

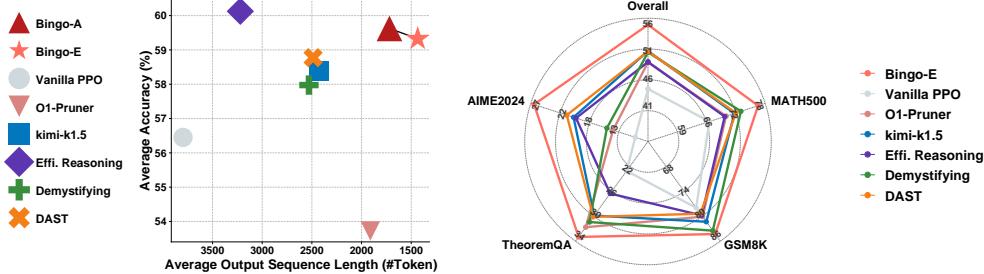
000 BINGO: BOOSTING EFFICIENT REASONING OF LLMs 001 VIA DYNAMIC AND SIGNIFICANCE-BASED 002 REINFORCEMENT LEARNING

006 **Anonymous authors**

007 Paper under double-blind review

010 ABSTRACT

013 Large language models have demonstrated impressive reasoning capabilities, yet
014 they often suffer from inefficiencies due to unnecessarily verbose or redundant
015 outputs. While many works have explored reinforcement learning (RL) to enhance
016 reasoning abilities, most primarily focus on improving accuracy, with limited
017 attention to reasoning efficiency. Some existing approaches introduce direct length-
018 based rewards to encourage brevity, but this often leads to noticeable drops in
019 accuracy. In this paper, we propose BINGO, an RL framework that advances
020 length-based reward design to boost efficient reasoning. BINGO incorporates two
021 key mechanisms: a significance-aware length reward, which gradually guides the
022 model to reduce only insignificant tokens, and a dynamic length reward, which
023 initially encourages elaborate reasoning for hard questions but decays over time
024 to improve overall efficiency. Experiments across multiple reasoning benchmarks
025 show that BINGO improves both accuracy and efficiency. It outperforms the vanilla
026 reward and several other length-based reward baselines in RL, achieving a favorable
027 trade-off between accuracy and efficiency. These results underscore the potential
028 of training LLMs explicitly for efficient reasoning. Our code can be found at
029 <https://anonymous.4open.science/r/Bingo-luck-1124>.



039 **Figure 1: Performance overview of BINGO and other baselines.** **Left:** Scatter plot of average
040 accuracy versus average response length on four benchmarks (MATH500, GSM8K, TheoremQA,
041 AIME2024) using DeepSeek-R1-Distill-Qwen-1.5B as the base model. Points nearer the top-right
042 corner represent a better balance of accuracy and efficiency. **Right:** Radar chart of length-normalized
043 accuracy for each method. Greater radial distances denote higher efficiency.

045 1 INTRODUCTION

047 Large language models (LLMs) (OpenAI, 2024; Gunasekar et al., 2023) have demonstrated impressive
048 reasoning capabilities across a variety of tasks, from arithmetic problem solving (Uesato et al., 2022;
049 Hendrycks et al., 2021; Veeraboina, 2023) to commonsense reasoning (Chen et al., 2023). A key
050 observation from recent work is that sufficiently large models can exhibit emergent reasoning abilities,
051 such as chain-of-thought (CoT) reasoning (Wei et al., 2022b), without explicit supervision (Wei et al.,
052 2022a; Suzgun et al., 2022). Despite these successes, a major challenge persists: LLMs often generate
053 unnecessarily verbose or redundant reasoning traces, leading to inefficiencies in computational cost,
054 redundancy, and latency.

Improving reasoning efficiency of LLMs has thus emerged as an important research direction (Qu et al., 2025; Sui et al., 2025; Li et al., 2025; Wang et al., 2025). Prior work in this area can be broadly categorized into supervised fine-tuning (SFT) approaches (Xia et al., 2025; Xu et al., 2025a; Zhang et al., 2025; Kang et al., 2024) and reinforcement learning (RL) approaches (Luo et al., 2025; Team et al., 2025; Arora & Zanette, 2025; Aggarwal & Welleck, 2025; Yeo et al., 2025; Shi et al., 2025). SFT-based methods focus on constructing compressed reasoning traces and training models to imitate them. While these approaches can be effective, they rely on high-quality compressed supervision, which is costly to obtain and often lacks generalizability across diverse tasks. RL-based methods typically introduce length-based rewards that penalize overly long responses to encourage brevity. However, the design of such rewards or penalties in RL-based methods remains underexplored and is often overly simplistic. For example, O1-Pruner (Luo et al., 2025) applies a uniform penalty to all samples, assuming that every response should be shortened. This assumption often leads to performance degradation, as not all reasoning traces are equally verbose—some require more detailed steps to arrive at the correct answer. To address this, other works have proposed more selective penalty strategies, conditioning penalties on sample correctness (Qu et al., 2025; Team et al., 2025; Yeo et al., 2025) or estimated difficulty (Shi et al., 2025). These approaches typically assign stronger penalties to simpler questions and weaker ones to more challenging cases. However, accurately estimating question difficulty remains a fundamental challenge, and unresolved hard questions often lead to unnecessarily long responses, further undermining reasoning efficiency.

Despite growing interest, current designs of length-based rewards remain limited, as they often fail to adequately promote concise reasoning while preserving answer accuracy. For example, prior work has largely overlooked the impact of token-level contributions on the overall efficiency of reasoning. In this work, we approach the problem from a novel perspective grounded in the concept of *token significance*. Our motivation arises from observed token redundancy in LLMs (Hou et al., 2022; Lin et al., 2025), where many tokens in chain-of-thought (CoT) reasoning contribute little to the final answer. We posit that *not all tokens are equally important for efficient reasoning*—many are insignificant, such as redundant phrases or unnecessary intermediate steps, and can be removed without degrading performance. Existing reward designs often overlook this distinction. In contrast, we introduce a *significance-aware length reward* that selectively penalizes only those insignificant tokens which do not meaningfully contribute to the final answer, while preserving essential reasoning steps.

We also observe that effectively handling hard questions is essential for efficient reasoning. Prior work (Muennighoff et al., 2025; Wu et al., 2025) has shown that encouraging extended CoT reasoning can improve performance by enabling deeper exploration, which may help solve more difficult questions. Therefore, it is intuitive to use length as an incentive for hard questions. However, LLMs should not only solve difficult questions accurately but also do so concisely. Applying a static length incentive can lead to unnecessarily long responses, which may still fail to produce correct answers. To address this, we incorporate a *dynamic length reward* that adapts over the course of training. This reward is applied to significant tokens in incorrect samples to balance exploration and efficiency. Specifically, it encourages longer reasoning in the early training phase to promote exploration, and gradually shifts toward penalizing excessive length in later stages to promote conciseness.

Building on these insights, we introduce **BINGO** (Boosting Efficient ReasonING in Policy Optimization), a RL framework that incorporates our two proposed reward mechanisms into standard RL algorithms such as Proximal Policy Optimization (PPO) (Schulman et al., 2017). This enables joint optimization of both reasoning accuracy and efficiency. Extensive experiments across diverse reasoning benchmarks show that BINGO significantly reduces redundant computation while maintaining—or even improving—task accuracy, consistently outperforming strong baselines. As shown in Figure 1, our method delivers substantial gains on both simple and challenging datasets. For example, on the relatively straightforward GSM8K benchmark, it improves accuracy by 1.6 percentage points while reducing response length by 57%. On the more challenging TheoremQA dataset, it achieves a 4.5-point accuracy improvement and a 60% reduction in response length.

In summary, this paper makes the following key contributions:

- **Token Significance Insight.** We introduce the concept of *token significance* in policy optimization, distinguishing between *significant* and *insignificant* tokens in reasoning traces. This insight motivates our *significance-aware length reward*, which explicitly penalizes uninformative tokens while preserving critical reasoning content, enabling more targeted and effective length control.

- **Dynamic Length Control.** We propose a *dynamic length reward* strategy that adjusts the reward signal over the course of training—encouraging longer reasoning in the early stages to foster exploration, and gradually promoting conciseness as the model converges.
- **Efficiency-Oriented RL Framework.** We develop **BINGO**, a new reinforcement learning framework that integrates both reward strategies. Extensive experiments across multiple reasoning benchmarks, along with comprehensive analyses, demonstrate its effectiveness.

2 RELATED WORK

Reinforcement Learning for Large Language Models. Reinforcement Learning (RL) (Kaelbling et al., 1996) has emerged as a powerful paradigm for aligning large language models (LLMs) with human preferences. In Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022), the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) is employed alongside human preference data to train a reward model that steers the fine-tuning of LLMs. Building on PPO, subsequent works have proposed improved variants to address its limitations. For instance, GRPO (Shao et al., 2024) improves the stability of reward modeling, while REINFORCE++ (Hu, 2025) focuses on enhancing training efficiency. Beyond alignment, RL has also shown promise in improving the reasoning capabilities of LLMs. Early studies (Lightman et al., 2023; Uesato et al., 2022) demonstrated that reward-guided training can enhance multi-step reasoning performance. More recently, DeepSeek-R1 (DeepSeek-AI et al., 2025) demonstrated that large-scale RL can substantially boost reasoning ability across a wide range of tasks, pointing to a promising direction for future work. RL has also been effectively applied in domain-specific scenarios. For example, DeepRetrieval (Jiang et al., 2025) trains models to reason over search engine interactions for improved information retrieval. Fortune (Cao et al., 2025) applies RL to enhance symbolic table reasoning abilities in LLMs through formula.

Efficient Reasoning with Large Language Models. Recent advances have empowered language models to perform strong reasoning via inference-time techniques such as chain-of-thought prompting (Wei et al., 2023; Yao et al., 2023; Cao, 2024; Wang et al., 2023) and post-training (Lightman et al., 2023; Uesato et al., 2022; DeepSeek-AI et al., 2025). More recent work has shifted to optimizing both accuracy and efficiency. Some approaches improve efficiency at inference time, such as token-budget-aware reasoning (Han et al., 2025), or prompting strategies like “reason-without-thinking” (Ma et al., 2025) and chain-of-draft (Xu et al., 2025b). Others apply post-training optimization via supervised fine-tuning (SFT), including TokenSkip (Xia et al., 2025), TwT (Xu et al., 2025a), LightThinker (Zhang et al., 2025), and C3oT (Kang et al., 2024). These SFT methods primarily construct high-quality compressed reasoning paths containing key information, and train the models on them. In parallel, RL-based approaches often improve efficiency by incorporating length controls or penalties into their reward functions. For instance, O1-Pruner (Luo et al., 2025) uses offline length rewards comparing samples against mean lengths. Kimi k1.5 (Team et al., 2025) applies online penalties to correct samples only. Efficient Reasoning (Arora & Zanette, 2025) scales rewards inversely with output length. L1 (Aggarwal & Welleck, 2025) optimizes accuracy under user-defined length constraints. Demystifying (Yeo et al., 2025) uses cosine-based penalties—reducing length for correct outputs while encouraging extended reasoning for incorrect ones. DAST (Shi et al., 2025) employs Token Length Budgets to dynamically adjust reasoning length based on problem difficulty. Building on prior RL-based approaches, we advance length-based reward design to enable LLMs to balance reasoning accuracy with computational efficiency.

3 METHODOLOGY

In this section, we introduce the design of the significance-aware length reward and the dynamic length reward, and explain how these two reward mechanisms are integrated into the BINGO framework, as illustrated in Figure 2. All notations are list at Appendix S.

3.1 TASK FORMULATION

Chain of Thought Reasoning. Let x denote a prompt, and let $y = (y_1, y_2, \dots, y_n)$ represent the sequence generated by a language model parameterized by θ , where y_i is the i -th token in the sequence, and n is the total length of the sequence. Tokens are generated autoregressively from the

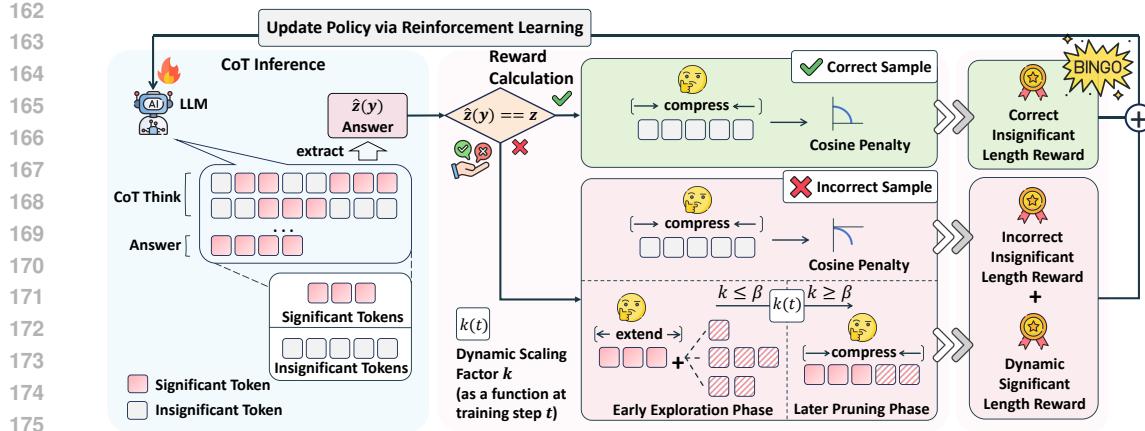


Figure 2: **Illustration of the BINGO framework.** Given a generated CoT trace, the LLM first distinguishes between *significant* and *insignificant* tokens. A dynamic length reward is then computed based on token type and sample correctness. During the early exploration phase of training ($k(t) \geq \beta$), the reward encourages extended reasoning for significant tokens in incorrect samples while penalizing insignificant tokens in all cases. As training progresses ($k(t) < \beta$), the reward shifts toward promoting conciseness by discouraging both significant and insignificant length where appropriate. This two-stage strategy allows the model to first explore broadly and then compress effectively. The aggregated rewards are then used to update the policy via RL, resulting in more accurate and efficient reasoning.

conditional distribution:

$$\pi_\theta(y | x) = \prod_{t=1}^n \pi_\theta(y_i | x, y_{1:i-1}), \quad (1)$$

where the product runs over all tokens in the sequence, with each token y_i (i.e., the action a_i) conditioned on the prompt x and the previous tokens $y_{1:i-1}$ (i.e., the state s_i). Generation continues until an end-of-sequence (EOS) token is produced, signaling the completion of the response. During this process, the model may produce intermediate reasoning tokens, referred to as a *chain of thought* (CoT) (Wei et al., 2022b), before generating the final answer. Therefore, the full output sequence, denoted as y , consists of both the chain of thought and the final answer.

Optimization Objective of Efficient Reasoning. The performance of the model in efficient reasoning is assessed along two key dimensions: *accuracy* and *efficiency*.

Accuracy is measured by the *Exact Match* (EM) metric, which evaluates whether the model’s final answer matches the ground truth. Let $\hat{z}(y)$ denote the final answer extracted from the model-generated sequence y , typically corresponding to its final segment. Let z be the ground-truth answer. Then EM is defined as:

$$\text{EM} = \mathbb{E}_{x \sim \phi} \mathbb{E}_{y \sim \pi_\theta(\cdot | x)} \mathbb{1} [\hat{z}(y) = z], \quad (2)$$

where ϕ denotes the distribution over prompts. The indicator function returns 1 if the predicted answer exactly matches the ground truth, and 0 otherwise.

