Generalization in Deep Reinforcement Learning

Sam Witty[1], Jun Ki Lee[2], Emma Tosch[1], Akanksha Atrey[1], Michael Littman[2], David Jensen[1]
[1]University of Massachusetts Amherst, [2]Brown University

Abstract

Observations of trained deep reinforcement-learning agents give the impression that agents have constructed a generalized representation that supports insightful action decisions. We re-examine what is meant by generalization in RL, and propose several definitions and techniques based on an agent's performance in on-policy, off-policy, and unreachable states. We demonstrate these techniques on a common benchmark task for deep RL, and show that more experimentation and analysis is necessary before claims of generalization can be supported.

1 Introduction

Deep reinforcement learning (RL) has produced agents that can perform complex tasks using only pixel-level visual input data. Given the apparent competence of some of these agents, it is tempting to see them as possessing a deep understanding of their environments. Unfortunately, this intuition can be misguided, leading to overconfidence in an agent's abilities.

We propose a view of generalization in RL based on an agent's performance in states it *couldn't* have encountered during training, yet that only differ from on-policy states in minor ways. The intuition is simple: To understand how an agent will perform across parts of the state space it could easily encounter and should be able to handle, expose it to states it could never have observed and measure its performance. Agents that perform well under this notion of generalization could be rightfully viewed as having mastered their environment. In this work, we make the following contributions:

Recasting generalization. We define a view of generalization for value-based RL agents, based on an agent's performance in on-policy, off-policy, and unreachable states. We discuss some of the undesirable implications of alternative definitions for generalization in RL, including those actively used in the literature. We do so by establishing a correspondence between the well-understood notions of interpolation and extrapolation in prediction tasks with off-policy and unreachable states in RL.

Analysis case-study. We demonstrate these techniques on a custom implementation of a common benchmark task for deep RL, the Atari 2600 game of AMIDAR. Our version, INTERVENIDAR, is fully parameterized, allowing us to manipulate the game's latent state, thus enabling an unprecedented set of experiments on a state-of-the-art deep Q-network architecture. We provide evidence that DQNs trained on pixel-level input can fail to generalize in the presence of non-adversarial, semantically meaningful, and plausible changes in an environment.

Related Work in Generalization. Generalization has long been a concern in RL [7]; the perspective we take is most related to sampling from diverse environments [8, 9]. Other work has focused on generalization as improved performance in off-policy states [6]. Techniques such as adding stochasticity to the policy [1], having the agent take random steps, no-ops, steps from human play [5], or probabilistically repeating the agent's previous action [4], all force the agent to transition to off-policy states. These existing methods diversify the training data via exposure to on-policy and off-policy states, but none discuss generalization over states that are logically plausible but unreachable.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

2 Recasting Generalization

Using existing notions of generalization in machine learning, such as held-out set performance, is complicated when applied to RL for two reasons: (1) training data is dependent on the agent's policy; and (2) the vastness of the state space in real-world applications means it is likely for novel states to be encountered at deployment time.

One could imagine a procedure in RL that directly mimics evaluation on held-out samples by omitting some subset of training data from any learning steps. However, this methodology only evaluates the ability of a model to *use* data after it is collected, and ignores the effect of exploration on generalization. Using this definition, we could incorrectly claim that an agent has learned a general policy, even if this policy performs well on a very small subset of states. Instead, we focus on a definition that encapsulates the trained agent as a standalone entity, agnostic to the specific data it encountered during training.

Generalization via State-Space Partitioning. We partition the universe of possible input states to a trained agent into three sets, according to how the agent can encounter them following its learned policy π from initial states $s_0 \in S_0 \subset S$.[1] Here, Π is the set of all policy functions, and α, δ, and β are some small positive values close to 0. We can think of δ and β as thresholds on estimation accuracy and optimality performance. These definitions assume that the agent explicitly produces an estimate, $\hat{q}(s,a)$, of the actual state-action value, $q_\pi(s,a)$, given a policy π which is upper-bounded by the optimal state-action value, $q^*(s,a)$. Given these, comparable quanities can be computed for the estimated, actual, and optimal state value, $\hat{v}(s)$, $v_\pi(s)$, and $v^*(s)$ respectively. The set of reachable states, $S_{\text{reachable}}$, is the set of states that an agent encounters with probability greater than α by following any policy $\pi \in \Pi$.

