Learning Interaction-Based Representations for Reinforcement Learning Using Recursive State Similarity Metrics

Kevin Black, Caleb Chuck* and Scott Niekum*
The University of Texas at Austin

***Abstract*—We investigate the problem of learning meaningful state representations from interaction alone, without an extrinsic signal such as pixel reconstruction, reward, or expert behavior. Good representations should encode only "relevant" information in a generalizable way, with the goal of accelerating reinforcement learning on multiple downstream tasks. To this end, we introduce recursive state similarity metrics, a generalization of bisimulation metrics that can encode complex information about future dynamics. We present an algorithm for representation learning using these recursive state similarity metrics. We apply the algorithm to a learning objective based on inverse and forward dynamics modeling rather than an extrinsic signal. We then perform experiments in a 2D navigation environment, learning a representation from offline interaction data that improves sample efficiency on a downstream goal-reaching task regardless of the offline data quality. We also demonstrate the ability of the recursive state similarity metric to produce rich semantic representations that encode information about future interactions.**

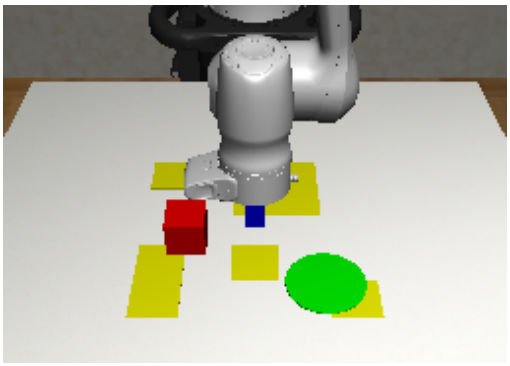

Fig. 1. An example block pushing robotic manipulation environment. A good representation should encode the pusher and the block, but not the yellow squares or green circle, which do not affect the motion of the block or the robot arm.

I. Introduction

Deep reinforcement learning (RL) has enjoyed enormous recent success in a wide variety of tasks. However, it still faces many barriers that forestall its application to certain real-life tasks. One such barrier is data efficiency. Many successful RL algorithms, especially those that learn from high-dimensional image observations, require millions of environment interactions before performing well. Another barrier is generalization. RL algorithms can overfit even to large datasets of different tasks, and rarely generalize well to tasks that they were not specifically trained on.

Representation learning is a general technique with the potential to solve both of these problems. In theory, a good representation should encode only aspects of the observation which are "relevant" to the agent, while ignoring any "irrelevant" distractors. While relevancy is often defined with respect to a particular task, in many real-world environments, there are intuitive notions of relevancy that generalize across tasks. For example, a good representation in a robot manipulation environment (Figure 1) would encode the configuration of the gripper and any manipulable objects, but not any background objects that are out of reach. Equipping an RL algorithm with such a representation would make learning easier and improve data efficiency on any tasks from the same environment.

*Not enrolled in CS391R

Traditional representation learning relies on reconstruction in the pixel space as the learning signal, which is easily susceptible to "irrelevant" distractors. For RL, alternative signals such as reward [21] and expert behavior [1] have found recent success; however, these signals may not always be accessible, and are fundamentally associated with a particular task. In this work, we aim to be as general as possible, learning a representation from pre-collected offline data which does not necessarily come from a particular policy nor contain task-specific annotations. Our methods fit into the existing paradigm of pre-training a model on a large and diverse dataset to enable data-efficient fine-tuning on downstream tasks; this paradigm has found enormous success in other fields such as computer vision [13] and natural language processing [8], and has more recently seen growing popularity in the field of robotics [17, 7].

The contributions of this work are as follows: 1) inspired by recent work using bisimulation metrics for representation learning [21], we propose the generalized idea of the recursive state similarity metric, as well as an algorithm using it for representation learning, 2) we apply this algorithm to an existing method based on inverse and forward dynamics modeling that can learn offline from task-agnostic environment interaction, 3) we apply our method to a 2D navigation environment and demonstrate better-than-baseline performance.

