When and Why Does Unsupervised RL Succeed in Mathematical Reasoning? A Manifold Envelopment Perspective

Anonymous ACL submission

Abstract

Although outcome-based reinforcement learning (RL) significantly advances the mathematical reasoning capabilities of Large Language Models (LLMs), its reliance on computationally expensive ground-truth annotations imposes a severe scalability bottleneck. Unsupervised RL guided by intrinsic rewards offers a scalable alternative, yet it suffers from opaque training dynamics and catastrophic instability, such as policy collapse and reward hacking. In this paper, we **first** design and evaluate a suite of intrinsic rewards that explicitly enforce concise and certain generation. **Second**, to discover the boundaries of this approach, we test base models across a spectrum of intrinsic reasoning capabilities, revealing how a model's foundational logical prior dictates its success or failure. **Finally**, to demystify why certain configurations stabilize while others collapse, we introduce a novel geometric diagnostic lens, showing successful cases are **enveloped by manifolds**. **Ultimately**, our work goes beyond merely demonstrating that enforcing concise and certain responses successfully boosts mathematical reasoning; we reveal *when* this unsupervised approach breaks down and geometrically diagnose *why*.

1 Introduction

Recent advances in mathematical reasoning for Large Language Models (LLMs) rely on outcome-based reinforcement learning frameworks, such as Group Relative Policy Optimization (GRPO; Shao et al., 2024), often categorized as Reinforcement Learning with Verifiable Rewards (RLVR). These paradigms operate on verifiable supervision: models are trained by generating multiple solution paths for a given prompt, and policy updates are guided by reward signals derived from ground-truth labels. This dependence on ground truth imposes a scalability bottleneck, as verifiable answers are frequently unavailable or computationally expensive to obtain (Lightman et al., 2024).

Unsupervised RL, which optimizes models using intrinsic rewards such as entropy minimization (Prabhudesai et al., 2025; Agarwal et al., 2025) or self-consistency (Zhang et al., 2025; Zuo et al., 2025) without ground-truth labels, provides a scalable alternative. However, deploying unsupervised RL for mathematical tasks presents challenges. To prevent optimization instability such as reward hacking, a phenomenon where the model exploits loopholes in the reward function to achieve high scores without fulfilling the intended task (Gao et al., 2022), the design of rewards often requires adaptation to specific datasets or models. Studies on unsupervised RL generally treats the optimization process as a black box, omitting the explanation of *why* the policy succeeds or collapses during training. The internal mechanisms of unsupervised rewards require further investigation.

Our primary research objective is to uncover the underlying mechanisms and boundaries of unsupervised mathematical reasoning in LLMs. To systematically unpack this, we try to answer the following questions: **(a)** *Will penalizing a model for being verbose and uncertain naturally force it to discover correct mathematical reasoning?* **(b)** *How do factors like the base model's inherent capacity define the boundaries where this unsupervised method inevitably fails?* **(c)** *What analytical lens can effectively capture the underlying dynamics to distinguish successful unsupervised settings from failed ones?*

To answer question **(a)**, as illustrated in Figure 1 (a), we systematically design and evaluate a series of intrinsic rewards based on two penalty dimensions: uncertainty and response length. We compare five specific methods: Shannon Entropy (Ent) (Agarwal et al., 2025), Cumulative Rényi Entropy (CH2), Averaged Shannon Entropy (AvgEnt) (Prabhudesai et al., 2025), Collision Proba-

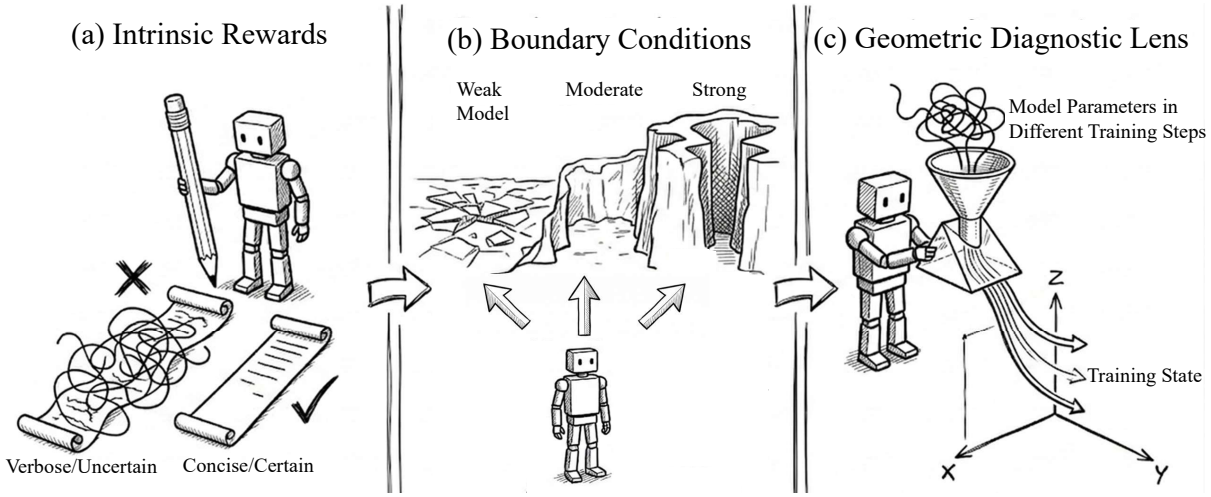


Figure 1: **Overview of Research Pipeline. (a)** Design Rewards: enforce concise and certain. **(b)** Find Boundary Conditions: evaluate across models with different reasoning abilities. **(c)** Find Diagnostic Lens: DTW clustering to 3D phase space.

bility (CP), and Length Penalty (LP). Among them, CH2, CP, and LP are newly designed methods introduced to cover different penalties, demonstrating the generality of the subsequent mechanistic analysis.

To answer question **(b)**, as illustrated in Figure 1 (b), we evaluate different model families across a spectrum of intrinsic logical reasoning priors from Llama to Qwen. We establish a developmental progression by systematically testing different configurations across base models with weak, moderate, and high capabilities and their corresponding reward methods. This design allows us to observe how a model's evolutionary stage dictates its training stability, failure modes, and ultimate success.

To answer question **(c)**, as illustrated in Figure 1 (c), we introduce a novel two-step analytical framework. **(Step 1)** We cluster tokens based on how their average entropy evolves throughout the training process. We apply Dynamic Time Warping (DTW) (Berndt and Clifford, 1994) for this clustering because these token-level entropy trajectories naturally vary in length. **(Step 2)** We project these high-dimensional trajectories into a measurable 3D phase space where each semantic cluster or entropy level corresponds to one dimension, providing a unified geometric lens to diagnose stability.

The results of **Step 1** indicate that across different training configurations, the temporal trends of token entropy group into distinct entropy levels, each corresponds to a semantic cluster: a low-entropy "Execution State" (e.g., *1, +, by*), a medium-entropy "Logic State" (e.g., *Let, So, Therefore*), and a high-entropy "Thinking State" (e.g., *perhaps, maybe, suppose*). In **Step 2**, we apply a geometric diagnostic lens to analyze the training trajectories on the manifold within this projected space. We observe that successful training settings result in trajectories that are tightly enveloped on a well-defined manifold, demonstrating ordered and structured exploration. In contrast, failing settings are either loosely enveloped or display chaotic wandering across the geometric space.

Our main contributions are summarized as follows:

- **Systematic Evaluation of Unsupervised Rewards:** Addressing question **(a)**, we design and compare a suite of rewards including our novel designs (LP, CH2 and CP) based on response length and predictive uncertainty. We empirically demonstrate how explicitly penalizing verbose and uncertain outputs can naturally force a model to discover correct mathematical reasoning without ground-truth labels.

- **Identification of Boundary Conditions:** Addressing question **(b)**, we establish a comparative evaluation pipeline across base models with weak, moderate, and highly capable foundational reasoning skills. This developmental progression reveals how a model's intrinsic logical prior fundamentally dictates its training stability, failure modes, and ultimate success under unsupervised RL.

- **Novel Diagnostic Lens for Training Dynamics:** Addressing question **(c)**, we intro-

Method	Uncertainty	Length
Ent	Penalty	Strong penalty
AvgEnt	Penalty	None
LP*	None	Strong penalty
CH2*	Penalty	Weak penalty
CP*	Penalty	Encourage (Confusing)

Table 1: **Formulation of the unsupervised rewards.** We decompose each reward into two orthogonal dimensions: uncertainty penalization and length penalization. Methods marked with an asterisk (*) are newly designed formulations introduced in this work to complete the design matrix across both dimensions. Ent (Agarwal et al., 2025) and AvgEnt (Prabhudesai et al., 2025) are previous methods.

duce a two-step analytical framework using DTW clustering and 3D phase space projection. This lens uncovers a consistent mapping between temporal entropy and cognitive states (Execution, Logic, and Thinking), and identifies the "Manifold Envelopment" phenomenon as a geometric metric to physically distinguish successful, stable reasoning from chaotic policy collapse.

2 Training Methods

During the RL training phase, the language model acts as a policy to generate a reasoning path token-by-token in an auto-regressive manner. This generation process occurs during the rollout stage of training, which practically functions as an inference step to collect trajectories. Once the full reasoning path is completed, a reward is assigned to update the model's parameters. Supervised RL for mathematical reasoning evaluates the final outcome against a ground-truth label. In contrast, unsupervised RL operates entirely without access to correct answers. Instead, the reward R is calculated at the response sequence level, derived purely from the intrinsic properties of the generated trajectory. Specifically, for a generated reasoning response of length T, let $p_t(v)$ denote the predicted probability of token v from the vocabulary at generation step t. We detail the mechanical intent behind each unsupervised formulation as follows.

2.1 Reward Formulation

Building on the effectiveness of *Ent* identified by Agarwal et al. (2025), we decompose it into two core optimization factors: **uncertainty penalization** and **length penalization**. Accordingly, we propose five reward formulations that systematically isolate and vary these dimensions to provide a comprehensive coverage across the penalty spectrum, as detailed in Table 1. These diverse formulations serve to verify the generalizability of our diagnostic framework across various unsupervised objectives, while simultaneously allowing us to identify the most effective penalty mechanisms for eliciting mathematical reasoning. Furthermore, this setup functions as an extensive ablation study, providing empirical insights into how rewards across different dimensions and intensities impact the model's reasoning performance.

Shannon Entropy (Ent). Ent represents **the most aggressive joint optimization**. The reward employs the exact vocabulary-level Shannon entropy:

$$R_{\text{Ent}} = -\sum_{t=1}^{T} \mathcal{H}(p_t) = \sum_{t=1}^{T} \sum_{v \in \mathcal{V}} p_t(v) \log(p_t(v))$$

Because token entropy $\mathcal{H}(p_t)$ is strictly non-negative, the cumulative sum monotonically decreases as the sequence elongates. The model is pressured to become certain and concise at the same time.

Averaged Shannon Entropy (AvgEnt). To completely remove the implicit length penalty found in cumulative metrics, we formulate AvgEnt as

$$R_{\text{AvgEnt}} = -\frac{1}{T} \sum_{t=1}^{T} \mathcal{H}(p_t)$$

By averaging the entropy across the entire sequence, the pressure to output short responses is eliminated. This formulation **isolates uncertainty penalization**.

Length Penalty (LP). To isolate the effect of sequence length from certainty, we establish LP as a pure length punishment.

$$R_{\text{LP}} = -T/T_{\max}$$

where $T_{\max}$ denotes the maximum response length. This metric applies an explicit penalty based solely on the token count, stripping away all feedback regarding the model's internal confidence. It serves to answer whether forcing brevity alone can induce mathematical reasoning.

Cumulative Rényi Entropy (CH2). Representing the formulation based on Second-order Rényi Entropy, CH2 is defined as

$$R_{\mathrm{CH}_2} = \sum_{t=1}^{T} \log \left(\sum_{v \in \mathcal{V}} p_t(v)^2 \right)$$

Although the reward looks similar to Ent, its mechanical effect diverges from Ent. It provides a **weaker length penalty** by annihilating the "long-tail tax." In Ent, Shannon entropy evaluates $-p \log p$ over the entire vocabulary ($|\mathcal{V}| \approx 10^5$). The logarithmic term aggressively amplifies the impact of the remaining probability mass distributed across the long tail ($\lim_{p_t(v) \to 0} \log p_t(v) \to -\infty$). In contrast, CH2's squaring operation mathematically obliterates the infinitesimal probabilities in the long tail ($\lim_{p_t(v) \to 0} p_t(v)^2 \to 0$). Consequently, the term is **dominated by the top-1 probability**.

Collision Probability (CP). Representing Collision Probability, CP introduces a mathematically inverted length dynamic formulated as

$$R_{\mathrm{CP}} = \sum_{t=1}^{T} \sum_{v \in \mathcal{V}} p_t(v)^2$$

The reward is strictly positive ($p_t^2 > 0$). Consequently, adding more tokens increases the total cumulative reward. This creates a tension: the model tends to maximize certainty, but is simultaneously encouraged to generate verbose responses to accumulate more reward points. CP is designed to stress-test the training dynamics and observe which factor (certainty vs. verbosity) dominates when they are placed in conflict. Furthermore, it also represents a confusing situation to disentangle the impact of the reward signal from the base model's inherent capabilities.

2.2 Policy Optimization

To translate our unsupervised intrinsic rewards into policy updates without the memory overhead of a separate critic network, we employ GRPO (Shao et al., 2024). GRPO estimates the baseline directly from the scores of multiple rollouts generated for the same prompt, making it particularly well-suited for our scalable unsupervised framework.

For a given mathematical problem query q, we sample a group of G response token trajectories $\{o_1, o_2, \ldots, o_G\}$ from the old policy $\pi_{\theta_{old}}$. For each trajectory o_i, we compute its intrinsic scalar reward r_i using one of the formulations defined in Section 2.1.

The advantage A_i for each rollout is then calculated by standardizing the rewards within the sampled group:

$$A_i = \frac{r_i - \mu_{\mathbf{r}}}{\sigma_{\mathbf{r}} + \epsilon_{std}}$$

where $\mu_{\mathbf{r}}$ and $\sigma_{\mathbf{r}}$ are the mean and standard deviation of the group rewards $\mathbf{r} = \{r_1, \ldots, r_G\}$, respectively, and ϵ_{std} is a small constant to prevent division by zero. This relative advantage stabilizes the training by ensuring the model is rewarded only when a trajectory outperforms its own average generation for that specific query.

The policy model π_θ is then optimized by maximizing the objective function of GRPO:

$$\mathcal{J}(\theta) = \mathbb{E}\left[\frac{1}{G} \sum_{i=1}^{G} \left(\min \left(\rho_i(\theta) A_i,\ \mathrm{clip}(\epsilon) A_i \right) \right) \right]$$

where $\rho_i(\theta) = \pi_\theta(o_i|q) / \pi_{\theta_{old}}(o_i|q)$ is the importance sampling ratio, ϵ is the clipping hyperparameter to prevent destructively large policy updates, and π_{ref} is the reference model. Note that we **explicitly exclude the KL divergence term** $\mathbb{D}_{KL}$ across all settings. While typically used to anchor the policy to a reference model and prevent over-optimization (Schulman et al., 2017), the KL penalty functionally overlaps with our entropy-based rewards. Because our primary objective is to mechanistically elucidate the working principles of unsupervised RL, introducing this redundant constraint would obscure the pure effects of the reward signals and hinder accurate theoretical analysis.

