Extended Abstract

Motivation Reinforcement learning fine-tuning (RLFT) is widely used to adapt pretrained policies to specific downstream tasks using simple reward signals. However, a known failure mode in RLFT is *entropy collapse*, where the policy becomes increasingly deterministic and forgets valuable behaviors learned during pretraining. To counteract this, many algorithms introduce an entropy bonus, intended to promote continued exploration. Yet the precise role of this bonus in maintaining policy diversity and supporting effective learning remains poorly understood. Our work critically examines whether the entropy bonus truly encourages exploration, or merely injects randomness without improving performance.

Method We conduct a systematic investigation of entropy regularization by fine-tuning behaviorally cloned policies across four Atari environments—*Gravitar*, *Breakout*, *Berzerk*, and *Private Eye*. We vary the entropy coefficient ($c_2 \in \{0.0, 0.05, 0.5, 0.9\}$) while holding other hyperparameters fixed, allowing us to isolate the effects of entropy bonuses during fine-tuning. Our analysis combines empirical evaluation (rewards, success rates, entropy trends, KL divergence, critic loss) with theoretical analysis showing how entropy bonuses introduce competing gradients that trade off between maximizing reward and distributing probability mass uniformly.

Implementation We begin by training a PPO agent from scratch to generate expert rollouts. These are used to train a behaviorally cloned policy, which serves as the initialization for all RLFT experiments. The cloned policy is fine-tuned for 1M PPO steps under each entropy setting. For each configuration, we log metrics such as entropy over time, critic loss, and changes in action distributions. Our implementation follows standard PPO practices and uses a convolutional architecture consistent with prior work on Atari.

Results Our experiments reveal that entropy bonuses significantly impact policy fine-tuning dynamics. Without entropy regularization, policies rapidly collapse to deterministic behaviors, sharply reducing entropy and failing to explore. Moderate bonuses delay this collapse and improve critic convergence, evidenced by lower critic loss across environments. However, high entropy coefficients often lead to premature unlearning of expert behaviors and sustained underperformance—especially in complex games like *Berzerk* and *Private Eye*. Crucially, increased entropy does not induce qualitatively new behaviors; it merely broadens existing action modes rather than discovering new ones. Action distribution analyses confirm that entropy widens policy variance without shifting modal preferences. KL divergence trends also show that higher entropy accelerates divergence from the pretrained policy, but not toward more successful behavior. Finally, critic learning benefits most from moderate entropy: losses stabilize and better approximate state values, particularly in exploratory regions of the state space.

Discussion These findings complicate the standard view that entropy bonuses robustly enhance exploration. Instead, entropy often causes destructive unlearning early in RLFT, especially when coefficients are high. While entropy improves critic learning by encouraging broader data collection, its exploratory value saturates quickly. All policies, regardless of coefficient, converge to a similar entropy ceiling—suggesting diminishing returns from increasing c_2. Moreover, entropy mostly adds variance around existing modes rather than prompting discovery of new ones. This trade-off—between exploration and preservation of pretrained behavior—proves difficult to manage without adaptive tuning. Our analysis also reveals that RLFT reinforces pretrained action biases: even when better alternatives exist, fine-tuned policies may favor suboptimal expert actions due to prior bias.

Conclusion Entropy regularization stabilizes value learning and delays policy collapse, but it is insufficient for inducing meaningful exploration or preserving pretraining benefits. Larger coefficients often undermine performance by accelerating unlearning and flattening useful action distributions. Despite improved critic loss and temporarily higher entropy, agents fail to discover novel strategies. Overall, entropy acts more as a noise injection mechanism than a true driver of exploration. Effective RLFT in complex environments likely requires more targeted techniques—such as adaptive entropy decay or explicit diversity rewards—that go beyond static entropy bonuses.

A Critical Study of the Entropy Bonus for Exploration

Ifdita Hasan Orney
Department of Computer Science
Stanford University
ifdi1101@stanford.edu

Iddah Mlauzi
Department of Computer Science
Stanford University
iddah@stanford.edu

George Kojo Frimpong Birikorang
Department of Computer Science
Stanford University
george25@stanford.edu

Advisor: Jubayer Ibn Hamid
Department of Computer Science
Stanford University
jubayer@stanford.edu

Abstract

Reinforcement learning fine-tuning (RLFT) often leads to entropy collapse, where the policy prematurely narrows its behavior and loses generalization. This project investigates how varying entropy regularization affects exploration and critic stability during RLFT in PPO-based agents. We pretrain agents via behavior cloning and fine-tune them with different entropy coefficients across four Atari games of varying exploration difficulty. Our results show that moderate entropy improves critic loss convergence and maintains policy diversity, while high entropy degrades performance and erodes useful pretraining signals. Action diversity gains saturate quickly, and fine-tuned policies tend to reweight pretrained behaviors rather than discover new ones. We conclude that entropy helps delay collapse but does not solve the broader exploration challenge in RLFT.

1 Introduction

Online reinforcement learning (RL)(Sutton and Barto, 2018) has demonstrated significant promise in enabling agents to acquire complex behaviors through sequential decision-making and continuous interaction with their environments. RL methods have achieved remarkable success in diverse domains, including game playing(Silver et al., 2017), robotics (Luo et al., 2025), and natural language processing (DeepSeek-AI et al., 2025; Yang et al., 2025; Lambert et al., 2025; OpenAI et al., 2024). More recently, large-scale deep RL has advanced the frontier of large language models (LLMs), particularly in verifiable domains such as mathematical reasoning and programming, enabling LLMs to tackle complex logical tasks.

In these settings, RL typically begins with a pretrained model that is optionally fine-tuned on high-quality data—for example, long chains of thought (CoT) for reasoning—before being further optimized using reinforcement learning on simple, automatically computable rewards. These rewards are often based on whether the model's output matches a ground-truth solution in mathematics or passes unit tests in code, facilitating scalable optimization without human labeling. This framework has garnered significant attention due to its simplicity and practical effectiveness.

However, it remains unclear whether reinforcement learning fine-tuning (RLFT) enables models to discover novel behaviors beyond those acquired during pre-training or supervised fine-tuning. Recent work suggests that RLFT may primarily sharpen the policy distribution around already successful behaviors present in the pretrained model (Yue et al., 2025; Cui et al., 2025). This phenomenon, often termed "entropy collapse," may cause models to abandon alternative beneficial behaviors and focus narrowly on the most successful one. As a result, policies become less stochastic and less exploratory, potentially limiting progress when encountering unfamiliar or challenging problems. The ability to explore and generate diverse solution strategies is especially crucial in these settings: a model that collapses onto a single behavior may fail when that behavior is invalid at test time or when users require different approaches. Retaining a repertoire of learned behaviors is therefore essential for robustness and adaptability.

A popular approach to mitigating entropy collapse is the addition of an entropy bonus (Cui et al., 2025; Schulman et al., 2017b), which augments the RL objective to simultaneously maximize expected rewards and the expected entropy of the policy. In principle, maximum entropy RL encourages exploration by promoting more stochastic policies. However, in practice, large-scale RLFT often omits the entropy bonus due to various practical considerations.

In this paper, we investigate the mechanisms and effects of the entropy bonus in RLFT. Specifically, we address the following questions:

1. Does the entropy bonus enable the model to retain successful behavioral modes that would otherwise be forgotten during RLFT? In particular, does it genuinely promote exploration of alternative action modes, or does it simply increase variance around the modes already favored by RLFT?
2. How does the entropy bonus affect critic learning and, by extension, the quality of the extracted policy?

Our findings indicate that incorporating an entropy bonus in PPO does not lead to the discovery of fundamentally new behaviors, and can instead accelerate the unlearning of expert behaviors acquired during pre-training. On the other hand, we observe that the entropy bonus can stabilize critic learning and lead to more accurate value estimates.

