ORBIT: On-policy Exploration-Exploitation for Controllable Multi-Budget Reasoning

Anonymous ACL submission

Abstract

Recent Large Reasoning Models (LRMs) achieve strong performance by leveraging long-form Chain-of-Thought (CoT) reasoning, but uniformly applying overlong reasoning at inference time incurs substantial and often unnecessary computational cost. To address this, prior work explores various strategies to infer an appropriate reasoning budget from the input. However, such approaches are unreliable in the worst case, as estimating the minimal required reasoning effort is fundamentally difficult, and the trade-off between reasoning cost and accuracy is implicitly fixed during training, limiting flexibility under varying deployment scenarios. Motivated by these limitations, we propose **ORBIT**, a controllable multi-budget reasoning framework with well-separated reasoning modes triggered by prompt. ORBIT employs multi-stage reinforcement learning to discover Pareto frontier reasoning behaviors at each effort, followed by On-Policy Distillation to fuse these behaviors into a single unified model. Experiments show that ORBIT achieves (1) controllable reasoning behavior over multiple modes, (2) competitive reasoning density within each mode, and (3) integration of these frontier policies into a single unified student model while preserving clear mode separation and token-efficient per-mode performance.

1 Introduction

Recent advances in Reinforcement Learning with Verifiable Rewards (RLVR) (Guo et al., 2025; Team et al., 2025; Xiaomi et al., 2025) have significantly pushed the frontier of complex task solving and long-form Chain-of-Thought (CoT) (Wei et al., 2022) reasoning in large language models. By encouraging explicit reasoning trajectories, these methods enable models to explore richer solution paths and achieve strong performance on challenging benchmarks (Guo et al., 2025). However, the continued push toward greater reasoning depth introduces a practical trade-off between reasoning effort and operational efficiency: increasingly long reasoning traces incur higher latency and computational cost, while often being unnecessary for simpler queries (Yang et al., 2025b).

Notably, this observation concerns not the use of long-form reasoning during training, but its uncalibrated application at inference time, as longer reasoning chains often facilitate exploration of richer solution paths and contribute to higher peak accuracy (Setlur et al., 2025). The challenge primarily emerges at deployment time, where models trained to reason deeply may apply the same heavy reasoning patterns indiscriminately across inputs.

To optimize inference-time efficiency, prior work has explored various strategies for adjusting reasoning effort according to task demands (Luo et al., 2025a; Lou et al., 2025; Luo et al., 2025b; Zhang et al., 2025). Across these approaches, the underlying goal is essentially the same: to estimate an optimal allocation of reasoning effort.

However, this estimation problem is fundamentally difficult. Determining the minimal amount of reasoning required to solve an instance can be viewed as attempting to estimate its Kolmogorov complexity (Li and Vitnyi, 2008), which is known to be uncomputable and connected to the halting problem theoretically (Graves, 2017). Empirically, adaptive mechanisms can also misallocate reasoning effort—truncating reasoning on instances that would benefit from deeper deliberation (Aggarwal et al., 2025; Jin et al., 2025), while preserving unnecessarily long reasoning for simpler instances (Chen et al., 2025a).

Moreover, these inference-agnostic approaches implicitly determine the trade-off between reasoning effort and expected accuracy during training, reflecting a fixed preference over cost and performance. While suitable for standardized benchmarks, this design may be less flexible in deployment scenarios where user preferences and system constraints vary across queries.

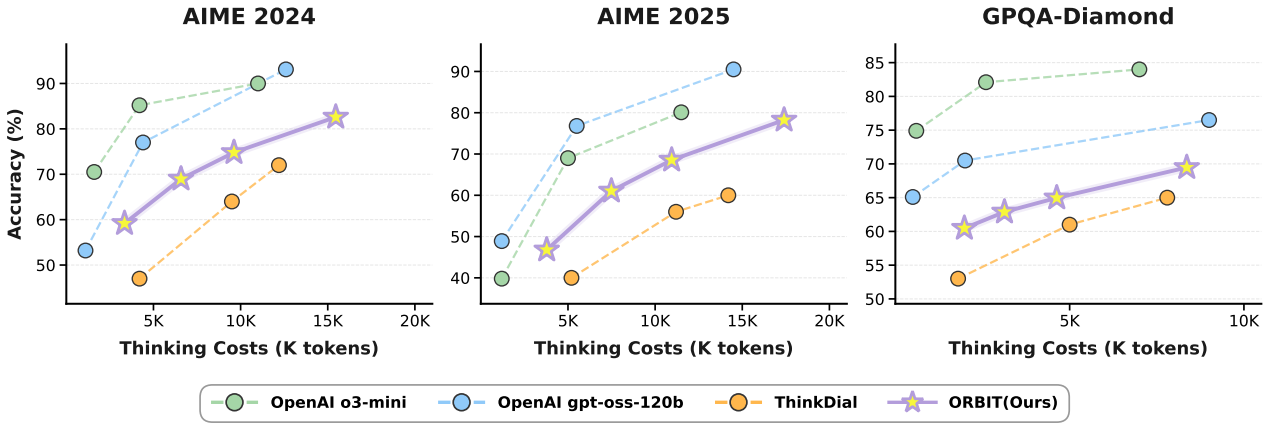

Figure 1: ORBIT demonstrates controllable reasoning behaviors comparable to advanced systems, covering multiple reasoning modes: low/mid/high for o3-mini and gpt-oss-120B, and an extended set of low/mid/high/extra-high for ORBIT, illustrating its ability to flexibly trade off reasoning cost and performance across modes.

Motivated by these limitations, we rethink a controllable reasoning model that can flexibly adjust its reasoning effort according to user preferences by switching reasoning policies across modes at inference time, while achieving competitive reasoning density within each effort level.

We introduce **ORBIT**, a fully on-policy training framework for controllable multi-budget reasoning. By exploring under different training context windows, ORBIT identifies a Pareto frontier of reasoning behaviors and effectively integrates these into a single model whose discrete reasoning modes can be triggered by specific prompts. As illustrated in Figure 1, ORBIT achieves clear and well-separated reasoning modes similar to state-of-the-art controllable systems, while maintaining strong per-mode accuracy and token-efficient reasoning.

Our contributions are as follows:

1. We propose **ORBIT**, a fully on-policy training framework for controllable multi-budget reasoning, enabling a single model to generate responses across multiple reasoning modes triggered by mode-specific prompts.

2. We introduce an **Expansion-Compression Loop** exploration strategy that efficiently discovers mode-specific reasoning frontiers under varying context constraints, yielding high quality policies at each reasoning effort.

3. We design a **Multi-Teacher On-Policy Distillation** pipeline with mode-aware initialization, which fuses these frontier policies into a single unified student model while preserving clear mode separation and token-efficient performance across all modes.

2 Preliminary

Problem Statement We study controllable multi-budget reasoning, aiming to learn a single, unified policy $\pi_\phi(o \mid q, c)$ that generates distinct, mode-specific reasoning behaviors for a query $q \in \mathcal{D}$, where the mode $c \in \mathcal{C} = \{c_1, \ldots, c_K\}$ can be selected or triggered at inference time, corresponding to different levels of reasoning effort.

