Utilizing Priors to Guide Exploration

Jyothish Pari
New York University

Denis Yarats
New York University
Facebook AI Research

Lerrel Pinto
New York University

***Abstract*—Exploration is a important component of any reinforcement learning algorithm. Consequently, recent works have developed reward-free methods of exploration that seek to obtain maximum coverage of an environment which is then utilized for a downstream task. However, we show that reward-free exploration is hard in complex environments. In addition, when there is knowledge of the downstream tasks, there is no straightforward method to prioritize exploration to the desired region while still being able to eventually reach regions outside of the desire one(s). We propose to modify existing entropy maximizing exploration algorithms by adding a varying distance metric before estimating entropy. Specifically, by dynamically stretching and compressing the representation space we are able to direct the exploration of entropy maximizing agents. We show that, in a fewer number of steps, our method yields better coverage of desired regions compared to reward-free method while being able to achieve full coverage given enough time.**

I. Introduction

There has been multiple works such as [13] [8] [6] that aim to develop unsupervised exploration methods. As the name suggests, these methods do not use the environments rewards and instead create their own reward. Typically these reward functions measure the novelty of a state. Some of these methods are entropy maximization and model disagreement. While they successfully are able to explore most of an environment given enough time, they can not utilize any prior to guide their exploration to make the exploration more time efficient.

In our work, we formulate a problem where we have a region of interest that we want the agent to explore. One can treat the region of interest as a Gaussian mixture model (GMM). We represent the region of interest through a finite set of points, which can be treated as the centers of the GMM. Therefore, we ideally want an exploratory agent that is able to direct its exploration towards the region(s) of interest. We achieve this by adding changing the distance metric of the aforementioned entropy maximization methods. By dynamically stretching and compressing the space, we are able to directly control where an entropy maximizing agent explores.

We work on various maze environments and plot the agent's visitation distribution to convey that our method is able to explore the region of interest initially while still being able to explore outside of the region given enough time. In addition, through the setup proposed in [14] we train an offline reinforcement learning agent to reach a goal location in the desired region, and show that our method is more time efficient at generating exploratory data that covers the region of interest.

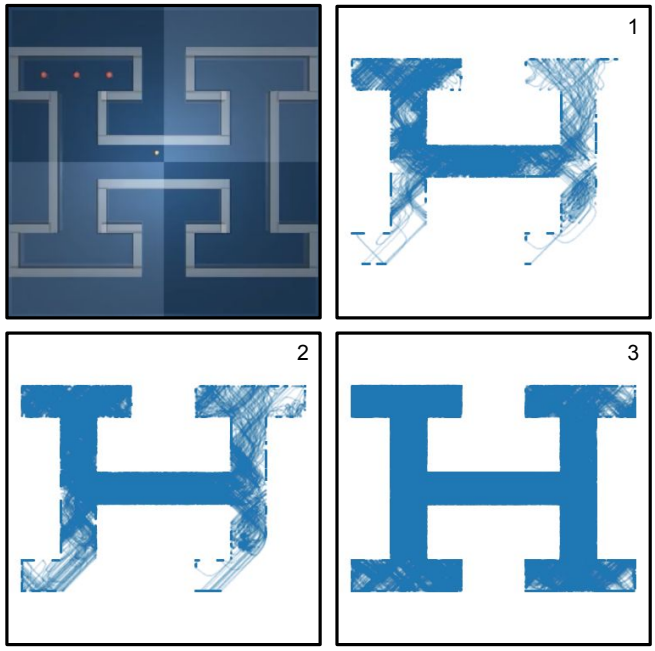

Fig. 1. Given the maze environment we prioritize exploration by placing 3 guide points, which are the red points, so that the agent explores the top right branch first. By showing 3 snapshots of the agent's visitation we see that the agent is still able to achieve uniform coverage of the environment

II. Related Work

The simplest form of exploration is in a tabular setting where one can utilize count based methods [10] and even extend it to non-tabular methods [11]. In addition there have been many method that perform intrinsic exploration in continuous settings. One class of them rely on model disagreement such as ICM [8]. By leveraging the error of a forward model, the novelty of a state can be determined. Other works such as [1] utilize two networks that map the agent's state to a feature space. One of the networks is frozen and the other network is trained to match the frozen network's output, which is known as distillation. Therefore there will be higher error in novel states, which is then utilized as an intrinsic reward.

