Behavior Predictive Representations for Generalization in Reinforcement Learning

Siddhant Agarwal
IIT Kharagpur

Aaron Courville
MILA, Université de Montréal

Rishabh Agarwal
Google Research, Brain Team
MILA, Université de Montréal

Abstract

Deep reinforcement learning (RL) agents trained on a few environments, often struggle to generalize on unseen environments, even when such environments are semantically equivalent to training environments. Such agents learn representations that overfit the characteristics of the training environments. We posit that generalization can be improved by assigning similar representations to scenarios with similar sequences of long-term optimal behavior. To do so, we propose behavior predictive representations (BPR) that capture long-term optimal behavior. BPR trains an agent to predict latent state representations multiple steps into the future such that these representations can predict the optimal behavior at the future steps. We demonstrate that BPR provides large gains on a jumping task from pixels, a problem designed to test generalization.

1 Introduction

Deep reinforcement learning (RL) agents, even when trained on diverse environments with similar high level goals but different dynamics and visual appearances, often struggle to generalize on unseen environments, even when such environments are semantically equivalent to training environments [Farebrother et al., 2018, Cobbe et al., 2020, Agarwal et al., 2021a, Packer et al., 2018]. Such agents learn state representations from high-dimensional observations that typically overfit to the peculiarities of training environments [Song et al., 2019, Raileanu and Fergus, 2021] rather than capturing generalizable skills which can be transferred to unseen environments. Such overfitting hinders the real-world applicability of RL, making generalization in RL an important challenge.

To improve generalization using better representations, we revisit predictive representations [Littman et al., 2001, Rafols et al., 2005] that describe the environment in terms of predictions about future observations, such as representations that encode the underlying environments dynamics. While learning such temporally predictive representations has been shown to improve sample efficiency [Oord et al., 2018, Schwarzer et al., 2021] within a training environment, it is unclear whether such representations would improve performance in unseen environments. More recently, Agarwal et al. [2021a] enhance generalization by learning similar state representations for observations with similar long-term optimal behavior. Inspired by their findings, we posit that predictive representations that capture long-term optimal behavior might be better suited for generalization. We expect such *behavior predictive representations* to generalize as two observations, possibly across different environments, are assigned similar representations if they exhibit similar sequences of optimal behaviors, irrespective of their differences in obtained rewards, visual appearances, or even the underlying dynamics.

For learning behavior predictive representations (BPR), we train the agent to predict latent state representations multiple steps into the future such that these representations can predict the optimal behavior at the future steps (Figure 1). BPR can be viewed as a representation learning approach where the agent predicts the optimal behavior at future states resulting from following a sequence of actions from a given state. We show the efficacy of BPR on the jumping task on pixels and

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

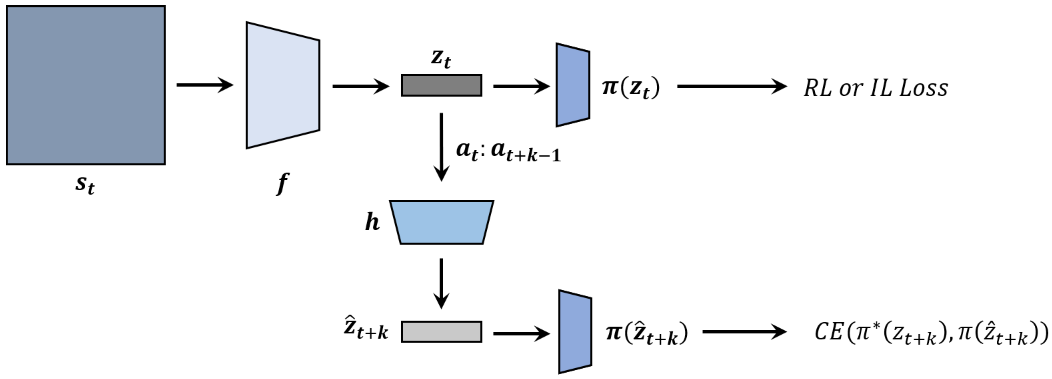


Figure 1: **Behavior Predictive Representations**. A schematic diagram showing how behavior predictive representations are learned using an auxiliary task on training environments. Representations z_t from the policy network are trained to predict the optimal behavior using either a reinforcement learning (RL) or imitation learning (IL) loss. These representations z_t, in conjunction with actions $a_t, a_{t+1}, \cdots, a_{t+k-1}$, are also trained to predicting latent representations $\hat{z}_{t+k}$ via the transition model h such that the $\hat{z}_{t+k}$ can predict the optimal behavior $\pi^*(z_{t+k})$ at time step $t + k$.

show it improves generalization upon existing methods including PSEs [Agarwal et al., 2021a] and SPR [Schwarzer et al., 2021]. We also provide ablations demonstrating the effect of predicting suboptimal policies as well as the horizon for predicting future behavior.