Efficiency is measured by the *response length* L , typically defined as the number of tokens n in the generated sequence $y = (y_1, y_2, \dots, y_n)$. While longer sequences may offer detailed reasoning, they often result in higher computational cost. Thus, reducing unnecessary tokens without harming accuracy is crucial for practical deployment. An ideal model achieves high EM while minimizing the average response length L , striking a balance between correctness and conciseness.

3.2 SIGNIFICANCE-AWARE LENGTH REWARD

To enhance the efficiency of CoT generation, it is crucial to recognize that not all tokens in a CoT sequence contribute equally to deriving the final answer. **Significant tokens** (such as key concepts, essential terms, or mathematical equations) directly influence the final answer, whereas **insignificant tokens** (including filler words or semantic connectors) contribute little to correctness. Distinguishing

216 between these two token types is essential for improving CoT efficiency by directing computational
217 resources toward the most informative content.

218 We leverage LLM Lingua-2 (Pan et al., 2024) as an off-the-shelf tool for estimating token-level
219 information content. LLM Lingua-2 is an encoder language model, denoted as \mathcal{M}_e , specifically trained
220 to assess the significance of individual tokens for text compression. A more detailed discussion of
221 our token significance measurement approach is provided in Appendix E. We define the significance
222 score for each token as follows:

$$224 \quad S(y_i) = P(y_i \mid \mathbf{y}_{\leq n}; \theta_{\mathcal{M}_e}), \quad (3)$$

225 where n is the total number of tokens in the output sequence.

226 Tokens with low importance scores are considered insignificant, while those with high scores are
227 deemed significant. Specifically, we classify tokens as follows:

$$229 \quad \text{Token } y_i \text{ is } \begin{cases} \text{insignificant,} & \text{if } S(y_i) < \tau, \\ \text{significant,} & \text{if } S(y_i) \geq \tau. \end{cases} \quad (4)$$

231 We then compute the total number of significant tokens L^s and insignificant tokens L^{is} in the response
232 as:

$$234 \quad L^s = \sum_{i=1}^n \mathbb{1}(S(y_i) \geq \tau), \quad L^{is} = \sum_{i=1}^n \mathbb{1}(S(y_i) < \tau), \quad (5)$$

236 where $\mathbb{1}(\cdot)$ is the indicator function, and τ is a pre-defined threshold. To encourage brevity while
237 maintaining reasoning quality, we introduce a significance-aware length reward that penalizes the
238 excessive use of insignificant tokens through a cosine-based decay:

$$239 \quad r_{is}(y) = \cos \left(\text{clip} \left(\frac{L^{is}}{L_{\text{ref}}}, 0, \frac{\pi}{2} \right) \right) + \mathbb{1}[\hat{z}(y) = z] \quad (6)$$

242 where L_{ref}^{is} denotes the number of insignificant tokens in a reference response. The cosine function
243 ensures a smooth, non-linear penalty that gradually decreases the reward as L^{is} increases, while the
244 clipping operation bounds the angle to the interval $[0, \frac{\pi}{2}]$, preventing negative rewards. The final
245 reward combines this length-based penalty with an answer reward derived from the EM indicator,
246 ensuring that answer correctness is preserved.

247 This reward formulation ensures that shorter or equally concise responses—measured in terms of
248 insignificant content—receive higher rewards, while excessively verbose outputs are gently penalized.
249 Notably, our approach preserves natural fluency and coherence in generated text by constraining only
250 the aggregate length of insignificant tokens, without dictating specific token selections or sequences
251 in RL-based training. Compared to standard length-based penalties, our significance-aware approach
252 achieves equal or greater length reductions with less accuracy degradation by selectively penalizing
253 insignificant tokens, as theoretically justified in Appendix F.

254 3.3 DYNAMIC LENGTH REWARD FOR SIGNIFICANT TOKENS

256 While insignificant tokens are consistently penalized to reduce redundancy, significant tokens warrant
257 a more nuanced approach. In the early stages of training, allowing longer reasoning with significant
258 content can facilitate exploration and support the development of robust problem-solving strategies.
259 However, as training progresses, conciseness becomes increasingly important for improving
260 efficiency.

261 To accommodate this shift, we introduce a **dynamic length reward** for significant tokens that evolves
262 over time based on the model’s learning trajectory. This adaptive mechanism is guided by a dynamic
263 scaling factor that captures trends in accuracy and modulates the reward accordingly. Formally, the
264 length-based reward for significant tokens is defined as:

$$266 \quad r_s(y) = \begin{cases} k \cdot \frac{L^s}{L_{\text{ref}}^s}, & \text{if } k \geq \beta \\ -\alpha \cdot t \cdot \frac{L^s}{L_{\text{ref}}^s}, & \text{if } k < \beta \end{cases} \quad (7)$$

268 where L^s represents the number of significant tokens in the generated output, L_{ref}^s is the corresponding
269 value from the reference model, and k is a dynamic scaling factor that reflects the reasoning trend

270 during training. The training step t begins at 1 and increments gradually when k first falls below the
 271 threshold β , which determines when the model transitions from incentivizing longer significant token
 272 lengths to penalizing them. α is a weight that determines the rate of decay in this process. The value
 273 of k is estimated by fitting a linear model to recent training steps:
 274

$$275 \quad k = \frac{\sum_{t=S_a}^{S_b} (t - \bar{t})(acc_t - \bar{acc})}{\sum_{t=S_a}^{S_b} (t - \bar{t})^2} \quad (8)$$

$$276$$

$$277$$

278 where acc_t denotes the training batch accuracy at training step t , \bar{t} and \bar{acc} are the mean step index
 279 and mean accuracy over the interval $[S_a, S_b]$. A positive k indicates an upward accuracy trend,
 280 suggesting that the model is still in an improvement phase. As training progresses and accuracy
 281 plateaus, k approaches zero or becomes negative. The theoretical rationale behind the design of
 282 our dynamic length reward schedule is discussed in detail in Appendix G. This dynamic adaptation
 283 allows the model to balance early-stage exploration with late-stage compression, fostering reasoning
 284 strategies that are both effective and efficient.
 285

286 3.4 BOOSTING EFFICIENT REASONING IN POLICY OPTIMIZATION

287 We propose a novel reinforcement learning algorithm, BINGO (Boosting Efficient ReasonING in
 288 Policy Optimization), designed to jointly optimize reasoning performance and efficiency. BINGO
 289 extends the reinforcement learning framework—primarily based on Proximal Policy Optimization
 290 (PPO) in this work—by introducing two key innovations: a **significance-aware length reward** and a
 291 **dynamic length reward**.

292 As discussed in Section 3.2, we begin by categorizing tokens into *significant* and *insignificant* based
 293 on their significance scores. To promote concise yet informative responses, we introduce a cosine-
 294 based reward function that adjusts penalties according to the length composition of the response. For
 295 correctly answered samples, the reward penalizes only the length of the insignificant portion, reducing
 296 verbosity while preserving essential reasoning. For incorrect samples, the reward both penalizes the
 297 use of insignificant tokens and encourages the generation of more significant reasoning content.
 298

299 To balance exploration and efficiency over the course of training, we incorporate a time-dependent
 300 mechanism that gradually reduces the incentive for longer responses. As detailed in Section 3.3, this
 301 dynamic reward decays as the model converges, shifting the focus from exploration to conciseness.
 302

303 The overall reward formulation integrates these components into a unified objective:

$$304 \quad R^{\text{BINGO}}(y) = \begin{cases} \underbrace{\lambda_c \cdot r_{is}(y)}_{\text{Correct insignificant length reward}} & , \quad \text{if correct,} \\ \underbrace{\lambda_w^{is} \cdot [r_{is}(y) - 1]}_{\text{Incorrect insignificant length reward}} + \underbrace{\min(0, r_s(y) - \lambda_w^s)}_{\text{Dynamic significant length reward}} & , \quad \text{if incorrect.} \end{cases} \quad (9)$$

$$305$$

$$306$$

$$307$$

$$308$$

309 where the coefficient λ_c controls the strength of the penalty applied to correct responses, while λ_w^{is}
 310 determines the magnitude of the penalty for incorrect ones. The parameter λ_w^s serves as a dynamic
 311 threshold to balance exploration when the model generates incorrect outputs.
 312

313 We optimize the policy using the proximal policy optimization objective with the reward R^{BINGO}
 314 defined by Equation 9. The surrogate objective is:
 315

$$316 \quad \mathcal{J}_{\text{BINGO}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip} (r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right], \quad (10)$$

$$317$$

318 where:
 319

- 320 • $r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)}$ is the importance sampling ratio,
 321
- 322 • \hat{A}_t is the advantage estimate at time step t , computed via generalized advantage estimation
 323 using the final sequence-level reward R^{BINGO} and the value predictions $V(s_t)$.
 324
- 325 • ϵ is a clipping parameter.
 326

324
 325 **Table 1: Comparison of different length-based rewards on reasoning benchmarks.** Each method
 326 is evaluated using DeepSeek-R1-Distill-Qwen-1.5B as the base model by answer accuracy (Acc,
 327 %), response length (Len), and length-normalized accuracy (L-Acc, %). The best performance is
 328 highlighted in **dark blue**, and the second-best in **light blue**.

Length-based Reward	MATH500			GSM8K			TheoremQA			AIME2024		
	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑
Vanilla PPO (Schulman et al., 2017)	81.4	2,771	66.2	85.4	1,310	78.2	32.3	4,146	22.7	26.7	6,961	10.3
O1-Pruner (Luo et al., 2025)	74.4	991	69.8	81.4	211	80.3	32.4	485	31.4	26.7	5,958	13.9
kimi-k1.5 (Team et al., 2025)	80.4	1,692	71.6	85.4	739	81.5	34.4	2,136	29.6	33.3	5,159	20.3
Effi. Reasoning (Arora & Zanette, 2025)	82.6	2,395	69.5	86.4	1,155	80.0	34.8	3,560	26.2	36.7	5,771	19.9
Demystifying (Yeo et al., 2025)	80.2	1,411	73.0	86.6	548	83.6	35.1	1,976	30.6	30.0	6,183	14.9
DAST (Shi et al., 2025)	81.2	1,770	71.9	82.0	456	79.6	35.2	2,325	29.8	36.7	5,400	21.4
<i>Bingo (Ours)</i>												
Bingo-A	82.2	894	77.6	87.0	563	83.9	36.8	1,648	32.9	33.3	2,943	26.7
Bingo-E	80.6	779	76.7	86.7	345	84.9	36.7	1,584	33.0	33.3	2,943	26.7

337
 338 Therefore, BINGO achieves a favorable trade-off between accuracy and efficiency by maximizing the
 339 objective function $\mathcal{J}_{\text{BINGO}}(\theta)$.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP AND EVALUATION METRICS

345 We fine-tune two reasoning models, *DeepSeek-R1-Distill-Qwen-1.5B* and *DeepSeek-R1-Distill-Qwen-7B*, along with an instruction-tuned model, *Qwen2.5-Math-7B-Instruct*, on the MATH (Hendrycks
 346 et al., 2021) training split. Evaluation is conducted on four test sets: MATH500 (Hendrycks et al.,
 347 2021), GSM8K (Cobbe et al., 2021), AIME2024 (Veeraboina, 2023), and THEOREMQA (Chen
 348 et al., 2023). Among them, MATH500 serves as the in-distribution benchmark, while the others are
 349 used as out-of-distribution test sets.

350 We compare our models with several baselines: the frozen *Base* model (zero-shot), a model fine-
 351 tuned with the *Vanilla PPO* algorithm, and five state-of-the-art methods—*DAST* (Shi et al., 2025),
 352 *Demystifying* (Yeo et al., 2025), *Efficient Reasoning* (Arora & Zanette, 2025), *Kimi-k1.5* (Team
 353 et al., 2025), and *O1-Pruner* (Luo et al., 2025). To isolate and evaluate the effectiveness of different
 354 length-based reward designs, we re-implement the core *length-reward components* from all methods
 355 within a unified PPO framework. This controlled setup allows direct assessment of whether the
 356 reward designs themselves drive performance. For a more detailed description of the experimental
 357 settings, please refer to Appendix I. A thorough hyperparameter study can be found in Appendix R.
 358

359 We report two variants of our model: BINGO-A, the *accuracy-preferred* checkpoint, selected when
 360 validation accuracy reaches its peak; and BINGO-E, the *efficiency-preferred* checkpoint, chosen when
 361 response length stabilizes during continued training. This dual-reporting strategy enables practitioners
 362 to choose a model variant based on their preference for accuracy or efficiency. They may correspond
 363 to the same checkpoint.

364 To evaluate the reasoning efficiency of LLMs in this study, we report not only accuracy (Acc) and
 365 response length (Len), but also introduce an additional metric: **length-normalized accuracy** (L-Acc).
 366 This metric provides a more comprehensive measure of a model’s reasoning efficiency by jointly
 367 considering correctness and conciseness. It is defined as:

$$368 \quad \text{L-Acc} = \text{Acc} \times \sqrt{1 - \frac{L}{L_{\max}}}, \quad (11)$$

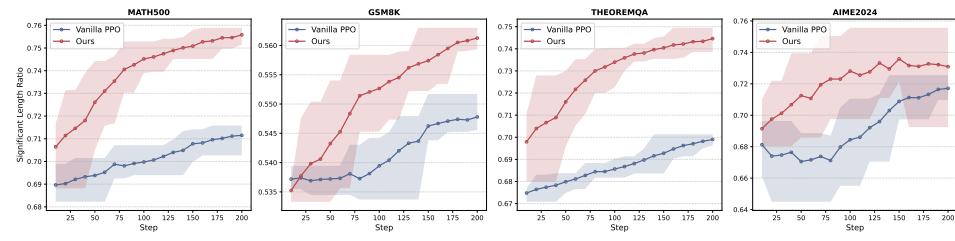
369 where L is the average response length and L_{\max} is the maximum allowable length. A detailed
 370 definition and theoretical analysis of L-Acc are provided in Appendix H.

4.2 PERFORMANCE COMPARISON WITH BASELINE METHODS

371 **BINGO outperforms existing methods in L-Acc:** As shown in Table 1, both BINGO-A and BINGO-
 372 E achieve the highest L-Acc across all four benchmarks, outperforming previous baselines such as
 373 Vanilla PPO, Efficient Reasoning, and DAST.

378
 379 **Table 2: Performance comparison across model scales and types.** Accuracy (Acc), average
 380 output length (Length), and length-normalized accuracy (L-Acc, %) on four benchmarks. The best
 381 performance is highlighted in **dark blue**, and the second-best in **light blue**.

