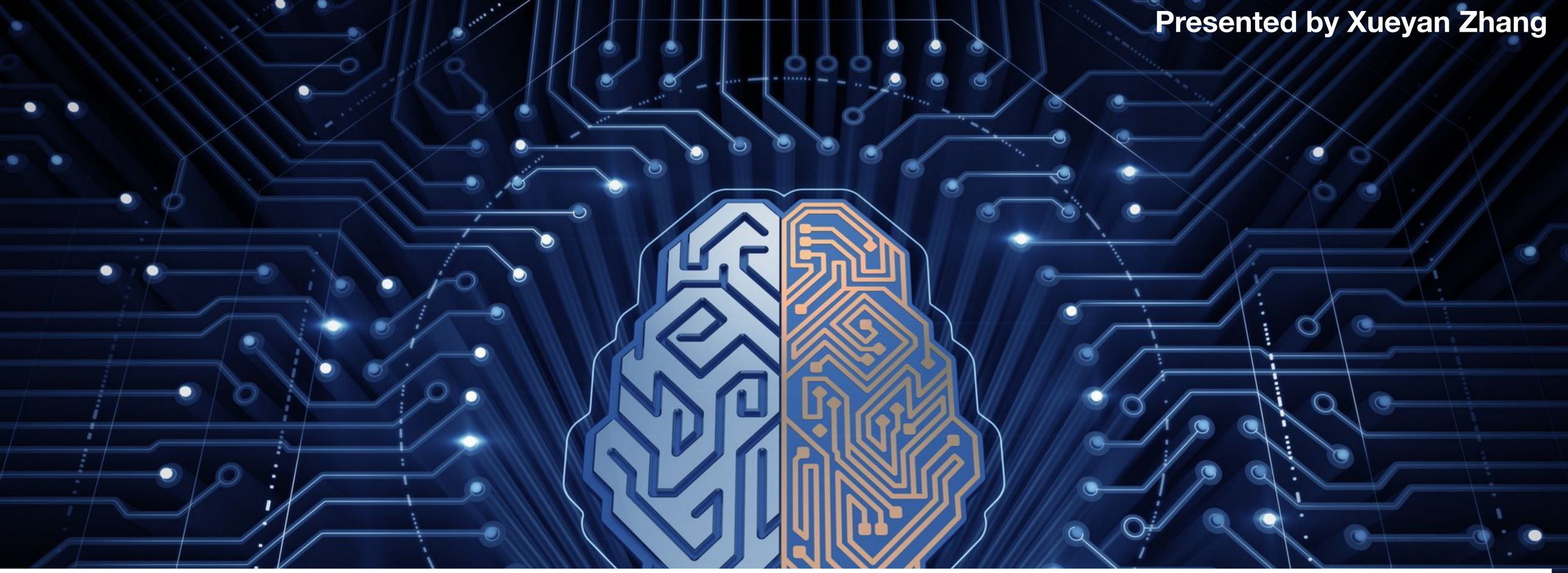
Presented by Xueyan Zhang



Composable Deep Reinforcement Learning for Robotic Manipulation

Tuomas Haarnoja, Vitchyr Pong, Aurick Zhou, Murtaza Dalal, Pieter Abbeel, Sergey Levine

Composable Deep Reinforcement Learning for Robotic Manipulation

Tuomas Haarnoja[1], Vitchyr Pong[1], Aurick Zhou[1], Murtaza Dalal[1], Pieter Abbeel[1,2], Sergey Levine[1]

***Abstract*— Model-free deep reinforcement learning has been shown to exhibit good performance in domains ranging from video games to simulated robotic manipulation and locomotion. However, model-free methods are known to perform poorly when the interaction time with the environment is limited, as is the case for most real-world robotic tasks. In this paper, we study how maximum entropy policies trained using soft Q-learning can be applied to real-world robotic manipulation. The application of this method to real-world manipulation is facilitated by two important features of soft Q-learning. First, soft Q-learning can learn multimodal exploration strategies by learning policies represented by expressive energy-based models. Second, we show that policies learned with soft Q-learning can be composed to create new policies, and that the optimality of the resulting policy can be bounded in terms of the divergence between the composed policies. This compositionality provides an especially valuable tool for real-world manipulation, where constructing new policies by composing existing skills can provide a large gain in efficiency over training from scratch. Our experimental evaluation demonstrates that soft Q-learning is substantially more sample efficient than prior model-free deep reinforcement learning methods, and that compositionality can be performed for both simulated and real-world tasks.**

