

Demystifying Entropy Conflict: Why Reasoning Compression Stalls in Large Reasoning Models

Hourun Zhu¹ Yang Gao^{1*} Wenlong Fei¹ Jiawei Li¹ Huashan Sun¹
Bowen Ren¹ Chengzhi Hu²

¹School of Computer Science and Technology,
Beijing Institute of Technology, Beijing, China

²Taiyuan University of Technology, Taiyuan, China

{hrzhu, gyang, wenlongfei, jwli, hssun, bwren-bit}@bit.edu.cn
2024511300@link.tyut.edu.cn

Abstract

Large reasoning models achieve strong problem-solving performance, but often generate excessively long chain-of-thought outputs, which substantially increase inference cost and limit practical deployment. Existing reasoning compression methods reduce length only partially and often suffer from unstable optimization or accuracy degradation. We trace this limitation to an underlying **trajectory-level objective conflict**: when the compression objective and the accuracy objective are optimized jointly, compression favors short-and-correct trajectories while accuracy optimization continues to preserve long-but-correct ones that remain useful on hard problems. This conflict creates opposing gradients on the same behaviors and can trap training in a local trade-off where further compression harms accuracy while accuracy recovery reintroduces verbose reasoning. At the model-decision level, we refer to the resulting failure mode as **entropy conflict**: the policy fails to fully collapse onto concise trajectories because accuracy-oriented updates keep restoring broader support over long-correct reasoning paths. To address this issue, we propose an entropy-guided training framework. Instead of forcing a mixed objective to follow an unstable path toward the final efficient-output set, we first contract the policy to a short-output set and then expand from this compact regime to recover accuracy and robustness. Experiments on multiple mathematical reasoning benchmarks show that our method achieves performance comparable to the original model using only about 25% of its reasoning tokens.

1 Introduction

Large reasoning models (LRMs) achieve strong performance by generating chain-of-thought (CoT) reasoning traces, but this strength often comes with unnecessary verbosity (DeepSeek-AI,

Q John writes 20 pages a day. How long will it take him to write 3 books that are 400 pages each?
A Okay, let me see the question. I need to find... Wait...many tokens...The answer is 60. (4581 tokens)



Figure 1: Trajectory-level objective conflict and its entropy manifestation in reasoning compression. The compression objective concentrates probability mass on short-and-correct trajectories, reducing the entropy of the reasoning distribution. Accuracy-oriented optimization preserves a broader set of correct trajectories, including long-but-correct ones, which re-expands the effective support of the policy and increases entropy.

2025; OpenAI, 2024a,b; Huang et al., 2024; Singhal et al., 2024; Saito et al., 2023). Long reasoning traces substantially increase inference cost and limit practical deployment, making reasoning compression an important problem for scalable LRMs. Recent RLVR-based methods attempt to shorten reasoning while preserving problem-solving ability (Xiang et al., 2025; Ling et al., 2025; Song and Zheng, 2025; Li et al., 2025a). However, they still face two challenges: maintaining the balance between brevity and accuracy, and keeping compression training stable.

We trace these challenges to an underlying **trajectory-level objective conflict** between the compression objective and the accuracy objective when they are optimized jointly. Compression pushes the model toward short-and-correct trajectories, whereas the accuracy objective continues to reward long-but-correct trajectories that remain useful on hard problems. These incompatible preferences create opposing gradients on the same behav-

*Corresponding author.

iors, so optimization can be trapped in a local trade-off where further compression harms accuracy while accuracy recovery reintroduces verbose reasoning. Once training enters this stagnant regime, the conflict becomes observable at the model-decision level as **entropy conflict**: the policy can no longer reliably collapse onto concise trajectories, because accuracy-oriented updates keep restoring broader support over long-correct reasoning paths.

We further analyze this phenomenon at the token level: high-entropy tokens often receive larger gradients, and many of them are reasoning connectors such as *therefore*, *then*, and *similarly*. Reinforcing these tokens encourages additional intermediate steps and increases decision entropy, making further compression even harder.

To resolve this conflict, we propose an entropy-guided training strategy that decomposes optimization according to the evolution of policy entropy. We first apply compression-focused training to suppress redundant continuations and drive the policy toward a concise, low-entropy response mode. Once this entropy contraction stabilizes, we switch to an accuracy-oriented stage that restores necessary exploration and task capability within the already compact policy support, gradually recovering the model’s decision entropy.

Across multiple mathematical reasoning benchmarks, our method achieves performance comparable to the original model using only about 25% of its reasoning tokens. We organize our experiments around five questions aligned with the proposed explanation. First, we verify whether entangling compression and accuracy objectives causes training stagnation. Second, we analyze why the accuracy objective tends to favor longer responses, especially through high-entropy reasoning tokens. Third, we test whether resolving this conflict enables a better compression–accuracy trade-off than existing baselines. Fourth, we examine the reasoning pattern that emerges after compression, showing that the model learns a more compact planning-and-execution behavior rather than merely truncating outputs. Finally, we conduct ablations to identify the critical roles of stage ordering, length clipping, and reward shaping.

2 Related Work

2.1 Large Reasoning Models and Verbose Reasoning

Chain-of-thought prompting and search-based reasoning methods (Wei et al., 2022; Yao et al., 2023; Besta et al., 2024) improve LLMs on complex multi-step tasks. Recent large reasoning models further strengthen this ability through reinforcement learning and large-scale post-training (OpenAI, 2024b; DeepSeek-AI, 2025; Yang et al., 2025; Team, 2025; OpenAI, 2024a), but often produce long intermediate reasoning traces. Prior studies also report verbosity bias in model preferences and generated responses (Saito et al., 2023; Huang et al., 2024; Singhal et al., 2024), suggesting that longer outputs are not always more informative. Our work studies why reducing this redundancy becomes difficult once accuracy-oriented training is involved.

2.2 Reasoning Compression

Reasoning-compression methods are commonly data-driven or reward-driven. Data-driven methods use supervised fine-tuning or constructed datasets to teach shorter reasoning behaviors, such as distilling compact rationales (Yu et al., 2024), introducing supervised reasoning modes (Fang et al., 2025), learning stopping decisions (Chen et al., 2025), or training on adaptive CoT data (Ma et al., 2025). These methods can control length, but often rely on curated supervision and may overfit to specific reasoning styles.

Reward-driven methods modify the RL objective to penalize unnecessary length or adapt reasoning depth. Some works use explicit length penalties or pruning objectives (Shen et al., 2025; Aggarwal and Welleck, 2025; Yeo et al., 2025; Luo et al., 2025a; Arora and Zanette, 2025), while others combine length control with accuracy rewards (Liu et al., 2025a; Dai et al., 2025; Fatemi et al., 2025; Xiang et al., 2025; Ling et al., 2025; Song and Zheng, 2025; Li et al., 2025a). Although these methods improve efficiency, jointly optimizing accuracy and compression can still be unstable because the two rewards may prefer different correct trajectories. Our focus is not to claim that entropy-based analysis itself is new, but to identify a more specific trajectory-level objective conflict that explains why reasoning-compression training stalls.

