Cyclical Entropy Eruption: Entropy Dynamics of Agent Reinforcement Learning

Wendi Li **Shawn Im** **Sharon Li**
Department of Computer Sciences
University of Wisconsin-Madison
{wli679, shawnim, sharonli}@cs.wisc.edu

Abstract

Agentic large language models are increasingly used to solve real-world tasks by reasoning over goals, invoking tools, and interacting with external environments. Reinforcement learning provides a natural framework for improving these behaviors, and recent agent RL methods have achieved strong results across domains. However, the training dynamics of agent RL remain poorly understood, limiting our ability to diagnose instabilities and design more effective training algorithms. In this work, we identify a previously underexplored phenomenon in agent RL, which we term *cyclical entropy eruption*. Unlike single-turn reasoning RL, where entropy typically collapses and stays low, agent RL training exhibits unique recurring cycles of sharp entropy eruption and gradual subsidence. We decompose this dynamic into three phases and provide theoretical and empirical analyses of each, explaining the mechanisms underlying its cyclical oscillation. We further show that degenerate patterns such as sentence duplication and hallucination, once acquired during eruption, can persist and accumulate across cycles. Motivated by these findings, we propose SEAL (Separation-Enhanced Agent Learning), a lightweight auxiliary loss that separates correct and incorrect trajectories in representation space, directly targeting the root cause of entropy eruption. Experiments across multiple benchmarks, models, and RL algorithms demonstrate that SEAL stabilizes training and yields stronger downstream agent performance. Code:

1 Introduction

Large language model agents are rapidly emerging as a practical interface between foundation models and real-world tasks, enabling models to reason over goals, invoke tools, and interact with external environments. This capability has already created value in settings such as web navigation [1, 2, 3], software engineering [4, 5, 6, 7], scientific assistance [8, 9, 10], and embodied decision making [11, 12, 13], where agents can decompose complex objectives into actionable intermediate steps. Reinforcement learning for agents provides a natural framework for improving such behaviors, since it optimizes long-horizon decision quality. Consequently, a growing class of agent RL algorithms has emerged [14, 15, 16], many of which adapt techniques originally developed for LLM reasoning by maximizing advantage-weighted objectives over trajectories. The policy can learn to favor action sequences that improve planning, tool use, and error correction.

Despite the rapid progress of agent RL, our understanding of its training dynamics remains limited. Most existing work focuses on improving benchmark performance or designing new optimization objectives, while offering relatively little insight into how the policy distribution evolves throughout training. Moreover, training-dynamics analyses developed for single-turn LLM RL [17, 18] may not directly transfer to agent settings, where learning unfolds over long-horizon trajectories with structured actions, tool interactions, and validity constraints. Understanding these dynamics is

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

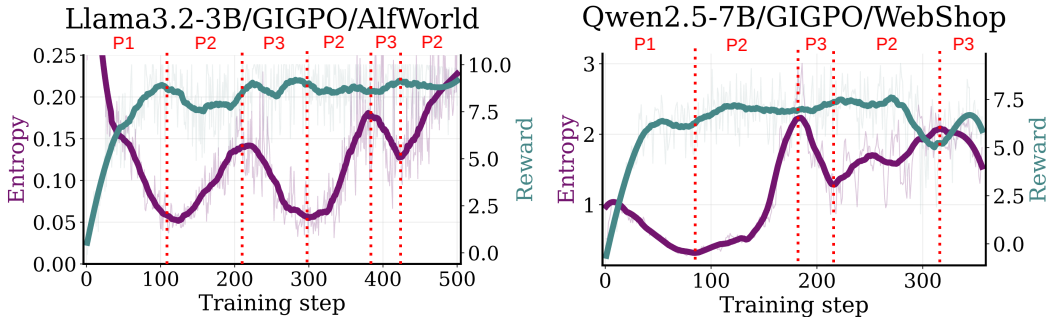

Figure 1: Cyclical entropy eruption in agent RL across models, tasks, and algorithms. P1, P2, and P3 represent three phases, respectively. See more examples in Figure 12.

important not only for explaining empirical successes and failures, but also for guiding algorithm design and diagnosing instability in increasingly complex agent environments.

In this work, we provide an in-depth understanding of agent RL by identifying and analyzing a previously unexplored phenomenon, which we term ***cyclical entropy eruption***. As shown in Figure 1, the policy entropy in agent RL does not simply decrease and stabilize as in single-turn reasoning RL [17, 18]; instead, it undergoes unique recurring cycles of eruption and subsidence throughout training. We decompose this dynamic into three phases: ❶ *Entropy descent*: the entropy descends at the beginning phase of training, where the policy model learns rapidly about how to use tools, call functions, and output in the correct format. ❷ *Entropy eruption*: an entropy surge phase in which high representation similarity among trajectories causes gradient interference that suppresses valid-trajectory likelihood. As a result, this flattens the policy distribution and amplifies degenerate patterns such as sentence duplication and hallucination. ❸ *Entropy subsidence*: As the distribution flattens, the policy samples more diverse trajectories, which in turn reduces representation similarity. As a result, the training gradually reinforces correct responses by increasing the likelihood of correct trajectories, which then triggers the next eruption. The dynamics therefore form a self-perpetuating cycle: convergence raises similarity $\rightarrow$ similarity causes eruption $\rightarrow$ eruption increases diversity $\rightarrow$ diversity enables subsidence $\rightarrow$ and subsidence leads back to convergence.

Motivated by our theoretical understanding of entropy dynamics, we propose SEAL (Separation-Enhanced Agent Learning), a lightweight modification to the original agentic RL objective, with no additional cost at inference time. The objective encourages the model to separate correct and incorrect trajectories in representation space. Therefore, harmful gradient interference is reduced: incorrect and correct trajectories become less likely to induce conflicting updates when their representations are no longer highly similar, which mitigates the likelihood decrease of valid trajectories and helps suppress entropy eruption. While the primary contribution of this paper is the dynamical analysis, SEAL demonstrates that our theoretical understanding can guide practice: even a simple intervention motivated by the analysis meaningfully stabilizes training, improves valid-trajectory likelihood, and yields stronger downstream agent performance across different tasks, backbones, and RL algorithms. For example, SEAL improves the average AlfWorld accuracy of Qwen2.5-7B with GRPO by 2.81% and raises its WebShop success rate by 3.13%. Moreover, on the Llama-series model, where GRPO training collapses and yields a 0% success rate, adding SEAL recovers training and boosts performance to 79.69%. We expect future work to deepen this line of inquiry with more sophisticated methods. Our main contributions are:

1. We identify *cyclical entropy eruption*, a recurring training instability in agent RL in which policy entropy repeatedly erupts and subsides. This phenomenon is qualitatively not observed in single-turn reasoning RL.

2. We provide in-depth theoretical and empirical analyses of each phase, explaining the mechanisms underlying its cyclical oscillation.

3. We propose SEAL, an agent RL loss motivated directly by our analysis, and demonstrate that it stabilizes training and improves agent performance across multiple benchmarks, models, and RL algorithms.

2 Preliminary

LLM agents and structured trajectories. Given a task prompt x, an LLM agent autoregressively generates a *trajectory* $y = (y_1, \ldots, y_T)$ that may contain intermediate reasoning, tool calls, returned observations, and a final answer. Because such tasks typically cannot be completed from parametric knowledge alone, agents invoke external tools such as search engines, calculators, code interpreters, databases, or domain-specific APIs. Tool use is usually implemented through structured action formats, *e.g.* JSON arguments, XML-style tags, or function-calling schemas. Dedicated special tokens demarcate each stage of the trajectory, enabling the environment to parse and execute tool calls. A trajectory y is called valid if every tool call conforms to the required schema and formatting constraints (correct syntax, expected argument types, proper use of special tokens). We use $r_{\text{fmt}}(y) \in \{0, 1\}$ as the indicator of format validity. A trajectory y is called semantically correct if the trajectory produces the right final answer. We use $r_{\text{sem}}(y) \in \{0, 1\}$ as the indicator of semantic correctness.

Agent reinforcement learning. Reinforcement learning provides a natural objective for improving the agentic capabilities of LLMs, since it can directly optimize long-horizon task success rather than next-token prediction alone. In particular, RL enables the model to learn effective reasoning, tool selection, and action sequencing from trajectory-level feedback. For a given prompt x, let $y_1, \ldots, y_G$ denote a group of G trajectories sampled independently from the current policy $\pi_\theta(\cdot \mid x)$. Each trajectory y_i is assigned a normalized advantage A_i computed from the within-group rewards, satisfying $\sum_{i=1}^{G} A_i = 0$ in the standard centered case. The RL training objective is formulated as

$$\mathcal{L}_{\text{RL}}(\theta) \;=\; -\sum_{i=1}^{G} A_i \, \ell_i(\theta), \qquad \ell_i(\theta) := \log \pi_\theta(y_i \mid x) = \sum_{k=1}^{|y_i|} \log \pi_\theta\big(y_{i,k} \mid x,\, y_{i,<k}\big). \tag{1}$$

Minimizing $\mathcal{L}_{\text{RL}}$ encourages the policy to increase the likelihood of trajectories whose advantage exceeds the group mean and to suppress those that fall below it. This within-group relative update underlies several recent agent RL algorithms, including GRPO [19] and GIGPO [16], and serves as the foundation for the training-dynamics analysis in Section 3.

3 Cyclical Entropy Eruption in Agent Reinforcement Learning

Despite rapid progress in agent RL, relatively little attention has been paid to *how* the policy distribution evolves over the course of training. Understanding the dynamics matters practically: entropy dynamics govern the exploration-exploitation balance during training, and pathological entropy behavior can cause reward stagnation, the reinforcement of degenerate outputs, or outright training collapse. In this section, we identify a special and recurring entropy pattern in agent RL that we term *cyclical entropy eruption*.

3.1 Overview: The Cyclical Entropy Eruption Phenomenon

Experimental setup. We conduct agent RL training with GRPO [19] and GIGPO [16] on two tasks, AlfWorld [20] and WebShop [21][1]. AlfWorld is a text-based embodied benchmark for evaluating LLM agents on multi-step household tasks, where agents must interpret goals and interact through multiple turns to complete activities such as picking, cleaning, heating, cooling, and placing objects. WebShop is a realistic web-interaction benchmark that tests LLM agents in online shopping scenarios, requiring them to search, browse, compare, and purchase products on a simulated e-commerce website with a large and diverse product catalog. Additional results on a search-augmented QA task [14] and experiments with other backbone models are in Appendix B.1.

