Density-Based Bonuses on Learned Representations for Reward-Free Exploration in Deep Reinforcement Learning

Anonymous Authors[1]

Abstract

In this paper, we study the problem of representation learning and exploration in reinforcement learning. We propose a framework to compute exploration bonuses based on density estimation, that can be used with any representation learning method, and that allows the agent to explore without extrinsic rewards. In the special case of tabular Markov decision processes (MDPs), this approach mimics the behavior of theoretically sound algorithms. In continuous and partially observable MDPs, the same approach can be applied by learning a latent representation, on which a probability density is estimated.

1. Introduction

Exploration is a major challenge in reinforcement learning (RL), where an agent must continually gather information about its environment in order to learn how to act *optimally*. In an online setting and when rewards are available, the performance of an RL algorithm is usually defined in terms of its regret, which is the different between the rewards gathered by an optimal policy and the rewards gathered by the algorithm. In a *reward-free* setting, the performance can be evaluated by the number of trajectories (i.e., the sample complexity) required for the algorithm to build a dataset that can be used by a planning algorithm to compute a near-optimal policy for any possible reward function (Jin et al., 2020a).

In finite MDPs, the regret, the sample complexity, and the computational complexity of near-optimal algorithms[1] scale with the number of states and actions (Jaksch et al., 2010; Jin et al., 2020a; Domingues et al., 2021). Hence, beyond small finite environments, we are faced with the challenge of designing provably efficient algorithms that are, at the same time computationally efficient and able to handle arbitrarily large state spaces.

Under structural assumptions on the MDP that allow generalization, it is possible to derive algorithms whose performance bounds depend on the *dimension* of the state-action space with respect to some representation, instead of its *cardinality*. The representation is often in the form of a metric, kernel, or linear function approximation (Ortner & Ryabko, 2012; Chowdhury & Gopalan, 2019; Jin et al., 2020b; Song & Sun, 2019; Sinclair et al., 2019; Yang & Wang, 2020; Yang et al., 2020; Domingues et al., 2020; Chowdhury & Oliveira, 2020; Wang et al., 2020). Although such algorithms present theoretical guarantees, they may suffer from two main drawbacks: a high computational complexity, preventing them from being applicable to large-scale MDPs; and the requirement that a *good* representation should be provided, e.g. in the form of a feature map or a kernel function.

Deep reinforcement learning algorithms have shown success in tackling large-scale reinforcement learning problems, and allow combining representation learning methods and RL in an end-to-end fashion. However, deep RL methods may have an excessively large sample complexity in hard exploration problems, and we have a limited theoretical understanding of such methods, especially in terms of how to perform exploration optimally. In this paper, we focus on the problem of reward-free exploration in deep RL, and study an exploration method that draws inspiration from theoretically grounded algorithms for finite or low-dimensional MDPs.

Theoretically-inspired exploration bonuses In finite MDPs where a reward function is available, algorithms relying on exploration bonuses scaling with $\sqrt{1/n(s,a)}$, where $n(s,a)$ is the number of visits to a state-action pair (s,a), have been shown to achieve near-optimal performance (Strehl & Littman, 2008; Azar et al., 2017). Such bonuses depending on $n(s,a)$ have inspired many algorithms that improve exploration in deep RL. For instance, Bellemare

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]An algorithm is *near-optimal* if its performance bound is equal to the lower bound, up to constants and logarithmic terms with respect to the learning horizon.

et al. (2016) propose a method to compute pseudo-counts approximating $n(s, a)$ using density estimation on images, and Tang et al. (2017) use locally-sensitive hashing to map continuous states to discrete embeddings, where explicit counts are computed. A common property among all these approaches is that the more a state-action pair (s, a) is visited, the smaller the bonus at (s, a). This property is also satisfied for algorithms such as Random Network Distillation (RND) (Burda et al., 2019) and Never Give Up (NGU) (Badia et al., 2020), that manage to tackle many hard exploration problems.

Reward-free exploration In the absence of rewards from the environment, a major question is to decide *what the agent should maximize*. Hazan et al. (2019) and (Guo et al., 2021) propose algorithms that search a policy maximizing the entropy of its induced state-space distribution. One disadvantage of this approach is that the learned policy may avoid regions of the state-space that do not increase the entropy, but may be important for downstream tasks.[2] Another approach is to evaluate the agent by its ability to explore and *build a dataset of transitions*, which can then be used to build a model and compute a near-optimal policy for *any possible reward function* (Jin et al., 2020a). Kaufmann et al. (2021) and Ménard et al. (2020) show that under this criterion, near-optimal algorithms can be obtained using bonuses of the form $\sqrt{1/n(s, a)}$ or $1/n(s, a)$. This motivates the use of such exploration bonuses both in the presence and in the absence of extrinsic rewards. A key difference between the dataset approach and entropy maximization, is that the former will recompute a policy in every episode, aiming to visit different regions of the state space, whereas the latter aims to find a single maximum-entropy policy that may fail to cover some regions of the state-space.