Definition 1 (Repetition)**.** An RL agent has high repetition performance, G_R, if $\delta > |\hat{v}(s) - v_\pi(s)|$ and $\beta > v^*(s) - v_\pi(s), \forall s \in S_{\text{on}}$. The set of on-policy states, S_{on}, is the set of states that the agent encounters with probability greater than α by following π from $s_0 \in S_0$.[2]

Definition 2 (Interpolation)**.** An RL agent has high interpolation performance, G_I, if $\delta > |\hat{q}(s,a) - q_\pi(s,a)|$ and $\beta > q^*(s,a) - q_\pi(s,a), \forall s \in S_{\text{off}}, a \in A$. The set of off-policy states, S_{off}, is defined as $S_{\text{reachable}} \setminus S_{\text{on}}$.

Definition 3 (Extrapolation)**.** An RL agent has high extrapolation generalization, G_E, if $\delta > |\hat{q}(s,a) - q_\pi(s,a)|$ and $\beta > q^*(s,a) - q_\pi(s,a), \forall s \in S_{\text{unreachable}}, a \in A$. The set of unreachable states, $S_{\text{unreachable}}$, is defined as $S \setminus S_{\text{reachable}}$.

Note that a large body of work implicitly uses G_R as a criteria for performance, even though this is the weakest of generalization capabilities. It is what you get when testing a learned policy in the environment in which it was trained. Some readers may doubt that it is possible to learn policies that extrapolate well. However, it has been shown that, with an appropriate representation, reinforcement learning can produce policies that extrapolate well under similar conditions to what we describe in this paper [3]. What has not been shown to date is that deep RL agents can learn policies that generalize well from pixel-level input.

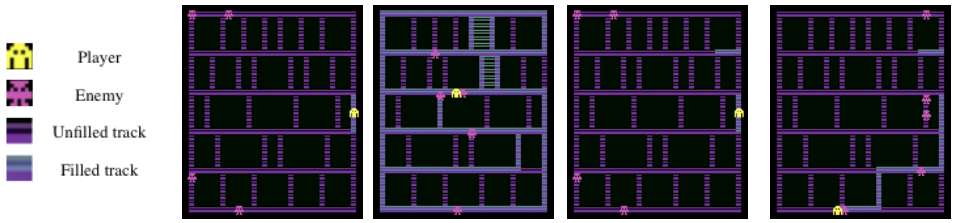
Consider an agent with tabular q-values, i.e. $\hat{q}(s,a)$ is represented as a table of size $|S| \times |A|$. Given an adequate exploration strategy, this agent could conceivably visit every reachable state during training, resulting in $\hat{v}(s)$ converging to $v^*(s), \forall s \in S_{\text{reachable}}$. This agent would satisfy G_R and G_I for arbitrarily small values of δ and β. Despite this positive outcome, most observers would not say that this agent "generalizes", because it lacks any function-approximation method. Only the definition G_E is consistent with this conclusion.

3 Analysis Case Study: AMIDAR

AMIDAR is a Pac-Man-like video game wherein an agent moves a player around a two-dimensional grid, accumulating reward for each vertical and horizontal line segment the first time that the player

[1] We use the standard formulation of a discrete-time Markov Decision Process (MDP) relating states $s \in S$ and actions $a \in A$ to subsequent states and rewards.

[2] These definitions can be customized with alternative metrics for value estimation and optimality, such as replacing $|\hat{v}(s) - v_\pi(s)|$ with $(\hat{v}(s) - v_\pi(s))^2$.



(a) Default start (b) Default death (c) Modified start (d) Modified death

Figure 1: Minor changes in AMIDAR game state can dramatically reduce a trained agent's reward.

