LOSS IS ITS OWN REWARD: SELF-SUPERVISION FOR REINFORCEMENT LEARNING

Evan Shelhamer[†‡] **Parsa Mahmoudieh**[†], **Max Argus**[†], **Trevor Darrell**[†]
[†]UC Berkeley
‡OpenAI
{shelhamer,trevor}@cs.berkeley.edu; parsa.m@berkeley.edu; argus.max@gmail.com

ABSTRACT

Reinforcement learning, driven by reward, addresses tasks by optimizing policies for expected return. Need the supervision be so narrow? Reward is delayed and sparse for many tasks, so we argue that reward alone is a noisy and impoverished signal for end-to-end optimization. To augment reward, we consider self-supervised tasks that incorporate states, actions, and successors to provide auxiliary losses. These losses offer ubiquitous and instantaneous supervision for representation learning even in the absence of reward. Self-supervised pre-training improves the data efficiency and returns of end-to-end reinforcement learning.

1 INTRODUCTION

End-to-end reinforcement learning (RL) addresses representation learning at the same time as policy optimization and value estimation. Of these dual pursuits, current work focuses on the reinforcement learning aspects of the problem such as stochastic optimization, exploration, and more. Having defined a loss on reward, the representation is delegated to backpropagation without further attention. However, representation learning is a bottleneck in current approaches that are bound by reward.

To illustrate the critical role of representation learning, we show that re-training an agent after destroying the action and value outputs is far faster than the initial training (Figure 1). Although the policy distribution and value function are lost, they are readily recovered given a representation from RL, even though the optimization and exploration issues remain. We turn to self-supervision to take an ambient approach to RL attuned to reward and environment alike.

Self-supervision defines losses via surrogate annotations that are readily synthesized from bare, unlabeled inputs. In the context of RL, reward captures the task while self-supervision helps capture the environment. In this setting, every transition contributes gradients of ambient environmental signals. While loss from reward might be delayed and sparse, the losses from self-supervision are instantaneous and ubiquitous. Augmenting RL with these auxiliary losses enriches the representation through multi-task learning and improves policy optimization.

We focus on auxiliary losses with discriminative formulations for state, dynamics, inverse dynamics, and reward. We transfer pre-training by these self-supervised tasks to RL. Policy optimization to 95% of best return is sped-up $1.4\times$ on average for a number of Atari environments.

2 SELF-SUPERVISION OF POLICIES

Self-supervised learning defines surrogate losses and synthesizes the targets from the data. To relate it to supervised and unsupervised learning, consider the general form of the objectives:

- supervised learning $\min_\theta \mathbb{E}\left[L_{\text{dis}}(f_\theta(x), y)\right]$
- unsupervised learning $\min_\theta \mathbb{E}\left[L_{\text{gen}}(f_\theta(x), x)\right]$
- self-supervised learning $\min_\theta \mathbb{E}\left[L_{\text{dis}}(f_\theta(x), s(x))\right]$ with surrogate annotation function $s(\cdot)$

for data x, annotation y, losses L either discriminative or generative, and parametric model f_θ.

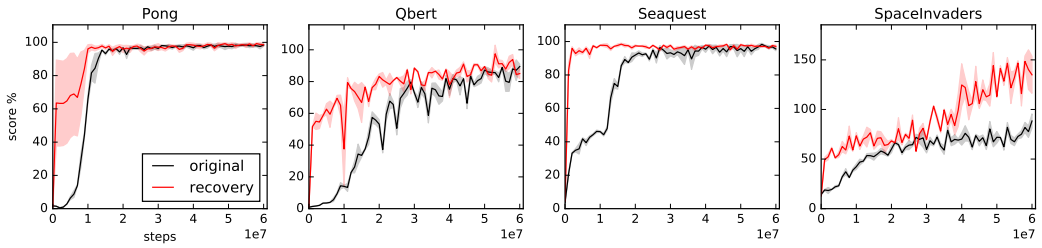

Figure 1: To separate reinforcement learning from representation learning, we decapitate trained agents by destroying the policy and value output weights, and then re-train end-to-end. Although the policy distribution and value estimates are obliterated, most of the weights are preserved and the policy is swiftly recovered. The gap between the initial optimization and recovery illustrates a representation learning bottleneck.