3 Experimental Settings

3.1 Models and Datasets

To identify the boundaries of this unsupervised approach, we evaluate a diverse set of open-weights LLMs across a spectrum of intrinsic reasoning capabilities. This cross-model evaluation allows us to observe how foundational model abilities dictate the transition from stable reasoning convergence to catastrophic policy collapse.

We primarily utilize *Qwen3-8B* as our main testbed. Having undergone rigorous post-training, it represents a highly aligned state with robust foundational reasoning capabilities, striking an optimal balance between state-of-the-art performance

at its parameter scale and computational feasibility (Yang et al., 2025). To evaluate generalizability across parameter sizes, we also test *Qwen3-1.7B*. Note that in all our experimental settings, the Qwen3 models are configured to use their native 'think' mode to ensure a fair evaluation of their reasoning potential.

Furthermore, to understand the evolutionary trajectory of reasoning, we need to observe models at different stages of alignment. Because we cannot access the intermediate checkpoints of Qwen's training process, we select *Llama-3.1-8B-Instruct* as a baseline reference, as it has only undergone standard supervised fine-tuning (SFT) for instruction following (Grattafiori et al., 2024). We then compare it with *DeepSeek-R1-Distill-Llama-8B*, which is distilled from the Llama-3 architecture. Through the distillation process, the model leverages additional reasoning data generated by a stronger teacher, which inherently boosts its mathematical capabilities and allows it to represent a higher aligned state similar to Qwen3 (DeepSeek-AI, 2025).

Training Data. Our primary training dataset is **DAPO-Math-17K** (Yu et al., 2025), an open-source dataset specifically designed for RL. We utilize it to ensure that our models are trained on high-quality mathematical tasks. Additionally, to ensure that our observations are not biased toward a specific data distribution, we further train the model on **DeepMath-103K** (He et al., 2025) in Appendix D.

Evaluation Benchmarks. All models are evaluated on a comprehensive suite of standardized benchmarks. Our test suite includes the traditional mathematical benchmarks: **MATH500** (Lightman et al., 2024), **Minerva Math** (Lewkowycz et al., 2022), **OlympiadBench** (He et al., 2024), **AIME24** (Zhang and Math-AI, 2024), and **AMC23**.[1] Furthermore, we integrate **AIME26** (Zhang and Math-AI, 2026) into our test suite. Because AIME26 was released strictly after the release dates of all evaluated models, it serves as the out-of-distribution (OOD) touchstone to verify zero-contamination reasoning improvements.

3.2 Early Stopping

In our experiments, we observe a common phenomenon: all unsupervised RL methods eventually suffer from policy collapse on the validation set as training progresses. Because of this inherent instability, training until convergence is not feasible. Instead, we adopt an early stopping strategy. As long as a method demonstrates a performance improvement over the initial base model, we halt the training at its peak validation accuracy and test the checkpoint.

Conversely, we define a training run as a complete "*collapse*" only if the model's performance continuously degrades from the very beginning of the training process, showing no signs of improvement. For these explicitly collapsed cases, we do not report the detailed numerical metrics in our main results, as the base model represents the best performance.

Other details of the implementation are in the Appendix C.

4 Results

As illustrated in Table 2, our unsupervised reinforcement learning methods except CP demonstrate highly competitive performance. On the Qwen3-1.7B and Qwen3-8B models, the LP method achieves the highest overall accuracy even outperforming supervised baseline S-RL across all benchmarks. Besides, other unsupervised rewards also dramatically outperform base model. Other additional discussions about boundary cases are in Appendix D.

Aligned Stages of Base Models. Interestingly, the base model capability significantly impacts training stability under unsupervised settings. For the DeepSeek-Distill-Llama-8B model, the performance gains are more modest than Qwen series. Although unsupervised methods (particularly LP and Ent) still manage to improve upon the base model, the margin of improvement is significantly smaller than that observed in the Qwen family, and S-RL maintains a clear lead. In contrast, Llama3.1-8B, representing the weakest reasoning ability, suffers from an immediate collapse across all unsupervised methods. According to Section 3.1, this suggests that aligned stages of base models affect the robustness of unsupervised RL.

Reward Formulations. Our controlled variations of the rewards reveal how unsupervised rewards shape reasoning. LP emerges as the most effective unsupervised method across almost all settings. Stripping away the entropy calculation

[1] https://maa.org/math-competitions/

Model	Method	MATH-500			Olympiad			Minerva			AIME 24			AMC 23			AIME 26			AVG		
		P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10
Qwen3-1.7B	Base	70.2	81.4	84.9	30.1	41.6	46.9	30.6	42.0	46.0	6.0	14.2	19.1	47.2	61.4	68.0	7.1	15.5	20.0	42.7	54.0	58.5
	S-RL	78.8	89.5	91.8	39.5	52.0	56.0	35.4	49.2	53.9	17.9	31.0	37.8	56.6	76.0	82.1	12.7	26.0	31.1	51.0	63.3	67.0
	Ent	79.8	90.4	92.4	41.4	54.9	58.8	34.4	48.3	52.9	**22.5**	**42.6**	**49.3**	**61.3**	**81.8**	**88.1**	14.0	27.9	31.6	52.2	**65.1**	**68.7**
	AvgEnt	78.2	89.7	91.9	39.8	52.8	56.7	**36.1**	**50.0**	**54.5**	20.2	38.8	49.1	58.3	80.3	87.9	**16.0**	**34.7**	**42.0**	51.2	64.3	68.1
	LP	**80.0**	90.4	92.4	**42.6**	**55.3**	**59.1**	34.4	47.9	52.4	20.0	34.2	39.3	57.8	79.7	86.8	14.4	28.0	33.6	**52.7**	65.0	68.5
	CH2	79.5	**90.6**	**93.2**	41.3	53.5	57.3	35.2	48.6	52.7	20.6	35.8	40.4	59.5	78.7	85.6	14.8	29.8	35.0	52.1	64.5	68.1
	CP	*Collapse* (×)																				
Qwen3-8B	Base	67.9	80.5	84.4	28.4	41.2	46.7	29.8	39.9	44.8	7.7	19.9	23.1	41.7	64.1	73.6	9.2	16.6	22.1	41.0	53.4	58.3
	S-RL	88.4	93.8	95.0	51.7	60.1	63.4	46.0	55.7	58.3	36.2	52.7	**60.4**	76.2	87.9	89.1	36.2	48.8	50.0	62.6	70.6	73.1
	Ent	86.6	93.1	94.5	49.9	58.2	61.0	43.9	54.4	58.2	31.7	48.0	53.5	74.1	89.4	91.5	34.2	49.8	**54.6**	60.7	69.3	71.9
	AvgEnt	81.5	89.5	91.2	43.9	54.0	57.2	44.2	54.4	57.9	25.4	38.6	42.9	62.0	81.6	88.4	19.2	36.3	43.1	55.7	65.6	68.6
	LP	**88.7**	**94.2**	**95.2**	**52.3**	**61.1**	**64.2**	**46.1**	**56.2**	**59.1**	**39.2**	**55.8**	59.1	**78.8**	**93.0**	**94.9**	**36.7**	**51.1**	54.1	**63.1**	**71.5**	**73.8**
	CH2	86.0	92.9	94.3	49.0	59.2	62.5	44.0	53.4	56.4	30.8	46.4	50.4	73.1	88.9	91.8	28.1	39.9	42.5	59.9	69.2	71.8
	CP	*Collapse* (×)																				
Llama3.1-8B	Base	45.5	**69.4**	**77.2**	13.2	29.8	38.2	21.4	39.8	47.1	**5.2**	**14.3**	**20.8**	22.7	**51.6**	**66.4**	0.4	2.1	**4.2**	24.9	44.1	52.1
	S-RL	**46.3**	69.0	76.4	**14.1**	**31.0**	**38.8**	**25.7**	**45.5**	**52.3**	4.6	13.2	18.7	**23.6**	46.8	56.1	**1.2**	**3.1**	3.3	**26.4**	**45.4**	**52.7**
	Unsupervised	*Collapse* (×)																				
DeepSeek-Distill Llama-8B	Base	69.4	87.1	90.8	35.7	52.9	58.8	22.7	40.2	47.8	16.5	30.3	37.9	50.8	80.6	86.9	14.4	27.4	32.7	43.9	61.5	67.0
	S-RL	**72.7**	**90.2**	**93.7**	**39.3**	**56.9**	**62.4**	**24.4**	**42.7**	**49.9**	**24.4**	**44.9**	**54.9**	**59.4**	**86.9**	**91.1**	**22.5**	**40.0**	**44.6**	**47.4**	**65.4**	**70.6**
	Ent†	70.0	88.1	91.9	36.4	54.6	60.6	21.8	39.9	47.2	18.1	37.9	47.5	53.4	83.8	88.5	17.3	31.2	36.1	44.4	62.8	68.4
	AvgEnt†	69.3	86.6	90.4	35.6	52.6	58.5	23.2	40.6	48.0	16.7	36.8	45.2	53.8	81.7	87.5	15.2	27.2	29.6	44.0	61.4	66.9
	LP†	70.5	88.5	92.5	36.6	53.9	59.5	22.2	40.7	48.3	21.7	43.1	52.8	50.8	81.4	87.0	17.5	33.8	39.4	44.7	62.9	68.4
	CH2†	69.5	86.8	90.6	36.7	54.3	59.9	21.8	38.6	46.1	20.6	40.9	50.3	54.7	83.7	88.7	17.3	30.4	34.0	44.5	62.1	67.5
	CP	*Collapse* (×)																				

Table 2: **Main Results on Mathematical Reasoning Benchmarks.** Performance comparison across models, reward formulations, and training settings. All training is conducted on the DAPO-Math-17K dataset. *S-RL* refers to Supervised RL. **Bold** denotes the overall best performance, and underlined indicates the best result among unsupervised methods within each model. *P@k* means Pass@k. † indicates a *semi-collapse* phenomenon. For the DeepSeek-Distill model, these unsupervised methods experience a very brief initial period of improvement where the peak performance slightly exceeds the base model, but the training rapidly collapses shortly afterward.

and strictly punishing the length is sufficient to force the model into generating concise, logical steps. Ent and CH2 also show strong gains over the Base model. However, they slightly lag behind pure LP. AvgEnt removes the implicit length penalty by averaging the entropy drastically reduces performance. CP reverses the penalty to encourage longer responses universally results in a complete collapse across all models, which means this reward's length penalty strength is stronger than uncertainty penalty. These prove our theoretical claim: penalizing uncertainty without strictly bounding the structural length is not strong enough. Length penalty is more effective than uncertainty penalty.

5 Why it Doesn't Collapse: Manifold Envelopment

To demystify the internal mechanisms of these unsupervised rewards, we introduce a two-step analytical framework using DTW clustering and 3D phase space projection.

Step1. Temporal Clustering of Token Entropies. Instead of analyzing static entropy values, we focus on the *temporal dynamics*: how the entropies of specific tokens evolves throughout the training process. Because a model's generated response for the same prompt changes across training steps, we use the (prompt, token) pair as a unique anchor to track the exact entropy trajectory of each token over time. We then apply Soft-DTW (Cuturi and Blondel, 2018) to group these trajectories based on the shape of their evolution rather than their absolute values. Furthermore, while previous work (Wang et al., 2025) simplifies reasoning into binary high and low entropy states, we deliberately set the number of clusters to $K = 3$. This allows us to capture the crucial intermediate phase of logical connectives and transitions that bridge direct calculations and complex decision-making. Detailed formalizations of this algorithm are provided in Appendix A.

Bridging Trajectories and Semantics. As illustrated in Table 3, applying Time-Series K-means consistently yields three distinct clusters. Remarkably, **across all our experimental settings**, the entropy centroids always stratify into a clear low, medium, and high pattern. Crucially, these three entropy levels naturally align with the semantic roles of the tokens during mathematical reasoning.

Entropy Cluster	Semantic	Representative Tokens (Top Frequency)	Characteristics
Low (Execution)	Arithmetic	*Qwen*: 1, 2, 3, 0, +, - *DeepSeek*: +, 2, 4, -, 3, 1 *Llama*: 4, 2, =, +, 1, 3	Operators and concrete numerical digits.
	Syntax	*Qwen*: by, 't, to, be, (,) *DeepSeek*: 't,), be, by, of, to *Llama*: of, {, }{, \, frac, $	Structural formatting and LaTeX syntax.
Medium (Logic)	Formulation	*Qwen*: Let, The, this, it *DeepSeek*: Let, this, here, each *Llama*: need, let, given, find, solve	Establishing variables and conditions.
	Transitions	*Qwen*: Wait, But, ?, Therefore, So *DeepSeek*: ?, but, should, So, Wait *Llama*: However, To, must, also, Now	Step-by-step reasoning deductions.
High (Thinking)	Exploration	*Qwen*: perhaps, maybe, suppose, Hmm *DeepSeek*: maybe, perhaps, wait, Hmm, think *Llama*: First, start, break, analyze	Macro-planning and expressing uncertainty.
	Verification	*Qwen*: confirm, verify, actually, If, assume *DeepSeek*: consider, double, Because, Or *Llama*: denote, determine, considering	Halting to verify or initiating branch paths.

Table 3: **Semantic Interpretation of Entropy Clusters Across Models.** We extract the representative tokens for the Qwen, DeepSeek-Distill, and Llama. To ensure objective categorization, these tokens are rigorously filtered by retaining only the core samples (the 50% closest to the respective cluster centroids in the DTW space) and ranked strictly by their occurrence frequency. Across all three model families, these high-frequency tokens consistently align with the semantic functions of Execution, Logic, and Thinking.

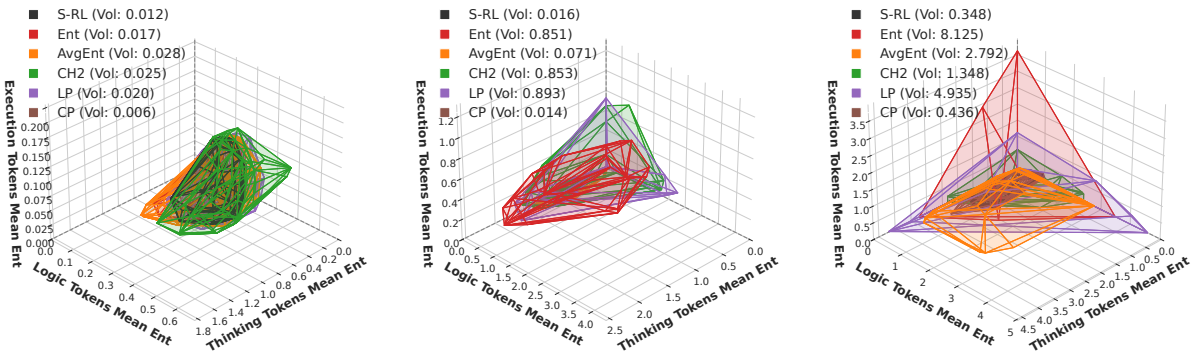
Specifically, they correspond to the *Execution* (low entropy), *Logic* (medium entropy), and *Thinking* (high entropy) phases. Detailed visualizations of these clustering trajectories across different models are provided in Appendix B.