2.3 Entropy in RL-Based Reasoning Optimization

Prior entropy-based studies (Wang et al., 2025; Hao et al., 2025; Deng et al., 2025a; Li et al., 2025b) mainly use entropy as a diagnostic or control variable for stabilizing RL-based reasoning optimization. Our work addresses a different question. We ask why reasoning-compression training stalls even when both compression and accuracy rewards appear well defined. We trace this failure to a trajectory-level objective conflict between compression and accuracy optimization. In this problem, entropy conflict is the decision-level manifestation of the underlying support mismatch between objectives, directly indicating that the model fails to converge to the optimal trajectory subset. This distinction motivates our entropy-guided training strategy: rather than applying an entropy intervention to stabilize a single objective, we decouple the conflicting objectives so that the model first enters a compact reasoning regime and then improves accuracy within that regime.

3 Conflict in Reasoning Compression

3.1 Formalizing Trajectory-Level Conflict

To formalize why direct reasoning compression tends to stall, we formulate it as a joint reinforcement learning problem that balances solution correctness and reasoning compactness. For an input x , let $\Omega(x)$ denote the set of complete reasoning trajectories and let $\pi_\theta(\tau \mid x)$ be the policy over $\tau \in \Omega(x)$. This objective corresponds to the objective-entangled (OE) training setting studied in our experiments. We consider

$$J_\lambda(\theta) = J_{\text{acc}}(\theta) - \lambda J_{\text{len}}(\theta), \quad (1)$$

where $\lambda > 0$ is the compression coefficient,

$$\begin{aligned} J_{\text{acc}}(\theta) &= \mathbb{E}_{x, \tau \sim \pi_\theta} [R_{\text{acc}}(\tau)], \\ J_{\text{len}}(\theta) &= \mathbb{E}_{x, \tau \sim \pi_\theta} [L(\tau)]. \end{aligned} \quad (2)$$

Here $R_{\text{acc}}(\tau) \in [0, 1]$ denotes the accuracy reward and $L(\tau) \geq 0$ denotes the trajectory length.

Lemma 1 (Conflict Existence). *Suppose that for some input x , there exist two correct trajectories $\tau_s, \tau_l \in \Omega(x)$ such that $L(\tau_s) < L(\tau_l)$. Then under the joint objective in Equation 1, the short-correct trajectory is strictly preferred to the long-correct one.*

Since both trajectories receive the same accuracy reward, their joint-score difference depends only on the output length:

$$\begin{aligned} & (R_{\text{acc}}(\tau_s) - \lambda L(\tau_s)) - (R_{\text{acc}}(\tau_l) - \lambda L(\tau_l)) \\ &= \lambda (L(\tau_l) - L(\tau_s)) > 0. \end{aligned} \quad (3)$$

Thus, whenever the same problem admits both short-correct and long-correct solutions, the compression term strictly suppresses the latter, even though both trajectories are equally valid from the perspective of correctness. Importantly, this preference becomes an optimization conflict on hard inputs where long-correct trajectories remain useful for maintaining or improving accuracy under the current policy: accuracy-oriented updates reinforce these trajectories, while compression-oriented updates suppress the same behaviors.

Theorem 1 (Conflict-Induced Stagnation). *If improving accuracy on hard inputs requires moving the policy toward long-correct trajectories along a parameter-space direction that also increases expected reasoning length, then there exists a compression coefficient $\lambda^* > 0$ such that the directional derivative of the joint objective vanishes along that direction. Consequently, joint optimization can exhibit a stationary trade-off direction: moving further toward compression reduces accuracy, whereas moving toward accuracy recovery increases reasoning length.*

Let v denote such a direction, satisfying

$$\langle \nabla_\theta J_{\text{acc}}(\theta), v \rangle > 0, \quad \langle \nabla_\theta J_{\text{len}}(\theta), v \rangle > 0. \quad (4)$$

The two signals cancel when

$$\begin{aligned} \lambda^* &= \frac{\langle \nabla_\theta J_{\text{acc}}(\theta), v \rangle}{\langle \nabla_\theta J_{\text{len}}(\theta), v \rangle} > 0, \\ \langle \nabla_\theta J_{\lambda^*}(\theta), v \rangle &= 0. \end{aligned} \quad (5)$$

Theorem 1 therefore lifts the conflict from individual trajectories to optimization dynamics. Once long-correct trajectories remain useful for preserving accuracy on difficult samples, the accuracy objective and the compression objective can cancel along the same update direction. As a result, direct joint optimization can stall after partial compression instead of converging to a shorter reasoning regime.

This stationary trade-off also explains the entropy-level behavior. Because accuracy-oriented updates continue to retain probability mass on long-correct trajectories, the policy support cannot fully

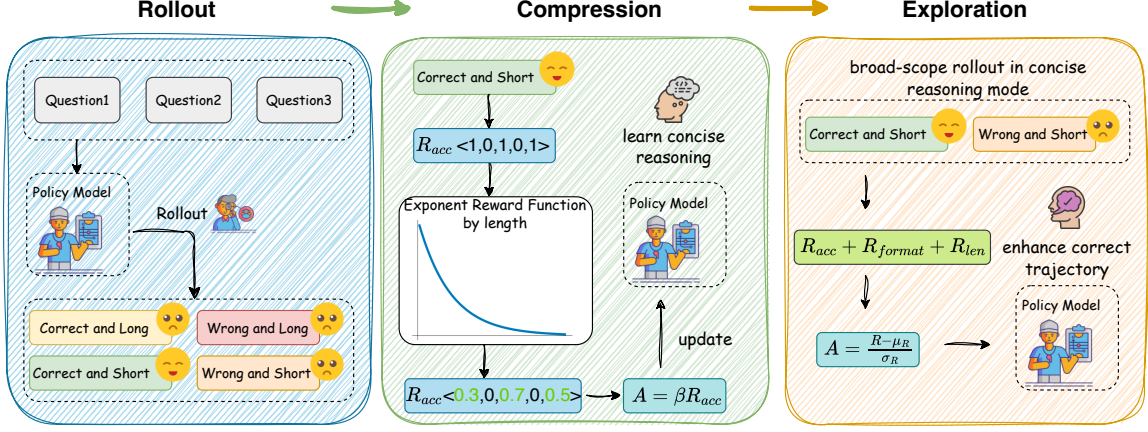


Figure 2: Overview of our method. The model first learns concise reasoning through compression. Then it enhances reasoning ability in the exploration phase via broader rollouts.

contract onto concise solutions. We refer to this decision-level failure to concentrate on concise trajectories as entropy conflict.

