Boosting LLM Reasoning via Human-Inspired Reward Shaping

Wenze Lin[1,*,†] **Zhen Yang**[2,*] **Xitai Jiang**[1,*] **Xiaoteng Ma**[3] **Gao Huang**[1,✉]

[1]Tsinghua University [2]Purdue University [3]MindLab

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a promising paradigm for enhancing reasoning in Large Language Models (LLMs). However, existing reward formulations typically treat exploration and consolidation as a monolithic process, resulting in entangled stage-wise learning dynamics. This contradicts the natural learning behavior of human learners. In human learning, individuals adopt distinct behavioral patterns toward mastered versus unfamiliar problems. When confronting unmastered challenges, humans prioritize broad exploration to seek viable solutions. By contrast, for well-mastered problems, they focus instead on reasoning condensation and knowledge abstraction to distill concise underlying principles. Motivated by this gap, we introduce **T2T** (Thickening-to-Thinning), a dynamic reward framework inspired by human learning processes. Specifically, it implements a dual-phase mechanism: (1) On incorrect attempts, T2T incentivizes "thickening" to broaden the search space and explore novel solution paths; (2) Upon achieving correctness, it shifts to "thinning", imposing length penalties to discourage redundancy, thereby fostering model confidence and crystallizing reasoning capabilities. Extensive experiments on mathematical benchmarks (MATH-500, AIME, AMC) across 5 mainstream LLMs demonstrate that T2T significantly outperforms standard GRPO and recent baselines, achieving superior performance.

1 Introduction

Psychological studies reveal that when faced with unfamiliar and unsolved problems, humans instinctively engage in diverse trials to seek viable solutions [23, 7, 1], a tendency particularly prominent in reasoning tasks [35]. For instance, when encountering difficult problems in mathematical examinations, students often attempt a variety of approaches and fill their answer sheets with trial reasoning even without a clear final solution. In contrast, as a problem gradually becomes familiar and mastered, humans naturally condense their problem-solving procedures to refine internal knowledge [27, 7, 1]. A typical example is the Pythagorean theorem: primary school students tend to re-derive the theorem every time they apply it shortly after learning it. As their knowledge accumulates and proficiency grows, however, they directly deploy the theorem without repeated derivation, since the knowledge has already been internally consolidated and refined. Taken together, human learning follows a clear two-stage pattern [7, 16, 1]. For unmastered challenges, learning starts with an expansive exploration phase: humans try multiple approaches, explore alternative decompositions, and tolerate verbosity and redundancy as a necessary cost of discovery. As mastery deepens, learning transitions into a second phase—reasoning is summarized and abstracted, and unnecessary details are stripped away to form compact, precise representations for efficient retention and reuse.

[*]Equal contribution. [†]Project lead. [✉]Corresponding author: gaohuang@tsinghua.edu.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

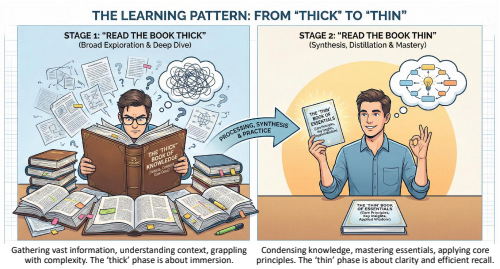


Figure 1: A two-stage learning pattern in human learning, where understanding is first expanded through exploration and later refined into concise and efficient knowledge.

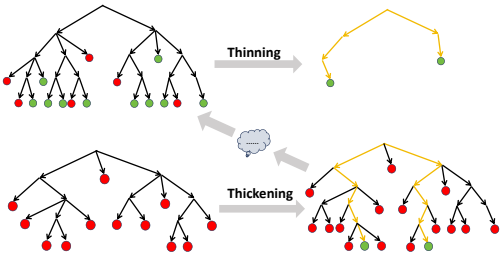


Figure 2: (1) On incorrect attempts, T2T incentivizes "thickening" to broaden the search space; (2) Upon correctness, it shifts to "thinning" to discourage redundancy, fostering model confidence.

Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a practical paradigm for enhancing the reasoning capabilities of Large Language Models (LLMs), particularly in automatically verifiable domains such as mathematics and code [8, 25, 13, 33, 38]. However, existing RLVR methods misalign with natural human learning dynamics we mentioned before, failing to make distinction between exploration and condensation. While some existing RLVR extensions have attempted to differentiate between positive and negative samples, their optimization objectives are either focused on compressing reasoning trajectories or maintaining entropy stability [45, 40, 19, 31]. The inherent learning dynamics between exploration for unmastered problems and condensation for mastered ones remain entangled without structural separation.

To address this gap, we propose Thickening-to-Thinning (T2T), a reward shaping method that explicitly embeds this stage-dependent learning dynamic into RLVR. Instead of redefining correctness, T2T modulates reward magnitude based on correctness, difficulty, and response length. It encourages expansive, exploratory reasoning when correctness is unachieved and favors concise, precise solutions once success is attained. What's more, T2T adopts a difficulty-aware incentive principle: for hard incorrect problems, T2T offers stronger encouragement for exploratory reasoning compared with easy failed cases; for simple solved problems, it imposes greater emphasis on concise reasoning than it does for correctly solved hard problems. Notably, T2T retains the simplicity of standard sequence-level RLVR: verification stays at the sequence level, no auxiliary models or token-level supervision are required. Consequently, T2T can be seamlessly integrated into existing RLVR pipelines and extended to various reinforcement learning variants, offering a minimal yet principled mechanism to align optimization objectives with the natural exploration-to-compression progression.

We conduct extensive experiments across five mainstream LLMs and four standard mathematical reasoning benchmarks. Empirical results demonstrate that T2T consistently achieves substantial gains over vanilla GRPO and a series of strong RLVR baselines. On the most capable model in our suite, Qwen3-14B, T2T improves AIME'25 Pass@64 from 51.5 to 57.9 ($+6.4$), while pushing AMC'23 Pass@64 from 94.6 to 98.4. Comparable improvements hold at smaller scales (e.g., Qwen3-4B AIME'24 Pass@64: $48.6 \rightarrow 56.3$) and across architectures (Llama-3.2-3B-Instruct AMC'23 Pass@64: $43.7 \rightarrow 49.1$), confirming T2T's strong cross-model transferability and broad applicability to diverse mathematical reasoning tasks.

2 Background and Related Works

Reinforcement Learning for Reasoning Reinforcement learning (RL) has become standard for aligning language models with complex logical constraints [21, 13, 33, 38]. Iterative training frameworks like STaR [42] and ReST [28] utilize the model's own successful trajectories as bootstrap supervision. More recent policy optimization methods, such as PPO [24] and Group Relative Policy Optimization (GRPO) [8, 25], directly optimize reasoning performance against outcome-based or process-based rewards [18]. However, standard RL objectives for reasoning typically treat the reasoning chain length as a byproduct or explicitly penalize verbosity to reduce costs

[6, 19, 29, 11, 12]. In contrast, our T2T approach treats trajectory length as instrument to realize human learning dynamic, rather than a goal itself.

Variants and Extensions of RLVR. A growing body of work has sought to address limitations of standard RLVR, including the mismatch between pass@1 improvements and large-k behavior, entropy collapse, and excessive verbosity. One line of work explicitly targets pass@k-style objectives or their surrogates [34, 2, 43, 4, 22, 14]. Another line focuses on theoretical or statistical corrections to the optimization process itself, for example via entropy control [40, 17, 15, 32, 5], structured pruning or brevity-promoting objectives [11, 19], or bias-aware alternatives to GRPO [20, 39]. These approaches are primarily motivated by formal objective mismatch or optimization pathologies, and propose principled modifications to the learning objective or update rule. In a different vein, NSR [45] demonstrates that training solely on negative samples can still yield substantial capability improvements, highlighting the robustness of reward-driven learning signals. Our work follows a different design philosophy. Rather than deriving corrections from theoretical optimality criteria, we adopt a heuristic perspective inspired by human learning behavior, and modify the reward objective accordingly. Since our proposed T2T method operates at the level of reward design, it is orthogonal to such framework-level extensions and can be applied alongside them without additional algorithmic overhead.

RLVR with Length Control. A series of work have incorporated length control mechanisms into RLVR [19, 29, 36, 9, 11, 12]. However, most prior works adopt explicit length penalties to compress inference overhead or reduce overthinking, aiming to maintain reasoning performance at lower computational cost. Their focus is all on efficient reasoning. In contrast, our method differs fundamentally from these approaches. For T2T, length modulation is not an optimization goal in itself, but an instrument to instantiate our human-inspired learning dynamics. Our primary objective is to boost the reasoning capability of LLMs by embedding the thickening-to-thinning learning paradigm into RLVR. We treat reasoning length as a dynamic exploration budget: it encourages the model to expand its reasoning search space on challenging unsolved problems, and drives concise, confident responses once the corresponding reasoning skill is fully mastered. In T2T, adaptive length modulation serves only as a supportive means to achieve stronger reasoning performance, rather than being pursued as an end goal.

3 Preliminaries

Reinforcement Learning with Verifiable Rewards

Let $\mathcal{V} : \mathcal{Q} \times \mathcal{O} \to \{0, 1\}$ denote a verifier that checks whether an output o solves query q:

$$\mathcal{V}(q, o) = \begin{cases} 1, & \text{if } o \text{ is verified as correct for } q, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Fix an interaction protocol (prompting, decoding strategy, and any hard constraints) and a model policy $\pi_\theta(o \mid q)$ over outputs $o \in \mathcal{O}$ given a query q. Define the single-sample success probability

$$p_\theta(q) \;:=\; \Pr_{o \sim \pi_\theta(\cdot|q)}[\,\mathcal{V}(q, o) = 1\,]. \tag{2}$$

For clarity, we consider sequence-level reinforcement learning with verifiable rewards and use it to illustrate the algorithms throughout this paper. Given a query $q \sim \mathcal{D}$ and a sequence-level output $o \sim \pi_\theta(\cdot|q)$, the learning objective is to maximize the expected reward:

$$\mathcal{J}(\pi_\theta) = \mathbb{E}_{q \sim \mathcal{D},\, o \sim \pi_\theta(\cdot|q)}\big[R(q, o)\big]. \tag{3}$$

In GRPO and related RLVR methods, the reward function $R(q, o)$ is typically defined as a binary indicator reflecting the correctness of the generated output. Under this definition, the objective reduces to maximizing the model's success probability under single-sample decoding:

$$\mathcal{J}(\pi_\theta) = \mathbb{E}_{q \sim \mathcal{D}}\Big[\mathbb{E}_{o \sim \pi_\theta(\cdot|q)}\big[\mathcal{V}(q, o)\big]\Big] = \mathbb{E}_{q \sim \mathcal{D}}\big[p_\theta(q)\big], \tag{4}$$

where $p_\theta(q) \triangleq \Pr_{o \sim \pi_\theta(\cdot|q)}\big(\mathcal{V}(q, o) = 1\big)$ denotes the probability that the model produces a verifier-accepted output for query q.

Using the policy gradient theorem, the gradient of the objective can be written as

$$\nabla_\theta \mathcal{J}(\pi_\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_\theta(\cdot|q)}\Big[A^{\pi_\theta}(q,o)\,\nabla_\theta \log \pi_\theta(o \mid q)\Big], \tag{5}$$

where the advantage function is defined as

$$A^{\pi_\theta}(q,o) = R(q,o) - \mathbb{E}_{o'\sim\pi_\theta(\cdot|q)}\big[R(q,o')\big]. \tag{6}$$

In practice, the expectation over the policy is approximated using N sampled outputs $\{o_j\}_{j=1}^{N}$, yielding the empirical estimator

$$\hat{A}^{\pi_\theta}(q,o_i) = R(q,o_i) - \frac{1}{N}\sum_{j=1}^{N} R(q,o_j), \quad o_j \sim \pi_\theta(\cdot|q). \tag{7}$$

This group-wise baseline corresponds to the advantage estimator used in GRPO.