Step2. 3D phase space projection By treating the entropy values of these three distinct cognitive states as independent coordinate axes, we can project the high-dimensional, opaque training dynamics into a measurable 3D phase space. This mapping represents a **dimensionality reduction process**. We compress the high-dimensional model parameters and internal states into the average entropy of three semantic clusters. This reduction is effective in unsupervised RL because entropy serves as the primary optimization objective. Consequently, the three-dimensional phase space captures the most significant behavioral shifts during training. This coordinate system allows us to visualize and measure how the policy navigates the reasoning manifold.

Exploration Boundaries. By calculating the Convex Hull of these temporal trajectories across the validation set, we can quantify the model's exploration boundaries. Visualizing these boundaries in Figure 2 reveals three distinct spatiotemporal phenomena: *Strong Manifold Constraints* (Success), *Exploration Stagnation* (Failure Type I), and *Weak Constraints* (Failure Type II). To ensure a fair comparison, we dynamically truncate these trajectories as declared in Section 3.2. Chronologically, the accuracy of collapsed configurations drops to zero significantly earlier than the performance peak of successful runs. Therefore, the trajectory of a successful run naturally stops at its validation peak. Conversely, the trajectory of a collapsed model terminates exactly at this premature zero-accuracy step. This precise truncation explicitly excludes the meaningless post-collapse stages. Additional visualizations demonstrating this consistent pattern across other configurations are provided in Appendix E.

Success: Strong Manifold Constraints. A highly capable base model natively restricts the reasoning space. For example, Qwen3-8B under LP successfully maintains an optimal exploration volume. Figure 2 (a) visually confirms this tightly bounded manifold.

Failure Type I: Exploration Stagnation. Qwen3-8B trained with CP suffers from severe *Exploration Stagnation*. The exploration manifold for CP shrinks drastically to a negligible volume of 0.006. Figure 2 (a) clearly illustrates this trapped phase space. The model completely loses its



(a) Qwen3-8B Boundaries (b) DS-Distill Boundaries (c) Llama3.1-8B Boundaries

Figure 2: **Visualization of 3D Semantic Manifold Boundaries.** The three axes define the phase space corresponding to the mean entropy of the *Thinking*, *Logic*, and *Execution* semantic clusters, as established in Section 5. **(a-c) Exploration Manifolds:** The translucent polygons denote the 3D convex hulls encompassing all aggregated trajectory points for specific training methods. The legend identifies the training method and provides the calculated volume of its corresponding hull. *DS-Distill* denotes DeepSeek-R1-Distill-Llama-8B.

generative diversity.

Failure Type II: Weak Constraints. Model capability progressively weakens from Qwen3-8B down to Llama3.1-8B. This decreasing capacity triggers a severe loss of boundary control. DS-Distill represents a transitional state with moderately expanded boundaries in Figure 2 (b). We observe the absolute opposite extreme when applying Ent to the weakest Llama3.1-8B model. Figure 2 (c) reveals a pathological **Manifold Explosion**. The Ent convex hull abnormally expands to an enormous volume of 8.125. This massive space indicates a complete loss of constraints instead of healthy diversity.

6 Related Work

While entropy was traditionally employed as a regularization term in standard supervised PPO (Schulman et al., 2017), recent unsupervised RL approaches have adopted uncertainty as a primary optimization signal. In this domain, existing literature generally falls into two main directions. The first direction focuses on self-consistency and semantic clustering, such as Test-Time RL (TTRL; Zuo et al., 2025) and Entropy Minimized Policy Optimization (EMPO; Zhang et al., 2025), which incentivize models to converge on consistent reasoning pathways. The second, emerging direction relies on direct token-level entropy minimization. For instance, RENT (Prabhudesai et al., 2025) utilizes negative token entropy as an intrinsic reward, and Agarwal et al. (2025) comprehensively demonstrates the effectiveness of entropy minimization across various fine-tuning and RL stages.

Our work distinguishes itself from these studies through two primary dimensions: **(a) Algorithmic Decoupling.** By decoupling uncertainty from length penalization within a systematic experimental matrix, we reveal that the implicit length penalty serves as another catalyst for mathematical reasoning. **(b) Mechanistic Interpretability.** Rather than treating the unsupervised RL training process as a black box, we elucidate the internal mechanisms driving this capability emergence. The geometric diagnostic lens allows us to identifies when a model successfully envelops robust reasoning manifolds and why specific optimization trajectories deteriorate into policy collapse.

7 Conclusion

In this work, we presented a comprehensive investigation into unsupervised RL for eliciting latent reasoning capabilities in LLMs. Moving beyond uncertainty penalty, we newly introduce length penalty. To uncover the underlying mechanisms of these rewards, we propose a novel interpretability framework. By applying DTW clustering to token-level entropy trajectories over the training progress, we successfully disentangled the response tokens into three distinct semantic phases. Our geometric analysis revealed the difference between successful cases and failures. In conclusion, our findings bridge the gap between empirical algorithmic design and mechanistic understanding in unsupervised LLM reasoning.

Limitations

While this work provides a novel geometric perspective on unsupervised RL, several limitations warrant acknowledgment and outline avenues for future research.

Scale. Due to the substantial computational overhead required by unsupervised RL, our evaluation is currently restricted to base models up to 8B parameters.

Theoretical Gaps in Training Dynamics. Our DTW clustering shows that the entropy centroids of different cognitive states (Execution, Logic, Thinking) remain stable throughout the entire training process. We hypothesize this stability is an intrinsic structural bias inherited from the base model's pre-training, but a formal optimization proof is left for future work.

Mathematical Formalization of the Manifold. In this study, the "Reasoning Manifold" and its topological barriers are primarily introduced as geometric metaphors grounded in empirical thermodynamic metrics. We do not provide a strict formulation of these boundaries. Formalizing the exact mathematical representation of these reasoning manifolds and proving bounds on the policy's KL divergence or gradient norm when confined by these barriers remain a next step for the interpretability.

References

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. 2025. The unreasonable effectiveness of entropy minimization in llm reasoning. *Preprint*, arXiv:2505.15134.

Donald J. Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, page 359–370. AAAI Press.

Marco Cuturi and Mathieu Blondel. 2018. Soft-dtw: a differentiable loss function for time-series. *Preprint*, arXiv:1703.01541.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Leo Gao, John Schulman, and Jacob Hilton. 2022. Scaling laws for reward model overoptimization. *Preprint*, arXiv:2210.10760.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng 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. *Preprint*, arXiv:2402.14008.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *Preprint*, arXiv:2206.14858.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. 2025. Maximizing confidence alone improves reasoning. *Preprint*, arXiv:2505.22660.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. 2025. Right question is already half the answer: Fully unsupervised LLM reasoning incentivization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yifan Zhang and Team Math-AI. 2024. American invitational mathematics examination (aime) 2024.

Yifan Zhang and Team Math-AI. 2026. American invitational mathematics examination (aime) 2026.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. 2025. TTRL: Test-time reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A Detailed Algorithm for Trajectory Construction

To provide a rigorous definition of the methodology described in Section 5, we formalize the data collection, temporal anchoring, and clustering process in Algorithm 1.

The core challenge is that the generated response for the same prompt p changes across training steps t (e.g., a model might output "Hmm, 5" at step 10, but simply "5" at step 50). To solve this, we use the tuple (p, y) as a unique semantic anchor, mapping the discrete token y back to its prompt p. This allows us to track the exact temporal evolution of the token's uncertainty (entropy) regardless of when it appears in the generation sequence, ultimately forming the trajectory $\mathcal{T}_{p,y}$ for DTW clustering.

Data Construction. We track the generation process on the validation set every 5 training steps. A trajectory is defined as a sequence of entropy values anchored to a specific (prompt, token) pair. Let $\mathcal{T}_{i,j} = [h_{t_1,j}, h_{t_2,j}, \ldots, h_{t_M,j}]$ denote the entropy trajectory for the i-th token in j-th prompt's response. The endpoint of training corresponds to the *effective convergence point*, defined as: (1) the step of peak validation accuracy for unsupervised methods (early stopping), (2) the onset of the accuracy plateau for supervised baselines, or (3) the point of collapse (0% accuracy) for failed runs.

Clustering Algorithm. We employ **Time-Series K-means** with a Soft-DTW metric to cluster these trajectories. Unlike Euclidean distance, DTW aligns sequences that may vary in phase or speed, allowing us to group tokens based on the *shape* of their entropy evolution (e.g., "rising" vs. "converging") rather than their absolute values at specific steps.

Cluster Selection. While Wang et al. (2025) simplifies the reasoning process into a binary classification of high- and low-entropy tokens, we hypothesize that mathematical derivations contain an inherent intermediate state. From a semantic perspective, this middle phase consists of logical connectives and sequential transitions that bridge direct calculation and complex decision-making. To capture this structural nuance, we deliberately set the number of clusters to $K = 3$.

B Detailed DTW Clustering Trajectories

As established in the main text (Table 3), the Time-Series K-means clustering consistently stratifies the generated tokens into three distinct semantic phases based on their temporal entropy evolution: Execution (low entropy), Logic (medium entropy), and Thinking (high entropy). Crucially, as visualized in the following figures, the most frequent tokens within each cluster perfectly match the semantic categories we defined in Table 3.

An insight from this empirical data is the algorithm's ability to capture the cognitive state behind token casing. For instance, the capitalized token `Wait` is consistently categorized into the Medium Entropy (Logic) cluster, whereas its lowercase counterpart `wait` robustly falls into the High Entropy (Thinking) cluster. This distinction reflects the underlying generation mechanism: `Wait` typically initiates a new sentence, serving as a structured, template-driven transition. Conversely, a lowercase `wait` generally emerges mid-sentence as a spontaneous self-interruption or correction, which naturally carries a much higher degree of

uncertainty and branching potential.

To demonstrate that this clear phase separation is a robust and universal phenomenon, this section provides the visual trajectories of these clusters across all evaluated models, context lengths, and reward formulations. A universal geometric feature observable across all these figures is that the middle portion of the centroid trajectories remains remarkably flat. This extended stable plateau directly marks the magnitude of the uncertainty, serving as a clear and intuitive visual indicator that distinctly separates the Low, Medium, and High entropy levels throughout the training process.

Figure 3 shows the clustering results on the smaller Qwen3-1.7B model, confirming that the phase separation occurs even at a reduced parameter scale. For our primary testbed, Qwen3-8B, we display the optimization dynamics under the standard training setup in Figure 4. To ensure these patterns are robust against varying experimental conditions, we further evaluate the Qwen3-8B model with an extended 8K context window in Figure 5, and under a deeper RL intervention setting in Figure 6.

Furthermore, to verify that these findings are not strictly tied to the Qwen family's pre-training, we extend our visual analysis to other architectures. Figure 7 illustrates the trajectories for DeepSeek-R1-Distill-Llama-8B, a model that already possesses strong initial alignment for reasoning tasks. Finally, Figure 8 presents the clustering results on the Llama3.1-8B base architecture. Across all these diverse models and settings, the structural separation of the Execution, Logic, and Thinking clusters remains highly consistent.

C Experiments' Settings

The detailed hyperparameters for our unsupervised reinforcement learning training are provided in Appendix Table 5. Additionally, our generation hyperparameters for the validation phase roughly follow the recommended settings from the Qwen3 Technical Report (Yang et al., 2025).

D Additional Boundary Conditions

As illustrated in Table 4, we divide the additional boundary conditions into:

Maximum Response Length. Previous studies typically constrain the generation by setting a maximum sequence length of 4K (Agarwal et al., 2025). However, this joint constraint is sensitive to the varying lengths of the input questions, leading to inconsistent actual generation budgets across different prompts. To eliminate this factor, our initial evaluations illustrated in Section 4 adopt a more rigorous setting by independently bounding the maximum *response* length to 4K tokens. Under this setting, LP achieves the strongest performance among all unsupervised RL methods. We suspect this is because LP explicitly penalizes lengthy generations, forcing the model to conclude its reasoning within the budget and thereby avoiding early truncation. To verify this hypothesis and observe the methods' effectiveness under an abundant generation budget, we extended the maximum response length to 8K. When evaluated at 8K, CH2 achieves the highest overall average score, while LP remains highly effective.

Datasets. Furthermore, to ensure these observations are not artifacts of a specific data distribution, we conducted cross-dataset generalization tests using the DeepMath corpus. The results demonstrate that the performance trends remain consistent: LP achieves the highest average score among unsupervised methods, strictly comparable to the performance of S-RL. This confirms that our findings regarding these unsupervised reward formulations generalize across different mathematical datasets.

E 3D Exploration Boundaries (Convex Hulls)

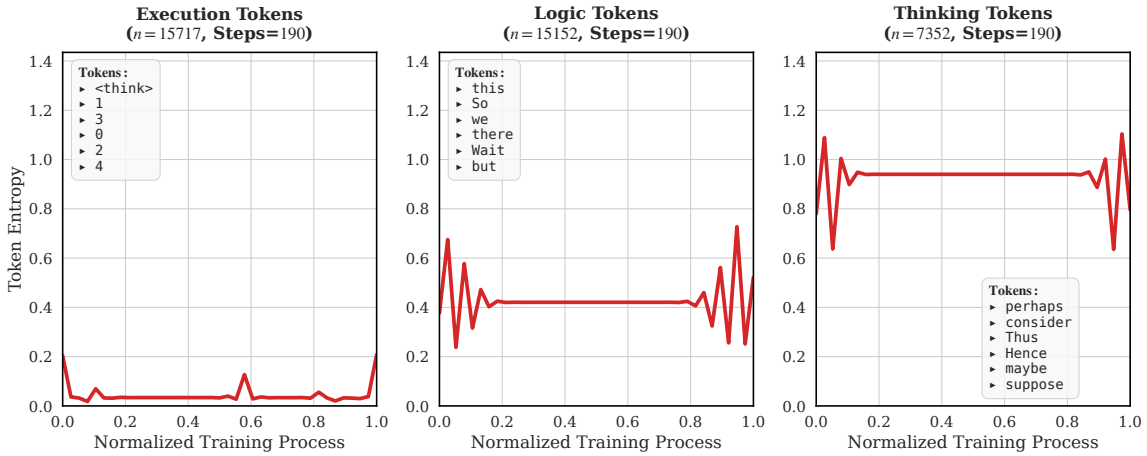
To quantitatively compare how different reward formulations shape the reasoning space, we visualize the 3D exploration boundaries of the semantic phases. By computing the convex hull of the prompt-level trajectories in the 3D entropy space (Execution, Logic, and Thinking), we can assess the diversity and extent of the model's exploratory behavior. A well-regularized reward (e.g., Ent) maintains a healthy, expansive exploration volume, whereas overly aggressive or misaligned penalties result in restricted boundaries or entropy collapse. A larger hull volume indicates a broader exploration of reasoning patterns, while a collapsed volume (e.g., converging to a plane or a point) signifies mode collapse or rigid generation.