2 Related Work

Policy Gradient Methods. Reinforcement learning algorithms based on policy gradients directly optimize the policy parameters by following an estimate of the gradient of expected return. Early methods like REINFORCE (Sutton and Barto, 2018) suffered from high variance, which spurred development of variance-reduction techniques (e.g., baseline subtraction (Williams, 1992); advantage estimation (Schulman et al., 2018)). Actor-critic architectures combine policy gradient actors with value function critics to stabilize learning (**?**). Notably, the Asynchronous Advantage Actor-Critic (A3C) algorithm (Mnih et al., 2016) demonstrated that parallel actor learners can effectively train deep policies across many Atari games. Trust-region methods introduced theoretical guarantees for stable policy updates: Trust Region Policy Optimization (TRPO) (Schulman et al., 2017a) enforced a small KL-divergence between old and new policies to ensure monotonic improvement. Proximal Policy Optimization (PPO) (Schulman et al., 2017b) later simplified TRPO by using a clipped surrogate objective, becoming a popular on-policy method due to its ease of implementation and strong empirical performance. PPO has been widely used as a baseline for fine-tuning large pre-trained policies, thanks to its robustness against unstable gradient updates. On the other hand, off-policy policy gradient algorithms have also been explored. Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2019) extended actor-critic methods to continuous action spaces by combining a deterministic policy with an off-policy Q-learning update. A significant advance in this category is Soft Actor-Critic (SAC) (Haarnoja et al., 2018), which maximizes a maximum entropy objective: the agent aims to maximize expected reward while also maximizing the entropy of its policy. By explicitly including an entropy bonus in the objective, SAC encourages continued exploration and trains stochastic policies.

Entropy Regularization and Collapse. It is common in deep RL to add an entropy bonus to the reward or objective to encourage exploration (Williams, 1992; Mnih et al., 2016; Schulman et al., 2017b). Entropy regularization has become standard in policy gradient methods to prevent the policy's probability distribution from collapsing to a single action. Without sufficient entropy encouragement, entropy collapse can occur (Cui et al., 2025; West and Potts, 2025), wherein the policy becomes nearly deterministic early in training and gets stuck exploiting suboptimal actions. This issue is especially pronounced during fine-tuning of pre-trained models: a strong pre-trained policy may quickly converge with minimal exploration if the entropy coefficient is too low. However, tuning the entropy coefficient is non-trivial – too high of an entropy bonus can hinder convergence, while too low fails to prevent collapse. Techniques like automatic entropy tuning in SAC address this by adjusting the coefficient on-line to maintain a desired entropy level (Haarnoja et al., 2018).

Exploration Strategies and Intrinsic Motivation. Encouraging exploration in RL goes beyond entropy bonuses. A rich line of research has developed intrinsic motivation techniques, where an agent receives internal rewards for novel or informative experiences. One approach is curiosity-driven exploration: Pathak et al. (Pathak et al., 2017) introduced an Intrinsic Curiosity Module (ICM) that rewards an agent for dynamics prediction error – essentially incentivizing the agent to seek states that are harder to predict. This method enabled agents to efficiently explore sparse-reward environments, even in the absence of extrinsic rewards. Similarly, Random Network Distillation (RND) (Burda et al., 2018) provides an intrinsic reward by measuring an agent's prediction error on a fixed random function; the agent thus continually seeks states that produce high prediction error, which correlates with novel states. Such curiosity-based methods have yielded substantial gains on hard-exploration games (e.g., Montezuma's Revenge), where naive exploration fails (Burda et al., 2018). Another paradigm is count-based exploration adapted to high-dimensional spaces: pseudo-count methods use density models to reward rarely visited states (Bellemare et al., 2016), achieving progress on games with very sparse rewards. In summary, a variety of exploration-enhancement techniques have been developed, and they can complement entropy regularization: while entropy encourages random action selection to a degree, intrinsic rewards and other strategies bias the exploration toward novel or informative states.

Atari and RL Benchmark Results. The Arcade Learning Environment (ALE) (Bellemare et al., 2013) – a suite of dozens of Atari 2600 video games – has long served as a standard benchmark for deep RL algorithms. Value-based methods initiated deep RL's success on Atari: the Deep Q-Network (DQN) (Mnih et al., 2016), combining convolutional neural networks with Q-learning, famously reached human-level performance on many games. Subsequent improvements like Double DQN, dueling networks, and prioritized replay were combined in the Rainbow agent (Hessel et al., 2017), further advancing the state of the art in value-based learning on Atari. In parallel, policy gradient and actor-critic methods have also been validated on Atari. A3C (Mnih et al., 2016) achieved competitive results with a much simpler, synchronous training setup. PPO (Schulman et al., 2017b) has likewise been extensively applied to Atari; its on-policy nature typically yields slightly lower sample efficiency than off-policy DQN variants, but PPO's stability makes it a strong choice for fine-tuning large neural policies on Atari benchmarks. Indeed, many recent studies use PPO as the backbone for Atari experiments, sometimes in conjunction with auxiliary losses or pre-training, because it reliably learns a good policy without divergence issues. The continued challenge in Atari has been hard-exploration games (like Montezuma's Revenge, Pitfall, Gravitar), where even advanced agents would score little to no reward due to sparse feedback. Intrinsic motivation approaches (ICM, RND, etc.) were introduced to tackle these, and they led to significant, though not complete, improvements (Burda et al., 2018).

3 Method

In PPO (Schulman et al., 2017b), we collect data using $\pi_{\theta_{\text{old}}}$ and consider finding π_θ that maximizes $V_{\pi_\theta} - V_{\pi_{\theta_{\text{old}}}}$. Using the performance difference lemma (Schulman et al., 2017a; Hamid, 2025), this can be formulated as the following optimization problem:

$$\max_\theta \mathbb{E}_{s_t \sim d^{\pi_{\theta_{\text{old}}}}(\cdot),\, a_t \sim \pi_{\theta_{\text{old}}}(\cdot|s_t)} \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t \right] \tag{1}$$

$$\text{subject to} \quad \mathbb{E}_{s_t \sim d^{\pi_{\theta_{\text{old}}}}(\cdot)} \left[\mathrm{KL}\big(\pi_{\theta_{\text{old}}}(\cdot \mid s_t) \mid\mid \pi_\theta(\cdot \mid s_t)\big) \right] \leq \delta. \tag{2}$$

PPO solves this problem by optimizing the following clipped surrogate opjective function:

$$L_t^{\text{CLIP}+VF}(\theta) = \hat{\mathbb{E}}_t \left[L_t^{\text{CLIP}}(\theta) - c_1 L_t^{VF}(\theta) \right], \tag{3}$$

where c_1 is a coefficient, and L_t^{VF} is a squared-error loss $(V_\theta(s_t) - V_t^{\text{targ}})^2$.

However, this objective can cause the learned policy to not explore and rapidly collapse on actions that yield high rewards. Since the data is collected in an on-policy manner and since the policies are trained to optimize the rewards, this objective can result in overly narrow action distributions, causing the policy to be stuck in local optima. As such, (Schulman et al., 2017b) proposed adding an entropy bonus:

$$L_t^{\text{CLIP}+VF+S}(\theta) = \hat{\mathbb{E}}_t \left[L_t^{\text{CLIP}}(\theta) - c_1 L_t^{VF}(\theta) + c_2 \mathcal{S}[\pi_\theta](s_t) \right], \tag{4}$$

where c_2 is a coefficient and $\mathcal{S}[\pi_\theta](s_t) = \mathcal{H}(\pi_\theta(\cdot \mid s_t))$ is the entropy of the distribution at state s_t. Intuitively, entropy measures the "width" of a policy's action distribution; the maximum entropy distribution over any sample space is the uniform distribution, whereas the minimum entropy distribution is one that adds all probability mass to only one sample. As such, maximizing this objective via the entropy bonus $\mathcal{S}$ in the objective function, at least in principle, encourages exploration.

