Stochastic Policies, Deterministic Minds: A Calibrated Evaluation Protocol and Diagnostics for Deep Reinforcement Learning

Sooyoung Jang[1], Seungho Yang[2], Changbeom Choi[3]*
Department of Computer Engineering, Hanbat National University, Daejeon, 34158, Republic of Korea[1,3]
Department of Urban Engineering, Hanbat National University, Daejeon, 34158, Republic of Korea[2]

***Abstract*—Deep reinforcement learning (DRL) typically involves training agents with stochastic exploration policies while evaluating them deterministically. This discrepancy between stochastic training and deterministic evaluation introduces a potential objective mismatch, raising questions about the validity of current evaluation practices. Our study involved training 40 Proximal Policy Optimization agents across eight Atari environments and examined eleven evaluation policies ranging from deterministic to high-entropy strategies. We analyzed mean episode rewards and their coefficient of variation while assessing one-step temporal-difference errors related to low-confidence actions for value-function calibration. Our findings indicate that the optimal evaluation policy is highly dependent on the environment. Deterministic evaluation performed best in three games, while low-to-moderate-entropy policies yielded higher returns in five, with a significant improvement of over 57% in Breakout. However, increased policy entropy generally degraded stability—evidenced by a rise in the coefficient of variation in Pong from 0.00 to 2.90. Additionally, low-confidence actions often revealed an over-optimistic value function, exemplified by negative TD errors, including -10.67 in KungFuMaster. We recommend treating evaluation-time entropy as a tunable hyperparameter, starting with deterministic or low-temperature softmax settings to optimize both return and stability on held-out seeds. These insights provide actionable strategies for practitioners aiming to enhance their DRL-based agents.**

***Keywords*—Deep reinforcement learning; policy evaluation; stochastic policy; temporal difference error; Atari; PPO**

I. Introduction

Deep reinforcement learning has emerged as a powerful paradigm for solving complex sequential decision-making problems, with notable successes in domains ranging from game playing to robotics [1], [2]. Among the most successful families of algorithms are policy gradient methods, which directly optimize the parameters θ of an agent's policy, π_θ. Algorithms like Proximal Policy Optimization (PPO) [3] learn a stochastic policy, which maps states to a probability distribution over actions, $\pi_\theta(a|s)$. For discrete action spaces, as found in the Atari Learning Environment, this policy is typically represented by a categorical distribution, often parameterized by the softmax output of a neural network.

The stochastic nature of the policy is not an incidental feature; it is fundamental to the learning process. During training, sampling actions from the policy distribution enables exploration, allowing the agent to discover novel state-action pathways that may lead to higher rewards. This exploration is essential for escaping local optima and preventing premature convergence to a suboptimal policy. Indeed, the policy gradient theorem [4], the theoretical foundation of these methods, relies on this stochasticity to estimate the gradient of the expected return.

In stark contrast to the training phase, the common practice during evaluation or deployment is to render the policy deterministic by selecting the action with the highest probability for any given state (i.e., taking the argmax of the policy's output distribution). This deterministic approach is predicated on the principle of exploitation. Once the agent has been sufficiently trained, the optimal strategy is to consistently choose the action it believes to be the best. This creates a fundamental duality: agents are trained as stochastic decision-makers but are often deployed as deterministic ones. This practice, while widespread, rests on an assumption that has received surprisingly little systematic scrutiny. Earlier empirical studies have examined evaluation discrepancies for value-based agents, but only recently has the gap for policy-gradient methods been quantified. In study [5], the authors demonstrate that apparent gains can disappear when stochastic and deterministic evaluations are compared under matched conditions; however, their study does not isolate the causal role of value-function mis-calibration. We extend this thread by providing the first large-scale analysis that links evaluation stochasticity, TD-error structure, and overfitting across eight Atari domains.

This dichotomy between training and evaluation methodologies raises a critical question: is the deterministic policy derived via an argmax operation the most effective implementation of the knowledge encoded within the trained stochastic policy? While intuitively appealing, this assumption is not guaranteed to hold. The optimization objective of PPO maximizes the expected return under the stochastic policy, π_θ, not necessarily the return of its deterministic counterpart. This creates a potential objective mismatch, where the criteria for success during training (effective stochastic exploration and exploitation) differ from the criteria during evaluation (pure deterministic exploitation). There are well-established scenarios where a stochastic policy is not just a tool for exploration but is fundamentally optimal for the task itself. In game-theoretic or multi-agent settings, a deterministic policy can be predictable and easily exploited by an adversary. A classic example is Rock-Paper-Scissors, where the Nash equilibrium corresponds to a uniform stochastic policy [4]. Similarly, in Partially Observable Markov Decision Processes (POMDPs),

*Corresponding author.

where different underlying states may appear identical to the agent (a phenomenon known as perceptual aliasing), a stochastic policy can be crucial for breaking symmetry and avoiding getting stuck. Although Atari games are typically single-agent environments, they are not fully observable, as the agent only sees the current screen's pixels and must infer dynamic information like velocity and trajectory. This partial observability could favor a stochastic approach. The degree of a policy's randomness can be quantified by its entropy. Higher entropy corresponds to a more uniform and exploratory distribution, while lower entropy indicates a more deterministic and exploitative one. During training, policy entropy is often explicitly encouraged via an entropy bonus in the loss function to promote exploration and smooth the optimization landscape. However, its role and optimal level during the evaluation phase remain largely unexamined.

This paper aims to bridge this research gap through a large-scale, systematic empirical investigation. We trained 40 agents using the PPO algorithm across eight distinct Atari environments. These trained agents were then evaluated using a suite of eleven different evaluation policies, ranging from fully deterministic to highly stochastic, allowing us to precisely control and measure the impact of policy entropy on performance. This comprehensive experimental framework allows us to address the following research questions:

- Performance and Stability: How do evaluation policies with varying degrees of stochasticity, as measured by entropy, affect agent performance (mean reward) and stability (reward variance) across different Atari environments?
- Action Pruning: Can a simple heuristic—pruning the least likely actions from the policy's distribution, a form of manual entropy reduction—consistently improve performance?
- Underlying Mechanisms: How do stochastic policies reveal weaknesses in the learned value function? Specifically, can we use the Temporal Difference (TD) error—conceptually, the difference between the predicted value of a state and a more accurate value estimated after taking an action— on low-confidence actions to diagnose inaccurate value predictions and explain performance degradation?

Novelty and contribution. In contrast to prior work that either 1) proposes entropy-regularized objectives to train intrinsically stochastic policies [6], [7] or 2) benchmarks evaluation heuristics without examining the critic, our study treats evaluation stochasticity as a post-training intervention and interrogates its causal impact on performance through a TD-error lens. This framing enables a principled diagnosis of value-function overfitting in cases where stochasticity harms performance and yields actionable guidelines for selecting evaluation protocols that account for this risk.

The remainder of this paper is structured to answer these questions. Section II details the related works, situating our investigation within the context of policy representation and overfitting in RL. Section III describes the methodologies we utilized for the study. Section IV presents a detailed analysis of the experimental results, addressing each research question in turn. Finally, Section V discusses the broader implications of our findings for RL practitioners and outlines directions for future research, concluding with a summary in Section VI.

II. Related Work

Our investigation is situated at the intersection of several key areas in reinforcement learning: policy representation, overfitting and generalization, action space modification, and the role of stochasticity.

A. Policy Representation in Reinforcement Learning

RL algorithms can be broadly categorized by how they represent the agent's policy.