2 Preliminaries

We describe an environment as a Markov decision process (MDP) that corresponds to a tuple $\mathcal{M} = (S, A, P, R, \gamma)$ where S is the state space, A is the action space, $P : S \times A \times S \to [0, 1]$ is the state transition function, $R : S \times A \to \mathbb{R}$ is the reward function and $\gamma \in [0, 1]$ is the discount factor. A policy $\pi(\cdot|s)$ maps a state $s \in S$ to a distribution over the action space A. A trajectory is defined as the sequence of states, actions and corresponding rewards i.e. $s_0, a_0, r_1, s_1, \cdots$. The goal of a reinforcement learning agent is to maximize the cumulative expected return $\mathbb{E}_{p(\tau)}[\sum_t \gamma^t r(s_t, a_t)]$ where $p(\tau) = p(s_0) \prod_t p(s_{t+1}|s_t, a_t)\pi(a_t|s_t)$.

3 Behavior Predictive Representations

In this work, we aim to learn a policy that can generalize across related environments. Specifically, we train an agent using a finite number of environments (or tasks) sampled from a distribution of environments. The performance of this agent is evaluated using unseen environments sampled from the same distribution. For example, consider the generalization problem in a jumping task from pixels [Tachet des Combes et al., 2018], where an agent needs to jump over an obstacle (Figure 2). Standard deep RL agents trained on a small number of training tasks with different obstacle positions struggle to generalize to unseen obstacle positions [Agarwal et al., 2021a].

Inspired from the recent success of representation learning to improve generalization [Agarwal et al., 2021a, Raileanu and Fergus, 2021, Zhang et al., 2020], we also focus on learning better representations to improve generalization. We posit that learning better representations requires understanding which states are similar in terms of their long-term optimal behavior. To do so, we aim to learn latent state representations that not only capture the behaviour at the current state but will also be able to predict the behaviour at future states, which we call *behavior predictive representations* (BPR). Since BPR simply uses an auxiliary objective L^{BPR}, it can be easily combined with any RL or imitation learning setup, as shown in Figure 1.

To describe the auxiliary loss L^{BPR} for predicting the long-term optimal behavior, we define some notation first. Let s_t be the state at the time step t and z_t be the corresponding latent representation learned by the policy network f. The policy π predicts the action distribution given the latent representation z_t. We use an encoder network f to generate these latent representations from states as, $z_t = f(s_t)$. A transition function $h : S^* \times A \to S^*$ learns the state dynamics and predicts the representations at the next step, $\hat{z}_{t+1} = h(s_t, a_t)$.

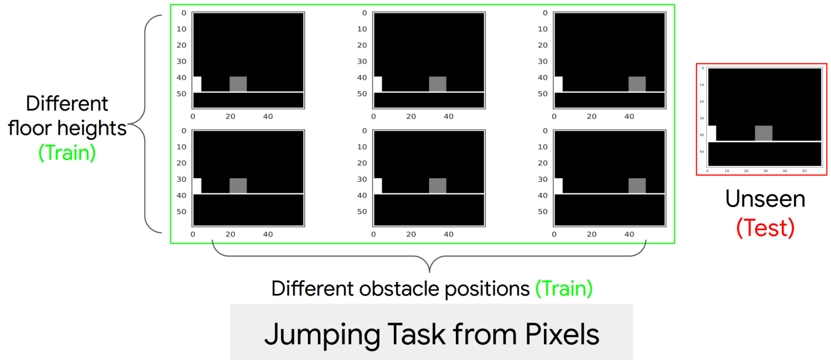


Figure 2: **Generalization on Jumping Task**. In this task, the agent needs to jump over an obstacle. The agent needs to time the jump precisely, at a specific distance from the obstacle, otherwise it will eventually hit the obstacle. Training environments consists of different obstacle positions as well as floor heights. At test time, the agent needs to generalize to environments with unseen positions and heights. The obstacle can be in 26 different locations while the floor has 11 different heights, totaling 286 environments.