II. Related Work

This work builds heavily upon bisimulation metrics [10], which measure the similarity between states in a Markov decision process (MDP) based on their current and future reward. Zhang et al. [21] use bisimulation metrics to perform task-specific representation learning by matching ℓ_1 distance in the latent space with bisimulation distance. We also utilize this method of ℓ_1 distance matching. Agarwal et al. [1] extend the idea of bisimulation metrics to use expert behavior as the learning signal instead of reward, and additionally introduce a contrastive method for learning representations based on a similarity metric. Castro [5], Castro et al. [6] propose further techniques for effectively learning representations based on bisimulation metrics. One key difference between our method and prior work is that prior work has used an *on-policy* state similarity metric in practice, which is tied to a particular policy. Our method has no such restriction.

Unrelated to bisimulation, there is a large body of work regarding intrinsic motivation [4, 18] which often has a component that learns a representation without reward. Chebotar et al. [7] and Stooke et al. [19] also learn reward-free representations offline from demonstrations using various techniques. However, our proposed method based on state similarity metrics provides an orthogonal approach.

III. Preliminaries

Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma, s_0)$ define a Markov decision process (MDP), where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the transition dynamics, R is the reward function, γ is the discount factor, and s_0 is the initial state distribution. The goal of RL is to find a policy π that maximizes the cumulative expected return $\mathbb{E}_{\pi,P}\left[\sum_t \gamma^t R(s_t, a_t)\right]$ where $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$.

A bisimulation metric [10] is a pseudometric satisfying the following recursive equation for all $s, t \in \mathcal{S}$:

$$d(s,t) = \max_{a \in \mathcal{A}} \left[|R(s,a) - R(t,a)| + \gamma \mathcal{W}_1^d(P(\cdot|s,a), P(\cdot|t,a)\right]$$

where $\mathcal{W}_1^d$ is the 1-Wasserstein distance [20] on the metric itself, d. Intuitively, two states are bisimilar if, for all actions, they produce similar reward and transition to states that are also bisimilar.

IV. Methods

We propose a generalization of bisimulation metrics as follows:

$$d(s,t) = \Phi_{a \in \mathcal{A}} \left[C(s,t,a) + \gamma \mathcal{W}_1^d(P(\cdot|s,a), P(\cdot|t,a))\right] \quad (1)$$

Where Φ is any reduction over the actions (e.g. max, mean), and $C(s,t,a)$ is an existing distance function. Intuitively, this metric takes the information contained in C and "extends" it through time. $C(s,t,a)$ is the base case, and the Wasserstein distance is the recursive term that accounts for future states. The original bisimulation metric uses $\max$ as the reduction operator to account for the *worst-case* sequence of actions that maximize the difference between the two states; therefore, two states will be bisimilar if and only if they produce similar rewards for *all* action sequences. While this allows for some appealing properties in the case of bisimulation, such as the bounding of optimal value function differences [9], in general we may not care about *all* action sequences. Therefore, a different reduction operator may be desired, such as the mean or the expected value over some policy.

A. Learning The State Similarity Representation

Algorithm 1 Learning The State Similarity Representation

1: **for** Time $t = 0$ to ∞ **do**
2: Sample batch B from dataset
3: Train forward model $\hat{P}$ using $\phi_{\bar{\theta}}$
4: Train representation: $\mathbb{E}_{(s,t)\sim B\times B}[L_\theta(s,t)]$ ▷ Eq. (2)
5: Update target parameters: $\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}$
6: **end for**