To enable multiple optimization steps on the same batch of data, we introduce the importance sampling ratio between the current policy π_θ and the behavior policy $\pi_{\theta_{\text{old}}}$:

$$r_\theta(q,o) = \frac{\pi_\theta(o \mid q)}{\pi_{\theta_{\text{old}}}(o \mid q)}. \tag{8}$$

Consequently, the final surrogate objective with clipping—designed to constrain policy updates and ensure training stability—is formulated as:

$$\begin{aligned}&\mathcal{J}_{\text{CLIP}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\\ &\left[\min\Big(r_\theta(q,o)\hat{A}(q,o), \text{clip}\big(r_\theta(q,o), 1-\epsilon, 1+\epsilon\big)\hat{A}(q,o)\Big)\right],\end{aligned} \tag{9}$$

where ϵ is a hyperparameter determining the clipping range, and $\hat{A}(q,o)$ is the group-wise advantage estimator defined previously.

4 Method

4.1 Thickening-to-Thinning Reward Design

We introduce the *Thickening-to-Thinning (T2T)* reward, a competence-aware reward shaping scheme for RLVR. The design augments the standard 0/1 verifier score with two ingredients: an on-policy pass-rate and a length score.

On-policy pass-rate. Following GRPO, for a fixed query q we sample a group of K outputs $\{o_i\}_{i=1}^{K} \sim \pi_\theta(\cdot \mid q)$ and compute the on-policy pass-rate

$$\hat{p}_\theta(q) \;\triangleq\; \frac{1}{K}\sum_{i=1}^{K} \mathcal{V}(q,o_i). \tag{10}$$

Length score. Let $\ell(o)$ denote the length of o (e.g., number of tokens). We define a normalized length score $s_L(o) \in [0,1]$ via clipping:

$$s_L(o) \;\triangleq\; \text{clip}\left(\frac{\ell(o) - L_{\min}}{L_{\max} - L_{\min}},\, 0,\, 1\right). \tag{11}$$

Here $L_{\min}$ and $L_{\max}$ are hyper-parameters specifying the effective range for the length score. The length scores of outputs shorter than $L_{\min}$ or longer than $L_{\max}$ are clipped to 0 and 1, respectively, so that the shaping signal does not grow unboundedly at extreme lengths.

Thickening-to-Thinning (T2T) reward. We define

$$R_{\mathrm{T2T}}(q,o,\theta)=\begin{cases}1-\alpha\, s_L(o)\, p_\theta(q), & \mathcal{V}(q,o)=1,\\ \alpha\, s_L(o)\,(1-p_\theta(q)), & \mathcal{V}(q,o)=0.\end{cases}\qquad \alpha\in(0,\tfrac{1}{2}) \tag{12}$$

where α is a hyper-parameter controlling the strength of the T2T shaping. In implementation, $p_\theta(q)$ is replaced by its on-policy estimate $\hat{p}_\theta(q)$:

$$\hat{R}_{\mathrm{T2T}}(q,o,\theta)=\begin{cases}1-\alpha\, s_L(o)\, \hat{p}_\theta(q), & \mathcal{V}(q,o)=1,\\ \alpha\, s_L(o)\,(1-\hat{p}_\theta(q)), & \mathcal{V}(q,o)=0.\end{cases} \tag{13}$$

We compute advantages and perform policy optimization following the same procedure as GRPO. In implementation, $\hat{p}_\theta(q)$ is treated as a stop-gradient statistic: although $\hat{p}_\theta(q)$ is computed from on-policy rollouts and thus implicitly depends on θ, we detach it before computing the policy gradient, so that backpropagation only flows through $\log \pi_\theta(o \mid q)$. This avoids high-variance gradient terms from the group-level estimator and matches the standard practice in GRPO-style methods. Consequently, the resulting update signal does not directly correspond to the gradient of any fixed expected reward; we characterize the objective it actually optimizes in Section 4.2.

4.2 Analysis of T2T

Reward-induced ordering. For any fixed query q and $\alpha \in (0, \frac{1}{2})$, since $s_L, p_\theta(q) \in [0,1]$, we have

$$\inf_{\mathcal{V}=1} R_{\mathrm{T2T}} = 1-\alpha \;>\; \alpha \;\geq\; \sup_{\mathcal{V}=0} R_{\mathrm{T2T}},$$

so any correct output is rewarded above any incorrect one. Within each group, the shaping term breaks ties: among correct outputs, shorter is preferred ($-\alpha s_L p_\theta(q)$); among incorrect ones, longer is preferred ($+\alpha s_L(1-p_\theta(q))$). This yields the ordering

$$\text{correct short} \;\succ\; \text{correct long} \;\succ\; \text{incorrect long} \;\succ\; \text{incorrect short},$$

which prioritizes correctness, encourages longer attempts when the model struggles, and favors concise solutions once correctness is achieved. Unlike binary-reward GRPO, T2T provides non-trivial signals even when sampled outputs are all correct or all incorrect. Furthermore, the introduction of $p_\theta(q)$ ensures that for hard incorrect problems, T2T offers stronger encouragement for exploratory reasoning compared with easy incorrect cases; for simple solved problems, it imposes greater emphasis on concise reasoning than it does for correctly solved hard problems.

Gradient analysis. Recall that in implementation $\hat{p}_\theta(q)$ is detached from the computation graph. With standard policy-gradient computation, the per-query update signal is then

$$\frac{1}{K}\sum_{i=1}^{K} \hat{R}_{\mathrm{T2T}}(q,o_i,\theta)\, g_\theta(q,o_i), \tag{14}$$

which is an estimator of

$$\mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\left[R_{\mathrm{T2T}}(q,o,\theta)\, g_\theta(q,o)\right], \tag{15}$$

where $g_\theta(q,o) := \nabla_\theta \log \pi_\theta(o \mid q)$. Because R_{T2T} depends implicitly on θ through $p_\theta(q)$, Eq. (15) is *not* equal to $\nabla_\theta \mathbb{E}_{o\sim\pi_\theta}[R_{\mathrm{T2T}}]$, so the expression alone does not directly reveal the underlying objective. To clarify its effect, we provide a decomposition. Define the auxiliary reward

$$\tilde{R}(q,o,\theta) := \mathcal{V}(q,o) + \alpha\, s_L(o)\big(1-p_\theta(q)\big), \tag{16}$$

and its expected objective

$$\tilde{J}_q(\theta) \;:=\; \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}[\tilde{R}(q,o,\theta)] \;=\; p_\theta(q) + \alpha\, \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}[s_L(o)]\big(1-p_\theta(q)\big). \tag{17}$$

Notice that

$$R_{\mathrm{T2T}}(q,o,\theta) \;\triangleq\; \mathcal{V}(q,o)\big(1-\alpha\, s_L(o)\, p_\theta(q)\big) \;+\; \big(1-\mathcal{V}(q,o)\big)\,\alpha\, s_L(o)\big(1-p_\theta(q)\big), \tag{18}$$

thus, a direct calculation gives $R_{\mathrm{T2T}}(q,o,\theta) = \tilde{R}(q,o,\theta) - \alpha\, s_L(o)\, \mathcal{V}(q,o)$.

Theorem 4.1 (T2T update-signal decomposition)**.** *For any fixed query q,*

$$\mathbb{E}_{o\sim\pi_\theta(\cdot|q)}[R_{\mathrm{T2T}}\, g_\theta] \;=\; \nabla_\theta \tilde{J}_q(\theta) \;+\; \alpha\, \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}\big[\big(\mathbb{E}[s_L(o)] - s_L(o)\big)\, \mathcal{V}(q,o)\, g_\theta(q,o)\big]\,. \quad (19)$$

The proof is given in Appendix A.

This decomposition admits a clean interpretation. Viewed as a function of $p_\theta(q)$ with $\mathbb{E}[s_L(o)]$ held fixed, $\tilde{J}_q$ is monotone increasing in $p_\theta(q)$, so its optimum is still attained at $p_\theta(q) = 1$; correctness remains the dominant target. The shaping term $\alpha\,\mathbb{E}[s_L(o)]\big(1 - p_\theta(q)\big)$ matters most when $p_\theta(q)$ is small: in that regime, it supplies a non-trivial optimization signal through length, whereas vanilla binary-reward GRPO often provides almost no gradient when nearly all sampled outputs are incorrect. This is the **thickening** effect. The second term acts only on correct samples ($\mathcal{V} = 1$). Its coefficient $\mathbb{E}[s_L(o)] - s_L(o)$ is positive for shorter-than-average correct responses and negative for longer-than-average ones. Hence, once the model starts producing correct answers, T2T adds an explicit update signal that favors shorter correct trajectories. This is the **thinning** effect.

5 Experiment

5.1 Experimental Setup

Models. To investigate the scalability and effectiveness of our proposed method across different model capacities, we conduct experiments on the Qwen and DeepSeek series, which are widely recognized for their mathematical reasoning capabilities. Specifically, we utilize `DeepSeek-R1-Distill-Qwen-1.5B`, `Llama-3.2-3B-Instruct`, `Qwen2.5-3B`, `Qwen3-4B`, and `Qwen3-14B` as our base policies. For the Qwen3 series, we utilize the **non-thinking mode**. These models serve as the initialization for all reinforcement learning experiments.

Training Setup. We construct our training set using `DAPO-MATH-17k` [41], a high-quality dataset specifically filtered for mathematical reasoning. We train for a total of 5 epochs. All models are trained using the **Verl** framework [26]. The maximum new tokens is set to 8192 and the group size is set to 8. Detailed hyperparameter configurations are provided in Appendix B.1.

Baseline Settings. We compare **T2T** against standard GRPO and three advanced baselines that explicitly differentiate between positive and negative samples to refine the learning signal. Detailed implementation specifics and hyperparameter configurations are provided in Appendix B.2. The specific methods are as follows:

- **GRPO** [8, 25]: The vanilla Group Relative Policy Optimization algorithm without additional length or gradient modifications.
- **LASER** [19]: A reward shaping method that encourages efficient reasoning by augmenting the reward for positive samples with a length penalty term $\alpha \cdot \mathbb{I}(L < L_{\text{target}})$, while keeping rewards for negative samples unchanged.
- **W-REINFORCE** [45]: A gradient reweighting approach that retains the full gradient for negative samples to facilitate unlearning, while scaling down the gradient for positive samples by a factor ρ.
- **EntroPIC** [40]: An entropy-stabilization method that utilizes Proportional-Integral (PI) control to dynamically adjust the loss weights of positive and negative samples based on the discrepancy between the current policy entropy and a target entropy.

We note that due to computational constraints, for `Qwen3-14B`, only the standard GRPO baseline is implemented.

Evaluation Setup. We evaluate the models on four widely used and challenging mathematical reasoning benchmarks: **MATH-500** [10, 18], **AIME'24**, **AIME'25**, and **AMC'23**. During evaluation, we sample $n = 64$ responses per prompt for MATH-500, and $n = 128$ for AIME'24, AIME'25, and AMC'23. The sampling temperature is set to 0.7 with a top-p of 0.95.

We report the **Pass@k** metric ($k \in \{1, 2, 4, 8, 16, 32, 64\}$). Specifically, **Pass@1** is calculated as the mean accuracy over the n generated samples. For $k > 1$, to avoid high variance from limited

Table 1: **Main Results** on mathematical reasoning benchmarks. We report **Pass@1** and **Pass@64** performance using the unbiased estimator. The best results are highlighted in **bold**, and the second best results are underlined.