Crucially, each induced behavior is expected to be *strong in its own right*: for every mode c_k, the conditional policy $\pi_\phi(\cdot \mid q, c_k)$ should achieve high task accuracy relative to its reasoning budget, rather than representing a compromised linear interpolation performance.

The core challenge, therefore, lies in learning such a unified yet controllable policy under a fixed training budget, while maintaining clear behavioral separation across modes.

RLVR We model language generation as autoregressive sampling from a conditional probability distribution $\pi_\theta(o|q)$, and our training goal is to maximize the expected correctness reward $r(q, o) \in \{0, 1\}$:

$$\max_\theta \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_\theta}[r(q, o)], \tag{1}$$

where q represents the input query and o represents the output response.

In this work, we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as our RLVR algorithm for its simplicity and effectiveness in training reasoning models (Guo et al., 2025). Unlike traditional methods (Schulman et al., 2017), GRPO bypasses the value model by computing

advantages relative to a group of G responses $\{o_1, \ldots, o_G\}$ sampled from $\pi_{\theta_{\text{old}}}$. The objective is:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \min\Big(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\Big)\Bigg], \tag{2}$$

where the importance ratio $r_{i,t}(\theta)$ and the group-relative advantage $\hat{A}_i$ are defined as:

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})}, \tag{3}$$

$$\hat{A}_i = \frac{r(q, o_i) - \text{mean}(\{r(q, o_j)\})}{\text{std}(\{r(q, o_j)\})}. \tag{4}$$

3 Methodology

We approach controllable multi-budget reasoning through an *exploration–exploitation* perspective. To operationalize, we decompose the problem into two complementary phases:

Exploration Phase. Multi-stage RL allocates computational effort to sample sequences in semantic space under varying context budgets, discovering high-quality per-mode strategies.

Exploitation Phase. Model merging and On-Policy Distillation consolidate these policies into a single student model, preserving mode-specific behaviors while enabling an efficient unified model for all reasoning modes.

Figure 2 provides an overview of this framework.

3.1 Expansion-Compression Loop for Mode-Specific Policy Exploration

During exploration, each reasoning mode $c \in \mathcal{C}$ is subject to a context budget reflecting its intended reasoning effort. The training objective is to discover high-quality strategies that maximize task accuracy under a given budget. To achieve this, we adopt the simple length regularizing method–truncation, along with a scheduling strategy and formalize the exploration stage as traversal along the accuracy–length Pareto frontier.

Dual-Objectivity under Truncation For a given context budget L_k and reasoning mode $c \in \mathcal{C}$, we consider a policy π_{θ_k} and define its truncated, mode-specific version:

$$\tilde{\pi}^{(c)}_{\theta_k}(o; L_k) := \pi_{\theta_k}(o \mid q, c)\, \mathbb{I}(|o| \le L_k), \tag{5}$$

For a given mode c, the expected reward (accuracy) under truncation is

$$J^{(c)}(\pi_{\theta_k}) = \mathbb{E}_{q\sim\mathcal{D}}\mathbb{E}_{o\sim\tilde{\pi}^{(c)}_{\theta_k}(\cdot|q)}[r(q, o)], \tag{6}$$

We define the non-truncation probability

$$p^{\pi,(c)}_{\text{non-trunc}}(q) := \mathbb{P}_{o\sim\pi_{\theta_k}(\cdot|q,c)}(|o| \le L_k), \tag{7}$$

and the corresponding renormalized truncated policy

$$\pi^{T_k,(c)}_{\theta_k}(o \mid q) := \frac{\pi_{\theta_k}(o \mid q, c)\, \mathbb{I}(|o| \le L_k)}{p^{\pi,(c)}_{\text{non-trunc}}(q)}. \tag{8}$$

Then the truncated expected reward decomposes as

$$J^{(c)}(\pi_{\theta_k}) = \mathbb{E}_{q\sim\mathcal{D}}\Big[p^{\pi,(c)}_{\text{non-trunc}}(q)\, \bar{r}^{\pi,(c)}_{\text{non-trunc}}(q)\Big], \tag{9}$$

$$\bar{r}^{\pi,(c)}_{\text{non-trunc}}(q) := \mathbb{E}_{o\sim\pi^{T_k,(c)}_{\theta_k}(\cdot|q)}[r(q, o)]. \tag{10}$$

This decomposition shows that, for each mode c, optimization under truncation involves two coupled directions:

- increasing $p^{\pi,(c)}_{\text{non-trunc}}$, i.e., concentrating probability mass on valid-length trajectories;
- increasing $\bar{r}^{\pi,(c)}_{\text{non-trunc}}$, i.e., improving the semantic quality of non-truncated trajectories.

Harvesting Along the Frontier via an Expansion–Compression Loop To efficiently obtain per-mode strategies under truncation, we introduce a one-time **Expansion–Compression Loop** for scheduling context window in RLVR training:

- **Expansion (Semantic Optimality).** We first enlarge the context budget to $L_{\text{exp}} \gg \mathbb{E}[|o^+|]$, where o^+ denotes correct, within-window trajectories, ensuring truncation is inactive and the policy focuses on semantic correctness. This allows exploration of harder instances and establishes a global performance ceiling.
- **Sequential Compression (Density Optimality).**

  Starting from the expanded policy, we progressively tighten the context budget L_k. This

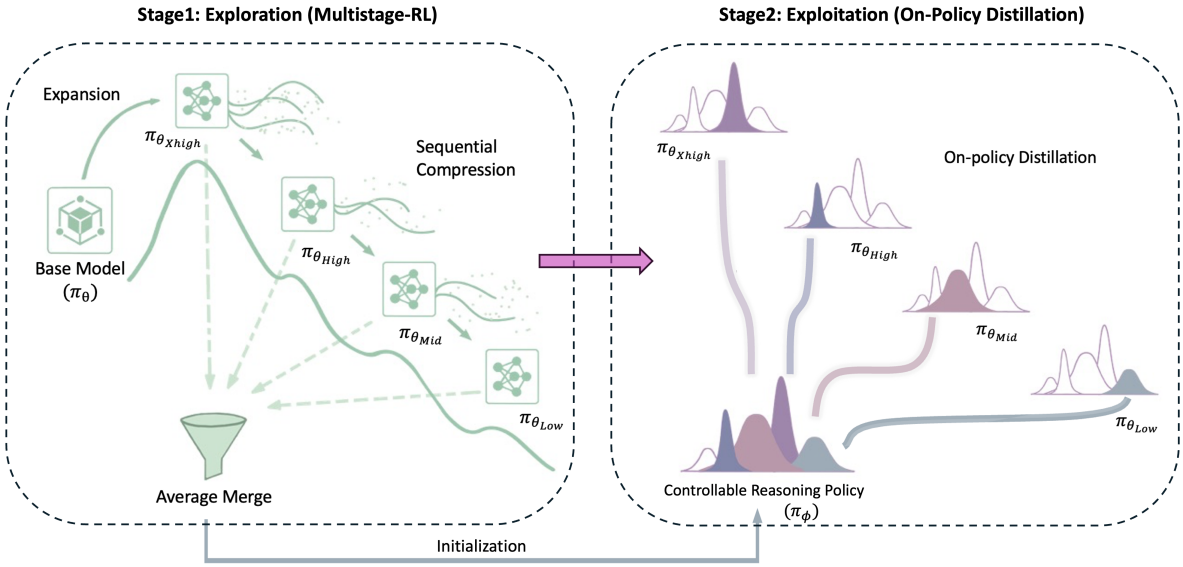

Figure 2: The ORBIT framework overview: (1) In the **Exploration** stage, multi-stage RL discovers specialized reasoning frontiers under varying context from the Expansion-Compression Loop. (2) In the **Exploitation** stage, these behaviors are unified into a single student model via Model Merging and Multi-Teacher On-Policy Distillation.

constraint imposes truncation, inducing a discontinuous length distribution that can destabilize training through two mechanisms: (i) increased reward variance due to the scarcity of valid-length correct trajectories, which leads to noisy gradient estimates; and (ii) "survival mode," where the model collapses to the shortest, trivial solutions, suppressing the complex reasoning required for harder instances.