In this paper, we examine the effects of adding this entropy bonus to the PPO objective when fine-tuning a pretrained policy. To do so, in our experiments, we use a range of entropy coefficients and analyze rewards, success rates, critic losses, the effects on entropy of the trained policy and KL divergence of the trained policy from the pretrained policy.

3.1 Theoretical Analysis

In this section, we theoretically analyze how the reinforcement learning fine-tuned policy diverges from the pre-trained policy. In particular, we will compare this divergence pattern between a learning algorithm that uses an entropy bonus and one that does not. Throughout this section, we consider a softmax policy which can be expressed as

$$\pi_\theta(a \mid s) = \frac{\exp(z_{sa})}{\sum_{a'} \exp(z_{sa'})} \tag{5}$$

where $s \sim d^{\pi_\theta}(\cdot)$ is sampled from the stationary state distribution induced by the policy π_θ, $a \sim \pi_\theta(\cdot \mid s)$, and z_{sa} is the output logit of action a given input s.

Proposition 1. *Let $\psi(\pi_\theta^k \mid s) = D_{KL}(\pi_\theta^k(\cdot \mid s) \parallel \pi_{ref}(\cdot \mid s))$. Then, for vanilla policy gradient methods without any entropy bonus,*

$$\psi(\pi_\theta^{k+1} \mid s) - \psi(\pi_\theta^k \mid s) = \operatorname{Cov}_{a \sim \pi_\theta^k(\cdot \mid s)} \left(\log \frac{\pi_\theta^k(a \mid s)}{\pi_{ref}(a \mid s)} + 1, \pi_\theta^k(a \mid s) A^{\pi_\theta^k}(s, a) \right).$$

Proof Sketch: Since we are using the softmax policy, we can use a Taylor expansion of $\psi(\pi_\theta^{k+1} \mid s) - \psi(\pi_\theta^k \mid s)$ centered around $\psi(\pi_\theta^k \mid s)$. Then, we use that, for the softmax policy,

$$\frac{\partial \log \pi_\theta^k(a' \mid s)}{\partial z_{sa}^k} = 1\{a = a'\} - \pi_\theta^k(a \mid s)$$

and, for vanilla policy gradient methods,

$$z_{sa}^{k+1} - z_{sa}^k = \eta \pi_\theta^k(a \mid s) A^{\pi_\theta^k}(s, a).$$

Remark 1: This proposition shows that the divergence from the reference policy is positive only insofar as there is a strong positive covariance between the log probability of an action and the advantage. In other words, without any entropy bonus, the divergence happens only due to *sharpening* the distribution around high reward actions.

Proposition 2. *Let $\psi(\pi_\theta^k \mid s) = D_{KL}(\pi_\theta^k(\cdot \mid s) \parallel \pi_{ref}(\cdot \mid s))$. Then, for vanilla policy gradient methods with an entropy bonus coefficient α,*

$$\begin{aligned}\psi(\pi_\theta^{k+1} \mid s) - \psi(\pi_\theta^k \mid s) = {} & \operatorname{Cov}_{a \sim \pi_\theta^k(\cdot \mid s)} \left(\log \frac{\pi_\theta^k(a \mid s)}{\pi_{ref}(a \mid s)} + 1, \pi_\theta^k(a \mid s) A^{\pi_\theta^k}(s, a) \right) \\ & + \alpha \operatorname{Cov}_{a \sim \pi_\theta^k(\cdot \mid s)} \left(\log \frac{\pi_\theta^k(a \mid s)}{\pi_{ref}(a \mid s)} + 1, -\pi_\theta^k(a \mid s) \log \pi_\theta^k(a \mid s) \right) + C\end{aligned}$$

where C is a constant.

Proof Sketch: The proof is similar to that of Proposition 1, except we must add the contribution of the entropy bonus to $z_{sa}^{k+1} - z_{sa}^{k}$.

Remark 2: In this case, we see that the divergence can also increase when there is a positive covariance between the log probability of an action and its contribution to the entropy. In other words, the divergence comes from an incentive to approach a uniform distribution over all actions. It is clear that these two goals are not necessarily mutually inclusive and can be quite difficult to balance. This proposition sheds light on the importance of carefully tuning the entropy coefficient α.

This theoretical analysis confirms the following intuition:

Policy gradient methods incorporating an entropy bonus must balance two conflicting incentives: the incentive to sharpen the distribution around actions yielding high advantages, and the incentive to add equal probability mass to all actions. The ability to balance these incentives relies heavily on the entropy bonus coefficient α and adapting it throughout the learning process.

While this confirms the intuition that reinforcement learning fine-tuning sharpens the distribution around high advantage actions, we ask whether there is any bias from the pre-training that affects this sharpening. In particular, suppose the pre-trained policy has a large bias towards one action. However, during RLFT, if the policy discovers that there is an alternative action with larger advantage, which action does RLFT sharpen the distribution around? To do so, we first look at the optimal policy. In particular, we consider the original optimization problem that PPO considers (Schulman et al., 2017b,a) (see equation 1 and equation 2). Next, we consider the theoretical optimal policy satisfying the (weakened) Lagrangian formulation (Peng et al., 2019) and observe the following:

Proposition 3. *Consider a fixed state s and action a. Consider the optimal policy π^* that solves the optimization problem considered by PPO: maximize expected returns while minimizing divergence from the pre-trained behavior policy used to collect data, π_{ref}. Then, if $\pi_{ref}(a' \mid s) = x \cdot \pi_{ref}(a \mid s)$. Then, π^* satisfies:*

$$\frac{\pi^*(a' \mid s)}{\pi^*(a \mid s)} = x \cdot \exp\left(\frac{1}{\beta}(\hat{A}^{\pi_{ref}}(s, a') - \hat{A}^{\pi_{ref}}(s, a))\right).$$

Remark 3: Observe that if both actions, a and a', are equally advantageous, the optimal policy π^* remains x-times more biased towards a'. In particular, if x is positive i.e. the pre-trained policy is more biased towards a' than a and if the *true* advantage of a and a' are equal, then the optimal policy will forego this bias only if the estimated advantage is biased towards a.

This suggests the following takeaway:

Reinforcement learning fine-tuning does not only sharpen the distribution around high advantage actions but also around actions that our pre-trained policy is biased towards.

4 Experimental Setup

Our experiments follow a four-stage pipeline: (1) training a suboptimal expert policy from scratch, (2) collecting expert rollouts, (3) pre-training via behavior cloning, and (4) reinforcement learning fine-tuning with varying entropy coefficients.

Environments. We evaluate on four Atari games—*Breakout*, *Gravitar*, *Berzerk*, and *Private Eye*—selected to reflect a range of exploration difficulty. Environments are processed using standard Atari wrappers: grayscale conversion, downsampling to 84×84, frame skip of 4, and a stack of the last 4 frames.

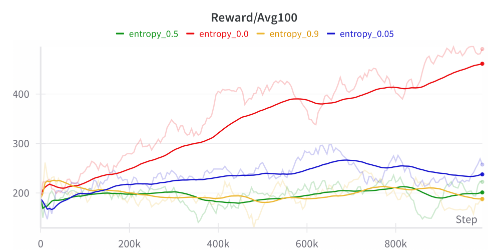
Policy. We use a CNN architecture similar to Mnih et al. (2016), with three convolutional layers and a 512-unit fully connected layer. The actor outputs logits over discrete actions; the critic predicts a scalar value. Both networks are trained using Adam.

