Extended Abstract

Motivation In this project, we aim to improve exploration in PPO. Exploration-exploitation trade-off is crucial in RL algorithms. Exploration helps in finding pathways to the solutions. Whereas, exploitation helps the agent to move towards the goal faster. PPO provides entropy bonus as a mechanism to explore the environment. While this helps the agent to get out of the local minima, it does not learn diverse solutions. We take a closer look at the entropy and propose a solution to obtain diverse paths to goal.

Method We first analyze the entropy bonus of the PPO and study its effect on the exploration as it is weighed. We find that both low and high entropies weights slow down PPO's solution search. While the low entropy could not search due to getting stuck in local minima, high entropy incentivizes PPO to explore a lot before finding a solution. We then study if running PPO longer after finding a solution helps it in finding newer solutions. While PPO does find new paths to goal, they are only a perturbation of the initial path that it finds. Upon a closer look at the entropy distribution, we find that this is due to collapse of the entropy; majority of the states are on low entropy and maximum entropy is very less. On the contrary, there should be some states with high entropy to encourage exploration. Therefore, we introduce an inductive bias of exponential distribution over the entropies. According to our objective, the entropy distribution over the states should be of the form of exponential distribution and ensure that some states get high entropy. We use moment matching loss between empirical entropy distribution and exponential distribution.

Implementation We perform our experiments on an obstacle course game. The game environment contains a spawn block and goal block. The agent has to avoid the obstacle (lava) in the environment to get to the goal block. Obstacle placement incentivizes multiple paths to the solution. Reward is 10 for reaching the goal and minor positive reward inversly proportional to distance goal. Observation space is agent's position, velocity, agent's distance from obstacles, and flags for agent's interaction with the environment. Both policy and value networks have a shared backbone with 2 linear layers. policy and value heads have a single linear layers.

Results Our experiments show that with the prior of exponential distribution on the entropies, the diversity of the solutions is qualitatively improved. It could find two starkly different solutions, compared to a multiple solutions perturbed around one mode, as seen in case of the entropy bonus. We compare the diversity produced by the entropy bonus vs proposed moment matching with exponential distribution, and show that our proposed loss performs 194% better relative to the baseline on the diversity as computed using pairwise DTW distance.

Discussion The results align with the expectation of entropy distribution. As we increase the number of moments used, we get more diverse solutions. The randomness induced with exponential distribution is different compared to entropy bonus. While entropy bonus shows completely random paths, our proposed loss explores multiple directed paths. While the higher entropy values control randomness, the lower values give direction to the paths. Thus achieving control over exploitation (directness of path) vs exploration (randomly choosing action). This route of exploration-exploitation balance is more controlled because the algorithm designer can replace the distribution to achieve desired trade-off.

Conclusion We investigated into exploratory properties of entropy bonus in PPO. We found that entropy bonus is helpful in getting out of local minima, but could not find diverse solutions. Our observation was the policy decreases the entropy close to 0 irrespective entropy bonus. This hinders the search for diverse solutions, which requires the policy to place high action entropy over a few number of states. Therefore, we propose the entropy distribution closer to an exponential distribution, thus ensuring representation for high-entropy states. Our results show the diverse trajectories found by this entropy objective. As this proposed approach hints towards a exploration-exploitation control based on distribution, the choice of distribution remains a key design decision requiring domain knowledge and understanding of the desired entropy distribution against the baseline, as shown in our environment.

Towards Exponential Exploration

Tejan Karmali
Department of Computer Science
Stanford University
tejan@stanford.edu

