Survey on Transfer Reinforcement Learning for Domains Differing in State-Action Spaces

Payal Mohapatra,[1] Akash Pandey,[2]

[1] *PMS4829, Department of Electrical and Computer Engineering, Northwestern University*
[2] *APA2237, Department of Mechanical Engineering, Northwestern University*

1. Motivation

Reinforcement learning (RL) is a framework to model sequential decision-making process via multiple incentivized interactions with the environment [1]. RL is inspired from interdisciplinary ideas from psychology, behavioral sciences, computer science and statistics. Traditionally RL modelling used tabular methods and model free methods like function approximations capable of handling lower dimensional state-spaces. However, with the introduction of deep neural networks as a promising candidate for model-free approaches Deep RL (DRL) [2] has been successfully used in many complex tasks. In spite of DRL's meteoric rise to popularity over the past few years, it still suffers with a trade-off between exploration vs. exploitation [1]. This is one of the major limitations in deploying RL frameworks in real-world use cases since collecting sufficient samples from a high-dimensional or continuous environment with sparse rewards can be prohibitive. As a result the training of an *agent* for an RL task in complex environments is an expensive affair. This is an open issue with reinforcement learning that it does not generalize well which means even with slight perturbation in environment it cannot perform well. This makes training robust agents difficult. ***Transfer Learning*** is a technique where the knowledge from a similarly motivated prior task can be used to leverage learning a new task. Transfer learning gained its popularity in supervised learning settings [3]. Many RL applications have also demonstrated the effectiveness of transfer learning to train agents in a target domain with lesser interactions with the environment.
In this survey report[1] we present the following :

- state-of-the art work on transfer learning in reinforcement learning.
- we identify some gaps in the current work of transfer reinforcement learning and formulate a setting with a proposed methodology and evaluation technique.

2. Literature Survey

2.1. Notations and Preliminaries

Throughout the paper we keep in mind that a Markov Decision Process (MDP) can be represented as a tuple $M = (S, A, \gamma, R, S')$ where S is the current state, S' is the next state, A is the current action taken, R is the reward associated with the state-action pair and γ is the discounting factor. Formally put, transfer learning aims to transfer the latent *knowledge* from a source domain,D_s and task, T_s to a learning task, T_t in the target domain, D_t where $D_s \neq D_t$ and/or $T_s \neq T_t$. In most applications the sample space of D_s is much greater than D_t but it is not a mandatory condition. Transfer Learning in general can be broadly divided into the following categories [3]:

- *Instance Based* [4] approach utilises some selected samples from the source domain to with appropriate weight as training samples in the target domain.
- *Mapping Based* [5] approach relies on arriving at a similar representation of the samples from target and source domains to train.
- *Network Based* [6] approach is most popular for deep learning applications and is intuitively the most similar to the neural processing in humans. The idea is that only the final few layers responsible for learning domain specific features need to be retrained while the other layers are assumed to capture the domain invariant features can can be reused across (similar)domains.
- *Adversarial Based* [7] approach relies on identifying indiscriminate features int he source and target domains and transfers only those. In more recent times, generative modelling is the choice of architecture for such models.

[1] This survey is prepared for the fulfilment of the Deep Reinforcement Learning course in the Spring quarter 2022 at Northwestern University. It naturally assumes the reader's familiarity with reinforcement learning preliminaries. For further details and comments please contact PayalMohapatra2026@u.northwestern.edu or AkashPandey2026@u.northwestern.edu

2.2. *Transfer Learning in the Context of Reinforcement Learning*

The general setting of transfer learning in RL assumes the knowledge of source domain(s) M_s which can be leveraged as an exterior information to learn a policy in target domains M_t. Domain Adaptation is a special case of transfer learning where the $S_s S_t$ and $A_s A_t$ but the tasks can be different $T_s \neq T_t$ resulting in a different reward distribution. Another related area of transfer learning is zero-shot learning where the objective is slightly different .i.e. knowledge from M_s can be directly applied to M_t without any interaction between the agent and environment in M_t. Such a formulation finds use in autonomous vehicles and robotic applications where an agent trained on simulated experiences is expected to perform well in real-world.
In this survey we focus on the transfer learning where we have a limited budget for the agent to gain experience in the target domain and the source and target domain can vary in their state and action spaces. Some of the common approaches used for transfer learning in RL are detailed in the sections 2.2.1 - 2.2.4.