Cyclical entropy eruption in agent RL. As shown in Figure 1, the policy entropy in agent RL often undergoes a striking non-monotonic evolution: after an initial decrease, the entropy erupts sharply and then enters a recurring cycle of eruption and subsidence throughout training. This behavior is qualitatively different from the entropy dynamics typically observed in single-turn reasoning tasks such as math [17, 18]. Empirically, entropy eruption is often accompanied by reward stagnation or even degradation (Figure 1), as well as the emergence of toxic generation patterns such as sentence

[1]The implementation details follow `Agent-RL` [16] with the `VeRL` [22] framework, detailed in Section 4.2 and Appendix C.1

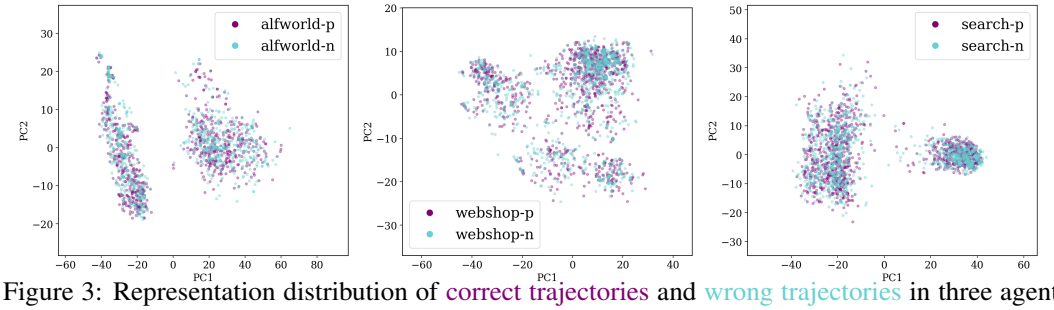

Figure 3: Representation distribution of correct trajectories and wrong trajectories in three agent tasks. -p and -n denote correct and wrong trajectories in the figures.

duplication and the incorrect use of format-specific special tokens (Figure 6). Understanding this phenomenon is therefore important for achieving stable and continual policy improvement in agent RL. We divide the entropy dynamics into three phases: (1) *entropy descent*, (2) *entropy eruption*, and (3) *entropy subsidence*. The remainder of this section analyzes each phase in turn, and explains the mechanism underlying its cyclical oscillation.

3.2 Phase 1: Entropy Descent and Validity Surge

In the first phase of training, agent RL exhibits a sharp entropy decrease as the policy rapidly reallocates probability mass toward a narrower set of valid trajectories. We explain this below.

Format learning dominates early training. We measure the fraction of sampled trajectories that satisfy formatting constraints and perform valid actions. As shown in Figure 2, we observe that the fraction of sampled trajectories that satisfy format-validity constraints surges and quickly saturates in the opening steps of training. This indicates that the policy first acquires protocol-level competence, including how to invoke tools and functions correctly, and how to satisfy schema and formatting constraints. This behavior is natural because while semantic competence can be partially inherited from pretraining, agent-specific formatting conventions—such as tool-call syntax, schema tokens, and special control markers—are often absent or weakly represented in the pretrained distribution. Consequently, the model begins RL with a much larger deficiency in format validity than in semantic understanding, so early gradient updates are dominated by correcting this format bottleneck, which in turn suppresses further gains from semantic learning until valid trajectories become sufficiently likely. We formalize this intuition into Lemma A.1, which shows that when the policy is weaker on agent-specific formatting than on semantic correctness, RL updates are dominated by increasing the format validity of trajectories. As a result, probability mass is rapidly shifted from a diverse set of malformed or invalid trajectories to a relatively concentrated subset of valid ones. We also found that mid-training can shrink phase 1 since mid-training already instills the basic formatting and validity constraints. We analyze this phenomenon in Appendix B.2.

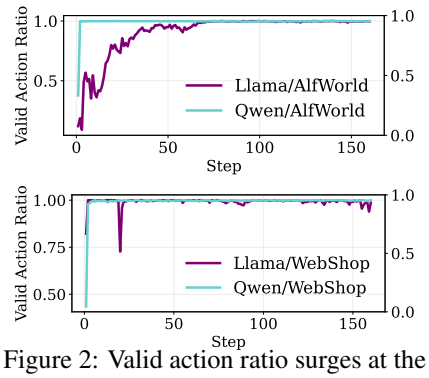

Figure 2: Valid action ratio surges at the beginning phase of agent RL.

3.3 Phase 2: Entropy Eruption

After the policy acquires format validity in Phase 1, training enters a regime in which entropy rises sharply rather than continuing to fall (see Figure 1). This behavior is distinctive to agent RL and stands in contrast with single-turn reasoning RL, where entropy typically remains low after its initial collapse. Below, we explain this unique phenomenon by two linked mechanisms: high representation similarity among trajectories and the gradient interference it induces.

Agent trajectories cluster together in the representation space. We begin by showing that *correct* and *wrong* agent trajectories are not well separated in the LLM representation space. Figure 3

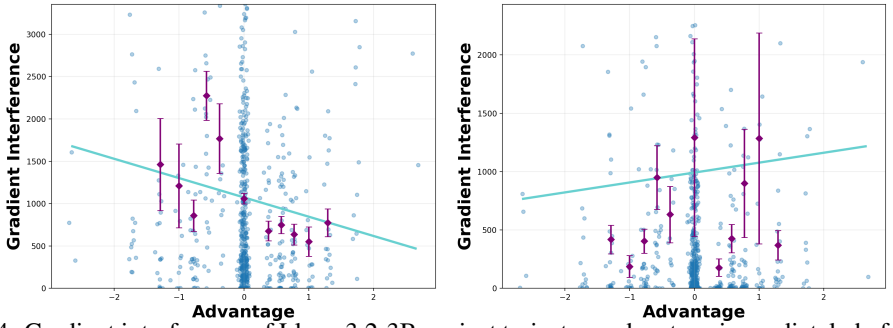

Figure 4: Gradient interference of Llama3.2-3B against trajectory advantage immediately before (left) and after (right) an entropy-eruption event on WebShop. Following the definition of c_j in Lemma 3.2, each point corresponds to trajectory j, with the x-axis representing its normalized advantage and the y-axis representing its average gradient inner product with all other trajectories in the rollout group

visualizes the last-layer hidden representations of Qwen-2.5-7B, averaged over response tokens, for sampled trajectories from three agent tasks: AlfWorld, WebShop, and Search-augmented QA. Across all three tasks, correct and wrong trajectories exhibit substantial overlap rather than forming clearly separated clusters, indicating strong representation similarity between positively and negatively rewarded trajectories. In Appendix B.3, we further show that this representation overlap is markedly more pronounced in agent tasks than in other LLM tasks. This high similarity results in important optimization consequences: negatively rewarded trajectories can interfere with correct trajectories during RL updates, thereby reducing the likelihood of valid trajectories and contributing to the onset of entropy eruption.

Representation similarity causes gradient interference. When trajectories are close in representation space, their per-token gradients align, so that suppressing one trajectory's likelihood inadvertently suppresses others. The following lemma makes this precise.

Lemma 3.1 (High representation similarity induces gradient interference)**.** *Fix a prompt x, and let $\theta = (W, \phi)$, where $W \in \mathbb{R}^{|V|\times d}$ is the output matrix and ϕ denotes all remaining model parameters. For any trajectory $y_i = (y_{i,1}, \ldots, y_{i,|y_i|})$, let $h_{i,k} := h_\phi(x, y_{i,<k}) \in \mathbb{R}^d$, $p_{i,k} := \pi_\theta(\cdot \mid x, y_{i,<k}) \in \mathbb{R}^{|V|}$, and let $e_{y_{i,k}} \in \mathbb{R}^{|V|}$ be the one-hot vector of token $y_{i,k}$. Define $\ell_i(\theta) := \sum_{k=1}^{|y_i|} \log p_{i,k}$. If $g_i := \nabla_\theta \ell_i(\theta)$ is the full gradient of the trajectory log-likelihood, then for any two trajectories y_i and y_j, their gradient interference can be formulated as*

$$\langle g_i, g_j \rangle = \underbrace{\sum_{k=1}^{T_i}\sum_{k'=1}^{T_j} \alpha_{i,k;j,k'} \langle h_{i,k}, h_{j,k'} \rangle}_{\textit{representation similarity}} + \underbrace{\sum_{k=1}^{T_i}\sum_{k'=1}^{T_j} \left\langle J_{i,k}^\top W^\top q_{i,k},\ J_{j,k'}^\top W^\top q_{j,k'} \right\rangle}_{\textit{relevant to LLM backbone geometry}}.$$

where $\alpha_{i,k;j,k'} := \langle q_{i,k}, q_{j,k'} \rangle = \langle e_{y_{i,k}} - p_{i,k},\ e_{y_{j,k'}} - p_{j,k'} \rangle$, and $J_{i,k} := \frac{\partial h_{i,k}}{\partial \phi} \in \mathbb{R}^{d\times \dim(\phi)}$ denote the Jacobian of the hidden state with respect to the non-output parameters.

This lemma shows that high cosine similarity between hidden states $h_{i,k}$ and $h_{j,k'}$ directly amplifies gradient interference between the two trajectories. This prediction is also supported empirically: in Figure 4, we compare gradient interference before and after entropy eruption, and show that negatively advantaged trajectories exhibit substantially stronger interference before the eruption, consistent with the proposed mechanism.

Gradient interference suppresses valid trajectories. Strong gradient interference means that when the RL objective suppresses a negatively-advantaged trajectory, it simultaneously pulls down the likelihood of other trajectories that share similar representations—including correct ones. The next lemma formalizes the likelihood reduction effect.

Lemma 3.2 (Gradient interference leads to likelihood decrease)**.** *Fix a prompt x and a group of sampled trajectories $y_1, \ldots, y_G$ with advantages $A_1, \ldots, A_G$. For each response y_i, let $g_i := \nabla_\theta \ell_i(\theta) = \nabla_\theta \log \pi_\theta(y_i \mid x)$ be the gradient of the log-likelihood of response y_i. We further define $c_j := \frac{1}{G}\sum_{i=1}^{G} \langle g_i, g_j \rangle$, which measures how strongly trajectory y_j's gradient aligns with the group on average. Under a gradient step $\frac{d}{dt}\theta = -\eta \nabla_\theta \mathcal{L}_{\mathrm{RL}}(\theta)$ with $\eta > 0$, the average log-likelihood*

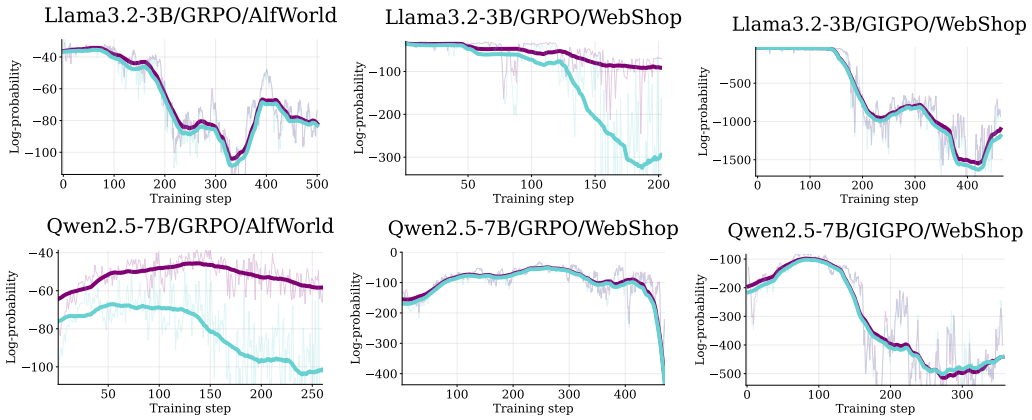

Figure 5: The likelihood of correct trajectories and valid trajectories decreases during agent RL. The reported log-probabilities are averages computed over all correct trajectories (purple) and all valid trajectories (orange) at each training step. More examples are shown in Appendix B.4.