A core distinction in RL is between on-policy and off-policy learning. On-policy algorithms, such as PPO [3], update the policy using only the data collected from the most recent version of that same policy. In contrast, off-policy algorithms, like DDPG [8], can learn from data generated by a different policy, often by using a replay buffer of past experiences. Another key distinction is between stochastic policies, which map states to a probability distribution over actions, and deterministic policies, which map states to a single action. These two distinctions are not synonymous.

1) Stochastic policy methods: Modern on-policy RL was significantly advanced by Trust Region Policy Optimization (TRPO) [9], which introduced a constraint on the Kullback-Leibler (KL) divergence between successive policies to guarantee monotonic improvement and stabilize training. PPO [3] emerged as a more practical and scalable successor, replacing the complex second-order optimization of TRPO with a more straightforward clipped surrogate objective that achieves similar stability. Its robustness and ease of implementation have made PPO a ubiquitous algorithm in the field, with recent advances continuing to refine its optimization process and theoretical underpinnings [10], [11]. Another influential on-policy method, Asynchronous Advantage Actor-Critic (A3C) [12], demonstrated that using multiple parallel actors to interact with the environment could decorrelate the training data and stabilize learning without a replay buffer. These methods all directly learn a stochastic policy, where exploration is inherent to the policy representation itself. While these examples are on-policy, off-policy algorithms that learn stochastic policies, like Soft Actor-Critic (SAC) [7], also exist.

2) Deterministic policy methods: In contrast, another line of research focuses on learning deterministic policies. The Deterministic Policy Gradient (DPG) theorem [13] showed that for deterministic policies, the policy gradient can be computed more efficiently as it does not require integration over the action space. This theoretical advantage motivated the development of algorithms like Deep DPG (DDPG) [8], which combines DPG with deep function approximators for continuous control tasks. These methods are typically off-policy and separate exploration from the policy itself, typically by adding noise to the actions during training only.

This fundamental split in RL approaches—between learning a stochastic policy with built-in exploration and learning a deterministic policy with external exploration—sets the stage for our central research question.

3) Value-based methods and evaluation: The pioneering Deep Q-Network (DQN) [1] learns a value function, Q(s,a), and derives a policy by acting greedily with respect to it. Early work on DQN evaluated agents using an ϵ-greedy policy with a small ϵ (e.g., 0.05) to prevent deterministic cycling in the Atari environments. However, subsequent common practice shifted towards fully greedy evaluation ($\epsilon = 0$) to assess the agent's maximum performance capacity. This historical trend shows the field's implicit convergence on deterministic evaluation as the standard, an assumption our work aims to scrutinize. Complementary evidence comes from [5], who revisit the Atari evaluation methodology and advocate measuring on-policy returns under test-time stochasticity. Our experimental design follows these recommendations but augments them with a TD-error-based causal probe, thereby clarifying why mismatched evaluation protocols degrade performance.

B. Overfitting in Deep Reinforcement Learning

The problem of generalization, a fundamental challenge in deep reinforcement learning (DRL) [14], provides a critical lens through which to view our results. High-capacity models like deep neural networks are prone to memorizing the training data rather than learning generalizable features, a phenomenon that is particularly acute in RL. An agent that overfits to the limited set of trajectories collected through its own on-policy interactions may perform well on states and actions it has seen frequently but fail catastrophically in novel situations [15]. For example, a study [16] demonstrated that overfitting is a primary cause of performance collapse when an agent is transferred from a limited set of training levels to unseen ones, highlighting the need for explicit regularization techniques to promote generalization.

Recent work has provided a more specific vocabulary for diagnosing these failures. The issue is not merely one of standard supervised learning overfitting, but involves instabilities unique to the RL process. A key failure mode is the chain effect of churn [17]. This phenomenon, identified in [17], describes how the value function's predictions for states not in the batch can change uncontrollably after each batch update. This creates a vicious cycle where churn in value estimation and policy improvement compound each other, progressively biasing the learning dynamics. This churn is symptomatic of a deeper issue termed "representation collapse," where the agent's neural networks lose plasticity and feature rank throughout training, rendering them incapable of adapting to new observations or fitting new targets. This collapse is exacerbated by the non-stationarity inherent in RL, where the agent's improving policy continually shifts the data distribution.

This perspective aligns with the concept of specification overfitting, where a system focuses excessively on optimizing a specific, narrow metric—in this case, the TD error on its on-policy data distribution—to the detriment of broader, high-level requirements, such as robustness and generalization. In [18], the authors provided a crucial diagnostic insight, identifying statistical overfitting in the temporal-difference (TD) error as a primary bottleneck. Their key idea is that as an agent trains, its value function becomes highly accurate on its on-policy data distribution but remains inaccurate on out-of-distribution transitions. They propose that this can be diagnosed by measuring the TD error on a held-out validation set of transitions, where an extensive validation TD error indicates poor generalization. This theoretical framework provides a powerful explanation for our empirical findings. The low-confidence actions sampled by our stochastic evaluation policies serve as a naturally occurring validation set. The TD errors on these actions thus act as a direct probe into the value function's generalization capabilities, allowing us to empirically test and validate the overfitting hypothesis proposed in the recent literature.

C. Action Elimination and Pruning

Our experiments with drop policies are related to a line of research on action elimination. An Action Elimination Network (AEN) learns to predict and mask out suboptimal or irrelevant actions in environments with very large discrete action spaces [19]. By reducing the effective size of the action space at each step, their method significantly speeds up learning and improves robustness in complex domains like text-based games. Our drop policies can be seen as a simple, non-learned heuristic inspired by this principle. Instead of learning which actions to eliminate, we manually prune the actions with the lowest logits, allowing us to test the hypothesis that performance can be improved simply by preventing the agent from taking actions it is already least confident about.

D. The Dual Role of Stochasticity: Exploration, Optimality, and Causality

The stochastic nature of the policy in algorithms like PPO is typically framed as a mechanism for exploration, which is essential for discovering novel, high-reward trajectories and preventing premature convergence. However, this view is incomplete. A growing body of research highlights that policy stochasticity plays a more fundamental role in optimization and even optimality itself. In some contexts, a stochastic policy is not merely a means to an end but is the optimal solution. This is well-established in game-theoretic settings [4]. More broadly, the framework of maximum entropy RL formalizes the benefit of randomness by augmenting the standard reward objective with an entropy term [6]. Algorithms like SAC [7] explicitly optimize this objective, learning policies that are not only effective but also as random as possible. The resulting policies can capture diverse strategies for accomplishing a task, thereby improving robustness.

Furthermore, the stochasticity inherent in policy gradient methods is a key mechanism for escaping local optima in the complex, non-convex optimization landscapes of DRL [20]. A purely deterministic policy can easily get trapped in a simple but suboptimal strategy. In contrast, the noise introduced by sampling from a policy distribution can provide the necessary perturbation to jolt the agent out of such traps and discover more sophisticated, higher-reward solutions. This provides a compelling theoretical lens through which to interpret our empirical results in environments like Breakout.