Base Model	Method	MATH-500		AIME'24		AIME'25		AMC'23	
		Pass@1	Pass@64	Pass@1	Pass@64	Pass@1	Pass@64	Pass@1	Pass@64
DeepSeek-R1-Distill-Qwen-1.5B	Initial Model	66.0	85.0	20.5	46.9	14.8	28.2	50.9	86.4
	GRPO	**68.9**	84.0	20.6	47.7	14.8	**28.9**	57.3	**92.7**
	LASER	66.5	85.4	20.8	48.3	14.9	28.4	57.8	91.5
	W-REINFORCE	67.5	85.8	20.4	46.8	14.7	28.0	58.2	91.6
	EntroPIC	66.2	84.8	21.0	49.5	15.0	28.5	58.5	91.4
	T2T (Ours)	66.9	**86.1**	**21.3**	**50.7**	**15.2**	28.8	**59.0**	91.7
Llama-3.2-3B-Instruct	Initial Model	24.9	32.5	12.6	13.3	0.0	0.0	15.4	22.5
	GRPO	34.3	45.0	14.7	20.9	3.2	5.3	29.2	43.7
	LASER	35.1	46.8	14.2	21.5	**3.8**	6.2	30.8	45.6
	W-REINFORCE	34.8	46.2	**15.7**	25.1	3.0	5.6	31.2	46.8
	EntroPIC	33.9	44.6	14.0	20.2	**3.8**	**7.0**	28.6	42.8
	T2T (Ours)	**36.2**	**47.6**	15.5	**25.4**	3.6	6.6	**32.0**	**49.1**
Qwen2.5-3B	Initial Model	40.6	49.1	7.1	14.7	3.4	4.6	43.9	59.0
	GRPO	47.4	59.9	4.8	14.7	**7.7**	15.6	49.1	63.4
	LASER	48.4	57.9	6.0	12.2	4.6	12.2	51.9	66.2
	W-REINFORCE	46.5	60.2	**12.6**	**22.8**	5.8	15.9	52.5	70.9
	EntroPIC	44.6	61.3	7.2	18.3	3.6	9.2	54.7	71.9
	T2T (Ours)	**49.8**	**64.6**	9.6	21.7	5.4	**16.7**	**57.0**	**77.0**
Qwen3-4B	Initial Model	80.2	**91.2**	22.2	37.0	14.7	35.4	71.4	92.4
	GRPO	**81.2**	90.5	29.3	48.6	21.3	48.5	75.8	91.2
	LASER	80.6	90.7	28.2	45.4	21.3	46.9	76.1	91.1
	W-REINFORCE	80.9	90.6	28.5	50.7	20.8	**50.9**	75.8	93.9
	EntroPIC	80.5	90.8	28.8	47.9	21.0	48.0	75.9	91.5
	T2T (Ours)	**81.2**	91.1	**33.4**	**56.3**	**23.0**	49.6	**78.0**	**95.3**
Qwen3-14B	Initial Model	82.2	87.2	27.1	48.3	19.6	47.3	74.6	91.2
	GRPO	83.3	88.8	29.1	56.1	22.4	51.5	80.7	94.6
	T2T (Ours)	**85.1**	**92.7**	**39.6**	**59.6**	**24.5**	**57.9**	**81.4**	**98.4**

sampling, we employ the unbiased estimator proposed by Chen [3]. Given n generated samples with c correct responses, Pass@k is calculated as:

$$\text{Pass@}k = \mathbb{E}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right]. \tag{20}$$

5.2 Main Results on Mathematical Benchmarks

The main results across four mathematical reasoning benchmarks are summarized in Table 1. The full experimental results are provided in Appendix H.1. Our experiments yield several key observations regarding the effectiveness and scalability of the proposed method:

Competitive Performance and Robust Improvements. As shown in Table 1, **T2T** demonstrates highly competitive performance across most tested model capacities. Notably, on the Qwen2.5-3B and Qwen3-4B models, our method yields consistent gains, particularly excelling in the AMC'23 and AIME benchmarks compared to the standard GRPO baseline and other variants like LASER and W-REINFORCE. On the largest evaluated model, Qwen3-14B, T2T exhibits dominant performance, securing the top spot in all Pass@1 and Pass@64 metrics, which validates the method's strong scalability.

Effectiveness of RL Training. Comparing the results with the *Initial Model*, it is evident that Reinforcement Learning significantly boosts mathematical reasoning capabilities. All RL-based methods show a substantial jump in performance compared to the base model. However, T2T generally maximizes this gain on models with sufficient capacity (3B+), suggesting that our approach utilizes the training data more efficiently by better distinguishing between high-quality and low-quality reasoning paths. Detailed analysis of T2T training dynamics are provided in Appendix D.

5.3 Controlled Analysis of Reasoning Trajectory Length

To isolate the effect of our T2T method on reasoning trajectory length, we conduct a controlled pairwise comparison between T2T and vanilla GRPO. Specifically, we restrict the analysis to samples

Table 2: Controlled analysis of reasoning trajectory lengths (token counts) on samples solved correctly by both GRPO and T2T.

Model	Method	MATH500	AIME24	AIME25	AMC23
DS-1.5B	GRPO	1453	3124	3510	2726
	T2T	1417	2786	3348	2814
Llama3.2-3B	GRPO	367	797	1483	651
	T2T	338	653	1237	584
Qwen2.5-3B	GRPO	1534	607	1521	1173
	T2T	1416	574	1392	1068
Qwen3-4B	GRPO	664	1718	1443	1732
	T2T	612	1543	1316	1577
Qwen3-14B	GRPO	687	1804	1496	1798
	T2T	534	1327	1384	1435

Table 3: **Ablation Study on Qwen2.5-3B.** We investigate the impact of Difficulty Awareness, Thickening (exploration reward), and Thinning (efficiency penalty). **Pass@1** and **Pass@64** are reported. The best results are highlighted in **bold**.

Method Variant	**MATH-500**		**AIME'24**		**AIME'25**		**AMC'23**	
	Pass@1	Pass@64	Pass@1	Pass@64	Pass@1	Pass@64	Pass@1	Pass@64
w/o Difficulty Awareness	49.5	64.4	8.9	20.5	4.9	15.8	56.2	75.8
w/o Thinning (Thickening Only)	49.1	63.3	8.5	20.8	4.6	16.0	55.1	75.2
w/o Thickening (Thinning Only)	48.3	60.1	8.2	18.5	4.2	13.9	55.8	72.5
T2T (Full)	**49.8**	**64.6**	**9.6**	**21.7**	**5.4**	**16.7**	**57.0**	**77.0**

that are solved correctly by both methods, eliminating confounding factors from differences in problem difficulty or correctness. We report the average reasoning length (in tokens) across five model families and four mathematical reasoning benchmarks, as shown in Table 2.

The results reveal a clear and consistent pattern: across nearly all model scales and benchmarks, T2T produces shorter average reasoning trajectories than GRPO on samples correctly solved by both methods. For instance, on the challenging AIME24 benchmark, T2T reduces the average reasoning length of Qwen3-14B from 1804 tokens to 1327 tokens, and that of Llama3.2-3B from 797 tokens to 653 tokens. This controlled comparison directly validates the thinning phase of our T2T framework: while standard GRPO assigns uniform rewards to all correct outputs and lacks incentives for conciseness, T2T dynamically encourages compact reasoning for mastered problems. The model accordingly prunes redundant exploratory steps and unnecessary derivations to converge on efficient solution paths, which aligns with the human learning progression from verbose exploration to condensed knowledge consolidation. More analysis of trajectory length are provided in Appendix F and Appendix E.

5.4 Ablation Study

To explicitly evaluate the contribution of each component in our proposed method, we conduct an ablation study using the `Qwen2.5-3B` model. We compare the full **T2T** against three variants:

- **w/o Difficulty Awareness:** We remove the group pass rate term from the reward formulation, applying a static length reward/penalty regardless of problem difficulty.

$$R_{\text{w/o Difficulty Awareness}}(q,o) = \begin{cases} 1 - \alpha\, s_L(o)\,, & \mathcal{V}(q,o) = 1, \\ \alpha\, s_L(o)\,, & \mathcal{V}(q,o) = 0. \end{cases} \tag{21}$$

- **w/o Thinning (Thickening Only):** We retain difficulty awareness but only apply the positive reward for incorrect responses (encouraging exploration on hard tasks). The penalty for correct responses is removed.

- **w/o Thickening (Thinning Only):** We retain difficulty awareness but only apply the length penalty for correct responses (encouraging efficiency on easy tasks). The reward for incorrect responses is removed.
- **T2T (Full):** The complete method incorporating difficulty-aware Thickening and Thinning.

The results are summarized in Table 3.

Impact of Difficulty Awareness. Comparing the full model with the *w/o Difficulty Awareness* variant, we observe a clear performance drop when the adaptive pass-rate scaling is removed. This confirms that treating all samples equally is suboptimal. A static length bias fails to distinguish between the need for exploration in hard problems and the need for conciseness in solved problems, leading to conflicting learning signals.

Impact of Thickening and Thinning Mechanics. As shown in the ablation results, removing either the "Thickening" or "Thinning" component leads to a general degradation in performance across most metrics, validating that both phases are essential for the full effectiveness of T2T. The *w/o Thinning* variant (Thickening Only) retains the incentive for expansive exploration on difficult or incorrect problems. However, its performance drops indicate that without the subsequent "Thinning" phase—which encourages abstraction and compression on correct solutions—the model fails to refine its verbose reasoning into efficient knowledge, mirroring the human learning process where understanding must be condensed after expansion. Conversely, the *w/o Thickening* variant (Thinning Only) applies penalties for redundancy but lacks the reward mechanism to encourage longer chains when the model fails. The significant decline in performance here suggests that skipping the "Thickening" phase prevents the model from engaging in the deep exploration necessary to solve hard problems, effectively trying to "thin" the knowledge before it has been sufficiently "thickened." These findings confirm our hypothesis based on Hua Luogeng's principle: mathematical reasoning capabilities are best developed by combining expansive exploration (Thickening) with disciplined abstraction (Thinning), rather than relying on either mechanism in isolation.

6 Discussion

Positioning T2T within the RLVR algorithmic stack. A useful way to situate T2T is to view RLVR algorithms as comprising three loosely coupled layers: the *reward* layer, which specifies $R(q, o)$; the *advantage* layer, which estimates $\hat{A}$ from sampled rewards (e.g., GRPO's group-relative baseline, Dr. GRPO's debiased estimator, or a learned value critic); and the *update* layer, which converts advantages into policy updates (e.g., PPO-style clipping, KL regularization, or W-REINFORCE-style reweighting). T2T modifies only the reward layer. The competence statistic $\hat{p}_\theta(q)$ is computed from on-policy rollouts and detached from the computation graph (Section 4), so the resulting reward enters the optimization pipeline as an ordinary scalar signal and does not interact with importance sampling, clipping, or KL terms. As a consequence, T2T is *orthogonal* to RLVR variants that target the advantage or update layers: it can in principle be combined with frameworks like Dr. GRPO [20], GSPO [44], DAPO [41], or off-policy extensions such as Luffy [37], without algorithmic conflict and without additional computational overhead.

Beyond RLVR: thickening-to-thinning as an implicit curriculum. More broadly, the thickening-to-thinning pattern can be viewed as an implicit, reward-driven curriculum: rather than scheduling task difficulty externally or annealing exploration coefficients by hand, the model's own competence on each query determines how much exploratory budget it receives. We see this as a small step toward training procedures whose exploration–exploitation trade-off adapts continuously to the learner's evolving state, and a direction worth investigating in settings beyond verifiable mathematical reasoning.