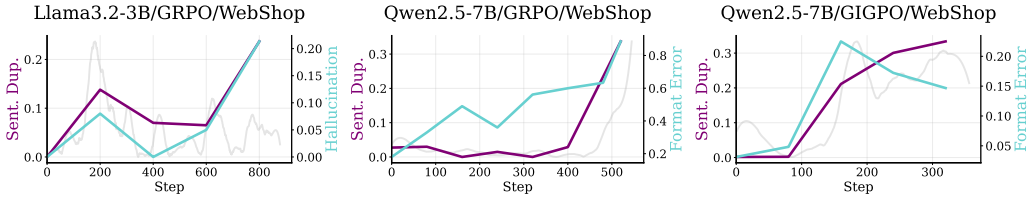

Figure 6: Degenerate patterns such as Sentence Duplication and Hallucination occur in entropy eruption. Sent. Dup. represents sentence duplication. y-axis represents percentage. Gray background curve is the policy entropy.

change of the group satisfies

$$\frac{1}{G}\sum_{i=1}^{G}\frac{d}{dt}\ell_i(\theta) \;=\; \sum_{j=1}^{G} A_j\, c_j.$$

This quantity is negative, i.e. the group's average likelihood decreases, when $\sum_{j=1}^{G} A_j\, c_j < 0$.

The lemma implies that the average likelihood of the response group could decrease when the gradient of a negatively advantaged trajectory y_j strongly interferes with other trajectories, namely when c_j is large. In that case, suppressing y_j also suppresses other trajectories that are semantically similar. Consequently, even the likelihood of correct responses in the group may be pulled down if they share strong representation similarity with negatively advantaged trajectories.

Likelihood decrease drives entropy growth. We next show, through the following lemma, that this likelihood decrease on valid trajectories increases entropy and thus drives the policy toward a flatter distribution.

Lemma 3.3 (Likelihood decline on high-probability responses increases entropy)**.** *Fix a prompt* x*, and let* $\pi_t(y) := \pi_{\theta_t}(y \mid x)$ *be the distribution at the training step* t*. If at time* t*,* $\mathrm{Cov}(\log \pi_t(y), \frac{\mathrm{d}}{\mathrm{d}t}\ell_t(y)) < 0$*, then* $\frac{\mathrm{d}}{\mathrm{d}t}H_t > 0$.

Once the valid-action ratio saturates in Phase 1, valid trajectories tend to occupy the high-likelihood region of the policy, but their likelihood then decreases during training (Figure 5), as demonstrated in Lemma 3.2. Thus, $\mathrm{Cov}(\log \pi_t(y), \frac{\mathrm{d}}{\mathrm{d}t}\ell_t(y)) < 0$ holds empirically. This implies, based on Lemma 3.3, that entropy must increase, leading to the eruption.

Entropy eruption flattens the policy and amplifies degenerate patterns. As entropy erupts, the policy distribution becomes flatter, making low-quality yet non-negligible trajectory modes more likely to be sampled during training. In experiments, we find that this phase is strongly associated with the emergence of degenerate patterns, such as *sentence duplication* and *hallucination*. To evaluate this phenomenon, we collect 1024 sampled trajectories at different training checkpoints and compute the

corresponding statistics. For sentence duplication, we split each response into sentence-level chunks and measure the fraction of sentences that are repeated within the same response. For hallucination, we measure the rate at which the model selects product attributes that do not actually appear on the webpage. As shown in Figure 6, across different backbones and both GRPO and GIGPO training, both metrics significantly rise during entropy-eruption periods, indicating that the entropy eruption regime systematically amplifies pathological behaviors and degrades the quality of sampled trajectories.

3.4 Phase 3: Entropy Subsidence and the Recurring Cycle

Why does the cycle repeat? Entropy eruption does not persist indefinitely. As the distribution flattens, the policy samples more diverse trajectories, which in turn reduces representation similarity among them. By Lemma 3.1 and Lemma 3.2, this reduction in trajectory similarity means less gradient interference, so the suppressive effect of negatively-advantaged trajectories on correct ones is reduced and the likelihood of correct trajectories recovers. As a result, training gradually shifts from a highly entropic exploration back toward reinforcing correct responses. However, the subsidence phase carries the seed of the next eruption. As the policy reconcentrates on the correct trajectories discovered during the high-entropy phase, representation similarity rises once again, restoring the conditions that triggered Phase 2. The dynamics therefore form a self-perpetuating cycle, illustrated in Figure 7: convergence raises similarity → similarity causes eruption → eruption increases diversity → diversity enables subsidence → and subsidence leads back to convergence.

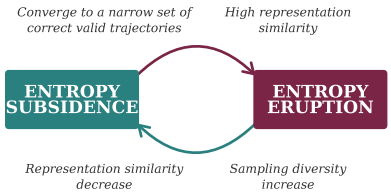

Figure 7: The cyclical dynamics between entropy subsidence and entropy eruption.

The lasting damage of entropy eruption. Although entropy subsides and valid-trajectory likelihood recovers, the damage introduced during eruption can persist. When the distribution is flat, degenerate trajectories, such as those containing sentence duplication and hallucination, are sampled more frequently. If such trajectories still produce a correct final answer and receive a positive reward, the policy reinforces these patterns. As shown in Figure 6, degenerate behaviors acquired during eruption often persist into subsequent subsidence phases and can even be amplified as the policy concentrates on the (now-corrupted) correct trajectories. We provide additional analysis and statistics in Appendix B.6. In more severe cases, entropy eruption can trigger a semantic collapse and completely derail RL training. As shown in Figure 12, for Llama3.2-1B on WebShop, an entropy eruption is followed by a collapse of the reward curve to zero.

4 How to Alleviate Entropy Eruption?

While the primary focus of this paper is on understanding the training dynamics of agent RL, we show here that our analysis directly suggests a simple practical intervention—and that even a minimal approach motivated by the theory can meaningfully stabilize training and improve performance. We expect future work to deepen this line of inquiry with more sophisticated methods; our goal here is to demonstrate that the theoretical understanding developed above is actionable and guides practice.

4.1 SEAL: Separation-Enhanced Agent Learning

The analysis in Section 3 traces entropy eruption to a single root cause: high representation similarity between correct and incorrect trajectories amplifies gradient interference, suppressing valid-trajectory likelihood and flattening the distribution. A natural remedy is to push these two trajectory classes apart in representation space. We propose **SEAL** (**S**eparation-**E**nhanced **A**gent **L**earning), a lightweight modification to the original RL objective, with no additional cost at inference time.

For each sampled response $y_i = (y_{i,1}, \ldots, y_{i,T_i})$, let $h_{i,t} \in \mathbb{R}^d$ denote the hidden representation of the t-th response token produced by the policy model. We attach a binary classifier on top of these token representations, $p_{i,t} = \sigma(\mathrm{MLP}(h_{i,t}))$, where $p_{i,t}$ is the predicted probability that the token comes from a correct trajectory. We assign every token in trajectory y_i the same binary label $z_i \in \{0, 1\}$, where $z_i = 1$ if y_i is correct and $z_i = 0$ otherwise. The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{RL}} + \alpha \mathcal{L}_{\mathrm{SEAL}}, \tag{2}$$

Table 1: Comparison of GRPO and GIGPO performance with and without SEAL on AlfWorld and WebShop for Qwen2.5- and Llama3.2-series models.

Model	Methods	ALFWORLD							WEBSHOP	
		Pick	Look	Clean	Heat	Cool	Pick2	**Avg.**	Score	**Succ.**
Llama3.2 -1B	GRPO	81.48	87.50	92.86	93.33	70.83	69.57	82.60	00.00	00.00
	GRPO + SEAL	96.43	84.62	100.00	83.33	87.50	84.00	**89.31**	92.42	**79.69**
	GIGPO	91.43	75.00	100.00	92.86	57.14	95.83	85.38	86.09	75.00
	GIGPO + SEAL	83.33	94.25	95.65	100.00	95.46	90.00	**93.12**	89.39	**78.90**
Qwen2.5 -1.5B	GRPO	90.91	87.50	83.87	91.67	84.85	84.21	87.17	85.97	73.04
	GRPO + SEAL	97.22	88.89	96.00	100.00	93.10	91.30	**94.42**	89.92	**81.25**
	GIGPO	93.75	86.77	84.62	84.62	95.00	77.27	87.01	86.11	71.88
	GIGPO + SEAL	100.00	85.00	91.68	75.00	95.83	91.30	**89.80**	91.14	**80.47**
Llama3.2 -3B	GRPO	96.88	100.00	96.15	76.92	90.00	86.36	91.05	87.15	63.28
	GRPO + SEAL	97.22	95.00	95.46	86.67	100.00	100.00	**95.73**	89.86	**77.73**
	GIGPO	94.44	100.00	96.67	90.00	76.47	96.29	92.31	89.07	71.88
	GIGPO + SEAL	96.30	100.00	100.00	86.67	100.00	86.96	**94.99**	90.26	**78.91**
Qwen2.5 -7B	GRPO	100.00	71.42	91.30	86.67	80.95	55.56	80.98	90.81	76.95
	GRPO + SEAL	94.60	85.71	100.00	86.67	85.74	50.00	**83.79**	91.13	**80.08**
	GIGPO	96.55	84.61	95.65	94.44	80.77	85.71	89.62	89.61	75.00
	GIGPO + SEAL	94.60	85.71	100.00	86.67	85.71	91.67	**90.72**	90.23	**79.69**

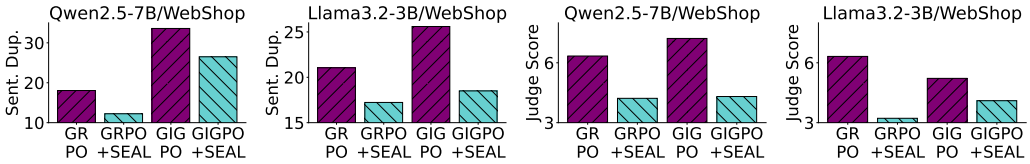

Figure 8: Comparison of sampling quality in terms of sentence duplication ratio (Sent. Dup.%) and GPT-4-based degeneracy score. SEAL reduces degenerate patterns in agent RL.

with the binary cross-entropy loss for the classification task as follows,

$$\mathcal{L}_{\text{SEAL}} = -\frac{1}{\sum_{i=1}^{G} T_i} \sum_{i=1}^{G} \sum_{t=1}^{T_i} \Big[z_i \log p_{i,t} + (1 - z_i) \log(1 - p_{i,t}) \Big]. \tag{3}$$

This auxiliary objective encourages token representations from correct trajectories to be distinguishable from those of incorrect trajectories, thereby reducing harmful representation overlap between them. As a result, the interference from incorrect trajectories on correct ones is weakened, which mitigates the likelihood decrease of valid trajectories and helps suppress entropy eruption.