Finally, recent work on the causal foundations of RL provides a formal way to disentangle the sources of randomness in an agent's experience [21]. This framework decomposes the total return of a trajectory into components attributable to the agent's actions and components attributable to the environment's inherent stochasticity. This decomposition is achieved by interpreting the advantage function as the causal

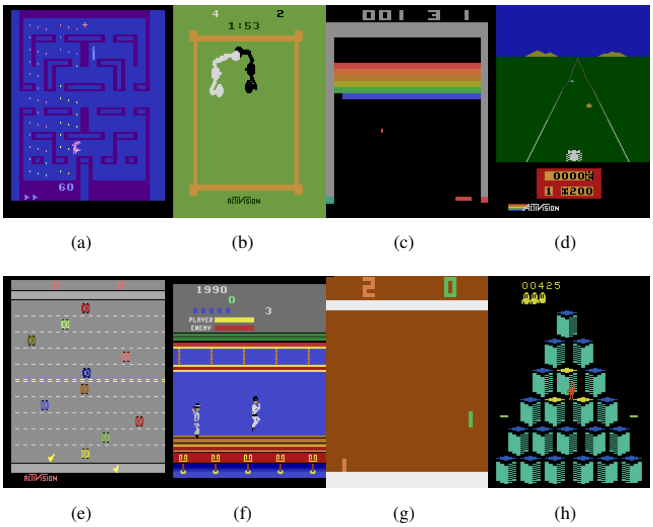

Fig. 1. The eight Atari 2600 environments utilized for the study: (a) Alien, (b) Boxing, (c) Breakout, (d) Enduro, (e) Freeway, (f) KungFuMaster (KFM), (g) Pong, and (h) Qbert.

effect of an action on the return [21]. This distinction clarifies that our investigation focuses on the consequences of the agent's policy stochasticity rather than randomness in the environment's dynamics.

III. Experimental Framework

To systematically investigate the effects of evaluation policy stochasticity, we designed a comprehensive experimental setup involving multiple agents, environments, and evaluation strategies.

A. Environments and Agent Training

Experiments were performed on eight Atari 2600 games from the Arcade Learning Environment (ALE)—Alien, Boxing, Breakout, Enduro, Freeway, KungFuMaster (KFM), Pong, and Qbert (Fig. 1). These titles span a broad spectrum of interaction modalities, reward densities, and action-space cardinalities, thereby providing a representative benchmark suite. The characteristics of each environment, particularly the size of the discrete action space and the corresponding entropy of a random policy, are crucial for interpreting the results and are summarized in Table I.

For each game, five independent PPO agents were trained, yielding a total of 40 trained policies and thereby mitigating the variance induced by random seeds. Training lasted for 1×10^7 environment steps per agent and used the canonical convolutional network for Atari inputs introduced by study [1]. Table II lists the set of PPO hyperparameters that were held constant across all environments.

B. Evaluation Policies

For each of the 40 trained agents, we conducted evaluations using 11 distinct action selection policies. These policies were designed to span a wide spectrum of stochasticity, from fully deterministic to highly random. Each agent-policy combination was evaluated for 100 independent episodes to gather statistically robust performance data. The definitions of the evaluation policies are provided in Table III. Table III provides a concise definition for each of the 11 evaluation policies used to assess the trained agents. logits refers to the raw, pre-softmax output of the policy network.

Justification for linear and drop policies:

1) Linear: The linear policy was designed as a high-entropy alternative to a purely random policy. By preserving the ordinal ranking of the network's logits while drastically flattening the probability distribution, it serves as a hard test case for how the agent performs when it must consider its less-preferred actions more seriously.

2) Drop: The softmax_drop_k and linear_drop_k policies are simple, non-learned heuristics inspired by action-pruning techniques. They allow us to test the hypothesis that manually preventing the agent from taking actions it is already least confident about can function as a form of targeted entropy reduction and improve performance, particularly on otherwise highly random policies.

C. Analysis Metrics

To evaluate performance and diagnose mechanisms, we record the following metrics during evaluation.

1) Performance metrics: The primary performance metric is the *Mean Episode Reward*, averaged over 100 episodes per agent–policy pair. Stability is quantified by the *Coefficient of Variation* (CV), defined as the standard deviation of episode returns divided by their mean. A lower CV indicates more consistent performance relative to the average score.

2) Stochasticity metric: The degree of randomness of the evaluation policy is measured by the average policy entropy,

$$H(q_T) \;=\; \mathbb{E}_t\left[-\sum_{a\in A} q_T(a \mid s_t)\,\log q_T(a \mid s_t)\right], \qquad (1)$$

where $q_T(\cdot \mid s)$ is the evaluation policy (e.g., `softmax_T`) and the expectation is over timesteps.

3) Diagnostic metrics: To investigate the underlying mechanisms of performance differences, we defined two key diagnostic metrics:

a) Low-Confidence Action Ratio (LCAR): Given an evaluation policy $q_T(a \mid s)$ and threshold $\tau \in (0, 1]$, define

$$\mathrm{LCAR}_\tau(s) \;=\; \sum_{a\in A} q_T(a \mid s)\,\mathbf{1}\Big\{\, q_T(a \mid s) < \tfrac{\tau}{|A|}\Big\}. \qquad (2)$$

This quantity equals the total probability mass that $q_T(\cdot \mid s)$ assigns to the *low-confidence* set $\{a : q_T(a \mid s) < \tau/|A|\}$. We report **LCAR-50** ($\tau = 0.5$) and **LCAR-10** ($\tau = 0.1$) as the time-averaged frequencies of such events. *Note on softmax_10.0.* As $T \to \infty$, $q_T(\cdot \mid s)$ approaches the uniform distribution, implying $\mathrm{LCAR}_\tau(s) \to 0$ because $1/|A| > \tau/|A|$

TABLE I. ACTION SPACE SIZE AND ENTROPY OF A RANDOM POLICY IN ATARI ENVIRONMENTS

	Alien	Boxing	Breakout	Enduro	Freeway	KFM	Pong	Qbert
Action Space Size (N)	18	18	4	9	3	14	6	6
Entropy (nats)	2.890	2.890	1.386	2.197	1.099	2.639	1.792	1.792

TABLE II. PPO HYPERPARAMETER CONFIGURATION USED FOR THE ATARI EXPERIMENTS

Hyper-parameter	**Value**
Frame stack	4
Number of parallel environments (n_{envs})	8
Rollout length per environment (n_{steps})	128
SGD epochs per update (n_{epochs})	4
Mini-batch size	256
Total timesteps	1×10^7
Learning-rate schedule	linear, $2.5 \times 10^{-4} \to 0$
Value-function loss coefficient (c_v)	0.5
Entropy coefficient (c_H)	0.01

TABLE III. EVALUATION POLICIES AND THEIR DESCRIPTIONS

Policy	**Description**
deterministic	Selects the action with the highest logit value: $a = \arg\max(\text{logits})$.
softmax_T	Samples an action from the distribution given by $\text{Softmax}(\text{logits}/T)$. Tested for temperatures $T \in \{0.1, 0.5, 1.0, 2.0, 10.0\}$.
linear	Applies an affine shift to make all logits non-negative ($w = v - \min(v)$), then normalizes these values linearly to form a probability distribution.
softmax_drop_k	Removes the k actions with the lowest logit values, then applies a softmax function with $T = 1.0$ to the remaining logits. Tested for $k \in \{1, 2\}$.
linear_drop_k	Removes the k actions with the lowest logit values, then applies the linear normalization scheme to the remaining logits. Tested for $k \in \{1, 2\}$.

for $\tau \in \{0.5, 0.1\}$. At the finite temperature T=10, however, the policy remains only approximately uniform and retains residual structure from the logits. Consequently, a small but nonzero fraction of probability mass can still fall below the threshold. Empirically, this is visible in Qbert, where softmax_10.0 attains LCAR-50 = 5.489% (Table XI). We therefore include softmax_10.0 as a high-entropy reference policy rather than a recommended operating point.

b) Conditional TD error: The one-step TD error at timestep t is

$$\delta_t \;=\; R_{t+1} + \gamma\, V(S_{t+1}) - V(S_t), \tag{3}$$

with discount factor $\gamma = 0.99$ and V the learned value function. We report the average TD error *conditioned* on low-confidence events (e.g., timesteps contributing to LCAR-50 or LCAR-10). This conditional analysis probes the value function's accuracy for actions that the policy itself deems unlikely.

