts at $G = 16$ relative to the base are GSM8K ($+22.96$), MATH-500 ($+13.85$), MIVA ($+7.20$), MMLU ($+6.35$), and MMLU-PRO ($+6.20$), while GPQA is unchanged (27.51 vs 27.52). Moving from $G = 16$ to $G = 32$ slightly lowers *Overall* to 41.75 (-0.42), driven mainly by GPQA (-2.13), with other tasks nearly flat (e.g., MMLU-PRO $+0.79$, MMLU -0.24). This suggests diminishing returns once the group-relative estimator is already well-sampled.

Completion length. Capacity on the thought channel dominates performance. Very short completions underperform sharply: at length 64, *Overall* is 11.50 and *Math* averages 1.12. Increasing to 512 raises *Overall* to 24.65 and *Math* to 22.63. The main jump occurs between 512 and 1024 (*Overall* $+14.24$ to 38.89; GSM8K $+28.55$; MATH-500 $+36.85$). Extending to 2048 adds a smaller but consistent gain (*Overall* 42.17, $+3.28$ over 1024; *Math/Science* $48.06/36.29$). Pushing to 4096 gives only a marginal change (*Overall* 42.21, $+0.04$; small taskwise shifts such as MMLU-PRO $+0.64$ and GSM8K -0.62), so 2048 is the preferred trade-off.

KL coefficient. Adding a token-level KL toward a fixed reference does not help overall. At $\beta = 10^{-4}$ and 10^{-3}, *Overall* is 41.35 and 41.44 (-0.82 and -0.73 vs $\beta = 0$). There are isolated improvements (MMLU-PRO $+1.43$ at 10^{-4}; AMC23 $+1.88$ at 10^{-3}), but these are offset by broader declines (e.g., GSM8K -1.26 and -2.82; GPQA -2.01 and -1.51). The KL term also increases memory use and step time. We therefore keep $\beta = 0$ in the main recipe.

In summary, the appendix table provides the taskwise breakdown behind these trends, and the figure shows the smooth saturation with rollouts, the strong length-driven regime change between 512 and 1024 tokens,

Table S.1: Ablations on rollout count, completion length, and KL weight β with QWEN3-1.7B-BASE. All numbers denote accuracy (%).

Model / Variant	Tasks (%)							Macro avg (%)		
	MATH500	GSM8K	AMC23	Minerva	MMLU	MMLU-Pro	GPQA	Math	Science	Overall
Baseline										
Qwen3-1.7B-Base	48.45	54.16	25.94	15.30	44.85	23.95	27.52	35.96	32.11	34.03
Ablation: # rollouts										
num_rollouts=4	59.45	74.79	33.44	21.78	50.83	28.81	26.52	47.37	35.39	41.38
num_rollouts=8	61.70	76.93	30.62	22.06	50.88	30.55	26.77	47.83	36.07	41.95
num_rollouts=16†	62.30	77.12	30.31	22.50	51.20	30.15	27.51	48.06	36.29	**42.17**
num_rollouts=32	60.45	77.26	30.94	22.29	50.96	30.94	25.38	47.74	35.76	41.75
Ablation: completion length										
completion_length=64	1.00	2.84	0.62	0.00	33.26	15.46	16.92	1.12	21.88	11.50
completion_length=128	1.73	3.17	0.94	0.05	29.04	13.94	12.37	1.47	18.45	9.96
completion_length=256	2.95	13.86	2.81	0.46	37.19	17.09	15.15	5.02	23.14	14.08
completion_length=512	21.35	46.58	16.25	6.34	42.27	19.82	17.93	22.63	26.67	24.65
completion_length=1024	58.20	75.13	28.80	20.47	48.36	27.74	20.31	45.65	32.14	38.89
completion_length=2048†	62.30	77.12	30.31	22.50	51.20	30.15	27.51	48.06	36.29	42.17
completion_length=4096	62.00	76.50	30.60	22.80	51.30	30.79	27.27	47.98	36.45	**42.21**
Ablation: KL weight β										
$\beta = 10^{-4}$	61.35	75.86	28.00	21.50	51.00	31.58	25.50	46.68	36.03	41.35
$\beta = 10^{-3}$	60.90	74.30	32.19	20.73	50.73	30.80	26.00	47.03	35.84	41.44
$\beta = 0^{\dagger}$	62.30	77.12	30.31	22.50	51.20	30.15	27.51	48.06	36.29	**42.17**

and the lack of net benefit from KL.

11. Additional Ablations

Does the improvement sustain if we make Pretraining compute equivalent to RLP?

For both comparisons, the configuration for RLP remains fixed, based on a budget of $T_{inp} = 170M$ input tokens. First, we establish a baseline by continuing the pretraining of the base model on an identical 170M tokens (Base + CPT, Input Token). Second, to create a FLOP-equivalent baseline, we first approximate the total computational cost of RLP. The effective token budget, T_{flop}, can be estimated by summing the tokens used for gradient updates (T_{inp}) and the tokens processed during the rollout phase:

$$T_{flop} = (n \times l_{seq} \times bs \times iters) + T_{inp}$$

where n is the number of rollouts per instance, l_{seq} is the sequence length, bs is the batch size and $iters$ is the number of steps RLP has gone through. This calculation results in an effective budget of approximately 6B tokens for our model. We therefore train a second, more powerful CPT baseline on 6B tokens (Base + CPT, Flop Usage), holding all other hyperparameters constant.

RLP resonates well in presence of multidomain data.

Model	Dataset	Math Avg@1[8]	Science Avg	Science Avg@1[4]	Average
$\mathcal{M}_{\text{base}}$	-	35.96	34.50	32.11	34.19
$\mathcal{M}_{\text{RLP}}$	Only Math	48.23	41.64	36.77	42.21
	Only Science	49.17	39.65	38.26	42.36
	Combined	49.76	42.54	37.78	43.36

Table S.2: **Ablation on math, science, and combined domains.** RLP shows particularly strong generalization in presence of multi-domain data.

Recent works have shown tremendous improvement in reasoning tasks, particularly in mathematics, through RLVR (Liu et al., 2025a; Luo et al., 2025; Hu et al., 2025). However, these methods are often tied to the complexity of queries, limiting their scalability. To draw a parallel, we evaluate RLP on Nemotron-Crossthink using different blends of math and science data. As shown in Table S.2, training only on math yields substantial

math improvements, but comes at the cost of weaker generalization to science. Conversely, training only on science improves science accuracy, but underperforms in math compared to math-only training. Strikingly, combining both domains provides the best overall average, indicating that RLP is able to leverage complementary signals from multiple domains without diluting the benefits within each. This suggests that RLP not only scales beyond single-domain specialization but also thrives in multidomain settings where diverse reasoning styles reinforce one another.