4.2 Experimental Settings

We first train the LLM agents on two challenging benchmarks: ALFWorld [20] and WebShop [21]. ALFWorld is an embodied environment designed to assess the ability of LLM agents to perform multi-step decision-making. In each episode, the agent receives a text goal and must accomplish it through multi-turn interaction with the environment. It includes 3,827 task instances across six categories of common household activities: Pick & Place (Pick), Examine in Light (Look), Clean & Place (Clean), Heat & Place (Heat), Cool & Place (Cool), and Pick Two & Place (Pick2). WebShop is a complex, web-based interactive environment designed to test the LLM agents in realistic online shopping scenarios. To complete the task, the agent must interact with a simulated HTML-based shopping website to search for, navigate to, and ultimately purchase a suitable item. It contains over 1.1 million products and 12k user instructions, providing a rich and diverse action space. We also conduct further experiments on search-augmented QA in Appendix C.5. We use Qwen2.5-1.5B/7B-Instruct [23] and Llama3.2-1B/3B-Instruct [24] as our base models. Since Llama3.2 instruct models cannot sample valid trajectories throughout the RL training, we perform mid-training on Llama3.2 with a subset of OpenManus [25]. For hyperparameter settings, we mostly follow GIGPO [16]. For fair comparison, all methods use the same learning rate, batch size, and training steps, and each pair of runs with and without SEAL is evaluated after the same amount of training. Implementation details are in Appendix C.1.

4.3 SEAL Improves the Performance and Suppresses Degenerate Patterns.

SEAL improves the performance of agent RL across different tasks and backbones. Table 1 shows that SEAL consistently improves agent RL performance across both tasks and model backbones. On AlfWorld, adding SEAL generally raises the average success over subtasks for both GRPO and GIGPO, with especially large gains for smaller models such as Llama3.2-1B and Qwen2.5-1.5B. On WebShop, SEAL also leads to clear improvements in both score and success rate across the Qwen and Llama series. Notably, SEAL turns a near-failed run of Llama3.2-1B on WebShop into strong performance. Overall, these results indicate that SEAL is a robust and effective enhancement for agent RL, delivering gains across different backbones, training algorithms, and task environments.

SEAL alleviates entropy eruption and suppresses degenerate patterns. As shown in Appendix C.2, SEAL consistently mitigates the entropy-eruption phenomenon across different tasks, backbones, and RL algorithms. This stabilization also translates into better sample quality. In Figure 8, we evaluate degenerate behavior on WebShop using two complementary metrics: the sentence-duplication ratio and an LLM-as-a-judge degenerate score with GPT-4. Across both Qwen2.5-7B and Llama3.2-3B, SEAL consistently reduces sentence duplication for both GRPO and GIGPO. These results suggest that by preventing the policy from entering an excessively diffuse high-entropy regime, SEAL not only improves final task performance but also suppresses pathological generation behaviors during training.

Ablations. SEAL is not overly sensitive to the auxiliary-loss weight α. As shown in Appendix C.3, SEAL remains effective over a broad range of α, with the best or near-best performance typically achieved at intermediate values. Importantly, across a wide range of α, SEAL generally remains better than the corresponding baselines without SEAL, suggesting that its gains are robust rather than dependent on delicate hyperparameter tuning.

5 Related Works

Reinforcement learning for LLM agents. LLM agents [26, 27, 28, 4] extend static text generation to sequential decision making, requiring models to plan, invoke tools, interact with environments, and maintain state over long horizons. Thus, reinforcement learning is a natural framework for improving agent behavior from outcome feedback. Recent work [29, 14, 30, 15] has largely built on PPO/GRPO-style objectives [31, 19] originally developed for single-turn reasoning and alignment, while adapting them to the agent setting with techniques such as context compression [32, 33, 34, 35, 36], decoupled execution-training pipelines, and memory-aware optimization [37, 38, 39, 40]. These advances have substantially improved empirical performance, but the optimization dynamics of agent RL remain poorly understood. Our work complements this line of research by studying how the policy distribution evolves during training and by identifying the phenomenon of cyclical entropy eruption.

Entropy dynamics in LLM RL. Policy entropy is a central quantity in policy-gradient RL, governing the exploration-exploitation trade-off: higher entropy supports broader search, whereas rapid entropy reduction can lead to overly deterministic behavior [18, 41, 42]. In LLM RL, prior work has shown that entropy often collapses early in training and then remains persistently low, reducing response diversity and contributing to performance saturation [17, 18, 43]. To address this issue, existing methods have introduced entropy regularization [17, 44] or alternative distribution-matching objectives [45, 46]. A few recent works observe entropy fluctuations in LLM RL [47, 48], but do not explain the mechanism behind them. In contrast, we focus on the agent setting, where long-horizon trajectories, tool interactions, and validity constraints induce qualitatively different dynamics. We provide a theoretical and empirical account linking entropy eruption to representation similarity and gradient interference, and show that the resulting behavior is inherently cyclical.

6 Conclusion

In this work, we study the training dynamics of agent RL and identify a previously underexplored phenomenon where agent RL exhibits recurring cycles of entropy descent, eruption, and subsidence. We term it *cyclical entropy eruption*. Through theoretical and empirical analysis, we show that this behavior is closely related to high representation similarity among trajectories, which induces

harmful gradient interference, suppresses the likelihood of valid trajectories, and amplifies degenerate patterns such as sentence duplication and hallucination. Building on this understanding, we propose SEAL, a simple training objective that encourages better separation between correct and incorrect trajectories in representation space. Despite its simplicity, SEAL effectively stabilizes training, reduces entropy eruption and degenerate behaviors, improving downstream agent performance across multiple benchmarks, model families, and RL algorithms.

Acknowledgement

This work is supported in part by the AFOSR Young Investigator Program under award number FA9550-23-1-0184, the Office of Naval Research (ONR) under award number N000142612508, the National Science Foundation under awards IIS-2237037 and IIS-2331669, Schmidt Sciences Foundation, Open Philanthropy (now Coefficient Giving), Alfred P. Sloan Fellowship, and gifts from Google and Amazon. Shawn Im is also supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 2137424. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 6864–6890. Association for Computational Linguistics, 2024.

[2] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samual Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[3] Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vázquez, Nicolas Chapados, and Alexandre Lacoste. Workarena: How capable are web agents at solving common knowledge work tasks? In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 11642–11662. PMLR / OpenReview.net, 2024.

[4] John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[5] John Yang, Kilian Leret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. Swe-smith: Scaling data for software engineering agents. *CoRR*, abs/2504.21798, 2025.

[6] Antonis Antoniades, Albert Örwall, Kexun Zhang, Yuxi Xie, Anirudh Goyal, and William Yang Wang. Swe-search: Enhancing software agents with monte carlo tree search and iterative refinement. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[7] Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. Training software engineering agents and verifiers with swe-gym. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

[8] Shanghua Gao, Ada Fang, Yepeng Huang, Valentina Giunchiglia, Ayush Noori, Jonathan Richard Schwarz, Yasha Ektefaie, Jovana Kondic, and Marinka Zitnik. Empowering biomedical discovery with ai agents. *Cell*, 187(22):6125–6151, 2024.

[9] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob N. Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *CoRR*, abs/2408.06292, 2024.

[10] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

[11] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, Weiyu Liu, Percy Liang, Li Fei-Fei, Jiayuan Mao, and Jiajun Wu. Embodied agent interface: Benchmarking llms for embodied decision making. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[12] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

[13] Chan Hee Song, Brian M. Sadler, Jiaman Wu, Wei-Lun Chao, Clayton Washington, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 2986–2997. IEEE, 2023.

[14] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025.

[15] Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 7909–7928. Association for Computational Linguistics, 2025.

[16] Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[17] Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *CoRR*, abs/2505.22617, 2025.

[18] Han Shen. On entropy control in LLM-RL algorithms. *CoRR*, abs/2509.03493, 2025.

[19] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024.

[20] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[21] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[22] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[23] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024.

[24] Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

[25] OpenManus-RL Team. Openmanus-rl: Open platform for generalist llm reasoning agents with rl optimization, 2025.

[26] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[27] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[28] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[29] Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: understanding self-evolution in LLM agents via multi-turn reinforcement learning. *CoRR*, abs/2504.20073, 2025.

[30] Cheng Qian, Emre Can Acikgoz, Qi He, Hongru WANG, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[31] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[32] Miao Lu, Weiwei Sun, Weihua Du, Zhan Ling, Xuesong Yao, Kang Liu, and Jiecao Chen. Scaling llm multi-turn rl with end-to-end summarization-based context management. *arXiv preprint arXiv:2510.06727*, 2025.

[33] Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, et al. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*, 2025.

[34] Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. Acon: Optimizing context compression for long-horizon llm agents. *arXiv preprint arXiv:2510.00615*, 2025.

[35] Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling long-horizon llm agent via context-folding. *arXiv preprint arXiv:2510.11967*, 2025.

[36] Qiaoyu Tang, Hao Xiang, Le Yu, Bowen Yu, Yaojie Lu, Xianpei Han, Le Sun, WenJuan Zhang, Pengbo Wang, Shixuan Liu, et al. Beyond turn limits: Training deep search agents with dynamic context window. *arXiv preprint arXiv:2510.08276*, 2025.

[37] Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. Memagent: Reshaping long-context LLM with multi-conv rl-based memory agent. *CoRR*, abs/2507.02259, 2025.

[38] Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *CoRR*, abs/2508.19828, 2025.

[39] Yaorui Shi, Yuxin Chen, Siyuan Wang, Sihang Li, Hengxing Cai, Qi Gu, Xiang Wang, and An Zhang. Look back to reason forward: Revisitable memory for long-context llm agents. *arXiv preprint arXiv:2509.23040*, 2025.

[40] Yuxiang Zhang, Jiangming Shu, Ye Ma, Xueyuan Lin, Shangxi Wu, and Jitao Sang. Memory as action: Autonomous context curation for long-horizon agentic tasks. *arXiv preprint arXiv:2510.12635*, 2025.

[41] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Proceedings of Machine Learning Research, pages 1856–1865. PMLR, 2018.

[42] Safa Messaoud, Billel Mokeddem, Zhenghai Xue, Linsey Pang, Bo An, Haipeng Chen, and Sanjay Chawla. S 2 ac: Energy-based reinforcement learning with stein soft actor critic. In *International Conference on Learning Representations*, volume 2024, pages 22323–22336, 2024.

[43] Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may or may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

[44] Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor, editors, *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 30377–30385. AAAI Press, 2026.

[45] Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, et al. Flowrl: Matching reward distributions for llm reasoning. *arXiv preprint arXiv:2509.15207*, 2025.

[46] Wendi Li and Sharon Li. Lad: Learning advantage distribution for reasoning. *arXiv preprint arXiv:2602.20132*, 2026.

[47] Wujiang Xu, Wentian Zhao, Zhenting Wang, Yu-Jhe Li, Can Jin, Mingyu Jin, Kai Mei, Kun Wan, and Dimitris N Metaxas. Epo: Entropy-regularized policy optimization for llm agents reinforcement learning. *arXiv preprint arXiv:2509.22576*, 2025.

[48] Junkang Wu, Kexin Huang, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. Quantile advantage estimation for entropy-safe reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

[49] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ULTRAFEEDBACK: boosting language models with scaled AI feedback. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 9722–9744. PMLR / OpenReview.net, 2024.