IV. RESULTS AND ANALYSIS

Our experiments yielded a rich dataset that provides clear answers to our research questions, with comprehensive statistics presented in Tables IV to XI for each environment and Table XII for policy entropies. The results, visualized in Fig. 2, reveal a split pattern: in five environments, a low- or moderate-entropy policy outperforms the deterministic baseline, while in three (Boxing, Pong, Qbert) deterministic remains best.

A. RQ1: How do Stochastic vs. Deterministic Policies Affect Performance and Stability?

1) Performance: Across the eight Atari tasks, deterministic evaluation achieves the highest mean reward in three games—Boxing (Table V), Pong (Table X), and Qbert (Table XI). In the remaining five games, a low-to-moderate-entropy policy outperforms deterministic evaluation: Alien (softmax_0.5; Table IV), Breakout (softmax_0.5; Table VI), Enduro (softmax_0.1; Table VII), Freeway (softmax_drop_1; Table VIII), and KFM (softmax_0.1; Table IX). This pattern indicates that modest stochasticity can yield higher returns in a majority of environments, while deterministic remains a strong baseline in the others. This pattern is evident in the summary bar charts in Fig. 2. In several environments, modest stochasticity yields higher returns. In Breakout (Table VI), the moderately stochastic softmax_0.5 policy surpasses the deterministic baseline, scoring 242.02 compared to 153.83 (57.3% improvement). In Freeway (Table VIII), performance differences are modest, but softmax_1.0 and drop variants slightly exceed the deterministic baseline (22.00 and 22.06 vs 21.41), indicating multiple near-optimal evaluation policies. However, in Qbert (Table XI), the deterministic policy scores 3894.00, significantly outperforming the standard stochastic softmax_1.0 policy, which scores 2404.60.

2) Stability: Stochastic evaluation generally inflates return variance and degrades stability. Across many games, higher-entropy evaluation tends to increase the coefficient of variation (CV). However, the relationship is not strictly monotonic in all environments, so we treat the stability impacts of entropy as environment-dependent(as detailed in Table XII). This instability is most pronounced in what we term 'brittle' environments, such as Pong (Table X), where a single suboptimal action can lead to immediate and irreversible negative consequences (e.g., losing a rally). The CV rises from a perfect 0.00 under the deterministic policy to 0.38 under the softmax_2.0 policy. Conversely, in 'robust' environments like Freeway (Table VIII), the agent's success is determined by a longer sequence of actions, and a single suboptimal move is less likely to be catastrophic, allowing for multiple near-optimal policies. However, stability decreases significantly with near-uniform action sampling, as seen with the softmax_10.0, which has a CV of 0.42.

To better understand the differences in both performance and stability, Fig. 3 visualizes the full reward distributions for

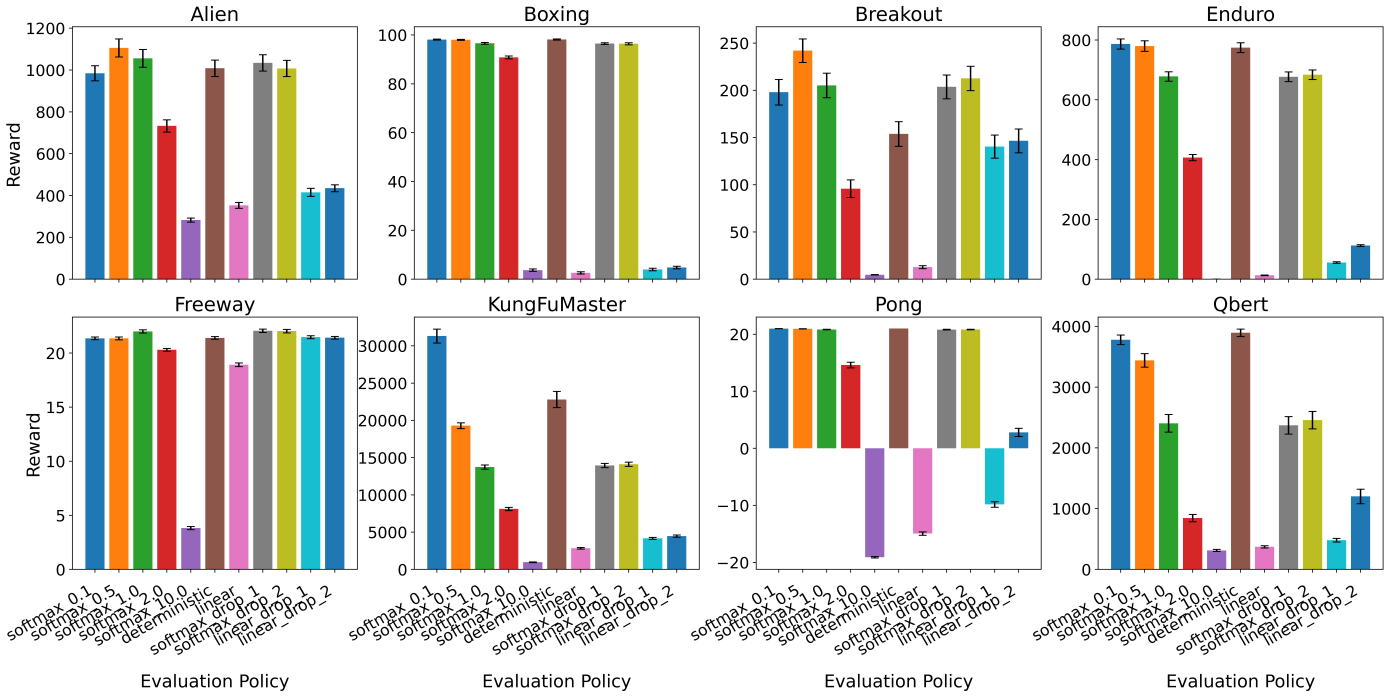

Fig. 2. Mean reward per evaluation policy across eight Atari environments. Error bars represent 95% confidence intervals.