7 Conclusion

This work introduces Thickening-to-Thinning (T2T), a competence and length conditioned reward shaping strategy that explicitly distinguishes between exploratory reasoning and efficient solution refinement within RLVR. Importantly, T2T is lightweight to implement, requires no additional models

or inference-time overhead, and can be readily combined with algorithmic extensions that modify optimization or sampling procedures rather than reward semantics. Extensive experiments demonstrate that T2T not only improves final performance and generalization across diverse benchmarks, but also induces healthier training dynamics, mitigating common RLVR pathologies such as brittle exploration or excessive verbosity. More broadly, our results suggest that mimicking key aspects of human learning—progressing from expansive exploration to deliberate compression—may offer a fruitful direction for advancing reasoning in large language models beyond static reward formulations.

Acknowledgments

This work is supported in part by the National Key R&D Program of China under Grant 2024YFB4708200, the National Natural Science Foundation of China under Grants U24B20173, U2541227 and 62625604, and the Scientific Research Innovation Capability Support Project for Young Faculty under Grant ZYGXQNJSKYCXNLZCXM-I20.

References

[1] John R. Anderson. Cognitive psychology and its implications. 1980. URL https://api.semanticscholar.org/CorpusID:145694528.

[2] Farid Bagirov, Mikhail Arkhipov, Ksenia Sycheva, Evgeniy Glukhov, and Egor Bogomolov. The best of n worlds: Aligning reinforcement learning with best-of-n sampling via max@ k optimisation. *arXiv preprint arXiv:2510.23393*, 2025.

[3] Mark Chen. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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

[5] Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30377–30385, 2026.

[6] Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

[7] Paul M Fitts and Michael I Posner. Human performance. 1967.

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

[9] Hasan Abed Al Kader Hammoud, Kumail Alhamoud, Abed Hammoud, Elie Bou-Zeid, Marzyeh Ghassemi, and Bernard Ghanem. Train long, think short: Curriculum learning for efficient reasoning. *arXiv preprint arXiv:2508.08940*, 2025.

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

[11] Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

[12] Chengyu Huang, Zhengxin Zhang, and Claire Cardie. Hapo: Training language models to reason concisely via history-aware policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31122–31130, 2026.

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

[14] Yuhua Jiang, Jiawei Huang, Yufeng Yuan, Xin Mao, Yu Yue, Qianchuan Zhao, and Lin Yan. Risk-sensitive rl for alleviating exploration dilemmas in large language models. *arXiv preprint arXiv:2509.24261*, 2025.

[15] Yuxian Jiang, Yafu Li, Guanxu Chen, Dongrui Liu, Yu Cheng, and Jing Shao. Rethinking entropy regularization in large reasoning models. *arXiv preprint arXiv:2509.25133*, 2025.

[16] Jong W Kim, Frank E Ritter, and Richard J Koubek. An integrated theory for improved skill acquisition and retention in the three stages of learning. *Theoretical issues in ergonomics science*, 14(1):22–37, 2013.

[17] Qingbin Li, Rongkun Xue, Jie Wang, Ming Zhou, Zhi Li, Xiaofeng Ji, Yongqi Wang, Miao Liu, Zheming Yang, Minghui Qiu, et al. Cure: Critical-token-guided re-concatenation for entropy-collapse prevention. *arXiv preprint arXiv:2508.11016*, 2025.

[18] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

[19] Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612*, 2025.

[20] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

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

[22] Ruotian Peng, Yi Ren, Zhouliang Yu, Weiyang Liu, and Yandong Wen. Simko: Simple pass@ k policy optimization. *arXiv preprint arXiv:2510.14807*, 2025.

[23] Walter Schneider and Richard M Shiffrin. Controlled and automatic human information processing: I. detection, search, and attention. *Psychological review*, 84(1):1, 1977.

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

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

[26] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *EuroSys*, pages 1279–1297, 2025. URL https://doi.org/10.1145/3689031.3696075.

[27] Richard M Shiffrin and Walter Schneider. Controlled and automatic human information processing: Ii. perceptual learning, automatic attending and a general theory. *Psychological review*, 84(2):127, 1977.

[28] Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

[29] Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

[30] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, 2023.

[31] Hongze Tan and Jianfei Pan. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *ArXiv*, abs/2508.04349, 2025. URL https://api.semanticscholar.org/CorpusID:280536765.

[32] Hongze Tan, Zihan Wang, Jianfei Pan, Jinghao Lin, Hao Wang, Yifan Wu, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*, 2025.

[33] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

[34] Christian Walder and Deep Tejas Karkhanis. Pass@k policy optimization: Solving harder reinforcement learning problems. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=W6WC6047X2.

[35] David Wood, Jerome S Bruner, and Gail Ross. The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, 17(2):89–100, 1976.

[36] Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *arXiv preprint arXiv:2506.05256*, 2025.

[37] Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=vO8LLoNWWk.

[38] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[39] Fengkai Yang, Zherui Chen, Xiaohan Wang, Xiaodong Lu, Jiajun Chai, Guojun Yin, Wei Lin, Shuai Ma, Fuzhen Zhuang, Deqing Wang, et al. Your group-relative advantage is biased. *arXiv preprint arXiv:2601.08521*, 2026.

[40] Kai Yang, Xin Xu, Yangkun Chen, Weijie Liu, Jiafei Lyu, Zichuan Lin, Deheng Ye, and Saiyong Yang. Entropic: Towards stable long-term training of llms via entropy stabilization with proportional-integral control. *arXiv preprint arXiv:2511.15248*, 2025.

[41] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

[42] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

[43] Kaichen Zhang, Shenghao Gao, Yuzhong Hong, Haipeng Sun, Junwei Bao, Hongfei Jiang, Yang Song, Hong Dingqian, and Hui Xiong. Rspo: Risk-seeking policy optimization for pass@ k and max@ k metrics in large language models. *arXiv preprint arXiv:2508.01174*, 2025.

[44] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[45] Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=ftVlLG9cks.

A Proof of Theorem 4.1

For brevity, throughout this appendix we write

$$\mathbb{E}[\cdot] \;:=\; \mathbb{E}_{o\sim\pi_\theta(\cdot|q)}[\cdot], \qquad g_\theta(q,o) \;:=\; \nabla_\theta \log \pi_\theta(o \mid q).$$

Recall that in implementation $\hat{p}$ is detached from the computation graph, so the policy gradient flows only through $\log \pi_\theta(o \mid q)$. The quantity we analyze, $\mathbb{E}[R_{\mathrm{T2T}}\, g_\theta]$, is the population-level counterpart of this update.

Proof. The proof proceeds in three steps: (i) expand $\nabla_\theta \tilde{J}_q$ via the score-function identity, (ii) evaluate $\nabla_\theta p_\theta(q)$, and (iii) compare with $\mathbb{E}[R_{\mathrm{T2T}}\, g_\theta]$.

Step 1: Score-function expansion of $\nabla_\theta \tilde{J}_q$. Since $\tilde{R}(q, o, \theta)$ depends on θ both through the sampling distribution $\pi_\theta(\cdot \mid q)$ and explicitly through $p_\theta(q)$, differentiating $\tilde{J}_q(\theta) = \mathbb{E}[\tilde{R}]$ yields

$$\nabla_\theta \tilde{J}_q(\theta) \;=\; \underbrace{\mathbb{E}\Big[\nabla_\theta \tilde{R}\Big]}_{\text{explicit dependence}} \;+\; \underbrace{\mathbb{E}\Big[\tilde{R}\, g_\theta\Big]}_{\text{score-function term}} . \tag{22}$$

Among the components of $\tilde{R} = \mathcal{V}(q, o) + \alpha\, s_L(o)\,(1 - p_\theta(q))$, only $p_\theta(q)$ depends explicitly on θ, so

$$\nabla_\theta \tilde{R}(q, o, \theta) \;=\; -\alpha\, s_L(o)\, \nabla_\theta p_\theta(q). \tag{23}$$

Step 2: Evaluating $\nabla_\theta p_\theta(q)$. Applying the score-function identity to $p_\theta(q) = \mathbb{E}[\mathcal{V}(q, o)]$,

$$\nabla_\theta p_\theta(q) \;=\; \mathbb{E}[\mathcal{V}(q, o)\, g_\theta(q, o)]\,. \tag{24}$$

Substituting Eqs. (23)–(24) into Eq. (22) and using the fact that $\mathbb{E}[s_L(o)]$ is a constant with respect to o,

$$\begin{aligned} \nabla_\theta \tilde{J}_q(\theta) \;&=\; \mathbb{E}[\tilde{R}\, g_\theta] \;-\; \alpha\, \mathbb{E}[s_L(o)]\, \mathbb{E}[\mathcal{V}(q, o)\, g_\theta] \\ &=\; \mathbb{E}\Big[\big(\tilde{R} - \alpha\, \mathbb{E}[s_L(o)]\, \mathcal{V}(q, o)\big)\, g_\theta\Big]\,. \end{aligned} \tag{25}$$

Step 3: Comparing with $\mathbb{E}[R_{\mathrm{T2T}}\, g_\theta]$. By the identity $R_{\mathrm{T2T}} = \tilde{R} - \alpha\, s_L(o)\, \mathcal{V}(q, o)$ established in the main text,

$$\mathbb{E}[R_{\mathrm{T2T}}\, g_\theta] \;=\; \mathbb{E}\Big[\big(\tilde{R} - \alpha\, s_L(o)\, \mathcal{V}(q, o)\big)\, g_\theta\Big]\,. \tag{26}$$

Subtracting Eq. (25) from Eq. (26),

$$\mathbb{E}[R_{\mathrm{T2T}}\, g_\theta] \;-\; \nabla_\theta \tilde{J}_q(\theta) \;=\; \alpha\, \mathbb{E}\big[\big(\mathbb{E}[s_L(o)] - s_L(o)\big)\, \mathcal{V}(q, o)\, g_\theta\big]\,,$$

which rearranges to

$$\mathbb{E}[R_{\mathrm{T2T}}\, g_\theta] \;=\; \nabla_\theta \tilde{J}_q(\theta) \;+\; \alpha\, \mathbb{E}\big[\big(\mathbb{E}[s_L(o)] - s_L(o)\big)\, \mathcal{V}(q, o)\, g_\theta(q, o)\big]\,.$$

□

B Implementation Details

B.1 Hyperparameters

We provide the detailed hyperparameter configurations used in our experiments in Table 4. All models are trained using the **Verl** [26] framework with the settings specified below.

B.2 Baseline Implementation Details

To ensure a fair comparison and reproducibility, we strictly follow the hyperparameter recommendations provided in the respective original publications for all baseline methods. The specific configurations are detailed below:

GRPO. We utilize the standard implementation of Group Relative Policy Optimization [8] without any additional reward shaping or gradient modification terms.

LASER. Following Liu et al. [19], we configure the length-penalty reward shaping to encourage conciseness in correct responses. We set the penalty coefficient and the target length threshold as follows:

- **Penalty Coefficient (α):** 0.2
- **Target Length (L_{target}):** 4096

This target length aligns with the maximum response length used in our main experiments.

Table 4: Hyperparameter settings for T2T.

Hyperparameter	**Value**
Learning Rate	1e-6
Group Size	8
Train Batch Size	256
PPO Mini-Batch Size	64
Max Response Length	8192
Max Prompt Length	1024
Rollout Temperature	1.0
Evaluation Temperature	0.7
Evaluation Top-p	0.95
Clip Ratio High	0.2
Clip Ratio Low	0.2
Total Training Steps	275
T2T Reward Parameters	
Scaling Factor (α)	0.2
Length Norm Max ($L_{\max}$)	4096
Length Norm Min ($L_{\min}$)	0

W-REINFORCE. We adopt the **W-REINFORCE** baseline as recommended by Zhu et al. [45]. This method scales down the gradients of positive samples while retaining full gradients for negative samples to encourage exploration.