A Proofs

Lemma A.1 (Format-gated learning dominates training at the beginning.)**.** *Suppose the trajectory reward factorizes as $r(y) = r_{\mathrm{fmt}}(y)\, r_{\mathrm{sem}}(y)$, where $r_{\mathrm{fmt}}(y) \in \{0,1\}$ and $r_{\mathrm{sem}}(y) \in \{0,1\}$ denote the format and semantic correctness. Let $p_\theta^{\mathrm{fmt}}(x) = \sum_{y:\, r_{\mathrm{fmt}}(y)=1} \pi_\theta(y \mid x)$, and $p_\theta^{\mathrm{sem}}(x) = \sum_{y:\, r_{\mathrm{sem}}(y)=1} \pi_\theta(y \mid x)$ denote the total probability the policy assigns to format-valid and semantically correct trajectories, respectively. If $\mathbb{E}_x p_\theta^{\mathrm{sem}}(x) > \mathbb{E}_x p_\theta^{\mathrm{fmt}}(x)$,* i.e.*, the policy is weaker on agent-specific formatting than on semantic correctness, RL updates are dominated by increasing the format validity of trajectories.*

Proof. Let

$$F := \{y :\, r_{\mathrm{fmt}}(y) = 1\}, \qquad S := \{y :\, r_{\mathrm{sem}}(y) = 1\}.$$

Since $r_{\mathrm{fmt}}, r_{\mathrm{sem}} \in \{0,1\}$, we have

$$r(y) = r_{\mathrm{fmt}}(y)\, r_{\mathrm{sem}}(y) = \mathbf{1}\{y \in F \cap S\}.$$

Hence, for RL objective $J_x(\theta) := \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r(y)] = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r_{\mathrm{fmt}}(y)\, r_{\mathrm{sem}}(y)]$, we have $J_x(\theta) = \sum_{y\in F\cap S} \pi_\theta(y \mid x)$. Note $p_\theta^{\mathrm{fmt}}(x) > 0$ and $p_\theta^{\mathrm{sem}}(x) > 0$, and define the conditional success rates

$$q_\theta^{\mathrm{fmt}|\mathrm{sem}}(x) := \mathbb{P}_\theta\big(r_{\mathrm{fmt}} = 1 \mid r_{\mathrm{sem}} = 1, x\big), \qquad q_\theta^{\mathrm{sem}|\mathrm{fmt}}(x) := \mathbb{P}_\theta\big(r_{\mathrm{sem}} = 1 \mid r_{\mathrm{fmt}} = 1, x\big).$$

By elementary probability,

$$J_x(\theta) = \mathbb{P}_\theta(F \cap S \mid x) = \mathbb{P}_\theta(S \mid x)\, \mathbb{P}_\theta(F \mid S, x) = p_\theta^{\mathrm{sem}}(x)\, q_\theta^{\mathrm{fmt}|\mathrm{sem}}(x),$$

and similarly

$$J_x(\theta) = \mathbb{P}_\theta(F \mid x)\, \mathbb{P}_\theta(S \mid F, x) = p_\theta^{\mathrm{fmt}}(x)\, q_\theta^{\mathrm{sem}|\mathrm{fmt}}(x).$$

Differentiating both factorizations gives

$$\nabla_\theta J_x = q_\theta^{\mathrm{fmt}|\mathrm{sem}}\, \nabla_\theta p_\theta^{\mathrm{sem}} + p_\theta^{\mathrm{sem}}\, \nabla_\theta q_\theta^{\mathrm{fmt}|\mathrm{sem}},$$

and

$$\nabla_\theta J_x = q_\theta^{\mathrm{sem}|\mathrm{fmt}}\, \nabla_\theta p_\theta^{\mathrm{fmt}} + p_\theta^{\mathrm{fmt}}\, \nabla_\theta q_\theta^{\mathrm{sem}|\mathrm{fmt}}.$$

The first identity shows that changes in format validity enter through the term $p_\theta^{\mathrm{sem}}\, \nabla_\theta q_\theta^{\mathrm{fmt}|\mathrm{sem}}$, so format improvement is weighted by the current semantic-correctness mass p_θ^{sem}. Likewise, the second identity shows that changes in semantic correctness enter through $p_\theta^{\mathrm{fmt}}\, \nabla_\theta q_\theta^{\mathrm{sem}|\mathrm{fmt}}$, so semantic improvement is weighted by the current format-valid mass p_θ^{fmt}. Therefore, when $E_x p_\theta^{sem}(x) > E_x p_\theta^{fmt}(x)$, the gating of semantic improvement would be more severe than format improvement, and RL updates are thus dominated by increasing the probability of protocol-valid trajectories. □

Lemma A.2 (Restatement of Lemma 3.1)**.** *Fix a prompt x, and let $\theta = (W, \phi)$, where $W \in \mathbb{R}^{|V|\times d}$ is the output matrix and ϕ denotes all remaining model parameters. For any trajectory $y_i = (y_{i,1}, \ldots, y_{i,|y_i|})$, define $\ell_i(\theta) := \sum_{k=1}^{|y_i|} \log \pi_\theta(y_{i,k} \mid x, y_{i,<k})$. For each token position (i,k), let $h_{i,k} := h_\phi(x, y_{i,<k}) \in \mathbb{R}^d$ denote the output hidden states, $p_{i,k} := \pi_\theta(\cdot \mid x, y_{i,<k}) \in \mathbb{R}^{|V|}$ denote the output distribution, and $e_{i,k} \in \mathbb{R}^{|V|}$ be the one-hot label of token $y_{i,k}$. If $g_i := \nabla_\theta \ell_i(\theta)$ is the full gradient of the trajectory log-likelihood, then for any two trajectories y_i and y_j,*

$$\langle g_i, g_j \rangle = \sum_{k=1}^{T_i} \sum_{k'=1}^{T_j} \alpha_{i,k;j,k'}\, \langle h_{i,k}, h_{j,k'} \rangle + \sum_{k=1}^{T_i} \sum_{k'=1}^{T_j} \Big\langle J_{i,k}^\top W^\top q_{i,k},\, J_{j,k'}^\top W^\top q_{j,k'} \Big\rangle.$$

where $\alpha_{i,k;j,k'} := \langle q_{i,k}, q_{j,k'} \rangle = \langle e_{i,k} - p_{i,k},\, e_{j,k'} - p_{j,k'} \rangle$., and $J_{i,k} := \frac{\partial h_{i,k}}{\partial \phi} \in \mathbb{R}^{d\times \dim(\phi)}$ denote the Jacobian of the hidden state with respect to the non-output parameters.

Proof. For a fixed token position (i,k), write $z_{i,k} := W h_{i,k} \in \mathbb{R}^{|V|}$. Then $\pi_\theta(\cdot \mid x, y_{i,<k}) = \mathrm{softmax}(z_{i,k})$, and therefore

$$\nabla_{z_{i,k}} \log \pi_\theta(y_{i,k} \mid x, y_{i,<k}) = e_{y_{i,k}} - \mathrm{softmax}(z_{i,k}) = e_{y_{i,k}} - p_{i,k} = q_{i,k}.$$

By the chain rule, the gradient with respect to the output matrix W is

$$\nabla_W \log \pi_\theta(y_{i,k} \mid x, y_{i,<k}) = q_{i,k} h_{i,k}^\top.$$

Similarly, since $z_{i,k} = W h_{i,k}$ and $h_{i,k} = h_\phi(x, y_{i,<k})$, we have

$$\nabla_\phi \log \pi_\theta(y_{i,k} \mid x, y_{i,<k}) = J_{i,k}^\top W^\top q_{i,k}.$$

Summing over all token positions of trajectory y_i yields

$$\nabla_W \ell_i(\theta) = \sum_{k=1}^{T_i} q_{i,k} h_{i,k}^\top, \qquad \nabla_\phi \ell_i(\theta) = \sum_{k=1}^{T_i} J_{i,k}^\top W^\top q_{i,k}.$$

Under the decomposition $\theta = (\mathrm{vec}(W), \phi)$, the inner product between the full gradients decomposes as

$$\langle g_i, g_j \rangle = \langle \nabla_W \ell_i(\theta), \nabla_W \ell_j(\theta) \rangle + \langle \nabla_\phi \ell_i(\theta), \nabla_\phi \ell_j(\theta) \rangle .$$

where we define the inner product between two matrices $\langle A, B \rangle$ as $\mathrm{tr}(A^T B)$. Expanding the first term bilinearly gives

$$\langle \nabla_W \ell_i(\theta), \nabla_W \ell_j(\theta) \rangle_F = \sum_{k=1}^{T_i} \sum_{k'=1}^{T_j} \left\langle q_{i,k} h_{i,k}^\top,\ q_{j,k'} h_{j,k'}^\top \right\rangle = \sum_{k=1}^{T_i} \sum_{k'=1}^{T_j} \langle q_{i,k}, q_{j,k'} \rangle \, \langle h_{i,k}, h_{j,k'} \rangle.$$

where we use the identity $\langle ab^\top, cd^\top \rangle = \langle a, c \rangle \langle b, d \rangle$. Likewise, expanding the ϕ-block gives

$$\langle \nabla_\phi \ell_i(\theta), \nabla_\phi \ell_j(\theta) \rangle = \sum_{k=1}^{T_i} \sum_{k'=1}^{T_j} \left\langle J_{i,k}^\top W^\top q_{i,k},\ J_{j,k'}^\top W^\top q_{j,k'} \right\rangle.$$

Adding the two contributions proves the claim. □

Meaning of the terms. The first double sum,

$$\sum_{k=1}^{T_i} \sum_{k'=1}^{T_j} \alpha_{i,k;j,k'} \, \langle h_{i,k}, h_{j,k'} \rangle,$$

is exactly the W-block interaction from the constrained / output-layer-only analysis. Here

$$\alpha_{i,k;j,k'} = \left\langle e_{y_{i,k}} - p_{i,k},\ e_{y_{j,k'}} - p_{j,k'} \right\rangle$$

measures how aligned the two token-level residual vectors are, while $\langle h_{i,k}, h_{j,k'} \rangle$ measures alignment between the corresponding hidden states. The second double sum,

$$\sum_{k=1}^{T_i} \sum_{k'=1}^{T_j} \left\langle J_{i,k}^\top W^\top q_{i,k},\ J_{j,k'}^\top W^\top q_{j,k'} \right\rangle,$$

is the additional contribution coming from the trainable non-output parameters ϕ. It captures how token-level errors backpropagate through the network into the hidden states. The factor $W^\top q_{i,k}$ is the gradient of the token log-probability with respect to the hidden state $h_{i,k}$, and $J_{i,k}^\top$ maps that hidden-state gradient back to parameter space through the Jacobian of the hidden state with respect to ϕ.