Building upon Zhang et al. [21], a representation can be learned by encouraging ℓ_1 distances in the representation space to directly match the value of the recursive state similarity metric from equation (1). Let $\phi_\theta : \mathcal{S} \to \mathcal{Z}$ be an encoder with parameters θ mapping observed states to the latent representation space $\mathcal{Z}$. We draw batches of state pairs (s,t) and train the encoder with the following mean-squared error loss:

$$L_\theta(s,t) = \left(||\phi_\theta(s) - \phi_\theta(t)||_1 - \hat{d}(s,t)\right)^2 \quad (2)$$

where $\hat{d}$ is an approximation of equation (1), defined by:

$$\hat{d}(s,t) = \Phi_{a \in \mathcal{A}} \left[C(s,t,a) + \gamma \mathcal{W}_1^{||\cdot||_1}(\hat{P}(\cdot|\phi_{\bar{\theta}}(s), a), \hat{P}(\cdot|\phi_{\bar{\theta}}(t), a))\right] \quad (3)$$

where $\hat{P}$ is a forward dynamics model trained in the latent space. For learning stability, $\hat{P}$ is trained in the latent space of a target encoder $\phi_{\bar{\theta}}$, where $\bar{\theta}$ is an exponential moving average of the online encoder parameters θ as introduced in Mnih et al. [16]. The procedure for learning the state similarity representation is summarized in Algorithm 1.

For this work, we assume deterministic transition dynamics, so that the forward model is a deterministic function $\hat{P} : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$ and the Wasserstein distance in equation (3) reduces to $||\hat{P}(\phi_{\bar{\theta}}(s), a) - \hat{P}(\phi_{\bar{\theta}}(t), a)||_1$. However, stochastic transition dynamics can be handled as in Zhang et al. [21] by letting $\hat{P}$ output a Gaussian distribution and using a closed-form computation of the Wasserstein distance. We also assume a discrete action space, meaning that the inner terms of equation (3) can easily be evaluated on every action. An extension to continuous action spaces is left to future work.

B. Learning The Interaction-Based Representation

We want to choose a "base case" $C(s,t,a)$ that captures meaningful information about the environment. However, $C(s,t,a)$ does not need to capture information about future

timesteps, only the current one; the recursive term will extend this information forward through time. Drawing upon previous work [18, 2], we learn a $C(s, t, a)$ using another representation learning technique based on a joint inverse and forward dynamics objective.

Let ψ_μ be another encoder with parameters μ that maps states into another latent representation space. We draw batches of (state, action, next state) samples (s, a, s'), and train with the following loss:

$$\begin{aligned} L_\mu(s, a, s') = \lambda L_{inv}[a, \text{ID}(\psi_\mu(s), \psi_\mu(s'))] \\ + (1 - \lambda) L_{fwd}[\psi_\mu(s'), \text{FD}(\psi_\mu(s), a)] \end{aligned} \tag{4}$$

where ID is an inverse dynamics model trained to predict the action that caused the transition between consecutive encoded states, and FD is a forward dynamics model trained to predict the next encoded state from the previous encoded state and the action. In discrete environments, L_{inv} is the standard cross-entropy loss. L_{fwd} is the mean squared error. The models are trained jointly using a mixing coefficient λ.

Intuitively, the inverse dynamics objective forces the representation to encode information that is predictive of the agent's actions. The forward dynamics model regularizes the representation space and encourages it to discard irrelevant information that is not correlated with the agent's actions. However, the objectives only encourage the representation to capture meaningful information about the environment's dynamics in the *current timestep*, which is why applying the recursive state similarity metric on top of it is useful.

We extract a similarity metric from ψ_μ by defining $C(s, t, a) = ||\psi_\mu(s) - \psi_\mu(t)||_1$.

V. Experiments

A. Environment

All experiments are carried out on a simple 2D navigation environment (Figure 3). The agent, in green, can take 4 discrete actions (up, down, left, right). The goal is to reach the blue square, and the agent is unable to move through the red obstacles. The obstacles, agent, and goal locations are randomized at the start of every episode. States are 20×20 image observations. A reward of 0 is provided for reaching the goal square, and a reward of -1 is provided for every timestep that the goal square is not reached. Episodes are automatically terminated when the goal square is reached or after 50 timesteps.