Other works [13] [6] seek to maximize the entropy over the state space and consequently rely on non-parametric methods to estimate entropy which utilizes nearest neighbors information for each state. These method also work on complex domains such as images, where the nearest neighbors are calculated on a feature space. Therefore, it requires that there

is a good encoder and the aforementioned works learn the encoder through self supervised techniques like prototypical clustering or contrastive learning.

In addition to guided exploration there are multiple alternative methods. One possibility is explicit demonstrations of performing a task. Works such as [12] utilize human data to initialize an agent through imitation learning. Naturally, this will guide exploration to better states when improving the agent through reinforcement learning.

We find previous works that do not require demonstrations and instead explore other forms of data to distill a prior to guide exploration. For example, language is one form, which allows the agents to reason about another mode of data as done in [4] in addition to [7].

III. Approach

Methods like [6] and [13] seek to maximize the entropy of the agent's explored states. They achieve this via a non-parametric estimation of the states. Specially they utilize [9] which calculates the entropy over a set of n points, $\{z_i\}$, $z_i \in \mathbb{R}^q$.

$$\hat{\mathbb{H}}(z_i) = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{k}{n v_i^k} + C_k \propto \sum_{i=1}^{n} \log v_i^k \tag{1}$$

$$v_i^k = \frac{||z_i - z_i^{(k)}||^{nq} \pi^{n\frac{q}{2}}}{\Gamma(n\frac{q}{2}+1)} \tag{2}$$

[6] finds that there is better stability when averaging over the k nearest neighbors which yields the following estimator.

$$\hat{\mathbb{H}}(z_i) := \log(c + \frac{1}{k} \sum_{z_i^{(j)} \in N_k(z_i)} ||z_i - z_i^{(j)}||^{nq}) \tag{3}$$

We propose to include a metric on the state or representation space which the nearest neighbor distances are calculated on. A metric is a function that calculates the distance between points. Specially we establish a metric scale factor $d(z)$, to scale the original nearest neighbor distance $||z_i - z_i^{(k)}||^{nq}$. This modifies the entropy estimator to the following.

$$\hat{\mathbb{H}}(z_i) := \log\left(\frac{c + \frac{1}{k}\sum_{z_i^{(j)} \in N_k(z_i)} ||z_i - z_i^{(j)}||^{nq}}{d(z_i)}\right). \tag{4}$$

$$\hat{\mathbb{H}}(z_i) := \log(c + \frac{1}{k} \sum_{z_i^{(j)} \in N_k(z_i)} ||z_i - z_i^{(j)}||^{nq}) - \log(d(z_i)), \tag{5}$$

where we set

$$d(z_i) = \exp\left(\frac{\alpha * \text{top}_r}{\text{top}_{dist} + d} ||z_i - G_z||\right), \tag{6}$$

with G_z being the set of guide points $\{g_i\}_1^{n_g}$. Guide points capture the region of interest where we want the agent to prioritize exploration. These set of points don't have to be in R^q and can reside in a subspace. top_r is the average of the top k_r rewards from a replay buffer sample calculated using the original entropy estimator without scaling distances. top_{dist} is the distance to the state in the sample from the replay buffer that is farthest from a guide point. In addition we have hyperparameters α and d which allow us to control how much density we want in the regions of interest. They can be treated as the centers of a Gaussian mixture model. Below we show the final form of our reward function.

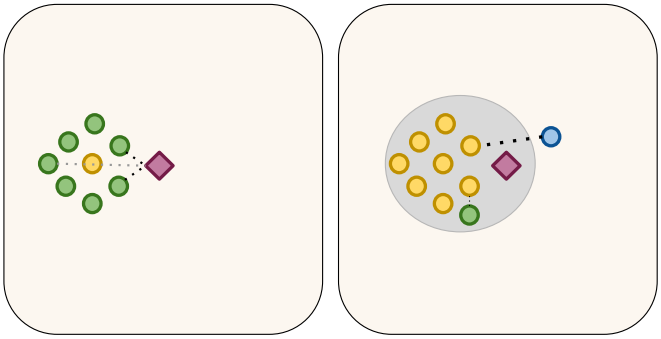

Fig. 2. We illustrate two possible scenarios, where the circles represents states, and the diamond is a guide point. In the left image, the yellow circle is present in the replay buffer and the green circles are the states where the agent currently explored. Based on our reward function the green states closet to the guide point will obtain the highest reward because all they are all equidistant from the present yellow state. However, on the right image if the agent currently explored the green state and blue state, then the blue state will obtain the higher reward. This shows how we prioritize guiding the agent towards the guide point as well as expanding the explored coverage around the guide point.