The state, action, successor, and reward (s, a, s', r) transition standard to RL admits many kinds of self-supervision. We explore the use of surrogate annotations that span different parts of the transitions to gauge what is informative for RL. These diverse, ambient losses mine further supervision from the same data available to existing RL methods.

2.1 Tasks

Our self-supervised tasks define auxiliary losses for pre-training an actor-critic network for RL.

Reward Reward can be cast into a proxy task as instantaneous prediction by regression or binning into positive, zero, and negative classes. This is equivalent to one-step value function estimation, and so may seem redundant for value methods. However, the gradient of the instantaneous prediction task is less noisy as it sidesteps bootstrapping error. Our self-supervised reward task is to bin r_t into $r'_t \in \{0, +, -\}$ with equal balancing of the classes as done independently by Jaderberg et al. (2017).

Dynamics and Inverse Dynamics Dynamics can be cast into a verification task by recognizing whether state-successor (s, s') pairs are drawn from the environment or not. Our self-supervised dynamics verification task is to identify the corrupted observation o_{t_c} in a history from t_0 to t_k, where o_{t_c} is corrupted by swapping it with $o_{t'}$ for $t' \notin \{t_0, \ldots, t_k\}$. Inverse dynamics, mapping $\mathcal{S} \times \mathcal{S} \to \mathcal{A}$, can be reduced to classification (for discrete actions) or regression (for continuous actions). Our self-supervised inverse dynamics task is to infer the intervening actions given a history of observations.

Reconstruction Auto-encoding (AE) and variational auto-encoding (VAE) learn to reconstruct the input subject to a representational bottleneck. While a popular line of attack for unsupervised learning, the representations learned by reconstruction are relatively poor for transfer (Donahue et al., 2017). Nevertheless we include reconstruction for comparison with our self-supervised tasks.

3 Policy Pre-Training Results

We show results on self-supervision for policy pre-training on Atari. The data for pre-training on each environment is collected by executing a random policy for $100,000$ transitions. The optimization of the auxiliary losses converges quickly (< 10 epochs) to reasonable task accuracy. Policies pre-trained by self-supervision converge to same or better return and do so in fewer updates.

Our self-supervised policies are instantiated as variations of the asynchronous advantage actor-critic (A3C) architecture of Mnih et al. (2016). The actor-critic network is taken as an encoder to which each task attaches its own decoder. For the environment we follow the specification from Mnih et al. (2015) by our own re-implementation with the OpenAI Gym (Brockman et al., 2016).

We compare simple initialization strategies—random initialization as well as calibrated and data-dependent initialization (Krähenbühl et al., 2016)—with our self-supervised tasks. These tasks include auxiliary losses that are agnostic to reward, letting learning make progress while waiting for reward. Table 1 reports data efficiency, and Figure 2 shows policy optimization progress.

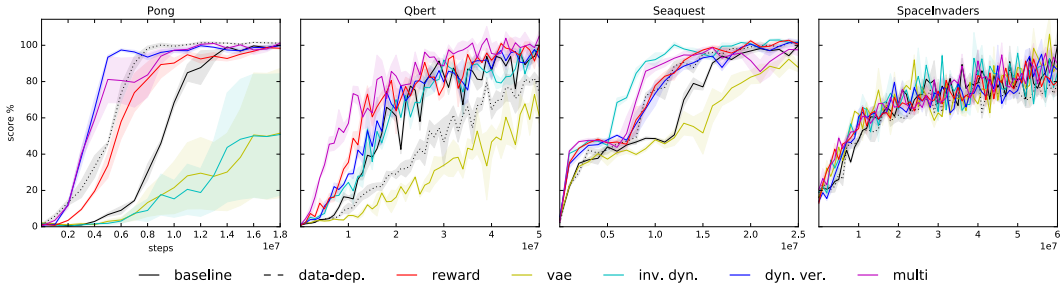

Figure 2: Policy optimization from self-supervised pre-training. Progress is reported as the percentage of the best baseline return. The mean and standard error (shading) are shown for three runs. Multi-task self-supervision reliably improves data efficiency and converges to comparable returns.