We predict the representations for K future steps by iteratively applying the transition function. While such latent state dynamics are typically learned by minimizing the mean squared loss between $\hat{z}_{t+k}$ and z_{t+k}, we instead use these predicted representations to predict the optimal action distributions in the future steps $t+1$ to $t+k$. To do so, the agent minimizes the cross entropy between the predicted action distribution and optimal action distributions at these steps. Specifically, given access to the optimal policy π^* on training environments, the auxiliary loss L^{BPR} is given by:

$$L^{BPR} = \sum_{k=1}^{K} L^{CE}(\pi^*(z_{t+k}), \pi(\hat{z}_{t+k})), \tag{1}$$

where $L^{CE}(\pi_1(\cdot|s), \pi_2(\cdot|s)) = -\sum_{a \in A} \pi_1(a|s) \log \pi_2(a|s)$.

In the general RL setting, where we do not have access to the optimal policy, we propose to use the learned policy on training environments for specifying the future behavior in the objective L^{BPR}. Specifically, the target distribution for the auxiliary cross entropy loss, L^{BPR}, comes from the same policy network that is being trained (f in Figure 1). To provide some stability from the continuous changes in the learned policy, we use a separate *target* policy network that is periodically updated with the learned policy network parameters, analogous to deep Q-learning [Mnih et al., 2013] and self-supervised learning [Grill et al., 2020]. So, the target representations $\hat{z}_{t+k}$ are derived form the learned transition function h while the action distribution $\pi_{\text{learned}}(z_{t+k})$ comes from the target policy network. For this setting, the auxiliary loss $\hat{L}^{BPR}$ is given by,

$$\hat{L}^{BPR} = \sum_{k=1}^{K} L^{CE}(\pi_{\text{learned}}(z_{t+k}), \pi(\hat{z}_{t+k})) \tag{2}$$

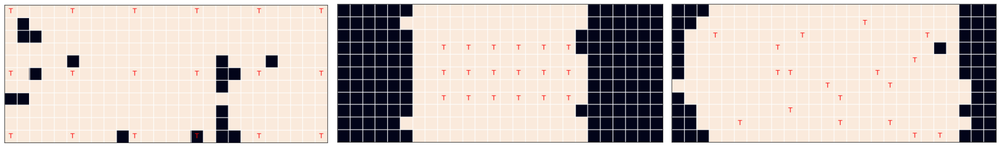
The final training objective is the combination of both of these loss functions. Let L^{RL} be the traditional model-free RL (or imitaion learning) objective. Then, the combined loss function for learning behavior predictive representations is $L = L^{RL} + \lambda_{BPR} L^{BPR}$, where λ_{BPR} is the weighting coefficient for the auxiliary loss.

4 Experiments

We first thoroughly investigate behavior predictive representations (BPR) on the jumping task [Tachet des Combes et al., 2018, Agarwal et al., 2021a] that captures whether agents can learn the correct invariances for generalization directly from image inputs.

4.1 Jumping Task From Pixels

Task Description. The task consists of an agent trying to jump over an obstacle using two actions: *right* and *jump*. Different tasks consist in shifting the floor height and/or the obstacle position (Fig-



(a) “Wide” grid (b) “Narrow” grid (c) Random grid

Figure 3: **Jumping Task**: Visualization of average performance of BPR with data augmentation across different configurations. We plot the median performance across 25 runs. Each tile in the grid represents a different task (obstacle position/floor height combination). For each grid configuration, the height varies along the y-axis (11 heights) while the obstacle position varies along the x-axis (26 locations). The red letter $\top$ indicates the training tasks. Random grid depicts only one instance, each run consisted of a different test/train split. Beige tiles are tasks BPR solved while **black** tiles are tasks BPR did not solve.

ure 2). To generalize, the agent needs to be invariant to the floor height while jump based on the obstacle position.

Problem Setup. Following Agarwal et al. [2021a], we use three different configurations (Figure 3), each consisting of 18 seen (training) and 268 unseen (test) tasks, to test generalization in regimes without and with data augmentation using RandConv [Lee et al., 2020]. As discussed by Agarwal et al. [2021a], the different grids configurations capture different types of generalization: the “wide” grid tests generalization via “interpolation”, the “narrow” grid tests out-of-distribution generalization via “extrapolation”, and the random grid instances evaluate generalization similar to supervised learning where train and test samples are drawn i.i.d. from the same distribution. Refer to Agarwal et al. [2021a] for more more experimental details.

