Learning Value Functions from Undirected State-only Experience

Matthew Chang* Arjun Gupta* Saurabh Gupta
University of Illinois at Urbana-Champaign
{mc48, arjung2, saurabhg}@illinois.edu

1. Introduction

Offline or batch reinforcement learning focuses on learning goal-directed behavior from pre-recorded data of undirected experience in the form of (s_t, a_t, s_{t+1}, r_t) quadruples. However, in many realistic applications, action information is not naturally available (*e.g.* when learning from video demonstrations), or worse still, isn't even well-defined (*e.g.* when learning from the experience of an agent with a different embodiment). Motivated by such use cases, this paper studies if, and how, intelligent behavior can be derived from undirected streams of observations: (s_t, s_{t+1}, r_t).

Our key conceptual insight is that while an observation-only dataset doesn't tell us the precise action to execute, it may still tell us which states are more likely to lead us to the goal than not, *i.e.* the value function $V(s)$. In this paper we present a method that learns value functions using Q-learning on discrete latent actions obtained through a latent-variable future prediction model. Our experiments show that using learned value functions as dense rewards can lead to quick policy learning through some small amount of interaction in the environment, or they can guide the behavior of low-level controllers directly, without any further training.

2. Latent Action Q-Learning

Our proposed approach decouples learning into three steps: mining *latent* actions from state-only trajectories, using these latent actions for Q-learning to obtain value functions, and learning a policy to act according to the learned value function. Refer to Figure 1 for an overview. If latent actions are a *state-conditioned refinement* of the original actions, Q-learning with latent actions will result in the same value function as Q-learning with ground-truth actions [2].

2.1. Latent Actions from Future Prediction

Given a dataset $\mathbf{D}$ of observations streams $\ldots, o_t, o_{t+1}, \ldots$, we learn the latent actions through future prediction. We train a future prediction model f_θ, that maps the observation o_t at time t, and a latent action $\hat{a}$ (from a set $\hat{\mathbf{A}}$ of discrete latent actions) to the observation o_{t+1} at time $t+1$, *i.e.* $f_\theta(o_t, \hat{a})$. f is trained to minimize the L2 loss between the prediction $f_\theta(o_t, \hat{a})$ and the ground truth observation o_{t+1}. Each training sample (o_t, o_{t+1}) is assigned to the action that leads to the lowest loss under the current forward model.

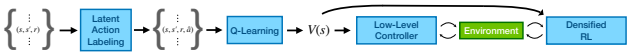

Figure 1. Approach Overview. Our proposed approach Latent Action Q-Learning (LAQ) starts with a dataset of (s, s', r) triples. Using the latent action learning process, each sample is assigned a latent action $\hat{a}$. Q-learning on the dataset of quadruples produces a value function, $V(s)$. Behaviors are derived from the value function through densified RL, or by guiding low-level controllers.

2.2. Q-learning with Latent Actions

Latent actions mined from Section 2.1 allow us to complete the given (o_t, o_{t+1}, r_t) tuples into $(o_t, \hat{a}_t, o_{t+1}, r_t)$ quadruples for use in Q-learning [8]. We note that this Q-learning still needs to be done in an *offline* manner from pre-recorded state-only experience. Value functions are obtained from the Q-functions as $V(s) = \max_{\hat{a} \in \hat{\mathbf{A}}} Q(s, \hat{a})$.

2.3. Behaviors from Value Functions

Given a value function, we derive behavior in the underlying MDP in the following two ways.

Densified Reinforcement Learning. We use the value function to create a *potential-based* shaping function $F(s, s') = V(s') - V(s)$, based on [5], and construct an augmented reward function $r'(s, a, s') = r(s, a, s') + F(s, s')$. This augmented reward is used with online RL to derive a policy.

Domain Specific Low-level Controllers. In more specific scenarios, behavior can directly be obtained by picking the low-level controller that conveys the agent to the state s' that has the highest value under the learned $V(s)$.

* denotes equal contribution. Project website: https://matthewchang.github.io/latent_action_qlearning_site/.

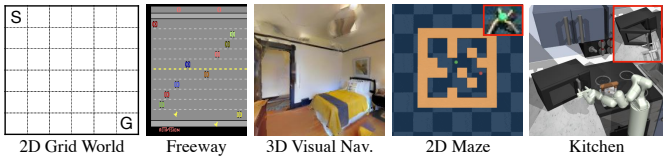

Figure 2. We experiment with five environments: 2D Grid World, Freeway (Atari), 3D Visual Navigation, Maze2D (2D Continuous Control), and FrankaKitchen.