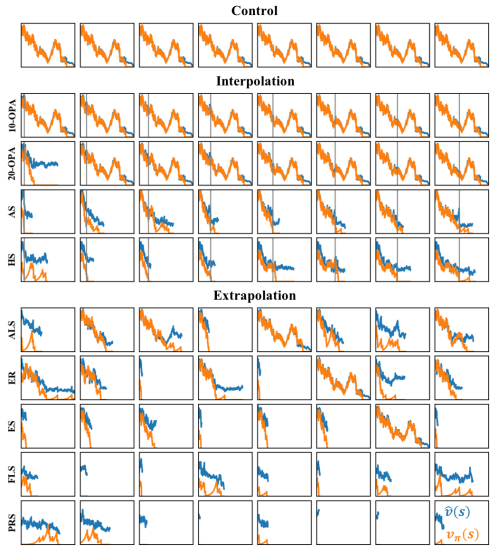
traverses them. An episode terminates when the player makes contact with one of the five enemies that also move along the grid.

Consider the two executions of an agent's learned policy in Figure 1 starting from two distinct states, *default* and *modified*. The default condition places the trained agent in the deterministic start position it experienced during training. The modified condition is identical, except that a single line segment has been filled in. While this exact state could never be observed during training, we would expect an agent that has learned appropriate representations and a generalized policy to perform well. Indeed, with a segment filled in, the agent is at least as close to completing the level as in the default condition. However, this small modification causes the agent to obtain an order of magnitude smaller reward. Importantly, this perturbation differs from an adversarial attack [2] for deep agents in that it influences the latent *semantics* of state, not solely the agent's *perception* of that state. Our experiments expand on this representative example, enumerating a set of perturbations.

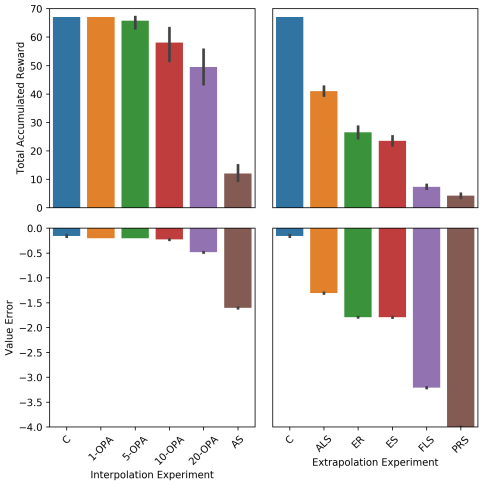
Off-policy States. We employed three strategies to generate off-policy states for an agent: human starts, agent swaps, and k-OPA. In each case, we passed control to the agent after n steps, where $n \in \{100, 200, \ldots, 900\}$. Human start states are generated using a human user-interface to INTERVENIDAR. Four individuals played 30 INTERVENIDAR games each. We randomly selected 75 action sequences lasting more than 1000 steps and extracted 9 states, taken at each of the n timesteps [5]. Agent swap states are generated by executing the policy of an alternative agent. We trained four *alternative agents*: (1) an agent that starts each training episode with 50 random actions, (2) an agent with half of the convolutional channels as the baseline architecture, (3) an agent with one less convoultional layer than baseline archiecture, and (4) an agent with half the hidden nodes as the baseline agent. We chose these agents with the belief that their policies would be sufficienctly different from each other to provide variation in off-policy states. k-OPA states are generated by following the agent's policy for n steps, before taking k random off-policy actions, where k was set to 10 and 20. None of these methods require the INTERVENIDAR system.

Unreachable States. With INTERVENIDAR, we generated unreachable states by modifying individual components of game state such as the filled/unfilled status of a line segment. This process guarantees that the agent begins an episode in a state it has never encountered during training, as the unmodified INTERVENIDAR environment is a nearly faithful replica of the deterministic game of AMIDAR. All modifications to the board happen before gameplay. We distinguish between existential modifications, modifications that alter the existence or quantity of game entities, and parameterized modifications, modification that alter the value of a particular latent state variable. We make one existential and one parameterized modification to enemies: We randomly remove between one and four enemies from the board (ER), and we shift one randomly selected enemy by n steps along its path, where n is drawn randomly between 1 and 20 (ES). We make one existential and one parameterized modification to line segments: We add one new vertical line segment to a random location on the board (ALS) and we randomly fill between one and four non-adjacent unfilled line segments (FLS). We start the player in a randomly chosen unoccupied tile location that has at least one tile of buffer between the player and any enemies (PRS).