The full policy-gradient derivation and formal proofs of Lemma 1 and Theorem 1 are provided in Appendix A.

3.2 Broader Implications for Entropy-guided Optimization

Although our focus is reasoning-chain compression, objective conflict reflects a more general challenge in multi-objective optimization. When one objective effectively defines a stricter behavioral subset of another, jointly optimizing them can preserve gradients from the broader objective and prevent full convergence to the target subspace. In our setting, the concise-and-correct trajectories preferred by compression form such a stricter subset of all correct trajectories preserved by accuracy optimization. Appendix B further discusses this nested-objective interpretation and its implications for staged optimization.

4 Method

4.1 Motivation

Guided by the objective-conflict analysis in Section 3, we view reasoning compression as finding an efficient output set that is both correct and concise. Direct mixed-objective training approaches this set under conflicting accuracy and length signals, which can make optimization unstable or stagnant. We therefore first contract the policy toward a short-output set, then expand from this compact

regime to recover accuracy and robustness, yielding the final efficient-output set.

4.2 Compressing Stage

4.2.1 Length Clipping

To encourage the model to generate concise reasoning trajectories, we introduce a length-clipping mechanism during sampling. When a generated reasoning sequence exceeds a predefined maximum length L without producing a final answer, the trajectory is terminated and assigned a zero reward. Formally, the clipped reward is defined as:

$$R_{\text{clip}} = \begin{cases} 0, & \text{if } |y| > L, \\ R(y), & \text{otherwise.} \end{cases} \quad (6)$$

We determine L through a one-time calibration using only the training data. Specifically, the base policy first generates a fixed set of M rollouts, denoted by $\mathcal{R}_{\text{cal}} = \{y_j\}_{j=1}^M$, on the training set. For each candidate threshold L , we compute the clipped accuracy

$$\text{Acc}_{\text{clip}}(L) = \frac{1}{M} \sum_{j=1}^M \mathbf{1}[\text{Correct}(y_j)] \cdot \mathbf{1}[|y_j| \leq L]. \quad (7)$$

We then choose the smallest candidate threshold that retains approximately 30% clipped accuracy:

$$L^* = \min \{L : \text{Acc}_{\text{clip}}(L) \geq \tau\}, \quad \tau = 0.30. \quad (8)$$

This calibration yields $L^* = 4096$ in our experiments. Importantly, the rollouts and threshold selection use the training set only; no test examples or

test-set performance are used to tune L . The value $\tau = 0.30$ is an empirical feasibility threshold that preserves a non-trivial amount of correct training signal under clipping, rather than a guarantee that the selected L is globally optimal.

In reasoning tasks, response length is closely correlated with the number of intermediate reasoning steps, and excessively long outputs often reflect redundant deliberation. By pruning trajectories that exceed the length threshold, this mechanism filters out samples with low information density and biases sampling toward concise, answer-producing paths. As a result, the sampled group contains a higher proportion of valid and efficient reasoning trajectories, providing higher-quality supervision for subsequent policy updates.

4.2.2 Positive-Only and Absolute-Advantage Updates

Previous study (Deng et al., 2025b) has reported an interesting phenomenon: some negative samples are highly similar to positive ones. So their gradients may decrease the probability of positive samples. Therefore, to ensure training efficiency, we update the model using only the gradients from the positive samples of the target dataset. The training objective can be expressed as

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{y \sim \pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(y) \cdot A(y)]_{R_{clip}(y) > 0}. \quad (9)$$

The advantage function is computed as $A(y) = \beta R_{clip}(y)$, where β denotes the scaling coefficient. We adopt this absolute advantage formulation instead of the relative advantage used in standard GRPO. This design choice helps to prevent instability introduced by normalization. When relative normalization is applied, certain correct trajectories with rewards slightly below the batch mean may be incorrectly treated as negative examples, thus introducing misleading penalty gradients. By contrast, the absolute advantage preserves the positivity of genuinely correct samples and yields more stable and reliable policy updates.

4.2.3 Exponent Reward Shaping

Reward shaping plays a critical role in reinforcement learning by guiding how models adapt their generation behavior. To encourage efficient reasoning, we employ an exponential reward shaping function, as defined in Equation 10. This formulation assigns larger gradients to shorter responses while progressively diminishing the contribution of

longer ones. The strength of compression is controlled by two parameters: L , which specifies the reference response length, and r , which determines the reward magnitude when the response length equals L . This exponential form induces a length-dependent scaling of rewards, resulting in stronger gradient signals for concise trajectories and weaker signals for verbose ones.

$$f(x) = e^{-\gamma x}, \quad \gamma = -\frac{\ln r}{L}. \quad (10)$$

This formulation leads to more efficient optimization. Empirical evidence supporting its effectiveness is provided in Section 5.7.

4.3 Enhancing Stage

After the model has learned concise reasoning, we increase the sampling temperature and broaden the sampling range to encourage freer exploration, thereby further improving its reasoning accuracy and format robustness.

In this stage, the policy is optimized over all sampled trajectories using the GRPO loss:

$$\mathcal{L}_{GRPO}(\theta) = -\mathbb{E}_{y \sim \pi_{\theta}} \left[\frac{\pi_{\theta}(y)}{\pi_{\theta_{old}}(y)} \cdot A(y) \right], \quad (11)$$

where π_{θ} and $\pi_{\theta_{old}}$ denote the current and reference policies, respectively. The advantage function is computed using the relative formulation:

$$A(y) = \frac{R(y) - \mu_R}{\sigma_R}, \quad (12)$$

where μ_R and σ_R represent the group mean and standard deviation of the rewards, respectively.

To provide richer and more interpretable supervision signals, we decompose the total reward into multiple components:

$$R(y) = R_{format} + R_{answer} + R_{length}. \quad (13)$$

To discourage excessively long reasoning, we set R_{length} to decay by 0.25 for every additional 1k tokens beyond the length clipping threshold.

5 Experiment

5.1 Experimental Setups

We evaluate entropy-guided compression on mathematical reasoning models, measuring both accuracy and response length under a standardized evaluation protocol. The full experimental setup, including base models, training data, benchmark suite, efficiency metric, baselines, and length-measurement rules, is provided in Appendix C.