382 Method	383 MATH500			384 GSM8K			385 TheoremQA			386 AIME2024		
	387 Acc↑	388 Len↓	389 L-Acc↑	390 Acc↑	391 Len↓	392 L-Acc↑	393 Acc↑	394 Len↓	395 L-Acc↑	396 Acc↑	397 Len↓	398 L-Acc↑
<i>DeepSeek-R1-Distill-Qwen-1.5B</i>												
384 Base	63.2	3,913	45.7	73.2	2,025	63.5	18.7	5,741	10.3	16.7	7,027	6.3
385 PPO	81.4	2,771	66.2	85.4	1,310	78.2	32.3	4,146	22.7	26.7	6,961	10.3
386 Bingo-A (Ours)	82.2	894	77.6	87.0	563	83.9	36.8	1,648	32.9	33.3	2,943	26.7
387 Bingo-E (Ours)	80.6	779	76.7	86.7	345	84.9	36.7	1,584	33.0	33.3	2,943	26.7
<i>DeepSeek-R1-Distill-Qwen-7B</i>												
388 Base	82.8	3,033	65.7	85.7	1,001	80.3	37.8	4,340	25.9	40.0	6,528	18.0
389 PPO	88.4	1,536	79.7	92.9	918	87.5	45.4	2,709	37.1	56.7	5,857	30.3
390 Bingo-A (Ours)	88.8	1,400	80.9	92.3	371	90.2	45.2	1,908	39.6	63.3	4,670	41.5
391 Bingo-E (Ours)	87.2	1,252	80.3	91.8	366	89.7	45.0	1,693	40.1	60.0	4,011	42.9
<i>Qwen2.5-Math-7B-Instruct</i>												
392 Base	80.8	727	70.3	95.8	331	90.3	36.8	919	30.7	16.7	1,310	12.5
393 PPO	82.0	670	72.3	96.6	305	91.5	37.6	759	32.5	20.0	1,260	15.2
394 Bingo-A (Ours)	82.6	656	73.0	96.1	283	91.5	37.9	598	34.0	20.0	892	16.8
395 Bingo-E (Ours)	81.6	559	73.6	96.0	241	92.0	37.1	552	33.5	16.7	811	14.2



402 Figure 3: **Significant Length Ratio dynamics during training.** The x-axis indicates training steps,
 403 and the y-axis denotes the proportion of significant tokens in the generated responses. Each subplot
 404 corresponds to one benchmark evaluated using DeepSeek-R1-Distill-Qwen-1.5B as the base model.
 405 The blue curve represents the baseline method (Vanilla PPO), and the red curve represents our
 406 approach (Ours).

408 **BINGO-A improves accuracy while significantly reducing response length:** BINGO-A reduces
 409 average response length by up to 68% compared to Vanilla PPO (e.g., 894 vs. 2,771 tokens on
 410 MATH500), demonstrating the model’s ability to generate concise and correct reasoning steps.

411 **Existing baselines struggle with the trade-off between accuracy and brevity:** Approaches like
 412 Efficient Reasoning produce verbose outputs, while methods such as O1-Pruner overly shorten
 413 responses, compromising accuracy.

415 4.3 PERFORMANCE EVALUATION ACROSS VARYING MODEL SCALES

417 **BINGO achieves the best trade-off between accuracy and response length across different model**
 418 **sizes:** As shown in Table 2, both BINGO-A and BINGO-E consistently outperform all other methods
 419 across various model sizes (1.5B and 7B parameters) and benchmarks, achieving the highest L-Acc
 420 while maintaining competitive or superior accuracy.

421 **BINGO-E offers a substantial reduction in response length without sacrificing accuracy:** BINGO-E
 422 reduces response length by up to 63% (e.g., 366 vs. 1,001 tokens on GSM8K) compared to the
 423 Base model, while also improving accuracy by 6.1 percentage points, demonstrating the model’s
 424 ability to generate concise and accurate reasoning steps.

426 4.4 ANALYSIS OF SIGNIFICANT VERSUS INSIGNIFICANT TOKEN RATIO

428 **Significance-aware reward increases the proportion of significant tokens:** As shown in Figure 3,
 429 our significance-aware reward consistently improves the significant-token ratio across all datasets.
 430 For example, the ratio increases from 0.71 to 0.75 (+4%) on the MATH500. Similarly, the STR
 431 improves by about 2% on GSM8K and 5% on TheoremQA, indicating that our method retains the
 432 essential reasoning steps while removing redundant or insignificant tokens.

432 **Table 3: Ablation study on reward components.** Each method is evaluated using DeepSeek-R1-
 433 Distill-Qwen-1.5B as the base model by answer accuracy (Acc, %), response length (Len), and
 434 length-normalized accuracy (L-Acc, %). Values in parentheses indicate the relative drop in L-Acc
 435 compared to BINGO-A. The best performance is highlighted in **dark blue**, and the second-best in
 436 **light blue**.

Method	MATH500			GSM8K			TheoremQA			AIME2024		
	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑
Bingo-A (Ours)	82.2	894	77.6	87.0	563	83.9	36.8	1,648	32.9	33.3	2,943	26.7
Vanilla PPO	81.4	2,771	66.2 (-14.7)	85.4	1,310	78.2 (-6.8)	32.3	4,146	22.7 (-31.0)	26.7	6,961	10.3 (-16.4)
Significance-Aware Length Reward	81.4	1,734	72.3 (-5.3)	86.7	742	82.6 (-1.3)	36.0	2,841	29.1 (-3.8)	40.0	5,138	24.4 (-2.3)
w/o Cosine	78.6	1,750	69.7 (-7.9)	85.7	509	83.0 (-0.9)	35.3	2,414	29.7 (-3.2)	33.3	6,454	15.4 (-11.3)
w/o Significance Separation	79.8	1,666	71.2 (-6.4)	86.6	604	83.3 (-0.6)	36.9	2,328	31.3 (-1.6)	26.7	5,702	14.7 (-12.0)
w/o Length Incentive	77.8	1,400	70.8 (-6.8)	82.6	425	80.5 (-3.4)	35.7	1,636	32.0 (-0.9)	30.0	4,157	21.1 (-5.6)
Dynamic Length Reward	79.0	2,204	67.5 (-10.1)	84.3	955	79.2 (-4.7)	33.9	2,632	27.9 (-5.0)	30.0	5,047	18.6 (-8.1)

445 **Improved reasoning efficiency with richer content:** The increase in significant tokens leads to
 446 shorter, more focused chains, as demonstrated by the reductions in response length shown in Table 2.
 447 These concise outputs are not only shorter but also contain more meaningful content, resulting in
 448 higher raw Acc and L-Acc, reinforcing the effectiveness of our reward strategy.

450 4.5 ABLATION STUDY

451 **Combining Significance-Aware and Dynamic Length rewards yields the best trade-off:** Table 3
 452 shows that the joint use of both the Significance-Aware and Dynamic Length rewards (BINGO-A)
 453 provides the best performance, achieving the highest accuracy and L-Acc across all four benchmarks,
 454 while maintaining competitive or superior raw accuracy compared to other methods.

455 **Removing key reward components degrades performance significantly:** Ablations show that
 456 removing any of the reward components leads to noticeable performance drops, particularly in terms
 457 of L-Acc. These results confirm the complementary nature of the reward components and their crucial
 458 role in optimizing the model’s efficiency and accuracy.

460 4.6 ADDITIONAL EXPERIMENTS AND ANALYSIS

462 **BINGO improves across multiple RL algorithms:** We evaluate the generalizability of our reward
 463 design by integrating it into other RL algorithms, including RLOO, GRPO, and Reinforce++. As
 464 shown in Appendix K, BINGO variants consistently outperform vanilla ones, achieving superior
 465 performance in both accuracy and L-Acc.

466 **BINGO effectively reduces response length, especially for incorrect samples:** The distribution of
 467 response lengths for correct vs. incorrect samples in Appendix O shows that BINGO significantly
 468 shortens incorrect sample lengths compared to PPO. Furthermore, Figure 8 in Appendix O illustrates
 469 that incorrect samples show a more significant reduction in response length during training, con-
 470 firming the dynamic reward’s effectiveness. Figure 7 in Appendix N further shows that our method
 471 consistently reduces response length more than PPO.

472 **Analysis of significant token ratio and token-level significance visualization:** Appendix L shows
 473 that BINGO increases the proportion of significant tokens compared to baselines, while Appendix M
 474 provides a token-level significance visualization, demonstrating how our approach retains essential
 475 reasoning steps and eliminates redundancy.

476 **Analysis confirms the effectiveness of dynamic and significance rewards:** The analysis in Ap-
 477 pendix P validates that our dynamic and significance rewards balance exploration and efficiency. A
 478 case study in Appendix Q further demonstrates the practical impact of BINGO on reasoning efficiency.

480 5 CONCLUSION

483 In this paper, we introduce BINGO, a RL framework that enhances reasoning efficiency in LLMs.
 484 By incorporating significance-aware and dynamic length rewards, BINGO strikes a strong balance
 485 between exploration and conciseness, outperforming existing methods across multiple benchmarks.

486 REFERENCES
487

488 Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with
489 reinforcement learning, 2025. URL <https://arxiv.org/abs/2503.04697>.

490 Daman Arora and Andrea Zanette. Training language models to reason efficiently, 2025. URL
491 <https://arxiv.org/abs/2502.04463>.

492 Lang Cao. GraphReason: Enhancing reasoning capabilities of large language models through a
493 graph-based verification approach. In Bhavana Dalvi Mishra, Greg Durrett, Peter Jansen, Ben
494 Lipkin, Danilo Neves Ribeiro, Lionel Wong, Xi Ye, and Wenting Zhao (eds.), *Proceedings of
495 the 2nd Workshop on Natural Language Reasoning and Structured Explanations (@ACL 2024)*,
496 pp. 1–12, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL
497 <https://aclanthology.org/2024.nlrse-1.1/>.

498 Lang Cao, Jingxian Xu, Hanbing Liu, Jinyu Wang, Mengyu Zhou, Haoyu Dong, Shi Han, and
499 Dongmei Zhang. Fortune: Formula-driven reinforcement learning for symbolic table reasoning in
500 language models. *arXiv preprint arXiv:2505.23667*, 2025.

502 Wenhui Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and
503 Tony Xia. Theoremqa: A theorem-driven question answering dataset. In *Proceedings of the 2023
504 Conference on Empirical Methods in Natural Language Processing*, pp. 7889–7901, Singapore,
505 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.489. URL
506 <https://aclanthology.org/2023.emnlp-main.489/>.

507 Paul Francis Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei.
508 Deep reinforcement learning from human preferences. *ArXiv*, abs/1706.03741, 2017. URL
509 <https://api.semanticscholar.org/CorpusID:4787508>.

511 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher
512 Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint
513 arXiv:2110.14168*, 2021.

514 Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen
515 Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for
516 reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

517 DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu,
518 Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu,
519 Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao
520 Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan,
521 Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao,
522 Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding,
523 Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang
524 Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong,
525 Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao,
526 Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang,
527 Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang,
528 Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L.
529 Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang,
530 Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng
531 Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng
532 Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan
533 Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang,
534 Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen,
535 Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li,
536 Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang,
537 Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan,
538 Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia
539 He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong
Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha,
Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhehan Xu, Zhenda Xie, Zhengyan Zhang,

540 Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li,
 541 Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen
 542 Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.
 543 URL <https://arxiv.org/abs/2501.12948>.

544 Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth
 545 Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital
 546 Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai,
 547 Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. URL <https://arxiv.org/abs/2306.11644>.

548 Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-
 549 budget-aware llm reasoning, 2025. URL <https://arxiv.org/abs/2412.18547>.

550 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
 551 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv*
 552 preprint *arXiv:2103.03874*, 2021.

553 Le Hou, Richard Yuanzhe Pang, Tianyi Zhou, Yuexin Wu, Xinying Song, Xiaodan Song, and
 554 Denny Zhou. Token dropping for efficient BERT pretraining. In Smaranda Muresan, Preslav
 555 Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association*
 556 *for Computational Linguistics (Volume 1: Long Papers)*, pp. 3774–3784, Dublin, Ireland, May
 557 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.262. URL
 558 <https://aclanthology.org/2022.acl-long.262/>.

559 Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models.
 560 *ArXiv*, abs/2501.03262, 2025. URL <https://api.semanticscholar.org/CorpusID:275342265>.

561 Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing
 562 prompts for accelerated inference of large language models, 2023. URL <https://arxiv.org/abs/2310.05736>.

563 Pengcheng Jiang, Jiacheng Lin, Lang Cao, R. Tian, S. Kang, Z. Wang, Jimeng Sun, and Jiawei
 564 Han. Deepretrieval: Hacking real search engines and retrievers with large language models via
 565 reinforcement learning. *arXiv preprint arXiv: 2503.00223*, 2025.

566 Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A
 567 survey. *J. Artif. Intell. Res.*, 4:237–285, 1996. URL <https://api.semanticscholar.org/CorpusID:1708582>.

568 Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. C3ot: Generating shorter chain-of-thought
 569 without compromising effectiveness, 2024. URL <https://arxiv.org/abs/2412.11664>.

570 Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference
 571 efficiency of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Pro-
 572 ceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6342–
 573 6353, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/
 574 2023.emnlp-main.391. URL <https://aclanthology.org/2023.emnlp-main.391/>.

575 Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. Compressing context to enhance inference
 576 efficiency of large language models, 2023b. URL <https://arxiv.org/abs/2310.06201>.

577 Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian
 578 Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhijiang
 579 Guo, Le Song, and Cheng-Lin Liu. From system 1 to system 2: A survey of reasoning large
 580 language models, 2025. URL <https://arxiv.org/abs/2502.17419>.

581 Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan
 582 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL
 583 <https://arxiv.org/abs/2305.20050>.

594 Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu
 595 Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Rho-1: Not all tokens are what you need, 2025.
 596 URL <https://arxiv.org/abs/2404.07965>.

597
 598 Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao,
 599 and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning, 2025.
 600 URL <https://arxiv.org/abs/2501.12570>.

601 Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning
 602 models can be effective without thinking, 2025. URL <https://arxiv.org/abs/2504.09858>.

603
 604 Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke
 605 Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time
 606 scaling, 2025. URL <https://arxiv.org/abs/2501.19393>.

607
 608 OpenAI. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>.

609
 610 Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong
 611 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton,
 612 Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan
 613 Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback.
 614 *ArXiv*, abs/2203.02155, 2022. URL <https://api.semanticscholar.org/CorpusID:246426909>.

615
 616 Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor
 617 Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. Llmlingua-
 618 2: Data distillation for efficient and faithful task-agnostic prompt compression, 2024. URL
 619 <https://arxiv.org/abs/2403.12968>.

620
 621 Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong
 622 Liu, Shuxian Liang, Junxian He, Peng Li, Wei Wei, Jing Shao, Chaochao Lu, Yue Zhang, Xian-
 623 Sheng Hua, Bowen Zhou, and Yu Cheng. A survey of efficient reasoning for large reasoning
 624 models: Language, multimodality, and beyond, 2025. URL <https://arxiv.org/abs/2503.21614>.

625
 626 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proxi-
 627 mal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017. URL <https://api.semanticscholar.org/CorpusID:28695052>.

628
 629 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
 630 Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of
 631 mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.

632
 633 Shuming Shi, Jian Zhang, Yi Shen, Kai Wang, Shiguo Lian, Ning Wang, Wenjing Zhang, Jieyun
 634 Huang, and Jiangze Yan. Dast: Difficulty-adaptive slow-thinking for large reasoning models, 2025.
 635 URL <https://arxiv.org/abs/2503.04472>.

636
 637 Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan J. Lowe, Chelsea Voss, Alec
 638 Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback.
 639 *ArXiv*, abs/2009.01325, 2020. URL <https://api.semanticscholar.org/CorpusID:221665105>.