Figures 9 and 10 visualize these convex hulls across the different model groups and settings.

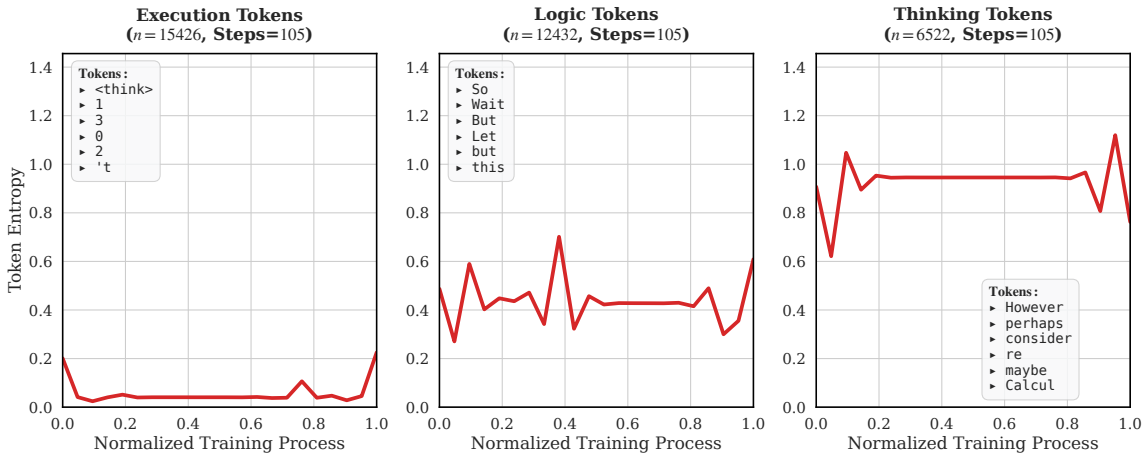
Algorithm 1 Token-Level Entropy Trajectory Construction and Semantic Clustering

Require: Validation dataset $\mathcal{D}_{val}$, Set of training evaluation steps $K = \{t_1, t_2, \ldots, t_M\}$
Require: Target number of clusters $C = 3$ (Execution, Logic, Thinking)
Ensure: Semantic clusters $\mathcal{C}$ and their corresponding centroids

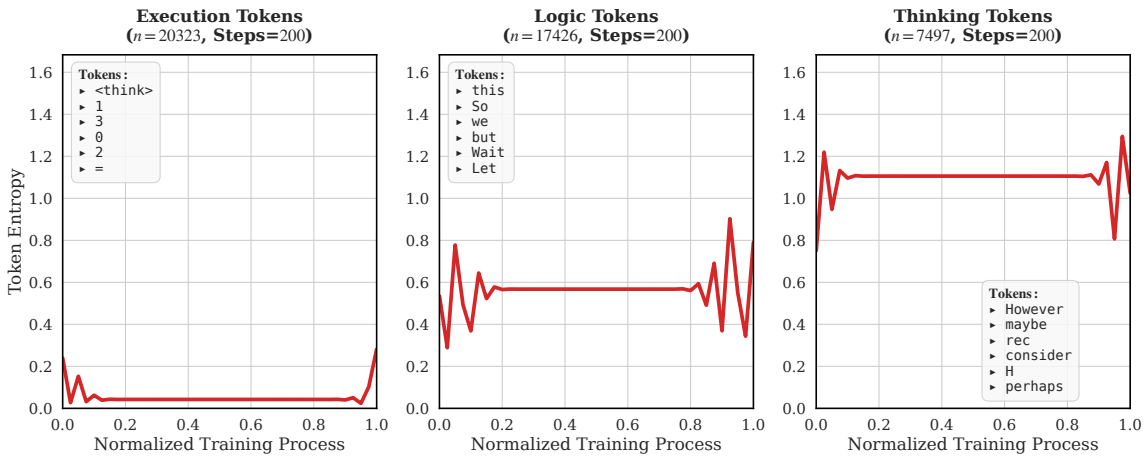
```
Initialize an empty hash map T to store trajectories, keyed by (p, y).
for each evaluation step t ∈ K do
    Load model checkpoint M_t
    for each prompt p ∈ D_val do
        Generate response sequence R_{p,t} using M_t(p)
        Extract response tokens Y_{p,t} = [y_1, y_2, ..., y_N]
        Extract token-level entropies H_{p,t} = [h_1, h_2, ..., h_N]
        for i = 1 to N do
            T[(p, y_i)].append_record(step = t, entropy = h_i)
        end for
    end for
end for
T_valid ← ∅
for each unique anchor (p, y) in T do
    Extract raw entropy sequence S = [h_{t_a}, ..., h_{t_b}] from T[(p, y)]
    if length(S) ≥ 2 then                ▷ Filter isolated tokens without temporal dynamics
        Normalize time axis of S to t̂ ∈ [0, 1] based on the effective convergence point
        T_valid ← T_valid ∪ {S}
    end if
end for
Initialize Time-Series K-Means model K_DTW with k = C and metric = Soft-DTW
C, Centroids ← K_DTW.fit_predict(T_valid)
return C
```



(a) Supervised RL

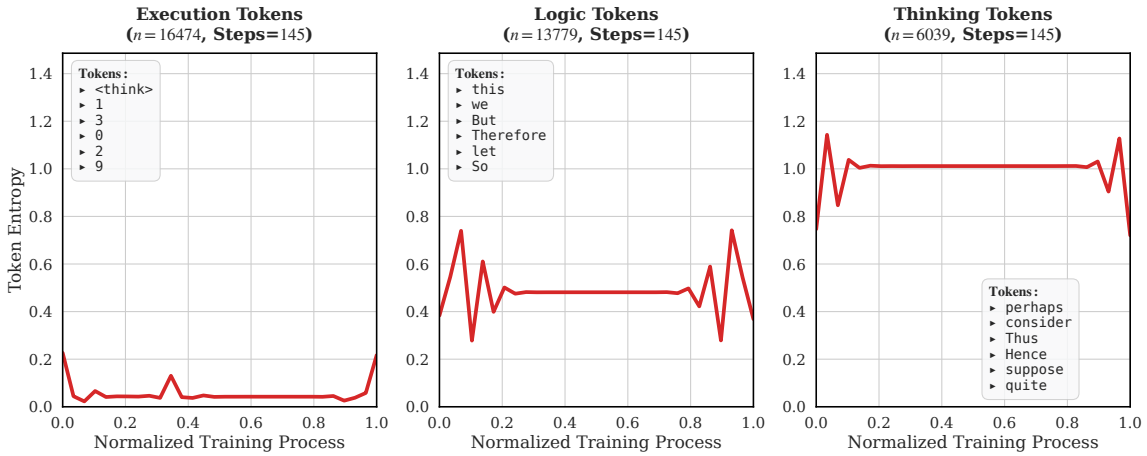


(b) Ent

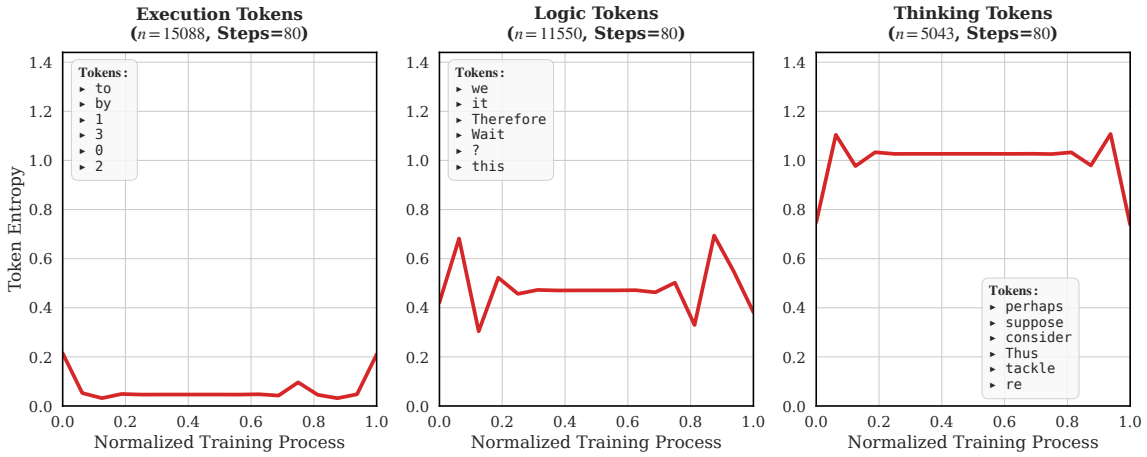


(c) AvgEnt

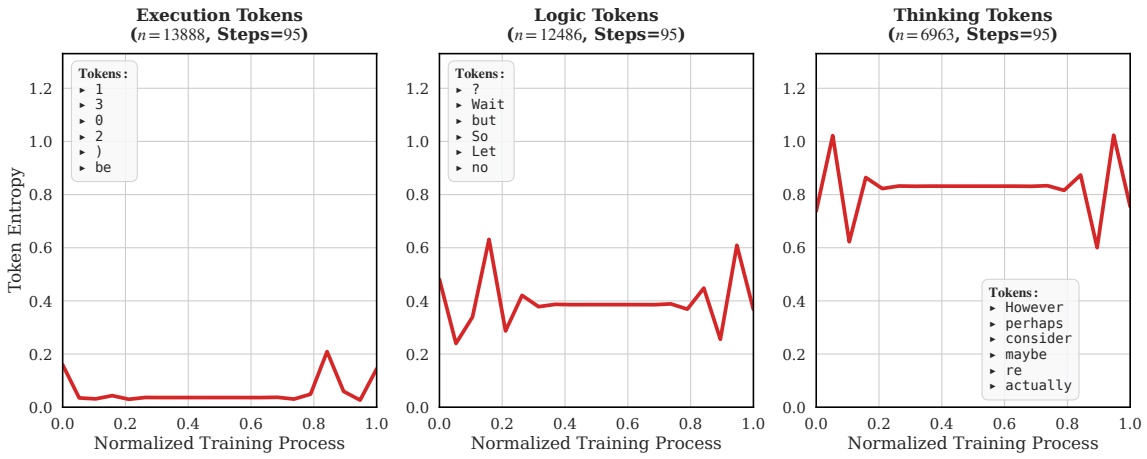
Figure 3: **Token Entropy Trajectories on Qwen3-1.7B (Part 1 of 2).** Clustering results evaluated on the AIME 24 dataset across six different reward formulations. The consistent emergence of Execution, Logic, and Thinking clusters is observed despite the smaller parameter scale. The text boxes within each subplot display the top frequency tokens corresponding to that cluster.



(d) CH2

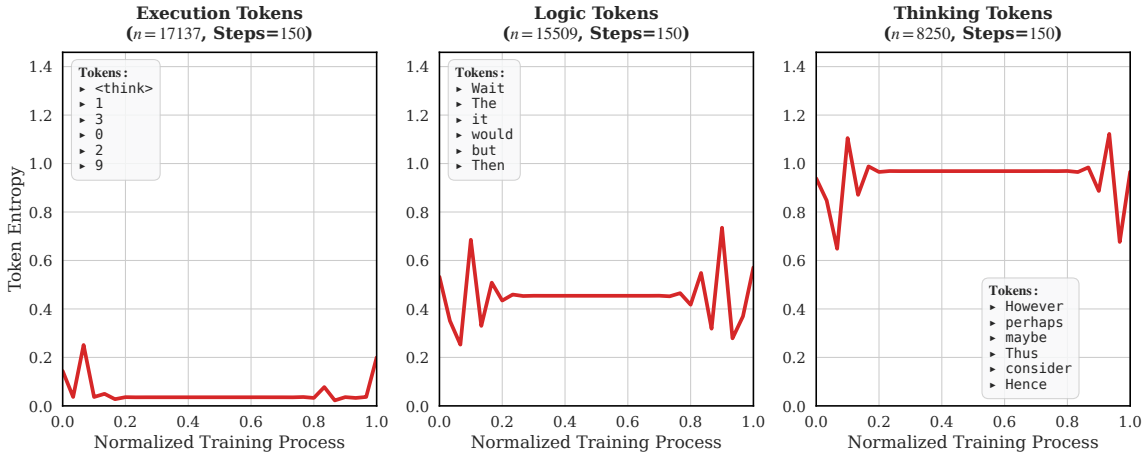


(e) LP

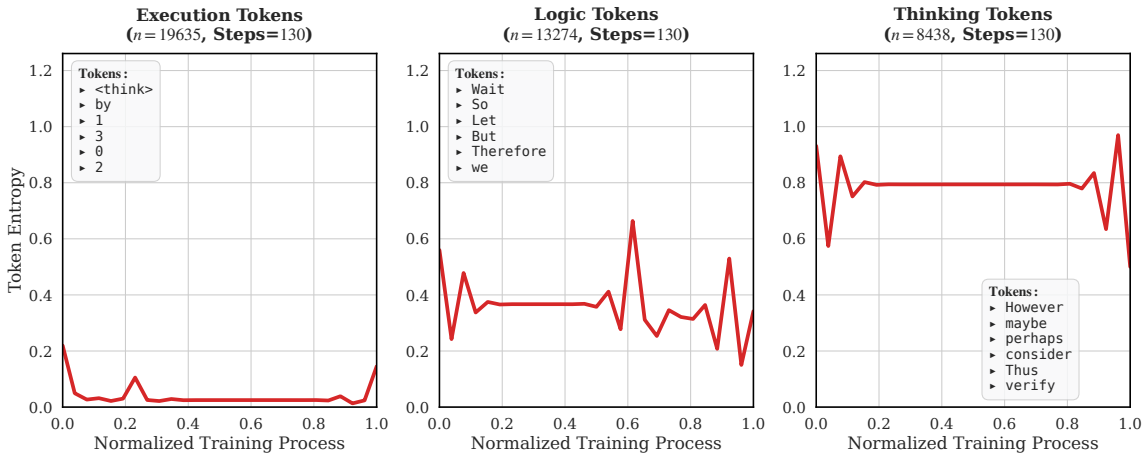


(f) CP

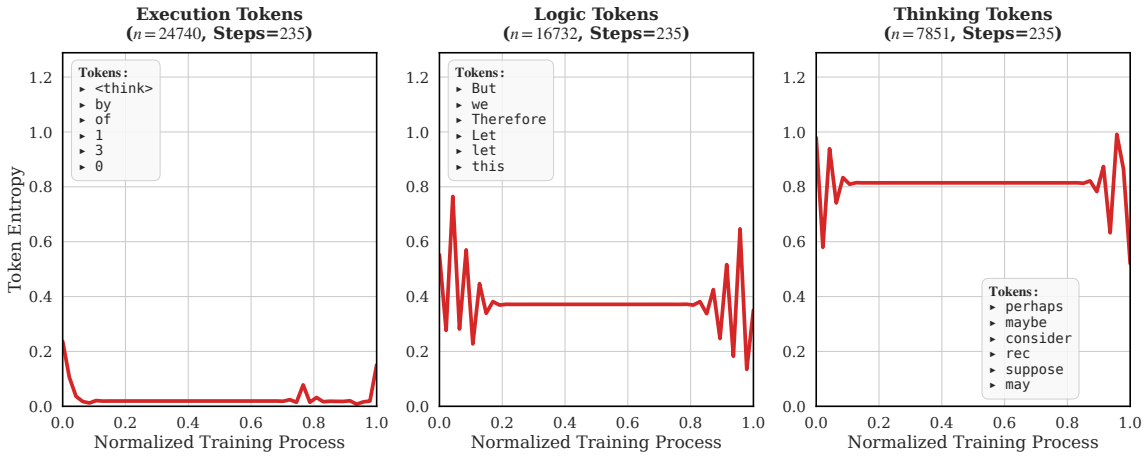
Figure 3: **Token Entropy Trajectories on Qwen3-1.7B (Part 2 of 2).** (Continued from previous page.)