TABLE IV. POLICY-WISE SUMMARY FOR ALIEN: REWARD, LOW CONFIDENCE ACTION RATIO, AND TD ERROR

Policy	Reward				Low Conf 50%		Low Conf 10%	
	Mean	Min	Max	CV	Ratio (%)	TD Error	Ratio (%)	TD Error
softmax_0.1	984.46	460.00	2850.00	0.41	0.489	0.765	0.100	0.518
softmax_0.5	1105.72	470.00	4400.00	0.44	3.075	0.205	0.534	-0.014
softmax_1.0	1055.94	330.00	4080.00	0.46	5.111	0.039	0.835	-0.253
softmax_2.0	732.78	280.00	2860.00	0.45	7.521	-0.497	0.896	-0.983
softmax_10.0	282.78	70.00	1450.00	0.40	2.567	-1.819	0.002	1.368
deterministic	1008.28	470.00	2650.00	0.44	0.000	0.000	0.000	0.000
linear	352.96	80.00	2090.00	0.45	5.438	-1.236	0.117	-0.796
softmax_drop_1	1034.18	270.00	2910.00	0.43	5.024	0.041	0.832	0.059
softmax_drop_2	1007.52	330.00	4370.00	0.44	5.050	0.028	0.826	-0.395
linear_drop_1	415.36	150.00	2170.00	0.53	4.697	-0.923	0.173	-1.274
linear_drop_2	435.18	90.00	1970.00	0.43	4.145	-0.910	0.176	-0.933

the deterministic policy and a representative stochastic policy (softmax_1.0). These plots reveal the underlying structure of the summary statistics. For example, in Pong, the deterministic policy's distribution is a single sharp spike at the maximum score of 21, illustrating its perfect stability (CV of 0.00). In contrast, the distribution for Breakout is bimodal; the stochastic softmax_1.0 policy is visibly more successful at avoiding the low-reward mode, explaining its higher average score. Conversely, in games like Qbert, the deterministic policy's distribution is clearly concentrated at a much higher reward level than that of the stochastic policy, reinforcing its superior performance in that environment.

B. RQ2: What is the effect of action pruning (drop) on performance?

Our analysis of the drop policies, which manually prune the least likely actions, reveals that this heuristic acts as a form of manual entropy reduction whose effectiveness is entirely dependent on the baseline policy's randomness.

- When applied to a focused, low-entropy policy like softmax_1.0, pruning has a negligible impact. The actions being pruned already have near-zero probability of being selected, so their explicit removal does not significantly alter the agent's behavior. For instance, in Alien (Table IV), the mean reward for softmax_1.0 is 1055.94, while softmax_drop_1 and softmax_drop_2 score a comparable 1034.18 and 1007.52, respectively.
- The effect is dramatically different when pruning is applied to a highly random policy such as linear, which assigns substantial probability to actions with low logits. In this context, pruning is highly effective at improving performance by forcing the agent to be more exploitative. In Qbert (Table XI), the linear policy scores a meager 371.55, whereas linear_drop_2 improves this to 1201.20, more than tripling the reward. This shows that performance is fundamentally

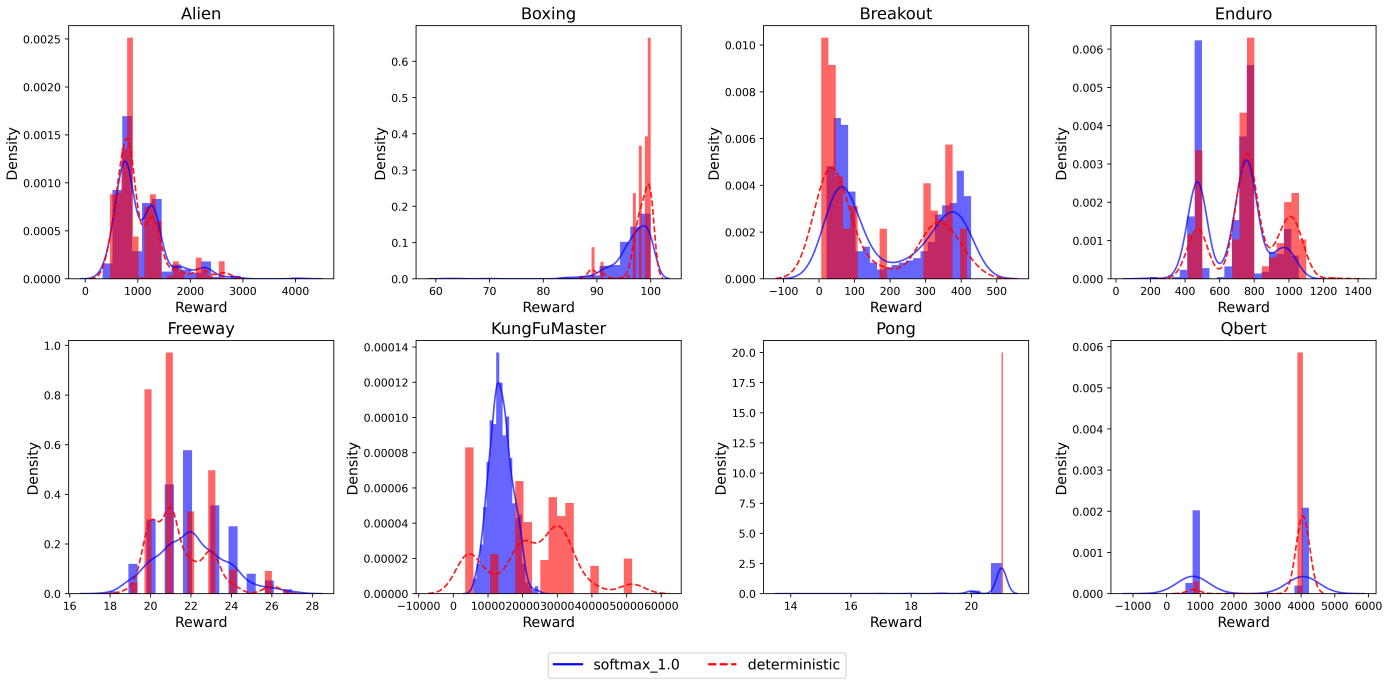

Fig. 3. Reward distribution histograms and kernel density estimates for 100 episodes across eight Atari environments. Only two representative policies (softmax policy with temperature 1.0 and deterministic policy) are shown for clarity.

TABLE V. POLICY-WISE SUMMARY FOR BOXING: REWARD, LOW CONFIDENCE ACTION RATIO, AND TD ERROR

Policy	Reward				Low Conf 50%		Low Conf 10%	
	Mean	Min	Max	CV	Ratio (%)	TD Error	Ratio (%)	TD Error
softmax_0.1	98.11	84.00	100.00	0.02	0.098	0.006	0.020	0.030
softmax_0.5	98.02	84.00	100.00	0.02	0.706	-0.011	0.146	-0.010
softmax_1.0	96.57	62.00	100.00	0.04	1.777	-0.022	0.422	-0.032
softmax_2.0	90.82	42.00	100.00	0.07	5.148	-0.078	1.065	-0.114
softmax_10.0	3.72	-32.00	22.00	1.52	4.056	-0.170	0.010	-0.328
deterministic	98.17	89.00	100.00	0.03	0.000	0.000	0.000	0.000
linear	2.59	-18.00	22.00	2.02	3.925	-0.139	0.074	-0.188
softmax_drop_1	96.48	66.00	100.00	0.04	1.769	-0.032	0.420	-0.046
softmax_drop_2	96.41	46.00	100.00	0.05	1.814	-0.035	0.433	-0.048
linear_drop_1	3.99	-20.00	25.00	1.53	4.896	-0.144	0.120	-0.163
linear_drop_2	4.80	-9.00	25.00	1.22	4.844	-0.137	0.142	-0.153

tied to avoiding low-confidence actions. However, it is crucial to note that even with this significant improvement, the linear_drop_2 policy's score of 1201.20 is still dramatically lower than the deterministic policy's score of 3894.00. This suggests that simply pruning the worst few actions is insufficient to recover high performance. The underlying issue is not just the presence of a few catastrophic actions, but the overall shape of the probability distribution. The linear normalization scheme, even after pruning, assigns a much more uniform probability to the remaining actions than the peaked distribution of a standard softmax policy, leading to continued suboptimal decision-making.