Abstract

Proximal Policy Optimization (PPO) employs entropy regularization to balance exploration and exploitation in reinforcement learning environments. However, we identify a fundamental limitation: while entropy bonuses help agents escape local minima, they fail to discover diverse solution pathways due to entropy collapse, where the majority of states exhibit uniformly low entropy values. Through empirical analysis on navigation tasks with multiple viable solution paths, we demonstrate that PPO's entropy distribution becomes highly concentrated near zero, limiting the agent's ability to maintain exploratory behavior in regions where diverse strategies could emerge. To address this limitation, we propose an entropy regularization scheme that enforces an exponential prior distribution over state-wise entropy values, ensuring adequate representation of high-entropy states throughout training. Our approach maintains the benefits of entropy regularization for escaping local optima while promoting sustained exploration in regions conducive to discovering alternative solution strategies. Experimental results on obstacle avoidance navigation tasks show that our method discovers qualitatively distinct solution trajectories, contrasting with standard PPO which finds only minor perturbations around a single dominant strategy. This work provides new insights into the role of entropy distribution in policy optimization and offers a principled approach to enhancing solution diversity in reinforcement learning.

1 Introduction

Balancing the exploration and exploitation trade-off is crucial in the success of reinforcement learning algorithms. Reinforcement learning algorithms has the objective to maximize the expected sum of rewards in an episode. While the original policy gradients approach has this vanilla objective, the later algorithms proposed multiple improvement over it to reduce the variance in this estimate. These solutions did not take into account the case of when an agent gets trapped in a local minima. Trying different actions and exploring becomes the more relevant to alleviate such an issue. Another aspect of exploration is to find diverse solution strategies to a problem, thus building a set of solutions.

Proximal Policy Optimization Schulman et al. (2017) (PPO) proposes a refined form of policy gradients objective which is stable and fast to train. It is able to reuse the rollout data collected using a recent version of the policy. This objective includes a term to reinforce most rewarding actions, an entropy bonus to encourage exploration of other actions in the action space, and a KL-divergence term to not to prevent deviation from the rollout policy. While PPO achieves the objective of the finding a rewarding path to the goal, it is not able to find multiple paths to the goal.

Learning based algorithms, like PPO, prioritize exploitation over exploration. That is, since the objective of the algorithm is to find the expected case of most rewarding solutions, it is biased to safely explore. On the contrary, a naive algorithm that purely explores would be random walk where the agent randomly takes actions at each state, until it reaches the goal state. GoExplore is an intelligent version of random walk algorithm where they follow a dynamic programming approach to reduce the

costs associated with repeating simulations at an already seen state and uses heuristics to choose a state to explore from. But a concern with GoExplore is that it assumes access to the state space of the environment. This implies that we can spawn the agent from any state in the environment rather than a fixed spawn state. This assumption helps it to reduce the costs associated with random walk exploration.

To induce more control over exploration in PPO, we take inspiration from GoExplore that some states are more worthy to explore from than the others. Our analysis of PPO's entrotpy bonus suggested that entropy collapses to values near 0, inspite of entropy bonus. This means the PPO agent's strategy is heavily biased to exploitation. To counter that, we ensure that in all the states that a given policy can visit some of them should have high entropy. Thus, creating a room for exploration from some states. Thus we also define entropy to be a heuristic to decide which states are worthy of exploration.

To enforce this, we propose matching entropy distribution with exponential distribution. We find that it alleviates the collapse of entropies to 0, thereby also having states with maximum possible entropy to encourage exploration. This further encourages the discovery of diverse paths to the goal. Another key property enables by this objective is a different way to look at the exploration-exploitation trade-off. Algorithm designer can control the desired distribution to be matched with the entropy's distribution. Thus if the desired distribution has more probability mass on high entropies, that would imply enforcing a more explorative policy, and vice-versa.

Overall, our key contributions are as follows:

1. We first analyze the existing entropy bonus to check how it affects the diversity of the solutions produced.
2. Based on the findings, we propose loss to match entropy distribution to a prior distribution, which in our case in exponential distribution.
3. We thoroughly analyze the baseline and our proposed loss quantitatively and qualitatively, showing the diverse solutions obtained by our approach.

2 Related Work

In model-free reinforcement learning, exploration-exploitation balancing is important because of lack of a world model. PPO encourages exploration using entropy bonus. The usage of maximum entropy in reinforcement learning was proposed in Ziebart et al. (2008); Fox et al. (2016); Haarnoja et al. (2017); Rawlik et al. (2012); Toussaint (2009). Maximum entropy objective modifies reinforcement learning (RL) objective to incorporate exploration along with reward maximization. As a result, this helps to escape the agent if it follows a deceptive path of rewards. Thus it helps construction of policies that are robust to such reward functions.