- **Positive Sample Weight (ρ):** 0.1

EntroPIC. For the entropy-stabilization method EntroPIC [40], we employ the Proportional-Integral (PI) control mechanism to dynamically adjust loss weights. The specific control parameters are set as:

- **Target Entropy (H_{target}):** 0.1
- **Proportional Gain (K_p):** 1.0
- **Integral Gain (K_i):** 0.01

These values correspond to the configuration for standard mathematical reasoning tasks as described in the original paper.

C Generalization Capabilities

A critical concern in domain-specific reinforcement learning is the potential risk of overfitting to the target domain (mathematics), which may degrade the model's performance on other tasks—a phenomenon often referred to as the "alignment tax." To evaluate the robustness and transferability of T2T, we conduct zero-shot evaluations on two out-of-domain benchmarks: **BBH** (Big Bench Hard) [30] for general logical reasoning and **HumanEval** [3] for code generation.

We evaluate the generalization capabilities of the *Initial Model*, standard GRPO, and other baseline methods (LASER, W-REINFORCE, EntroPIC) on Qwen2.5-3B and Qwen3-4B. The results are summarized in Table 5.

Generalization to Broader Reasoning Tasks. We observe that reinforcement learning demonstrates robust generalization capabilities beyond mathematical domains. As shown in Table 5, all evaluated RL methods maintain or slightly improve upon the Initial Model's performance on out-of-domain benchmarks, such as BBH (general reasoning) and HumanEval (coding). This suggests that the reasoning patterns acquired during mathematical training effectively transfer to broader logical and algorithmic tasks. T2T aligns with this trend, confirming that our method enhances domain-specific performance while preserving general reasoning abilities.

Table 5: **Generalization Performance on Out-of-Domain Tasks.** We report the **average 3-shot Exact Match (EM)** across all 27 subtasks of BBH (Reasoning) and **Pass@k** ($k \in \{1, 5, 10\}$) for HumanEval (Coding). **Detailed breakdowns of performance on each BBH subtask are provided in Appendix H.2.**

Base Model	Method	BBH (Reasoning)	HumanEval (Coding)		
		3-shot EM	Pass@1	Pass@5	Pass@10
Qwen2.5-3B	Initial Model	55.3	61.5	78.2	84.5
	GRPO	55.9	64.2	80.5	86.0
	LASER	56.0	64.5	80.8	86.4
	W-REINFORCE	55.9	64.1	80.4	86.1
	EntroPIC	55.7	63.8	80.0	85.8
	T2T (Ours)	55.8	64.2	80.5	86.2
Qwen3-4B	Initial Model	79.3	72.0	86.5	91.0
	GRPO	80.8	75.2	88.5	92.8
	LASER	80.7	75.5	88.8	93.0
	W-REINFORCE	80.7	75.1	88.4	92.6
	EntroPIC	80.0	74.8	88.0	92.2
	T2T (Ours)	80.6	75.7	89.0	93.5

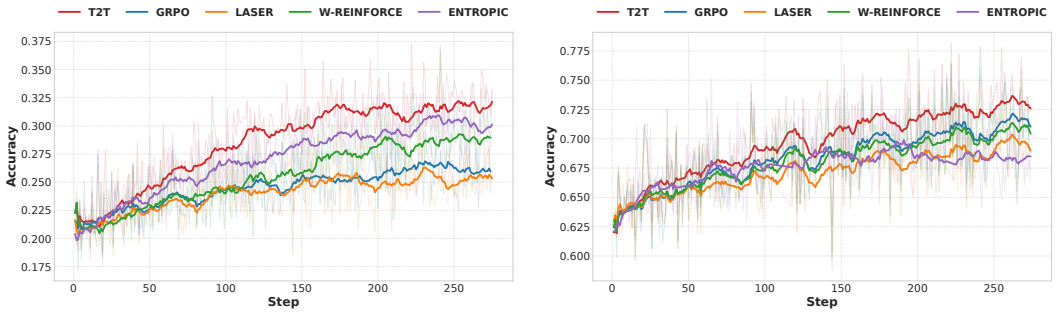


Figure 3: **Training Accuracy Evolution.** The plot on the left corresponds to `Qwen2.5-3B`, and the plot on the right corresponds to `Qwen3-4B`. Across both model scales, T2T demonstrates superior learning efficiency compared to the baseline.

D Analysis of Training Dynamics

To investigate the learning behavior of T2T, we visualize the evolution of Training Accuracy and Policy Entropy during the fine-tuning process. Figure 3 and figure 4 presents the comparative results on `Qwen2.5-3B` and `Qwen3-4B`.

Emergent High Entropy without Explicit Regularization. A critical observation from training dynamics is the behavior of policy entropy, which varies across model capacities yet consistently favors our method. It is important to emphasize that **T2T does not employ any explicit entropy regularization terms**.

On **Qwen2.5-3B** , the standard GRPO baseline exhibits a relatively flat and static entropy curve, suggesting limited variation in its search strategy. In contrast, T2T demonstrates a distinct dynamic: it exhibits a **steady increase in entropy** that **eventually converges** to a stable level. This trajectory aligns with our "Thickening-to-Thinning" philosophy: the model first expands its search space to explore diverse reasoning paths (Thickening), and then stabilizes as it learns to distinguish correct solutions (Thinning).

On **Qwen3-4B** , where the baseline shows a tendency towards entropy decay (signaling potential policy collapse), T2T effectively **counters this drop**, sustaining a significantly higher entropy level throughout training.

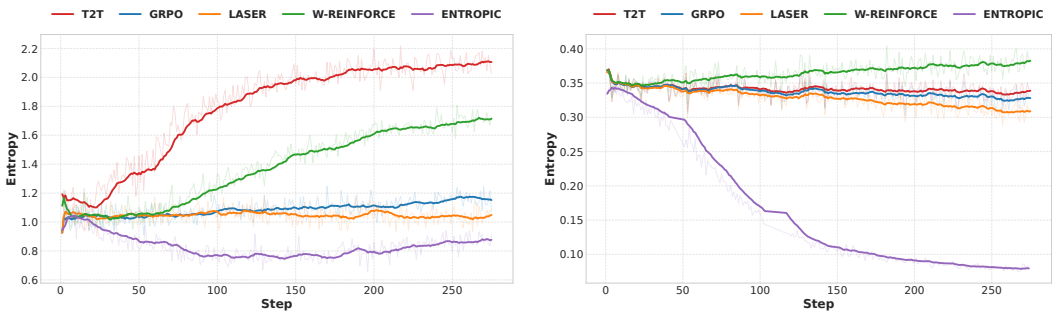


Figure 4: **Policy Entropy Evolution.** The plot on the left corresponds to `Qwen2.5-3B`, and the plot on the right corresponds to `Qwen3-4B`. Regardless of the absolute trend, T2T consistently maintains a higher relative entropy level than the baseline, indicating sustained exploration capabilities.

Despite these different underlying dynamics, the common thread is that T2T implicitly incentivizes the model to maintain a broader search space than the baseline. This emergent property directly supports the superior Pass@64 performance observed in our main results.

Accelerated Learning Efficiency. Figure 3 demonstrate that T2T achieves a significantly increase in training accuracy compared to the baseline during training. This rapid improvement stems from the **synergy of the Thickening-to-Thinning (T2T) strategy**. On one hand, the "Thickening" phase encourages the model to explore diverse trajectories for hard problems, increasing the likelihood of discovering correct reasoning chains that the baseline might miss. On the other hand, once a correct path is found, the "Thinning" phase applies a dynamic length penalty, filtering out verbose and irrelevant tokens. This dual mechanism ensures that the model not only finds the correct answers faster but also learns to refine them efficiently, leading to a consistently higher accuracy yield within the same training budget.

Length Modulation Mechanics. We further analyze the evolution of response lengths. We observe that T2T adaptively modulates generation length based on intrinsic **model capability**. Specifically, the method tends to increase length for less capable models (to encourage exploration of reasoning paths), while decreasing it for highly proficient models that have already mastered the tasks (to encourage conciseness). Moreover, a breakdown of positive and negative samples confirms that our method successfully learns a bi-modal strategy: shortening correct responses while elongating incorrect ones. **Detailed plots and analyses of these length dynamics are provided in Appendix E.**

E Detailed Analysis of Response Length in Training Dynamics

In this section, we provide a granular analysis of how T2T modulates the response length across different model scales and correctness outcomes.

E.1 Adaptive Length Modulation across Model Scales

Figure 5 illustrates the evolution of average response length during training. A striking divergence is observed between the two model scales, which strongly validates the adaptive nature of our T2T mechanism.

On **Qwen2.5-3B**, the response length for T2T increases relative to the initial stage. This suggests that for the smaller model, the training queries pose a higher level of intrinsic difficulty. Consequently, the "Thickening" incentive dominates the reward signal, encouraging the model to generate longer chains of thought to bridge reasoning gaps. This aligns with the entropy increase observed in Figure 4, confirming that the model is actively exploring via expansion. Conversely, on **Qwen3-4B**, the T2T response length exhibits a downward trend. Since the 4B model possesses stronger intrinsic reasoning capabilities, it achieves high success rates earlier in training. As the model gains confidence, the "Thinning" penalty takes precedence, incentivizing the pruning of redundant tokens to converge on more concise solutions. This contrast highlights a key advantage of T2T: it does not enforce a static

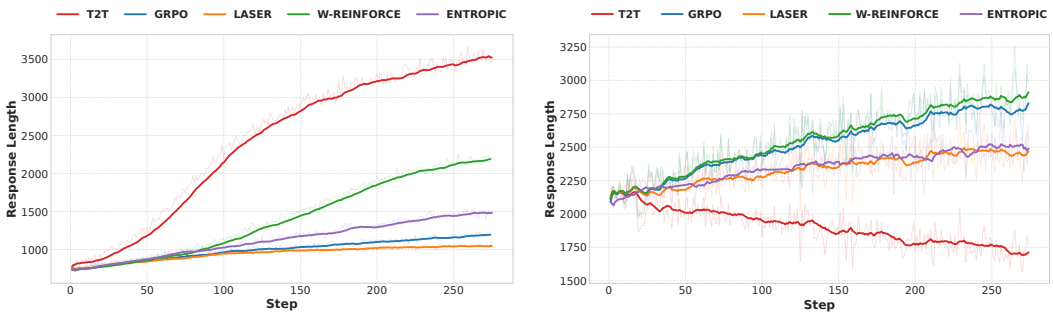
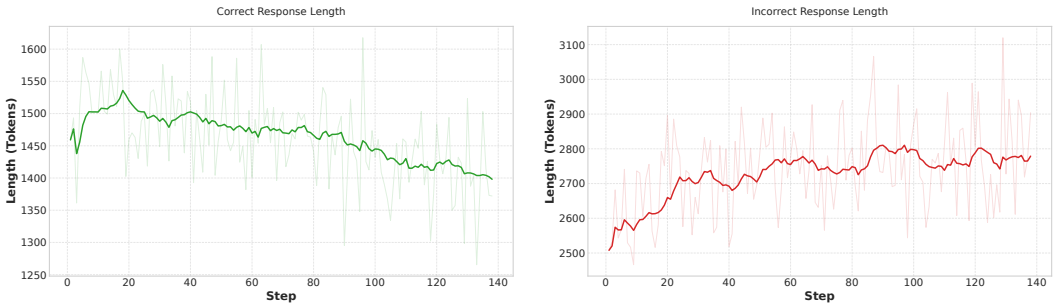


Figure 5: **Response Length Evolution.** The plot on the left corresponds to `Qwen2.5-3B`, and the plot on the right corresponds to `Qwen3-4B`.