C. RQ3: Can TD Error Explain the Performance Degradation?

While the preceding results demonstrate a correlation between high evaluation entropy and poor returns in many games, our analysis of the TD error provides a causal, mechanistic explanation for this performance degradation. The data reveal that performance degradation is a direct symptom of the agent's value function being poorly generalized outside of the high-density regions of the on-policy data distribution observed during training. The argument is built in three steps.

1) Step 1: Higher entropy forces more low-confidence actions. There is a direct causal link between the stochasticity of the evaluation policy and the frequency of low-confidence actions. As policy entropy increases (Table 12), the evaluation policy q_T allocates more probability to the low-probability tail, increasing the frequency of low-confidence events (e.g., LCAR-50 of Alien rises from 0.489% at $T = 0.1$ to 7.521% at $T = 2.0$; Table IV).

2) Step 2: The value function is systematically overoptimistic for low-confidence actions. Our analysis shows that when the agent is forced to take actions it has learned to avoid, the resulting outcomes are consistently worse than what its

TABLE VI. POLICY-WISE SUMMARY FOR BREAKOUT: REWARD, LOW CONFIDENCE ACTION RATIO, AND TD ERROR

Policy	Reward				Low Conf 50%		Low Conf 10%	
	Mean	Min	Max	CV	Ratio (%)	TD Error	Ratio (%)	TD Error
softmax_0.1	198.01	4.00	426.00	0.78	1.593	-0.122	0.311	-0.200
softmax_0.5	242.02	22.00	429.00	0.59	3.454	0.033	0.524	0.029
softmax_1.0	205.22	21.00	428.00	0.72	5.153	-0.008	0.667	-0.137
softmax_2.0	95.88	7.00	421.00	1.11	6.478	-0.127	0.529	-0.356
softmax_10.0	4.71	0.00	16.00	0.65	1.876	-0.180	0.015	-0.134
deterministic	153.83	6.00	417.00	0.96	0.000	0.000	0.000	0.000
linear	12.79	0.00	373.00	1.38	1.269	-0.032	0.029	-0.056
softmax_drop_1	203.68	15.00	430.00	0.70	5.540	-0.016	2.268	-0.020
softmax_drop_2	212.63	17.00	429.00	0.69	10.709	0.000	9.711	0.005
linear_drop_1	140.39	7.00	428.00	0.99	1.032	-0.091	0.037	-0.073
linear_drop_2	146.49	6.00	417.00	0.98	0.000	0.000	0.000	0.000

TABLE VII. POLICY-WISE SUMMARY FOR ENDURO: REWARD, LOW CONFIDENCE ACTION RATIO, AND TD ERROR

Policy	Reward				Low Conf 50%		Low Conf 10%	
	Mean	Min	Max	CV	Ratio (%)	TD Error	Ratio (%)	TD Error
softmax_0.1	786.96	390.00	1341.00	0.25	0.789	0.000	0.162	0.009
softmax_0.5	779.99	377.00	1359.00	0.26	3.652	-0.017	0.586	-0.027
softmax_1.0	678.18	197.00	1057.00	0.26	5.346	-0.041	0.728	-0.073
softmax_2.0	406.85	135.00	756.00	0.29	6.203	-0.103	0.824	-0.170
softmax_10.0	0.50	0.00	14.00	3.51	2.901	-0.271	0.001	-0.316
deterministic	774.72	415.00	1272.00	0.24	0.000	0.000	0.000	0.000
linear	13.17	0.00	107.00	0.98	4.072	-0.197	0.273	-0.195
softmax_drop_1	677.04	389.00	1057.00	0.27	5.334	-0.041	0.927	-0.065
softmax_drop_2	684.12	192.00	1083.00	0.26	5.396	-0.039	1.266	-0.051
linear_drop_1	55.74	0.00	140.00	0.54	2.297	-0.187	0.166	-0.224
linear_drop_2	112.62	25.00	334.00	0.31	0.956	-0.210	0.030	-0.237

TABLE VIII. POLICY-WISE SUMMARY FOR FREEWAY: REWARD, LOW CONFIDENCE ACTION RATIO, AND TD ERROR

Policy	Reward				Low Conf 50%		Low Conf 10%	
	Mean	Min	Max	CV	Ratio (%)	TD Error	Ratio (%)	TD Error
softmax_0.1	21.37	19.00	26.00	0.07	0.000	0.000	0.000	0.000
softmax_0.5	21.36	19.00	26.00	0.07	0.003	-0.010	0.003	-0.010
softmax_1.0	22.00	18.00	27.00	0.08	0.568	-0.010	0.568	-0.010
softmax_2.0	20.30	15.00	25.00	0.07	8.492	-0.015	1.409	-0.019
softmax_10.0	3.82	0.00	9.00	0.42	0.017	-0.127	0.000	-0.043
deterministic	21.41	19.00	26.00	0.07	0.000	0.000	0.000	0.000
linear	18.92	13.00	24.00	0.10	1.823	-0.017	0.000	0.000
softmax_drop_1	22.06	17.00	27.00	0.08	0.573	-0.009	0.573	-0.009
softmax_drop_2	22.03	17.00	29.00	0.09	0.573	-0.011	0.573	-0.011
linear_drop_1	21.47	19.00	26.00	0.07	0.000	0.000	0.000	0.000
linear_drop_2	21.42	19.00	26.00	0.07	0.000	0.000	0.000	0.000

value function predicted. This observation is consistent with recent evidence that actor-critic value networks become miscalibrated on out-of-distribution state–action pairs, systematically over-estimating returns in precisely the regions we probe here [18]. A negative TD error ($\delta_t < 0$) means the experienced outcome, $R_{t+1}+\gamma V(S_{t+1})$, was significantly worse than what the value function $V(S_t)$ had predicted. The data provides unambiguous evidence for this. In Alien (Table IV), when the softmax_2.0 policy takes a low-confidence (50%) action, the resulting average TD error is -0.497. For the lowest-confidence (10%) actions, the error is -0.983. A similar pattern is observed in KFM (Table IX), where the softmax_2.0 policy yields a TD error of -6.111 and -10.671 on low-confidence actions of 50% and 10%, respectively. When the agent is forced to take an action it deems unlikely, the consequence is almost always worse than it expected.

3) Step 3: This is a clear symptom of value function overfitting. This chain of evidence points to a single, compelling conclusion: the agent's value function is overfit to the high-density regions of the policy distribution it experienced during training. During PPO's on-policy training, the critic (the value function) is trained to minimize TD error on trajectories generated by the current actor policy. This training data is naturally dominated by high-confidence actions. Consequently, the critic becomes an expert at predicting the value of these frequently observed, "good" state-action pairs. Conversely, it sees very few examples of low-confidence actions and their outcomes, leaving the value function undertrained and inaccurate for this part of the state-action space.