I. INTRODUCTION

The intersection of expressive, general-purpose function approximators, such as neural networks, with general purpose model-free reinforcement learning algorithms that can be used to acquire complex behavioral strategies holds the promise of automating a wide range of robotic behaviors: reinforcement learning provides the formalism for reasoning about sequential decision making, while large neural networks provide the representation that can, in principle, be used to represent any behavior with minimal manual engineering. However, applying model-free reinforcement learning algorithms with multilayer neural network representations (i.e., deep reinforcement learning) to real-world robotic control problems has proven to be very difficult in practice: the sample complexity of model-free methods tends to be quite high, and is increased further by the inclusion of high-capacity function approximators. Prior work has sought to alleviate these issues by parallelizing learning across multiple robots [1], making use of example demonstrations [2], [3], or training in simulation and relying on an accurate model to enable transfer to the real world [4], [5]. All of these approaches carry additional assumptions and limitations. Can we instead devise model-free reinforcement learning algorithms that are efficient enough to train multilayer neural network models directly in the real world, without reliance on simulation, demonstrations, or multiple robots?

[1]Berkeley Artificial Intelligence Research, UC Berkeley, [2]Open AI
{haarnoja, vitchyr, azhou42, mdalal, pabbeel, svlevine}@berkeley.edu

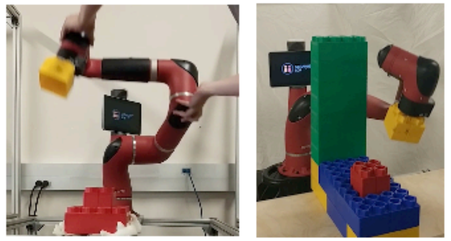
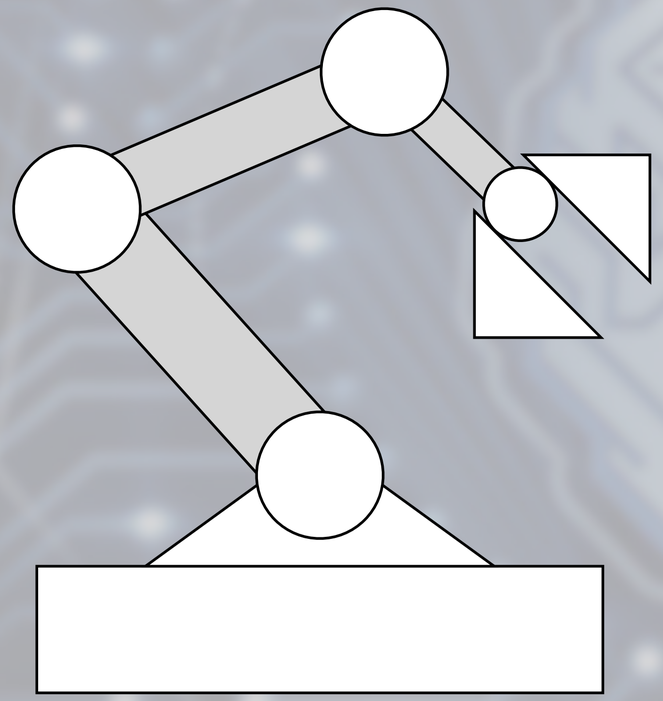
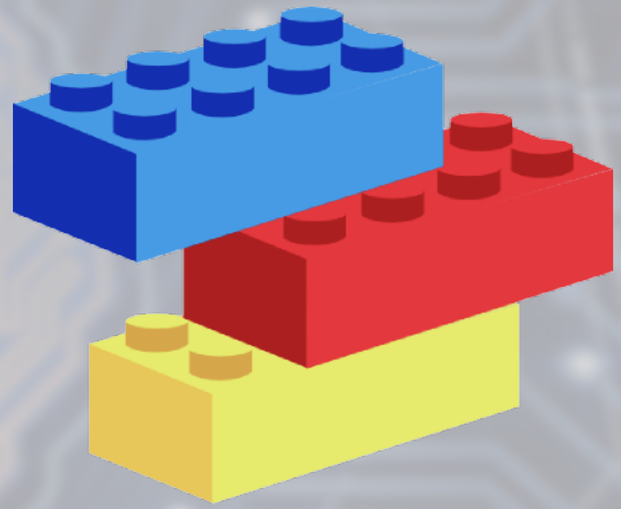