Baselines. We compare the efficacy of our method with a number of techniques that have been used to achieve generalization including regularization such as ℓ_2-regularization and dropout [Farebrother et al., 2018] and data augmentation [Lee et al., 2020].

Policy Similarity Embeddings (PSEs) [Agarwal et al., 2021a] are the state-of-the-art generalization method on the jumping task. PSEs form an important baseline for BPR as PSEs also use the future behaviour as a similarity metric between states. Specifically, PSEs learn contrastive metric embeddings using a policy similarity metric d (Equation 3) that uses policy to measure the long term behavior similarity between among states.

$$d(x, y) = DIST(\pi^*(x), \pi^*(y)) + \gamma W_1(d)(p_{\pi^*}(\cdot|x), p_{\pi^*}(\cdot|y)) \tag{3}$$

Self-predictive representations (SPR) [Schwarzer et al., 2021] is another relevant baseline which has been shown to improve sample-efficiency on training environments on the Atari 100k benchmark [Kaiser et al., 2019, Agarwal et al., 2021b]. SPR’s objective is that the agent learns to predict its own latent representations at future steps. Similar to BPR, it uses a transition function to iteratively generate these latent representations for the future steps. However, while BPR optimizes the latent representations to predict future behavior, SPR tries to maximize the similarity between the predicted latent representations $\hat{z}_{t+1} : \hat{z}_{t+K}$ with the true future state representations $z_{t+1} : z_{t+K}$. To do so, SPR uses a self-supervised learning objective [Grill et al., 2020] as the auxiliary loss,

$$L^{SPR}(s_t : s_{t+k}, a_t : a_{t+k}) = -\sum_{k=1}^{K} \Big(\frac{q(g_o(\hat{z}_{t+k}))}{||q(g_o(\hat{z}_{t+k}))||_2}\Big)^T \Big(\frac{g_m(z_{t+k})}{||g_m(z_{t+k})||_2}\Big) \tag{4}$$

where g_o, g_m and q are online projection network, target projection network and prediction networks respectively. SPR linearly combines the auxiliary objective, L^{SPR}, with the RL objective.

Results. Table 1 summarizes the performance of BPR and all the baselines with and without data augmentation. Without data augmentation, with only 18 training environments, BPR generalizes quite well in all the three grid configurations, significantly outperforming regularization and PSEs by a large margin. These results exhibit that BPR is effective even without data augmentation.

Data augmentation complements all the methods and boosts generalization performance. Comparing RandConv + BPR to RandConv, we see that BPR is much more effective on top of RandConv. Moreover, when used in conjunction with data augmentation, BPR performs comparably to the current state-of-the-art method PSEs. Compared to BPR, SPR degrades the generalization performance significantly and even performs poorly than simply using RandConv. We hypothesize

Table 1: Percentage (%) of test tasks solved by different methods without and with data augmentation. The "wide", "narrow", and random grids are described in Figure 2. For methods implemented in this work (BPR and SPR), we report average performance across 25 runs with different random initializations, with standard deviation between parentheses. Other results are taken from Agarwal et al. [2021a].

Data Augmentation	Method	Grid Configuration (%)		
		"Wide"	"Narrow"	Random
No	Dropout and ℓ_2 reg.	17.8 (2.2)	10.2 (4.6)	9.3 (5.4)
	PSEs	33.6 (10.0)	9.3 (5.3)	37.7 (10.4)
	BPR	**62.4** (18.6)	**15.3** (6.7)	**58.5** (20.0)
Yes	RandConv	50.7 (24.2)	33.7 (11.8)	71.3 (15.6)
	RandConv + SPR	23.3 (11.8)	30.6 (13.3)	64.1 (15.6)
	RandConv + PSEs	**87.0** (10.1)	**52.4** (5.8)	**83.4** (10.1)
	RandConv + BPR	**90.0** (18.6)	**52.0** (9.4)	**82.5** (15.1)

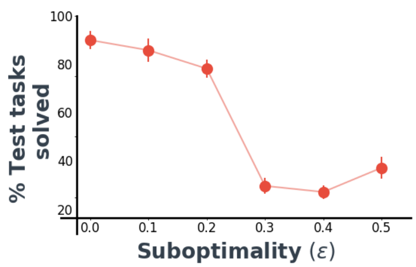


Figure 4: Percentage (%) of test tasks solved by BPR using ϵ-suboptimal policies on the "wide" configuration. We report the mean across 25 runs. Error bars show the standard error in mean results.