640
 641 Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu,
 642 Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient
 643 reasoning for large language models, 2025. URL <https://arxiv.org/abs/2503.16419>.

644
 645 Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung,
 646 Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-
 647 bench tasks and whether chain-of-thought can solve them, 2022. URL <https://arxiv.org/abs/2210.09261>.

648 Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun
 649 Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming
 650 Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han
 651 Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li,
 652 Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su,
 653 Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye,
 654 Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu,
 655 Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong,
 656 Weiran He, Weixiao Huang, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu,
 657 Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang
 658 Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du,
 659 Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu
 660 Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, and Zonghan Yang. Kimi k1.5: Scaling
 661 reinforcement learning with llms, 2025. URL <https://arxiv.org/abs/2501.12599>.
 662

662 Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia
 663 Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and
 664 outcome-based feedback, 2022. URL <https://arxiv.org/abs/2211.14275>.
 665

665 Hemish Veeraboina. Aime problem set 1983-2024, 2023. URL <https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024>.
 666

666 Rui Wang, Hongru Wang, Boyang Xue, Jianhui Pang, Shudong Liu, Yi Chen, Jiahao Qiu, Derek Fai
 667 Wong, Heng Ji, and Kam-Fai Wong. Harnessing the reasoning economy: A survey of efficient
 668 reasoning for large language models, 2025. URL <https://arxiv.org/abs/2503.24377>.
 669

669 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdh-
 670 ery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models,
 671 2023. URL <https://arxiv.org/abs/2203.11171>.
 672

672 Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama,
 673 Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals,
 674 Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022a.
 675 URL <https://arxiv.org/abs/2206.07682>.
 676

676 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny
 677 Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in
 678 neural information processing systems*, 35:24824–24837, 2022b.
 679

679 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le,
 680 and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.
 681 URL <https://arxiv.org/abs/2201.11903>.
 682

682 Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Un-
 683 derstanding chain-of-thought length in llms, 2025. URL <https://arxiv.org/abs/2502.07266>.
 684

684 Heming Xia, Yongqi Li, Chak Tou Leong, Wenjie Wang, and Wenjie Li. Tokenskip: Control-
 685 lable chain-of-thought compression in llms, 2025. URL <https://arxiv.org/abs/2502.12067>.
 686

686 Jingxian Xu, Mengyu Zhou, Weichang Liu, Hanbing Liu, Shi Han, and Dongmei Zhang. Twt:
 687 Thinking without tokens by habitual reasoning distillation with multi-teachers' guidance, 2025a.
 688 URL <https://arxiv.org/abs/2503.24198>.
 689

689 Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing
 690 less, 2025b. URL <https://arxiv.org/abs/2502.18600>.
 691

691 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik
 692 Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.
 693 URL <https://arxiv.org/abs/2305.10601>.
 694

702 Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long
703 chain-of-thought reasoning in llms, 2025. URL <https://arxiv.org/abs/2502.03373>.
704
705 Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun
706 Chen, and Ningyu Zhang. Lighthinker: Thinking step-by-step compression, 2025. URL <https://arxiv.org/abs/2502.15589>.
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755

756 **Contents of Appendix**

759	A Use of LLM Assistance	16
761	B Ethics Statement	16
763	C Reproducibility Statement	16
765	D Limitations and Future Work	16
767	E Discussion of Token Significance Measurement	17
770	F Theoretical Analysis of Significance-Aware Length Reward	17
772	G Theoretical Discussion of Dynamic Length Reward	19
774	H Definition and Theoretical Analysis of Length-normalized Accuracy	20
776	I Detailed Settings of Experiments	21
778	J Performance under Extended Sampling Settings	23
781	K Performance across Different Reinforcement Learning Algorithms	24
783	L Analysis of Significant Token Ratio	25
785	M Token-level Significance Visualization	25
788	N Analysis of Response Lengths Trends during Training	26
790	O Analysis of Response Lengths Dynamics for Correct vs. Wrong Samples	26
792	P Analysis of Incorrect Response Length under Different Reward Designs	28
794	Q Case Study	29
796	R Hyperparameter Study	32
799	S Notation Table	33

800
801
802
803
804
805
806
807
808
809

810
811

A USE OF LLM ASSISTANCE

812
813
814
815
816
817
In preparing this manuscript, we used large language models (LLMs) solely as writing assistants
to help with grammar checking, improving sentence structure and readability, ensuring consistent
technical terminology, condensing verbose passages, and formatting citations according to confer-
ence guidelines. All research contributions—including ideas, experimental design, data analysis,
mathematical derivations, and scientific conclusions—are entirely the authors' original work. The
authors take full responsibility for all content presented in this paper.818
819
820

B ETHICS STATEMENT

821
822
823
824
825
826
827
828
Our research focuses on improving the efficiency of large language model reasoning through rein-
forcement learning techniques, which poses no direct ethical concerns regarding human subjects, as
no human data collection or experimentation was conducted. All datasets used (MATH, GSM8K,
TheoremQA, AIME2024) are publicly available benchmarks with proper citations. We acknowledge
the broader implications of more efficient LLM reasoning, including potential dual-use concerns, but
emphasize that our contributions aim to reduce computational costs and environmental impact of AI
systems. The research was conducted with academic integrity, and all authors have reviewed and
agree with the content presented. There are no conflicts of interest to declare.830
831
832

C REPRODUCIBILITY STATEMENT

833
834
835
836
837
838
839
840
To ensure reproducibility, we provide comprehensive implementation details throughout the paper.
Section 3 describes our complete algorithmic framework, including all hyperparameters and reward
formulations. Section 4.1 details our experimental setup, while Appendix I provides comprehensive
settings including training and evaluation configurations, dataset settings, optimization parameters,
data splits, evaluation metrics, and computational requirements. All experiments use publicly
accessible pre-trained models (DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-Distill-Qwen-7B,
and Qwen2.5-Math-7B-Instruct) and datasets available on HuggingFace. Our code is available
through an anonymous link in the abstract and as a zip file in the supplemental materials, and will be
made publicly available upon acceptance.842
843
844

D LIMITATIONS AND FUTURE WORK

845
846
Limitations. Despite the promising results of the BINGO framework, there are some limitations that
need to be acknowledged:847
848
849
850
851
852
853
• **Task-Specific Performance Variability:** While BINGO performs well across several reasoning
benchmarks, its performance may vary on more domain-specific or highly complex tasks. Tasks
requiring intricate domain knowledge or long-term dependencies may still present challenges.
• **Computational Resources for Training:** The reinforcement learning framework utilized by
BINGO requires considerable computational resources for training, which may limit its scalability
to larger datasets and more complex tasks.854
855
Future Work. Several directions can be explored to improve upon BINGO:856
857
858
859
860
861
• **Expanding to Diverse Domains and Tasks:** To broaden the applicability of BINGO, it would be
beneficial to extend its evaluation to more complex, domain-specific reasoning tasks. This could
involve tasks in specialized fields like legal reasoning or advanced scientific modeling.
• **Handling Long-Term Dependencies:** Exploring ways to better handle tasks that require long-term
memory or reasoning across large spans of text could make the framework more effective for
complex problem-solving scenarios.
• **Improving Training Efficiency:** Future efforts could focus on reducing the computational cost of
training by utilizing techniques like transfer learning or distillation, making the framework more
accessible for large-scale applications.

864

E DISCUSSION OF TOKEN SIGNIFICANCE MEASUREMENT

865
866 As introduced in Section 3.2, we adopt *LLMLingua-2* (Pan et al., 2024) to measure token significance.
867868 A variety of methods have been proposed to mitigate token redundancy in large language models (Hou
869 et al., 2022; Lin et al., 2025), including prompt compression techniques (Li et al., 2023a; Pan et al.,
870 2024; Jiang et al., 2023). One intuitive approach, *Selective Context* (Li et al., 2023b), estimates token
871 importance using a semantic confidence score derived from language modeling:
872

873
$$S(y_i) = -\log P(y_i | y_{<i}; \theta_{M_L}), \quad (12)$$

874 where y_i is the i -th token, and θ_{M_L} denotes the parameters of a unidirectional language model. This
875 score reflects the model’s uncertainty about each token, using randomness as a proxy for importance.
876 However, this method suffers from two main limitations: (1) position bias, where tokens toward the
877 end of a sequence are systematically assigned lower importance, and (2) the architectural constraint
878 of unidirectional models like GPT, which lack access to future context and thus provide a limited
879 view of token-level informativeness.
880881 In contrast, when designing a significance-aware length reward, we leverage *LLMLingua-2* to compute
882 token significance using:
883

884
$$S(y_i) = P(y_i | y_{\leq i}; \theta_{M_e}), \quad (13)$$

885 where θ_{M_e} denotes the parameters of a bidirectional encoder model. Unlike unidirectional architectures,
886 *LLMLingua-2*—built on BERT-like models—leverages both preceding and succeeding context
887 to evaluate each token, thereby alleviating position bias and enabling a more accurate and holistic
888 assessment of informativeness. Therefore, *LLMLingua-2* serves as a more effective off-the-shelf tool
889 for estimating token-level information content. It is a language encoder model specifically trained to
890 assess the significance of individual tokens for the purpose of text compression. Given a generated
891 sequence $y = (y_1, y_2, \dots, y_n)$, each token y_i receives an importance score $S(y_i)$, which is used to
892 distinguish between *significant* and *insignificant* reasoning steps.
893894 The advantage of using *LLMLingua-2* over *Selective Context* for measuring token significance has
895 also been validated by recent work such as TokenSkip (Xia et al., 2025).
896897

F THEORETICAL ANALYSIS OF SIGNIFICANCE-AWARE LENGTH REWARD

898 **Preliminaries and Notation.** Given a prompt x , the policy π_θ generates a chain-of-thought (CoT)
899 sequence $Y = (y_1, \dots, y_T)$, from which a deterministic decoder produces a final answer $\hat{Z} \in \mathcal{Z}$. To
900 assess the relative informativeness of each token, we compute a *significance score* using *LLMLingua-2*
(Pan et al., 2024):
901

902
$$S(y_i) = P(y_i | Y; \theta_{M_e}), \quad (14)$$

903 where θ_{M_e} denotes the parameters of a bidirectional encoder model. Unlike unidirectional predictors,
904 *LLMLingua-2* uses both left and right context to provide a holistic estimate of token informativeness.
905 Based on a fixed threshold τ , we partition the sequence as:
906

907
$$\mathcal{Y}_{\text{sig}} = \{y_i : S(y_i) \geq \tau\}, \quad \mathcal{Y}_{\text{insig}} = \{y_i : S(y_i) < \tau\}, \quad (15)$$

908 where \mathcal{Y}_{sig} and $\mathcal{Y}_{\text{insig}}$ denote the sets of *significant* and *insignificant* tokens, respectively.
909910 **Motivation for a Mutual Information Proxy.** In principle, each token’s importance could be
911 measured by its mutual information with the final answer, $I(y_i; Z^*)$. However, computing the exact
912 joint distribution $p(y_i, Z^*)$ is intractable due to the vast generation space and limited supervision.
913 Instead, we employ a proxy that is (i) efficient to compute for each token and (ii) monotonically
914 correlated with $I(y_i; Z^*)$.
915916 *LLMLingua-2* (Pan et al., 2024) satisfies these requirements by training under an information bottle-
917 neck objective:
918

919
$$I(T; Y) - \beta I(T; Z^*), \quad (16)$$

918 where T is the retained subsequence. Tokens with low conditional probability typically carry little
 919 additional information about Z^* , while high-probability tokens preserve essential semantics.
 920

921 **Assumption 1** (Fidelity of the Mutual Information Proxy). There exist constants $c > \varepsilon > 0$ such that
 922

$$923 \quad I(y_i; Z^*) \leq \varepsilon \quad (\forall y_i \in \mathcal{Y}_{\text{insig}}), \quad I(y_j; Z^*) \geq c - \varepsilon \quad (\forall y_j \in \mathcal{Y}_{\text{sig}}). \quad (17)$$

925 **Lemma 1** (Bounded Accuracy Loss). Let \hat{Z}_{full} denote the answer decoded from the full CoT, and \hat{Z}_{sig}
 926 the answer decoded after removing $\mathcal{Y}_{\text{insig}}$. Under Assumption 1, the increase in error probability is
 927 bounded:
 928

$$929 \quad \left| \Pr[\hat{Z}_{\text{sig}} \neq Z^*] - \Pr[\hat{Z}_{\text{full}} \neq Z^*] \right| \leq \varepsilon. \quad (18)$$

931 *Proof.* Let $Y = (y_1, \dots, y_T)$ and $Y_{\text{sig}} = Y \setminus \mathcal{Y}_{\text{insig}}$. By the chain rule:
 932

$$933 \quad I(Y; Z^*) = I(Y_{\text{sig}}; Z^*) + \sum_{y_i \in \mathcal{Y}_{\text{insig}}} I(y_i; Z^* \mid Y_{<i}), \quad (19)$$

936 and each term in the sum is at most ε . Therefore,
 937

$$938 \quad I(Y; Z^*) - I(Y_{\text{sig}}; Z^*) \leq T\varepsilon, \quad (20)$$

939 and by Fano's inequality, this gap translates into an error increase of at most ε . \square
 940

942 **Definition 1** (General vs. Significance-Aware Length Reward). For a generated trace Y , define two
 943 reward functions:
 944

$$945 \quad R_{\text{len}}(Y) = \mathbb{1}[\hat{Z}(Y) = Z^*] - \lambda |Y|, \quad (21)$$

$$946 \quad R_{\text{sig}}(Y) = \mathbb{1}[\hat{Z}(Y) = Z^*] - \lambda |\mathcal{Y}_{\text{insig}}|. \quad (22)$$

948 Here, R_{len} penalizes total length, while R_{sig} penalizes only insignificant tokens.
 949

950 **Theorem 1** (Benefit of the Significance-Aware Reward). Let π_θ be updated by a single PPO step
 951 using either reward, with the same coefficient $\lambda > 0$. If
 952

$$953 \quad \lambda > \frac{\varepsilon}{\mathbb{E}_{\pi_\theta}[|\mathcal{Y}_{\text{sig}}|]}, \quad (23)$$

955 then

$$956 \quad \mathbb{E}_{\pi_\theta}[R_{\text{sig}}(Y)] > \mathbb{E}_{\pi_\theta}[R_{\text{len}}(Y)]. \quad (24)$$

959 *Proof.* Lemma 1 implies

$$961 \quad \mathbb{E}[R_{\text{sig}}] - \mathbb{E}[R_{\text{len}}] = \lambda \mathbb{E}[|\mathcal{Y}_{\text{sig}}|] - \Delta_{\text{acc}}, \quad (25)$$

962 where

$$963 \quad 0 \leq \Delta_{\text{acc}} \leq \varepsilon. \quad (26)$$

965 Under the stated bound on λ , the difference is strictly positive. \square
 966

967 **Practical Implication.** The significance-aware reward achieves the same or greater length reduction
 968 with provably smaller accuracy degradation than a general length reward. By selectively penalizing
 969 insignificant tokens, it still encourages conciseness while maintaining fidelity. With *LLMLingua-2*
 970 providing a fast proxy for token–answer informativeness, this reward design supports both principled
 971 and practical optimization for efficient reasoning.