Lemma A.3 (Restatement of Lemma 3.2)**.** *Fix a prompt x and a group of sampled responses $y_1, \ldots, y_G$. For each response y_i, define its log-likelihood under the current policy by $\ell_i(\theta) := \log \pi_\theta(y_i \mid x)$. Let A_i denote the (normalized) advantage assigned to response y_i, and consider the local RL objective $L_{\mathrm{RL}}(\theta) := -\sum_{i=1}^G A_i \, \ell_i(\theta)$. For each i, let $g_i := \nabla_\theta \ell_i(\theta)$ be the gradient of the log-likelihood of response y_i, and define $c_j := \frac{1}{G} \sum_{i=1}^G \langle g_i, g_j \rangle$. The quantity c_j measures how strongly the update direction induced by response y_j aligns, on average, with the gradients of all responses in the group; equivalently, it is a group-level* similarity centrality *of response y_j. Suppose we perform a gradient descent step $\frac{\mathrm{d}}{\mathrm{d}t}\theta = -\eta \nabla_\theta L_{\mathrm{RL}}(\theta)$ with $\eta > 0$. The average log-likelihood change of the group is negative when $\sum_{j=1}^G A_j c_j < 0$*

Proof. Denote $K_{ij} := \langle g_i, g_j \rangle$. The matrix K is the trajectory-level interaction kernel. From the objective,

$$\nabla_\theta L_{\mathrm{RL}}(\theta) = -\sum_{j=1}^{G} A_j \nabla_\theta \ell_j(\theta) = -\sum_{j=1}^{G} A_j g_j.$$

Hence under gradient flow,

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta = -\eta \nabla_\theta L_{\mathrm{RL}} = \eta \sum_{j=1}^{G} A_j g_j.$$

Therefore, for each trajectory i,

$$\frac{\mathrm{d}}{\mathrm{d}t} \log \pi(y_i|x) = \langle g_i, \frac{\mathrm{d}\theta}{\mathrm{d}t} \rangle = \eta \sum_{j=1}^{G} A_j \langle g_i, g_j \rangle.$$

Averaging over i gives

$$\frac{1}{G}\sum_{i=1}^{G} \frac{\mathrm{d}}{\mathrm{d}t} \log \pi(y_i|x) = \frac{1}{G}\sum_{i=1}^{G} \eta \sum_{j=1}^{G} A_j \langle g_i, g_j \rangle = \eta \sum_{j=1}^{G} A_j \left(\frac{1}{G}\sum_{i=1}^{G} \langle g_i, g_j \rangle \right) = \eta \sum_{j=1}^{G} A_j c_j.$$

This proves the claim. □

Empirical evidence consistent with the mechanism. To further probe the causal role of gradient interference, we compare the relationship between trajectory advantage and gradient interference immediately before and after an entropy-eruption event in Figure 4. Before the eruption, trajectories with negative advantage exhibit systematically larger gradient interference than those with positive advantage, as shown by the clear negative trend in the fitted line and the binned statistics. This indicates that poorly rewarded trajectories are more strongly coupled to the rest of the sampled group and therefore exert a larger suppressive effect on other trajectories, including valid or correct ones. This observation is consistent with our theory: when negatively advantaged trajectories are more central in the interaction kernel, they can drive a decrease in the average likelihood of the group and trigger entropy eruption. In contrast, after the eruption, this pattern is substantially weakened and even partially reversed: the dependence of gradient interference on advantage becomes much flatter, and negatively advantaged trajectories no longer dominate the interaction structure. This change is consistent with the role of entropy eruption in increasing trajectory diversity and reducing representation similarity, which in turn alleviates harmful interference. Taken together, the before-and-after comparison provides temporal evidence supporting our causal claim: strong interference from negatively advantaged trajectories appears before entropy eruption and is mitigated after the eruption occurs.

Corollary A.1 (Negatively-Advantaged trajectories reduce all trajectory likelihoods)**.** *Let $\mathcal{Y}^+$ be the set of indices corresponding to valid and correct trajectories and $\mathcal{Y}^-$ be the remaining trajectory indices. Suppose that the advantages are centered, $A_j > 0$ for all $j \in \mathcal{Y}^+$ and $A_j < 0$ for all $j \in \mathcal{Y}^-$. Given that all the gradients are similar, $\langle g_i, g_j \rangle > \|g_i\|\|g_j\|\delta$ for $1 \geq \delta > 0$ for any $i, j \in [G]$ and the negatively-advantaged trajectories dominate the gradient, $\delta \min_{j\in\mathcal{Y}^-} \|g_j\| > \max_{j\in\mathcal{Y}^+} \|g_j\|$, $\sum_{j=1}^{G} A_j \langle g_i, g_j \rangle < 0$ for all $i \in [G]$ and the likelihoods of the all trajectories decrease.*

Proof. From the proof of Lemma 3.2, we know that

$$\frac{\mathrm{d}}{\mathrm{d}t} \log \pi(y_i|x) = \eta \sum_{j=1}^{G} A_j \langle g_i, g_j \rangle.$$

We will decompose this sum into a $\mathcal{Y}^+$ component and a $\mathcal{Y}^-$ component.

$$\frac{\mathrm{d}}{\mathrm{d}t} \log \pi(y_i|x) = \eta \sum_{j\in\mathcal{Y}^+} A_j \langle g_i, g_j \rangle + \eta \sum_{j\in\mathcal{Y}^-} A_j \langle g_i, g_j \rangle.$$

We can upper bound the first term using Holder's inequality

$$\frac{\mathrm{d}}{\mathrm{d}t} \log \pi(y_i|x) \leq \eta (\sum_{j\in\mathcal{Y}^+} A_j) \max_{j\in\mathcal{Y}^+} \langle g_i, g_j \rangle + \eta \sum_{j\in\mathcal{Y}^-} A_j \langle g_i, g_j \rangle.$$

and since all the A_j in second term are negative and all the inner products are positive, we can upper bound the second term as

$$\eta(\sum_{j\in\mathcal{Y}^-} A_j)\min_{j\in\mathcal{Y}^-}\langle g_i, g_j\rangle$$

Then, as the advantages are centered, we know that $\sum_{j\in\mathcal{Y}^+} A_j = -\sum_{j\in\mathcal{Y}^-} A_j$ and we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\log\pi(y_i|x) \le \eta(\sum_{j\in\mathcal{Y}^+} A_j)(\max_{j\in\mathcal{Y}^+}\langle g_i, g_j\rangle - \min_{j\in\mathcal{Y}^-}\langle g_i, g_j\rangle).$$

By Cauchy-Schwarz and by the assumptions, we have that

$$\frac{\mathrm{d}}{\mathrm{d}t}\log\pi(y_i|x) \le \eta(\sum_{j\in\mathcal{Y}^+} A_j)(\|g_i\|\max_{j\in\mathcal{Y}^+}\|g_j\| - \delta\|g_i\|\min_{j\in\mathcal{Y}^-}\|g_j\|).$$

Then, as $\delta\min_{j\in\mathcal{Y}^-}\|g_j\| > \max_{j\in\mathcal{Y}^+}\|g_j\|$,

$$\frac{\mathrm{d}}{\mathrm{d}t}\log\pi(y_i|x) \le \eta(\sum_{j\in\mathcal{Y}^+} A_j)\|g_i\|(\max_{j\in\mathcal{Y}^+}\|g_j\| - \delta\min_{j\in\mathcal{Y}^-}\|g_j\|) < 0,$$

completing the proof. □

Lemma A.4 (Restatement of Lemma 3.3)**.** *Fix a prompt x, and let $\pi_t(y) := \pi_{\theta_t}(y \mid x)$ be the distribution at the training step t. If $\mathrm{Cov}(\log\pi_t(y), \dot{\ell}_t(y)) < 0$, then $\frac{\mathrm{d}}{\mathrm{d}t}H_t > 0$.*

Proof. Differentiate the entropy:

$$\frac{d}{dt}H_t = -\sum_y \frac{d\pi_t(y)}{dt}\big(1+\log\pi_t(y)\big). \tag{4}$$

Since $\sum_y \pi_t(y) = 1$, we have $\sum_y \frac{d\pi_t(y)}{dt} = 0$. Define $\dot{\ell}_t(y) = \frac{\mathrm{d}}{\mathrm{d}t}\log\pi(y|x)$, we have

$$\frac{d}{dt}H_t = -\sum_y \frac{d\pi_t(y)}{dt}\log\pi_t(y) = -\sum_y \pi_t(y)\dot{\ell}_t(y)\log\pi_t(y) = -\mathrm{Cov}(\log\pi_t(y), \dot{\ell}_t(y)). \tag{5}$$

Then, as we have assumed $\mathrm{Cov}(\log\pi_t(y), \dot{\ell}_t(y)) < 0$, $\frac{\mathrm{d}}{\mathrm{d}t}H_t > 0$. □

B Further Analyses

B.1 Further Evidence Supporting the Training Dynamic of Entropy Eruption in Agent RL

Beyond the examples we show in Figure 1, we exhibit more experimental instances with Llama3.2-1B and Qwen2.5-1.5B on AlfWorld, WebShop and Search tasks. For AlfWorld and WebShop, we adopt mid-training on Llama3.2-1B similarly to our main experiments with corpora from Openmanus-RL [25]. As shown in Figure 12, Llama3.2-1B and Qwen2.5-1.5B consistently exhibit the same entropy-eruption pattern across all three agent tasks, indicating that this phenomenon is not specific to a single model or environment. In AlfWorld, the entropy remains relatively controlled in the early stage and then rises noticeably in the middle-to-late stage, while the reward has already reached a plateau. In WebShop, the effect is even more pronounced: entropy undergoes a sharp late-stage escalation, accompanied by reward stagnation or even collapse, especially for Llama3.2-1B where the reward drops abruptly after the entropy spike. In Search, models also show a clear entropy burst whenever reward enters a plateau. Overall, these results provide further evidence that entropy eruption is a robust training dynamic in agent RL and is often associated with reward instability or degradation once training enters the later phase.

B.2 Mid-training shrinks the length of phase 1.

Figure 12 shows that models initiated with mid-training have a substantially shorter, or even nearly nonexistent, Phase 1. This is expected because mid-training already instills the basic formatting and validity constraints required for function calling. Meanwhile, it also improves the policy's overall trajectory quality. As a result, the policy enters RL from a substantially better initialization, with a higher valid-action ratio and a stronger average reward. Consequently, early RL no longer needs to spend many updates correcting format errors, and much of the coarse entropy-reduction process that typically characterizes Phase 1 has already occurred before RL begins.

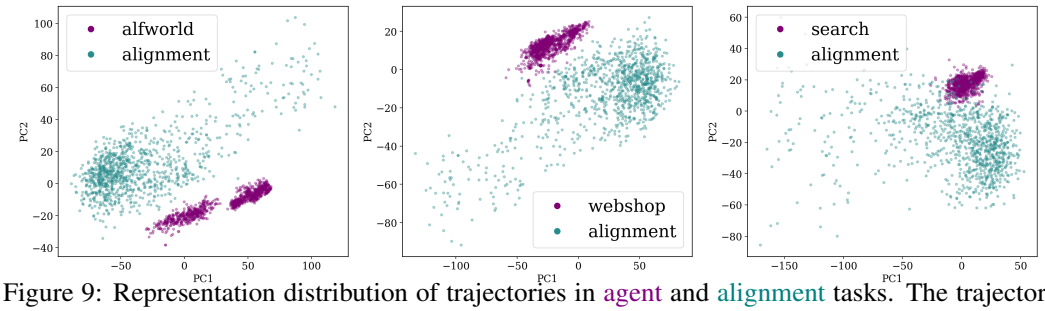

Figure 9: Representation distribution of trajectories in agent and alignment tasks. The trajectory representation in the agent task is significantly more concentrated.

B.3 Representations of Agent Trajectories are Significantly More Concentrated than Those of Alignment Trajectories.

From Phase 1, the policy model would quickly learn the basic rules of output format and function calling, thus making the sampling focus on valid trajectories. Here, we show that the representation of valid agent trajectories shares a very similar representation in the LLM semantic space. Figure 3 visualizes the last-layer hidden representations of Qwen2.5-7B-Instruct, averaged over response tokens, for trajectories sampled from agent tasks (ALFWorld, WebShop, Search) and from a non-agent alignment task (UltraFeedback [49]). Agent trajectories form significantly tighter clusters than alignment trajectories. Next, we show that the high representation similarity among agent trajectories leads to entropy eruption.