Representation Learning When dealing with environments with infinitely many states and partial observability, learning a good state representation is crucial. In general, this representation must satisfy at least two properties: (i) it should allow generalization across similar states; and (ii) it should be a sufficient statistic for the past transitions and discard irrelevant information. However, whether a representation is good or not will depend on the purpose for which we design the learning agent. For instance, we might look for representations that ignore uncontrollable features of the environment, that are useful to build world models, or that allow generalization across tasks (Pathak et al., 2017; Azar et al., 2019; Guo et al., 2020). Given the vast amount of representation learning methods, we believe that it is important to look for a unified exploration strategy that allows the agent to integrate different methods to learn representations and that do not rely on extrinsic rewards from the environment, which is the problem that we investigate in this paper.

[2]For instance, assume that our goal is to learn the transitions of a GridWorld MDP consisting of a large and a small room, which are disconnected. The agent starts outside both rooms, and, in the initial state, is given the choice to go to one of them. Maximum-entropy policies may always choose to go to the larger room and ignore the smaller one, and this would hurt the estimation of a transition model in the small room.

Our contribution We study a framework in which exploration is encouraged by intrinsic rewards (or exploration bonuses) computed as a function of a density estimation on top of learned representations, which are inspired by the $1/n(s, a)$ bonuses used in reward-free finite MDPs. We argue that the advantages of this approach are:

(i) It mimics the behavior of near-optimal algorithms in finite or low-dimensional MDPs and it generalizes naturally to large-scale problems.

(ii) It allows us to integrate and combine different representation learning methods, while keeping the same strategy for computing exploration bonuses.

(iii) It enables exploration in the absence of extrinsic rewards.

2. Method

We consider an agent interacting with an environment in episodes of length H, such that, at every time t, the agent chooses an action $a_t \in \mathcal{A}$ and it receives an observation $o_{t+1} \in \mathcal{O}$. Let $h_t = (o_i, a_{i-1})_{i \leq t} \in \mathcal{H}$ be the history of observations and actions up to time t, which are stored in a replay buffer. We propose the following method for reward-free exploration with learned representations:

- Given a history h_t, we use a representation function f to compute an embedding $x_t = f(h_t)$, where f is optimized according to *any representation learning algorithm* trained with data from the replay buffer;
- We compute $p(x_t, a_t)$ representing the probability of observing the embedding x_t and the action a_t in the replay buffer, where p is learned using *any density estimation method* and we define the intrinsic reward at (h_t, a_t) as $r(x_t, a_t) = 1/p(x_t, a_t)$;
- A reinforcement learning agent is trained to maximize the sum of intrinsic rewards.

Below, we argue that in the special case of finite environments, this algorithm becomes similar to the `RF-Express` algorithm of Ménard et al. (2020) that has sample complexity guarantees for reward-free exploration. Then, we propose a simple method to estimate the inverse density

$1/p(x_t, a_t)$ that is applicable to general environments and evaluate it experimentally combining different representation learning methods.

Fully observable finite MDPs In this case, the observation o_t completely describes the state of the environment, and we can take $x_t = f(h_t) = o_t \in \mathcal{O}$. Then, for any $(x, a) = (o, a) \in \mathcal{O} \times \mathcal{A}$ we can compute the probability density after T transitions as:

$$p(x, a) = p(o, a) = n_T(o, a)/T,$$

where $n_T(o, a) = \sum_{i=1}^{T} \mathbb{I}\{(o_i, a_i) = (o, a)\}$ is the number of times that the observation-action pair (o, a) appears in the history of T transitions. Consequently, the intrinsic reward at $(x, a) = (o, a)$ becomes $r(x, a) = 1/p(x, a) = T/n_T(o, a)$. An optimal policy π_T is invariant to the scaling of the rewards: we can multiply the intrinsic reward by any constant and π_T will remain optimal. This implies that π_T also maximizes the sum of rewards $r(x, a)/T$, which is equivalent to using $1/n_T(o, a)$ as intrinsic reward. Hence, an algorithm using intrinsic rewards defined as $1/p(x, a)$ is expected to have a behavior similar to that of `RF-Express` (Ménard et al., 2020) that uses bonuses proportional to $1/n_T(o, a)$ and has a near-optimal sample complexity. Notice that, however, the scaling $1/T$ is time-dependent, which is not an issue for `RF-Express` that recomputes a policy in every episode, but in deep RL we need to be careful when optimizing non-stationary rewards.