On another end, search-based algorithms such as GoExplore Ecoffet et al. (2019) rely more on the exploration to find at least one solution to the task. While exploring it forms a model of the world, which is used to teleport the agent to a promising state to explore from. GoExplore introduces heuristics to decide what are the promising states to explore from, which are based on the number of times a state is visited, or how recently a state statistic has been updated.

If we want to explore effectively in PPO to find diverse solutions, we would want to decide which states are promising to explore from. Since entropy has shown that trying a different route in PPO which eventually leads to success, we hypothesize that entropy could be repurposed as a heuristic to decide the promising states in PPO to explore from.

3 Method

3.1 Background

Proximal Policy Optimization (PPO): PPO introduces a clipped surrogate objective as described in Eq. 1. The algorithm operates by first collecting data using rollout policy $\pi_{\theta_{old}}$, which is the most recent checkpoint of the policy. It then updates the policy π_θ with PPO objective as we describe next, while keeping in mind to not to exceed a deviation margin from the rollout policy. Finally, after performing enough number of updates, $\pi_{\theta_{old}}$ is updated to be π_θ, and the training repeats.

The PPO objective as following main properties: a) the ratio r_t (eq. 2) measures the relative sharpening of the probability associated with an action. b) Clipping: One aspect of PPO is that it prevents the policy π_θ from diverging from the rollout policy $\pi_{\theta_{old}}$. ϵ control the divergence of the ratio of action probabilities. c) Entropy: Entropy maximization objective incentivizes trying out of different actions, balanced by weight w_1. d) KL Divergence: prevents divergence of the current action distributions $\pi_\theta(\cdot|s_t)$ and action distribution at the rollout $\pi_{\theta_{old}}(\cdot|s_t)$. Finally, PPO objective is to maximize the combined objective as described in eq. 5.

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t,\ \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right] \tag{1}$$

where

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{2}$$

$$H(\theta) = -\mathbb{E}_{a_t}\left[\log \pi_\theta(a_t|s_t)\right] \tag{3}$$

$$\text{KL}(\theta) = \mathbb{E}_t\left[\text{KL}\left(\pi_{\theta_{\text{old}}}(\cdot|s_t) \;||\; \pi_\theta(\cdot|s_t)\right)\right] \tag{4}$$

$$\mathcal{L}_{\text{PPO}}(\theta, \gamma) = L^{\text{CLIP}}(\theta) + w_1 H(\theta) - w_2 \text{KL}(\theta) - w_3 \mathcal{L}^{\text{VF}}(\gamma) \tag{5}$$

In the PPO framework, we first study the behavior of entropy bonus to understand how it helps in escaping local minima to eventually find a solution (Sec. 3.2). Then we study the distribution of the entropies and how it prevents the algorithm from exploring further to find newer solutions (Sec. **??**). Finally, we propose exponential exploration objective, an alternative to entropy bonus, which allows for continual exploration and finding new solutions (Sec. zz).

3.2 Escaping local minima

Problem of local minima: PPO objective, without entropy bonus, optimizes only to sharpen action distribution. This objective is more biased towards exploitation as it ensures that the policy does not take any sub-optimal action. As we observe in the Fig. 1, the policy suggested various actions that progressed the agent towards the goal. In their initial stages of the training, the policy starts with random action distribution. Upon encountering an obstacle where it got stuck (right side, EntCoeff 0 in Fig. 1), it was able to escape it because of higher entropy in the initial stages.

After escaping the minima, policy was eventually able to discover a different path (straight path) after some updates. But again got stuck at a different obstacle. As entropy continues to decrease, by this update the policy could insert any random action which could have helped in escaping this local minima.