(a) Correct Responses (Positive)

(b) Incorrect Responses (Negative)

Figure 6: **Divergent Length Dynamics based on Correctness (Qwen3-14B).** Analysis of token length evolution for (a) Correct and (b) Incorrect samples. Correct responses systematically shorten (Thinning) while incorrect responses lengthen (Thickening), confirming the bi-modal optimization strategy of T2T.

length bias (e.g., always longer or always shorter). Instead, it acts as a **competence-aware regulator**, dynamically allocating the computational budget based on the model's mastery of the task.

E.2 Disentangling Correct and Incorrect Response Dynamics

To rigorously verify that the observed length changes are causally linked to response correctness—as dictated by our T2T reward formulation—we decompose the response length evolution into positive (correct) and negative (incorrect) samples. Figure 6 presents this breakdown for the `Qwen3-14B` model.

As shown in Figure 6, the model exhibits a striking divergence in behavior depending on the outcome:

- **Thinning for Correctness:** The length of **positive samples** shows a consistent downward trend. This confirms that when the model successfully solves a problem, the adaptive length penalty effectively encourages the policy to converge towards the most efficient reasoning path.
- **Thickening for Failure:** Conversely, the length of **negative samples** increases over time. This indicates that when the model fails to reach the correct answer, the reward formulation incentivizes it to generate longer sequences,forcing the model to explore alternative reasoning steps.

Impact of Transitioning Samples. It is worth noting that the divergence shown in Figure 6 likely *underestimates* the true strength of the T2T mechanism due to the dynamic nature of training. As the model improves (see accuracy curves), many difficult problems that initially resulted in long, incorrect responses eventually transition into the "correct" category. Since these newly solved problems often require complex (and thus longer) reasoning chains compared to easy problems,

their migration into the positive pool partially offsets the length reduction of the originally correct samples. Similarly, their removal from the negative pool dampens the average length increase of the remaining incorrect samples. Therefore, the intrinsic drive of T2T to shorten correct paths and lengthen incorrect exploration is likely even stronger than what the aggregated statistics suggest.

This clear separation of dynamics provides strong empirical evidence that T2T creates a **bi-modal strategy**: efficient execution for known knowledge, and extensive exploration for unknown territories.

F Full Test Set Length Report

We report the average reasoning trajectory lengths across the full test set (including both correct and incorrect samples) for T2T and vanilla GRPO in Table 6.

Table 6: Average reasoning trajectory lengths (tokens) over the **full test set** for GRPO and T2T.

Model	Method	MATH500	AIME24	AIME25	AMC23
DS-1.5B	GRPO	3010	3674	4125	4289
	T2T	2842	3956	4412	4573
Llama3.2-3B	GRPO	1009	2709	2088	2247
	T2T	1146	2984	2357	2110
Qwen2.5-3B	GRPO	1764	2386	2917	3164
	T2T	1984	2612	3245	3418
Qwen3-4B	GRPO	1269	4824	4855	2251
	T2T	1124	4156	4287	2034
Qwen3-14B	GRPO	1384	4142	4370	2418
	T2T	1217	3784	4276	2156

As discussed, T2T's length control is not a static compression objective but a dynamic mechanism aligned with the model's mastery state: for problems the model struggles with, T2T encourages expansive exploration, leading to longer reasoning trajectories; for problems the model has mastered, it incentivizes concise reasoning, leading to shorter trajectories. Consequently, the average length over an entire dataset depends directly on the distribution of problem difficulty and the model's overall mastery of the test set at evaluation time.

T2T's length modulation is adaptive and context-dependent, rather than a one-size-fits-all compression strategy. We would like to clarify again that length control is not our major objective but an instrument to implement T2T dynamics. Our core objective is to fundamentally enhance the model's reasoning capabilities by simulating human learning dynamics.

G Hyperparameter Sensitivity Analysis

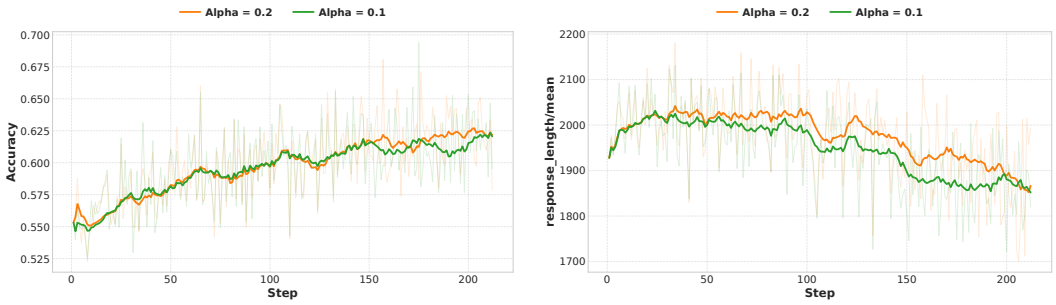
A practical reinforcement learning algorithm should be robust to hyperparameter choices and not require exhaustive tuning. To verify the sensitivity of T2T to the length scaling factor α, we conducted experiments on the `Qwen-14B` model comparing two distinct settings: $\alpha = 0.1$ and $\alpha = 0.2$.

The parameter α controls the magnitude of the length incentive relative to the correctness reward. As derived in Section 4, the theoretical constraint $\alpha < 0.5$ ensures that any correct response yields a higher reward than any incorrect response, regardless of length. Within this valid range, α merely adjusts the "strength" of the shaping signal rather than its direction.

The results are summarized in Table 7. We observe that the performance differences between $\alpha = 0.1$ and $\alpha = 0.2$ are marginal across benchmarks. For instance, the Pass@1 score on MATH-500 fluctuates by less than 1%, and the exploration capability (Pass@64) remains consistently high. This indicates that our method is **robust to the selection of** α: as long as the parameter is set within a reasonable range (sufficiently large to provide a gradient signal but satisfying $\alpha < 0.5$), the T2T mechanism effectively modulates the reasoning length without requiring precise tuning.

Table 7: **Sensitivity Analysis of α on Qwen3-14B.** We compare the performance with $\alpha = 0.1$ and $\alpha = 0.2$. The method demonstrates high stability, showing that precise tuning of the scaling factor is not critical for success.

Configuration	MATH-500		AIME'24		AIME'25		AMC'23	
	Pass@1	Pass@64	Pass@1	Pass@64	Pass@1	Pass@64	Pass@1	Pass@64
$\alpha = 0.1$	85.0	92.2	39.9	60.1	24.9	58.3	80.6	98.2
$\alpha = 0.2$	85.1	92.7	39.6	59.6	24.5	57.9	81.4	98.4



(a) Training Accuracy (b) Average Response Length

Figure 7: **Training Dynamics with different α.** Comparing $\alpha = 0.1$ and $\alpha = 0.2$ on Qwen-14B. (a) Both settings achieve similar convergence rates and final accuracy. (b) The response length dynamics show that a larger α induces a slightly stronger modulation in token generation, yet both stabilize effectively.

Analysis of Training Dynamics. To further understand how the scaling factor α influences the optimization process, we visualize the evolution of training accuracy and response length in Figure 7.

As illustrated in Figure 7, the model exhibits highly consistent behavior across different shaping magnitudes ($\alpha = 0.1$ and $\alpha = 0.2$). Both the learning trajectories (Figure 7a) and the response length dynamics (Figure 7b) follow nearly identical patterns. This stability confirms that the proposed T2T method is robust to the hyperparameter α, consistently guiding the model towards an efficient reasoning policy without requiring precise tuning.

H Full Experimental Results

H.1 Detailed Experimental Results on Mathematical Dataset

In this section, we provide the comprehensive performance metrics for all evaluated models across the four benchmarks. To thoroughly analyze the exploration capabilities of different methods, we report **Pass@**k for $k \in \{1, 2, 4, 8, 16, 32, 64\}$. Tables 8, 9, 10, and 11 present the results for MATH-500, AIME'24, AIME'25, and AMC'23, respectively.

Table 8: **Full Results on MATH-500**. Comparison of Pass@k ($k = 1, 2, 4, 8, 16, 32, 64$).

Base Model	Method	Pass@1	Pass@2	Pass@4	Pass@8	Pass@16	Pass@32	Pass@64
DeepSeek-R1-Distill-Qwen-1.5B	Initial Model	66.0	72.2	76.6	79.7	82.1	83.8	85.0
	GRPO	**68.9**	**73.8**	77.3	79.9	81.8	83.2	84.0
	LASER	66.5	72.6	77.0	80.2	82.5	84.1	85.4
	W-REINFORCE	67.5	73.5	77.5	80.5	82.8	84.5	85.8
	EntroPIC	66.2	72.4	76.8	79.5	81.9	83.5	84.8
	T2T	66.9	73.3	**77.8**	**81.0**	**83.4**	**85.0**	**86.1**
Llama-3.2-3B-Instruct	Initial Model	24.9	26.8	28.4	29.8	31.0	31.9	32.5
	GRPO	34.3	36.9	39.3	41.4	43.0	44.1	45.0
	T2T	**36.2**	**38.9**	**41.3**	**43.4**	**45.1**	**46.5**	**47.6**
Qwen2.5-3B	Initial Model	40.6	43.4	44.9	46.7	47.9	48.7	49.1
	GRPO	47.4	51.6	53.3	56.6	58.3	59.5	59.9
	LASER	48.4	50.9	53.2	55.2	56.5	57.3	57.9
	W-REINFORCE	46.5	51.1	53.1	56.7	58.6	59.9	60.2
	EntroPIC	44.6	49.0	52.7	55.8	58.1	60.0	61.3
	T2T	**49.8**	**53.1**	**56.3**	**58.9**	**61.1**	**63.0**	**64.6**
Qwen3-4B	Initial Model	80.2	83.3	85.7	87.6	89.1	90.3	**91.2**
	GRPO	**81.2**	84.1	86.3	88.0	89.1	89.9	90.5
	LASER	80.6	83.9	86.4	88.3	89.4	90.2	90.7
	W-REINFORCE	80.9	84.1	86.5	88.1	89.2	90.0	90.6
	EntroPIC	80.5	83.8	86.1	87.9	89.0	90.1	90.8
	T2T	**81.2**	**84.5**	**86.9**	**88.6**	**89.8**	**90.6**	91.1
Qwen3-14B	Initial Model	82.2	84.0	85.3	86.1	86.6	87.0	87.2
	GRPO	83.3	85.1	86.4	87.3	88.0	88.5	88.8
	T2T	**85.1**	**87.5**	**89.1**	**90.4**	**91.4**	**92.2**	**92.7**

Table 9: **Full Results on AIME'24**. Comparison of Pass@k ($k = 1, 2, 4, 8, 16, 32, 64$).