Pretraining. We first train a PPO agent from scratch for 1M steps to obtain an expert policy. This policy is used to generate 500 episodes of data, which are then used to train a new policy via behavior cloning using cross-entropy loss. This cloned policy is used as the initialization for all RLFT experiments.

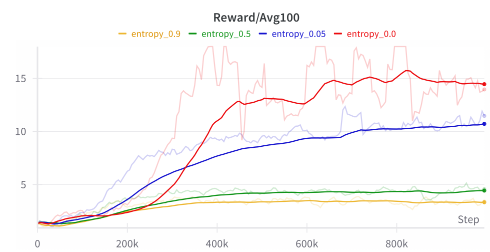
Fine-Tuning. The cloned policy is fine-tuned using PPO for an additional 1M steps under entropy coefficients $\{0.00,\ 0.05,\ 0.5,\ 0.9\}$. We track reward, success rate, entropy, KL divergence from the pretrained policy, and critic loss throughout training.

5 Analysis

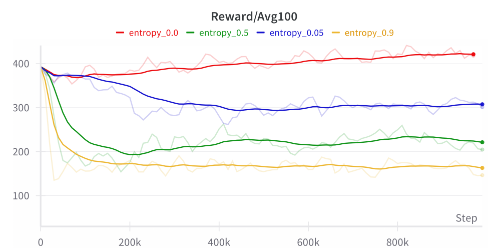
5.1 Reward and Success Rate Analysis



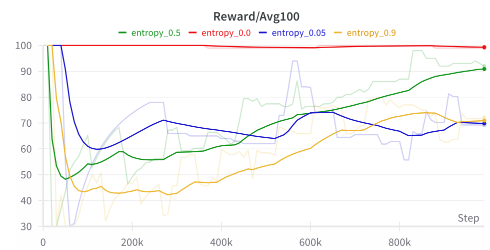
(a) Gravitar



(b) Breakout



(c) Berzerk



(d) Private Eye

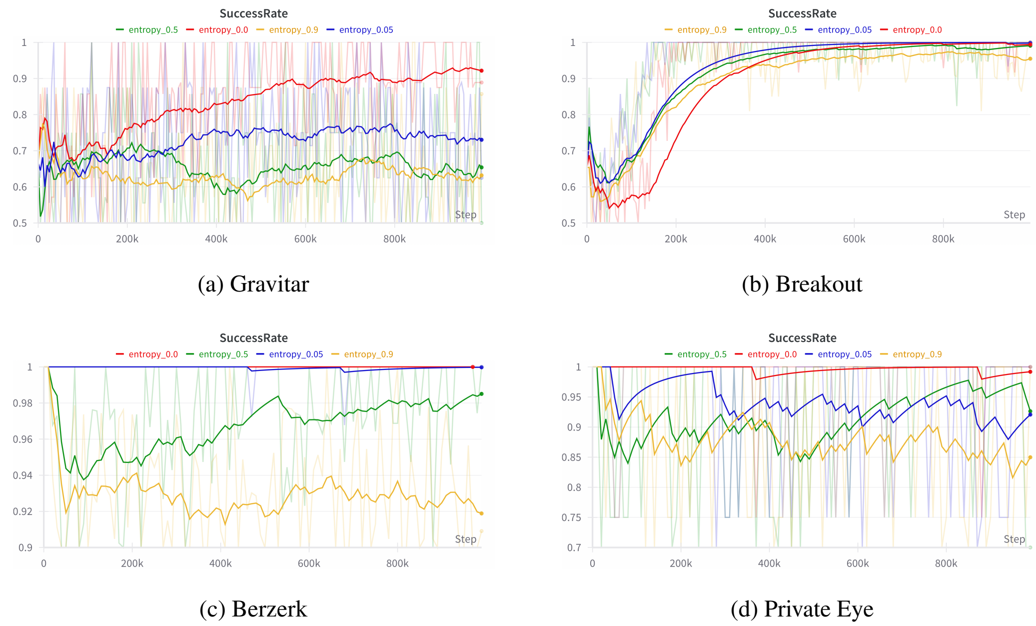
Figure 1: Reward curves across four Atari environments. Gravitar and Breakout show steady improvements; Berzerk and Private Eye display more unstable or plateauing learning behavior.

Our analysis of reward and success rate curves reveals several key insights regarding the role of entropy regularization in policy fine-tuning. First, introducing an entropy bonus leads to an immediate drop in rewards and success rates, indicating that the policy begins to "unlearn" behaviors acquired during pre-training. This effect is especially pronounced as the entropy coefficient increases: higher entropy coefficients cause the decline in performance to occur earlier in training, effectively accelerating the unlearning process.

Moreover, maintaining a large entropy coefficient throughout training often prevents the policy from regaining its previous performance. Even after one million training steps, policies with a high entropy bonus frequently fail to recover the rewards and success rates achieved by the pretrained policy, which calls into question the utility of reinforcement learning from feedback in such settings. In contrast, in relatively easier environments such as Gravitar and Breakout, the addition of entropy does not cause significant unlearning. However, we observe that increasing the entropy coefficient slows down performance improvements, and in some cases, such as Gravitar, larger entropy coefficients provide little to no additional benefit.

Taken together, these findings suggest the following:

Entropy bonuses in reinforcement learning fine-tuning may undermine the advantages of pre-training by causing rapid unlearning of behaviors learned from the expert, especially in more challenging environments with large action spaces. Careful tuning of the entropy coefficient is thus essential to balance exploration and retention of pretrained behaviors.



(a) Gravitar

(b) Breakout

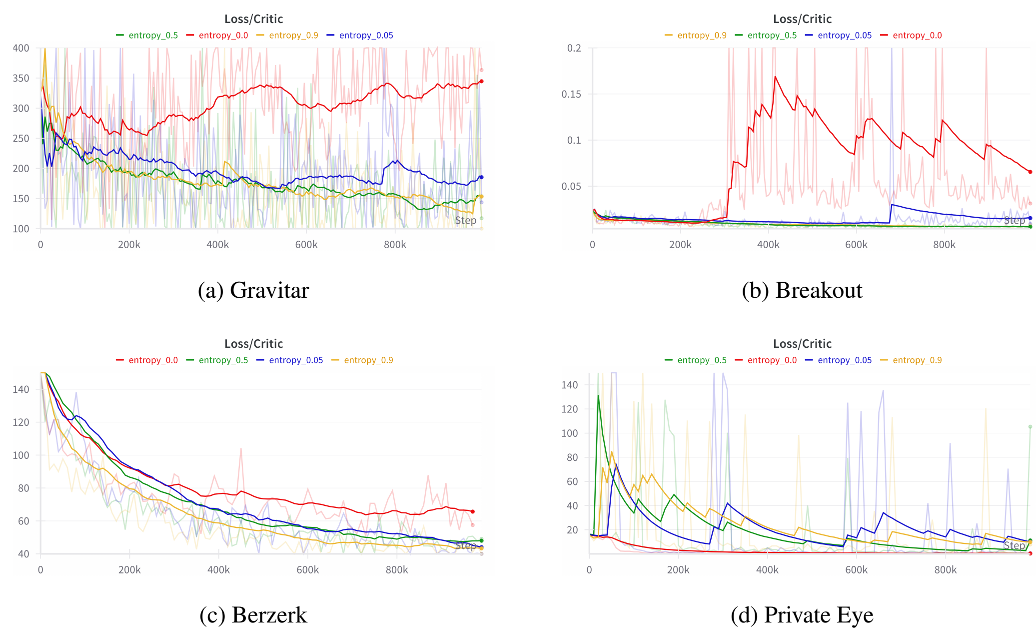
(c) Berzerk

(d) Private Eye

Figure 2: Success rate across four Atari environments. Breakout and Gravitar show stable gains, while Berzerk and Private Eye reveal more unstable patterns.