Escaping with entropy bonus: As can be seen in Fig. 1, as the entropy coefficient (EntCoeff) is increased the policy is more capable of escaping local minima. For eg. at EntCoeff=5×10^{-4}, the agent got stuck at multiple local minima during its course of training (multiple darker colored paths), but eventually found solutions to the goal. Further increasing the EntCoeff to 1×10^{-3}, the policy got stuck at only one local minima before reaching the goal. Finally, at very high EntCoeff such at 8×10^{-3}, the agent spends more time in taking random actions rather than progressing towards the goal. We also see this trend in the Fig. 2.

Here we see two extrema of entropy bonus. On one hand, if weighed too low it can cannot escape the minima, or can get stuck at multiple local minima. At high weightage, it never progresses to the solution. This leads to an interesting observation that we would want high entropy as a rescue whenever the agent gets stuck, but not necessarily at every state. Which is contrary to the entropy bonus objective which tries to maximize entropy at every state.

3.3 Evaluation of Entropy Distribution

To further evaluate the entropy, we seek to answer the question if training longer with entropy bonus makes the policy explore a diverse set of solutions. We observe that once PPO finds a solution, it continues to perturb it if we further continue to train it. The degree of perturbation varies wrt

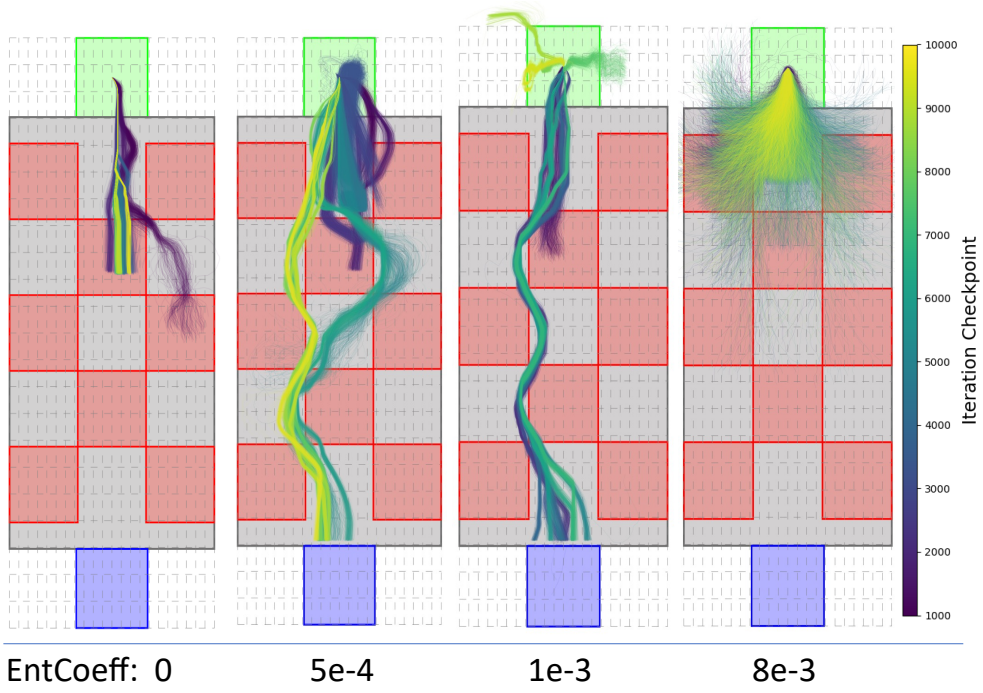


Figure 1: **Policy behavior at different Entropy coefficient values:** Green block is where the agent spawns. Blue block is the goal block. **red**Red are the lava obstacles that the agent has to avoid. We show here paths taken by the agent during the course of training. darker colored paths are taken earlier in the training, and vice-versa. At lower entropies coefficients (EntCoeff) the agent gets stuck while focussing on exploitation (EntCoeff 0). At higher EntCoeff the agent explores widely instead progressing towards the goal. The sweet spots lie in the middle which balances exploration and exploitation.