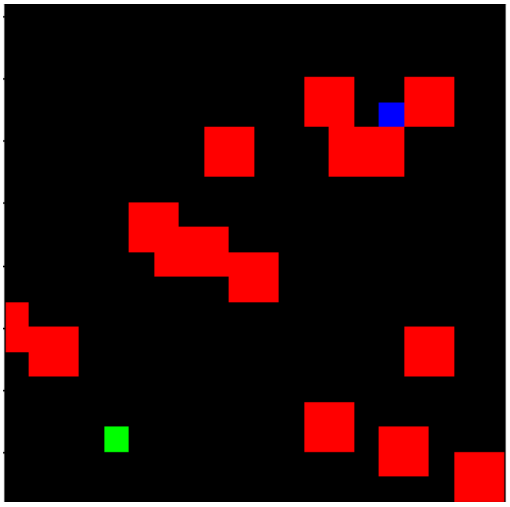

Fig. 3. An image observation from the 2D navigation environment.

Offline datasets are collected using an ϵ-greedy algorithm: with probability ϵ, the agent takes a uniformly random action, and with probability $1-\epsilon$, the agent takes a step along an optimal path toward the goal. Additionally, the goal channel (blue) is masked out in the offline datasets, preventing imitation learning and providing a rudimentary test of generalization: any algorithm trained on the offline datasets must learn from

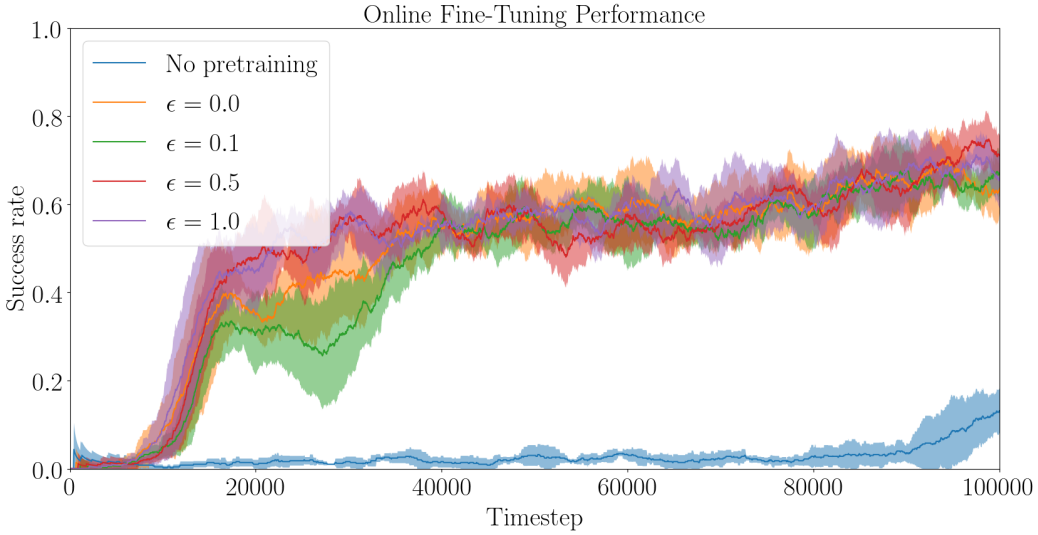

Fig. 2. Results of online fine-tuning on the 2D navigation environment. All runs are averaged over 5 seeds, with the standard deviation represented by the shaded area. A success is counted when the agent reaches the goal within 50 timesteps, and the success rate is a rolling average over the past 100 episodes.

the interaction of the agent with the obstacles alone.

B. Experimental Details

Both encoders, ϕ_θ and ψ_μ, are 6-layer convolutional networks with ReLU activations that induce a latent dimension of 128. As in CoordConv [14], two pixel coordinate channels are concatenated to the image observations before passing them through the network. The inverse dynamics model ID, as well as the forward dynamics models FD and $\hat{P}$, are all fully connected networks with one hidden layer of 256 neurons and ReLU activations. The state similarity components $(\phi_\theta, \hat{P})$ are trained with a learning rate of 0.0001, $\tau = 0.005$, $\gamma = 0.99$, and $\Phi = \text{mean}$. The interaction-based components $(\psi_\mu, \text{ID}, \text{FD})$ are trained with a learning rate of 0.001 and $\lambda = 0.2$. Both components are optimized using Adam [12].