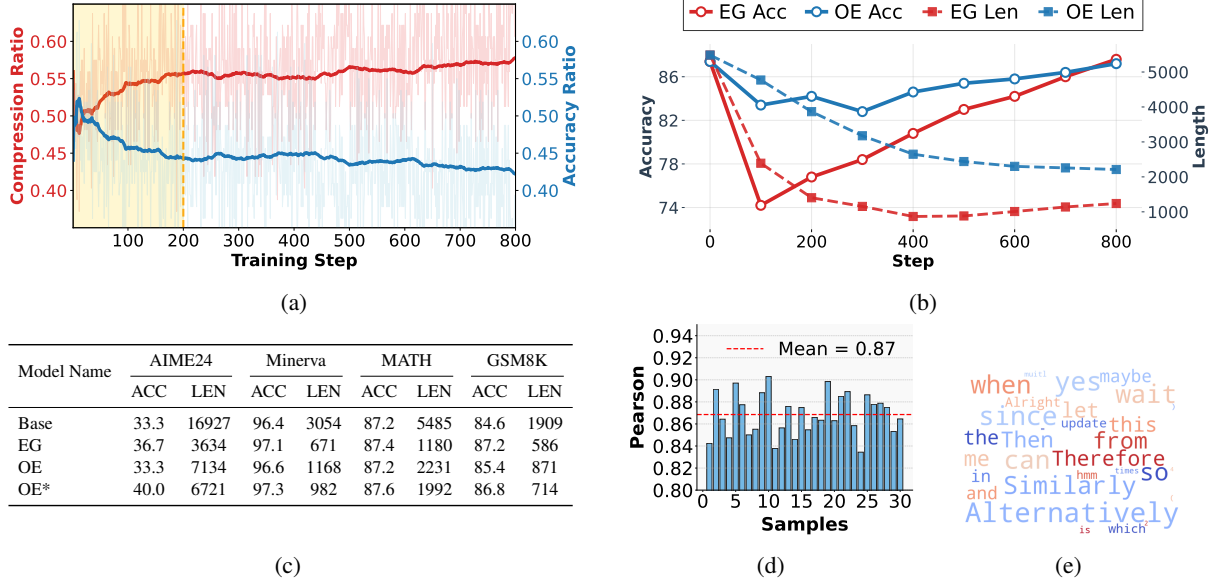


Figure 3: Overall empirical evidence of objective conflict. (a) Distribution of samples selected by the compression objective (red) and accuracy objective (blue) during training. The detailed setup is provided in Section 5.3. (b) Performance comparison on MATH under entropy-guided (EG) and objective-entangled (OE) training. (c) Evaluation of entropy-guided EG, objective-entangled OE, and an extended OE* variant (trained for 400 additional steps) across four benchmarks (AIME24, Minerva, MATH, GSM8K). (d) Pearson correlation between token-level entropy and gradient magnitude during training. (e) Word-cloud visualization of some high-entropy tokens frequently encountered during training.

5.2 Research Questions

We organize the experiments around five research questions. **RQ1** asks whether compression training truly stagnates because of objective conflict; we answer this question through Figure 3(a–c). **RQ2** asks why the accuracy objective tends to favor longer responses; we analyze this mechanism through Figure 3(d–e). **RQ3** asks whether resolving this conflict enables stronger compression; we answer it with the cross-benchmark comparison in Table 1. **RQ4** asks what reasoning pattern emerges after compression; we study this behavior through the qualitative example in Figure 4 and the distributional evidence in Figure 5. **RQ5** asks which components of the proposed method are essential; we investigate this through the ablation results summarized in Appendix E.

5.3 RQ1: Does Compression Training Stagnate Because of Objective Conflict?

To examine whether *objective conflict* emerges in practice, we construct an objective-coupled setup. In each group, samples are split into two categories based on their sampled accuracy. High-accuracy ($\geq 50\%$) samples are treated as compression samples, while low-accuracy ($< 50\%$) samples are used for accuracy optimization.

Answer to RQ1. Compression training does stagnate when the compression objective and the accuracy objective are entangled. The core reason is that the two objectives prefer different correct trajectories: compression favors short-and-correct solutions, whereas the accuracy objective still rewards long-but-correct trajectories. As training proceeds, these incompatible preferences induce opposing gradients on the same reasoning behavior and prevent the model from moving further toward a concise regime.

Figure 3(a–c) makes this failure mode explicit. In Figure 3(a), the ratio of compression-selected and accuracy-selected samples initially shifts, but both curves flatten after roughly 200 steps, showing that optimization becomes trapped in a stable yet suboptimal allocation state. Figure 3(b) reveals the same phenomenon from both response length and accuracy perspectives: under objective-entangled training, response length quickly plateaus even as training continues, while accuracy shows no sustained improvement. This indicates that the model is no longer moving toward a better compression–accuracy trade-off, but oscillates around a stagnant regime induced by the conflicting objectives. Figure 3(c) further confirms that OE cannot sustain strong compression, and even OE* yields only limited additional length reduction after 400 more

Model Name	AIME24			Minerva			MATH			GSM8K			AMC			OlymBench		
	ACC	LEN	LAC	ACC	LEN	LAC	ACC	LEN	LAC	ACC	LEN	LAC	ACC	LEN	LAC	ACC	LEN	LAC
Qwen2.5-1.5B-Ins	0.0	1300	-	77.7	993	-	51.7	855	-	70.2	466	-	0	730	-	7.3	682	-
Qwen2.5-1.5B-Math-Ins	10.0	1128	-	91.8	586	-	76.0	721	-	85.7	447	-	0	823	-	21.4	774	-
R1-Distill-Qwen2.5-1.5B	33.3	16927	-	96.4	3054	-	87.2	5485	-	84.6	1909	-	61.4	11157	-	44.7	11236	-
AutoThink	30.0	7016	23.0	95.5	959	79.1	87.0	1851	70.8	86.4	743	67.5	66.3	4656	50.6	45.2	5296	32.8
AdaptThink	33.3	8492	23.5	96.3	1267	73.7	82	2158	63.9	83.1	971	58.3	62.6	4194	49.5	44.5	5291	32.4
Thinkless	36.7	8848	25.3	95.3	1178	74.7	84.6	2623	61.1	84.5	637	69.0	57.8	6861	35.9	42.5	6625	27.2
ThinkPrune	26.7	8175	19.2	96.0	1403	70.6	85.8	2321	65.2	85.0	907	61.6	63.9	4354	49.9	43.0	4401	33.5
Laser	40.0	5635	32.7	96.6	1323	72.7	87.0	2073	68.6	86.2	996	59.6	65.1	4963	48.5	45.8	5512	32.7
LC-R1	23.3	11124	13.7	94.4	1374	70.0	83.6	2921	57.2	80.1	789	61.3	60.2	4778	45.5	39.3	4639	30.1
DLER	36.7	3365	32.8	97.3	1457	70.4	85.4	1960	68.5	87.5	1165	54.6	69.7	3403	58.1	47.3	3893	38.2
TokenSqueeze	36.7	11199	21.3	94.4	1436	68.7	84.8	3036	56.7	83.8	517	71.5	65.9	6356	43.3	44.8	5902	30.9
ours	36.7	3634	32.5	97.1	671	85.7	87.4	1180	77.4	87.2	586	72.6	66.3	2083	59.8	45.3	2275	40.5
Qwen2.5-7B-Ins	13.3	1016	-	91.2	649	-	74.2	567	-	90.9	279	-	47.5	801	-	39.2	827	-
Qwen2.5-7B-Math	20.0	1429	-	93.1	761	-	63.4	740	-	93.2	439	-	62.5	1022	-	31.5	1037	-
Qwen2.5-7B-Math-Ins	13.3	1363	-	94.2	730	-	81.4	670	-	95.2	323	-	60.0	1029	-	38.9	1027	-
R1-Distill-Qwen2.5-7B	56.7	14108	-	98.7	2435	-	93.4	4181	-	92.8	1700	-	81.9	7630	-	56.2	9031	-
AutoThink	56.7	6611	41.3	98.6	944	77.2	92.6	1808	69.8	93.4	695	71.8	84.3	3638	60.7	55.0	4415	42.9
AdaptThink	53.3	8023	35.0	98.1	1139	71.6	92	1902	67.9	91	812	65.8	83.1	3891	58.2	52.8	3782	43.0
Laser	50.0	6054	37.8	98.3	1276	67.8	93.0	1762	70.7	93.0	965	61.2	81.9	3437	60.7	54.3	3365	45.5
LC-R1	46.7	7384	32.2	97.1	850	78.4	92.4	1653	71.8	90.2	507	75.6	80.7	3418	60.0	53.0	4093	42.3
DLER	50.0	3248	43.9	98.4	1150	71.5	94.2	1518	75.2	93.6	934	62.8	82.8	2267	69.4	56.2	3025	45.8
TokenSqueeze	53.3	9361	30.9	96.3	1027	73.2	90.0	2427	58.3	89.5	429	77.4	83.2	5434	44.6	54.3	4793	37.2
ours	53.3	4094	44.9	98.0	651	83.9	92.4	1078	79.6	93.6	428	81.0	83.1	1838	72.4	53.5	1911	48.6