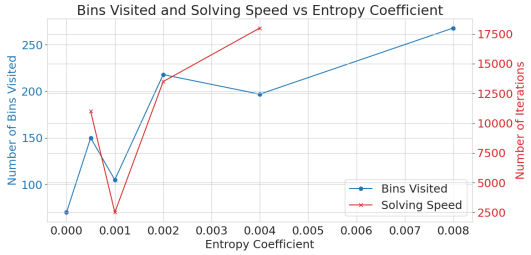
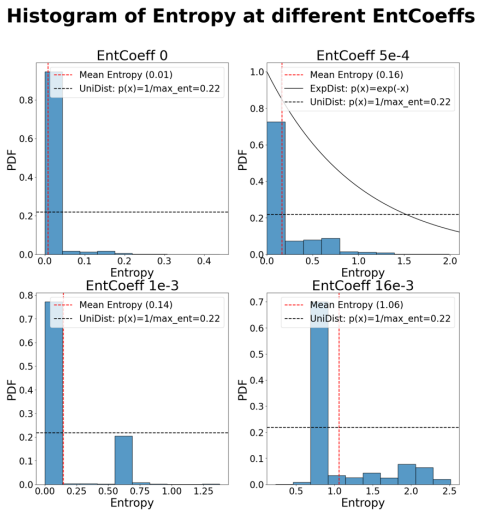


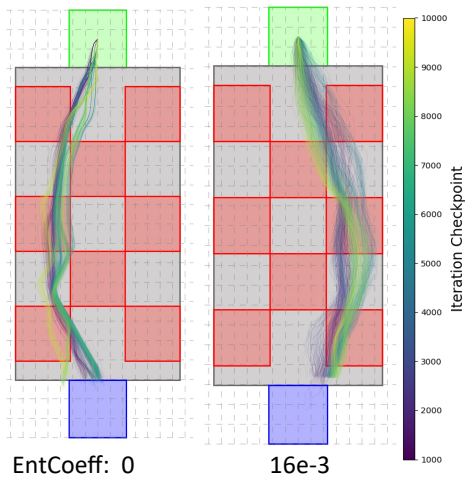
Figure 2: Number of bins explored is a proxy for exploration. Low and high entropy coefficients lead to higher exploration before finding a solution. But these explorations contribute to escaping from local minima.

the entropy coefficient. For eg. solutions at EntCoeff=0 are perturbed lesser as compared to the EntCoeff=16×10^{-3}.

This result can be correlated with the entropy distribution. Here, we compute entropy at each state along the paths taken by the agent with final trained policy. The entropies are collapsed near 0. As the entropy coefficient is increased, we make two observations: the mean of entropy distribution shifts and b) larger values of entropy start showing some representation. However, these values are not close to the maximum entropy possible in this setup. which is 4.56. If some states would have higher entropies, it could meant that the algorithm still has the scope to explore and find diverse solutions.



(a) Entropy distribution at different entropy coefficients (EntCoeff).



(b) Perturbations of solution paths upon longer training.

Figure 3: (a) Upon training PPO for long, the entropy distrubution collapses. Despite of EntCoeff. (b) This is evident in the agent's paths which different but essentially perturbed versions of each other

3.4 Exponential Entropy Distribution

As we establish the need for states with higher entropy values in order for PPO to produce diverse solutions, we seek to introduce a prior in the enrtropy distribution. Observing the waning of the counts for higher entropy values, we seek to fit an exponential distribution to the entropies. This helps in controlling the counts of low-entropy states and higher entropy states in the entropy distribution, unlike entropy bonus which does not offer any control over it.

Our proposed entropy objective makes use of moment matching Li et al. (2015) to match the entropy distribution to the exponential one. We match first k moments of a suitably chosen exponential distribution $p(x) = \lambda exp(-\lambda x)$.