Base Model	Method	Pass@1	Pass@2	Pass@4	Pass@8	Pass@16	Pass@32	Pass@64
DeepSeek-R1-Distill-Qwen-1.5B	Initial Model	20.5	**25.2**	**30.0**	34.9	39.7	43.7	46.9
	GRPO	20.6	22.5	27.1	32.7	38.8	44.1	47.7
	LASER	20.8	22.9	27.6	33.2	39.4	44.8	48.3
	W-REINFORCE	20.4	22.1	26.8	32.0	38.1	43.2	46.8
	EntroPIC	21.0	23.5	28.5	34.5	40.8	45.9	49.5
	T2T	**21.3**	23.7	29.9	**36.2**	**42.1**	**47.3**	**50.7**
Llama-3.2-3B-Instruct	Initial Model	12.6	12.9	13.0	13.1	13.2	13.3	13.3
	GRPO	14.7	16.0	16.9	17.5	18.1	19.4	20.9
	T2T	**15.2**	**17.5**	**19.3**	**21.0**	**22.4**	**23.9**	**25.4**
Qwen2.5-3B	Initial Model	7.1	7.4	8.1	9.1	10.8	12.7	14.7
	GRPO	4.8	5.8	7.0	8.4	10.0	12.1	14.7
	LASER	6.0	6.9	7.7	8.7	9.9	11.2	12.2
	W-REINFORCE	**12.6**	**14.5**	**16.4**	**17.6**	**18.8**	**20.3**	**22.8**
	EntroPIC	7.2	7.9	8.9	10.4	12.9	15.9	18.3
	T2T	9.6	11.3	13.6	15.8	18.1	20.0	21.7
Qwen3-4B	Initial Model	22.2	24.2	26.1	28.2	31.2	34.2	37.0
	GRPO	29.3	32.2	35.7	39.6	44.7	47.0	48.6
	LASER	28.2	31.0	32.6	34.5	39.8	43.3	45.4
	W-REINFORCE	28.5	31.1	34.1	37.6	44.2	48.2	50.7
	EntroPIC	28.8	31.5	33.8	36.2	42.5	45.8	47.9
	T2T	**33.4**	**37.6**	**41.9**	**46.1**	**50.7**	**54.5**	**56.3**
Qwen3-14B	Initial Model	27.1	32.1	35.9	38.7	41.4	44.6	48.3
	GRPO	29.1	34.3	39.3	44.4	49.3	53.5	56.1
	T2T	**39.6**	**45.7**	**50.6**	**53.6**	**55.4**	**57.3**	**59.6**

Table 10: **Full Results on AIME'25**. Comparison of Pass@k ($k = 1, 2, 4, 8, 16, 32, 64$).

Base Model	Method	Pass@1	Pass@2	Pass@4	Pass@8	Pass@16	Pass@32	Pass@64
DeepSeek-R1-Distill-Qwen-1.5B	Initial Model	14.8	17.5	19.0	20.5	22.6	25.8	28.2
	GRPO	14.8	17.6	20.2	22.4	24.4	26.8	**28.9**
	LASER	14.9	17.8	19.5	21.2	23.1	26.1	28.4
	W-REINFORCE	14.7	17.4	19.1	20.8	22.9	25.5	28.0
	EntroPIC	15.0	17.9	20.0	21.9	23.5	26.4	28.5
	T2T	**15.2**	**18.4**	**21.4**	**23.8**	**25.7**	**27.5**	28.8
Llama-3.2-3B-Instruct	Initial Model	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	GRPO	3.2	3.8	4.2	4.9	5.8	6.4	6.6
	T2T	**3.6**	**3.8**	**4.3**	**5.0**	**5.9**	**6.6**	**6.6**
Qwen2.5-3B	Initial Model	3.4	3.4	3.5	3.6	3.8	4.1	4.6
	GRPO	**7.7**	**8.5**	**9.7**	11.1	12.5	14.2	15.6
	LASER	4.6	6.9	7.7	8.7	9.9	11.1	12.2
	W-REINFORCE	5.8	7.2	8.8	10.5	12.6	14.4	15.9
	EntroPIC	3.6	4.0	4.5	5.4	6.7	8.1	9.2
	T2T	5.4	7.1	9.3	**11.7**	**13.8**	**15.2**	**16.7**
Qwen3-4B	Initial Model	14.7	17.5	21.6	26.4	31.0	33.9	35.4
	GRPO	21.3	24.6	29.2	33.8	39.3	43.9	48.5
	LASER	21.3	23.3	27.0	32.0	37.7	43.7	46.9
	W-REINFORCE	20.8	24.3	27.7	35.4	40.2	45.2	**50.9**
	EntroPIC	21.0	24.0	28.5	33.5	39.0	44.5	48.0
	T2T	**23.0**	**26.7**	**31.4**	**36.1**	**41.2**	**46.3**	49.6
Qwen3-14B	Initial Model	19.6	22.3	26.3	31.7	37.7	43.2	47.3
	GRPO	22.4	24.5	28.0	33.3	39.2	45.7	51.5
	T2T	**24.5**	**29.1**	**34.7**	**41.1**	**48.1**	**54.2**	**57.9**

Table 11: **Full Results on AMC'23**. Comparison of Pass@k ($k = 1, 2, 4, 8, 16, 32, 64$).

Base Model	Method	Pass@1	Pass@2	Pass@4	Pass@8	Pass@16	Pass@32	Pass@64
DeepSeek-R1-Distill-Qwen-1.5B	Initial Model	50.9	59.7	67.7	74.6	79.7	83.7	86.4
	GRPO	57.3	65.3	73.5	80.6	86.1	**89.9**	**92.7**
	LASER	57.8	65.8	73.9	80.8	85.8	89.2	91.5
	W-REINFORCE	58.2	66.1	74.0	81.0	86.0	89.4	91.6
	EntroPIC	58.5	66.4	74.2	81.1	85.9	89.3	91.4
	T2T	**59.0**	**66.9**	**74.5**	**81.3**	**86.2**	89.5	91.7
Llama-3.2-3B-Instruct	Initial Model	15.4	17.3	19.3	20.9	22.2	22.4	22.5
	GRPO	29.2	32.4	35.2	37.4	39.5	41.8	43.7
	T2T	**32.0**	**36.0**	**39.8**	**43.7**	**46.6**	**48.3**	**49.1**
Qwen2.5-3B	Initial Model	43.9	47.7	50.8	53.4	55.8	57.8	59.0
	GRPO	49.1	52.9	55.8	57.5	59.2	61.3	63.4
	LASER	51.9	54.5	57.1	60.0	62.4	64.6	66.2
	NSR	52.5	56.6	60.4	63.7	66.5	68.8	70.9
	EntroPIC	54.7	59.0	62.7	65.9	68.7	70.6	71.9
	T2T	**57.0**	**61.6**	**65.6**	**69.8**	**73.2**	**75.8**	**77.7**
Qwen3-4B	Initial Model	71.4	76.3	80.0	83.3	85.7	90.6	92.4
	GRPO	75.8	79.1	82.6	87.3	88.9	89.7	91.2
	LASER	76.1	78.1	80.1	86.4	88.8	90.2	91.1
	W-REINFORCE	75.8	78.7	81.2	88.3	89.1	92.6	93.9
	EntroPIC	75.9	78.5	81.5	87.0	88.5	90.0	91.5
	T2T	**78.0**	**81.7**	**85.1**	**90.0**	**92.0**	**93.5**	**95.3**
Qwen3-14B	Initial Model	74.6	80.1	84.6	87.5	89.0	90.0	91.2
	GRPO	80.7	85.1	87.4	89.4	91.6	93.5	94.6
	T2T	**81.4**	**86.7**	**90.0**	**92.2**	**94.4**	**96.7**	**98.4**

H.2 Detailed BBH Experimental Results

In this section, we present the comprehensive breakdown of the **Big Bench Hard (BBH)** benchmark results in Table 12 and Table 13. We report the exact scores for all 27 subtasks across `Qwen2.5-3B` and `Qwen3-4B`.

Table 12: **Full BBH Results on Qwen2.5-3B.**

Subtask	Initial Model	GRPO	LASER	W-REINFORCE	EntroPIC	T2T (Ours)
Boolean Expressions	78.00	82.00	82.40	**82.80**	76.80	82.40
Causal Judgement	59.89	60.96	60.96	60.43	58.82	**61.50**
Date Understanding	59.60	**60.40**	60.00	59.20	58.40	58.80
Disambiguation QA	58.40	58.40	58.40	58.40	57.60	58.40
Dyck Languages	4.00	3.60	3.60	4.00	4.00	4.00
Formal Fallacies	54.40	55.20	55.60	52.80	**56.40**	52.00
Geometric Shapes	32.80	**33.20**	**33.20**	32.80	31.60	32.00
Hyperbaton	80.40	80.40	80.00	80.40	**81.20**	80.80
Logical Deduction (Five)	40.40	**41.60**	40.40	40.80	40.80	41.20
Logical Deduction (Seven)	31.20	30.00	30.40	**31.20**	30.00	30.40
Logical Deduction (Three)	67.20	67.20	67.20	67.20	**67.60**	67.20
Movie Recommendation	66.80	66.40	66.40	66.40	**67.20**	66.40
Multi-Step Arithmetic	76.00	77.60	78.00	76.80	**78.40**	77.60
Navigate	84.40	84.40	84.40	84.00	**84.80**	84.00
Object Counting	78.40	78.80	78.80	**79.20**	76.80	**79.20**
Penguins in a Table	63.70	**65.07**	64.38	**65.07**	63.70	**65.07**
Reasoning about Colored Objects	72.80	71.60	**74.00**	72.80	73.20	72.40
Ruins Names	48.00	47.60	47.20	47.20	**49.60**	47.20
Salient Translation Error	30.80	40.00	40.00	40.40	35.60	**40.80**
Snarks	60.11	60.11	60.11	59.55	**61.24**	58.99
Sports Understanding	73.60	73.60	73.60	73.20	**74.40**	74.00
Temporal Sequences	**32.40**	30.40	30.80	30.80	31.60	30.80
Tracking Shuffled Objects (Five)	36.80	38.00	**38.80**	37.60	38.40	37.60
Tracking Shuffled Objects (Seven)	27.60	26.80	27.20	**29.60**	28.00	28.00
Tracking Shuffled Objects (Three)	60.80	60.80	61.20	61.20	**61.60**	61.20
Web of Lies	99.60	99.60	99.60	99.60	99.60	99.60
Word Sorting	14.40	14.80	**15.20**	14.40	**15.20**	14.80
Average Score	55.28	55.87	**55.99**	55.85	55.65	55.79

Table 13: **Full BBH Results on Qwen3-4B.**

Subtask	Initial Model	GRPO	LASER	W-REINFORCE	EntroPIC	T2T (Ours)
Boolean Expressions	95.70	**97.60**	**97.60**	96.80	97.20	96.00
Causal Judgement	59.70	58.82	59.89	55.08	**61.50**	56.68
Date Understanding	81.60	83.60	83.20	83.20	**84.00**	83.60
Disambiguation QA	58.50	60.80	60.40	62.00	59.20	**62.40**
Dyck Languages	38.50	**39.60**	38.80	39.20	36.80	**39.60**
Formal Fallacies	75.00	73.60	72.80	74.00	**78.40**	72.80
Geometric Shapes	51.50	**54.40**	52.80	53.20	46.80	51.20
Hyperbaton	91.50	90.80	**91.60**	90.80	91.20	**91.60**
Logical Deduction (Five)	67.50	**71.60**	68.80	68.40	70.40	69.60
Logical Deduction (Seven)	**37.80**	27.60	30.80	28.40	31.60	27.20
Logical Deduction (Three)	95.50	96.00	96.00	96.00	**96.80**	96.40
Movie Recommendation	67.50	68.80	69.20	69.20	**69.60**	68.80
Multi-Step Arithmetic	98.20	98.40	98.80	98.80	**99.60**	98.40
Navigate	90.50	92.80	92.80	92.80	**93.60**	92.80
Object Counting	84.50	**86.00**	**86.00**	**86.00**	84.80	85.60
Penguins in a Table	90.80	94.52	93.15	94.52	91.78	**95.21**
Reasoning about Colored Objects	94.80	96.00	96.00	96.40	**96.80**	96.40
Ruins Names	68.00	70.00	**72.40**	68.00	69.60	66.80
Salient Translation Error	60.80	62.80	61.60	62.00	**63.60**	62.40
Snarks	69.50	**69.66**	66.85	68.54	67.98	**69.66**
Sports Understanding	81.80	84.40	84.40	84.40	83.60	**84.80**
Temporal Sequences	84.00	86.00	86.00	**86.40**	**86.40**	86.00
Tracking Shuffled Objects (Five)	95.50	96.40	**97.60**	**97.60**	**97.60**	97.20
Tracking Shuffled Objects (Seven)	89.00	92.80	92.80	**93.20**	89.60	**93.20**
Tracking Shuffled Objects (Three)	85.50	91.60	91.20	**94.40**	75.60	92.40
Web of Lies	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Word Sorting	35.80	37.20	36.40	38.00	37.60	**39.60**
Average Score	79.33	**80.82**	80.73	80.70	80.05	80.57

I Case Study

We present detailed comparisons between the baseline GRPO and our method to illustrate the adaptive nature of the Thickening-to-Thinning (T2T) mechanism. The following visualizations display two contrasting scenarios: (1) a **Thickening Case** on a hard trigonometric problem, where our method is incentivized to explore a rigorous derivation path—solving a system of linear equations—to correct the baseline's failure; and (2) a **Thinning Case** on a simple arithmetic problem, where our method effectively prunes the baseline's redundant conversational fillers to achieve inference efficiency without compromising accuracy.