Kernel density estimation (KDE) When a kernel function k is available, such that $k((x, a), (x', a'))$ represents the similarity between the state-action pairs (x, a) and (x', a'), Domingues et al. (2020) show that exploration bonuses of the form

$$r(x, a) = \left(\beta + \sum_{i=1}^{T} k((x_i, a_i), (x, a))\right)^{-1} \tag{1}$$

allows us to obtain a sample efficient exploration algorithm for continuous MDPs, where $\beta > 0$ is a regularization factor. We can define a regularized kernel density estimator as

$$p(x, a) = \frac{1}{T}\left(\beta + \sum_{i=1}^{T} k((x_i, a_i), (x, a))\right) \tag{2}$$

and notice that the bonus (1) is proportional to $1/p(x, a)$.

Fitted KDE using learned representations As a simple method for density estimation, we propose the use of a learned version of (2). We define a kernel function based on the learned representation f as follows:

$$k((x, a), (x', a')) = \frac{1}{1 + ||f(x) - f(x')||_2}. \tag{3}$$

For simplicity, we ignore the actions a and a' in the definition of the kernel[3], and we consider $\beta = 0$. Now, due to computational issues, we need to avoid the sum over all T past transitions in (2). To do so, let J be a random variable uniformly distributed on $\{1, \dots, T\}$, and notice that, for $\beta = 0$, we have

$$\frac{1}{p(x, a)} = \frac{1}{\mathbb{E}_J\left[k((x_J, a_J), (x, a)\right]} \leq \mathbb{E}_J\left[\frac{1}{k((x_J, a_J), (x, a)}\right]$$

by Jensen's inequality. Consequently, we can learn a function $g(x, a)$ to estimate an upper bound on $1/p(x, a)$[4] by minimizing the following loss:

$$\min_g \mathbb{E}_{(x,a)}\left[\left(g(x, a) - \mathbb{E}_{\mathcal{B}}\left[\frac{1}{\frac{1}{|\mathcal{B}|}\sum_{(x_j, a_j \in \mathcal{B})} k((x_j, a_j), (x, a)}\right]\right)^2\right]$$

where $\mathcal{B}$ is a batch of histories and actions (h_j, a_j) sampled uniformly from the replay buffer and $x_j = f(h_j)$. Similarly, to compute the expectation over (x, a), we take samples (h, a) uniformly from the replay, and set $x = f(h)$.

Finally, we propose to use the following exploration bonus at any state-action pair (x, a):

$$r(x, a) = g(x, a) + \left|g(x, a) - \frac{1}{\mathbb{E}_J\left[k((x_J, a_J), (x, a)\right]}\right|$$

where the term added to $g(x, a)$ increases the bonus to compensate the approximation error that is made when learning the kernel density estimator (2), and is computed by sampling (x_J, a_J) uniformly from the replay buffer.

Although we propose the bonuses based on fitted KDE as a first method to test our approach, other density estimation methods can be used, such as normalizing flows (Rezende & Mohamed, 2015).

Also, notice the bonuses computed fitted KDE have a similar idea as the ones used by Never Give Up (NGU) (Badia et al., 2020), but NGU uses only data from a single episode for its kernel-based bonuses, whereas fitted KDE uses all data from the past stored in the replay buffer. Another similarity is that NGU includes a multiplicative term in the bonuses using prediction errors from RND, whereas we use an additive term representing the error in predicting the KDE.

[3]This is coherent with similar exploration methods in deep RL, which encourage visits to novel states, instead of novel state-action pairs, e.g. (Bellemare et al., 2016; Badia et al., 2020).

[4]Using an upper bound instead of $1/p(x, a)$ as exploration bonus increases the amount of exploration performed by the algorithm. As long as the bonuses preserve the property that frequently visited states receive smaller bonuses, this should not significantly impact the algorithm's performance.