To balance mode separability and training stability, we adopt a logarithmic compression schedule ($L_{k+1} = L_k/2$) and sequentially optimize the policy under progressively smaller budgets. This schedule applies comparable truncation pressure on the remaining trajectory mass at each stage, allowing the policy to re-equilibrate under the current constraint before the next compression is applied. Further discussion of this design is provided in Appendix A.

Notably, binary compression serves as the canonical instantiation of our framework. If finer-grained trade-offs between output quality and latency are desired, additional compression stages can be simply inserted into the current schedule.

The loop yields stage-wise per-mode experts along the accuracy–length frontier, which provide policy anchors in semantic space for consolidating multiple reasoning modes into a single model.

3.2 On-Policy Distillation for Frontier Policy Exploitation

While the exploration stage identifies frontier policies, REINFORCE-style updates provide only a sparse $O(1)$ supervision signal per rollout sequence (uniform $\hat{A}$ across all tokens). Such scalar feedback is insufficient not only for efficient consolidation, but also for reasoning behaviors anchoring across multiple mode-specific experts.

Transitioning from *discovery* to *consolidation*, we therefore adopt distillation as a high-bandwidth alternative, enabling supervision at a tokenwise $O(N)$ resolution. Specifically, we treat the frontier expert models obtained during the exploration stage, $\{\pi_{\theta_k}\}_{k=1}^{K}$, as teachers and fuse their mode-specific behaviors into a single unified student model π_ϕ. This is achieved through a unified pipeline that combines **Multi-Teacher On-Policy Distillation** with a **Mode-aware Initialization**, allowing diverse reasoning modes to be consolidated in a structured and controllable manner.

On-Policy Distillation Mode Fusion A common intuition is to perform distillation using an offline dataset of query–response pairs sampled from the teacher model, and then train the student in an SFT manner.

However, distilling an autoregressive student model in this way leads to a training–inference distribution mismatch, commonly known as *exposure bias* (Bengio et al., 2015), because supervision

is provided on teacher-generated trajectories that the student policy may never enter during inference, preventing it from learning to *calibrate its own behavior* on the trajectories induced by its own policy (Agarwal et al., 2023).

Based on these intuitions, we adopt **On-Policy Distillation** (Agarwal et al., 2023; Lu and Lab, 2025), minimizing the Reverse Kullback-Leibler (RKL) divergence:

$$\mathcal{L}_{\text{distill}}(\phi) = \mathbb{E}_{q\sim\mathcal{D}, o\sim\pi_\phi(\cdot|q,p_k)}\left[\log \frac{\pi_\phi(o|q,p_k)}{\pi_{\theta_k}(o|q,p_k)}\right]. \tag{11}$$

RKL is inherently **mode-seeking**, encouraging the student π_ϕ to focus on the high-density reasoning modes of the expert π_{θ_k} under prompt p_k for mode c_k. By concentrating probability mass on these modes, the student is less likely to assign weight to irrelevant trajectories or non-target ones. This ensures that the reasoning knowledge from each expert is effectively consolidated, while preserving the model's broader generative capabilities.

Formally, the objective function is defined as

$$\mathcal{L}_{\text{fusion}}(\phi) = \mathbb{E}_{k\sim U(1,K)}\mathbb{E}_{\substack{q\sim\mathcal{D}\\ o\sim\pi_\phi(\cdot|q,p_k)}}\left[\sum_t \log \frac{\pi_\phi(o_t|q,p_k,o_{<t})}{\pi_{\theta_k}(o_t|q,p_k,o_{<t})}\right] \tag{12}$$

Mode-aware initialization via Model Merging Despite the theoretical advantages of RKL, On-Policy Distillation suffers a "cold-start": if the student π_ϕ initially lacks confidence in the target modes, rollouts rarely reach the expert's high-density manifold, limiting the supervisory signal (Jang et al., 2026). We mitigate this by initializing the student via model merging, combining the stage-wise frontier models $\{\pi_{\theta_k}\}_{k=1}^{K}$ obtained during exploration, to align input-output mappings with the teacher π_{θ_k}. This provides a more stable start and accelerates convergence. Details can be found in Appendix B.

4 Experiments

4.1 Experimental Settings

We implemented ORBIT on three models—`DeepSeek-Distill-Qwen-1.5B`, `Qwen3-4B-2507-Thinking`, and `Openmath-Nemotron-7B`—each configured with four reasoning modes: Low, Mid, High, and Xhigh. The training context lengths for these modes were set according to the model: 2K, 4K, 8K, and 16K tokens for `DeepSeek-Distill-Qwen-1.5B`, and 4K, 8K, 16K, and 32K tokens for both `Qwen3-4B-2507-Thinking` and `Openmath-Nemotron-7B`. Specifically, `DeepSeek-Distill-Qwen-1.5B` was trained on the DAPO-Math-17K dataset (Yu et al., 2025), while `Qwen3-4B-2507-Thinking` and `Openmath-Nemotron-7B` were trained on Polaris-53K (An et al., 2025). Notably, the training procedure is scheduled in reverse order. Further implementation and training details are provided in Appendix C.

Baselines. We first evaluate our full pipeline against the recent controllable reasoning models, including o3-mini (OpenAI, 2025), gpt-oss-120b (OpenAI, 2025), and ThinkDial (He et al., 2025). Next, we compare ORBIT's within-mode performance against current RL-based adaptive and efficient reasoning methods (Aggarwal and Welleck, 2025; Arora and Zanette, 2025; Zhang et al., 2025; Fang et al., 2025; Tu et al., 2025; Liu et al., 2025) on `DeepSeek-Distill-Qwen-1.5B`. Finally, we ablate ORBIT with other potentially used methods and baselines.

Evaluation Benchmarks. We evaluate our methods on three mathematical reasoning tasks: AIME 2024 & 2025, BeyondAIME (ByteDance-Seed, 2025), as well as representative multi-task benchmarks spanning diverse domains in science, humanities, and professional knowledge: GPQA-Diamond (Rein et al., 2024) and MMLU-Pro (Wang et al., 2024).

4.2 Main Results

ORBIT Achieves Controllable Reasoning Behavior Across Domains Table 1 illustrates the reasoning behaviors learned by ORBIT under different reasoning budgets across domains. We observe a clear and consistent separation between reasoning modes, with each exhibiting a monotonic trade-off between reasoning effort and task performance. This separation indicates that ORBIT successfully learns prompt triggered inference-time separable distinct reasoning policies.