Datasets of 1,000,000 transitions are collected at various values of ϵ. The state similarity component and the interaction-based component are trained in tandem on the datasets with a batch size of 128 for 30 epochs. Then, the representation from the state similarity component is evaluated by performing 100,000 steps of online learning using double deep Q-learning [11] with hindsight experience replay [3], where the Q-network is initialized with the parameters from the state simliarity encoder ϕ_θ. For detailed hyperparameters for the online learning phase, see the code repository[1].

C. Results Discussion

Figure 2 shows the results of online fine-tuning in the 2D navigation environment. The representation learning method clearly outperforms the "no pretraining" baseline at all values of ϵ. This is somewhat surprising because $\epsilon = 0$ represents data collected from a perfectly optimal policy that contains no obstacle collisions, meaning it should be more difficult to learn a meaningful representation. We hypothesize that our method can implicitly learn from expert behavior: the optimal policy always navigates around obstacles, so the representation still learns obstacle information. However, a more detailed investigation is left to future work.

It does not appear in the chart, but an ablation was done where the Q-network was initialized with the parameters from ψ_μ, without the recursive state similarity component. This initialization performed similarly to the "no pretraining" baseline.

D. Visualizing Representations

We perform several visualizations to gain an intuitive understanding of the learned representations. Figure 4 compares obstacle attention between the interaction-based representation alone and the full recursive state similarity representation. As expected, the interaction-based representation mostly attends to information relevant in the *current* timestep — in this case, whether or not there is an obstacle right next to the agent — since this is the only information required for inverse and forward dynamics. The recursive state similarity representation extends this information through time, accounting for obstacles that may be encountered in future timesteps. However, the attention fades as the distance from the agent increases, due to the discount factor $\gamma = 0.99$.

[1]https://github.com/kvablack/nav2d-representation

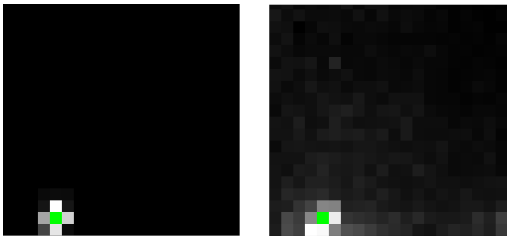

Fig. 4. A visualization of obstacle attention for the interaction-based representation ψ_μ (left) and the full recursive state similarity representation ϕ_θ (right). The agent is the green pixel, and the intensity of all other pixels is proportional to the amount that the representation changes (measured by ℓ_1 distance) when an obstacle is placed at that pixel.

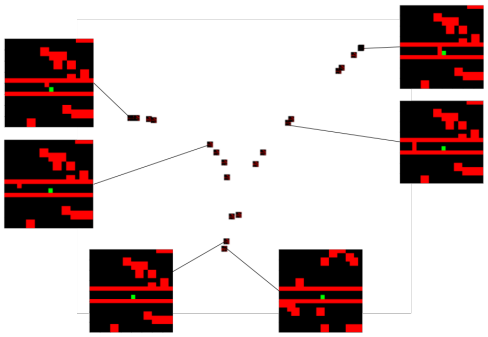

Fig. 5. T-SNE projection of the state similarity representation in the corridor environment. States are close together (bottom) when only the unreachable obstacles outside of the corridor change. States move upward along the left "branch" when the corridor is partially blocked, with the states being further away the closer the obstacle is to the agent. States move upward along the right "branch" when the corridor is fully blocked.

Figure 5 visualizes the recursive state similarity representation in a version of the 2D navigation environment where the agent is stuck in a fixed corridor. The representation space is structured intuitively, producing different "branches" depending on whether the corridor is partially or fully blocked, as this represents fundamentally different future dynamics. Unreachable obstacles outside the corridor do not affect the representation at all.