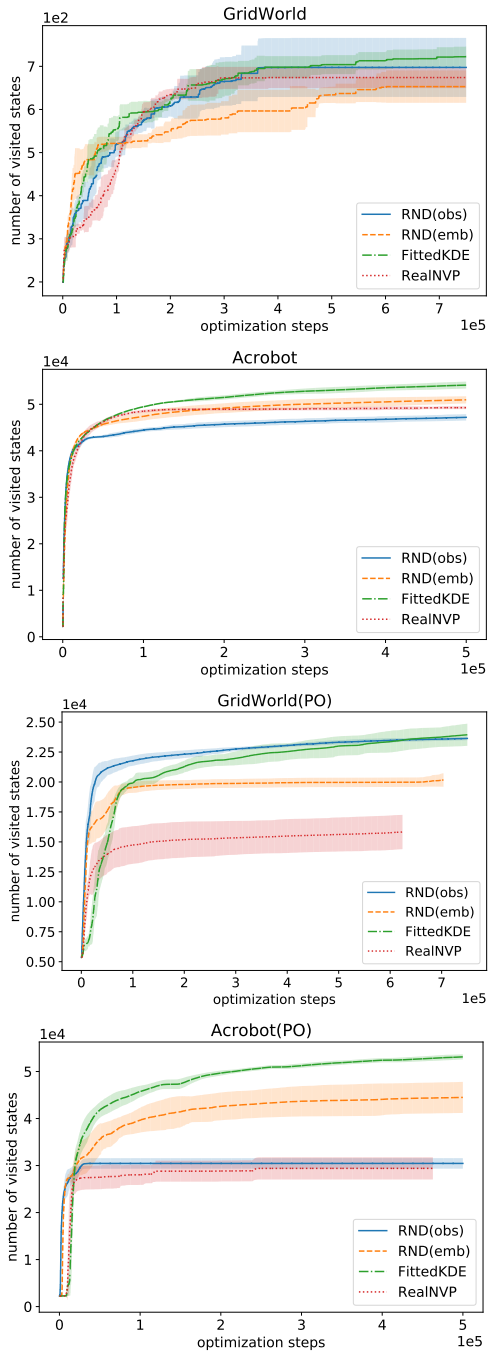

Figure 1. Number of visited states versus training steps for different algorithms in four different environments: GridWorld and Acrobot and their partially observable (PO) versions. RND(emb) and RND(obs) means that the RND predictions are applied to the embeddings and to the observations, respectively. Results averaged over 4 runs.

3. Experiments

To evaluate our framework, we consider four environments: (i) a fully observable GridWorld with 9 rooms, where the state and the observations are one-hot encodings of the position of the agent; (ii) a partially observable version GridWorld, where a velocity is included in the state, but the observation only shows the position; (iii) Acrobot[5]; and (iv) a partially observable version of Acrobot, where only the positions, and not the velocities, are observed.

For representation learning, we combined three methods: (i) action prediction (Pathak et al., 2017; Badia et al., 2020); (ii) adjacency regularization (Guo et al., 2021); and (iii) observation prediction, where the embedding of the past $f(h_t)$ is used to predict the next observation o_{t+1}. The latter was only used in the partially observable enviroments. For density estimation, we used the Fitted KDE method presented in Section 2, and Real NVP (Dinh et al., 2017). Finally, as reinforcement learning algorithm, we used Soft-DQN (Haarnoja et al., 2018; Vieillard et al., 2020).

Additionally, as a baseline, we consider RND (Burda et al., 2019) that computes intrinsic rewards based on the error incurred when predicting the output of a random neural network. We study the case where the predictions are made using the observations o_t only, and also the case where the predictions are made on top of the embeddings $x_t = f(h_t)$. Notice that the latter case is a way to generalize RND to environments where a representation needs to be learned.

We evaluate the algorithms with respect to how quickly they are able to discover new states in the environment.[6] Figure 1 shows the number of visited states versus the number of training steps with our framework, compared to the RND baselines. We observe that Fitted KDE allows us to discover at least as many states as the baselines in the GridWorld, and more states are discovered in the Acrobot environments.

4. Conclusion

We propose a unified framework to compute exploration bonuses in reward-free exploration for RL using density estimation. This framework is inspired by provably efficient algorithms in tabular MDPs and we generalize it to arbitrary environments by allowing the algorithm to integrate any set of representation learning and density estimation methods. We illustrate its empirical effectiveness in continuous and partially observable problems, by proposing a simple method to compute density based bonuses, Fitted KDE, that can be related to the intrinsic rewards used by state-of-the-art methods, such as RND (Burda et al., 2019) and NGU (Badia et al., 2020). Interesting directions for future work include testing our approach in more complex environments, such as hard exploration games in Atari, and investigate the impact of different representation methods.

[5]From OpenAI Gym.