5.2 Critic Loss Analysis

5.2.1 Change in Critic Loss through out finetuning



(a) Gravitar

(b) Breakout

(c) Berzerk

(d) Private Eye

Figure 3: Critic loss trends across four Atari environments. Stable and decreasing loss generally indicates effective learning of state values.

We examine the evolution of critic loss across our four environments under varying entropy coefficients as shown in Figure 3. Critic loss reflects how well the value function approximates the expected return; therefore, a stable and steadily decreasing critic loss, in general, demonstrates the effective learning of the state values, whereas highly unstable values indicate insufficient learning.

Our analysis reveals three key findings: First, adding entropy bonus in most environments (such as Gravitar, Berzerk, and Breakout) improves critic learning. We see that critic loss converged smoothly to a lower value for the non-zero entropy coefficients, unlike that of the zero entropy coefficient which sometimes never decreased at all. This suggests that added entropy not only improves exploration, but also leads to a more stable value estimation. Second, the high values and unstable nature of the zero entropy coefficient reinforces the hypothesis that entropy collapse reduces exploration, resulting in insufficient diverse dataset, which may negatively impact the critic's ability to learn.

Interestingly, Private Eye presents a unique case. Even though the critic loss for all the entropy coefficients differs initially, they all converge to approximately the same final value. This shows that despite the initial instability, the value approximator learned a similar solution - probably due to the deterministic gameplay in the Private Eye environment.

The overall insight from these findings is the following:

Entropy bonuses, aside from promoting exploration, play an essential role in stabilizing critic learning and learning more accurate critic networks during fine-tuning.

5.2.2 Average Critic Loss vs. Number of Unique Actions Taken from State

We further investigate the relationship between critic loss and the number of unique actions taken from a state in the Private Eye environment under varying entropy coefficients. Our grouped scatter plots Figures 4–6 visualize the average critic loss against the diversity of actions from a state, with bubble size representing the number of states in each group. For entropy coefficient 0.0 (Figure 4), we observe that states associated with a higher number of unique actions tend to exhibit higher critic loss, suggesting that the critic struggled to fit the data in regions of the state space where the policy was less deterministic. However, as entropy increases, this trend weakens. For entropy 0.5 (Figure 5), the relationship remains weakly positive but less pronounced, and for entropy 0.9 (Figure 6), we even observe a slight inverse correlation, with critic loss decreasing as action diversity increases. This reversal indicates that high entropy may facilitate better critic learning in more exploratory regions, mitigating the challenges of fitting high-variance data. However, due to the lack of a consistent pattern and the complexity of the dynamics involved, we conclude that this relationship warrants further investigation in future work.

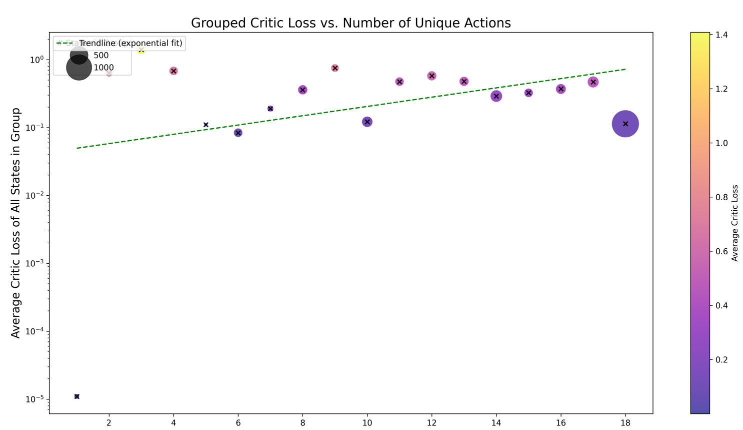


Figure 4: Private Eye (Entropy 0.0)

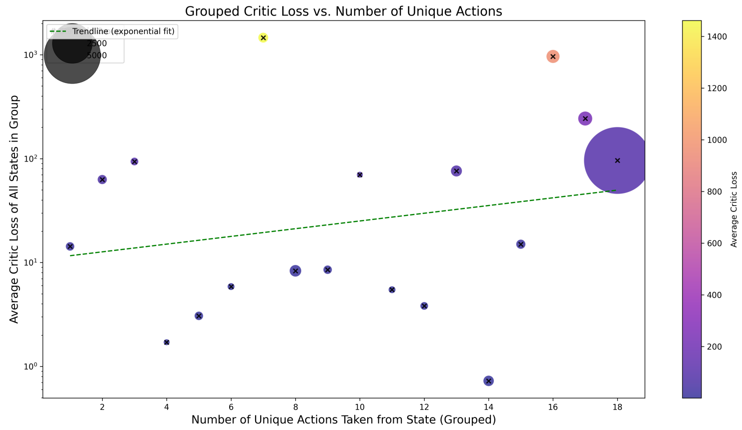


Figure 5: Private Eye (Entropy 0.5)

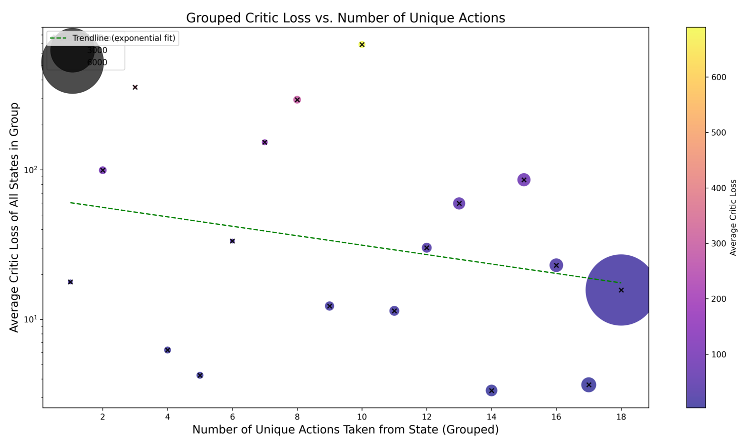
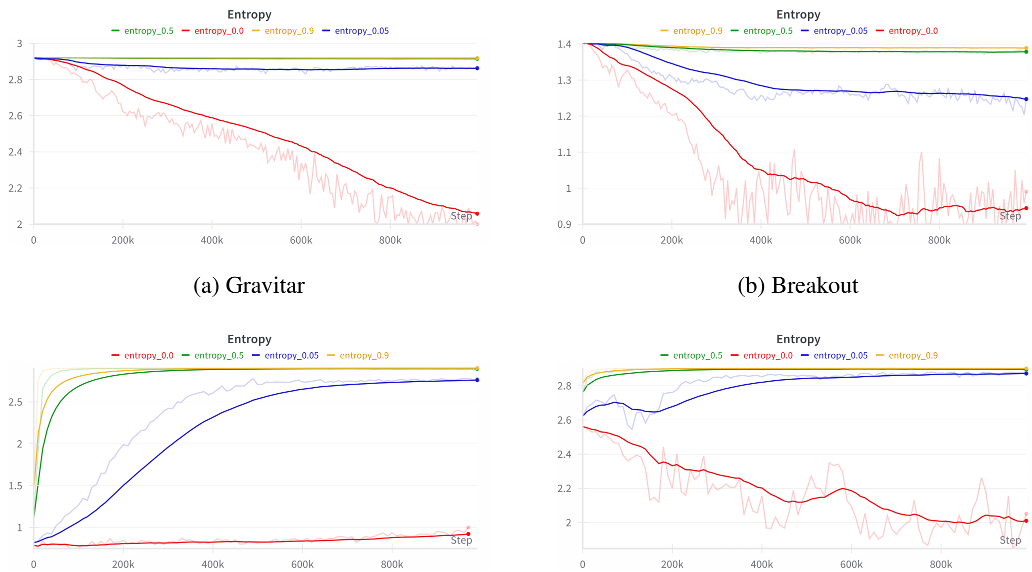


Figure 6: Private Eye (Entropy 0.9)

5.3 Entropy analysis

5.3.1 Change in Entropy through out finetuning



(a) Gravitar

(b) Breakout

(c) Berzerk

(d) Private Eye

Figure 7: Entropy trends across four Atari environments. A clear downward trend in entropy, especially for lower coefficients like 0.0 and 0.05, indicates increasing policy confidence and reduced randomness as training progresses. Higher entropy coefficients (e.g., 0.5, 0.9) maintain broader exploration longer.