Table 1: Performance comparison across mathematical reasoning benchmarks. Purple-shaded rows denote methods that directly compress reasoning chains, while green-shaded rows denote methods that learn adaptive reasoning behaviors. **Boldface** indicates the best performance among non-base models.

steps without producing a clear accuracy advantage. Together, these results answer RQ1 positively: objective conflict is a practical bottleneck that directly causes compression training to stall.

5.4 RQ2: Why Does the Accuracy Objective Prefer Longer Responses?

Answer to RQ2. The accuracy objective prefers longer responses because it disproportionately reinforces high-entropy tokens that often signal step transitions in reasoning. These are precisely the tokens that expand a chain of thought by introducing another intermediate step, reformulation, or branch of reasoning. As a result, optimizing for accuracy alone does not merely preserve correctness; it also biases the model toward a more verbose reasoning style and increases entropy. We interpret this token-level behavior as the observable entropy manifestation of the underlying trajectory-level objective conflict.

Figure 3(d-e) reveals this token-level mechanism. Figure 3(d) shows a strong positive correlation between token entropy and gradient magnitude, indicating that uncertain tokens receive larger optimization signals. Figure 3(e) further shows that many of these high-entropy tokens are discourse connectors such as *Then*, *Therefore*, *Similarly*, and *Alternatively*, which typically mark the beginning of a new reasoning segment. Once such connectors are repeatedly reinforced, the model becomes more

likely to generate additional intermediate steps, which naturally lengthens the response. This mechanism explains why the accuracy objective tends to favor longer reasoning traces and clarifies the source of its conflict.

5.5 RQ3: Can Resolving the Objective Conflict Enable Stronger Compression?

Answer to RQ3. Yes. Resolving the objective conflict leads to a substantially better compression-accuracy trade-off than entangled optimization. By separating compression and exploration in time, the model can first establish a compact reasoning regime and then recover capability without being repeatedly pushed back toward verbose trajectories. This optimization improves the attainable frontier rather than merely moving to a different point on the same one.

Table 1 shows this pattern consistently across both 1.5B and 7B scales. Although chain-of-thought reasoning improves mathematical performance, it also dramatically increases response length. Existing baselines reveal the usual trade-off: direct compression methods often shorten responses at the cost of accuracy, while adaptive reasoning methods preserve accuracy but usually compress less aggressively. In contrast, our method maintains strong benchmark accuracy while substantially reducing response length, leading to strong L-Acc across datasets. Appendix D fur-

Query

There exist real numbers x and y , both greater than 1, such that $\log_x(y^x) = \log_y(x^{4y}) = 10$. Find xy .



Response of Original Model

<think>Alright, so I've got this problem here: There exist real numbers x and y ...First, let's recall some logarithm properties...So now, I have two equations:...Hmm, okay. Let me think about how to relate these two equations. I notice that...Maybe I can express both equations...Hmm, this looks a bit complicated, but maybe I can manipulate it...Wait, it's another thought:<Algebraic operations>...So I get xy is 10.<think> ≈ 4000 tokens
So now I have expressions...Great, now I can...But before I go further down that path, maybe I can..Let me rearrange the terms...Bring all terms to the left side...Wait, let me denote...Hmm, not particularly helpful...Hmm, not sure...<Algebraic operations>... **Final Answer** \boxed{10} ≈ 3200 tokens



total 7159 tokens



Response of Compressed Model

<think>First, I recognize that each logarithmic equation Next, I simplify both equations to express themFinally, I calculate the product of x and y .<think> ≈ 200 tokens
We have two equations...Let's simplify both equations...So, first equation:...Second equation:<Algebraic operations>... **Final Answer** \boxed{25} ≈ 1100 tokens



total 1258 tokens

Figure 4: Qualitative case study comparing the original model and the compressed model. The compressed model removes redundant reasoning in the think phase and shifts detailed computation to the final answer, yielding a more compact planning-and-execution pattern.

ther shows that this behavior extends beyond the mathematical benchmarks: on GPQA-Diamond, our compressed models achieve performance comparable to the original models while using substantially fewer reasoning tokens. These results answer RQ3 affirmatively: once the conflict is explicitly resolved, substantially stronger compression becomes achievable in practice.

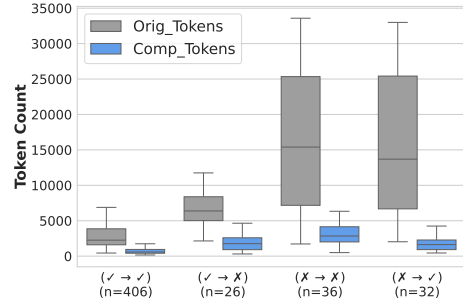
5.6 RQ4: What Reasoning Pattern Emerges After Compression?