2.2.1. Reward Shaping

In transfer learning reward shaping(RS) aims at providing auxiliary rewards using external knowledge from the source domain. The idea is to indicate most beneficial state-action pairs to the agent. This is especially beneficial when the rewards are sparse. RS modifies the target domain reward R to R' where $R' \rightarrow R + F$. Potential based reward shaping (PBRS) [8] is a well-known reward shaping function, F which is the difference between the two potential functions $\psi(.)$:

$$F(s,a,s') = \gamma\psi(s) - \psi(s')$$

A variation of PBRS is Potential based state-action advice(PBA) [9] where the $\psi(s,a)$ evaluates how beneficial it is to take a certain action from a given state s:

$$F(s,a,s',a') = \gamma\psi(s,a) - \psi(s',a')$$

PBA is limited to RL algorithms which are optimised on-policy.
Remark 1 : Reward shaping as an idea similar to recently popular reward-free/unsupervised RL [10,11] approaches where the potential function can be thought of a representation of the intrinsic/bonus reward awarded to an agent. More details on such approaches are given in Section 4.1. However, for transfer learning it is assumed that there exists a mapping function over the state and actions from the source to target domain, $M_S(s), M_A(a)$ which can be used to augment the reward using $\pi_s(M_S(s), M_A(a))$.
Reward shaping technique is very task-specific and is demonstrated mostly in MDPs with similar state-action spaces [8, 9]. One of the recent works uses dynamic reward shaping function [12] to transferring the knowledge of the source policy to a target with differing state-action space using graph convolution neural networks. Reward shaping needs to be combined with other ideas like Learning from Demonstrations and Policy Transfer to show its effectiveness.

2.2.2. Learning from Demonstration

Learning from Demonstration (LfD) aims at providing guidance to conduct efficient exploration. It can be categorised as offline and online methods. The offline method allows transfer of value functions, policy and state transitions as a pretrained model to initialise the target network [13–15]. In the online method, it is very specific to the underlying RL algorithm used for optimisation [16, 17]. One of the works using Deep Q networks [18] maintain one replay buffer for expert experience and one for the agent's sampled experience and use a stochastic policy for sampling between these two buffers. This ensures a non-zero probability of learning from the expert's experience at each step. An interesting work [19] which draws parallel to imitation learning is GAIL - Generative Adversarial Imitation Learning, also combines the reward shaping principle. GAIL proposes using an augmented reward based on the discriminative index between the expert policy π_E and the current policy π.
Two major challenges with LfD are tackling imperfect demonstrations and avoiding overfitting with the expert policy. Some research have shown success in using a self-adaption imitation learning [20] to selectively opt for optimal self-generated experience over the suboptimal expert demonstrations. Some literature studies use regularisation techniques like as a negative reward function to address overfitting [21–23]. An open research area for LfDs is to explore techniques that are agnostic to the underlying RL algorithm.

2.2.3. Policy Transfer

The general idea of policy transfer if the availability of multiple expert policies that can be leveraged by the student agent. The policies may be distilled [24] by the teacher or the student or directly reused [25]. Teacher refers to source domain and student is the target domain in this report's vernacular.
Remark 2 : Policy transfer is a popular approach in domain adaptation tasks [26] and are vulnerable to differing

state-action space between domains. Also, the training multiple expert policies is a high budget assumption. It is worthwhile to invest in research to relax these constraints and conduct successful policy transfer.

2.2.4. Representation Transfer

Model-free RL approaches have significantly benefited from the popularity of deep neural networks. Representation transfer assumes that the network between the source and target domains can be disentangled as - domain invariant and domain specific components. We can leverage from supervised learning the various techniques of representation transfer like directly use source representation [27] (typically when the source and target domains are similar) or learn a disentanglement function and learn only domain invariant properties [28] (when the source and target MDPs differ in observation space).

2.3. Challenges of Transfer Learning in Reinforcement Learning

One of the major challenges in transfer RL algorithms, is the lack of unified evaluation of transfer efficacy. One of the recent works [29] have proposed some definitions for performance metrics that can serve as standard reporting metrics for demonstrating the efficacy of transfer learning. We have consolidated and presented these in Table 1. There are two categories of metrics, 1)***Learning Process*** of the target agent and 2) ***Transferred Knowledge***. In general, most of the metrics can be thought to address two properties of the learner agent 1) ***Mastery*** : How well the learned agent performs in the target domains? and 2) ***Generalisation*** : Ability of the learning agent to quickly adapt to target domain using TL.
Another challenge is that most of these methods are very specific to the underlying RL algorithm. This can be prohibitive in their impact if the nature of application is changed. Recent research interest is targeted at developing RL algorithm agnostic [30] transfer techniques. Reward-free learning/ unsupervised RL [30,31] training are some of the currently explored algorithms which share the motivation of transfer learning in our opinion. These works are mostly demonstrated on MDPs with a visual observation space, but we conjecture that it can be extended to non-vision tasks as well with some tweaks in the technique. Such task-independent training naturally makes such agents better candidates as a teacher. We expound on this idea further in the Section 4.