When we evaluate with a high-entropy policy, we are effectively sampling from low-density regions of the policy distribution. We force the agent to explore these infrequently

TABLE IX. POLICY-WISE SUMMARY FOR KFM: REWARD, LOW CONFIDENCE ACTION RATIO, AND TD ERROR

Policy	Reward				Low Conf 50%		Low Conf 10%	
	Mean	Min	Max	CV	Ratio (%)	TD Error	Ratio (%)	TD Error
softmax_0.1	31318.40	7000.00	60800.00	0.34	0.436	2.323	0.132	1.797
softmax_0.5	19289.60	8000.00	33000.00	0.23	2.507	-0.629	0.354	-2.660
softmax_1.0	13739.80	5800.00	24500.00	0.24	2.784	-7.744	0.537	-5.201
softmax_2.0	8113.20	2600.00	14900.00	0.31	6.274	-6.111	2.491	-10.671
softmax_10.0	966.20	0.00	2900.00	0.56	0.000	0.000	0.000	0.000
deterministic	22802.80	3400.00	51600.00	0.54	0.000	0.000	0.000	0.000
linear	2835.40	500.00	7100.00	0.41	11.982	-9.045	0.000	0.000
softmax_drop_1	13951.80	3800.00	22700.00	0.23	2.778	-7.110	0.546	-4.115
softmax_drop_2	14117.20	5200.00	26200.00	0.23	2.781	-7.109	0.543	-3.958
linear_drop_1	4168.20	700.00	12500.00	0.38	4.888	-8.184	0.770	-9.098
linear_drop_2	4468.60	900.00	18400.00	0.39	3.271	-7.312	0.627	-8.463

TABLE X. POLICY-WISE SUMMARY FOR PONG: REWARD, LOW CONFIDENCE ACTION RATIO, AND TD ERROR

Policy	Reward				Low Conf 50%		Low Conf 10%	
	Mean	Min	Max	CV	Ratio (%)	TD Error	Ratio (%)	TD Error
softmax_0.1	20.99	20.00	21.00	0.01	1.306	0.000	0.260	0.000
softmax_0.5	20.95	20.00	21.00	0.01	5.144	-0.000	0.609	-0.001
softmax_1.0	20.84	14.00	21.00	0.03	6.116	-0.001	0.489	-0.009
softmax_2.0	14.61	-19.00	21.00	0.38	6.004	-0.020	0.388	-0.143
softmax_10.0	-19.08	-21.00	-12.00	-0.08	0.920	-0.058	0.002	-0.071
deterministic	21.00	21.00	21.00	0.00	0.000	0.000	0.000	0.000
linear	-14.93	-21.00	-3.00	-0.22	4.203	-0.032	0.145	-0.041
softmax_drop_1	20.81	13.00	21.00	0.03	6.956	-0.001	3.134	-0.001
softmax_drop_2	20.82	14.00	21.00	0.03	9.339	-0.001	7.152	-0.001
linear_drop_1	-9.83	-20.00	15.00	-0.56	2.434	-0.040	0.084	-0.044
linear_drop_2	2.80	-19.00	21.00	2.90	1.639	-0.042	0.070	-0.059

TABLE XI. POLICY-WISE SUMMARY FOR QBERT: REWARD, LOW CONFIDENCE ACTION RATIO, AND TD ERROR

Policy	Reward				Low Conf 50%		Low Conf 10%	
	Mean	Min	Max	CV	Ratio (%)	TD Error	Ratio (%)	TD Error
softmax_0.1	3777.80	800.00	4175.00	0.24	0.800	2.531	0.171	-0.199
softmax_0.5	3439.85	575.00	4200.00	0.37	3.279	1.720	0.486	-0.235
softmax_1.0	2404.60	125.00	4450.00	0.68	4.653	-0.332	0.733	0.577
softmax_2.0	844.15	150.00	4200.00	0.81	7.433	-0.515	0.730	-0.257
softmax_10.0	312.35	0.00	1200.00	0.64	5.489	-2.509	0.000	0.000
deterministic	3894.00	800.00	4050.00	0.18	0.000	0.000	0.000	0.000
linear	371.55	25.00	1325.00	0.56	2.374	-0.304	0.083	-0.803
softmax_drop_1	2370.35	225.00	4600.00	0.69	4.625	-0.186	1.045	0.132
softmax_drop_2	2457.75	325.00	4200.00	0.67	4.711	-0.030	2.299	-0.477
linear_drop_1	481.15	0.00	4050.00	0.73	0.954	-0.911	0.020	-1.776
linear_drop_2	1201.20	125.00	4150.00	1.14	1.296	-1.239	0.028	0.054

sampled transitions, and the significant, negative TD errors are the direct result. The critic, having not been adequately trained on these inputs, provides overly optimistic value estimates and is then surprised by the poor outcomes. This is a classic symptom of statistical overfitting, where a model fails to generalize from its training distribution to a different test distribution. The success of deterministic evaluation is therefore not just a matter of exploitation; it is a strategy that implicitly hides this overfitting by constraining the agent to the well-modeled, high-density regions where its value function is accurate.

V. DISCUSSION

Our empirical analyses provide multifaceted insights into the consequences of employing stochastic evaluation policies for agents trained with PPO. The findings have significant practical implications for RL practitioners and open up new avenues for future research.

A. Synthesis of Findings: A Unified View

The narrative that emerges from our results is clear and consistent. The central finding of this study is that the value functions of PPO-trained agents are often overfitted to the high-confidence, on-policy actions experienced during training. This overfitting, compounded by the objective mismatch between a stochastic training goal and a deterministic evaluation goal, is revealed through the following mechanism:

1) Stochastic evaluation policies act as a diagnostic probe: By increasing policy entropy, we force the agent to sample low-confidence actions more frequently. This pushes the agent

TABLE XII. AVERAGE ENTROPY PER POLICY AND ENVIRONMENT

Policy	Alien	Boxing	Breakout	Enduro	Freeway	KFM	Pong	Qbert
softmax_0.1	0.216	0.057	0.183	0.195	0.000	0.206	0.198	0.112
softmax_0.5	0.998	0.289	0.482	0.820	0.000	1.058	0.803	0.544
softmax_1.0	1.577	0.575	0.727	1.234	0.036	1.529	1.132	0.871
softmax_2.0	2.144	1.138	0.987	1.592	0.335	1.958	1.396	1.193
softmax_10.0	2.820	2.761	1.329	2.137	1.042	2.585	1.756	1.692
deterministic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
linear	2.677	2.699	1.015	1.905	0.514	2.367	1.416	1.501
softmax_drop_1	1.575	0.577	0.676	1.224	0.032	1.529	1.073	0.857
softmax_drop_2	1.572	0.575	0.495	1.209	0.000	1.527	0.974	0.796
linear_drop_1	2.593	2.575	0.565	1.806	0.000	2.163	1.202	1.279
linear_drop_2	2.517	2.475	0.000	1.704	0.000	2.112	0.895	0.888

to rely on its value function in low-density regions of the state-action space that were infrequently visited during training.

2) The probe reveals inaccurate and overly optimistic value estimates: The value function, having been poorly generalized in these regions, proves to be systematically over-optimistic. This inaccuracy is directly measured by the large, negative TD errors that result when these low-confidence actions are taken.

3) Inaccurate value estimates lead to performance degradation: This failure of the value function to accurately price the consequences of suboptimal actions is the direct cause of the observed performance degradation. The success of deterministic evaluation is therefore a strategy that succeeds primarily because it confines the agent to the well-modeled, high-density regions of its policy distribution where the value function is reliable. Conversely, the superior performance of low-to-moderate entropy policies in five of the eight environments (Alien, Breakout, Enduro, KFM, and Freeway) indicates that deterministic evaluation is not universally optimal. In these cases, the agent's performance is limited by factors other than value function overfitting. We hypothesize that modest stochasticity helps overcome challenges like perceptual aliasing (where different states appear identical) or helps the policy escape repetitive, suboptimal loops that a purely deterministic approach might get trapped in. This highlights a fundamental trade-off: while deterministic evaluation protects against the dangers of a poorly generalized value function, it may prevent the agent from achieving a higher score if the environment dynamics reward a degree of randomness.