Fig. 1. We trained a Sawyer robot to stack Lego blocks together using maximum entropy reinforcement learning algorithm called soft Q-learning. Training a policy from scratch takes less than two hours, and the learned policy is extremely robust against perturbations (left figure). We also demonstrate how learned policies can be combined to form new compound skills, such as stacking while avoiding a tower of Lego blocks (right figure).

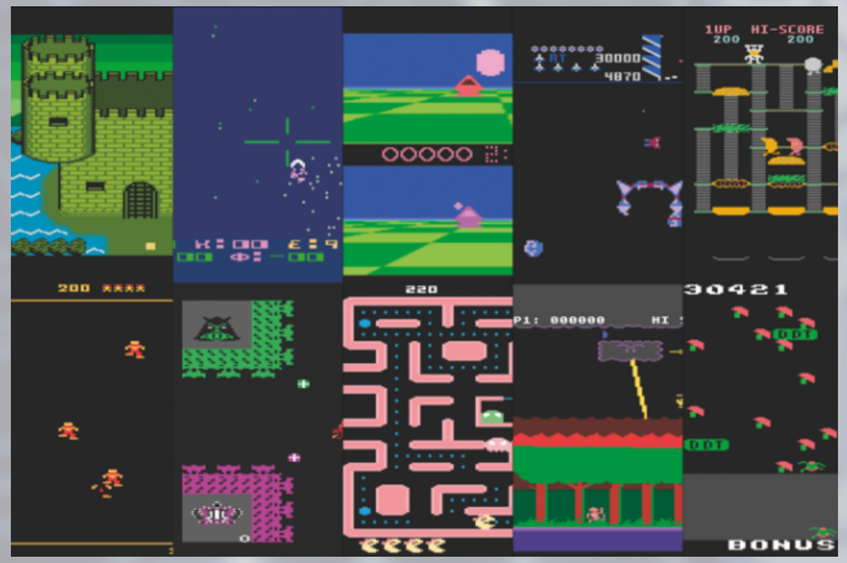
We hypothesize that the maximum entropy principle [6] can yield an effective framework for practical, real-world deep reinforcement learning due to the following two properties. First, maximum entropy policies provide an inherent, informed exploration strategy by expressing a stochastic policy via the Boltzmann distribution, with the energy corresponding to the reward-to-go or Q-function [7]. This distribution assigns a non-zero probability to all actions, but actions with higher expected rewards are more likely to be sampled. As a consequence, the policy will automatically direct exploration into regions of higher expected return. This property, which can be thought of as a soft combination of exploration and exploitation, can be highly beneficial in real-world applications, since it provides considerably more structure than ϵ-greedy exploration and, as shown in our experiments, substantially improves sample complexity. Second, as we show in this paper, independently trained maximum entropy policies can be composed together by adding their Q-functions, yielding a new policy for the combined reward function that is provably close to the corresponding optimal policy. Composability of controllers, which is not typically possible in standard reinforcement learning, is especially important for real-world applications, where reuse of past experience can greatly improve sample efficiency for tasks that can naturally be decomposed into simpler sub-problems. For instance, a policy for a pick-and-place task can be decomposed into (1) reaching specific x-coordinates, (2) reaching specific y-coordinates, and (3) avoiding certain obstacles. Such decomposable policies can therefore be learned in three stages, each yielding a sub-policy, which can later be combined offline without the need to interact with the environment.