$$\log(c + \frac{1}{k} \sum_{z_i^{(j)} \in N_k(z_j)} ||z_i - z_i^{(j)}||^{nq}) - \frac{\alpha * top_r}{top_{dist} + d} ||f_\theta(s) - G_z|| \tag{7}$$

Intuitively our method initially prioritizes reaching the guide points because calculating distances near guide points yeild larger reward values than those farther away. G_z is nearest guide point to z. However, once the agent explores the desired regions with enough density, our reward function prioritizes exploring the border of the coverage as shown in Figure 2.

We first perform n steps of greedy exploration where we set our reward function to be $r(s) = e_r - \beta * ||f_\theta(s) - G_z||$. This allows us to obtain a coverage to some of the guide points from which we then switch our reward function to equation 7 which then seeks to progressively keep expanding the coverage.

IV. Experimental Evaluation

We demonstrate that our method is able to successfully explore the desired regions first while still being able to obtain full coverage of the environment given enough time. We analyze our method against two baselines, being unsupervised exploration through ICM_APT [5] as well as setting the reward function to $e_r - ||f_\theta(s) - G_z||$, where e_r is the entropy reward calculated from ICM_APT. The later of the two methods does prioritize exploration near the guide points. However, as shown in Figure 3, after a certain radius away from the guide point, the distance term, $||f_\theta(s) - G_z||$ will be larger

than the entropy reward term, e_r. This causes the agent to not explore significantly past that radius, and thus exploration is confined to a fixed region. We test the method on two environments of significantly different sizes. On the smaller environment which is the bottom row, we can see the method obtains reasonable coverage while being able to prioritize the right room. However, the larger environment conveys that the method isn't able to obtain coverage of the entire environment. We then evaluate our method on both environments as shown in Figure 5. Using the same hyper parameters between both environments we show that our method is able to obtain coverage of the region of interest first while still being able to cover the entire environment.

In addition, we utilize offline reinforcement learning to evaluate whether our method is able to obtain sufficient coverage for an agent to learn to reach a goal in the desired region. This setup is taken from [14]. We run our method as well as unsupervised ICM_APT from [5]. We tested it on an environment that has distinct rooms as shown in Figure 1. As shown in Figure 5 we show that training an agent using td3 [3] yields better performance using the exploratory data collected by our method. The performance difference is most significant when using the exploratory data from an earlier time step. This is because our method prioritizes the region of interest whereas an unsupervised exploration method doesn't. Therefore, on average it will take longer for an unsupervised exploratory agent to explore a specific region. Eventually the unsupervised agent is able to explore the region of interest, however as the environment becomes larger it takes exponentially more time for an unsupervised method to obtain coverage of a desired region.

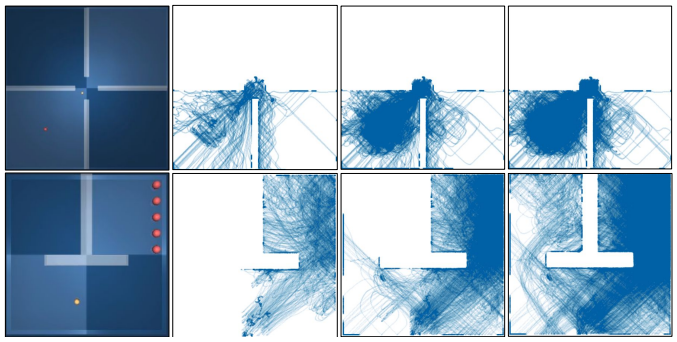

Fig. 3. We demonstrate why a simple reward function $entropy_r - \gamma * NN_{dist}$ is not sufficient to explore the entire environment, where $entropy_r$ is the reward from the entropy calculation. We show the visitation distribution at 3 timesteps, being 150k, 500k, 1M.