972 G THEORETICAL DISCUSSION OF DYNAMIC LENGTH REWARD 973

974 We provide a theoretical discussion of the motivation for our dynamic length reward schedule by
975 addressing three key questions:
976

- 977 1. Why does encouraging longer chains of thought (CoT) during early training help exploration?
- 978 2. Why does applying a fixed length penalty throughout training limit performance?
- 979 3. Why does dynamically flipping the reward from positive to negative upon convergence yield
980 better accuracy–efficiency trade-offs?
981

982 1. Longer CoT Enables Richer Exploration.

983 Let $P_t(L)$ denote the model’s distribution over output length L at training step t . Define the expected
984 accuracy given length L as
985

$$986 \text{Acc}(L) = \Pr[\hat{Z} = Z^* \mid L(Y) = L], \quad (27)$$

987 where \hat{Z} is the predicted answer and Z^* the ground-truth. Empirically, $\text{Acc}(L)$ follows a saturating
988 “S-curve”:
989

$$990 \frac{d}{dL} \text{Acc}(L) > 0 \quad \text{for } L < L^*, \quad \frac{d}{dL} \text{Acc}(L) \approx 0 \quad \text{for } L \geq L^*, \quad (28)$$

991 where L^* is the length at which accuracy saturates. The expected accuracy at step t is
992

$$993 \text{Acc}_t = \sum_L P_t(L) \text{Acc}(L). \quad (29)$$

995 Shifting probability mass toward longer CoT (up to L^*) thus increases Acc_t , since longer reasoning
996 expands exploration and raises the chance of discovering correct solution patterns.
997

998 **Takeaway.** Rewarding longer CoT early boosts exploration and accelerates convergence toward
999 high accuracy.
1000

1001 2. Static Length Penalty Causes Premature Compression.

1003 Consider a fixed penalty $\lambda > 0$, yielding reward

$$1004 J_{\text{static}}(L) = \text{Acc}(L) - \lambda L. \quad (30)$$

1006 The optimal length L_s under this objective satisfies

$$1008 \frac{d}{dL} \text{Acc}(L) \Big|_{L=L_s} = \lambda. \quad (31)$$

1010 Since $\frac{d}{dL} \text{Acc}(L)$ vanishes for $L \geq L^*$, any $\lambda > 0$ forces $L_s < L^*$, implying
1011

$$1012 \text{Acc}(L_s) < \text{Acc}(L^*). \quad (32)$$

1013 Thus the model truncates its CoT before accuracy has fully converged.
1014

1015 **Takeaway.** Static penalties enforce efficiency too early, sacrificing potential accuracy gains.
1016

1017 3. Dynamic Penalty Supports a Two-Phase Curriculum.

1019 We introduce a time-dependent penalty λ_t :

$$1021 \lambda_t = \begin{cases} \gamma, & t < t_0 \quad (\text{exploration phase}), \\ 1022 \alpha(t - t_0), & t \geq t_0 \quad (\text{compression phase}), \end{cases} \quad (33)$$

1023 where $\gamma < 0$, and t_0 is the step at which validation accuracy stabilizes, i.e.,
1024

$$1025 \Delta \text{Acc}_t = \frac{\text{Acc}_t - \text{Acc}_{t-\Delta}}{\Delta} < \beta. \quad (34)$$

1026 **Phase I (Exploration).** During early training, we set $\lambda_t < 0$, effectively turning the penalty into a
 1027 bonus:

$$1028 \quad J(L) = \text{Acc}(L) - \lambda_t L, \quad \text{with } -\lambda_t > 0,$$

1029 which encourages longer outputs. Since $\text{Acc}(L)$ increases with L up to L^* , this promotes
 1030

$$1031 \quad L_t \rightarrow L^*, \quad \text{Acc}_t \rightarrow \text{Acc}(L^*). \quad (35)$$

1032 **Phase II (Compression).** As training progresses, λ_t transitions from negative to positive and
 1033 increases gradually. When $\lambda_t > 0$, the derivative of the reward at L^* is
 1034

$$1035 \quad \frac{d}{dL} [\text{Acc}(L) - \lambda_t L] \Big|_{L=L^*} = \frac{d}{dL} \text{Acc}(L^*) - \lambda_t < 0, \quad (36)$$

1038 so extending beyond L^* reduces reward. The policy thus shortens to a new equilibrium L_d :

$$1039 \quad \frac{d}{dL} \text{Acc}(L) \Big|_{L=L_d} = \lambda_t, \quad L_d < L^*, \quad \text{Acc}(L_d) \approx \text{Acc}(L^*). \quad (37)$$

1042 **Comparison to Static Penalty.** The final dynamic reward is
 1043

$$1044 \quad J_{\text{dyn}} = \text{Acc}(L_d) - \lambda_T L_d, \quad (38)$$

1045 and under the concavity of $\text{Acc}(L)$ one can show

$$1047 \quad J_{\text{dyn}} - J_{\text{static}} = [\text{Acc}(L_d) - \text{Acc}(L_s)] - \lambda_T (L_d - L_s) \geq 0, \quad (39)$$

1048 i.e., dynamic scheduling yields no worse and often strictly better reward. This holds because for
 1049 concave functions

$$1050 \quad \text{Acc}(L_d) - \text{Acc}(L_s) \geq \text{Acc}'(L_d)(L_d - L_s), \quad (40)$$

1051 and with $\text{Acc}'(L_d) = \lambda_T$, the inequality follows.

1053 **Efficiency Metric.** Define length-normalized accuracy
 1054

$$1055 \quad \text{L-Acc}(L) = \text{Acc}(L) \sqrt{1 - \frac{L}{L_{\max}}}. \quad (41)$$

1057 In practice, dynamic scheduling often achieves similar or higher accuracy with shorter or comparable
 1058 length, leading to

$$1059 \quad \text{L-Acc}(L_d) > \text{L-Acc}(L_s). \quad (42)$$

1062 **Conclusion.** Our dynamic length reward realizes the curriculum

$$1063 \quad \text{explore freely } (\lambda \leq 0) \longrightarrow \text{accuracy convergence} \longrightarrow \text{gradual compression } (\lambda \uparrow). \quad (43)$$

1065 This schedule lets the model reach its accuracy ceiling $\text{Acc}(L^*)$ before enforcing brevity, achieving
 1066 better accuracy–efficiency trade-offs than static schemes.

1068 H DEFINITION AND THEORETICAL ANALYSIS OF LENGTH-NORMALIZED 1069 ACCURACY

1071 Length-Normalized Accuracy.

1073 To evaluate reasoning efficiency, we adopt a length-normalized accuracy metric, denoted as L-ACC,
 1074 which balances correctness with brevity. Formally, it is defined as:

$$1075 \quad \text{L-Acc} = \text{Acc} \times \sqrt{1 - \frac{L}{L_{\max}}}, \quad (44)$$

1078 where $\text{Acc} \in [0, 1]$ denotes exact-match accuracy, L is the number of tokens in the model’s response,
 1079 and L_{\max} is a dataset-specific upper bound on response length. The multiplicative factor penalizes
 longer outputs in a sub-linear manner, rewarding models that solve problems with fewer tokens.

1080 Specifically, we set $L_{\max} = 8192$ for the two
 1081 *DeepSeek*-based reasoning models, and $L_{\max} =$
 1082 3000 for the *Qwen2.5-Math-7B-Instruct* model, since
 1083 reasoning-oriented models generally generate longer
 1084 outputs than instruction-tuned models. The multi-
 1085 plicative factor $\sqrt{1 - \frac{L}{L_{\max}}}$ weights accuracy by a
 1086 sub-linear penalty on sequence length, so the met-
 1087 ric rewards correct solutions that are delivered with
 1088 fewer tokens. Normalizing by L_{\max} makes the score
 1089 comparable across datasets of very different scale,
 1090 while the square-root ensures a smooth, continuous
 1091 trade-off: the first tokens cut away improve the score
 1092 more than later ones, mirroring human tolerance for
 1093 moderate verbosity but aversion to extreme length.
 1094 When $L = L_{\max}$ the metric collapses to zero, pre-
 1095 venting models from exchanging unbounded length
 1096 for marginal accuracy gains; when $L = 0$ it reduces to the raw accuracy, preserving credit for
 1097 perfectly concise answers.

1098 **Penalty Behavior and Physical Intuition.** The penalty term $\sqrt{1 - \frac{L}{L_{\max}}}$ is continuous, monoton-
 1099 ically decreasing, and bounded between 0 and 1. It applies no penalty when $L = 0$, and reduces
 1100 the reward to zero when $L = L_{\max}$, even if the answer is correct. Crucially, the square-root form
 1101 introduces diminishing returns: trimming early redundant tokens provides larger gains in L-ACC
 1102 than removing tokens later in the sequence. This design mirrors human preferences—we tolerate
 1103 moderate verbosity, but disfavor excessive detail. It also echoes the behavior of L2 regularization,
 1104 where larger values are penalized more aggressively, while smaller deviations are softly constrained.

1105 **Gradient Analysis.** To understand its optimization implications, we analyze the gradient of the
 1106 penalty term with respect to L :

$$\frac{d}{dL} \left(\sqrt{1 - \frac{L}{L_{\max}}} \right) = -\frac{1}{2L_{\max}} \cdot \left(1 - \frac{L}{L_{\max}} \right)^{-1/2}. \quad (45)$$

1111 This derivative diverges as $L \rightarrow L_{\max}$, indicating that long outputs are heavily penalized. In contrast,
 1112 when L is small, the gradient approaches zero, and the penalty becomes negligible. This behavior
 1113 encourages models to first eliminate highly redundant tokens, while maintaining stability for shorter
 1114 outputs.

1115 **Optimization Benefits.** Unlike hard constraints on length, this formulation yields a smooth and
 1116 differentiable reward signal, making it well-suited for reinforcement learning algorithms such as
 1117 PPO. It provides stable guidance throughout training and enables the model to trade off between
 1118 accuracy and length in a controlled and interpretable manner. As shown in Figure 4, the penalty curve
 1119 strongly discourages excessively long outputs while allowing flexibility in moderately verbose cases,
 1120 contributing to more efficient and human-aligned reasoning. Notably, the curve becomes steep as
 1121 the response length approaches L_{\max} , meaning that small increases in length lead to sharp drops
 1122 in reward; conversely, it flattens near $L = 0$, where changes in length have only a minor effect on
 1123 the reward. This property ensures that the model is heavily penalized for extreme verbosity while
 1124 remaining tolerant of brief explanatory content.

1125 I DETAILED SETTINGS OF EXPERIMENTS

1128 **Prompt.** All experiments use the prompt: "Let's think step by step and output
 1129 the final answer within \boxed{}."

1130 **Models.** Our experiments involve a mix of proprietary and open-source models. The models evaluated
 1131 in this study include:

- 1133 • **DeepSeek-R1-Distill-Qwen-1.5B (MIT License):** A fine-tuned model with 1.5 billion parame-
 1134 ters, used to evaluate the proposed method.

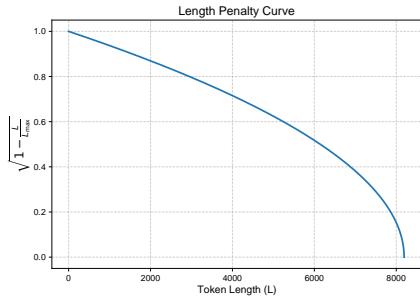


Figure 4: **Penalty curve:** $\sqrt{1 - \frac{L}{L_{\max}}}$.

1134
 1135 • **DeepSeek-R1-Distill-Qwen-7B (MIT License)**: A fine-tuned model with 7 billion parameters,
 1136 also used to evaluate the proposed method.
 1137 • **Qwen2.5-Math-7B-Instruct (Apache-2.0 License)**: An instruction-tuned model with 7 billion
 1138 parameters, used to further assess the efficiency and accuracy in reasoning tasks.

1139 **Datasets.** We evaluate our models on several datasets covering both in-distribution (ID) and out-of-
 1140 distribution (OOD) tasks. The evaluation framework encompasses:

1141 • **MATH** (Hendrycks et al., 2021): A comprehensive training dataset containing 7,500 mathemati-
 1142 cal problems across various difficulty levels and topics.
 1143 • **MATH500** (Hendrycks et al., 2021): A carefully selected 500-problem subset from the MATH
 1144 test set, serving as our primary in-distribution evaluation benchmark.
 1145 • **GSM8K** (Cobbe et al., 2021): Grade school math word problems requiring multi-step reasoning,
 1146 used for out-of-distribution evaluation on elementary-level mathematics.
 1147 • **TheoremQA** (Chen et al., 2023): A challenging dataset requiring theorem application and
 1148 symbolic reasoning across STEM domains, used for out-of-distribution evaluation.
 1149 • **AIME2024** (Veeraboina, 2023): Problems from the prestigious American Invitational Mathe-
 1150 matics Examination, representing the most challenging out-of-distribution evaluation.

1151 These datasets are arranged in increasing order of difficulty: $\text{GSM8K} < \text{MATH500} < \text{THEOREMQA}$
 1152 $< \text{AIME2024}$, offering a comprehensive evaluation of models' reasoning capabilities across varying
 1153 complexity levels, as summarized in Table 4.

1154
 1155 Table 4: Overview of datasets used for training and evaluation.
 1156

Type	Dataset	# Train	# Test	Domain	Task Type	Difficulty	Source
Training	MATH (Hendrycks et al., 2021)	7,500	–	Mathematics	Problem Solving	Mixed	Link
ID Test	MATH500 (Hendrycks et al., 2021)	–	500	Mathematics	Problem Solving	Medium-Hard	Link
OOD Test	GSM8K (Cobbe et al., 2021)	–	1,319	Elementary Math	Word Problems	Easy	Link
	TheoremQA (Chen et al., 2023)	–	800	STEM	Theorem Application	Hard	Link
	AIME2024 (Veeraboina, 2023)	–	30	Competition Math	Advanced Problem Solving	Very Hard	Link

1162 **Preprocessing and Tokenization.** Each model uses its corresponding tokenizer to process the input
 1163 sequences. Tokenization ensures compatibility with the model's input structure, using special tokens
 1164 to denote the start and end of sequences.

1165 **Training Procedure.** All models are trained for a total of 50 epochs using the Proximal Policy
 1166 Optimization (PPO) algorithm, optimizing for both accuracy and efficiency. The actor and critic
 1167 models are initialized with the same parameters, and training is conducted with the following
 1168 hyperparameters: actor learning rate = 5×10^{-5} , critic learning rate = 1×10^{-6} , mini-batch size
 1169 = 512, and KL-divergence coefficient = 0.001. Evaluation is performed periodically at every training
 1170 step to monitor progress, and the best model checkpoints are selected for final testing.

1171 **Decoding Configurations.** We conduct both training and evaluation under carefully controlled
 1172 decoding settings. During training, we adopt sampling generation with temperature = 1.0, top_k =
 1173 –1, and top_p = 1.0 to encourage exploration, following the default configuration of the VERL
 1174 framework for comparability with prior work. A single response ($n = 1$) is generated per prompt,
 1175 with the maximum prompt length capped at 1,024 tokens for efficiency. The maximum response
 1176 length is set to 8,192 tokens for DeepSeek models and 3,000 tokens for Qwen-Math models.