Let m_i be the ith moment of $p(x)$, calculated as $m_i = \mathbb{E}_{\curvearrowright \sim l(\curvearrowright)}[x^i], i \in \mathbb{Z}^+$. These are defined in a closed-form for exponential distribution. We estimate the respective moments of entropies of states seen during a rollout as follows. Suppose a rollout R_i sees states s_t^i using a policy $\pi_{\theta_{old}}$ for timesteps $t \in \{1, ..., T\}$. Entropy at s_t^i is H_t^i as defined in Eq. 3. Empirically, the entropies form the set $\mathbf{H} = {H_t}^{i=k,t=T}_{i=0,t=1}$. Estimated entropy moments are defined at $\hat{m}_j = \dfrac{\sum_{i,t}(H_t^i)^j}{kT}$. The final moment matching loss looks as follows:

$$\mathcal{L}(\theta) = \sum_{j=1}^{l}(m_j - \hat{m_j})^2 \tag{6}$$

4 Experimental Setup

We run our experiments on MadronaEngine Shacklett et al. (2023) setup with PPO. Our environment is built off obstacle course games inspired from Roblox obbies. The environments have a spawn block where the agent spawns and a goal block which it has to reach.

Action space of the agent is composed of 4 discrete distributions: 1) Move/not move 2) 8 orientations (uniformly spaced between 0-360) towards the step relative to the agent 3) turn (no turn, left, or right) 4) jump/no jump. For eg. is action is $[1, 2, 2, 0]$, it means that the agent should move, by first turning to right, then changing the orientation in counter-clockwise by 90 degrees, and not jump.

Observation Space: Observation space contains vectors corresponding to depth, velocity, agent's observation, and flags indicating interaction between agent and environment's entities such as obstacles. **Network:** Actor-Critic of PPO is implemented as a network with shared backbone

to extract features from observation and then passed through policy head (actor) and value head (critic). For the moment matching loss, we experiment with first 4 moments with distribution prior as $p(x) = \lambda \exp(-\lambda x), \lambda = 1$

5 Results

Metric: We compare entropy bonus and the proposed exponential moment matching method on mean Dynamic Time Warping (DTW) distance metric between successful trajectories. We collect 25 successful trajectories across every 1000 iterations during the training (total 10 checkpoints). Then we compute DTW distance between each pair of trajectories at every iteration and report its mean and standard deviation.

5.1 Quantitative Evaluation

Table 1: DTW Distance Analysis: Entropy bonus and Exponential Moment matching Comparison on successful trajectories

(a) Entropy Coefficient vs DTW Distance

Entropy Coefficient	**Mean DTW Distance**
0.0000	$30.335_{\pm 6.299}$
0.0005	$36.795_{\pm 12.496}$
0.0010	$23.451_{\pm 7.337}$
0.0020	$24.823_{\pm 5.435}$
0.0040	$31.970_{\pm 6.193}$
0.0080	$36.309_{\pm 6.788}$
0.0160	$\mathbf{55.253_{\pm 11.579}}$

(b) Moment matching comparison

Moments	**Mean DTW Distance**
M_1	$50.684_{\pm 5.056}$
M_{1-2}	$40.791_{\pm 4.707}$
M_{1-3}	$42.059_{\pm 18.874}$
M_{1-4} variants	
M_{1-4} (weight 0.25)	$104.366_{\pm 57.755}$
M_{1-4} (weight 0.5)	$102.450_{\pm 77.806}$
M_{1-4} (weight 1.0)	$\mathbf{162.567_{\pm 52.998}}$

As reported in Tab. 1, mean DTW distances over the course of training in entropy bonus vs. moment matching objectives. DTW distance measures distance between two signals that are shifted in time. Higher the DTW distance, better is the diversity. We compute DTW distance by treating agent's positions as a 1D signal. We find that the entropy bonus could achieve mean DTW distance till 55.253, with an increase in the distance as the EntCoeff is increased. However also note that at very high EntCoeff, the convergence to the solution is slower. Compared to entropy bonus, moment matching objective achieves superior diversity as measured by the DTW distance, achieving a mean distance of 162.567, which 194% relative incraese against the best entropy bonus for diversity.