Importantly, these controllable behaviors transfer reliably across domains. When transferring from mathematical reasoning to general question answering benchmarks, the ordinal ordering of reasoning modes is preserved, while only minimal stepwise performance degradation is incurred.

ORBIT Achieves Competitive Token Efficiency Within Each Mode We evaluate each ORBIT

Models	Mathematics						Multi-Task				Overall	
	AIME 24		AIME 25		Beyond AIME		GPQA		MMLU			
	avg@32	Toks	avg@32	Toks	avg@8	Toks	avg@8	Toks	avg@1	Toks	Avg.	Toks
DeepSeek-Distill-Qwen-1.5B												
Initial Model	28.6	18.8	22.8	18.5	10.1	18.6	28.0	12.7	35.5	7.2	25.0	15.2
- ORBIT Low	28.7	2.7	21.1	2.3	9.9	1.9	34.8	1.8	38.1	1.4	26.5	2.0
- ORBIT Mid	34.5	5.6	26.0	4.9	12.6	3.6	34.3	2.6	38.2	2.0	29.1	3.7
- ORBIT High	38.8	7.7	30.2	7.4	15.6	7.1	41.0	4.9	41.7	3.4	33.5	6.1
- ORBIT Xhigh	42.6	8.5	31.1	7.8	15.9	7.7	41.8	6.4	41.2	4.4	34.5	7.0
Qwen3-4B-2507-Thinking												
Initial Model	81.6	20.1	77.1	18.3	52.4	27.6	66.1	8.8	74.4	4.4	70.3	15.8
- ORBIT Low	59.7	3.3	46.8	3.8	31.4	3.4	60.5	2.0	71.7	1.1	54.0	2.7
- ORBIT Mid	69.5	6.6	61.0	7.5	40.6	7.4	62.9	3.1	72.8	1.9	61.4	5.3
- ORBIT High	75.0	9.6	69.2	11.0	48.3	11.8	65.2	4.6	73.5	2.8	66.2	8.0
- ORBIT Xhigh	82.6	15.5	78.2	17.4	53.5	20.7	69.5	8.4	74.5	5.4	71.7	13.5
Openmath-Nemotron-7B												
Initial Model	71.6	11.9	59.9	13.7	41.8	13.6	32.5	8.5	42.4	5.7	49.6	10.7
- ORBIT Low	60.0	5.2	36.5	5.7	31.5	4.7	31.5	2.4	39.3	1.8	39.8	4.0
- ORBIT Mid	66.4	7.4	45.2	8.3	35.0	7.5	32.3	4.2	41.5	2.9	44.1	6.1
- ORBIT High	71.4	10.0	57.1	11.6	41.4	10.9	31.9	6.1	41.7	4.4	48.7	8.6
- ORBIT Xhigh	72.0	11.0	57.8	12.1	41.8	12.1	34.8	6.6	42.2	5.0	49.7	9.4

Table 1: Comparison of reasoning modes of ORBIT across various reasoning tasks. "Avg@k" denotes the average accuracy (in %) over k generations (i.e., pass@1), and "Toks" indicates the average output length in thousands of tokens (K). Darker colors in the cell background denote better results.

Models	AIME 24		AIME 25		Beyond AIME		GPQA		MMLU		Overall	
	avg@32	Toks	avg@32	Toks	avg@8	Toks	avg@8	Toks	avg@1	Toks	Avg.	Toks
Initial Model	28.6	18.8	22.8	18.5	10.1	18.6	28.0	12.7	35.5	7.2	25.0	15.2
1) TLMRE-DS-1.5B ($\alpha = 0.1$)	28.1	14.7	22.1	14.2	8.4	12.7	30.8	7.8	34.3	4.7	24.7	10.8
2) AdaptThink-1.5B ($\delta = 0.05$)	25.2	7.6	21.3	8.1	9.9	5.1	31.3	6.7	36.9	4.1	24.9	6.3
3) AutoThink-DS-1.5B-Stage3	31.0	11.0	22.8	10.6	11.0	9.4	31.4	7.6	36.3	4.5	26.5	8.6
4) Laser-DE-L4096-1.5B	30.6	8.2	22.0	7.5	10.3	7.2	34.6	3.4	37.8	2.3	27.1	5.7
5) AutoThink-Stage3*	39.5	9.0	29.0	9.0	12.4	8.0	35.7	4.7	39.6	3.6	31.2	6.9
6) L1-1.5B-Max*	23.6	3.4	22.3	3.1	10.4	3.1	33.0	1.2	39.5	1.4	25.8	2.4
7) Thinkless-1.5B-RL*	27.8	11.1	21.8	11.1	11.1	11.6	28.9	12.8	35.2	6.6	25.0	10.6
- ORBIT Low	28.7	2.7	21.1	2.3	9.9	1.9	34.8	1.8	38.1	1.4	26.5	2.0
- ORBIT Mid	34.5	5.6	26.0	4.9	12.6	3.6	34.3	2.6	38.2	2.0	29.1	3.7
- ORBIT High	38.8	7.7	30.2	7.4	15.6	7.1	41.0	4.9	41.7	3.4	33.5	6.1
- ORBIT Xhigh	42.6	8.5	31.1	7.8	15.9	7.7	41.8	6.4	41.2	4.4	34.5	7.0

Table 2: Comparison of reasoning modes of ORBIT trained from DeepSeek-Distill-Qwen-1.5B with efficient reasoning baselines across various reasoning tasks. Each reasoning mode is evaluated and compared individually.

mode separately on DeepSeek-Distill-Qwen-1.5B, comparing its accuracy–length trade-off with RL-based efficient reasoning methods. We aim to determine if the multi-mode student maintains high reasoning density across all triggered efforts without performance compromise.

As shown in Table 2, ORBIT consistently establishes a superior accuracy–length Pareto frontier, reaching higher accuracy under the same reasoning budget or matching baseline accuracy with fewer tokens. Notably, these results are obtained with a minimalist truncation-based reward combined with our scheduling and fusion strategy, highlighting the effectiveness and simplicity of ORBIT.

5 Analysis

We conducted a series of analytical experiments on the best-performing Qwen3-4B-2507-Thinking.

Comparison with End-to-End Joint RL Training We first investigate whether mode separation can be integrated into the RL training, where we jointly train each single query under four different reasoning modes, each associated with the specific system prompt and corresponding context window.

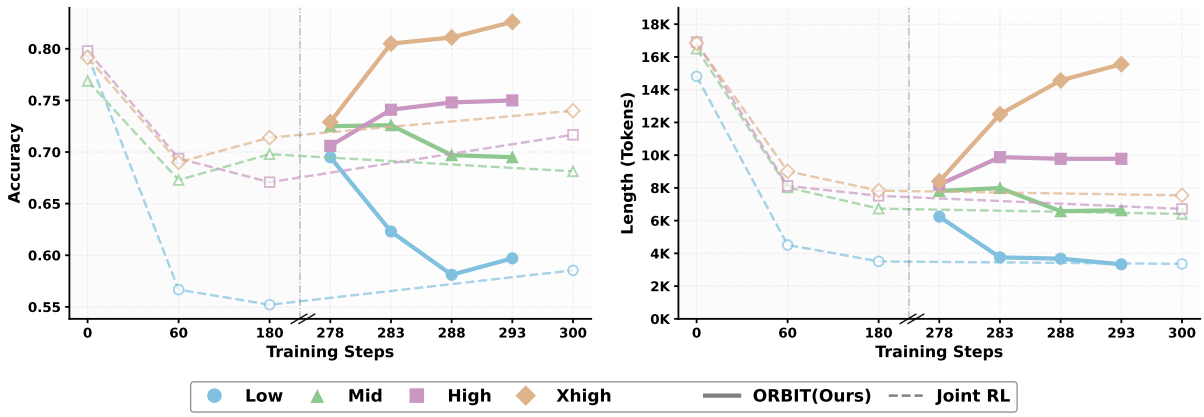

Figure 3: Step-wise performance of ORBIT and joint RL on AIME24, with steps aligned by total training tokens for fair comparison. ORBIT starts from step 278, where OPD training begins. Sequential compression stages are omitted for clarity.