(a) Supervised RL

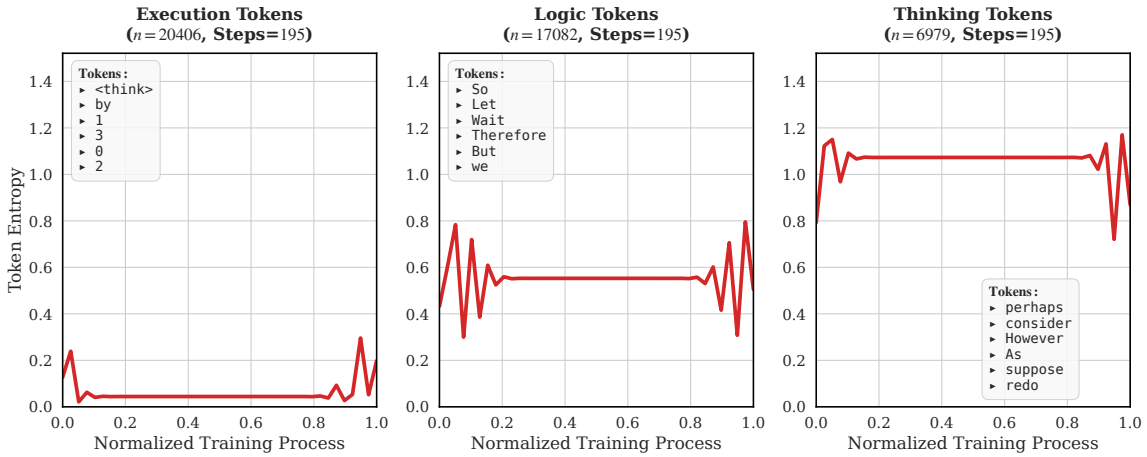


(b) Ent

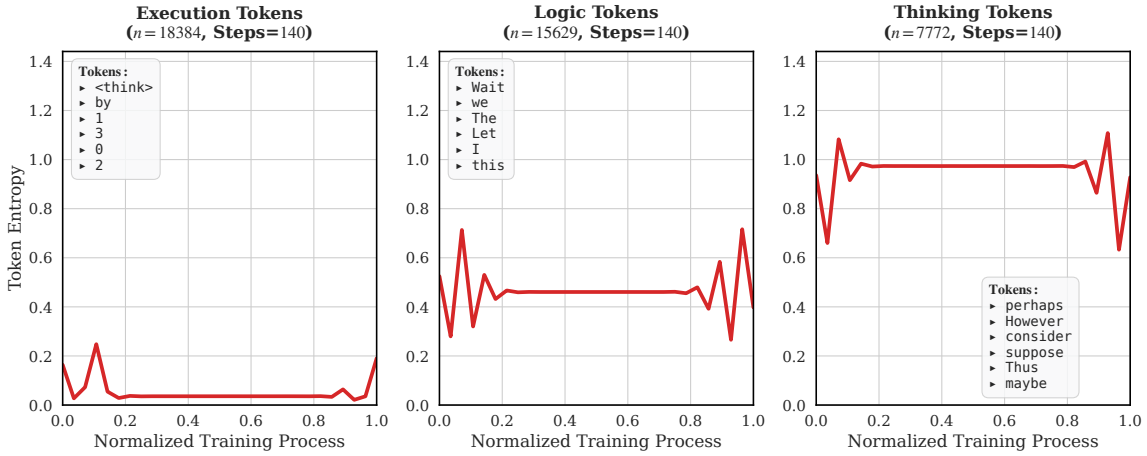


(c) AvgEnt

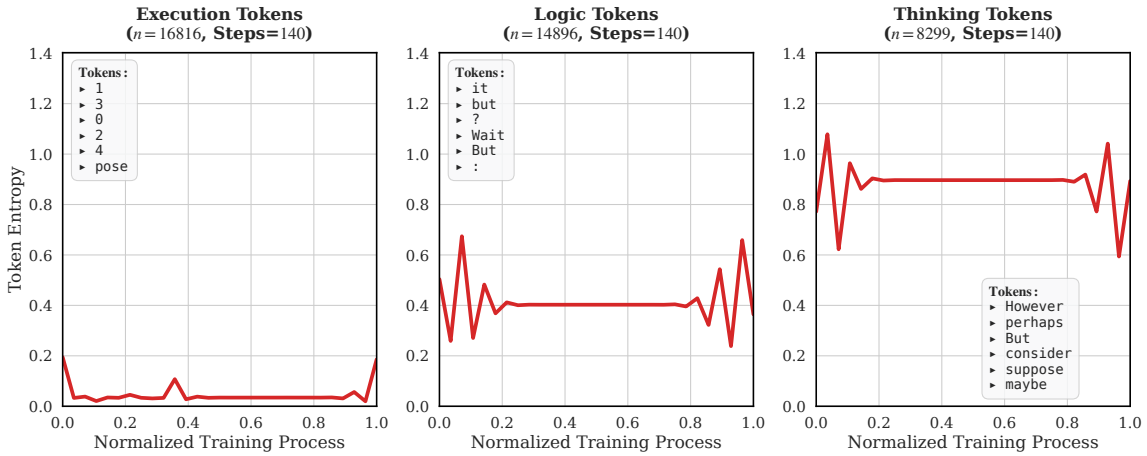
Figure 4: **Token Entropy Trajectories on Qwen3-8B (Standard Setup) (Part 1 of 2).** Visualizations of the optimization dynamics under standard context length on our primary testbed model. The text boxes within each subplot display the top frequency tokens corresponding to that cluster.



(d) CH2

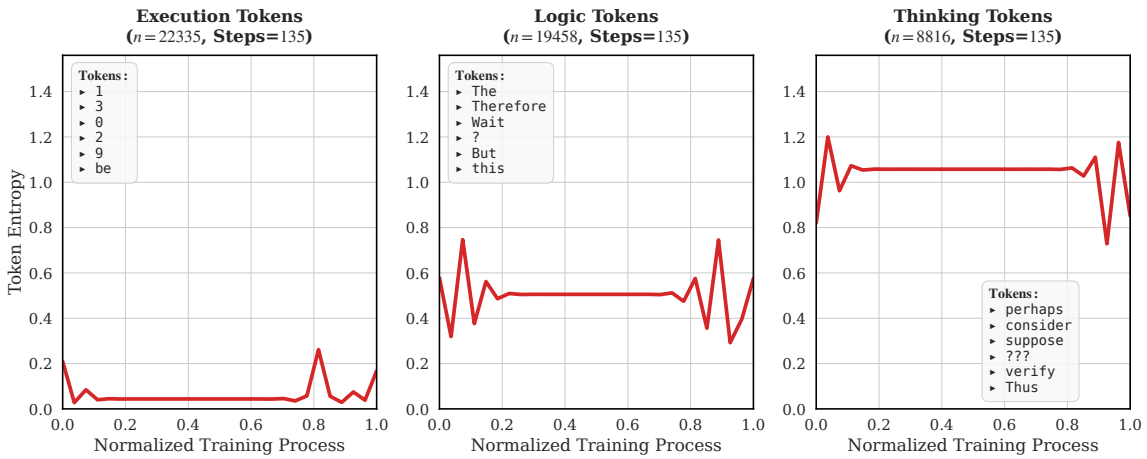


(e) LP

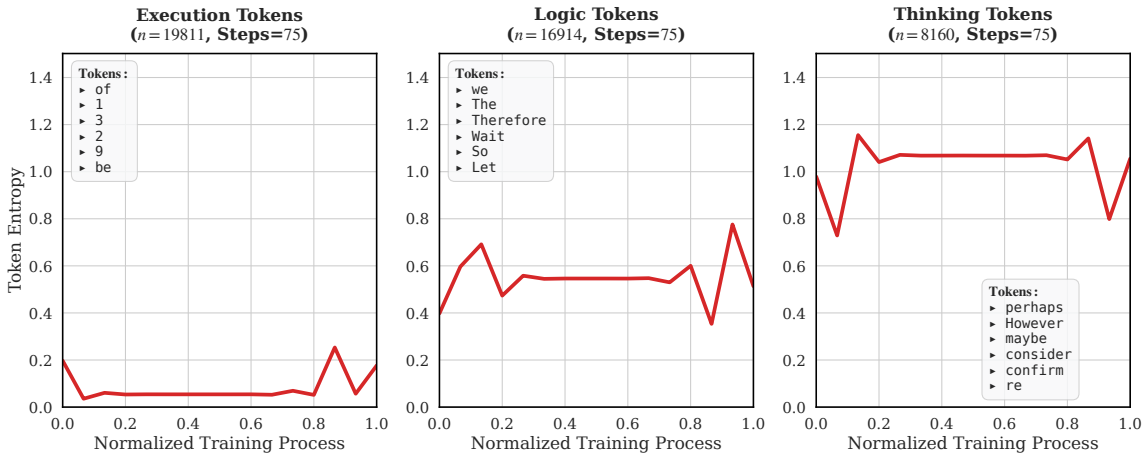


(f) CP

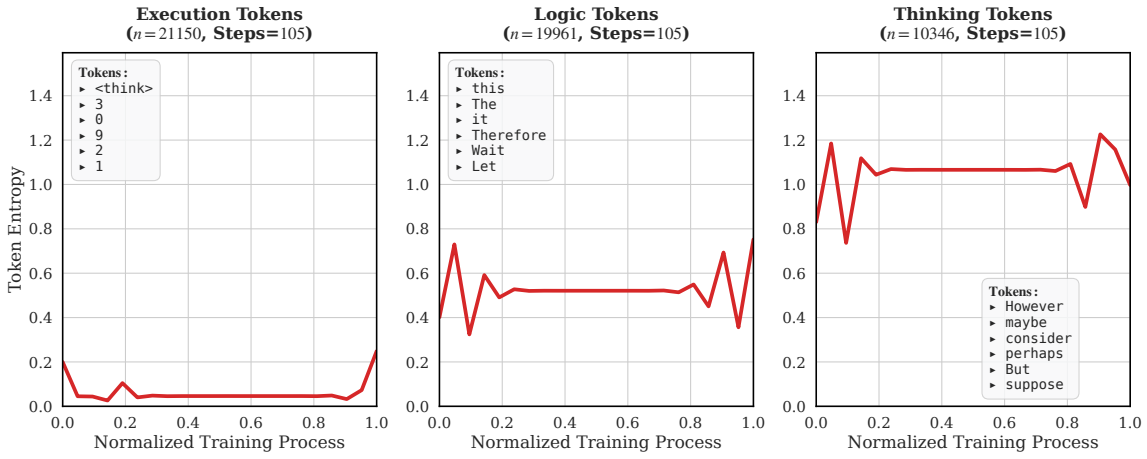
Figure 4: **Token Entropy Trajectories on Qwen3-8B (Standard Setup) (Part 2 of 2).** (Continued from previous page.)



(a) Supervised RL

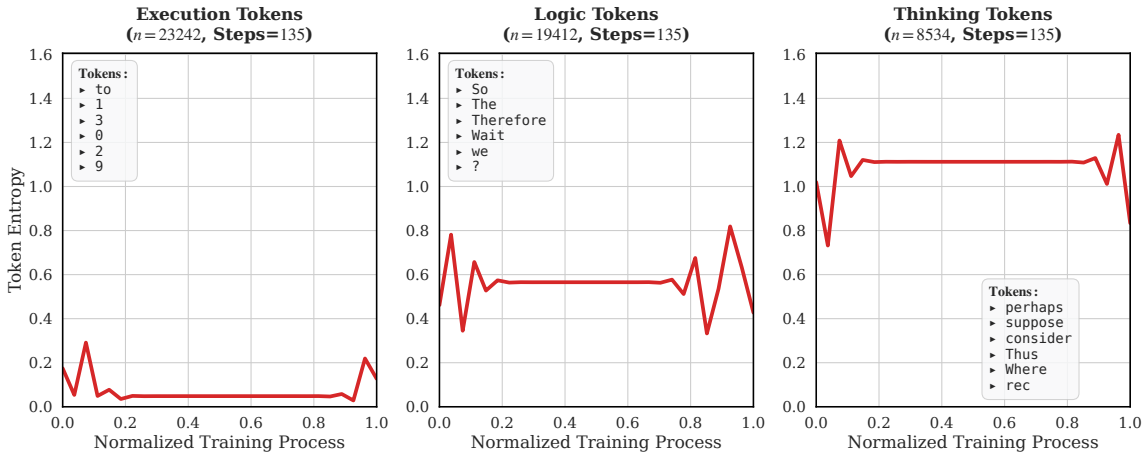


(b) Ent

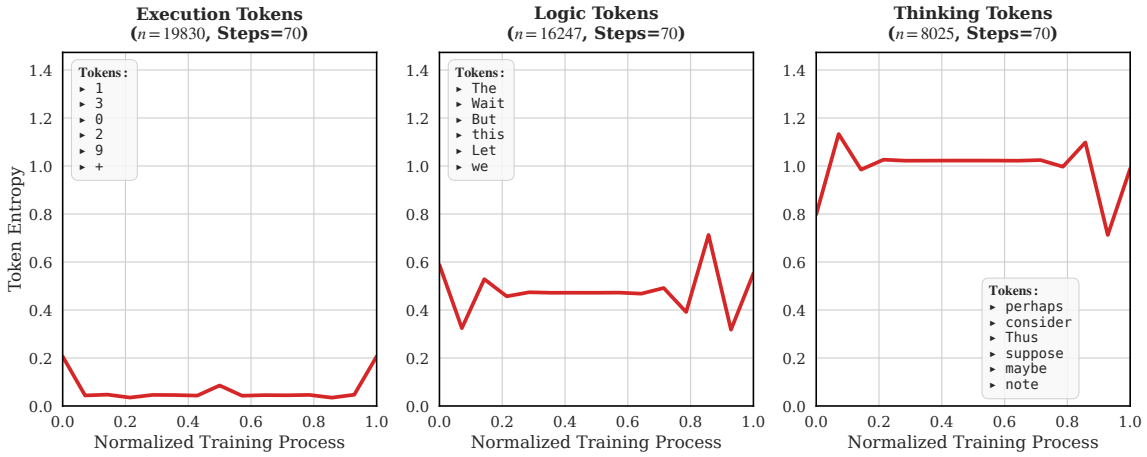


(c) AvgEnt

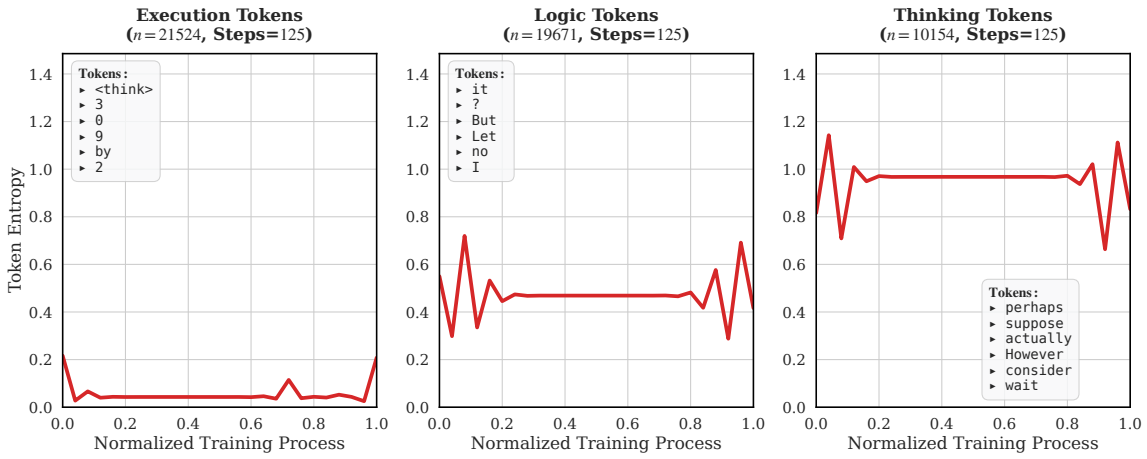
Figure 5: **Token Entropy Trajectories on Qwen3-8B (8K Context) (Part 1 of 2).** These visualizations demonstrate that the emergence of reasoning clusters remains consistent when scaling the context window to 8K. The text boxes within each subplot display the top frequency tokens corresponding to that cluster.