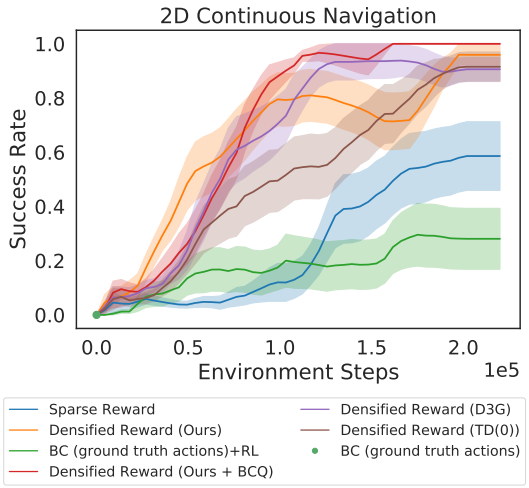

Figure 3. We show learning curves for acquiring behavior using learned value functions. We compare densified RL with sparse RL. Results are averaged over 5 seeds and show ± standard error.

3. Experiments

We design experiments to assess the quality of value functions learned by LAQ from undirected state-only experience. For each setting, we work with a pre-collected dataset of experience in the form of state, next state and reward triplets, (o_t, o_{t+1}, r_t). We provide action labels to these triplets, and produce value functions using the LAQ method described above. We evaluate the learned value functions in two ways.

First, we measure the extent to which value functions learned with LAQ without ground truth information agree with value functions learned with Q-learning with ground truth action information. We do this by measuring the Spearman's rank correlation coefficient between the different value functions. Our second evaluation measures the effectiveness of LAQ-learned value functions for deriving effective behavior in different settings: when using it as a dense reward, and when using it to guide low-level controllers. Figure 2 shows the environments which were used for our experiments.

3.1. Quality of Learned Value Functions

Table 1 reports the Spearman's coefficients of value functions obtained using different action labels: D3G [3], clustering, and latent actions (ours). Our method outperforms all baselines in settings with high-dimensional image observations. In state-based settings, where clustering state differences is a helpful inductive bias, our method is still on-par with, or superior to clustering state differences and even D3G, which predicts state differences.

Table 1. We report Spearman's correlation coefficients for value functions learned using various methods with DQN, against a value function learned offline using ground-truth actions (DQN for discrete action environments, DDPG for continuous).

Environment	D3G	Clustering (Diff)	Latent Actions
2D Grid World	0.959	**1.000**	0.985
Freeway	– (image input)	0.902	**0.961**
3D Visual Navigation	– (image input)	0.827	**0.927**
2D Continuous Control	0.673	0.490	**0.844**
Kitchen Manipulation	0.854	0.815	**0.905**

3.2. Using Value Functions for Downstream Tasks

Our experiments test the utility of LAQ-learned value functions for acquiring goal-driven behavior. Figure 3 measures the learning sample efficiency. We compare to only using the sparse reward, behavior cloning (BC) with ground truth actions, and BC followed by sparse reward RL. The following is a summary of our takeaways.

LAQ value functions speed up downstream learning. Learning plots in Figure 3 show that, using LAQ-learning value functions compares favorably to sparse reward (orange line *vs*. blue line). Our method learns more quickly than sparse reward and converges to a higher mean performance. The same trend holds across other tested environments.

LAQ discovers stronger behavior than imitation learning when faced with undirected experience. An advantage of LAQ over other imitation-learning based methods such as BCO [7] and ILPO [4] is LAQ's ability to learn from sub-optimal or undirected experience. To showcase this, we compare the performance of LAQ with BC with ground truth actions (which serves as an upper bound on all methods in this class). Learning plots in Figure 3 shows the effectiveness of LAQ over BC and BC followed by fine-tuning with sparse rewards. LAQ discovers stronger behavior than imitation learning when faced with undirected data.

LAQ value functions can guide low-level controllers for zero-shot control. Learned value functions can also be used to guide the behavior of low-level controllers directly at test time. We do this experiment in the context of 3D visual navigation in a realistic simulation based on 3D house scans [6]. The problem setup is based on the *branching environment* from [1]. We find that LAQ learned value functions result in a higher SPL (0.82) than clustering-based value functions (0.57) and other baselines.

References

[1] Matthew Chang, Arjun Gupta, and Saurabh Gupta. Semantic visual navigation by watching youtube videos. In *NeurIPS*, 2020. 2

[2] Matthew Chang, Arjun Gupta, and Saurabh Gupta. Learning value functions from undirected state-only experience. *arXiv preprint arXiv:2204.12458*, 2022. 1

[3] Ashley D. Edwards, Himanshu Sahni, Rosanne Liu, Jane Hung, Ankit Jain, Rui Wang, Adrien Ecoffet, Thomas Miconi, Charles Isbell, and Jason Yosinski. Estimating $q(s, s')$ with deep deterministic dynamics gradients. In *ICML*, 2020. 2

[4] Ashley D Edwards, Himanshu Sahni, Yannick Schroecker, and Charles L Isbell. Imitating latent policies from observation. In *ICML*, 2019. 2

[5] Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, pages 278–287. Morgan Kaufmann, 1999. 1

[6] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied AI research. In *ICCV*, 2019. 2

[7] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *IJCAI*, 2018. 2

[8] Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989. 1