[6]When the states are continuous, we compute the number of visits to discretized states.

References

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pp. 263–272. PMLR, 2017.

Azar, M. G., Piot, B., Pires, B. A., Grill, J.-B., Altché, F., and Munos, R. World discovery models. *arXiv preprint arXiv:1902.07685*, 2019.

Badia, A. P., Sprechmann, P., Vitvitskyi, A., Guo, D., Piot, B., Kapturowski, S., Tieleman, O., Arjovsky, M., Pritzel, A., Bolt, A., and Blundell, C. Never give up: Learning directed exploration strategies. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Sye57xStvB.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, 2016. URL https://proceedings.neurips.cc/paper/2016/file/afda332245e2af431fb7b672a68b659d-Paper.pdf.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1lJJnR5Ym.

Chowdhury, S. R. and Gopalan, A. Online learning in kernelized markov decision processes. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, 2019.

Chowdhury, S. R. and Oliveira, R. No-Regret Reinforcement Learning with Value Function Approximation: a Kernel Embedding Approach. *arXiv e-prints*, art. arXiv:2011.07881, November 2020.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Domingues, O. D., Ménard, P., Pirotta, M., Kaufmann, E., and Valko, M. Regret bounds for kernel-based reinforcement learning. *arXiv preprint arXiv:2004.05599*, 2020.

Domingues, O. D., Ménard, P., Kaufmann, E., and Valko, M. Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited. In Feldman, V., Ligett, K., and Sabato, S. (eds.), *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pp. 578–598. PMLR, 16–19 Mar 2021. URL http://proceedings.mlr.press/v132/domingues21a.html.

Guo, Z. D., Pires, B. A., Piot, B., Grill, J.-B., Altché, F., Munos, R., and Azar, M. G. Bootstrap latent-predictive representations for multitask reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. URL http://proceedings.mlr.press/v119/guo20g.html.

Guo, Z. D., Azar, M. G., Saade, A., Thakoor, S., Piot, B., Pires, B. A., Valko, M., Mesnard, T., Lattimore, T., and Munos, R. Geometric entropic exploration. *arXiv preprint arXiv:2101.02055*, 2021.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Hazan, E., Kakade, S., Singh, K., and Van Soest, A. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pp. 2681–2691. PMLR, 2019.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4), 2010.

Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. Reward-free exploration for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, 2020a. URL http://proceedings.mlr.press/v119/jin20d.html.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Proceedings of Thirty Third Conference on Learning Theory*, 2020b. URL http://proceedings.mlr.press/v125/jin20a.html.

Kaufmann, E., Ménard, P., Darwiche Domingues, O., Jonsson, A., Leurent, E., and Valko, M. Adaptive reward-free exploration. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, 2021. URL http://proceedings.mlr.press/v132/kaufmann21a.html.

Ménard, P., Domingues, O. D., Jonsson, A., Kaufmann, E., Leurent, E., and Valko, M. Fast active learning for pure exploration in reinforcement learning, 2020.

Ortner, R. and Ryabko, D. Online regret bounds for undiscounted continuous reinforcement learning. In *Advances in Neural Information Processing*

Systems, 2012. URL https://proceedings.neurips.cc/paper/2012/file/f9a40a4780f5e1306c46f1c8daecee3b-Paper.pdf.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pp. 2778–2787. PMLR, 2017.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538. PMLR, 2015.

Sinclair, S. R., Banerjee, S., and Yu, C. L. Adaptive discretization for episodic reinforcement learning in metric spaces. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–44, 2019.

Song, Z. and Sun, W. Efficient model-free reinforcement learning in metric spaces. *arXiv preprint arXiv:1905.00475*, 2019.

Strehl, A. L. and Littman, M. L. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Xi Chen, O., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. #Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3a20f62a0af1aa152670bab3c602feed-Paper.pdf.

Vieillard, N., Pietquin, O., and Geist, M. Munchausen reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/file/2c6a0bae0f071cbbf0bb3d5b11d90a82-Paper.pdf.

Wang, R., Du, S. S., Yang, L., and Salakhutdinov, R. R. On reward-free reinforcement learning with linear function approximation. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17816–17826. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/ce4449660c6523b377b22a1dc2da5556-Paper.pdf.

Yang, L. and Wang, M. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pp. 10746–10756. PMLR, 2020.

Yang, Z., Jin, C., Wang, Z., Wang, M., and Jordan, M. Provably efficient reinforcement learning with kernel and neural function approximations. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/file/9fa04f87c9138de23e92582b4ce549ec-Paper.pdf.