VI. Conclusion

In this paper, we extended prior work in bisimulation metrics to develop a generalized recursive state similarity metric along with an algorithm leveraging it to learn state representations. We applied this algorithm on top of an existing interaction-based representation learning technique that uses a joint inverse and forward dynamics objective to learn from environment interaction alone without any extrinsic signal such as reward or expert behavior. We quantitatively evaluated this method on a simple 2D navigation environment, demonstrating better-than-baseline performance. Additionally, we qualitatively demonstrated the ability of the state similarity metric to produce representations that encode more

meaningful, longer-horizon information about objects in the environment compared to the interaction-based representation alone.

While the results so far are promising, more work needs to be done to validate the efficacy of this method. Firstly, it should be compared against other baselines in offline pretraining for RL, such as actionable models [7]. Secondly, it needs to be applied to more complex, real-world tasks. Robotic manipulation and autonomous driving are good candidate domains for this work, as they can both benefit greatly from the task-agnostic offline pretraining paradigm, and they also exhibit similar interaction-based dynamics as the simple 2D navigation environment. For example, it is easy to acquire a lot of task-agnostic "play" data [15] from a general-purpose robot arm, which is rich with the kind of interaction information that is essential for downstream tasks. However, applying recursive state similarity metrics to these domains will require eliminating the assumption of deterministic transition dynamics as well as extending it to continuous action spaces.

There are many directions for future work. Aside from applying this method to more complex domains, the formulation of the recursive state similarity metric is quite general in that it can use any base case $C(s, t, a)$. There may be existing methods that, when used as a base case for the recursive state similarity metric, lead to much more expressive and useful representations than the base case used in this paper. Alternatively, the base case used in this paper has been applied to intrinsic motivation [18], so perhaps adding the recursive state similarity metric on top of it would provide new benefits in that line of work.

Author Information

Kevin Black performed all coding and paper-writing. Caleb Chuck and Scott Niekum contributed in an advisory capacity, assisting with the discussion of ideas.

References

[1] Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qda7-sVg84.

[2] Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 5092–5100, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

[3] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/453fadbd8a1a3af50a9df4df899537b5-Paper.pdf.

[4] Arthur Aubret, Laetitia Matignon, and Salima Hassas. A survey on intrinsic motivation in reinforcement learning, 2019.

[5] Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 10069–10076. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6564.

[6] Pablo Samuel Castro, Tyler Kastner, P. Panangaden, and Mark Rowland. Mico: Learning improved representations via sampling-based state similarity for markov decision processes. *ArXiv*, abs/2106.08229, 2021.

[7] Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jacob Varley, Alex Irpan, Benjamin Eysenbach, Ryan C Julian, Chelsea Finn, and Sergey Levine. Actionable models: Unsupervised offline reinforcement learning of robotic skills. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1518–1528. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/chebotar21a.html.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[9] Norm Ferns and Doina Precup. Bisimulation metrics are optimal value functions. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, page 210–219, Arlington, Virginia, USA, 2014. AUAI Press. ISBN 9780974903910.

[10] Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, page 162–169, Arlington, Virginia, USA, 2004. AUAI Press. ISBN 0974903906.

[11] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2094–2100. AAAI Press, 2016.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[14] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9628–9639, Red Hook, NY, USA, 2018. Curran Associates Inc.

[15] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. *Conference on Robot Learning (CoRL)*, 2019. URL https://arxiv.org/abs/1903.01973.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, Feb 2015. ISSN 1476-4687. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

[17] Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. {AWAC}: Accelerating online reinforcement learning with offline datasets, 2021. URL https://openreview.net/forum?id=OJiM1R3jAtZ.

[18] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 2778–2787. JMLR.org, 2017.

[19] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9870–9879. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/stooke21a.html.

[20] Cédric Villani. *Optimal transport: old and new*. Springer, 2008.

[21] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.