Our analysis of entropy dynamics across four Atari environments reveals several important trends in how entropy regularization shapes the evolution of the policy's action distribution. When no entropy bonus is applied ($c_2 = 0.0$), entropy decreases sharply throughout training, indicating that the policy rapidly collapses onto a narrow set of high-reward actions. This collapse is consistent with standard PPO behavior and reflects a lack of sustained exploration.

Surprisingly, when any nonzero entropy bonus is applied, we observe the opposite pattern: entropy initially increases rapidly during early training. This suggests that the bonus encourages the policy to temporarily diversify its action distribution and explore alternative behaviors. However, this effect saturates quickly. After the initial rise, entropy levels plateau and remain roughly constant for the remainder of training. Moreover, across all environments, entropy values converge to a common ceiling that corresponds to the maximum entropy achievable for the given action space.

This convergence implies that increasing the entropy coefficient beyond a small threshold (e.g., from 0.05 to 0.9) does not produce proportionally more exploration. Instead, larger coefficients merely accelerate the early rise in entropy, after which all policies stabilize at the same entropy level. This challenges the assumption that higher entropy bonuses always induce greater diversity or broader exploration during fine-tuning.

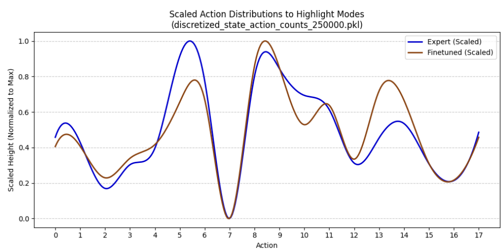
Taken together, these results suggest the following:

While Entropy bonuses do prevent premature entropy collapse, their long-term effect is bounded. The policy reaches a regime of maximal entropy early in training and remains there, regardless of the magnitude of the coefficient. This happens as the policy unlearns the behaviors learned during pre-training, approximating a uniform distribution and then spending the rest of the reinforcement learning fine-tuning phase attempting to sharpen the distribution around high advantage actions. As such, the entropy bonus helps in letting go of the pre-trained policy's biases. Unless the entropy bonus is lowered later on, this sharpening cannot be done causing the RLFT policy to consistently achieve low returns and successes.

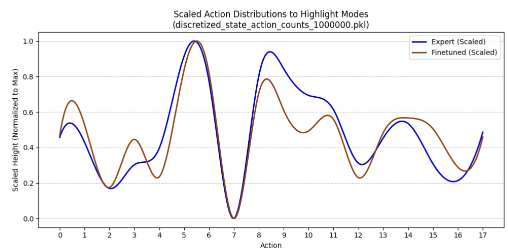
This raises the question of whether larger entropy coefficients meaningfully increase exploration depth or simply delay exploitation—an issue we explore further in the next section.

5.4 Action Distribution Evolution Across Entropy Coefficients

5.4.1 Berzerk

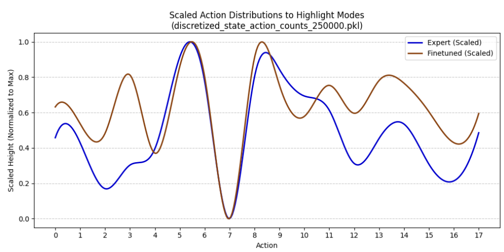


(a) Step 250,000 — Entropy 0.0

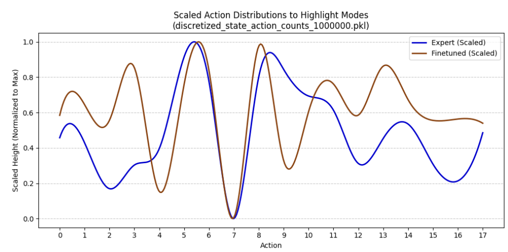


(b) Step 1,000,000 — Entropy 0.0

Figure 8: Action distribution over time for entropy coefficient 0.0.



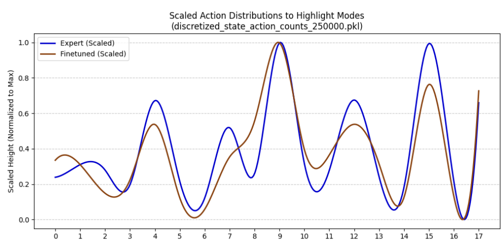
(a) Step 250,000 — Entropy 0.9



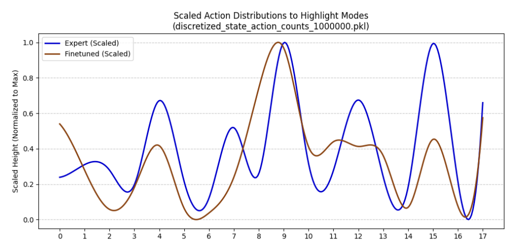
(b) Step 1,000,000 — Entropy 0.9

Figure 9: Action distribution over time for entropy coefficient 0.9.

5.4.2 Private Eye

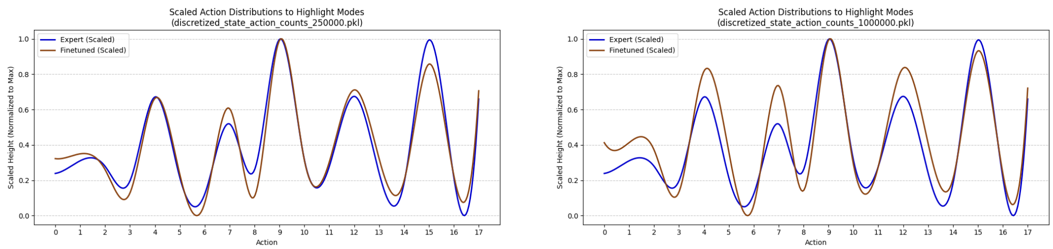


(a) Step 250,000 — Entropy 0.0



(b) Step 1,000,000 — Entropy 0.0

Figure 10: Private Eye action distribution over time for entropy coefficient 0.0.



(a) Step 250,000 — Entropy 0.9

(b) Step 1,000,000 — Entropy 0.9

Figure 11: Private Eye action distribution over time for entropy coefficient 0.9.

In this section, we examine how the action distribution changes when we use PPO with varying entropy coefficients. For very low entropy coefficients (0.0 and 0.05), reinforcement learning from feedback (RLFT) primarily sharpens the action distribution around the same modes as the pretraining policy. This can be seen in figure 10b and in figure 8b where we see that fine-tuned policy peaks around the same actions as the pre-trained policy. In these cases, the peaks of the RLFT policy align closely with those of the original expert, indicating that the overall structure of the action distribution remains largely unchanged. Interestingly, with very little entropy bonus, reinforcement learning fine-tuning causes the models' learned probability distribution to have the same local optima as the pre-trained policy - RLFT sharpens the distribution but preserves the local optima.