	Pong	Qbert	Seaquest	S. Invaders
Data-Dep. Init.	1.51×	0.69×	1.13×	0.93×
Reward	1.32×	1.16×	1.17×	1.01×
Dyn. Ver.	1.61×	1.09×	1.14×	1.00×
Inv. Dyn.	0.38×	1.02×	1.26×	1.04×
VAE	0.38×	0.46×	0.86×	1.02×
Multi-task	1.55×	1.32×	1.18×	1.04×

Table 1: We examine the data efficiency of RL with self-supervised pre-training. We calculate the area under the score/iteration curve and report the ratio to the baseline. Multi-task self-supervision improves $1.3\times$ on average, and early on it gives $3\times$ improvement for the first 10M iterations.

4 Related Work

Representation learning for reinforcement learning, robotics, and control is commonly known as state representation learning, as it yields the state for modeling the task as an MDP. This can be summarized formally as seeking a mapping ϕ such that the current state $s_t = \phi(o_{1:t}, a_{1:t}, r_{1:t})$ as in Jonschkowski & Brock (2015).

Unsupervised learning by auto-encoding is a common approach to state representation learning (Watter et al., 2015; Finn et al., 2016). These approaches optimize policies to achieve a goal state without a task reward, so it is not possible to fine-tune the representation to optimize return. In contrast our auxiliary, discriminative losses capture dynamics, inverse dynamics, and other aspects of the environment in tandem with RL.

The robotic priors of Jonschkowski & Brock (2015) are auxiliary losses for temporal coherence, repeatability, proportionality, and causality. Multi-task optimization of these losses defines a linear, low-dimensional state representation for RL. These losses are distances between states conditioned on action and reward, while we define discriminative losses on the (s, a, r, s') of transitions.

Concurrent work explores different methods to augment reinforcement learning with auxiliary losses (Jaderberg et al., 2017; Mirowski et al., 2017; Dosovitskiy & Koltun, 2017). In the same spirit as our work, these approaches seek to improve policy returns, data efficiency, and robustness of end-to-end RL. Our tasks do not require additional privileged information, we focus on discriminative formulations of auxiliary losses, and we compare a variety of ambient signals for self-supervision.

5 Discussion

It is encouraging that pre-training alone, with and without reward, can improve optimization for reinforcement learning. By augmenting reinforcement learning with self-supervision, transitions without reward need not be so unrewarding for the representation.

ACKNOWLEDGEMENTS

This work was supported in part by Berkeley AI Research, Berkeley Deep Drive, NSF, DARPA, NVIDIA, and Intel. We gratefully acknowledge NVIDIA for GPU donation.
We thank John Schulman and Chelsea Finn for advice and useful discussions.

REFERENCES

G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *ICLR*, 2017.

A. Dosovitskiy and V. Koltun. Learning to act by predicting the future. *ICLR*, 2017.

C. Finn, X.Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel. Deep spatial autoencoders for visuomotor learning. In *ICRA*, 2016.

M. Jaderberg, V. Mnih, W. Marian Czarnecki, T. Schaul, J.Z. Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *ICLR*, 2017.

R. Jonschkowski and O. Brock. Learning state representations with robotic priors. *Autonomous Robots*, 39(3):407–428, 2015.

Philipp Krähenbühl, Carl Doersch, Jeff Donahue, and Trevor Darrell. Data-dependent initializations of convolutional neural networks. In *ICLR*, 2016.

P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, et al. Learning to navigate in complex environments. *ICLR*, 2017.

V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

V. Mnih, A.P. Badia, M. Mirza, A. Graves, T.P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *ICML*, 2016.

M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *NIPS*, pp. 2746–2754, 2015.