1177 For evaluation, we emphasize efficiency and stability by adopting greedy decoding, consistent with
 1178 VERL defaults and prior studies (Yeo et al., 2025; Cui et al., 2025). Specifically, evaluation uses
 1179 greedy decoding with temperature = 0, and one response per input ($n = 1$). The same maximum
 1180 response lengths as in training are applied (8,192 for DeepSeek, 3,000 for Qwen-Math). We also
 1181 conducted experiments under extended sampling configurations, with comprehensive results presented
 1182 in Appendix J.

1183 **Evaluation Metrics.** We evaluate the models using the following metrics:

1184 • **Exact Match (EM)**: Measures the proportion of exact matches between the generated output
 1185 and the ground-truth answer.
 1186 • **Response Length (Len)**: Measures the number of tokens in the output sequence.

1188 • **Length-Normalized Accuracy (L-Acc):** A metric that balances accuracy and efficiency by
 1189 considering both correctness and response length.
 1190

1191 **Baselines.** We compare the BINGO framework with the following baselines:

1192 • **DAST** (Shi et al., 2025): Uses dynamic length penalties based on problem difficulty.
 1193 • **Efficient Reasoning** (Arora & Zanette, 2025): Scales down positive rewards to encourage
 1194 brevity.
 1195 • **Kimi-k1.5** (Team et al., 2025): Applies online length penalties.
 1196 • **O1-Pruner** (Luo et al., 2025): Applies offline length penalties based on length comparisons
 1197 with reference sequences.
 1198 • **Demystifying** (Yeo et al., 2025): Applies a symmetric penalty strategy for response lengths,
 1199 encouraging both shorter and more extensive reasoning depending on correctness.

1200 Rather than directly comparing published baselines—which employ diverse frameworks (e.g., SimPO,
 1201 GRPO) and differ in their on-policy versus off-policy implementations—we isolate and re-implement
 1202 only the length-based reward components proposed in each work. All these reward designs are
 1203 integrated into a unified PPO framework. This approach enables a fair comparison focused specifically
 1204 on the effectiveness of different reward formulations for improving reasoning efficiency.
 1205

1206 **Software and Hardware.** The experiments are conducted with Python 3.11, PyTorch v2.4.0, and
 1207 CUDA 12.8 for model training and inference. We use 4 NVIDIA A100 80GB PCIe GPUs for training
 1208 the 7B model and 2 NVIDIA H100 80GB PCIe GPUs for training the 1.5B model. For inference, 2
 1209 NVIDIA H100 80GB PCIe GPUs are used to accelerate processing.
 1210

1210 J PERFORMANCE UNDER EXTENDED SAMPLING SETTINGS

1211 We conducted additional experiments using sampling decoding to assess the robustness of our
 1212 approach under more exploratory conditions. These experiments employed an extended configuration
 1213 with a 32,768-token output limit, three samples per prompt, temperature of 0.6, and top-p of 1.0.
 1214 We evaluated the base DeepSeek-R1-Distill-Qwen-1.5B model, vanilla PPO, our proposed Bingo
 1215 method, and selected competitive baselines to ensure comprehensive comparison.
 1216

1217 **Table 5: Performance comparison under sampling decoding settings.** Each method is evaluated
 1218 using DeepSeek-R1-Distill-Qwen-1.5B as the base model with sampling parameters (32,768 token
 1219 limit, 3 samples, temperature = 0.6, top-p = 1.0). Metrics include answer accuracy (Acc, %), response
 1220 length (Len), and length-normalized accuracy (L-Acc, %). The best performance is highlighted in
 1221 **dark blue**, and the second-best in **light blue**.
 1222

Method	MATH500			GSM8K			TheoremQA			AIME2024		
	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑
Base	81.6	5,155	74.9	83.7	1,748	81.4	31.7	7,598	27.8	17.8	15,703	12.8
Vanilla PPO	82.3	2,694	78.8	86.5	1,050	85.1	33.4	3,616	31.5	28.9	7,389	25.4
O1-Pruner	80.1	1,283	78.5	85.2	352	84.7	34.7	1,095	34.1	28.9	4,636	26.8
Demystifying	81.3	1,945	78.9	86.8	483	86.2	35.2	1,863	34.2	30.0	5,891	27.2
DAST	83.5	2,053	80.8	84.1	375	83.6	35.4	2,954	33.8	36.7	5,072	33.7
Bingo (Ours)												
Bingo-A	85.1	1,114	83.6	88.4	483	87.7	37.9	1,592	37.0	38.9	3,110	37.0
Bingo-E	84.2	983	82.9	88.1	217	87.8	37.7	1,004	37.1	37.6	2,817	35.9

1232 Table 5 presents the accuracy and average response length across four benchmarks under these
 1233 sampling conditions. The results demonstrate that Bingo maintains its efficiency advantage even
 1234 with sampling decoding, achieving strong accuracy while generating substantially shorter outputs
 1235 than all baseline methods. This finding confirms that our reward design effectively promotes concise
 1236 reasoning regardless of the decoding strategy employed.
 1237

1238 Although these extended settings yielded accuracy improvements, they required approximately five
 1239 times the computational resources and training time compared to greedy decoding. Given this
 1240 substantial computational overhead, we selected single-response greedy decoding as our primary
 1241 evaluation protocol to maintain experimental feasibility while still providing meaningful performance
 1242 assessments. The sampling results presented here validate that our approach remains effective under
 1243 more computationally intensive conditions.
 1244

Table 6: **Comparison of reinforcement learning algorithms on four reasoning benchmarks.** Each method is evaluated using DeepSeek-R1-Distill-Qwen-1.5B as the base model by answer accuracy (Acc, %), response length (Len), and length-normalized accuracy (L-Acc, %). Bingo-based variants consistently outperform their vanilla counterparts across different RL optimizers (PPO, RLOO, GRPO, Reinforce++). Numbers in parentheses show the L-Acc gain over the corresponding vanilla baseline, with **green** indicating improvement.

Method	MATH500			GSM8K			TheoremQA			AIME2024		
	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑	Acc↑	Len↓	L-Acc↑
Base	63.2	3,913	45.7	73.2	2,025	63.5	18.7	5,741	10.3	16.7	7,027	6.3
Vanilla PPO	81.4	2,771	66.2	85.4	1,310	78.2	32.3	4,146	22.7	26.7	6,961	10.3
Bingo-PPO	82.2	894	77.6 (+11.4)	87.0	563	83.9 (+5.7)	36.8	1,648	32.9 (+10.2)	33.3	2,943	26.7 (+16.4)
Vanilla RLOO	76.8	2,413	64.5	77.3	1,588	69.4	30.0	3,162	23.5	26.7	6,025	13.7
Bingo-RLOO	78.0	1,985	67.9 (+3.4)	80.7	450	78.5 (+9.1)	32.0	2,230	27.3 (+3.8)	33.3	5,583	18.8 (+5.1)
Vanilla GRPO	76.4	2,533	63.5	77.8	804	73.9	29.2	2,946	23.4	26.7	6,096	13.5
Bingo-GRPO	79.4	1,753	70.4 (+6.9)	80.0	449	77.8 (+3.9)	31.9	2,298	27.0 (+3.6)	30.0	5,886	15.9 (+2.4)
Vanilla Reinforce++	76.2	2,842	61.6	82.0	1,291	75.2	28.0	3,977	20.1	30.0	6,168	14.9
Bingo-Reinforce++	78.4	2,070	67.8 (+6.2)	81.0	640	77.8 (+2.6)	33.1	2,566	27.4 (+7.3)	30.0	5,885	15.9 (+1.0)

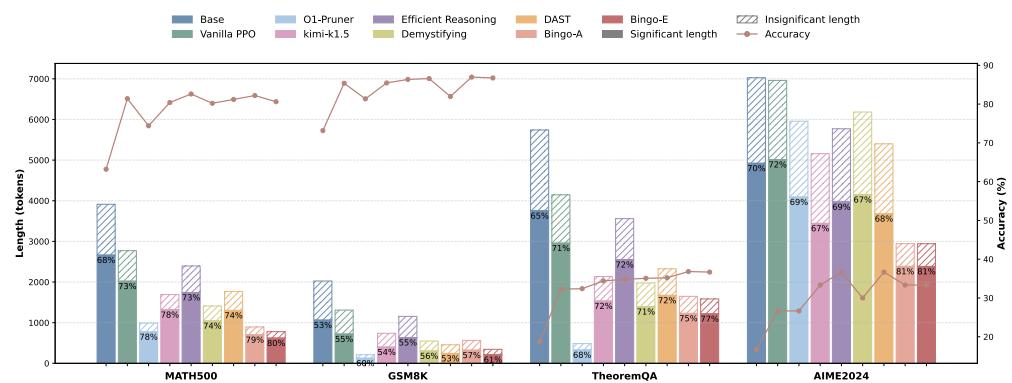


Figure 5: **Length-accuracy results for nine optimization algorithms on four datasets.** Bars show the number of tokens generated using DeepSeek-R1-Distill-Qwen-1.5B as the base model at the checkpoint that yields the reported accuracy (left axis). Each bar is partitioned into significant (dark) and insignificant (light) segments, and the percentage above the bar indicates the share of significant tokens. The solid line (right axis) gives the corresponding answer accuracy. Our methods, Bingo-A and Bingo-E, attain the highest accuracy while using the fewest tokens and achieving the greatest proportion of significant tokens, highlighting their superior reasoning efficiency.

K PERFORMANCE ACROSS DIFFERENT REINFORCEMENT LEARNING ALGORITHMS

To evaluate the generalizability of our reward design, we integrate it into multiple on-policy reinforcement learning (RL) algorithms, including PPO, RLOO, GRPO, and Reinforce++. As shown in Table 6, Bingo-enhanced variants consistently outperform their vanilla counterparts across all four benchmarks in both accuracy and length-normalized accuracy (L-Acc). Among them, BINGO-PPO delivers the strongest overall performance, achieving the highest or second-highest scores on all datasets while substantially reducing output length. Moreover, the benefits of Bingo extend beyond PPO: each Bingo variant improves L-Acc over its baseline by a clear margin, demonstrating that our reward formulation generalizes well across different policy optimization strategies. These gains are observed not only on in-distribution datasets like MATH500 and GSM8K, but also on more challenging out-of-distribution settings such as AIME2024, highlighting the robustness of our approach. Overall, the results confirm that a principled and learnable length-aware reward offers a consistent advantage across a variety of reasoning tasks and RL algorithms.

1296 **Problem:** Terry eats 2 yogurts a day. They are currently on sale at 4 yogurts for \$5.00. How much does he
 1297 spend on yogurt over 30 days?
 1298 **Response of BINGO:**
 1299 Okay, so Terry eats 2 yogurt each day, and they're on sale where 4 yogurt cost \$5.00. I need to figure out how much he
 1300 spends over 30 days. Hmm, let me break this down step by step. ... Since 4 yogurt cost \$5, and he needs 60 yogurt, which
 1301 is 15 sets, so 15 times \$5 is \$75. Terry spends 75.00 dollars on yogurt over 30 days.
 1302 (553 tokens with 57% Significant)
 1303 **Response of Vanilla PPO:**
 1304 Okay, so Terry eats 2 yogurt per day. He's planning to buy them on sale where they're 4 for \$5.00. I need to figure out
 1305 how much he'll spend over 30 days. Wait, let me double-check that. ... Wait, another way to think about it: ... so $12 * \$5$
 1306 = \$60. 52 yogurt is 13 sets, so $13 * \$5 = \65 . 56 yogurt is 14 sets, so $14 * \$5 = \70 . 60 yogurt is 15 sets, so $15 * \$5 = \75 .
 1307 Yep, that's consistent. I think that's solid. So, Terry spends \$75.00 on yogurt over 30 days. Terry spends 75 dollars on
 1308 yogurt over 30 days. (916 tokens with 52% Significant)
Final Answer: 75.

1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316 **Figure 6: Token-level significance visualization for a sample reasoning task.** Each token is colored
 1317 based on its predicted significance: red indicates significant tokens (darker = more significant), and
 1318 blue indicates insignificant tokens (darker = less significant). The response from BINGO (top) is
 1319 shorter and more concentrated around meaningful reasoning steps, while the Vanilla PPO response
 1320 (bottom) is longer and contains more exploratory and redundant language. The visualization illustrates
 1321 how Bingo encourages more efficient and focused reasoning.
 1322

L ANALYSIS OF SIGNIFICANT TOKEN RATIO

1323 We employ the DeepSeek-R1-Distill-Qwen-1.5B model, trained exclusively on the MATH
 1324 corpus. To evaluate its generalization beyond the training distribution, we test the model on the
 1325 in-distribution split MATH500 as well as three out-of-distribution (OOD) benchmarks: GSM8K,
 1326 THEOREMQA, and AIME2024. Figure 5 shows that our approaches, **Bingo-A** and **Bingo-E**, achieve
 1327 the most favorable length-accuracy trade-off across all four benchmarks.

- 1328 **Efficiency at peak accuracy.** At the checkpoints that obtain their highest accuracy, both Bingo
 1329 variants require only about 20% of the tokens used by the *Base* model on MATH500, with similarly
 1330 large reductions on GSM8K, THEOREMQA, and AIME2024.
- 1331 **Preservation of informative content.** Bingo increases the share of significant tokens to 75–81%,
 1332 showing that the shortened rationales shed mainly redundant rather than essential reasoning steps.
- 1333 **Difficulty-dependent length trends.** Token counts grow with task difficulty: the two harder
 1334 benchmarks, THEOREMQA and AIME2024, demand considerably longer rationales and yield
 1335 lower absolute accuracy than MATH500 and GSM8K. Even under these tougher conditions,
 1336 Bingo still delivers the highest accuracy while generating the fewest tokens.
- 1337 **Alleviating the length-accuracy trade-off.** Baselines that compress reasoning without account-
 1338 ing for token importance (e.g., *O1-Pruner*) exhibit marked accuracy declines, whereas Bingo
 1339 maintains—and in some cases slightly improves—task performance.
- 1340 **Robustness across tasks.** The same advantage holds for algebraic, commonsense, formal-logic,
 1341 and competition-style benchmarks, underscoring the generality of the significance-aware and
 1342 dynamic length rewards.

1343 These findings confirm that explicitly modeling token significance and adaptively scheduling length
 1344 rewards enables language models to reason both *accurately* and *efficiently*.

M TOKEN-LEVEL SIGNIFICANCE VISUALIZATION

1345 Figure 6 provides a token-level significance visualization for a sample reasoning task. The problem
 1346 involves calculating the cost of yogurt based on a given sale, and both the BINGO and Vanilla PPO
 1347 models generate responses to solve the problem. Each token in the generated response is color-coded
 1348 based on its predicted significance, with red indicating significant tokens and blue representing
 1349 insignificant ones. Darker shades of red and blue correspond to higher significance levels.