(d) CH2

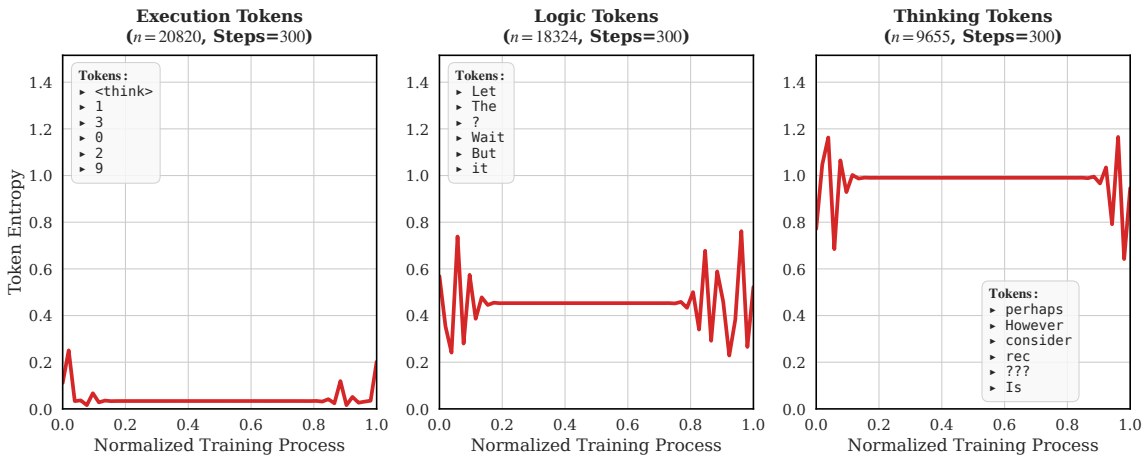


(e) LP

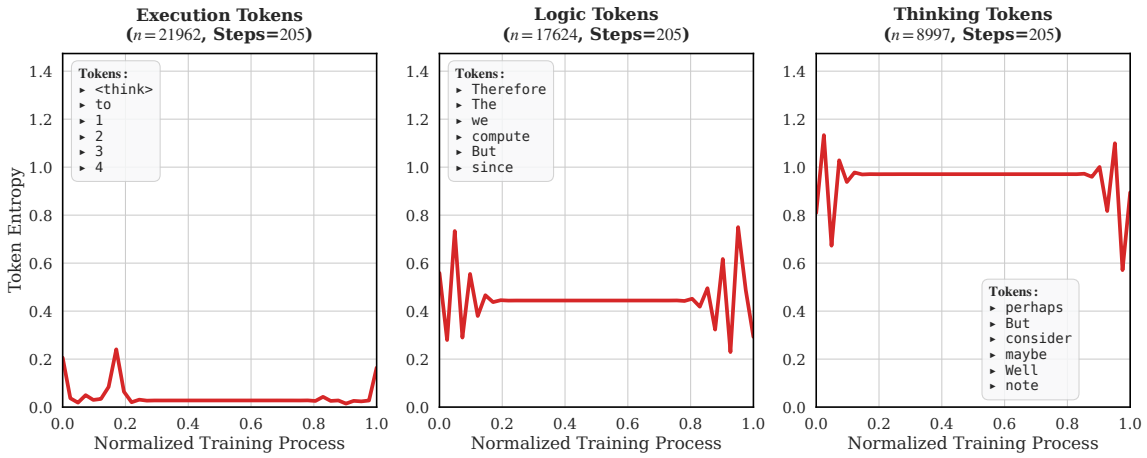


(f) CP

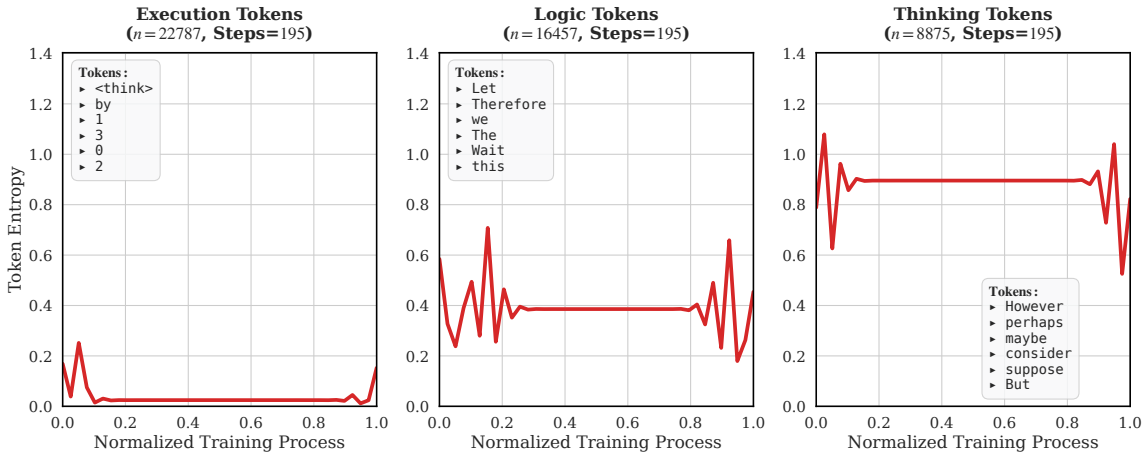
Figure 5: **Token Entropy Trajectories on Qwen3-8B (8K Context) (Part 2 of 2).** (Continued from previous page.)



(a) Supervised RL

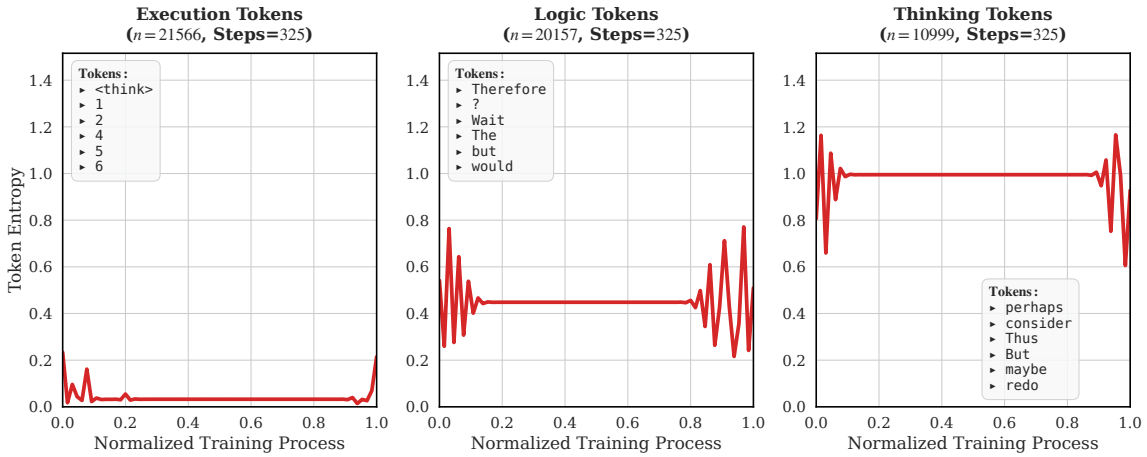


(b) Ent

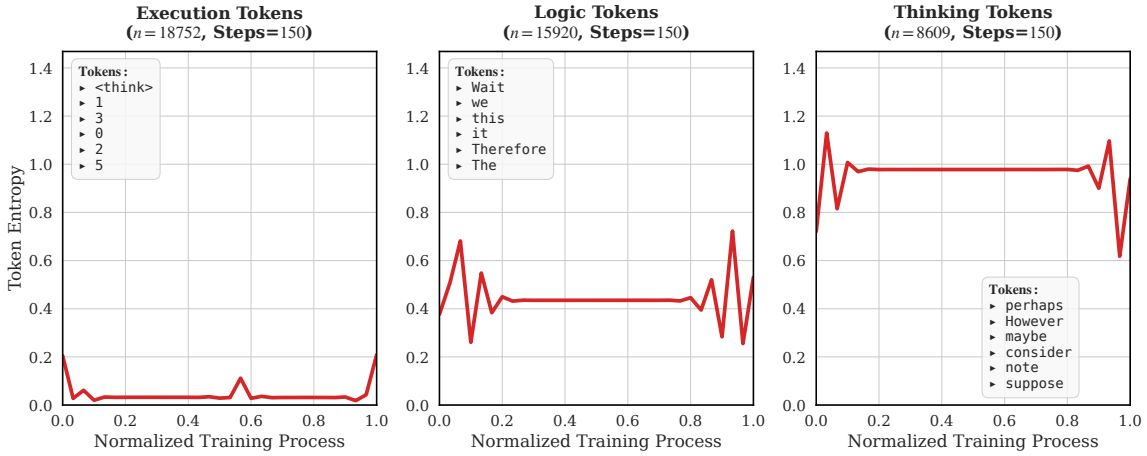


(c) AvgEnt

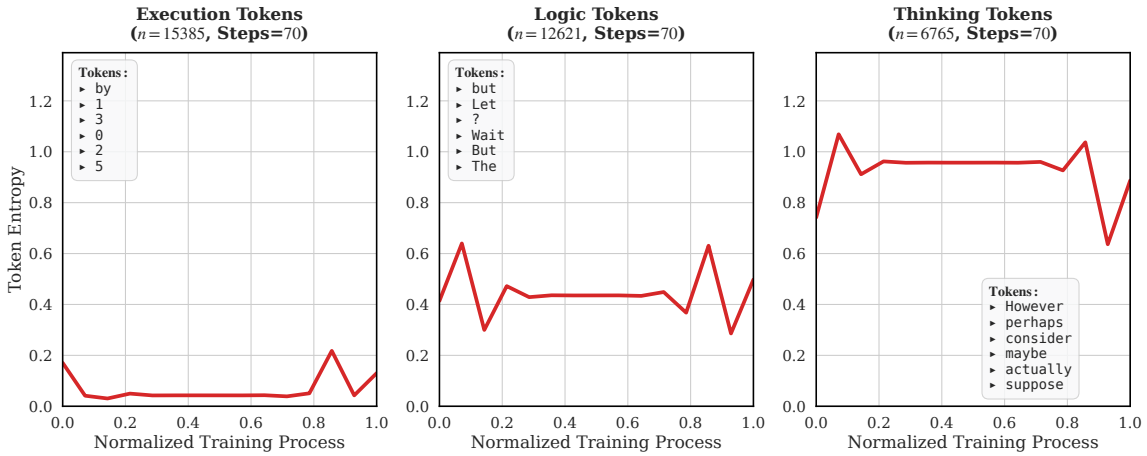
Figure 6: **Token Entropy Trajectories on Qwen3-8B (DeepMath Dataset) (Part 1 of 2).** Trajectories analyzed under deeper RL interventions, illustrating the robustness of the intrinsic phase separation. The text boxes within each subplot display the top frequency tokens corresponding to that cluster.