B.4 Further Evidence Supporting the Likelihood Decrease of Valid Trajectories

To complement our main results, we further track the training-time log-probability of *correct*, *wrong*, and *valid* responses for Llama3.2-1B and Qwen2.5-1.5B on AlfWorld, WebShop, and Search. As shown in Figure 10, the likelihood assigned to valid trajectories surprisingly does not increase along with training, instead, across all three agent tasks and both model families, it either gradually declines after an early improvement phase or undergoes an abrupt late-stage collapse. In AlfWorld, both models show a gentle fluctuation in the log-probability of valid responses, with a clear decreasing overall tendency. In Search, the effect is even more pronounced: after a relatively stable phase, the log-probability of valid trajectories drops sharply in the later stage and only partially recovers. In WebShop, the phenomenon is strongest, especially for Llama3.2-1B, where the likelihood of valid responses collapses dramatically during late training.

B.5 Further Evidence on Occurrence of Degenerate Patterns during Entropy Eruption.

As the entropy erupts, the distribution is flattened, which would make the degenerate patterns more likely to be sampled in the training. We found that the entropy eruption is strongly synchronized with the emergence of some degenerate patterns such as sentence duplication. To empirically show this, we collect the 1024 sampled trajectories at different checkpoints during training, and analyze their sentence duplication ratio. Specifically, we first split the responses into sentence chunks, and measure the fraction of sentences that appear multiple times in the same response. As shown by the purple lines in Figure 11, the sentence duplication ratio surges during the entropy eruption, and has the same fluctuation patterns as the entropy dynamics. Meanwhile, as the orange line shown in Figure 11, other kinds of error are also likely to appear during the entropy eruption, such as the format error.

B.6 Further Evidence on Lasting Effect of Entropy Eruption.

As shown in Figure 11, although training temporarily becomes normal after entropy eruption subsides, the damage introduced during the eruption can remain. When entropy erupts, the policy distribution flattens and valid trajectories receive lower probability, so sampled trajectories are more likely to contain degenerate patterns even among responses that still obtain the correct answer. If these degenerate trajectories receive positive reward, the policy can inadvertently learn and reinforce such garbage patterns. Consequently, the harmful effects of entropy eruption may persist after the eruption subsides and can even be amplified during entropy subsidence, as shown by the format-error curves

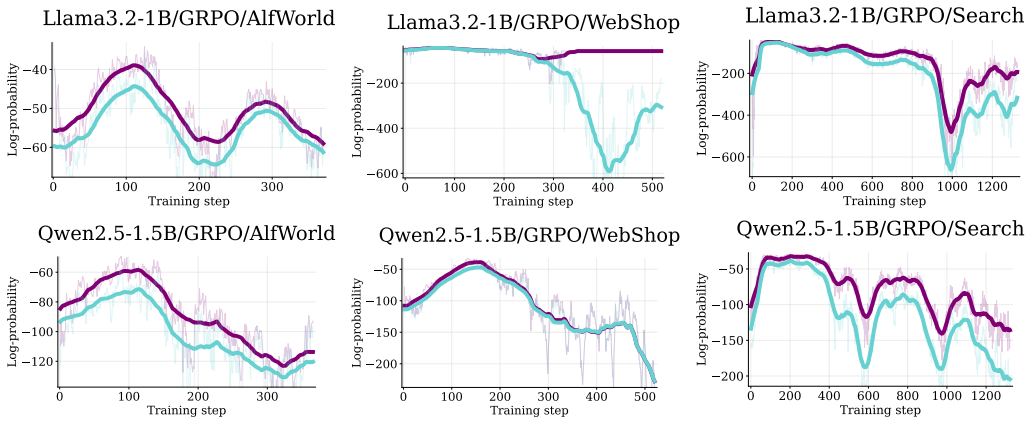

Figure 10: Further evidence supporting the likelihood decrease of valid trajectories during agent RL

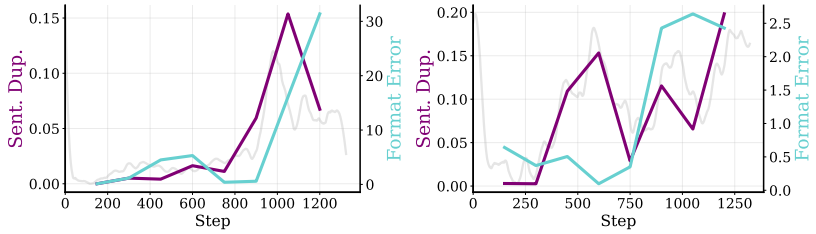

Figure 11: Sentence Duplication Ratio rises with entropy eruption, which also increases the occurrence of format error. The gray line in the background is the entropy curve.

(light blue) in Figure 11. We provide additional analysis and statistics in Appendix B.6. In more severe cases, entropy eruption can lead to semantic collapse and completely break RL training. As shown in Figure 12, for Llama3.2-1B on WebShop, entropy eruption is followed by a collapse of the reward curve to zero.

C Further Validation For SEAL

C.1 Implementation Details

The experiments of Qwen2.5-1.5B and Llama3.2-1B are conducted on 2 NVIDIA A100-SXM4-80GB GPUs, and the experiments of Qwen2.5-7B and Llama3.2-3B are conducted on 2 NVIDIA H200 GPUs. We mostly follow experimental settings in [16]. We fix the random seed of the environment to 0 for reproducibility.

For the AlfWorld task, the maximum prompt length is 2048 tokens, and the maximum response length is 512 tokens. Each episode allows up to 25 environment steps. The learning rate is set to 2e-6 for the actor. The reward is set to 10 for success and to 0 for failure. We also apply a reward penalty of -0.1 to punish the format errors. We use a group size of 8. The rollout temperature is set to 1.0, while the validation temperature is set to 0.4. The mini-batch size is 256, and the KL-divergence loss coefficient is set to 0.01. For GIGPO, we also keep the best hyperparameter according to the original paper [16], where the weighting coefficient is fixed at 1 without further tuning, and the discount factor is set to 0.95.

For the WebShop task, the maximum prompt length is 4096 tokens, and the maximum response length is 512 tokens. Each episode is limited to 10 environment steps. The learning rate is 2e-6 for the actor. As in ALFWorld, reward is set to 10 for success and 0 for failure. Invalid actions are penalized with a reward of -0.1. Group size is set to 8. The rollout temperature is set to 1.0, while the validation temperature is set to 0.4. The mini-batch size is 64, and the KL-divergence loss coefficient is set to 0.01. For GIGPO, we also keep the best hyperparameter according to the original paper [16],

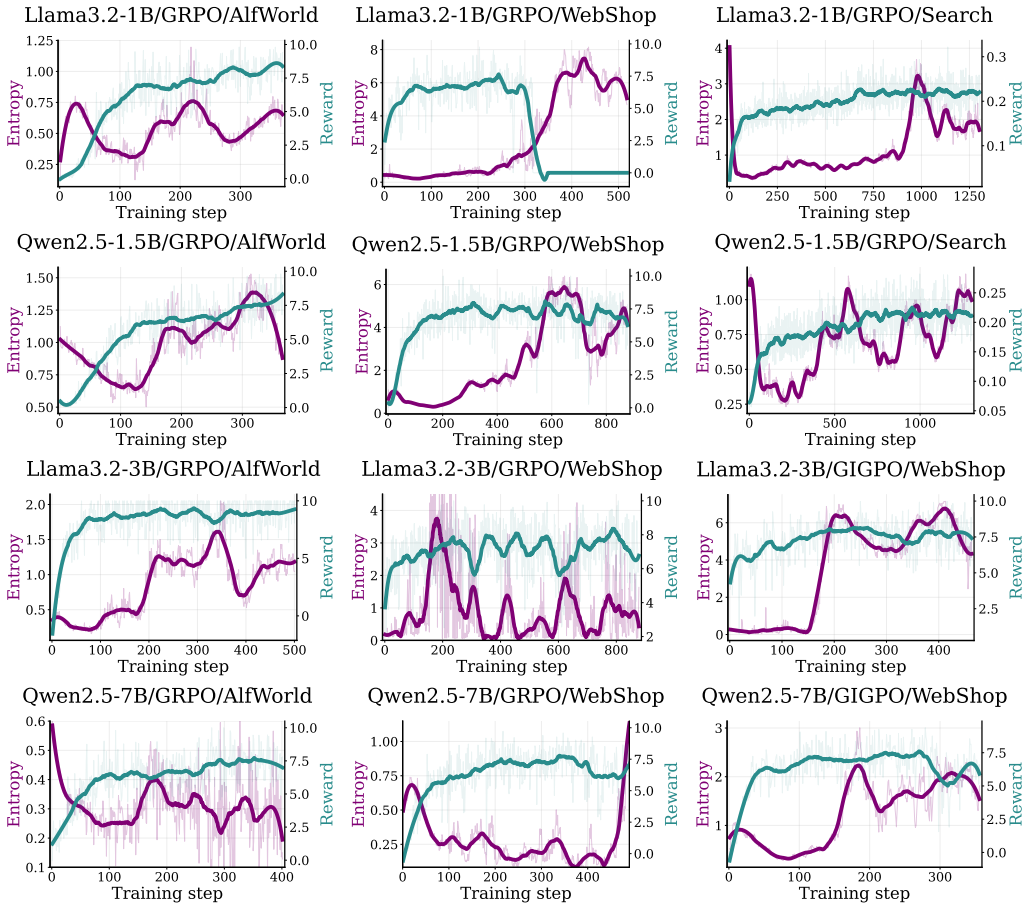

Figure 12: Further evidence supporting the training dynamic of entropy eruption in agent RL. In the AlfWord and WebShop tasks, the length of Phase 1 in Llama-series models is shortened due to mid-training as demonstrated in Section 3.

where the weighting coefficient is fixed at 1 without further tuning, and the discount factor is set to 0.95.

For the search-augmented QA task, the maximum prompt length is 4096 tokens, and the maximum response length is 512 tokens. The max turn is set to 3. The learning rate is 2e-6 for the actor. Reward is set to 1 for success and 0 for failure. Invalid actions are penalized with a reward of -0.01. We set the train data size to 256 and use a group size of 5 following GIGPO [16] and Search-R1 [14]. Rollout and validation temperatures are set to 1.0 and 0.0, respectively. The mini-batch size is 512, and the KL-divergence loss coefficient is set to 0.001. The weighting coefficient is fixed at 1 without further tuning, and the discount factor is set to 0.95.

To evaluate the degenerate score with GPT-4, we use the prompt in Box C.1.

C.2 SEAL Successfully Alleviates Entropy Eruption

As shown in Figure 13, SEAL consistently mitigates the entropy-eruption phenomenon across different tasks, backbones, and RL algorithms. Compared with standard GRPO/GIGPO, the corresponding runs with SEAL exhibit substantially smoother entropy trajectories, with smaller spike magnitudes and weaker recurrent oscillations. As shown in Figure 8, suppressing entropy eruption not only makes training more robust and improves performance, but also reduces degenerate patterns.

Box C.1: GPT-4 prompt for degeneration judgment

<System Prompt>
You are an expert evaluator of language-model generations.