To ensure the peak performance, we start our joint RL training after expansion stage, and stabilize the training with rejection sampling to prevent training oscillation under the most aggressive truncation (e.g. 32K $\rightarrow$ 4K). All other training settings are identical to those of our sequential counterpart.

As shown in Figure 3, joint RL leads to mode collapse, where the Mid, High, and Xhigh modes converge to nearly identical behaviors.

We identify two key factors contributing to this: (i) supervision from truncated sequences dominates optimization, which is consistent with our truncation-rate decomposition in Sec. 3.1; and (ii) higher-mode sequences can satisfy their truncation requirements by compressing below lower modes. Together, these effects encourage reward hacking on response length rather than the intended mode-selective reasoning behavior.

Analysis of Fusion Strategies We examine ORBIT's fusion strategy by replacing on-policy fusion with offline distillation, where consolidation proceeds without additional on-policy rollouts, relying instead on a pre-sampled dataset from the exploration-stage frontier models. Both variants are initialized from the same merged model to minimize behavioral inconsistency.

As shown in Figure 4, on-policy fusion achieves slightly higher performance than offline distillation, while both methods exhibit similar convergence trends. This outcome can be attributed to the strong framework-level alignment in ORBIT: all teacher and student models share a sequential training with same base model trained on same data with same algorithm, effectively reducing policy mismatch and rendering offline distillation in this setting closer to a semi-offline procedure.

Beyond performance, on-policy fusion provides operational advantages: it generates rollouts dynamically, adapts convergence on-the-fly, and can naturally integrate with RL-based correctness signals. In contrast, offline distillation decouples dataset construction from training, allowing for scalable evaluation via pre-collected datasets, which can be leveraged in techniques like Rejection Fine-tuning. The choice between these strategies, or their combination, can be tailored depending on specific application requirements.

ORBIT's Soft-Mode Constraint Elicits Cognition Separation To further analyze how ORBIT's mode control shapes reasoning behavior, we evaluate it on MATH500 (Lightman et al., 2023) and visualize the evolution of reasoning components in Figure 5.

We categorize ORBIT's reasoning behavior along three functional levels: (i) *Logical Deduction* steps grow linearly as modes increase, reflecting structured problem-solving calibrated by training context; (ii) *Self-Reflection* exhibits characteristics analogous to System 2 thinking via an exponential-like increase, allocating computational effort toward iterative reasoning as modes advance; and (iii) *Information Extraction* remains constant, suggesting that increased token usage stems from deeper reasoning rather than additional input parsing.

Together, these trends suggest a structured and hierarchical organization of reasoning behavior, in which different reasoning components occupy distinct functional levels and higher reasoning modes systematically amplify the contributions of more reflective, high-level processes.

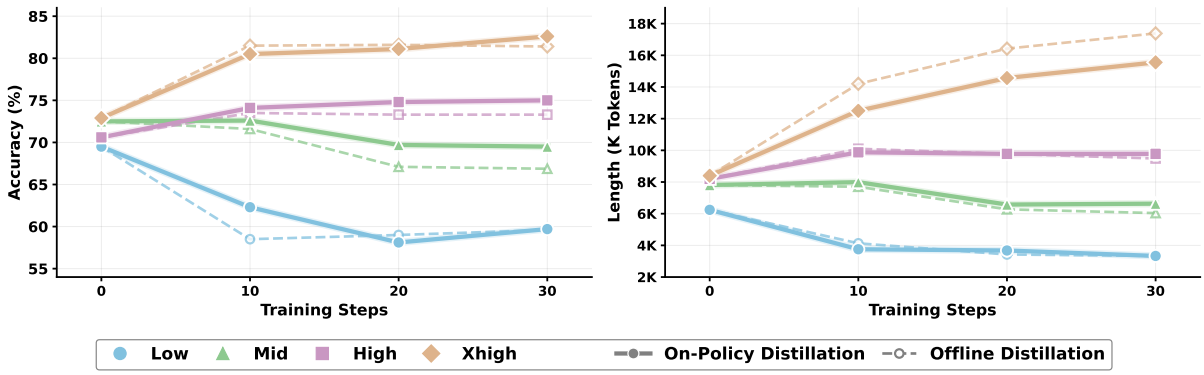

Figure 4: Comparison of OPD and offline distillation as the fusing strategy on AIME24, with steps aligned by total training samples for fair comparison. On-policy fusion achieves slightly higher performance while both strategies show similar trends.

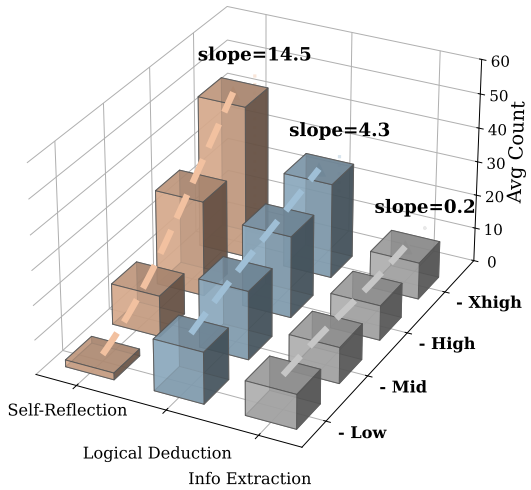

Figure 5: Cognition analysis of ORBIT across reasoning modes, vertical axis denotes average count per sample.

6 Related Work

Reasoning Effort Adjustment in LRMs Existing works on reasoning effort adjustment primarily target overthinking through RL-based length penalties (Aggarwal and Welleck, 2025; Arora and Zanette, 2025; Yi et al., 2025; Zhang et al., 2025; Fang et al., 2025; Lou et al., 2025), verification training (Chen et al., 2025b), early exit strategies (Yang et al., 2025a; Jiang et al., 2025), or inference-time interventions (Wang et al., 2025). Few lines of methods tackle the underthinking problem by decoding time intervention (Muennighoff et al., 2025; Jin et al., 2025). ORBIT addresses these problems by offering flexible reasoning modes at inference time.

Controllable Reasoning Generation Prior work for reasoning control can be broadly categorized in two main streams (1) *Hard Token Budget Constraint:* A common approach enforces reasoning efficiency through externally specified constraints, most notably by fixing explicit token budgets that cap the length of intermediate reasoning (Muennighoff et al., 2025; Sun et al., 2025; Aggarwal and Welleck, 2025; Anthropic, 2025), relying on an absolute tokenwise budget estimation across all inputs. (2) *Discrete-Mode Systems:* Offer intuitional discrete reasoning levels, enabling users to explicitly specify their preference for the efficiency-performance trade-off (OpenAI, 2025; He et al., 2025), representing a soft form of control based on model's own capability. ORBIT falls into this category, controlling reasoning effort relative to the task through explicit mode selection.