(d) CH2

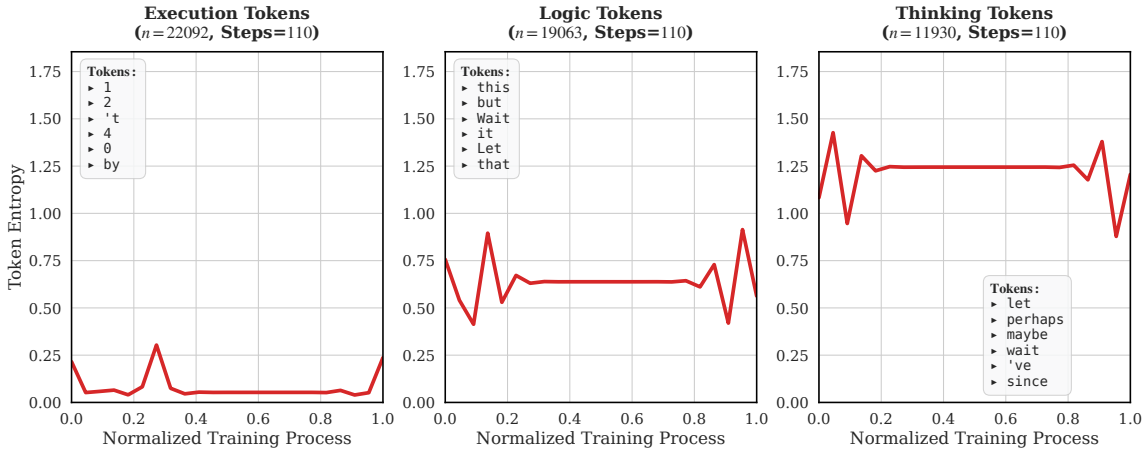


(e) LP

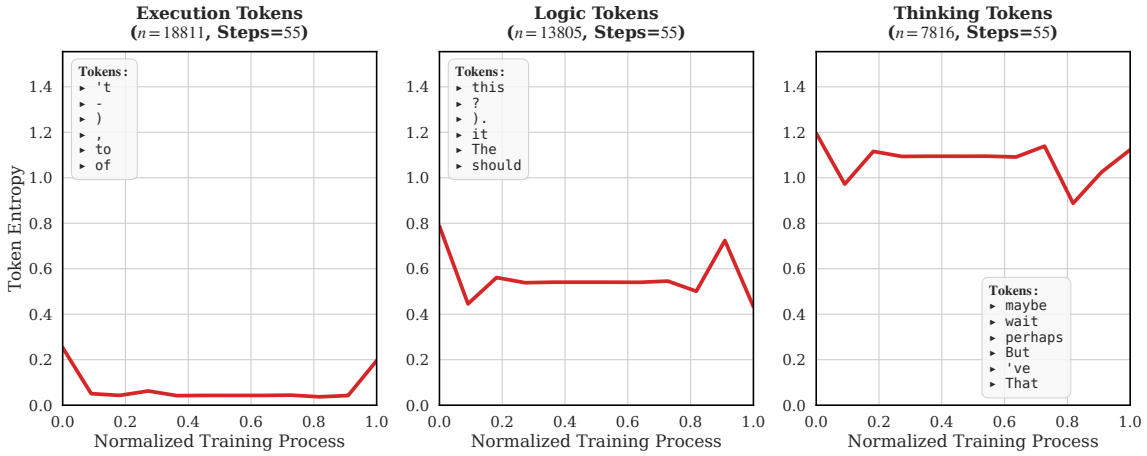


(f) CP

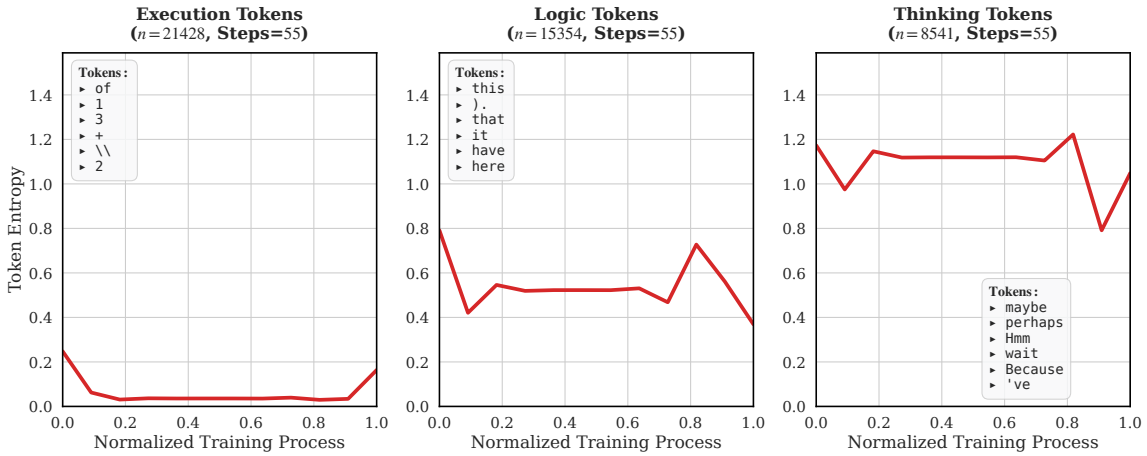
Figure 6: **Token Entropy Trajectories on Qwen3-8B (DeepMath Dataset) (Part 2 of 2).** (Continued from previous page.)



(a) Supervised RL

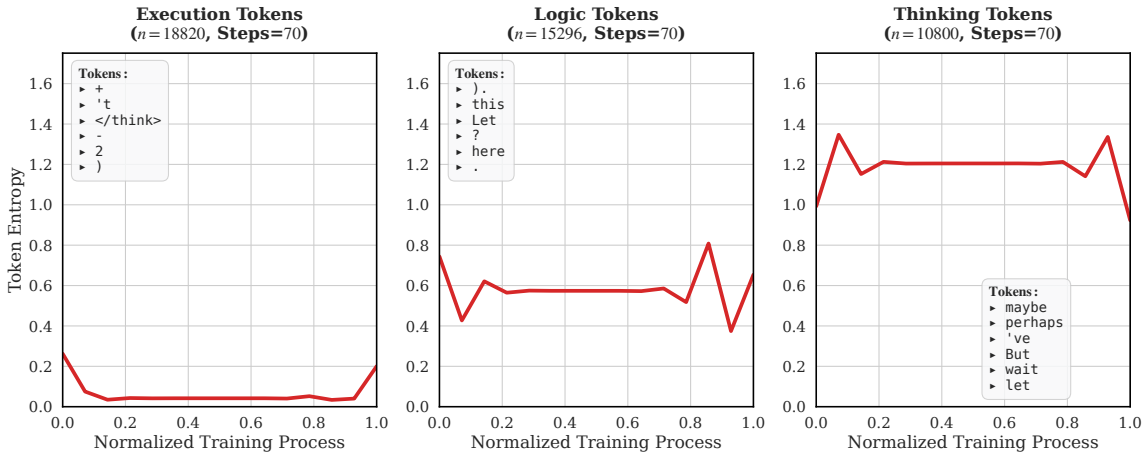


(b) Ent

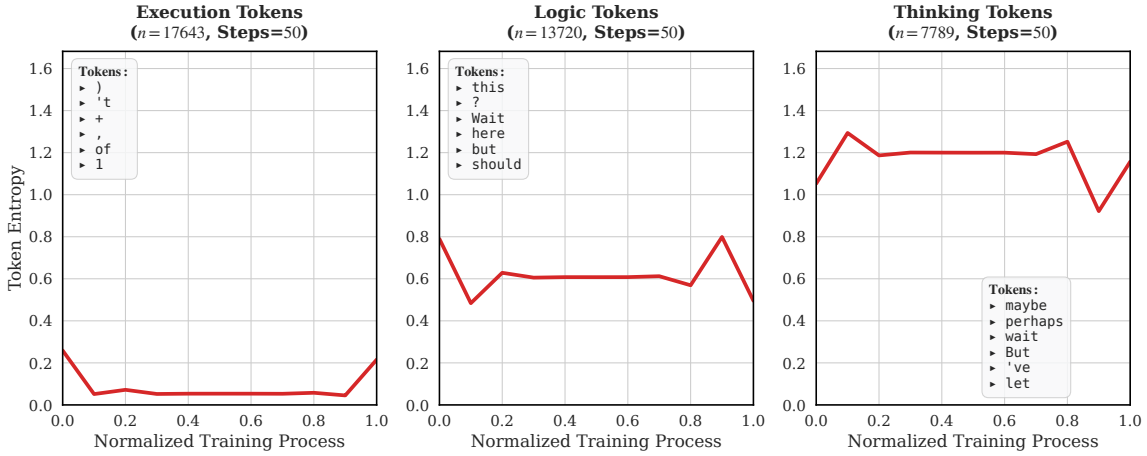


(c) AvgEnt

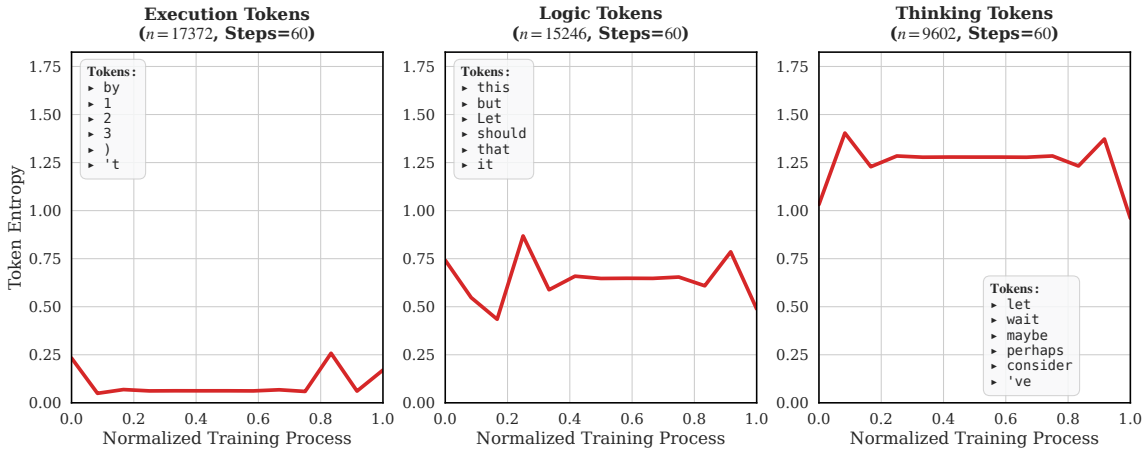
Figure 7: **Token Entropy Trajectories on DeepSeek-R1-Distill-Llama-8B (Part 1 of 2).** Analysis conducted on an architecture with a highly-aligned initial state, further isolating the impact of the reward signal. The text boxes within each subplot display the top frequency tokens corresponding to that cluster.



(d) CH2

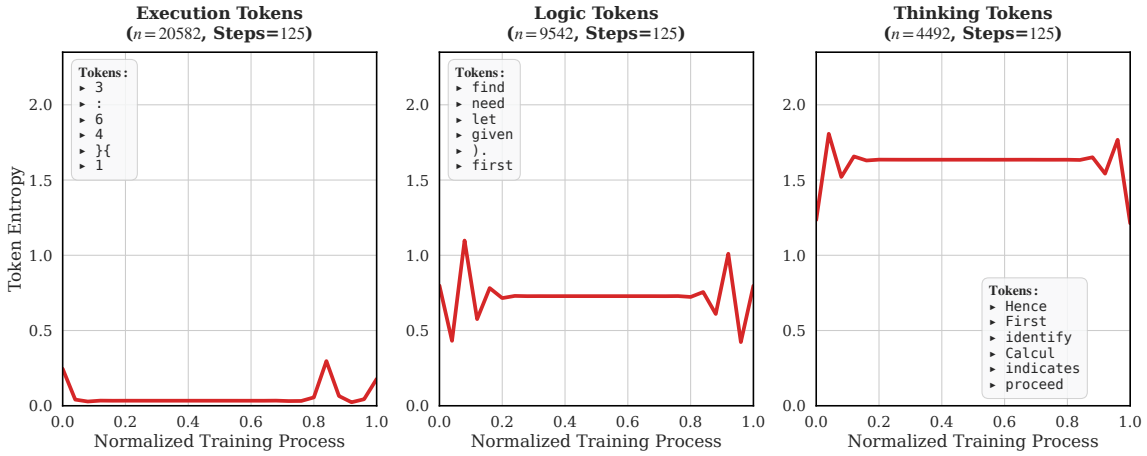


(e) LP

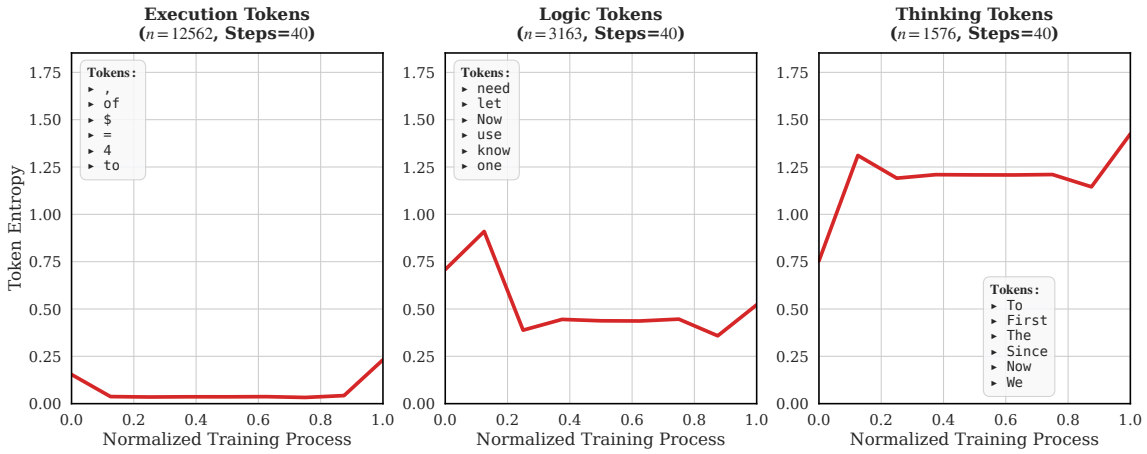


(f) CP

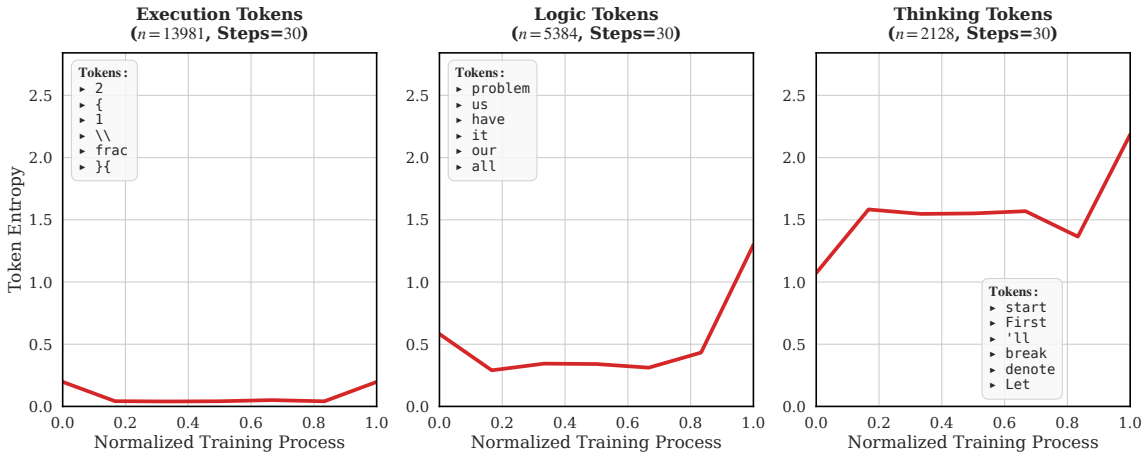
Figure 7: **Token Entropy Trajectories on DeepSeek-R1-Distill-Llama-8B (Part 2 of 2).** (Continued from previous page.)