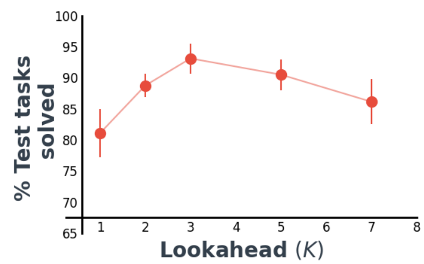


Figure 5: Percentage (%) of test tasks solved by BPR for different lookahead K on "wide" configuration. We report the mean across 25 runs. Error bars show the standard error in mean results.

that the self-supervised learning objective in SPR might be exacerbating the overfitting in learned representations by trying to predict the spurious features captured by the learned representations on training environments.

4.2 Effect of Policy Suboptimality on BPR

On the jumping task, we use the optimal policy on training environments to learn BPR. To understand the dependence of BPR on the optimal policy, we utilize -suboptimal policies to the auxiliary loss during training. Specifically, the prediction at future steps predicts the optimal action with probability $1-\epsilon$ and suboptimal action with probability ϵ.

We plot the performance of BPR on the "wide" configuration for the degree of suboptimality specified by ϵ, starting with the optimal policy ($\epsilon = 0$) to a uniform random policy ($\epsilon = 0.5$). It can be seen from Figure 4 that as ϵ increases, the performance decreases which is expected. For small values of ϵ, *i.e.*, $<= 0.2$, the performance decreases slightly but for larger values, the performance decreases sharply. This means that BPR is tolerant to certain levels of suboptimality.

4.3 Effect of look ahead on BPR

The lookahead of the agent is the number of future steps K for which the optimal action is to be predicted by latent representations. Greater the value of K, latent representations are required to predict actions on further in the future and possibly improve their generalizability. But it will be difficult for the latent representation to predict actions for steps that are far from the current step. Thus the performance drops for larger values of K.

Figure 5 shows the plot of the performance of BPR on the "wide" configuration with increase the value of K. The performance of BPR is good for even low values of K and it remains similar for

$K = 1$ to 7. But it increases slightly from $K = 1$to 3 and then decreases thereafter. As a result, we use $K = 3$ for all the results that we have reported so far.

5 Conclusion and Future Work

In this paper, we introduced Behavior Predictive Representations for improving generalization in reinforcement Learning. We show that predicting optimal actions at future steps is more beneficial than dynamics prediction or predicting future latent state representations. As seen from the results, BPR also performs well without data augmentations.

We plan to extend our work to more complex environments designed for testing generalization such as the Procgen benchmark [Cobbe et al., 2019] and Distracting DM Control Suite [Stone et al., 2021]. In these environments, we do not have access to the optimal policies and thus we plan to build BPR on top of RL objectives and show the effectiveness of BPR in such settings.

References

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*, 2021a.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *arXiv preprint arXiv:2108.13264*, 2021b.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. *arXiv preprint arXiv:1912.01588*, 2019.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pages 2048–2056, 2020.

Jesse Farebrother, Marlos C Machado, and Michael Bowling. Generalization and regularization in dqn. *arXiv preprint arXiv:1810.00123*, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. In *The International Conference on Learning Representations (ICLR)*, 2020.

Michael L Littman, Richard S Sutton, and Satinder P Singh. Predictive representations of state. In *Neural Information Processing Systems*, 2001.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

Eddie J Rafols, Mark B Ring, Richard S Sutton, and Brian Tanner. Using predictive representations to improve generalization in reinforcement learning. In *IJCAI*, 2005.

Roberta Raileanu and Rob Fergus. Decoupling value and policy for generalization in reinforcement learning. *International conference on machine learning*, 2021.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with momentum predictive representations. 2021.

Xingyou Song, Yiding Jiang, Yilun Du, and Behnam Neyshabur. Observational overfitting in reinforcement learning. In *The International Conference on Learning Representations (ICLR)*, 2019.

Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. The distracting control suite – a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

Remi Tachet des Combes, Philip Bachman, and Harm van Seijen. Learning invariances for policy generalization. In *Workshop track at the International Conference on Learning Representations (ICLR)*, 2018.

Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.