3. Gaps Identified : Transfer Reinforcement Learning in Differing Action Space

From our survey on transfer learning for reinforcement learning settings, we identify that most of the works assume a strong similarity among the domains. The limited works which address the differing state-action spaces use policy reuse methodology assuming availability of multiple source domains and expert policies at the disposal of the target domain [32, 33]. One of the recent works [34] have attempted using reward shaping techniques mention in Section 2.2.1 to address the differing action space, however the chosen evaluations are lower dimensional discrete observation space and in the continuous domain the results are not promising. One of the interesting works to transfer knowledge in differing action space is using actor-critic method [35] where apart from policy gradients, a mutual information between the source and target's latent representation is also used for optimisation.
We want to push the research in the direction of transferring knowledge between agents in differing state and particularly action spaces. Our motivation is that given a teacher policy for a locomotive humanoid robotic agent (let's say it has 100 dimension action space) comprising of actions like joint angular velocity, acceleration and so on. And in a manufacturing setting, the robotic agent malfunctions (or we have another less able agent) resulting in a smaller action space (let's say 80 dimensional or so). In such a scenario, we would like to leverage the prior experience since the the two domains share an inherent structure. We want t develop algorithms that can help jump start the target domain's learning with less interactions with the environment. We also want to focus on model free approaches and are motivated to analyse our performance formally using metrics reported in Table 1.

4. Proposed Methodology

In this section we are going to propose the methodology for the transfer learning from source to target domain when both the state and action space differs in both the domains. As discussed in the above sections, there are numerous work available where source domain is trained using many different scenarios at the same time [5, 28]. This makes the model generalization till one extent and hence helps in the transfer of the knowledge to the target domain
We want to take a step back and study if we can train the agent in the source domain on just one scenario and with more exploration using the methods in sections below. We are hypothesising that the agent which has explored more in the source domain can help the agent in the target domain to learn faster even when there is state-action mismatch.

Table 1. Formal Evaluation Metrics from definitions from literature [29]

heightMetrics	Definition	Context
Jumpstart Performance	the initial performance (returns) of the agent	Learning Process + Generalisation
Asymptotic Performance (AP)	the ultimate performance (returns) of the agent	Learning Process + Mastery
Accumulated Rewards (AR)	the area under the learning curve of the agent	Learning Process + Mastery
Time to Threshold (TT)	the learning time (iterations) needed for the target agent to reach certain performance threshold	Learning Process + Generalisation
Performance with fixed training epochs (PE)	the performance achieved by the target agent after a specific number of training iterations	Learning Process + Mastery
Performance Sensitivity (PS)	the variance in returns using different hyper-parameter settings	Learning Process + Generalisation
Necessary knowledge Amount (NKA)	the necessary amount of the knowledge required for TL in order to achieve certain performance thresholds	Transferred Knowledge + Generalisation
Necessary Knowledge Quality (NKQ)	the necessary quality of the knowledge required to enable effective TL	Transferred Knowledge + Generalisation

4.1. Exploration friendly Reinforcement Learning Methods

Exploration versus exploitation is an important topic in reinforcement learning. We want the agent to learn the optimum policy as fast as possible but at the same time we want it to gain the maximum possible knowledge about the environment. With the perspective of the transfer learning it is beneficial to maximize the knowledge about the environment because if the agent has more information about the environment dynamics in the source then it can be more useful in the target domain.
For the problem proposed in section 3, where the state space is the same but action space can vary; our hypothesis is that having the maximum knowledge about the environment can be helpful as the state space is the same. Therefore, in this section we discuss about the methods in the literature which incentives the agents when its knowledge of the environment increases.

4.1.1. Forward Dynamics Prediction Model

These models explore based on the prediction models i.e., a model is made to predict the consequences and if the error is high in the prediction then the agent is less familiar with that state. The action which reduces this error or which shows fastest error rate drop is chosen as the next action [36].

Intelligent Adaptive Curiosity (IAC):
This falls under the category of dynamic prediction model. In this architecture there are 2 models: (i) machine

learner **M** which predicts the consequence, and (ii) meta machine learner **metaM** which predicts the error in the error prediction. In the architecture there is a knowledge gain assessor which assesses the mean decline in the error rate prediction. If the decline is higher then the agent has gained more knowledge about the environment dynamics and based on that intrinsic reward is given to the agent [37].