B. Practical Implications for Reinforcement Learning Practitioners

Based on our findings, we propose the following guidelines for practitioners working with policy gradient algorithms:

1) Default to low-entropy evaluation (tune temperature): Deterministic (argmax) is a strong baseline, but not uniformly optimal across our Atari suite. Practitioners should treat evaluation entropy as a hyperparameter: begin with argmax or a low-temperature softmax, and select the setting that maximizes mean return and stability on held-out evaluation seeds.

2) Consider slightly stochastic evaluation in specific cases: In environments where there is a strong reason to suspect the presence of perceptual aliasing or where the policy might be trapped in a local optimum, evaluating with a small amount of stochasticity can be beneficial. As observed in Breakout, a low-temperature softmax policy (e.g., with temperature T=0.5) might allow the agent to escape repetitive patterns and achieve higher performance. In environments such as Breakout, we hypothesize that a purely deterministic policy may become trapped in a simple but less effective strategy. The evaluation temperature should be treated as a hyperparameter to be tuned based on the specific problem at hand.

3) Use TD error on low-confidence actions as a diagnostic tool: Our results highlight the diagnostic power of TD error. Monitoring the TD error on actions with low selection probability during the training process can serve as a powerful indicator of value function overfitting. A widening gap between the TD error on high-confidence versus low-confidence actions should be seen as a red flag, signaling a potential generalization problem.

C. Limitations and Future Work

While this study provides a comprehensive analysis, it has several limitations that point toward important directions for future research.

1) Algorithmic scope: Our investigation was confined to the PPO algorithm. It is an open question whether these findings generalize to other families of RL algorithms. Of particular interest are off-policy actor-critic methods, such as SAC, which explicitly optimize an entropy-regularized objective. Because an agent trained with SAC is optimized to act stochastically, it is plausible that such an agent would exhibit less performance degradation, or perhaps even an improvement, under stochastic evaluation. A comparative study is needed to resolve this.

2) Environment scope: The analysis was conducted on eight discrete-action Atari games. Future work should extend this investigation to other domains, such as continuous control tasks (e.g., MuJoCo benchmarks) and more complex, modern game environments, to assess the generality of our conclusions.

3) Exploring regularization: Our findings indicate that value function overfitting is a key limiting factor. This motivates future work to test empirically whether regularization techniques known to combat overfitting in supervised learning can enhance the value function's generalization in an RL context. Researchers and practitioners may apply methods such as weight decay, dropout, spectral normalization, or ensemble methods to the critic network during PPO training. An effective

regularization scheme should reduce the TD error on low-confidence actions and, as a result, diminish the performance gap between deterministic and stochastic evaluation.

VI. Conclusions

This paper has presented a rigorous empirical investigation into the common practice of training reinforcement learning agents with a stochastic policy but evaluating them deterministically. Our findings demonstrate that this practice is not merely a convention but is often a well-justified approach that mitigates a critical, underlying weakness in deep RL agents: the overfitting of the learned value function to its most frequently observed state-action pairs.

We have shown that higher evaluation-time entropy often degrades performance and stability—particularly in brittle environments—yet modest stochasticity can improve returns in several games (e.g., Alien, Breakout, Enduro, KFM, Freeway). Thus, evaluation entropy should be selected empirically, balancing exploitation against environment-specific benefits of limited randomness. The root cause of performance degradation, in cases where it occurs, is that stochastic policies force the agent to sample low-confidence actions, exposing it to low-density regions of the state-action space where its value function is demonstrably inaccurate and overly optimistic. This inaccuracy is a direct manifestation of overfitting to the on-policy training distribution, compounded by the inherent mismatch between the entropy-regularized stochastic objective used in training and the purely exploitative deterministic objective used in evaluation.

Ultimately, the choice of an evaluation policy transcends the simple dichotomy of exploration versus exploitation. It is a decision that interacts deeply with the generalization limits of the agent's learned models. Deterministic evaluation often succeeds because it acts as a safeguard, confining the agent to the high-density regions of its policy distribution where its internal value model is most reliable. However, our work also shows this is not a universal solution, as some environments reward modest stochasticity that can overcome other limitations of the learned policy. This work provides a clear empirical framework and a causal, TD-error-based mechanism for understanding this critical aspect of deep reinforcement learning applications, offering both practical guidance for practitioners and a diagnostic tool for future research aimed at building more robust and generalizable agents.

Acknowledgment

This research was supported by the 2024 KNUDP(Korea National University Development Project) funded by the Ministry of Education(MOE, Korea) and National Research Foundation of Korea(NRF)

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-Level Control through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] A. Karalakou, D. Troullinos, G. Chalkiadakis, and M. Papageorgiou, "Deep Reinforcement Learning Reward Function Design for Autonomous Driving in Lane-Free Traffic," *Systems*, vol. 11, no. 3, 2023.

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[4] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[5] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare, "Deep reinforcement learning at the edge of the statistical precipice," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 29 304–29 320.

[6] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *International conference on machine learning (ICML)*, 2017, pp. 1352–1361.

[7] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning (ICML)*, 2018, pp. 1861–1870.

[8] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[9] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International Conference on Machine Learning (ICML)*, 2015, pp. 1889–1897.

[10] N. Milosevic, J. Müller, and N. Scherf, "Central Path Proximal Policy Optimization," in *The Exploration in AI Today Workshop at International conference on machine learning (ICML)*, 2025.

[11] C. B. Tan, E. Toledo, B. Ellis, J. N. Foerster, and F. Huszár, "Beyond the Boundaries of Proximal Policy Optimization," *arXiv preprint arXiv:2411.00666*, 2024.

[12] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2016, pp. 1928–1937.

[13] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International Conference on Machine Learning (ICML)*, 2014, pp. 387–395.

[14] M. Schilling, "Avoid Overfitting in Deep Reinforcement Learning: Increasing Robustness Through Decentralized Control," in *International Conference on Artificial Neural Networks (ICANN)*, 2021, pp. 638–649.

[15] E. Korkmaz, "A Survey Analyzing Generalization in Deep Reinforcement Learning," *arXiv preprint arXiv:2401.02349*, 2024.

[16] C. Zhang, O. Vinyals, R. Munos, and S. Bengio, "A Study on Overfitting in Deep Reinforcement Learning," *arXiv preprint arXiv:1804.06893*, 2018.

[17] T. Hongyao and B. Glen, "Improving Deep Reinforcement Learning by Reducing the Chain Effect of Value and Policy Churn," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[18] Q. Li, A. Kumar, I. Kostrikov, and S. Levine, "Efficient Deep Reinforcement Learning Requires Regulating Overfitting," in *International Conference on Learning Representations (ICLR)*, 2023.

[19] T. Zahavy, M. Haroush, N. Merlis, D. J. Mankowitz, and S. Mannor, "Learn what not to learn: Action elimination with deep reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[20] K. Xue, X. Lin, Y. Shi, S. Kai, S. Xu, and C. Qian, "Escaping Local Optima in Global Placement," *arXiv preprint arXiv:2402.18311*, 2024.

[21] H.-R. Pan and B. Schölkopf, "Skill or Luck? Return Decomposition via Advantage Functions," in *International Conference on Learning Representations (ICLR)*, 2024.