V. Discussion

We convey that guiding exploration is useful when there is access to a prior of the downstream tasks. We utilize guide points which need to be manually created. This is a potentially a limitation of this work because in more complex environments and agents it may be difficult to create a guide point in the right space. However, one can treat guide points as more than points and instead representations. Works like [2] compress trajectories into representations and perform optimal transport to calculate distances. Therefore, one can substitute representations for our guide points and substitute a new distance metric instead of a euclidean metric. In addition, we see that having representations evolve over an agent's lifetime are more adaptable. Works such as [13], illustrate how combining exploration with representation learning yield more efficient exploration. Consequently, an interesting direction would be to treat the guide points as a set of observations and as the representations evolve over time so will the distances thus allowing the agent dynamically change which regions of the environment it prioritizes for exploration.

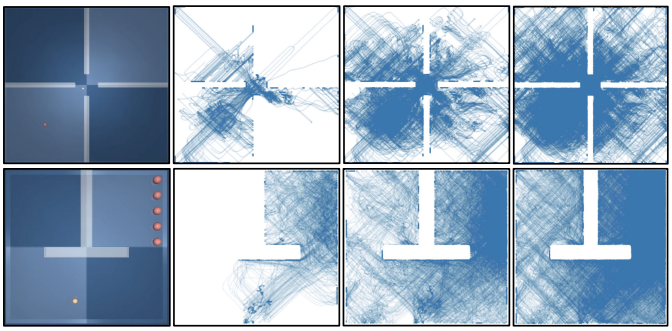

Fig. 4. We illustrate the visitation distribution of method for the given guide points across different time steps being 150k, 500k, 1M. We see that eventually our method is able to cover the entire environment.

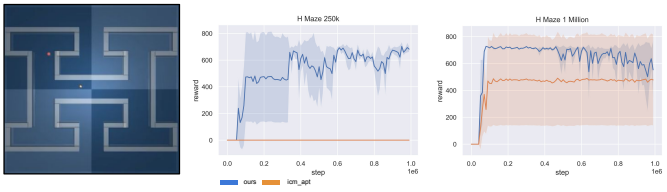

Fig. 5. We evaluate our method and the baseline on the shown goal location across different time steps of the exploratory data collection. At fewer number of time steps we show that our method is more successful.

Acknowledgment

I would like to thank David Brandfonbrener for guiding me on the theoretical end and discussing the applications of this work.

References

[1] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation, 2018. URL https://arxiv.org/abs/1810.12894.

[2] Samuel Cohen, Brandon Amos, Marc Peter Deisenroth, Mikael Henaff, Eugene Vinitsky, and Denis Yarats. Imitation learning from pixel observations for continuous control. 2021.

[3] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *CoRR*, abs/1802.09477, 2018. URL http://arxiv.org/abs/1802.09477.

[4] Brent Harrison, Upol Ehsan, and Mark O. Riedl. Guiding reinforcement learning exploration using natural language. *CoRR*, abs/1707.08616, 2017. URL http://arxiv.org/abs/1707.08616.

[5] Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. URLB: unsupervised reinforcement learning benchmark. *CoRR*, abs/2110.15191, 2021. URL https://arxiv.org/abs/2110.15191.

[6] Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. *CoRR*, abs/2103.04551, 2021. URL https://arxiv.org/abs/2103.04551.

[7] Jesse Mu, Victor Zhong, Roberta Raileanu, Minqi Jiang, Noah Goodman, Tim Rocktäschel, and Edward Grefenstette. Improving intrinsic exploration with language abstractions, 2022. URL https://arxiv.org/abs/2202.08938.

[8] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. *CoRR*, abs/1705.05363, 2017. URL http://arxiv.org/abs/1705.05363.

[9] Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American Journal of Mathematical and Management Sciences*, 23(3-4):301–321, 2003. doi: 10.1080/01966324.2003.10737616. URL https://doi.org/10.1080/01966324.2003.10737616.

[10] Alexander L. Strehl and Michael L. Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2007.08.009. URL https://www.sciencedirect.com/science/article/pii/S0022000008000767. Learning Theory 2005.

[11] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. exploration: A study of count-based exploration for deep reinforcement learning, 2016. URL https://arxiv.org/abs/1611.04717.

[12] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Caglar Gulcehre, Ziyun Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pages 1–5, 2019.

[13] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. *CoRR*, abs/2102.11271, 2021. URL https://arxiv.org/abs/2102.11271.

[14] Denis Yarats, David Brandfonbrener, Hao Liu, Michael Laskin, Pieter Abbeel, Alessandro Lazaric, and Lerrel Pinto. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *CoRR*, abs/2201.13425, 2022. URL https://arxiv.org/abs/2201.13425.