The response from BINGO (top) is notably shorter and more focused on the key reasoning steps,
 highlighting the model's ability to concentrate on relevant tokens while avoiding unnecessary elabora-

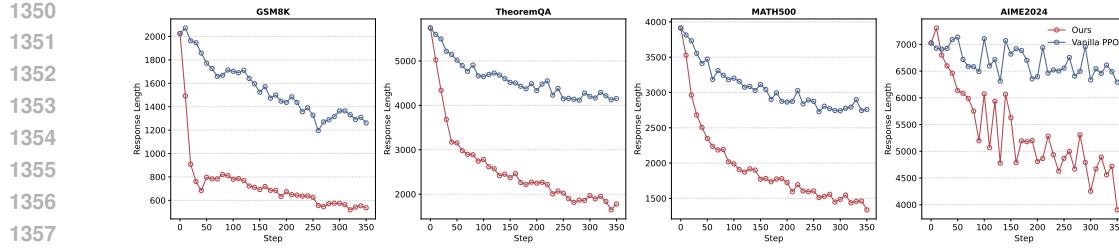


Figure 7: **Response length trends during training across four datasets.** The y-axis shows the number of tokens generated per response using DeepSeek-R1-Distill-Qwen-1.5B as the base model; the x-axis denotes training steps. The red line represents our method, and the blue line corresponds to Vanilla PPO. Across all tasks, our method consistently produces shorter and more stable responses, demonstrating improved reasoning efficiency without compromising task performance.

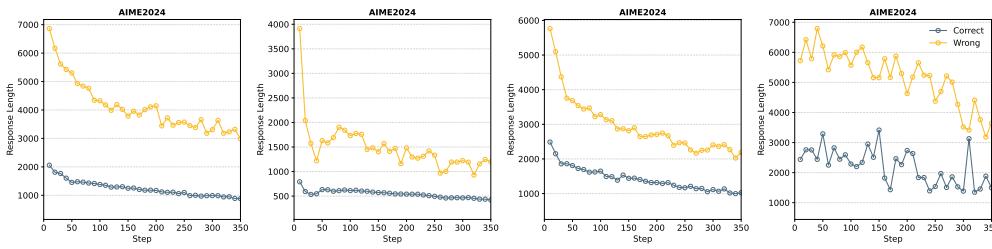


Figure 8: **Response length dynamics for correct vs. wrong samples during training.** The x-axis indicates training steps, and the y-axis denotes response length in tokens for models trained on DeepSeek-R1-Distill-Qwen-1.5B as the base model. The blue line tracks correctly answered samples, while the yellow line tracks incorrectly answered samples. In the early stages, incorrect samples produce substantially longer responses, reflecting the effect of our length-incentive mechanism. As the dynamic length reward gradually diminishes, the response length for incorrect samples falls more sharply than that for correct samples, illustrating the model’s adaptive pruning of redundant reasoning steps.

tion. In contrast, the Vanilla PPO response (bottom) is longer, with a higher proportion of redundant and less informative tokens, reflecting a less efficient reasoning process. This visualization clearly demonstrates how BINGO encourages more concise and targeted reasoning, optimizing for both accuracy and efficiency by emphasizing significant steps in the reasoning process.

N ANALYSIS OF RESPONSE LENGTHS TRENDS DURING TRAINING

Figure 7 presents the evolution of response length over training steps for Vanilla PPO and our method on four benchmarks. Our approach consistently yields substantially shorter outputs than Vanilla PPO throughout training, demonstrating effective removal of redundant tokens, and converges more smoothly, reflecting robust length regularization. The reduction in response length is most pronounced on the more demanding tasks—MATH500 and AIME2024—where Vanilla PPO produces very long sequences, yet our method maintains a compact reasoning footprint. Importantly, this improvement generalizes across diverse reasoning styles, from arithmetic problems in GSM8K and formal-logic questions in THEOREMQA to academic and competition-style challenges, confirming that our reward design enhances reasoning efficiency without compromising training stability.

O ANALYSIS OF RESPONSE LENGTHS DYNAMICS FOR CORRECT VS. WRONG SAMPLES

Figure 8 illustrates how response length evolves for correct and incorrect samples under our approach. In the early phase of training, incorrect samples produce markedly longer outputs than correct ones, demonstrating the impact of our length-incentive mechanism in promoting thorough exploration on

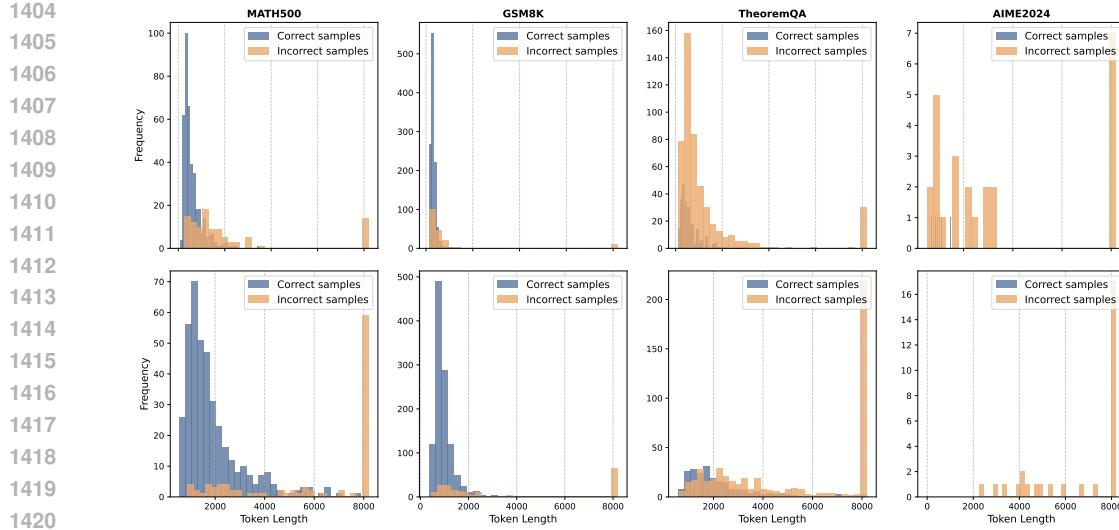


Figure 9: **Distribution of response lengths for correct vs. incorrect samples.** Histograms show the frequency of token lengths in model outputs across four benchmarks using DeepSeek-R1-Distill-Qwen-1.5B as the base model. Each plot compares correct responses (blue) and incorrect responses (orange). The top row corresponds to our method, while the bottom row shows results from Vanilla PPO. Across all datasets, incorrect samples are more likely to produce longer outputs, while correct samples tend to cluster in shorter length ranges. Compared to Vanilla PPO, our method produces a sharper, more compact distribution concentrated in shorter length regions.

challenging cases. As the dynamic length reward takes effect around mid-training, the length for wrong samples declines steeply—outpacing the reduction seen for correct samples—and the gap between the two curves narrows. By later stages, both curves converge toward similarly concise rationales, indicating that the model has learned to apply efficient reasoning uniformly. This behavior confirms that our combination of significance-aware and dynamic rewards not only drives exploration where needed but also enforces brevity once sufficient understanding is achieved, resulting in a balanced, adaptive pruning of redundant tokens.

To examine how response length relates to answer correctness, we compare output length distributions of our method and the Vanilla PPO baseline across four benchmarks using *DeepSeek-R1-Distill-Qwen-1.5B*. As shown in Figure 9, correct responses consistently exhibit shorter lengths than incorrect ones across all tasks. Our method further produces sharply concentrated distributions for correct samples, suggesting more focused and efficient reasoning. In contrast, Vanilla PPO outputs are generally longer and more dispersed, with substantial overlap between correct and incorrect cases. Notably, the length of incorrect samples is substantially reduced compared to Vanilla PPO, suggesting that the dynamic reward mechanism—which gradually penalizes verbosity during training—plays a role in guiding more efficient responses.

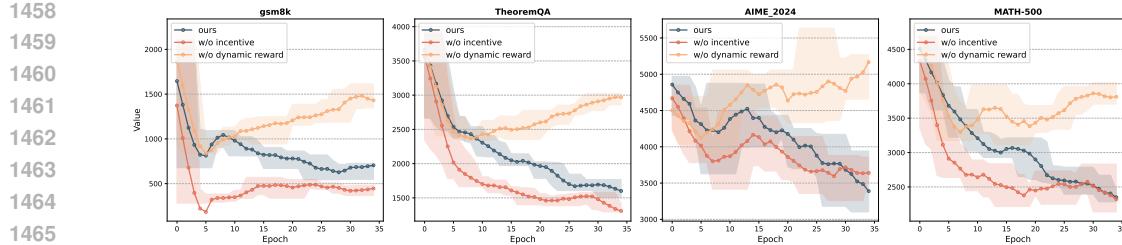


Figure 10: Effect of Reward Design on Incorrect Response Length. We visualize the average significant response length of incorrect predictions during training on four benchmarks using DeepSeek-R1-Distill-Qwen-1.5B as the base model. Compared to the variant without incentive, our full method produces longer responses for incorrect samples, suggesting that the significance-aware reward encourages more thorough exploration when the model is uncertain. In contrast, removing the dynamic reward leads to persistently longer outputs, whereas our full method shows a clear reduction in response length over time, confirming the effectiveness of dynamic reward scheduling in promoting concise reasoning. Together, these trends highlight the complementary roles of the two reward components in balancing exploration and efficiency.

P ANALYSIS OF INCORRECT RESPONSE LENGTH UNDER DIFFERENT REWARD DESIGNS

To gain deeper insights into the effect of our reward design during training, we analyze the significant response length of incorrect predictions—a proxy for the model’s exploratory behavior under uncertainty. As shown in Figure 10, our full method generates longer responses for incorrect samples compared to the version **without incentive** across all four benchmarks, indicating that the significance-aware reward successfully promotes deeper reasoning attempts when the model is uncertain. In contrast, removing the dynamic reward leads to consistently longer outputs throughout training, while our method gradually reduces response length over time. This trend confirms that dynamic reward scheduling effectively encourages concise reasoning as the model matures. Together, these findings highlight the complementary roles of the two reward components: one enhances exploration during failure, and the other improves efficiency during learning progression.

1512 Q CASE STUDY
1513

1514 To further assess the reasoning quality and output efficiency of our method, we conduct two case
1515 studies comparing the Base model, PPO, and our BINGO framework across two model scales. As
1516 shown in Figures 11 and 12, the Base model often produces excessively long and repetitive outputs,
1517 sometimes exceeding 5,000 words, and either fails to converge or hides the correct answer deep
1518 within verbose reasoning or terminated generations without a final answer. PPO improves conciseness
1519 but occasionally generates incorrect outputs or retains unnecessary repetition in intermediate steps. In
1520 contrast, BINGO consistently produces correct, well-structured solutions while significantly reducing
1521 output length—achieving up to 3x compression over PPO in the 7B setting. These examples demon-
1522 strate the effectiveness of our reward design in balancing reasoning depth and brevity, encouraging
1523 the model to generate focused and efficient reasoning even on complex tasks.

1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565

1566

1567

1568

1569

1570

1571

1572

1573

1574

1575

1576

1577

1578

1579

1580

1581

1582

1583

1584

1585

1586

1587

1588

1589

1590

1591

1592

1593

1594

1595

1596

1597

1598

1599

1600

1601

1602

1603

1604

1605

1606

1607

1608

1609

1610

1611

1612

1613

1614

1615

1616

1617

Problem: Find the greatest integer less than $(\sqrt{7} + \sqrt{5})^6$. (Do not use a calculator!)

X Response of Base Model with 2509 words:

Okay, so I need to find the greatest integer less than $((\sqrt{7} + \sqrt{5})^6)/4$. Hmm, that sounds a bit intimidating at first, but maybe I can break it down step by step. Let me think about how to approach this without a calculator, so I have to rely on algebraic manipulations and maybe some patterns or properties of exponents. First, I remember that expressions like $((\sqrt{a} + \sqrt{b})^n)$ can sometimes be simplified using binomial expansions, but that might get messy for the sixth power. Alternatively, I recall that sometimes pairing such terms can lead to simplifications. Maybe I can consider the conjugate of $(\sqrt{7} + \sqrt{5})$, which is $(\sqrt{7} - \sqrt{5})$, and see if their product is a nice number. Let me try compute ... Maybe I can find a recurrence relation or some pattern for $(x^n + y^n)$. Let me try computing some lower powers to see if I can spot a pattern. Let's compute $(x + y) : (x + y = (\sqrt{7} + \sqrt{5}) + (\sqrt{7} - \sqrt{5}) = 2\sqrt{7})$. Hmm, that's interesting. Let's compute ... Okay, so $(x^2 + y^2 = 24)$. Let's compute $(x^3 + y^3)$. Maybe I can use the identity $(x^3 + y^3 = (x + y)(x^2 - xy + y^2))$ But when I computed (x^6) directly, I got approximately 13535.988656. So, which is correct? Wait, perhaps I made a mistake in the direct expansion. Let me check the expansion again. Wait, in the direct expansion, I had: $(x^6 = 6768 + 1144\sqrt{35})$. So, $1144 \times \sqrt{35} \approx 1144 \times 5.916074 \approx 6767.988656$. So, ... So, which is correct? Wait, perhaps I made a mistake in the direct expansion. Let me check the expansion step by step. ... So, that's correct. But in the direct expansion, I have $(x^6 \approx 13535.988674)$. So, which is correct? Wait, perhaps I made a mistake in the direct expansion. Let me check the expansion again. Wait, ...