As the entropy coefficient increases, however, the policy increasingly scales up the probability mass across all actions, moving toward a more uniform distribution. This is especially evident for the highest entropy coefficients (see figure 11b and figure 9b), where the distinction between the modes becomes less pronounced. In particular, this entropy bonus does not cause the model to discover a different mode altogether, even with mild entropy coefficients like 0.05. Interestingly, even with larger entropy coefficients, the policy tends to preserve the local minima and maxima found in the original distribution, but the probability mass is distributed more broadly, and the peaks are less sharp. In effect, high entropy regularization widens the distribution while maintaining the overall directionality imposed by pretraining, rather than fundamentally altering the locations of the modes.

These observations highlight the following:

Entropy regularization during reinforcement learning fine-tuning mainly controls the sharpness and spread of the action distribution by adding large variance around the same modes as in the pre-training, rather than shifting the policy toward fundamentally different or new behaviors.

6 Discussion

For future works, we would like to explore adaptive methods for choosing the entropy coefficient. We would also like to explore other regularization methods than the entropy bonus. In particular, we would like to consider reward functions that, given k trajectories, rewards fundamentally diverse attempts that are also scaled by rewards similar to Tang et al. (2025).

7 Conclusion

Our results indicate that the entropy bonus can be a useful tool in enabling learning better critic functions via a more stable learning curve. However, the entropy coefficient can be a very difficult hyperparameter to tune and requires precise adapative methods. Otherwise, it can cause policies to unlearn behaviors learned during pre-training and can prevent the models from finding high reward behaviors. More importantly, the entropy bonus is insufficient in enabling the policy to truly explore and discover new behaviors.

8 Team Contributions

- **Ifdita Hasan Orney:** Implemented code for pretraining, PPO, and experiment plots; ran experiments for **Gravitar**, and analysis. Contributed to write up.
- **Iddah Mlauzi:** Wrote code for PPO. Ran pretraining and finetuning for **Berzerk** and **Private Eye**. Ran analysis experiments. Contributed to write up.
- **George Kojo Frimpong Birikorang:** Ran pretraining and finetuning for **Breakout**, ran experiments for analysis. Contributed to write up.

Code Repository: `github.com/ifdita-hasan/Exploration-Policy`

References

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research* 47 (June 2013), 253–279. https://doi.org/10.1613/jair.3912

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. 2016. Unifying Count-Based Exploration and Intrinsic Motivation. arXiv:1606.01868 [cs.AI] https://arxiv.org/abs/1606.01868

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. 2018. Exploration by Random Network Distillation. arXiv:1810.12894 [cs.LG] https://arxiv.org/abs/1810.12894

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. 2025. The Entropy Mechanism of Reinforcement Learning for Reasoning Language Models. arXiv:2505.22617 [cs.LG] https://arxiv.org/abs/2505.22617

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. arXiv:1801.01290 [cs.LG] https://arxiv.org/abs/1801.01290

Jubayer Ibn Hamid. 2025. Deep Reinforcement Learning Notes. https://jubayer-ibn-hamid.github.io/data/RL_Notes__final_.pdf. Available online..

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2017. Rainbow: Combining Improvements in Deep Reinforcement Learning. arXiv:1710.02298 [cs.AI] https://arxiv.org/abs/1710.02298

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. 2025. Tulu 3: Pushing Frontiers in Open Language Model Post-Training. arXiv:2411.15124 [cs.CL] https://arxiv.org/abs/2411.15124

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2019. Continuous control with deep reinforcement learning. arXiv:1509.02971 [cs.LG] https://arxiv.org/abs/1509.02971

Jianlan Luo, Zheyuan Hu, Charles Xu, You Liang Tan, Jacob Berg, Archit Sharma, Stefan Schaal, Chelsea Finn, Abhishek Gupta, and Sergey Levine. 2025. SERL: A Software Suite for Sample-Efficient Robotic Reinforcement Learning. arXiv:2401.16013 [cs.RO] https://arxiv.org/abs/2401.16013

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. arXiv:1602.01783 [cs.LG] https://arxiv.org/abs/1602.01783

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg

Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. 2024. OpenAI o1 System Card. arXiv:2412.16720 [cs.AI] `https://arxiv.org/abs/2412.16720`

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. 2017. Curiosity-driven Exploration by Self-supervised Prediction. arXiv:1705.05363 [cs.LG] `https://arxiv.org/abs/1705.05363`

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. arXiv:1910.00177 [cs.LG] `https://arxiv.org/abs/1910.00177`

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. 2017a. Trust Region Policy Optimization. arXiv:1502.05477 [cs.LG] `https://arxiv.org/abs/1502.05477`

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2018. High-Dimensional Continuous Control Using Generalized Advantage Estimation. arXiv:1506.02438 [cs.LG] `https://arxiv.org/abs/1506.02438`

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017b. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG] `https://arxiv.org/abs/1707.06347`

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. arXiv:1712.01815 [cs.AI] `https://arxiv.org/abs/1712.01815`

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). The MIT Press. `http://incompleteideas.net/book/the-book-2nd.html`

Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Rémi Munos. 2025. Optimizing Language Models for Inference Time Objectives using Reinforcement Learning. arXiv:2503.19595 [cs.LG] `https://arxiv.org/abs/2503.19595`

Peter West and Christopher Potts. 2025. Base Models Beat Aligned Models at Randomness and Creativity. arXiv:2505.00047 [cs.CL] `https://arxiv.org/abs/2505.00047`

R. J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8 (1992), 229–256.

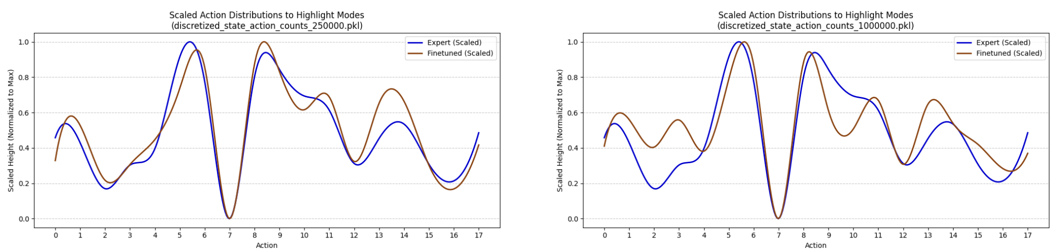
An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 Technical Report. arXiv:2505.09388 [cs.CL] `https://arxiv.org/abs/2505.09388`

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? arXiv:2504.13837 [cs.AI] https://arxiv.org/abs/2504.13837

A Additional Experiments

A.1 Additional plots for Action Distribution Evolution Across Entropy Coefficients

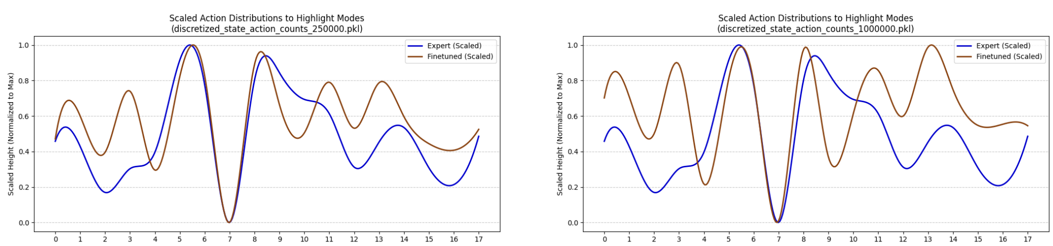
A.1.1 Berzerk



(a) Step 250,000 — Entropy 0.05

(b) Step 1,000,000 — Entropy 0.05

Figure 12: Action distribution over time for entropy coefficient 0.05.