Here we also ablate on the different moments. Firstly, we observe in Tab. 1b that first 3 moments have similar performance as the entropy bonus on the DTW metric. This can also be verified qualitatively with Fig. 4. We also see, in Fig. 4, that higher entropies count increases as more moments are added.

In Tab. 2 we measure all the moment losses as evaluation metric as we add on new moments in the objective, starting with M_1. This is done to qualitatively measure the closeness to the exponential distributional prior that we wanted to achieve. We find a steady decrease in losses as new moments are added, with minimum loss when all 4 moments are present.

Moments used	M_1 **loss**	M_2 **loss**	M_3 **loss**	M_4 **loss**
M_1	0.0929	2.1551	30.6824	553.6380
M_{1-2}	0.3083	1.8003	20.2531	407.2031
M_{1-3}	0.4828	1.8186	16.7182	304.6855
M_{1-4}	**0.0793**	**1.1228**	**11.6334**	**205.9728**

Table 2: Ablation of moments used for moment matching objective

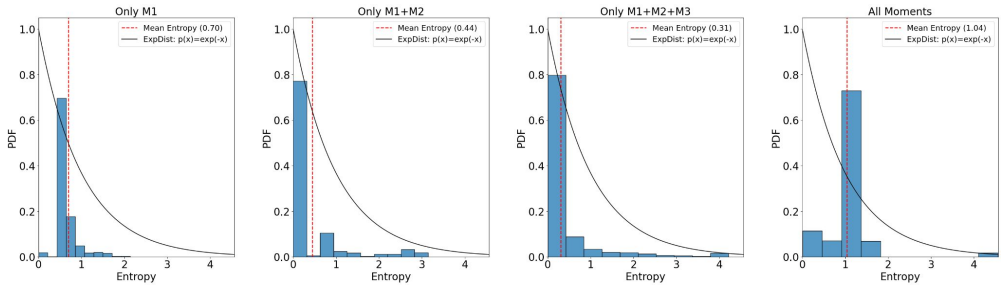


Figure 4: Learnt entropy distributions with different moment matching losses.

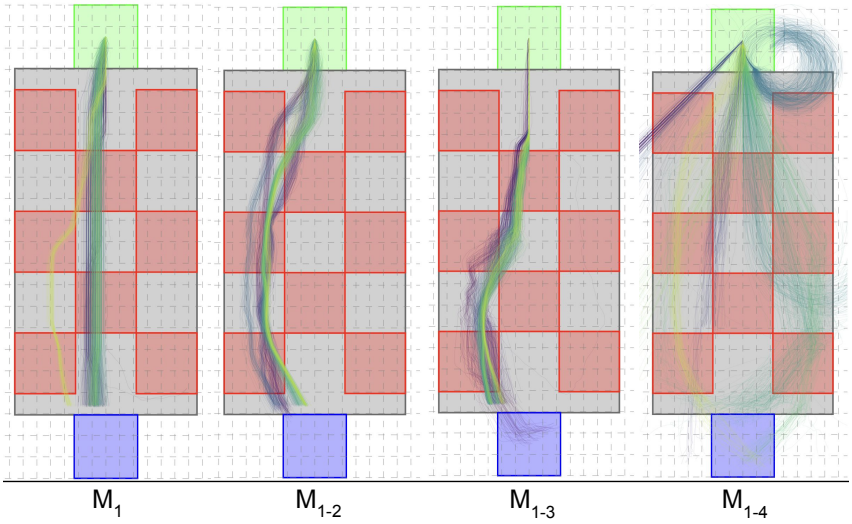


Figure 5: Visualization of agent trajectories at different moment matching objectives.

5.2 Qualitative Analysis

We report the qualitative results in Fig. 4. We see that as we include more moments in the loss objective, the distribution gets closer to the exponential distribution. This is reflected qualitatively in the Fig. 4. Using all 4 moments shifted the **Insight** Using lesser moments keeps the trajectories together whereas using more moments lets the trajectories go waywards. However, this is a different pattern of randomness than the one observed at higher EntCoeff in Fig. 1. Here the agent takes multiple well-defined paths but some do not make to the goal. All the paths are diverse from each other. These paths are well-defined due to expoenential distribution favouring low entropies while still having representation for the higher entropies.