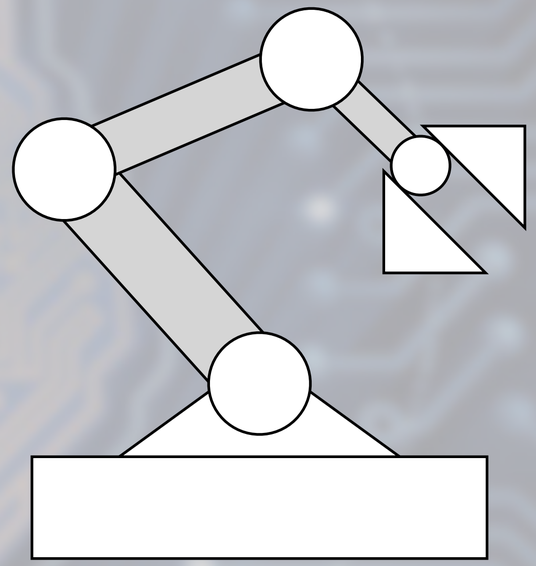
1) Robotic Manipulation





2) Policy Composing


1UP HI-SCORE
200 200
30000
4870
200
220
30421
P1: 000000 HI
BONUS



State of the Art

NAF: Normalized Advantage Functions

DDPG: Deep Deterministic Policy Gradient

SQL: Soft Q-Learning

Standard RL Policy:

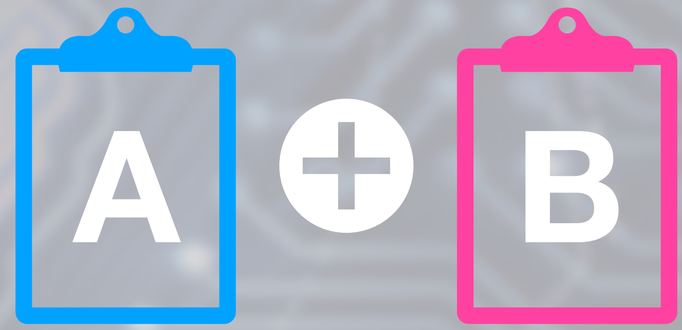
$$\pi^* = \underset{\pi}{\operatorname{argmax}} \sum_{n=0}^{N} \gamma^n E_{s_n, a_n | \pi}[R(s_n, a_n)]$$

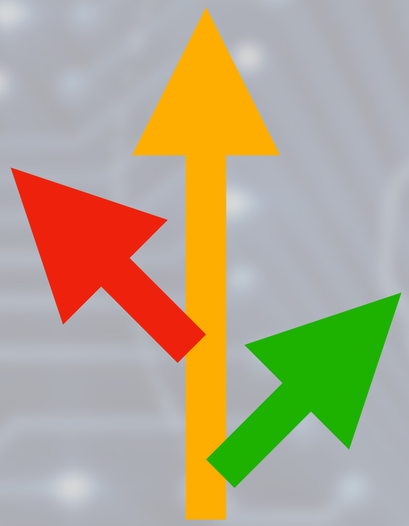
Maximum Entropy Policy:

$$\pi^*_{soft} = \underset{\pi}{\operatorname{argmax}} \sum_{n=0}^{N} \gamma^n E_{s_n, a_n | \pi}\left[R(s_n, a_n) + \lambda H\left(\pi(\cdot\,|s_n)\right)\right]$$