7 Conclusion

In this paper, we propose ORBIT, a fully on-policy training framework for learning controllable reasoning models that operates at a target reasoning level while maintaining high reasoning density within each mode. We employ a simple truncation reward and a one-time Expansion–Compression Loop to explore the reasoning frontier, followed by On-Policy Distillation to consolidate multiple modes into a unified model initialized from merged frontier policies, which enables inference-time mode control via natural language prompts. Experiments across LRMs of different sizes show that ORBIT enables controllable reasoning across modes while maintaining competitive accuracy–length trade-offs within each mode compared to existing RL-based efficient reasoning methods.

Limitations

We elucidate the limitations of this work as follows: First, long-form RL-based training can be computationally expensive for larger models, since each rollout requires autoregressive sampling, which may limit scalability to very large architectures or long-context reasoning tasks. Moreover, when the model output entropy is low, on-policy RL training may suffer from low sample efficiency, limiting further compression progress. Second, our evaluation mainly focuses on reasoning tasks with relatively structured verification signals, and we have not extended experiments to tasks where correctness is difficult to verify with rule-based criteria. Finally, despite the empirical results, we do not establish theoretical guarantees on the convergence or optimality of the proposed training procedure.

Ethical Considerations

Use of AI Assistants

The algorithmic design and main methodology presented in this work were developed through human-led research and reasoning. During implementation, we utilized GitHub Copilot 3[1] for coding assistance. We affirm that the primary content and logic of the code are entirely our own work.

Potential Risks

This work introduces ORBIT (On-policy Exploration-Exploitation for Controllable Multi-Budget Reasoning), a framework for training language models with controllable reasoning modes. We acknowledge potential risks associated with controllable reasoning models. Misuse could lead to over-reliance on automated reasoning, propagation of incorrect conclusions, or reinforcement of biases present in the training data. Moreover, the ability to choose reasoning modes in ORBIT may lead to misleading outputs if used improperly or adversarially.

[1]https://github.com/features/copilot

References

Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2023. Gkd: Generalized knowledge distillation for auto-regressive sequence models. *CoRR*.

Pranjal Aggarwal, Seungone Kim, Jack Lanchantin, Sean Welleck, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. Optimalthinkingbench: Evaluating over and underthinking in llms. *Preprint*, arXiv:2508.13141.

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *Preprint*, arXiv:2503.04697.

Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. 2025. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models.

Anthropic. 2025. Building with extended thinking.

Daman Arora and Andrea Zanette. 2025. Training language models to reason efficiently. *Preprint*, arXiv:2502.04463.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In *Advances in Neural Information Processing Systems*, volume 28.

ByteDance-Seed. 2025. Beyondaime: Advancing math reasoning evaluation beyond high school olympiads. https://huggingface.co/datasets/ByteDance-Seed/BeyondAIME.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025a. Do not think that much for 2+3=? on the overthinking of o1-like llms. *Preprint*, arXiv:2412.21187.

Zigeng Chen, Xinyin Ma, Gongfan Fang, Ruonan Yu, and Xinchao Wang. 2025b. Verithinker: Learning to verify makes reasoning model efficient. *Preprint*, arXiv:2505.17941.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Thinkless: Llm learns when to think. *Preprint*, arXiv:2505.13379.

Alex Graves. 2017. Adaptive computation time for recurrent neural networks. *Preprint*, arXiv:1603.08983.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638.

Qianyu He, Siyu Yuan, Xuefeng Li, Mingxuan Wang, and Jiangjie Chen. 2025. ThinkDial: An Open Recipe for Controlling Reasoning Effort in Large Language Models. *Preprint*, arXiv:2508.18773.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggu Lim, and Taesup Kim. 2026. Stable On-Policy Distillation through Adaptive Target Reformulation. *Preprint*, arXiv:2601.07155.

Guochao Jiang, Guofeng Quan, Zepeng Ding, Ziqin Luo, Dixuan Wang, and Zheng Hu. 2025. Flashthink: An early exit method for efficient reasoning. *Preprint*, arXiv:2505.13949.

Liuxuan Jiao, Chen Gao, Yiqian Yang, Chenliang Zhou, YiXian Huang, Xinlei Chen, and Yong Li. 2025. Analyzing and modeling LLM response lengths with extreme value theory: Anchoring effects and hybrid distributions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Hyunbin Jin, Je Won Yeom, Seunghyun Bae, and Taesup Kim. 2025. "well, keep thinking": Enhancing llm reasoning with adaptive injection decoding. *Preprint*, arXiv:2503.10167.

Ming Li and Paul M.B. Vitnyi. 2008. *An Introduction to Kolmogorov Complexity and Its Applications*, 3 edition. Springer Publishing Company, Incorporated.

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.

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

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. 2025. Adacot: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *arXiv preprint arXiv:2505.11896*.

Kevin Lu and Thinking Machines Lab. 2025. On-policy distillation. *Thinking Machines Lab: Connectionism*. Https://thinkingmachines.ai/blog/on-policy-distillation.

Haotian Luo, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao, Dacheng Tao, and Li Shen. 2025a. Ada-r1: Hybrid-cot via bi-level adaptive reasoning optimization. *Preprint*, arXiv:2504.21659.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025b. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *Preprint*, arXiv:2501.12570.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card.

OpenAI. 2025. Openai o3-mini system card.

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

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

Amrith Setlur, Matthew YR Yang, Charlie Snell, Jeremy Greer, Ian Wu, Virginia Smith, Max Simchowitz, and Aviral Kumar. 2025. e3: Learning to explore enables extrapolation of test-time compute for llms. *arXiv preprint arXiv:2506.09026*.

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

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*.

Yi Sun, Han Wang, Jiaqiang Li, Jiacheng Liu, Xiangyu Li, Hao Wen, Yizhen Yuan, Huiwen Zheng, Yan Liang, Yuanchun Li, and 1 others. 2025. An empirical study of llm reasoning ability under strict output length constraint. *arXiv preprint arXiv:2504.14350*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 77 others. 2025. Kimi k1.5: Scaling Reinforcement Learning with LLMs. *Preprint*, arXiv:2501.12599.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. 2025. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage rl. *Preprint*, arXiv:2505.10832.

Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. 2025. Wait, we don't need to "wait"! removing thinking tokens improves reasoning efficiency. *Preprint*, arXiv:2506.08343.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging

multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *Preprint*, arXiv:2203.05482.

LLM-Core Xiaomi, Bingquan Xia, Bowen Shen, Cici, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, Liang Zhao, Peidian Li, Peng Wang, Shihua Yu, Shimao Chen, Weikun Wang, Wenhan Ma, Xiangwei Deng, and 45 others. 2025. MiMo: Unlocking the Reasoning Potential of Language Model – From Pretraining to Posttraining. *Preprint*, arXiv:2505.07608.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Zheng Lin, Li Cao, and Weiping Wang. 2025a. Dynamic early exit in reasoning models. *Preprint*, arXiv:2504.15895.

Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. 2025b. Towards thinking-optimal scaling of test-time compute for llm reasoning. *Preprint*, arXiv:2502.18080.

Jingyang Yi, Jiazheng Wang, and Sida Li. 2025. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. *arXiv preprint arXiv:2504.21370*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025. Adaptthink: Reasoning models can learn when to think. *Preprint*, arXiv:2505.13417.

A Intuition for Logarithmic Compression Scheduling

The logarithmic schedule used in the expansion–compression loop can be motivated by a simple structural property under heavy-tailed trajectory length distributions in LLM responses (Jiao et al., 2025).

Let $T := |o|$ denote the trajectory length. Define the tail probability

$$S(L) := \mathbb{P}(T > L),$$

which measures the probability mass of trajectories whose length exceeds the context budget L.

Over the relevant range of budgets, we approximate the tail distribution by a heavy-tailed form:

$$S(L) = CL^{-\alpha}, \qquad C > 0,\ \alpha > 0. \tag{13}$$

Under a multiplicative compression schedule

$$L_{k+1} = \rho L_k, \qquad 0 < \rho < 1, \tag{14}$$

the truncated trajectory mass satisfies

$$\frac{S(L_{k+1})}{S(L_k)} = \frac{C(\rho L_k)^{-\alpha}}{CL_k^{-\alpha}} = \rho^{-\alpha}. \tag{15}$$

which is independent of k. Thus each compression stage imposes a constant relative truncation pressure on the trajectory distribution.

By contrast, for example, under an additive schedule

$$L_{k+1} = L_k - \Delta, \qquad \Delta > 0, \tag{16}$$

the corresponding change becomes

$$\frac{S(L_{k+1})}{S(L_k)} = \left(\frac{L_k}{L_k - \Delta}\right)^{\alpha}, \tag{17}$$

which depends on L_k and grows as L_k decreases.

We therefore take a logarithmic scheduling with $\rho = 0.5$ in ORBIT for a balance between stable truncation and clear separation of different reasoning policies.

B Mode-aware initialization via Model Merging

As described in Sec. 3.2, optimizing Eq. 12 in practice faces a "cold-start" challenge. To mitigate this misalignment, we proposed using model merging as an initialization of the student model.

Model merging acts as a simple linear combination of model parameters. Numerous methods have been proposed, we use average merging (Wortsman et al., 2022) here for simplicity. This process can be defined as follows:

$$\phi = \frac{1}{K}\sum_{k=1}^{K}\theta_k \tag{18}$$

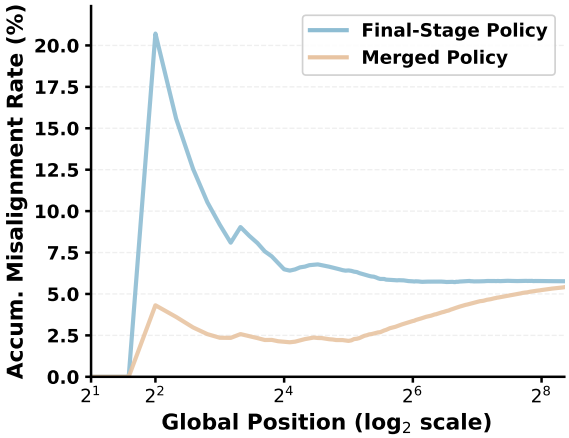

Figure 6: Misalignment rate on sequences generated by the merged model and the low-mode model.

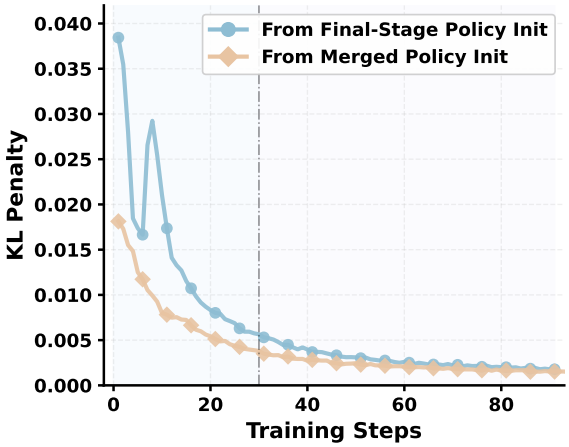

Figure 7: RKL training loss dynamics under different initializations in OPD. 30 steps denotes convergence.

We compare two initialization strategies: the merged model and the final-stage policy from sequential multi-stage RL, as these are the only models trained under all reasoning modes.

We first use the Accumulated Misalignment Rate (AMR), which measures whether the top-1 token probabilities differ between two models. This metric aligns naturally with the on-policy distillation process and provides an intuitive visualization of behavioral differences. Specifically, let $a_t = \arg\max \pi(o_t|q, o_{<t})$ and $\hat{a}_t = \arg\max \pi_{\theta_k}(o_t|q, o_{<t})$. The AMR is defined as:

$$\mathrm{AMR}(\pi, \pi_{\theta_k}) = \mathbb{E}_{q\sim\mathcal{D}, o\sim\pi(\cdot|q)} \left[\frac{1}{|o|} \sum_{t=1}^{|o|} \mathbb{I}(a_t \neq \hat{a}_t) \right] \tag{19}$$

As shown in Figure 6, this merged initialization significantly reduces the initial misalignment gap, providing a more stable starting point and accelerating the convergence of subsequent RKL distillation.

We further examine the training performance under different initialization, as shown in Figure 7, merged model consistently outperforms final stage policy by a lower per token RKL penalty, demonstrating the effectiveness of such initialization.

C Training Details

We build on the VeRL codebase (Sheng et al., 2024) with RLVR. Following the DAPO recipe (Yu et al., 2025), which allows for higher clipping—a practice that has been widely adopted in RL training, we also use rejection sampling as a further ensurance for training stability.

Training Configuration. We maintain consistent hyperparameters throughout the exploration RL stage, using a batch size of 256, rejection pools of 512, and a learning rate of 1×10^{-6} without warm-up. For generation, we apply standard rollout settings: temperature 1, `top-p` 1, `top-k` -1, and 8 rollouts per problem. We train `DeepSeek-Distill-Qwen-1.5B` using the DAPO-Math-17K dataset (Yu et al., 2025), while `Qwen3-4B-2507-Thinking` and `Nemotron-7B` are trained on the Polaris-53K (An et al., 2025).

For exploitation, we train OPD with a batch size of 1024, single generation per prompt, and a token-wise KL penalty coefficient of 1.0.

Context Window Scheduling. We implement an expansion-compression loop strategy to dynamically manage context windows in Exploration stage RL. Specifically, the maximum output length follows a schedule of 32K $\rightarrow$ 16K $\rightarrow$ 8K $\rightarrow$ 4K for `Qwen3-4B-2507-Thinking` and `Openmath-Nemotron-7B`, and 16K $\rightarrow$ 8K $\rightarrow$ 4K $\rightarrow$ 2K for `DeepSeek-Distill-Qwen-1.5B`. In both cases, the first stage of the context window serves as the expansion phase.

D System Prompts For Each Reasoning Mode

To build a semantically meaningful guidance for discrete mode separated reasoning behavior, we follow (Yang et al., 2025b) and generalized it to four modes, where each one of them represents semactically aligned expected reasoning behavior under corresponding mode.