X Response of PPO with 403 words:

To find the greatest integer less than $(\sqrt{7} + \sqrt{5})^6$, we can use the property that $(\sqrt{7} + \sqrt{5})^n + (\sqrt{7} - \sqrt{5})^n$ is an integer for any positive integer n . Since $\sqrt{7} - \sqrt{5}$ is a small positive number (approximately 0.4142), $(\sqrt{7} - \sqrt{5})^6$ will be a small number close to 0. Let's denote $x = \sqrt{7} + \sqrt{5}$ and $y = \sqrt{7} - \sqrt{5}$. Then, $x + y = 2\sqrt{7}$ and $x - y = 2\sqrt{5}$. However, directly using this might not help. Instead, we know that $x^6 + y^6$ is an integer. Since $y < 1$, y^6 is very small, so x^6 is slightly less than an integer. Therefore, the greatest integer less than x^6 is $x^6 + y^6 - 1$. We need to compute $x^6 + y^6$. To find this, we can use recurrence relations or expand step by step. However, a more efficient way is to compute powers step by step: First, compute $x^2 = (\sqrt{7} + \sqrt{5})^2 = 7 + 2\sqrt{35} + 5 = 12 + 2\sqrt{35}$. Next, $y^2 = (\sqrt{7} - \sqrt{5})^2 = 7 - 2\sqrt{35} + 5 = 12 - 2\sqrt{35}$. Then, $x^3 + y^3 = (x + y)(x^2 + y^2 - xy) = 2\sqrt{7} \cdot (24 - 5) = 2\sqrt{7} \cdot 19 = 38\sqrt{7}$. Similarly, $x^3 - y^3 = (x - y)(x^2 + y^2 + xy) = 2\sqrt{5} \cdot (24 + 5) = 2\sqrt{5} \cdot 29 = 58\sqrt{5}$. However, this is getting complex. Instead, we can compute $x^6 + y^6$ using the recurrence relation. Let $S_n = x^n + y^n$. Then, $S_0 = 2$, $S_1 = 2\sqrt{7}$, $S_2 = 24$, $S_3 = 38\sqrt{7}$, $S_4 = 24 \cdot S_2 - S_1^2 = 24 \cdot 24 - (2\sqrt{7})^2 = 576 - 28 = 548$, $S_5 = S_4 \cdot S_1 - S_3 \cdot S_2 = 548 \cdot 2\sqrt{7} - 38\sqrt{7} \cdot 12 = (1096\sqrt{7} - 456\sqrt{7}) = 640\sqrt{7}$, $S_6 = S_5 \cdot S_1 - S_4 \cdot S_2 = 640\sqrt{7} \cdot 2\sqrt{7} - 548 \cdot 24 = 640 \cdot 2 \cdot 7 - 548 \cdot 24 = 8960 - 13152 = -4192$. Thus, $x^6 + y^6 = -4192$. Since y^6 is positive, $x^6 = -4192 - y^6$. Therefore, x^6 is slightly less than -4192. But this contradicts since $x > 1$. Wait, actually, $y = \sqrt{7} - \sqrt{5} \approx 0.4142$, so $y^6 \approx (0.4142)^6 \approx 0.004$. Thus, $x^6 = -4192 - y^6 \approx -4192.004$, which is negative. This [Generation Termination without final answer]

✓ Response of Bingo with 470 words:

Okay, so I need to find the greatest integer less than $(\sqrt{7} + \sqrt{5})^6$. Hmm, that sounds like I need to compute $(\sqrt{7} + \sqrt{5})^6$ and then take the floor of that value. But since I can't use a calculator, I have to figure out a way to compute this without approximating. I remember that expressions like $(\sqrt{a} + \sqrt{b})^n$ can sometimes be simplified using binomial expansion or by recognizing patterns. Maybe I can pair $(\sqrt{7} + \sqrt{5})$ with $(\sqrt{7} - \sqrt{5})$ because their product is $7 - 5 = 2$, which is nice. Let me denote $x = \sqrt{7} + \sqrt{5}$ and $y = \sqrt{7} - \sqrt{5}$. Then, $x \cdot y = 2$. I also know that $x + y = 2\sqrt{7}$ and $x - y = 2\sqrt{5}$. Since x and y are conjugates, maybe I can find a recurrence relation for $x^n + y^n$. Let me compute $x^1 + y^1 = 2\sqrt{7} \approx 5.2915$, $x^2 + y^2 = (x + y)^2 - 2xy = (2\sqrt{7})^2 - 2 \times 2 = 28 - 4 = 24$, $x^3 + y^3 = (x + y)(x^2 + y^2 - xy) = 2\sqrt{7} \times (24 - 2) = 2\sqrt{7} \times 22 = 44\sqrt{7} \approx 44 \times 2.6458 \approx 116.315$, $x^4 + y^4 = (x^2 + y^2)^2 - 2(xy)^2 = 24^2 - 2 \times (4) = 576 - 8 = 568$, $x^5 + y^5 = (x + y)(x^4 + y^4) - xy(x^3 + y^3) = 2\sqrt{7} \times 568 - 2 \times (44\sqrt{7}) = 1136\sqrt{7} - 88\sqrt{7} = 1048\sqrt{7} \approx 1048 \times 2.6458 \approx 2767.318$, $x^6 + y^6 = (x^3 + y^3)^2 - 2(xy)^3 = (44\sqrt{7})^2 - 2 \times (8) = 1936 \times 7 - 16 = 13552 - 16 = 13536$. So $x^6 + y^6 = 13536$. But since $y = \sqrt{7} - \sqrt{5}$, which is less than 1 (since $\sqrt{7} \approx 2.6458$, $\sqrt{5} \approx 2.236$, $so y \approx 0.4098$). Therefore, y^6 is a very small positive number, less than 1. Therefore, $x^6 = 13536 - y^6$. Since y^6 is positive, $x^6 < 13536$. Thus, the integer part of x^6 is 13535. Therefore, the greatest integer less than $(\sqrt{7} + \sqrt{5})^6$ is 13535. **Final Answer**. 13535 </think> To find the greatest integer less than $(\sqrt{7} + \sqrt{5})^6$, we denote $x = \sqrt{7} + \sqrt{5}$ and $y = \sqrt{7} - \sqrt{5}$. We know that $x \cdot y = 2$, $x + y = 2\sqrt{7}$, and $x - y = 2\sqrt{5}$. We use the recurrence relation for $x^n + y^n$: $x^1 + y^1 = 2\sqrt{7}$, $x^2 + y^2 = 24$, $x^3 + y^3 = 44\sqrt{7}$, $x^4 + y^4 = 568$, $x^5 + y^5 = 1048\sqrt{7}$, $x^6 + y^6 = 13536$. Since $y = \sqrt{7} - \sqrt{5}$ is less than 1, y^6 is a very small positive number. Therefore, $x^6 = 13536 - y^6$, which is slightly less than 13536. Thus, the greatest integer less than $(\sqrt{7} + \sqrt{5})^6$ is 13535.

Final Answer: 13535.

Figure 11: Case study under the DeepSeek-R1-Distill-Qwen-1.5B model with three settings: Base, PPO, and Bingo. Blue highlights some redundant and repetitive tokens, while red marks omitted content and the final answer.

1620

1621

1622

1623

1624

1625

1626

1627

1628

1629

1630

1631

1632

1633

1634

1635

1636

1637

1638

1639

1640

1641

1642

1643

1644

1645

1646

1647

1648

1649

1650

1651

1652

1653

1654

1655

1656

1657

1658

1659

1660

1661

1662

1663

1664

1665

1666

1667

1668

1669

1670

1671

1672

1673

Problem: In how many ways can 8 people sit around a round table if 3 of the people – Pierre, Rosa, and Thomas – all want to sit together? (Two seatings are considered the same if one is a rotation of the other.)

✗ Response of Base Model with 5783 words:

Okay, so I have this function $f(x) = \sqrt{(x-1)/(x-2)}$, and I need to find the smallest possible integer value for x such that $f(x)$ is a real number. Hmm, let me think about how to approach this. First, I remember that for a square root to be real, the expression inside the square root, which is called the radicand, must be greater than or equal to zero. So, in this case, the radicand is $(x-1)$. That means $x-1$ has to be greater than or equal to zero. Let me write that down: $x-1 \geq 0$. Solving for x , I add 1 to both sides: $x \geq 1$. Okay, so x has to be at least 1. But wait, there's also the denominator in the function, which is $(x-2)$. For the function to be defined, the denominator can't be zero because division by zero is undefined. So, I need to make sure that $x-2$ is not equal to zero. Let me write that condition: ... But x can't be 2, so the next integer after 1 is 2, which is excluded. So, the next integer is 3. wait, is 3 the smallest integer that satisfies both conditions? Let me check. ... wait, if x is 1, then $f(x)$ is 0, which is real. So, 1 is a valid value. But is 1 the smallest integer? ... wait, but hold on, the problem says "the smallest possible integer value for x ." So, 1 is the smallest integer in the domain. So, 1 is the answer. But wait, hold on, I just thought of something. The function $f(x)$ is defined as $\sqrt{(x-1)/(x-2)}$. So, for $x=1$, it's 0, which is fine. But is 1 the smallest integer? Or is there a smaller integer? But integers go down to negative infinity, but x can't be less than 1 because of the square root. So, 1 is the smallest possible integer. ... So, 1 is the smallest integer in the domain, so that's the answer. But just to make sure, let me think about the function's domain. ... But wait, hold on, I just thought of something else. The problem is asking for the smallest possible integer value for x such that $f(x)$ has a real number value. So, 1 is the smallest integer, but is 1 the smallest integer in the domain? Yes, because x can't be less than 1. But wait, let me think about the function again. If x is 1, $f(x)$ is 0, which is real. So, 1 is acceptable. So, 1 is the answer. ... So, the integers in the domain are 1, 3, 4, 5, etc. So, the smallest integer is 1. Therefore, the [Generation Termination without final answer]

✓ Response of PPO with 437 words:

To determine the smallest possible integer value for x such that $f(x) = \frac{\sqrt{x-1}}{x-2}$ has a real number value, we need to consider the conditions under which the function is defined. 1. **Numerator Condition**: The expression inside the square root, $x-1$, must be non-negative because the square root of a negative number is not a real number. $x-1 \geq 0 \Rightarrow x \geq 1$. 2. **Denominator Condition**: The denominator, $x-2$, must not be zero because division by zero is undefined. $x-2 \neq 0 \Rightarrow x \neq 2$. 3. **Combined Domain**: Combining the above conditions, x must be greater than or equal to 1 but not equal to 2. $x \geq 1$ and $x \neq 2$. **Smallest Integer Value**: The smallest integer greater than or equal to 1 is 1. However, we need to check if $x=1$ satisfies the conditions. - For $x=1$: $f(1) = \frac{\sqrt{1-1}}{1-2} = \frac{0}{-1} = 0$. Since 0 is a real number, $x=1$ is valid. Therefore, the smallest possible integer value for x is 1. To determine the smallest possible integer value for x such that $f(x) = \frac{\sqrt{x-1}}{x-2}$ has a real number value, let's analyze the function step by step. 1. **Numerator Condition** - The expression inside the square root, $x-1$, must be non-negative because the square root of a negative number is not a real number. $x-1 \geq 0 \Rightarrow x \geq 1$. 2. **Denominator Condition** - The denominator, $x-2$, must not be zero because division by zero is undefined. $x-2 \neq 0 \Rightarrow x \neq 2$. 3. **Combined Domain** - Combining the above conditions, x must be greater than or equal to 1 but not equal to 2. $x \geq 1$ and $x \neq 2$. **Smallest Integer Value** - The smallest integer greater than or equal to 1 is 1. However, we need to check if $x=1$ satisfies the conditions. - For $x=1$: $f(1) = \frac{\sqrt{1-1}}{1-2} = \frac{0}{-1} = 0$. Since 0 is a real number, $x=1$ is valid. Therefore, the smallest possible integer value for x is: ¶

✓ Response of Bingo with 155 words:

First, I need to determine the domain of the function $f(x) = \frac{\sqrt{x-1}}{x-2}$ to find the smallest integer x for which $f(x)$ is a real number. I know that for the square root to be real, the expression inside must be non-negative. So, $x-1 \geq 0$, which implies $x \geq 1$. Additionally, the denominator $x-2$ cannot be zero, so $x \neq 2$. Combining these conditions, the domain is $x \geq 1$ with $x \neq 2$. The smallest integer greater than or equal to 1 is 1, but I need to check if $x=1$ is valid. Plugging in $x=1$, the denominator becomes -1 , which is allowed, and the square root is 0. So $f(1)=0$, which is a real number. Therefore, the smallest possible integer value for x is 1. </think> The function $f(x) = \frac{\sqrt{x-1}}{x-2}$ is defined when the expression inside the square root is non-negative and the denominator is not zero. 1. ****Square Root Condition**: $x-1 \geq 0$. $x \geq 1$. 2. ****Denominator Condition**: $x-2 \neq 0$. $x \neq 2$. Combining these conditions, the domain of $f(x)$ is $x \geq 1$ with $x \neq 2$. The smallest integer value for x that satisfies these conditions is $x=1$.

Final Answer: 1.

Figure 12: Case study under the DeepSeek-R1-Distill-Qwen-7B model with three settings: Base, PPO, and Bingo. Blue highlights some redundant and repetitive tokens, while red marks omitted content and the final answer.

1674 R HYPERPARAMETER STUDY
16751676 We evaluated several combinations of hyperparameters for BINGO on the GSM8K dataset using
1677 DeepSeek-R1-Distill-Qwen-1.5B as the base model. Table 7 reports the accuracy and output length
1678 across different settings.
16791680 Table 7: Performance of BINGO under different hyperparameter settings on GSM8K.
1681

λ_c	λ_w^{is}	λ_w^s	S	β	α	τ	Acc.	Len.
2	2	5	5	2	0.5	0.5	86.6	570
2	2	5	10	5	0.2	0.8	86.7	585
5	5	5	10	2.5	0.4	0.6	86.9	578
5	5	5	10	2.5	0.5	0.5	87.0	563

1688 **Hyperparameter Definitions:**
16891690 • λ_c : Insignificant Length Reward Weight for Correct Samples.
1691 • λ_w^{is} : Insignificant Length Reward Weight for Incorrect Samples.
1692 • λ_w^s : Significant Length Reward Weight for Incorrect Samples.
1693 • S : Slope interval for the Dynamic Length Reward.
1694 • β : Threshold for Training Phase Transition.
1695 • α : Decay Factor for Dynamic Length Reward.
1696 • τ : Threshold for Significant Tokens.
16971698 As shown in Table 7, the performance of BINGO remains stable, with both accuracy and output length
1699 exhibiting only minor fluctuations across the tested hyperparameter ranges. This indicates that the
1700 method is robust to hyperparameter choices. Since the last configuration achieves the best overall
1701 performance, we fixed these hyperparameters for methods and datasets to ensure consistency and
1702 fairness in comparison.
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727

S NOTATION TABLE

Table 8 offers a detailed overview of the notations utilized in this paper, along with their respective explanations. It serves as a handy reference to assist readers in grasping the concepts discussed in our work.

Table 8: Notation used throughout the paper

Notation	Description
<i>General</i>	
y	Sequence of tokens generated by the language model
x	Input prompt for the language model
n	Total length of the sequence y
y_i	i -th token in the generated sequence y
$\hat{z}(y)$	Extracted final answer from the generated sequence y
z	Ground-truth answer
\mathbb{E}_{π_θ}	Expectation over policy π_θ
$A(L)$	Expected accuracy as a function of output length L
L	Length of the output sequence generated by the model
L_{\max}	Maximum response length in the dataset
Acc	Exact match accuracy of the final output
L-Acc	Length-normalized accuracy, defined as $\text{Acc} \times \sqrt{1 - \frac{L}{L_{\max}}}$
$S(y_i)$	Significance score of token y_i
L^s	Number of significant tokens in the response
L^{is}	Number of insignificant tokens in the response
τ	Threshold for classifying a token as significant or insignificant
<i>Reinforcement Learning</i>	
π_θ	Policy parameterized by θ
\hat{A}_t	Advantage estimate at time step t
$r_t(\theta)$	Importance sampling ratio for policy optimization
R^{BINGO}	Reward function in the BINGO framework
$\mathcal{J}_{\text{BINGO}}(\theta)$	PPO objective with BINGO reward function
ϵ	Clipping parameter in the PPO objective
<i>Rewards and Penalties</i>	
$r_{is}(y)$	Reward for insignificant tokens in sequence y
$r_s(y)$	Reward for significant tokens in sequence y
λ_c	Coefficient for penalty on correct responses
λ_w	Coefficient for penalty on incorrect responses
k	Dynamic scaling factor for length reward
α	Scaling factor for the decay in dynamic length reward
β	Threshold for transition between exploration and compression in dynamic reward
<i>Length Penalty</i>	
L_{ref}^{is}	Reference number of insignificant tokens
L_{ref}^s	Reference number of significant tokens
$k(t)$	Dynamic scaling factor for adjusting length reward over time
<i>Miscellaneous</i>	
\mathcal{M}_e	Model used to estimate token significance (LLMLingua-2)
$\mathbb{1}[\cdot]$	Indicator function (1 if true, 0 otherwise)
\mathcal{Y}_{sig}	Set of significant tokens in the sequence y
$\mathcal{Y}_{\text{insig}}$	Set of insignificant tokens in the sequence y