Case 1: Thickening Case

Prompt

There exist constants a, b, c, and d such that

$$(\sin x)^7 = a\sin 7x + b\sin 5x + c\sin 3x + d\sin x$$

for all angles x. Find d. Let's think step by step and output the final answer in boxed format.

GRPO

We are given the identity:

$$(\sin x)^7 = a\sin 7x + b\sin 5x + c\sin 3x + d\sin x$$

for all angles x, and we are to find the constant d.

Step 1: Use the identity for $\sin^7 x$
We can use the **multiple-angle identity** or **power-reduction formulas** to express $\sin^7 x$ in terms of sines of multiple angles. Alternatively, we can use the **complex exponential form** of sine:

$$\sin x = \frac{e^{ix} - e^{-ix}}{2i}.$$

So:

$$\sin^7 x = \left(\frac{e^{ix} - e^{-ix}}{2i}\right)^7.$$

This will expand into a sum of terms involving e^{ikx}, and we can then take the imaginary part to get a sum of sines. But this might be tedious. Instead, we can use a known identity or the **orthogonality of sine functions**.

Step 2: Use the identity for $\sin^7 x$
We can use the identity:

$$\sin^7 x = \frac{7}{8}\sin x - \frac{7}{8}\sin 3x + \frac{7}{8}\sin 5x - \frac{1}{8}\sin 7x.$$

This is a known identity derived from the **multiple-angle expansions** of sine powers. Let us verify this identity by comparing coefficients.

Step 3: Compare with the given expression
We are given:

$$(\sin x)^7 = a\sin 7x + b\sin 5x + c\sin 3x + d\sin x.$$

From the identity above:

$$\sin^7 x = -\frac{1}{8}\sin 7x + \frac{7}{8}\sin 5x - \frac{7}{8}\sin 3x + \frac{7}{8}\sin x.$$

So matching terms:

- $a = -\frac{1}{8}$
- $b = \frac{7}{8}$

- $c = -\frac{7}{8}$
- $d = \frac{7}{8}$

Final Answer:

$$\boxed{\frac{7}{8}}$$

T2T

We are given the identity:

$$(\sin x)^7 = a\sin 7x + b\sin 5x + c\sin 3x + d\sin x$$

for all angles x, and we are to find the constant d.

Step 1: Use the identity for $\sin^7 x$
We can use the **multiple-angle identities** or **power-reduction formulas** to express $\sin^7 x$ in terms of sines of multiple angles. Alternatively, we can use the **complex exponential form** of sine:

$$\sin x = \frac{e^{ix} - e^{-ix}}{2i}.$$

So:

$$\sin^7 x = \left(\frac{e^{ix} - e^{-ix}}{2i}\right)^7.$$

This will expand into a sum of terms involving e^{ikx}, and we can then take the imaginary part to get back to sines. But this might be tedious. Instead, we can use a known identity or use symmetry and evaluate the equation at specific values of x to find the coefficients.

Step 2: Use symmetry and evaluate at specific values of x
Let us consider the original equation:

$$(\sin x)^7 = a\sin 7x + b\sin 5x + c\sin 3x + d\sin x.$$

We can plug in specific values of x to get equations involving a, b, c, d.

Case 1: $x = 0$

- $\sin 0 = 0$, so LHS $= 0^7 = 0$
- RHS $= a\sin 0 + b\sin 0 + c\sin 0 + d\sin 0 = 0$

So this gives no new information.

Case 2: $x = \frac{\pi}{2}$

- $\sin\frac{\pi}{2} = 1$, so LHS $= 1^7 = 1$

The RHS is:

$$a\sin\frac{7\pi}{2} + b\sin\frac{5\pi}{2} + c\sin\frac{3\pi}{2} + d\sin\frac{\pi}{2}.$$

We compute each term:

- $\sin\frac{7\pi}{2} = \sin\left(3\pi + \frac{\pi}{2}\right) = -1$
- $\sin\frac{5\pi}{2} = \sin\left(2\pi + \frac{\pi}{2}\right) = 1$
- $\sin\frac{3\pi}{2} = -1$
- $\sin\frac{\pi}{2} = 1$

So:

$$\text{RHS} = -a + b - c + d.$$

Setting RHS equal to LHS gives:

$$-a + b - c + d = 1 \tag{1}$$

Case 3: $x = \frac{\pi}{6}$

- $\sin\frac{\pi}{6} = \frac{1}{2}$, so LHS $= \left(\frac{1}{2}\right)^7 = \frac{1}{128}$

The RHS is:

$$a\sin\frac{7\pi}{6} + b\sin\frac{5\pi}{6} + c\sin\frac{3\pi}{6} + d\sin\frac{\pi}{6}.$$

Compute each sine:

- $\sin\frac{7\pi}{6} = \sin\left(\pi + \frac{\pi}{6}\right) = -\frac{1}{2}$
- $\sin\frac{5\pi}{6} = \sin\left(\pi - \frac{\pi}{6}\right) = \frac{1}{2}$
- $\sin\frac{3\pi}{6} = \sin\frac{\pi}{2} = 1$
- $\sin\frac{\pi}{6} = \frac{1}{2}$

So:

$$\text{RHS} = -\frac{a}{2} + \frac{b}{2} + c + \frac{d}{2}.$$

Setting equal to LHS:

$$-\frac{a}{2} + \frac{b}{2} + c + \frac{d}{2} = \frac{1}{128} \tag{2}$$

Case 4: $x = \frac{\pi}{3}$

- $\sin\frac{\pi}{3} = \frac{\sqrt{3}}{2}$
- LHS $= \left(\frac{\sqrt{3}}{2}\right)^7 = \frac{27\sqrt{3}}{128}$

The RHS is:

$$a\sin\frac{7\pi}{3} + b\sin\frac{5\pi}{3} + c\sin\pi + d\sin\frac{\pi}{3}.$$

Compute each sine:

- $\sin\frac{7\pi}{3} = \sin\left(2\pi + \frac{\pi}{3}\right) = \frac{\sqrt{3}}{2}$
- $\sin\frac{5\pi}{3} = -\frac{\sqrt{3}}{2}$
- $\sin\pi = 0$
- $\sin\frac{\pi}{3} = \frac{\sqrt{3}}{2}$

So:

$$\text{RHS} = \frac{\sqrt{3}}{2}(a - b + d).$$

Setting equal to LHS:

$$\frac{\sqrt{3}}{2}(a - b + d) = \frac{27\sqrt{3}}{128}$$

$$\Rightarrow a - b + d = \frac{27}{64} \tag{3}$$

Now we have three equations:

- (1) $-a + b - c + d = 1$
- (2) $-\frac{a}{2} + \frac{b}{2} + c + \frac{d}{2} = \frac{1}{128}$
- (3) $a - b + d = \frac{27}{64}$

We are to find d.
From (3):

$$a = b - d + \frac{27}{64} \tag{4}$$

Substitute into (1):

$$2d - c = \frac{91}{64} \tag{5}$$

Substitute into (2):

$$d + c = \frac{7}{32} \tag{6}$$

Adding (5) and (6):

$$3d = \frac{105}{64} \Rightarrow d = \frac{35}{64}.$$

Final Answer:

$$\boxed{\frac{35}{64}}$$

Case 2: Thinning Case

Prompt

How many positive whole-number divisors does 196 have? Let's think step by step and output the final answer in boxed format.

GRPO

We are asked to find how many **positive whole-number divisors** the number **196** has.

Step 1: Prime Factorization
We start by factoring 196 into its prime factors.

$$196 = 2 \times 98 = 2 \times 2 \times 49 = 2^2 \times 7^2$$

So, the **prime factorization** of 196 is:

$$196 = 2^2 \times 7^2$$

Step 2: Use the Divisor Count Formula
If a number n has the prime factorization:

$$n = p_1^{e_1} \times p_2^{e_2} \times \cdots \times p_k^{e_k},$$

then the number of **positive divisors** of n is:

$$(e_1 + 1)(e_2 + 1) \ldots (e_k + 1).$$

For 196, the exponents are:

- $2^2 \rightarrow$ exponent = 2
- $7^2 \rightarrow$ exponent = 2

So the number of positive divisors is:

$$(2 + 1)(2 + 1) = 3 \times 3 = 9.$$

Final Answer:

$$\boxed{9}$$

So, **196 has 9 positive whole-number divisors**.

T2T

To find the number of **positive whole-number divisors** of 196, we start by finding its **prime factorization**.

Step 1: Prime Factorization of 196

$$196 = 2 \times 98 = 2 \times 2 \times 49 = 2^2 \times 7^2$$

So, the prime factorization is:

$$196 = 2^2 \times 7^2$$

Step 2: Use the Formula for Number of Divisors
If a number

$$n = p_1^{e_1} \times p_2^{e_2} \times \cdots \times p_k^{e_k},$$

then the number of positive divisors is:

$$(e_1 + 1)(e_2 + 1) \ldots (e_k + 1).$$

For 196:

- Exponent of 2 is 2
- Exponent of 7 is 2

So, the number of divisors is:

$$(2 + 1)(2 + 1) = 3 \times 3 = 9$$

Final Answer:

$$\boxed{9}$$

J Limitations

This work has several limitations that leave room for future improvement.

First, we estimate problem difficulty solely based on the **on-policy pass rate** during training. Such a single-point online estimation is susceptible to sampling randomness and training fluctuations, which may lead to inaccurate difficulty calibration across benchmarks. To mitigate this issue, future work can adopt statistical smoothing strategies such as Exponential Moving Average (EMA) to stabilize pass-rate statistics, filter transient noise, and yield more reliable difficulty scores for adaptive T2T reward modulation.

Second, our current implementation and experiments are constrained to **binary-verifiable RLVR settings**. Nevertheless, the Thickening-to-Thinning (T2T) principle originates from general human cognitive learning dynamics and is not inherently bounded to rigid verifiable tasks. Extending T2T to open-domain reasoning remains a promising direction. In non-verifiable scenarios such as open-domain QA and commonsense reasoning, explicit binary verification is no longer available; task difficulty and model mastery can instead be proxied by soft reward signals, including reward model scores or self-consistency entropy across multiple reasoning paths. The core T2T mechanism still holds: we encourage expansive, thickened reasoning for low-confidence or low-reward queries, and promote concise, thinned reasoning for high-confidence, well-mastered ones. Beyond open-domain reasoning, T2T is also compatible with modern on-policy and self-distillation paradigms. The T2T-adjusted reward signal can be incorporated into reverse KL divergence distillation objectives, guiding student models to learn dynamically optimized reasoning trajectories adapted to implicit task difficulty. We leave these broader extensions to future research.

K Hardware Setup

All experiments are conducted on a Linux-based computing cluster, with GPU configurations tailored to different model scales:

1. **1.5B-scale models**: Trained on NVIDIA RTX3090 GPUs.
2. **3B–4B-scale models**: Trained on NVIDIA A100-PCIE-40GB GPUs.
3. **14B-scale models**: Trained on NVIDIA A800-PCIE-80GB GPUs.