/ Low Mode */*

You have extremely limited time to think and respond to the user's query. Every additional second of processing and reasoning incurs a significant resource cost, which could affect efficiency and effectiveness. Your task is to prioritize speed without sacrificing essential clarity or accuracy. Provide the most direct and concise answer possible. Avoid unnecessary steps, reflections, verification, or refinements UNLESS ABSOLUTELY NECESSARY. Your primary goal is to deliver a quick, clear and correct response.

/ Mid Mode */*

You have sufficient time to think and respond to the user's query, allowing for a more thoughtful and in-depth answer. However, be aware that the longer you take to reason and process, the greater the associated resource costs and potential consequences. While you should not rush, aim to balance the depth of your reasoning with efficiency. Prioritize providing a well-thought-out response, but do not overextend your thinking if the answer can be provided with a reasonable level of analysis. Use your reasoning time wisely, focusing on what is essential for delivering an accurate response without unnecessary delays and overthinking.

/ High Mode */*

You have a substantial amount of time to think and respond. The pressure on reasoning cost is low, allowing you to prioritize accuracy and robustness over speed. You should verify your key steps and consider alternative interpretations to ensure your answer is reliable. While you are encouraged to think deeply, aim for a confident solution rather than an endless exploration. If you encounter ambiguity, take the time to resolve it, but remain focused on deriving the correct conclusion efficiently within a generous budget.

/ Xhigh Mode */*

You have unlimited time to think and respond. There are absolutely no constraints on reasoning length or associated costs. Your singular goal is to achieve the highest possible level of accuracy and certainty. You must exhaustively explore the problem space, considering all edge cases, theoretical nuances, and alternative methodologies. Engage in rigorous, iterative self-correction and deep verification to eliminate even the slightest possibility of error. Do not settle for a 'likely' correct answer; explore every angle until you are certain. Perfection is the only standard.

Table 3: System prompts used for triggering Low, Mid, High, Xhigh modes

Models	Mathematics						Multi-Task				Overall	
	AIME 24		AIME 25		Beyond AIME		GPQA		MMLU			
	avg@32	Toks	avg@32	Toks	avg@8	Toks	avg@8	Toks	avg@1	Toks	Avg.	Toks
DeepSeek-Distill-Qwen-1.5B												
Initial Model	28.6	18.8	22.8	18.5	10.1	18.6	28.0	12.7	35.5	7.2	25.0	15.2
- Teachers Low	28.3	2.6	19.0	2.0	11.0	2.0	34.3	1.8	37.6	1.4	26.0	2.0
- Teachers Mid	35.2	4.9	26.4	4.5	10.4	3.6	32.9	2.9	38.2	2.2	28.6	3.6
- Teachers High	40.0	7.0	29.8	6.7	14.0	6.3	39.8	4.8	40.7	3.8	32.9	5.7
- Teachers Xhigh	42.5	10.7	32.9	11.0	16.0	11.7	39.5	8.3	40.0	7.7	34.2	9.9
Qwen3-4B-2507-Thinking												
Initial Model	81.6	20.1	77.1	18.3	52.4	27.6	66.1	8.8	74.4	4.4	70.3	15.8
- Teachers Low	64.4	5.0	54.3	5.5	36.6	5.2	57.3	1.9	71.5	0.9	56.8	3.7
- Teachers Mid	66.3	5.9	57.7	6.4	40.0	6.4	63.8	3.1	72.3	1.9	60.0	4.7
- Teachers High	75.1	9.4	69.1	10.6	46.0	11.7	65.4	4.9	73.4	3.0	65.8	7.9
- Teachers Xhigh	83.2	16.8	78.2	19.0	52.7	22.7	67.1	7.9	74.1	5.8	71.1	14.4
Openmath-Nemotron-7B												
Initial Model	71.6	11.9	59.9	13.7	41.8	13.6	32.5	8.5	42.4	5.7	49.6	10.7
- Teachers Low	55.2	4.5	33.1	4.5	29.8	3.7	29.3	2.2	39.2	1.7	37.3	3.3
- Teachers Mid	67.1	6.7	44.5	7.6	36.3	6.6	32.4	4.4	41.2	3.0	44.3	5.7
- Teachers High	74.3	11.1	62.1	12.9	41.9	12.1	30.9	7.9	42.6	5.3	50.4	9.9
- Teachers Xhigh	75.9	13.2	65.1	14.9	43.8	15.0	33.7	9.8	43.1	6.5	52.3	11.9

Table 4: Comparison of reasoning modes of Teachers across various reasoning tasks.

Setting	AIME 24		AIME 25		Beyond AIME		GPQA		MMLU		Overall	
	avg@32	Toks	avg@32	Toks	avg@8	Toks	avg@8	Toks	avg@1	Toks	Avg.	Toks
DeepSeek-Distill-Qwen-1.5B												
Prompt Steering Low	26.1	15.1	21.9	14.2	8.5	13.8	28.7	8.4	34.5	5.4	23.9	11.4
Prompt Steering Mid	26.8	15.5	22.6	14.9	10.0	15.0	28.6	9.9	35.3	6.3	24.7	12.3
Prompt Steering High	28.4	16.2	22.5	15.9	10.5	16.2	28.9	11.4	35.3	6.8	25.1	13.3
Prompt Steering Xhigh	28.2	16.1	20.0	15.7	10.0	15.9	29.0	12.3	35.1	7.0	24.5	13.4
Qwen3-4B-2507-Thinking												
Prompt Steering Low	79.3	16.6	77.0	17.1	47.1	20.4	63.6	7.3	73.1	3.8	68.0	13.0
Prompt Steering Mid	79.0	19.9	77.3	20.4	54.4	26.8	66.7	8.5	72.8	4.6	70.0	16.0
Prompt Steering High	81.1	20.4	77.7	20.8	53.5	27.7	65.4	9.1	74.1	5.0	70.4	16.6
Prompt Steering Xhigh	81.9	21.1	79.1	21.9	53.6	28.8	66.9	9.9	73.7	6.2	71.0	17.6
Openmath-Nemotron-7B												
Prompt Steering Low	70.8	11.3	57.5	13.7	41.3	13.1	31.9	8.2	42.6	5.5	48.8	10.4
Prompt Steering Mid	70.0	11.8	57.5	13.1	40.3	13.6	33.4	8.2	42.4	5.7	48.7	10.5
Prompt Steering High	73.3	11.1	55.8	13.3	41.8	13.5	34.4	8.6	42.5	5.7	49.6	10.4
Prompt Steering Xhigh	69.1	12.1	61.0	13.1	42.6	13.3	34.1	8.5	42.3	5.8	49.8	10.6

Table 5: Prompt steering results on initial models across various reasoning tasks.

E Additional Evaluation Results

E.1 Frontier Teacher Model Performance

We provide the evaluation results for each teacher model from exploration stage in Table 4.

E.2 Prompt Steering

We evaluate prompt-based steering as a baseline to assess whether controllable behavior can be achieved through simple system prompts (e.g., instructing a vanilla LRM to be concise). This comparison quantifies the benefits of our specialized training (Table 5). Results show limited controllability: performance remains close to the default across effort levels, with no consistent mode-dependent scaling. Output length varies slightly, but accuracy changes are modest and often non-monotonic, indicating that ORBIT's clear mode separation arises from specialized training rather than prompt steering.