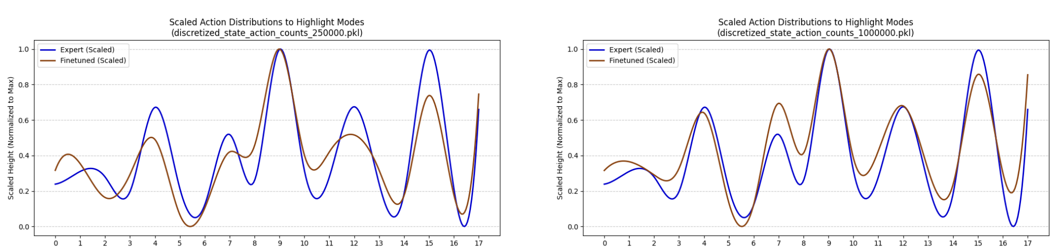


(a) Step 250,000 — Entropy 0.5

(b) Step 1,000,000 — Entropy 0.5

Figure 13: Action distribution over time for entropy coefficient 0.5.

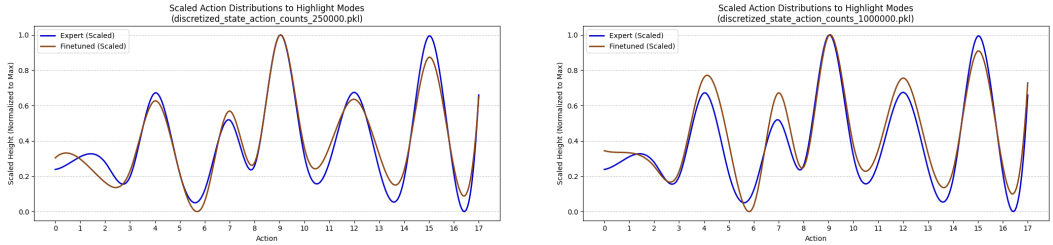
A.1.2 Private Eye



(a) Step 250,000 — Entropy 0.05

(b) Step 1,000,000 — Entropy 0.05

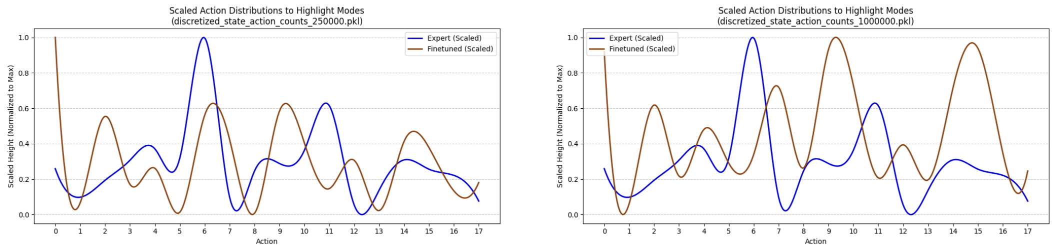
Figure 14: Private Eye action distribution over time for entropy coefficient 0.05.



(a) Step 250,000 — Entropy 0.5 (b) Step 1,000,000 — Entropy 0.5

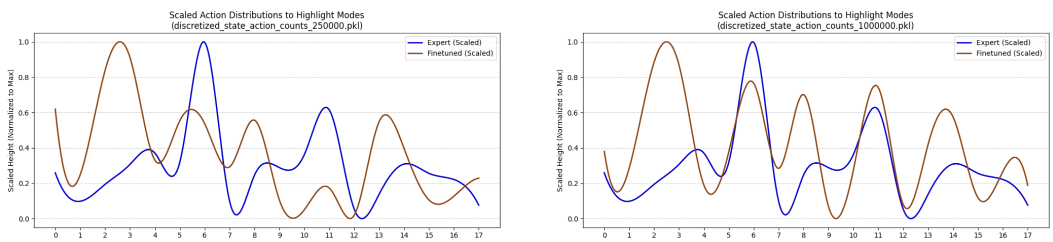
Figure 15: Private Eye action distribution over time for entropy coefficient 0.5.

A.1.3 Gravitar



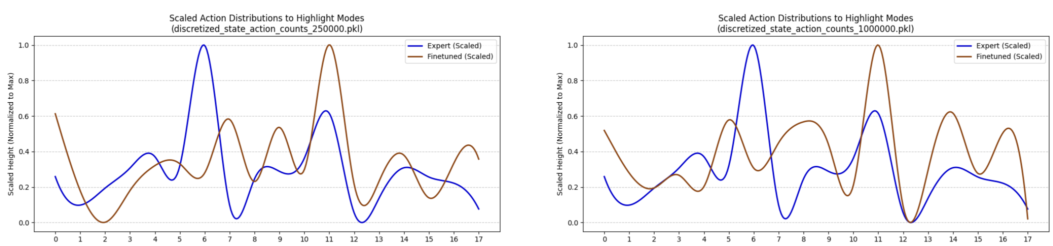
(a) Step 250,000 — Entropy 0.0 (b) Step 1,000,000 — Entropy 0.0

Figure 16: Gravitar action distribution over time for entropy coefficient 0.0.



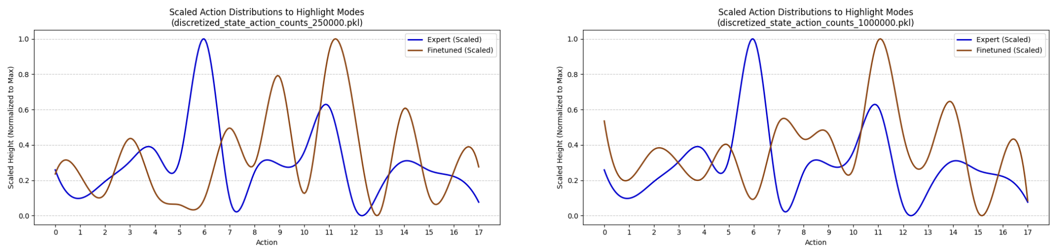
(a) Step 250,000 — Entropy 0.05 (b) Step 1,000,000 — Entropy 0.05

Figure 17: Gravitar action distribution over time for entropy coefficient 0.05.



(a) Step 250,000 — Entropy 0.5 (b) Step 1,000,000 — Entropy 0.5

Figure 18: Gravitar action distribution over time for entropy coefficient 0.5.

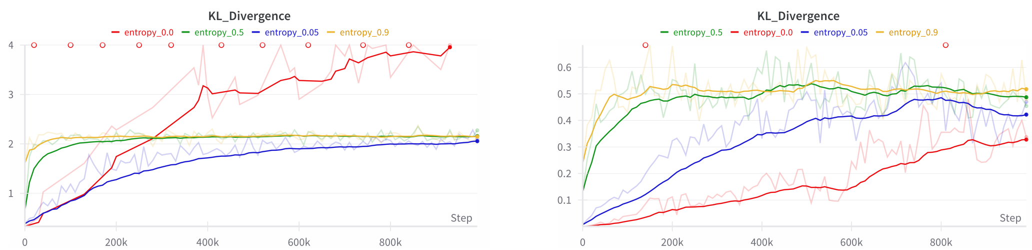


(a) Step 250,000 — Entropy 0.9

(b) Step 1,000,000 — Entropy 0.9

Figure 19: Gravitar action distribution over time for entropy coefficient 0.9.

A.2 KL Analysis



(a) Berzerk

(b) Private Eye

Figure 20: KL Divergence from Pretrained Policy across Entropy Coefficients.

We plot the KL divergence between the fine-tuned and pretrained policies to assess how much the policy shifts during training.

In Berzerk (Figure 20a), higher entropy coefficients lead to larger and faster deviations, indicating more exploration. In contrast, Private Eye (Figure 20b) shows smaller, more stable shifts across all entropy levels—likely due to its deterministic structure.

These trends confirm that entropy affects not only exploration but also how far the policy moves from its initialization.