Metrics. Generalization in Q-value-based RL can be encapsulated by two measurements for off-policy and unreachable states, one that accounts for the condition $\delta > |\hat{q}(s,a) - q_\pi(s,a)|$—whether the agent's estimate is close to the actual Q-value after executing π—and another for the condition $\gamma > q^*(s,a) - q_\pi(s,a)$—whether the actual Q-value is close to the optimal Q-value. In our work, we use value estimate error, $\text{VEE}_\pi(s) = \hat{v}(s) - v_\pi(s)$, and total accumulated reward, $\text{TAR}_\pi(s) = \mathbb{E}_\pi\left[\sum_{k=1}^{\infty} R(s_{t+k}) \mid s_t = s, a_t = a\right]$, respectively.



(a) $\hat{v}(s)$ and $v_\pi(s)$ for replicated trajectories for all experiments. Each subplot is a single independent trial. For the interpolation experiments, the vertical grey line shows the point where the agent takes random actions (in the k-OPA experiments) or regains control (in the agent swaps and human-starts experiments). The length of each episode is consistently lower and the difference between $\hat{v}(s)$ and $v_\pi(s)$ is consistently higher for the extrapolation experiments.



(b) TAR and average VEE for control, extrapolation, and interpolation experiments. The agent consistently overestimates the state value. TAR and VEE are strongly anti-correlated. All TAR bars are normalized by the TAR of the control condition. All VEE bars are normalized by their respective TAR.

Figure 2: Evaluation Results

4 Results and Conclusions

Our experiments demonstrate that the state-of-the-art DQN has poor generalization performance for AMIDAR gameplay. Figures 2a and 2b show that the fully trained state-of-the-art DQN dueling architecture produces a policy that is exceptionally brittle to small non-adversarial changes in the environment. The most egregious examples can be seen in Figure 2b, in the filling line segments (FLS) and player random starts (PRS) interventions. Visual inspection of the action sequences proceeeding these states showed the agent predominantly remaining stationary, often terminating the epsisode without traversing a single line segment. This behavior can be seen in Figure 2a, where PRS and FLS episodes terminate prematurely.

Furthermore, Figure 2b shows that VEE and TAR are very highly anti-correlated across the experiments, indicating that the agent's ability to select appropriate actions is related to its ability to correctly measure the value of a particular state. We observe that the model always overestimates the value of off-policy and unreachable states. In contrast, the agent's value estimates are small and approximately symetrically distributed around 0 in the control condition.

Generalization in RL needs to be discussed more broadly, as a capability of an arbitrary agent. We propose framing generalization as the performance metric of the researcher's choice over a partition of on-policy, off-policy, and unreachable states. Our custom, parameterizable AMIDAR simulator is a proof of concept of the type of simulation environments that are needed for generating unreachable states and training truly general agents.

Acknowledgements

This material is based upon work supported by the United States Air Force under Contract No, FA8750-17-C-0120. Any opinions, findings and conclusions or recommendations expressed in this

material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

References

[1] Matthew J Hausknecht and Peter Stone. The impact of determinism on learning atari 2600 games. In *AAAI Workshop: Learning for General Competency in Video Games*, 2015.

[2] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. 2017.

[3] Ken Kansky, Tom Silver, David A Mély, Mohamed Eldawy, Miguel Lázaro-Gredilla, Xinghua Lou, Nimrod Dorfman, Szymon Sidor, Scott Phoenix, and Dileep George. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. In *International Conference on Machine Learning*, pages 1809–1818, 2017.

[4] Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *arXiv preprint arXiv:1709.06009*, 2017.

[5] Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, et al. Massively parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1507.04296*, 2015.

[6] Ali Nouri, Michael L Littman, Lihong Li, Ronald Parr, Christopher Painter-Wakefield, and Gavin Taylor. A novel benchmark methodology and data repository for real-life reinforcement learning. In *Proceedings of the 26th international conference on machine learning*, 2009.

[7] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[8] Shimon Whiteson, Brian Tanner, Matthew E Taylor, and Peter Stone. Protecting against evaluation overfitting in empirical reinforcement learning. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2011 IEEE Symposium on*, pages 120–127. IEEE, 2011.

[9] Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. A study on overfitting in deep reinforcement learning.