6 Discussion

As reported so far, our proposed loss achieves more diverse solutions when combined with PPO as compared against vanilla entropy bonus. Using a distributional prior, such as exponential in our case, gives a control in algorithm designer's hands where they can control exploration and exploitation at a finer level by playing with the desired distribution. It is finer because in case of the entropy bonus, the control is using only the entropy coefficient, whereas in case of entropy distribution matching we can control the probability mass over low/high entropies. More mass on lower entropies would mean an exploitative policy and vice versa. The choice of the prior distribution also depends on domain and the phase of training. We observed that enforcing our loss at initial stages of training does not allow it to explore, and therefore we apply our loss only if mean entropy of a rollout is below 66.66%-ile of maximum entropy.

7 Conclusion

In this project, our goal was to diversify the successful paths explored by PPO during its training. We started with an analysis of existing PPO obejctive, specifically that of the entropy bonus which encourages the exploration. We found that exploration objective helps in getting out of local minima while the agent searches its way to the goal but it does not find diverse paths to the goal. We hypothesized that a cause of this could be entropy collapse, which gets near-zero entropy for all the paths to the goal. To fix this, we proposed to fit the entropy distribution to exponential distribution using moment matching, so that some states get higher entropy values to facilitate exploration. We found that this results in exploration of diverse trajactories compared to the baseline of entropy bonus. We analyzed the effect of moments used to construct the loss, and found that at sufficiently higher moments the method gives desired results. Introduccing distributional prior for entropy distribution open new avenues to control exploration-exploitation trade-off to the algorithm deisgner by with a choice of prior.

Changes from Proposal Original Hypothesis: Initially we planned to characterize the exploration strategy followed by PPO and SAC. Then, we aimed to transfer the exploration strategies between them.
Revised Hypothesis: We retain the spirit of the hypothesis to characterize the exploration, but limit ourselves to analyzing PPO's exploration in depth. after characterizing we experimentd with new loss to control entropy.

Acknowledgements We would like to acknowledge Zander Majercik, William Wang, Sharon Zhang, Vishnu Sarukkai, Brennan Shacklett for the wonderful mpbRbx and obby-dsl repo upon which we have built and tested our ideas. We would also like to thank Prof. Kayvon Fatahalian for their valuable comments and support throughout this project.

References

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. 2019. Go-Explore: a New Approach for Hard-Exploration Problems. *arXiv preprint arXiv:1901.10995* (2019). https://arxiv.org/abs/1901.10995

Roy Fox, Ari Pakman, and Naftali Tishby. 2016. Taming the Noise in Reinforcement Learning via Soft Updates. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence (UAI)*. 202–211. https://auai.org/uai2016/proceedings/papers/219.pdf

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement Learning with Deep Energy-Based Policies. In *Proceedings of the 34th International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 1352–1361. https://proceedings.mlr.press/v70/haarnoja17a.html

Yujia Li, Kevin Swersky, and Richard Zemel. 2015. Generative Moment Matching Networks. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*. 1718–1727. https://proceedings.mlr.press/v37/li15.html

Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. 2012. On Stochastic Optimal Control and Reinforcement Learning by Approximate Inference. In *Robotics: Science and Systems VIII (RSS 2012)*. 1–8. https://www.roboticsproceedings.org/rss08/p45.pdf

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347* (2017). https://arxiv.org/abs/1707.06347

Brennan Shacklett et al. 2023. An Extensible, Data-Oriented Architecture for High-Performance, Many-World Simulation. In *SIGGRAPH 2023*.

Marc Toussaint. 2009. Robot Trajectory Optimization using Approximate Inference. In *Proceedings of the 26th International Conference on Machine Learning*. 1049–1056. https://icml.cc/Conferences/2009/papers/271.pdf

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum Entropy Inverse Reinforcement Learning. In *AAAI*. 1433–1438.