Answer to RQ4. As shown in Figure 4, after compression, the model adopts a different reasoning pattern that is shorter, more selective, and more execution-oriented.

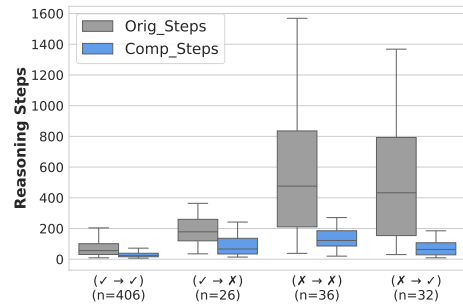
Figure 5 shows that the compressed model uses fewer tokens and steps across correctness-transition groups. In preserved cases, it retains correctness with shorter traces, indicating that much of the original reasoning is redundant; in gained cases, removing distracting intermediate steps can even turn failures into correct answers.

5.7 RQ5: Which Design Choices Are Critical?

Answer to RQ5. Ablations confirm the importance of stage ordering, length clipping, and reward shaping. Reversing the stage order causes compression to disrupt the model's established reasoning paradigm, resulting in severe performance degradation. The length threshold governs the



(a) Token distribution by correctness-transition group.



(b) Step distribution by correctness-transition group.

Figure 5: Comparison of the original and compressed models on Math500.

compression-accuracy trade-off: overly aggressive clipping harms accuracy, whereas loose clipping limits efficiency gains. Finally, exponential reward shaping provides stronger compression while preserving competitive accuracy, validating its emphasis on concise correct trajectories. Detailed results

are provided in Appendix E.

6 Conclusion

We identify trajectory-level objective conflict as a key obstacle to reasoning-chain compression: accuracy preserves long-but-correct trajectories that compression removes, causing entropy conflict and training stagnation. We address this issue with an entropy-guided training framework that separates compression from capability recovery, producing shorter responses while maintaining strong reasoning performance. We further discuss the broader implications of objective conflict in multi-objective optimization, where staged optimization provides a general strategy for nested objectives.

Limitations

While our compression strategy is effective in reducing redundant reasoning, it introduces additional sensitivity to training dynamics. Within the compression stage, identifying an appropriate stopping criterion is non-trivial, as excessive compression in later phases can degrade accuracy, while insufficient compression in earlier phases limits efficiency improvements. Developing more principled or automated criteria for selecting optimal compression stages is an important direction for future work. In addition, our method is designed as a refinement step prior to deployment, where the model adapts its reasoning behavior within a target domain or task distribution. As a result, the learned compression patterns may be specialized to the domains encountered during training and may not generalize equally well to unseen or substantially different tasks. Improving cross-domain generalization of staged entropy-guided compression remains an open challenge.

Acknowledgements

We sincerely thank the anonymous reviewers for their helpful feedback and the conference committee for their hard work. This work has received partial funding from the National Natural Science Foundation of China General Program (Grant No. 62576038) and the Major Research Plan of the National Natural Science Foundation of China (Grant No. 92370110).

References

- Pranjal Aggarwal and Sean Welleck. 2025. [L1: Controlling how long a reasoning model thinks with reinforcement learning](#). In *Second Conference on Language Modeling*.
- Daman Arora and Andrea Zanette. 2025. [Training language models to reason efficiently](#). *Preprint*, arXiv:2502.04463.
- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.
- Zigeng Chen, Xinyin Ma, Gongfan Fang, Ruonan Yu, and Xinchao Wang. 2025. [Verithinker: Learning to verify makes reasoning model efficient](#). *Preprint*, arXiv:2505.17941.
- Zhengxiang Cheng, Dongping Chen, Mingyang Fu, and Tianyi Zhou. 2025. [Optimizing length compression in large reasoning models](#). *Preprint*, arXiv:2506.14755.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Muzhi Dai, Shixuan Liu, and Qingyi Si. 2025. [Stable reinforcement learning for efficient reasoning](#). *Preprint*, arXiv:2505.18086.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Jia Deng, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2025a. [Decomposing the entropy-performance exchange: The missing keys to unlocking effective reinforcement learning](#). *Preprint*, arXiv:2508.02260.
- Wenlong Deng, Yi Ren, Muchen Li, Danica J. Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. 2025b. [On the effect of negative gradient in group relative deep reinforcement optimization](#). *Preprint*, arXiv:2505.18830.
- Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. [Thinkless: Llm learns when to think](#). *Preprint*, arXiv:2505.13379.
- Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. 2025. [Concise reasoning via reinforcement learning](#). *Preprint*, arXiv:2504.05185.
- Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. 2025. [Rethinking entropy interventions](#)

- in rlvr: An entropy change perspective. *Preprint*, arXiv:2510.10150.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand. Association for Computational Linguistics.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.
- Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *Preprint*, arXiv:2504.01296.
- Zeyu Huang, Zihan Qiu, Zili Wang, Edoardo M Ponti, and Ivan Titov. 2024. Post-hoc reward calibration: A case study on length bias. *arXiv preprint arXiv:2409.17407*.
- Ruosen Li, Ziming Luo, Quan Zhang, Ruochen Li, Ben Zhou, Ali Payani, and Xinya Du. 2025a. Aalc: Large language model efficient reasoning via adaptive accuracy-length control. *Preprint*, arXiv:2506.20160.
- Zeju Li, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang, and Qiang Xu. 2025b. Compressing chain-of-thought in llms via step entropy. *Preprint*, arXiv:2508.03346.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. *Preprint*, arXiv:2305.20050.
- Zehui Ling, Deshu Chen, Hongwei Zhang, Yifeng Jiao, Xin Guo, and Yuan Cheng. 2025. Fast on the easy, deep on the hard: Efficient reasoning via powered length penalty. *Preprint*, arXiv:2506.10446.
- Hanbing Liu, Lang Cao, Yuanyi Ren, Mengyu Zhou, Haoyu Dong, Xiaojun Ma, Shi Han, and Dongmei Zhang. 2025a. Bingo: Boosting efficient reasoning of llms via dynamic and significance-based reinforcement learning. *Preprint*, arXiv:2506.08125.
- Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. 2025b. Dler: Doing length penalty right - incentivizing more intelligence per token via reinforcement learning. *Preprint*, arXiv:2510.15110.
- Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. 2025c. Learn to reason efficiently with adaptive length-based reward shaping. *Preprint*, arXiv:2505.15612.
- Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025a. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *Preprint*, arXiv:2501.12570.
- Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. 2025b. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl.
- Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. CoT-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- OpenAI. 2024a. Gpt-4o system card. *Preprint*, arXiv:2410.21276.
- OpenAI. 2024b. Openai o1 system card. *Preprint*, arXiv:2412.16720.
- Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. Verbosity bias in preference labeling by large language models. *Preprint*, arXiv:2310.10076.
- Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. 2025. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *Preprint*, arXiv:2503.04472.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*.
- Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2024. A long way to go: Investigating length correlations in rlhf. *Preprint*, arXiv:2310.03716.
- Mingyang Song and Mao Zheng. 2025. Walk before you run! concise llm reasoning via reinforcement learning. *Preprint*, arXiv:2505.21178.
- Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.
- Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. 2025. Learning when to think: Shaping adaptive reasoning in rl-style models via multi-stage rl. *Preprint*, arXiv:2505.10832.
- Hemish Veeraboina. 2023. Aime problem set 1983-2024.

- Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025. [Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning](#). *Preprint*, arXiv:2506.01939.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. 2025. [Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning](#). *Preprint*, arXiv:2506.05256.
- An Yang, Anfeng Li, and Baosong Yang. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.
- Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. [Demystifying long chain-of-thought reasoning in llms](#). *Preprint*, arXiv:2502.03373.
- Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. [Distilling system 2 into system 1](#). *Preprint*, arXiv:2407.06023.
- Yuxiang Zhang, Zhengxu Yu, Weihang Pan, Zhongming Jin, Qiang Fu, Deng Cai, Binbin Lin, and Jieping Ye. 2025. [Tokensqueeze: Performance-preserving compression for reasoning LLMs](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A Objective Conflict Proof

In this section, we provide the formal assumptions and proofs for Lemma 1 and Theorem 1 in Section 3.

Setup. For an input x , let $\Omega(x)$ denote the set of complete reasoning trajectories and let $\pi_\theta(\tau | x)$ be the policy over $\tau \in \Omega(x)$. We consider the joint objective

$$J_\lambda(\theta) = J_{\text{acc}}(\theta) - \lambda J_{\text{len}}(\theta), \quad (14)$$

where

$$\begin{aligned} J_{\text{acc}}(\theta) &= \mathbb{E}_{x, \tau \sim \pi_\theta} [R_{\text{acc}}(\tau)], \\ J_{\text{len}}(\theta) &= \mathbb{E}_{x, \tau \sim \pi_\theta} [L(\tau)]. \end{aligned} \quad (15)$$

Here $R_{\text{acc}}(\tau) \in [0, 1]$ is the accuracy reward and $L(\tau) \geq 0$ is the trajectory length. The corresponding policy-gradient update is

$$\begin{aligned} \nabla_\theta J_\lambda(\theta) &= \mathbb{E}_{x, \tau \sim \pi_\theta} \left[(R_{\text{acc}}(\tau) - \lambda L(\tau) - b(x)) \cdot \right. \\ &\quad \left. \nabla_\theta \log \pi_\theta(\tau | x) \right], \end{aligned} \quad (16)$$

where $b(x)$ is any baseline independent of τ .

Assumption 1. There exists a non-empty set of inputs $\mathcal{X}_{\text{conf}}$ such that for every $x \in \mathcal{X}_{\text{conf}}$, there exist two correct trajectories $\tau_s, \tau_l \in \Omega(x)$ satisfying

$$\begin{aligned} R_{\text{acc}}(\tau_s) &= R_{\text{acc}}(\tau_l) = 1, \\ L(\tau_s) &< L(\tau_l). \end{aligned} \quad (17)$$

Proof of Lemma 1. Under Assumption 1, for every $x \in \mathcal{X}_{\text{conf}}$, the short-correct trajectory is strictly preferred by the joint objective:

$$\begin{aligned} (1 - \lambda L(\tau_s)) - (1 - \lambda L(\tau_l)) \\ = \lambda (L(\tau_l) - L(\tau_s)) > 0. \end{aligned} \quad (18)$$

Since both trajectories are correct, they receive the same accuracy reward. Therefore, their joint reward difference depends only on the compression term:

$$\begin{aligned} (R_{\text{acc}}(\tau_s) - \lambda L(\tau_s)) - (R_{\text{acc}}(\tau_l) - \lambda L(\tau_l)) \\ = \lambda (L(\tau_l) - L(\tau_s)) > 0. \end{aligned}$$

Thus, among correct trajectories for the same input, every unnecessarily long one receives a strictly smaller joint score than a shorter correct alternative. $\square$

Assumption 2. There exists a parameter-space direction v associated with long-correct trajectories on hard inputs such that

$$\begin{aligned} \langle \nabla_\theta J_{\text{acc}}(\theta), v \rangle &> 0, \\ \langle \nabla_\theta J_{\text{len}}(\theta), v \rangle &> 0. \end{aligned} \quad (19)$$

This means that moving along v improves accuracy, but only by increasing expected reasoning length.

Proof of Theorem 1. Under Assumptions 1 and 2, there exists a critical coefficient

$$\lambda^* = \frac{\langle \nabla_\theta J_{\text{acc}}(\theta), v \rangle}{\langle \nabla_\theta J_{\text{len}}(\theta), v \rangle} > 0 \quad (20)$$

such that the directional derivative of the joint objective along v vanishes:

$$\langle \nabla_\theta J_{\lambda^*}(\theta), v \rangle = 0. \quad (21)$$

By definition,

$$\begin{aligned} \langle \nabla_\theta J_\lambda(\theta), v \rangle &= \langle \nabla_\theta J_{\text{acc}}(\theta), v \rangle \\ &\quad - \lambda \langle \nabla_\theta J_{\text{len}}(\theta), v \rangle. \end{aligned} \quad (22)$$

Under Assumption 2, both terms on the right-hand side are strictly positive. Therefore, choosing $\lambda = \lambda^*$ makes the directional derivative exactly zero:

$$\langle \nabla_\theta J_{\lambda^*}(\theta), v \rangle = 0. \quad (23)$$

This shows that the joint objective reaches a Pareto-stationary regime along v : further compression harms accuracy, while further accuracy recovery necessarily increases reasoning length. $\square$

Interpretation. Lemma 1 establishes the existence of objective conflict at the trajectory level: the compression objective always pushes probability mass away from long-but-correct trajectories when shorter correct alternatives exist. Theorem 1 lifts this conflict to the optimization level: when such long-correct trajectories remain useful for preserving accuracy on hard inputs, the accuracy gradient and the compression penalty can cancel along a shared direction in parameter space. This explains why direct joint optimization tends to stall after partial compression.

B Broader Implications for Staged Optimization

Section 3.2 gives the main takeaway: objective conflict is not unique to reasoning compression, but

reflects a more general issue in multi-objective optimization. This appendix expands that claim in a way that directly matches the main-text discussion. We first restate the trajectory-level conflict in reasoning compression, then reinterpret it through nested objective sets, and finally explain why this structure naturally motivates staged rather than simultaneous optimization.