Your task is to judge how much a model output contains:
1. semantic garbage / nonsensical content
2. repetition / looping / duplicated phrases
3. off-task filler that does not help solve the task

You are NOT judging whether the output is polite, safe, or toxic in the moderation sense.
Here, "toxicity" means degeneration: nonsense, repetition, meaningless filler, or severe loss of grounding.

Return a score from 0 to 10, where a HIGHER score means WORSE generation quality:
- 0 = clean, coherent, non-repetitive
- 2.5 = mild repetition or mild filler
- 5 = noticeable nonsense/repetition affecting usefulness
- 7.5 = severe degeneration; much of the text is repetitive or semantically broken
- 10 = almost entirely garbage, repetitive looping, or incoherent

Important rules:
- Penalize repeated phrases, repeated constraints, repeated boilerplate, repeated clauses, and low-information loops.
- Penalize semantic garbage: irrelevant claims, contradictory statements, hallucinated filler, strange topic drift, or sentences that are grammatical but meaningless in context.
- Penalize generic filler when it replaces task-relevant reasoning.
- Do NOT penalize short necessary restatements once.
- Do NOT penalize correct task-specific repetition if it is needed and minimal.
- If the output includes actions, also penalize actions that are inconsistent with the stated action space or observation.

Scoring principle:
Estimate the proportion and severity of degenerate content in the output.
Higher score = larger portion of nonsense/repetition and/or stronger severity.

Return ONLY a number from 0 to 10, with no additional text or explanation.

<User Prompt>
PROBLEM:
problem
SOLUTION:
response

C.3 Hyperparameter Sensitivity

We study the sensitivity of SEAL to the auxiliary-loss weight α in Figure 14. Across both Llama3.2-3B and Qwen2.5-7B on WebShop, the method remains relatively stable over a broad range of α, indicating that SEAL does not require highly delicate tuning to be effective. In terms of task performance, both GRPO+SEAL and GIGPO+SEAL achieve their best or near-best accuracy at intermediate values of α, while still maintaining competitive results under smaller or larger settings. At the same time, the LLM-as-a-Judge score on degenerate patterns also remains favorable across most choices of α, with lower scores typically obtained in the same intermediate range. Overall, these results suggest a consistent trade-off: moderate auxiliary-loss strength is usually sufficient to improve representation separation without over-regularizing the policy. Importantly, across a wide

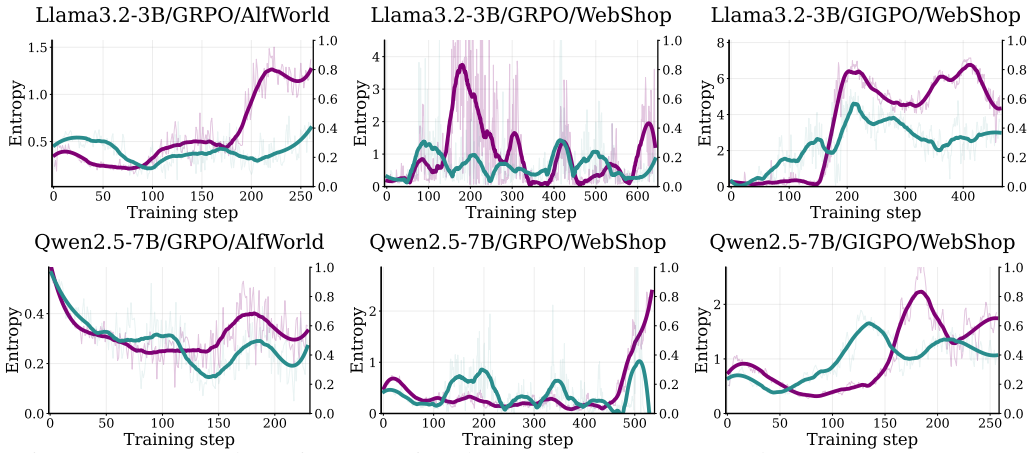

Figure 13: Entropy dynamics comparison between GRPO/GIGPO and GRPO/GIGPO+SEAL.

Table 2: Comparison of GRPO performance with and without SEAL on Search-augmented QA for Qwen2.5-7B and Llama3.2-3B.

Model	Methods	In-domain		Out-of-domain					**Avg.**
		NQ	HotpotQA	TriviaQA	PopQA	2Wiki	MuSiQue	Bamboo.	
Llama3.2 -3B	GRPO	48.22	37.98	64.60	45.19	35.21	11.62	67.74	44.37
	GRPO + SEAL	47.51	44.21	63.95	47.31	44.56	17.99	72.98	48.36
Qwen2.5 -7B	GRPO	46.08	40.12	63.78	45.55	38.77	16.32	65.72	45.19
	GRPO + SEAL	46.08	40.46	65.20	46.53	41.58	17.72	69.76	46.76

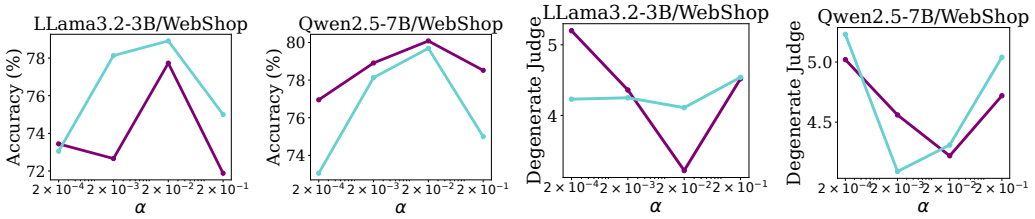

Figure 14: Accuracy and LLM-as-a-Judge score on degenerate patterns of GRPO+SEAL and GIGPO+SEAL with different hyperparameters.

range of α, the accuracy and degeneracy metrics of SEAL remain generally better than those of the corresponding baselines without SEAL (see Table 1 and Figure 8), showing that the benefit of SEAL is robust rather than tied to a single hyperparameter choice.

C.4 Statistics over three runs

Table 3 reports the mean performance over three random seeds. The results show that the gains from SEAL remain consistent after averaging across runs. On Llama3.2-3B, adding SEAL improves the AlfWorld average from 91.29 to 94.23, while also increasing the WebShop score from 87.06 to 89.81 and the success rate from 67.11 to 76.95. On Qwen2.5-7B, SEAL raises the AlfWorld average from 82.20 to 83.40, and also improves WebShop from 89.63 to 90.26 in score and from 76.95 to 77.73 in success rate. Overall, the multi-run statistics confirm that the benefits of SEAL are stable and reproducible, rather than arising from a favorable single run.

C.5 Performance on search-augmented QA.

Table 2 reports the performance of GRPO with and without SEAL on the search-augmented QA benchmark. Across both model backbones, SEAL consistently improves the average performance. These results suggest that the benefits of SEAL are not limited to interactive agent environments such as AlfWorld and WebShop, but also extend to search-augmented QA, where multi-step retrieval

Table 3: Statistics of GRPO performance with and without SEAL on AlfWorld and WebShop for Qwen2.5- and Llama3.2-series models. Results are the average of three runs.

Model	Methods	ALFWORLD							WEBSHOP	
		Pick	Look	Clean	Heat	Cool	Pick2	**Avg.**	Score	**Succ.**
Llama3.2 3B	GRPO	97.17	99.95	96.44	77.21	90.29	86.68	91.29 ± 2.35	87.06	67.11±4.11
	GRPO + SEAL	95.73	93.50	93.97	85.17	98.50	98.51	94.23± 2.57	89.81	76.95±3.12
Qwen2.5 7B	GRPO	99.95	78.89	92.77	88.14	82.42	51.03	82.20± 2.57	89.63	76.95 ± 1.81
	GRPO + SEAL	94.21	85.32	99.61	86.28	85.35	49.63	83.40± 2.76	90.26	77.73± 1.17

and reasoning are both required. Overall, the results provide further evidence that reducing harmful gradient interference can improve policy optimization across a broader range of agent-style tasks.

D Limitations and Future Work

Our study is conducted on three representative agent tasks—AlfWorld, WebShop, and search-augmented QA—and on two widely used model families, the Llama and Qwen series. While these settings already cover multiple environments, reward structures, and backbone scales, they do not exhaust the full space of agent RL problems. It would therefore be valuable to examine whether the same entropy dynamics arise as broadly in other domains, such as longer-horizon agents, richer tool-use settings, or multimodal environments. At the same time, the consistency of our observations across all of our current settings suggests that cyclical entropy eruption is unlikely to be an isolated artifact of a single benchmark or model family. In addition, our proposed mitigation, SEAL, is designed as a practical intervention during RL training. This choice matches the focus of this work, namely, to diagnose the training dynamics of entropy eruption and to show that explicitly reducing harmful representation overlap can alleviate it. However, it remains an open question whether similar benefits could be obtained even earlier in the training pipeline, for example, through pre-training, instruction tuning, or mid-training strategies that better separate correct and incorrect trajectories before RL begins. Exploring such earlier-stage interventions is a promising direction for future work. More broadly, future work could investigate whether improved curricula, trajectory filtering, reward design, or representation regularization can further stabilize agent RL while preserving useful exploration. We hope this work provides both an empirical starting point and a theoretical lens for studying these questions.

E Broader Impact

This work studies the training dynamics of reinforcement learning for LLM agents, with the goal of improving the stability and reliability of agent training. As discussed in the paper, agent systems are increasingly being deployed in settings such as web interaction, software engineering, scientific assistance, and other long-horizon decision-making tasks, making it important to understand not only how to improve benchmark performance, but also how and why training can become unstable. Our analysis of cyclical entropy eruption and our proposed mitigation, SEAL, are intended to contribute to safer and more controllable agent optimization by reducing pathological behaviors such as likelihood collapse on valid trajectories, degenerate generation patterns, and severe reward instability.

At the same time, more capable and stable agent RL can have dual-use implications. Techniques that improve the robustness of agent training may also make it easier to build agents that are more persistent, autonomous, and effective in real-world environments. Such systems could be beneficial in domains like scientific discovery, productivity assistance, and automation.

We therefore view this work as part of a larger effort toward principled and responsible agent development. A better understanding of training dynamics can support improved monitoring, evaluation, and intervention during RL, and future work should continue to study how stability-oriented methods interact with broader concerns such as alignment, robustness, and misuse prevention.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We explicitly mentioned our core contributions: We identify a unique entropy dynamics in agent RL and analyze its cause. At last, we propose an easy-to-implement approach based on our understanding of this dynamics.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed limitations in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We clearly exhibit the formulation of question in each lemma. For specific conditions that should be satisfied, we discussed why it is satisfied for real experiments in context.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provided implementation details in Appendix C.1. Meanwhile, we provide data and code.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We released the data and code, and provide command scripts to directly run it.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: As shown in Appendix C.1 and command scripts in code, we include all necessary implementation details and data sources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We include statistics in Appendix C.4. Meanwhile, we fix the random seed for reproducibility. Moreover, we conducted hyperparameter searches and experiments across multiple tasks/backbones, which can validate the consistent effectiveness of our approach.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: we include information in Appendix C.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We followed the terms in NeurIPS code of ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discussed potential impacts of our paper in introduction, approach, and limitation part.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release any new data source or pretrained language models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited every data source, pretrained models and code repository we based on. The license and terms of use are properly respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release any new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not conduct crowdsourcing experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.