Variational Information Maximizing Exploration (VIME):
This architecture also falls under dynamics prediction model. IAC is not suitable for continuous system but VIME is made to deal with the continuous systems. This method makes the agent to take the action which leads to the largest increase in the information gain or the largest decrease in the entropy [38].

4.1.2. Random Networks

Random networks methods are used for model-free approaches where the task is to learn random tasks rather than the environment dynamics.

Directed Outreaching Reinforcement Action-Selection (DORA):
This method depends on two parallel MDPs: (i) the original MDP of the system , and (ii) the other similar MDP with each state-action pair designed to have value 0. The Q-value learned for the second MDP is called as E-value and if the E-value is not predicted to be 0 then the model has still some missing information and it should continue the exploration [39].

Random Network Distillation (RND):
This method incentives the agent based on the error in the prediction of two neural networks: target and predictor network [40]. Target network is randomly assigned with the fixed weights in the beginning and they produce the embedding f. The predictor network is the one whose weights are not fixed and they produce the embedding $\bar{f}$. The weights of the predictor network are updated based on the error $E = ||f - \bar{f}||^2$. So the error predicted will be higher for the novel states as the predicted network would not have seen it before. Therefore to encourage exploration, the bonus reward is made to be proportional to the error predicted E.

4.2. Architecture

A proposed implementation of our hypothesis mentioned is shown in the Figure 1. First the agent in the source domain is trained during which apart from the actual reward it also receives reward for more exploration. This the novelty of our approach, where we aim to design a reward shaping technique to award a bonus for a given state-action pair. Then the source model is transferred to the target domain using representation transfer methods discussed Section 2.2.4. Since the state-action space is different between our source and target domains, we additionally learn a mapping function and reuse the source representation.
In the architecture, **NN1** and **NN2** are used to account for the dimensionality mismatch between source and target domain due to state-action dimensionality mismatch. The weights in these two networks change to learn the domain-specific dynamics of the target. The hypothesis is that in the beginning of the learning process the behaviour of the target agent will be very much guided by the frozen network from source and as the optimization progresses the agent will start developing more target domain specific knowledge. Note that such an approach results in a longer training time in source domain since we encourage exploraion. We believe this is a reasonable side-effect since, in most supervised learning cases it is assumed that the source domain has more samples available and we can extend the same argument to RL that we have the budget of one-time longer training if the knowledge can be effectively transferred.

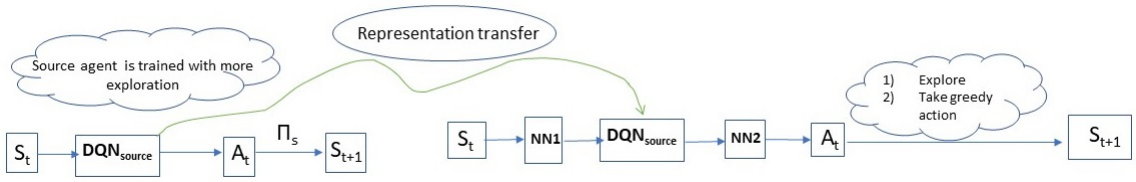

Fig. 1. Proposed architecture when action and state space is different in source and target domain

4.3. Proposed Evaluation Scheme

As the part of the future plan based on our literature survey and proposed architecture, plan is to take a fairly complicated environment from OpenAI[2] and firstly train the source agent which has explored a lot to gain environment knowledge. Then the source domain knowledge is transferred to target domain using the architecture discussed above. The efficiency of the transfer learning will be measured using the metrics discussed in Table 1.

5. Conclusion

In this survey we have reviewed the current research in the area of transfer reinforcement learning. There are two major drawbacks in most of the research in this area 1) lack of unified metrics for performance evaluation and 2) dependence on underlying RL algorithm. We identify that one of the gaps in the works on knowledge transfer in RL agents is that they are not evaluated for setting where the source and target MDPs vary in there action spaces. We highlight the importance of such a setting. We also propose a technique inspired from incentivized exploration for source agents to develop an algorithm to transfer knowledge between MDPs differing in state-action space. An accompanying video presentation can be accessed here[3].