(a) Supervised RL

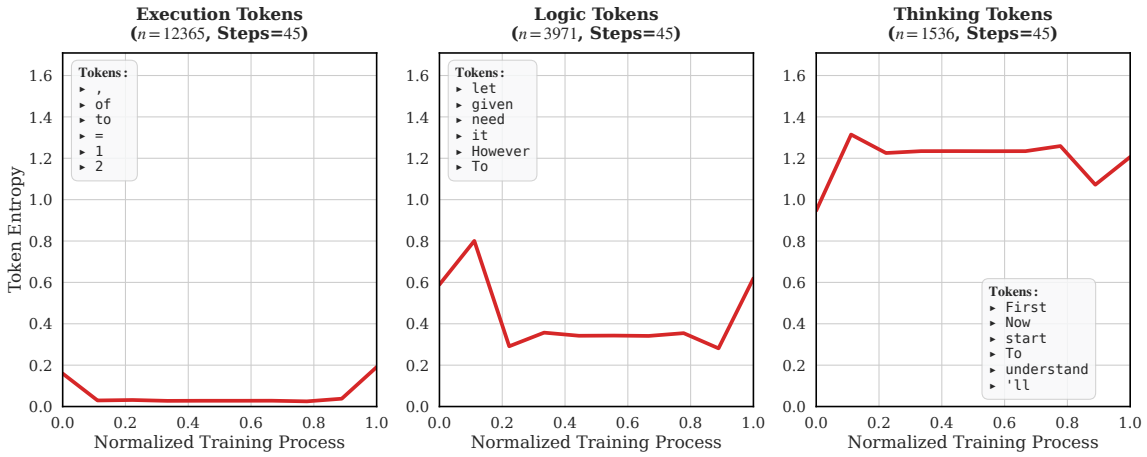


(b) Ent

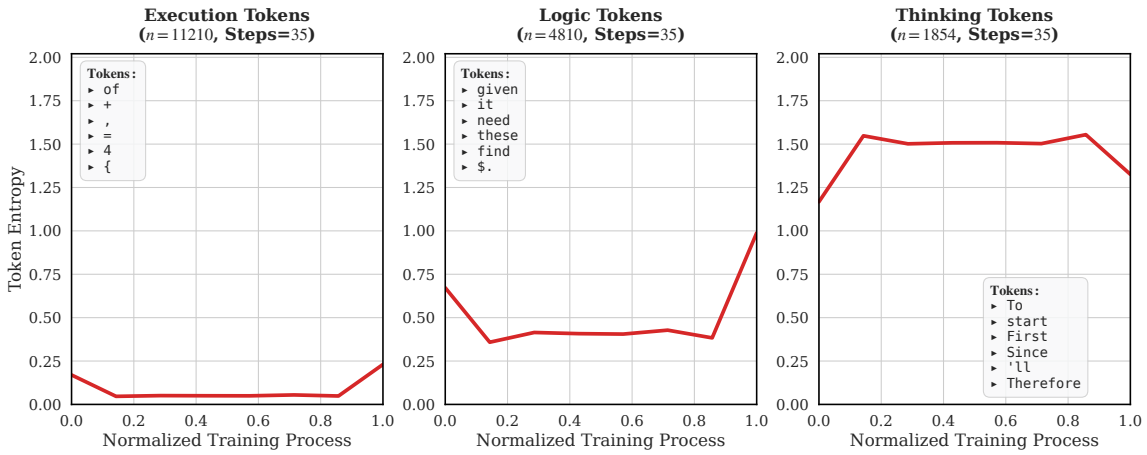


(c) AvgEnt

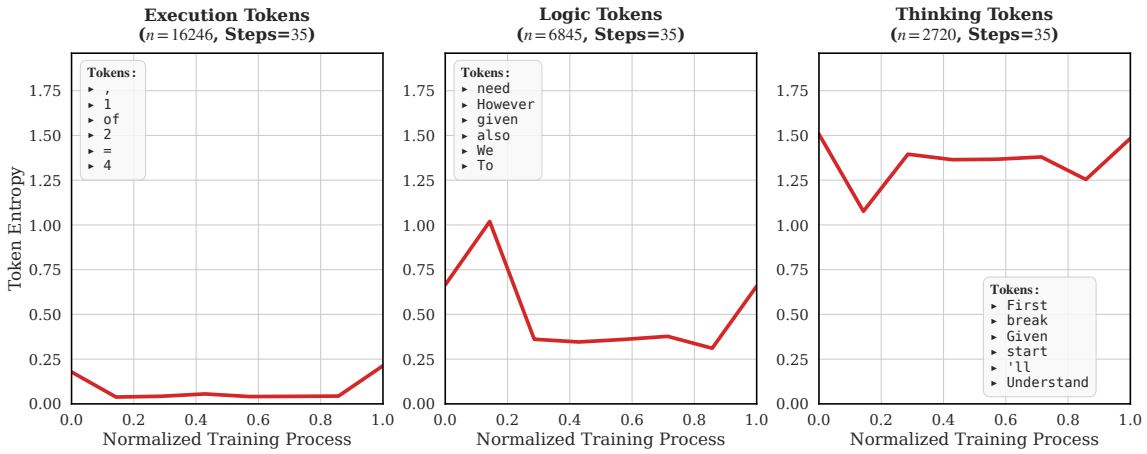
Figure 8: **Token Entropy Trajectories on Llama3.1-8B (Part 1 of 2).** Clustering evaluation on the Llama base architecture to verify cross-model generalizability of the mechanistic interpretations. The text boxes within each subplot display the top frequency tokens corresponding to that cluster.



(d) CH2

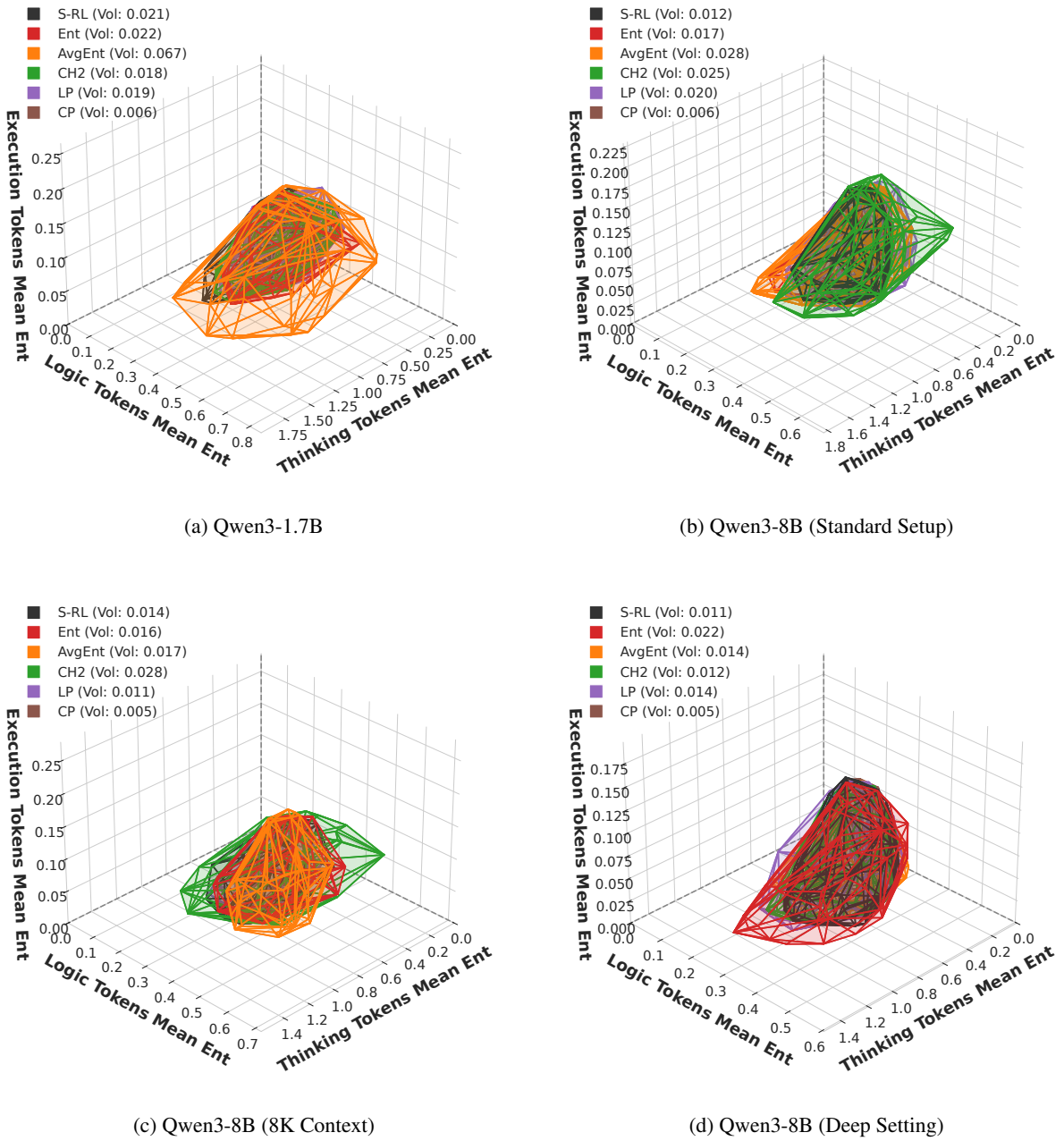


(e) LP



(f) CP

Figure 8: **Token Entropy Trajectories on Llama3.1-8B (Part 2 of 2).** (Continued from previous page.)



(a) Qwen3-1.7B

(b) Qwen3-8B (Standard Setup)

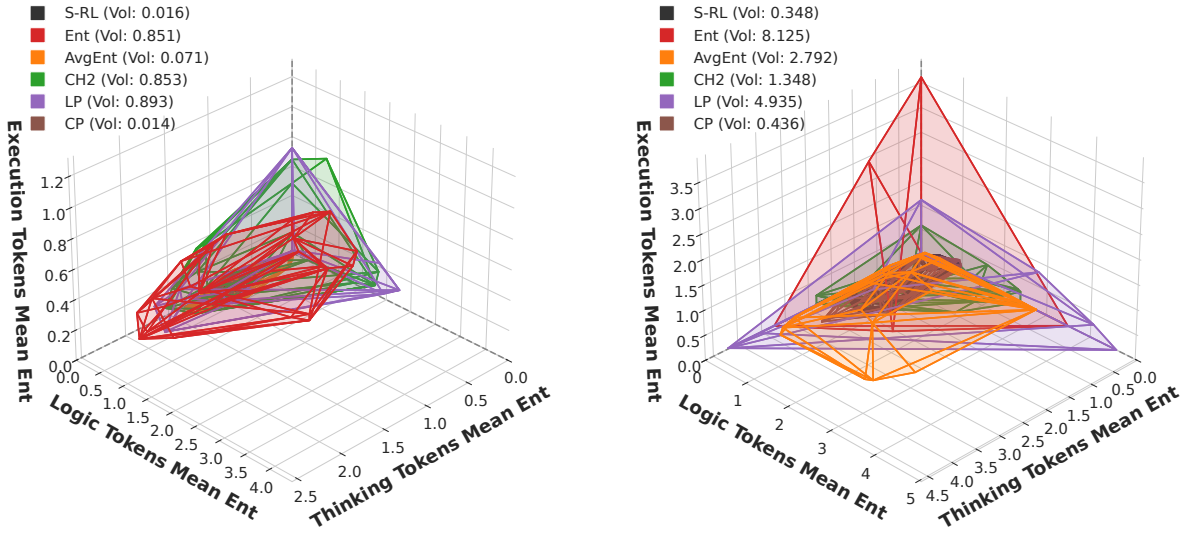
(c) Qwen3-8B (8K Context)

(d) Qwen3-8B (Deep Setting)

Figure 9: **3D Exploration Boundaries of Semantic Phases (Qwen Models).** The convex hulls illustrate the volume of the entropy phase space explored by the models under six different reward formulations.

Model	Method	MATH-500			Olympiad			Minerva			AIME 24			AMC 23			AIME 26			AVG		
		P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10
Qwen3-8B *(8K)*	Base	85.1	91.9	93.7	46.9	56.7	60.8	42.2	51.4	54.4	38.3	53.2	59.0	70.5	85.9	90.4	28.1	42.0	46.9	58.5	67.5	70.8
	S-RL	90.9	94.8	95.7	56.4	64.9	68.0	47.7	56.1	58.7	50.2	66.5	71.8	86.1	93.1	94.6	43.8	58.3	63.6	66.4	73.6	76.0
	Ent	91.7	95.7	96.5	58.3	67.7	70.8	47.7	55.9	58.1	**53.5**	**76.5**	82.1	86.4	**96.4**	97.5	49.4	**66.4**	**70.4**	67.7	75.6	77.8
	AvgEnt	89.9	94.4	95.5	54.3	62.3	65.3	46.8	56.0	58.9	44.4	61.7	69.6	82.8	93.4	94.7	44.4	60.9	66.7	64.8	72.3	74.8
	LP	**91.9**	**95.8**	**96.6**	58.4	67.3	70.2	47.5	56.2	59.4	53.1	75.3	**82.8**	**87.8**	94.5	96.2	**50.0**	65.4	70.0	67.8	75.4	77.7
	CH2	91.4	95.6	96.5	**59.0**	**69.0**	**72.0**	**53.8**	**64.8**	**68.1**	51.2	72.9	79.1	87.0	95.4	**98.0**	47.5	64.9	69.5	**68.9**	**77.5**	**80.0**
	CP	*Collapse* (×)																				
Qwen3-8B *(DeepMath)*	Base	67.9	80.5	84.4	28.4	41.2	46.7	29.8	39.9	44.8	7.7	19.9	23.1	41.7	64.1	73.6	9.2	16.6	22.1	41.0	53.4	58.3
	S-RL	90.1	94.7	95.3	**55.9**	**64.2**	66.8	**54.4**	**65.9**	**68.8**	**41.2**	61.2	**68.7**	**81.9**	**95.7**	**98.1**	**39.6**	55.0	60.2	**66.8**	**74.9**	**77.1**
	Ent	89.0	94.4	95.3	52.5	63.2	66.3	51.4	61.9	64.6	35.8	56.5	64.2	77.5	93.6	96.9	30.0	53.7	61.1	64.0	73.5	76.0
	AvgEnt	79.9	88.5	91.1	41.7	52.2	56.0	38.9	49.0	52.7	24.2	37.9	43.7	61.1	75.6	80.1	19.2	30.8	35.8	53.3	63.3	66.8
	LP	**90.2**	**94.9**	**95.7**	54.8	**64.2**	**67.3**	53.1	63.9	67.1	39.4	**63.1**	68.4	**81.9**	93.8	94.9	38.3	**57.3**	**61.2**	66.0	74.7	**77.1**
	CH2	89.7	94.6	95.3	54.3	64.1	67.1	52.5	64.2	67.7	36.2	57.0	65.2	79.5	93.1	94.9	34.0	55.5	60.6	65.3	74.4	76.9
	CP	*Collapse* (×)																				

Table 4: **Boundary Conditions Results.** We expand our evaluation across two dimensions: (1) 8K training and evaluations results; and (2) cross-dataset generalization tests utilizing DeepMath. Notions follow Table 2.



(a) DeepSeek-R1-Distill-Llama-8B

(b) Llama3.1-8B

Figure 10: **3D Exploration Boundaries of Semantic Phases (Other Architectures).** The convex hulls illustrate the volume of the entropy phase space explored by the models under six different reward formulations.

Hyperparameter	**Value**
Optimization	
Actor Learning Rate	1×10^{-6}
Learning Rate Warmup Steps	10
Weight Decay	0.1
Mini-batch Size	24
Prompt Batch Size	8
Reinforcement Learning	
Responses per Prompt (K)	16
PPO Clip Ratio (ϵ)	0.2
Loss Aggregation Mode	seq-mean-token-mean
Max Prompt Length	1024
Generation (Rollout & Validation)	
Training Temperature	0.6
Training Top-p	1.0
Training Top-k	-1
Validation Top-p	0.95
Validation Top-k	20

Table 5: Hyperparameters for the RL trainings.