In our setting, the conflict is most visible on trajectories that are correct yet unnecessarily long. The compression objective penalizes such trajectories to discourage verbosity, whereas the accuracy objective still rewards them because they remain valid solutions. As a result, the same behavior receives contradictory training signals. Empirically, this appears as stagnation: the model stops getting shorter, yet cannot safely discard those trajectories without hurting performance.

More broadly, this phenomenon can be understood through nested objective sets. The compression objective effectively defines a stricter subset of behaviors: among correct trajectories, it prefers only those that are also concise. The accuracy objective is broader, because it preserves all correct trajectories, including long-but-correct ones. When these two objectives are optimized simultaneously, gradients from the broader objective remain active and keep pulling the policy back toward behaviors that the stricter objective is trying to eliminate.

This interpretation explains why staged optimization is preferable in such cases. Rather than jointly optimizing the subset and superset objectives from the beginning, it is more effective to first optimize the stricter sub-objective so that the model internalizes the desired core behavior. Only after this compact regime is established should the broader objective be reintroduced to recover robustness and exploration. Conceptually, the first stage fixes the behavioral rules, while the second stage enlarges the feasible exploration space without fully undoing those rules.

Viewed in this way, the staged procedure is not merely a heuristic for reasoning compression. It is a general strategy for settings where one objective encodes a refined version of another and direct joint optimization would otherwise preserve conflicting gradients from the larger objective set.

C Implementation Details

Table 2 summarizes the training configuration used in all experiments. We fine-tune both base mod-

Configuration	Value
Base models	DeepSeek-R1-Distill-Qwen-1.5B and 7B
Training dataset	DeepScaleR-Preview
Training framework	Verl
Optimizer	AdamW
Learning rate	1×10^{-6}
Batch size	64
Model update interval	16 samples
Responses per input	8
KL-divergence penalty	None
Compression stage	400 steps
Exploration stage	400 steps
Hardware	8 NVIDIA A800 GPUs

Table 2: Training configurations used in our experiments.

els via reinforcement learning on the DeepScaleR-Preview dataset (Luo et al., 2025b), using an implementation based on the Verl framework (Sheng et al., 2024).

We evaluate the trained models on six mathematical reasoning benchmarks, reporting both Pass@1 accuracy and response length. The benchmarks include AIME24 (Veeraboina, 2023), AMC, MATH (Lightman et al., 2023), MinervaMATH (Hendrycks et al., 2021), Olympiad-Bench (He et al., 2024), and GSM-8K (Cobbe et al., 2021). To better assess reasoning efficiency, we additionally adopt the evaluation metric proposed in (Liu et al., 2025a)

$$\text{L-Acc} = \text{Acc} \times \sqrt{1 - \frac{L}{L_{\text{base}}}},$$

where L is the average response length and L_{base} is the length of the base model.

We compare our approach against representative open-source baselines with strong performance, including ThinkPrune (Hou et al., 2025), Laser (Liu et al., 2025c), AutoThink (Tu et al., 2025), LC-R1 (Cheng et al., 2025), Thinkless (Fang et al., 2025), DLER (Liu et al., 2025b), and Token-Squeeze (Zhang et al., 2025). Following Thinkless, we standardize the chat templates across all models when computing response length and include the content within <think> tags in the length measurement.

D General-Domain Evaluation

To evaluate whether the compressed model preserves capability beyond mathematical reasoning benchmarks, we additionally test it on GPQA-

Model	AIME24		Minerva		MATH		GSM8K	
	ACC	LEN	ACC	LEN	ACC	LEN	ACC	LEN
Base	33.3	16927	96.4	3054	87.2	5485	84.6	1909
Stage Ordering Ablation								
$model_{exploration}$	40.0	15127	97.5	2953	88.6	5135	87.9	2202
$model_{exp \rightarrow compression}$	30.0	3763	94.0	714	83.6	1089	80.9	460
$model_{comp \rightarrow exploration}$	36.7	3634	97.1	671	87.4	1180	87.2	586
Length Clip Ablation								
LC = 2048	33.3	3418	94.8	590	84.6	1017	85.1	412
LC = 4096 (ours)	36.7	3634	97.1	671	87.4	1180	87.2	586
LC = 8192	43.3	6662	97.3	829	87.6	1690	87.3	700
Reward Function Ablation								
cosine	36.7	5705	96.6	871	86.2	1428	86.5	683
linear	33.3	4783	96.5	761	86.8	1509	86.0	612
exponent (ours)	36.7	3634	97.1	671	87.4	1180	87.2	586

Table 3: For stage ordering ablation, *exp* denotes the exploration stage and *comp* denotes the compression stage. For the $exp \rightarrow comp$ setting, we report both the intermediate model after exploration and the final model after compression. For length clip ablation, *LC* indicates the maximum length threshold used for clipping overlong reasoning trajectories. For reward function ablation, different reward shaping functions are evaluated under the same training setup. Results are reported in terms of accuracy (ACC) and average response length (LEN) across benchmarks.

Diamond. Table 4 shows that our compressed models achieve performance comparable to the original models while using substantially fewer reasoning tokens. For the 1.5B model, compression improves accuracy from 30.81 to 33.84 while reducing token usage from 10,308 to 5,089. For the 7B model, the compressed model maintains comparable accuracy with a much shorter response length.

Model	ACC	Tokens
R1-Distill-Qwen-1.5B	30.81	10,308
Ours-1.5B	33.84	5,089
R1-Distill-Qwen-7B	49.49	8,570
Ours-7B	48.48	4,944

Table 4: General-domain evaluation on GPQA-Diamond. Our compressed models retain comparable accuracy while using substantially fewer reasoning tokens.

E Ablation Details

Detailed ablation results for stage ordering, length clipping, and reward shaping are reported in Table 3. First, stage ordering is critical. Although exploration alone improves accuracy, applying compression afterwards sharply degrades performance:

for example, MATH accuracy drops from 88.6 to 83.6 and GSM8K accuracy drops from 87.9 to 80.9. In contrast, the proposed $comp \rightarrow exploration$ ordering preserves substantially stronger accuracy while achieving comparable compression, suggesting that exploration is more effective after a compact reasoning regime has been established.

Second, the length-clipping threshold governs the compression–accuracy trade-off. A tight threshold ($LC = 2048$) yields the shortest responses but sacrifices accuracy, whereas a loose threshold ($LC = 8192$) improves accuracy at the cost of substantially longer outputs. The adopted setting ($LC = 4096$) provides a balanced operating point, retaining strong accuracy while keeping response lengths substantially below those of the base model.

Finally, exponential reward shaping consistently provides the strongest length reduction without compromising accuracy. Compared with cosine and linear shaping, the exponential reward produces much shorter responses on AIME24, Minerva, and MATH while maintaining comparable or higher accuracy. This supports the design choice of assigning stronger optimization signals to concise correct trajectories and rapidly diminishing the contribution of verbose ones.