References

1. R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction* (MIT press, 2018).
2. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602 (2013).
3. C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks,* (Springer, 2018), pp. 270–279.
4. Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *2010 IEEE computer society conference on computer vision and pattern recognition,* (IEEE, 2010), pp. 1855–1862.
5. E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," arXiv preprint arXiv:1412.3474 (2014).
6. M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* (2014), pp. 1717–1724.
7. Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning,* (PMLR, 2015), pp. 1180–1189.
8. Y. Gao, H. Xu, J. Lin, F. Yu, S. Levine, and T. Darrell, "Reinforcement learning from imperfect demonstrations," arXiv preprint arXiv:1802.05313 (2018).
9. S. Devlin, L. Yliniemi, D. Kudenko, and K. Tumer, "Potential-based difference rewards for multiagent reinforcement learning," in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems,* (2014), pp. 165–172.
10. Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," arXiv preprint arXiv:1808.04355 (2018).
11. L. Weng, "Exploration strategies in deep reinforcement learning," lilianweng.github.io (2020).
12. M. Klissarov and D. Precup, "Reward propagation using graph convolutional networks," Adv. Neural Inf. Process. Syst. **33**, 12895–12908 (2020).
13. S. Schaal, "Learning from demonstration," Adv. neural information processing systems **9** (1996).
14. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," nature **529**, 484–489 (2016).
15. X. Zhang and H. Ma, "Pretraining deep actor-critic reinforcement learning algorithms with expert demonstrations," arXiv preprint arXiv:1801.10459 (2018).
16. T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband *et al.*, "Deep q-learning from demonstrations," in *Proceedings of the AAAI Conference on Artificial Intelligence,* , vol. 32 (2018).
17. K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," arXiv preprint arXiv:1708.05866 (2017).
18. A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE international conference on robotics and automation (ICRA),* (IEEE, 2018), pp. 6292–6299.
19. I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson, "Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning," arXiv preprint arXiv:1809.02925 (2018).

[2] https://github.com/openai/gym
[3] https://youtu.be/bLRksZ-wD7A

20. C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE conference on computer vision and pattern recognition,* (IEEE, 2009), pp. 951–958.
21. G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531 **2** (2015).
22. S. M. Devlin and D. Kudenko, "Dynamic potential-based reward shaping," in *Proceedings of the 11th international conference on autonomous agents and multiagent systems,* (IFAAMAS, 2012), pp. 433–440.
23. M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel, "Continuous adaptation via meta-learning in nonstationary and competitive environments," arXiv preprint arXiv:1710.03641 (2017).
24. J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," arXiv preprint arXiv:1804.10332 (2018).
25. A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. P. van Hasselt, and D. Silver, "Successor features for transfer in reinforcement learning," Adv. neural information processing systems **30** (2017).
26. B. Huang, F. Feng, C. Lu, S. Magliacane, and K. Zhang, "Adarl: What, where, and how to adapt in transfer reinforcement learning," arXiv preprint arXiv:2107.02729 (2021).
27. A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," arXiv preprint arXiv:1606.04671 (2016).
28. T. D. Kulkarni, A. Saeedi, S. Gautam, and S. J. Gershman, "Deep successor reinforcement learning," arXiv preprint arXiv:1606.02396 (2016).
29. Z. Zhu, K. Lin, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," arXiv preprint arXiv:2009.07888 (2020).
30. A. Srinivas, M. Laskin, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," arXiv preprint arXiv:2004.04136 (2020).
31. H. Liu and P. Abbeel, "Behavior from the void: Unsupervised active pre-training," Adv. Neural Inf. Process. Syst. **34** (2021).
32. F. Fernández and M. Veloso, "Policy reuse for transfer learning across tasks with different state and action spaces," in *ICML Workshop on Structural Knowledge Transfer for Machine Learning,* (Citeseer, 2006).
33. Y. Heng, T. Yang, Y. ZHENG, H. Jianye, and M. E. Taylor, "Cross-domain adaptive transfer reinforcement learning based on state-action correspondence," in *The 38th Conference on Uncertainty in Artificial Intelligence,* (2022).
34. N. Beck, A. Rajasekharan, and T. H. Tran, "Transfer reinforcement learning for differing action spaces via q-network representations," arXiv preprint arXiv:2202.02442 (2022).
35. M. Wan, T. Gangwani, and J. Peng, "Mutual information based knowledge transfer under state-action dimension mismatch," arXiv preprint arXiv:2006.07041 (2020).
36. L. Wang, "Exploration strategies in deep reinforcement learning," in *https://lilianweng.github.io/posts/2020-06-07-exploration-drl/,* .
37. P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," IEEE Transactions on Evol. Comput. **11**, 265–286 (2007).
38. R. H. et.al., "Curiosity-driven exploration in deep reinforcement learning via bayesian neural networks," CoRR **abs/1605.09674** (2016).
39. L. Choshen, L. Fox, and Y. Loewenstein, "DORA the explorer: Directed outreaching reinforcement action-selection," CoRR **abs/1804.04012** (2018).
40. Y. B. et.al., "Exploration by random network distillation," CoRR **abs/1810.12894** (2018).