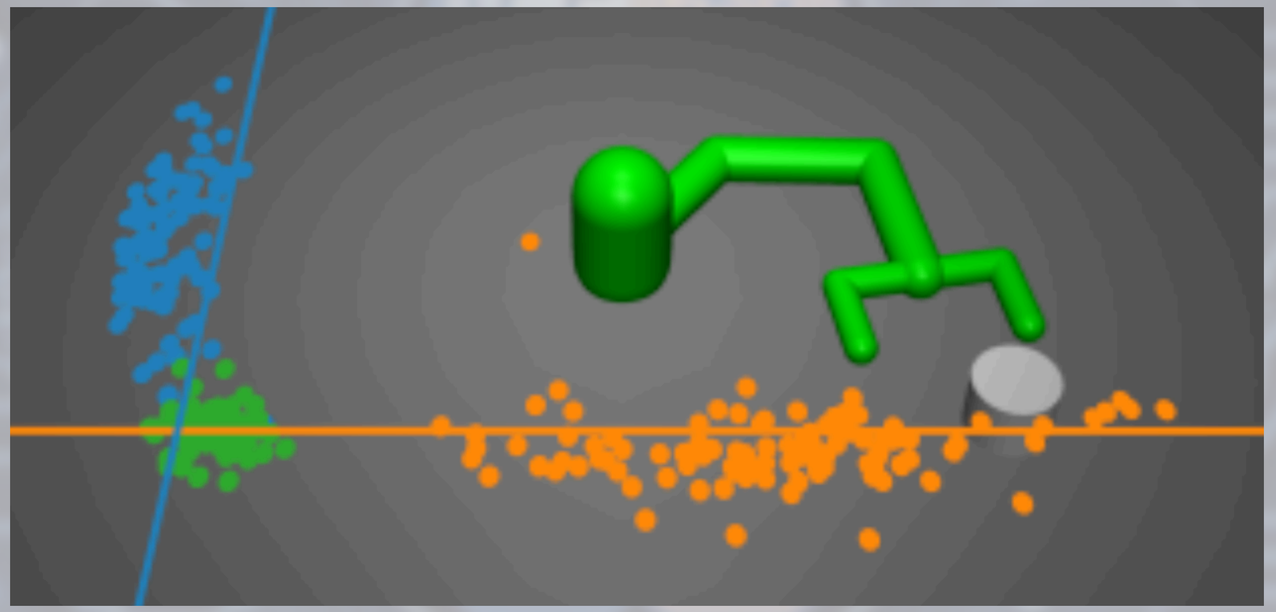
Why SQL?

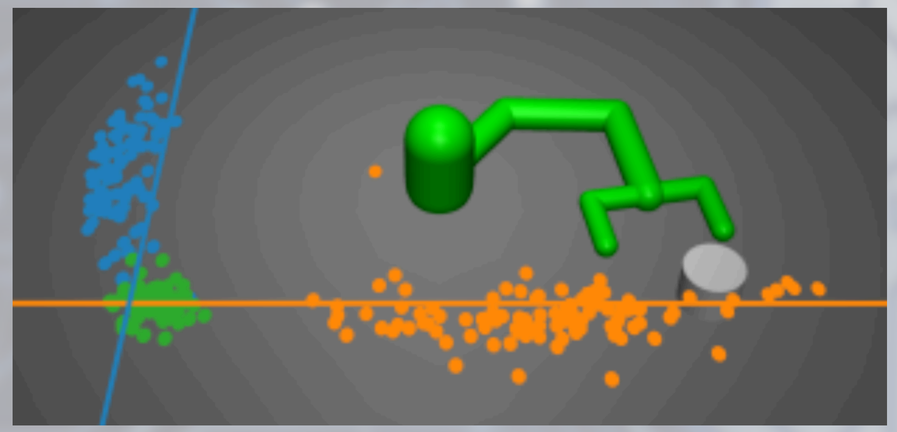
Compositionality



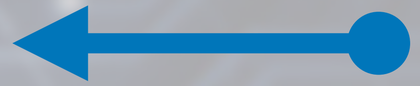


Multimodal Exploration





Policy A:

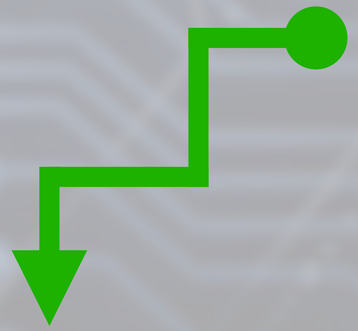


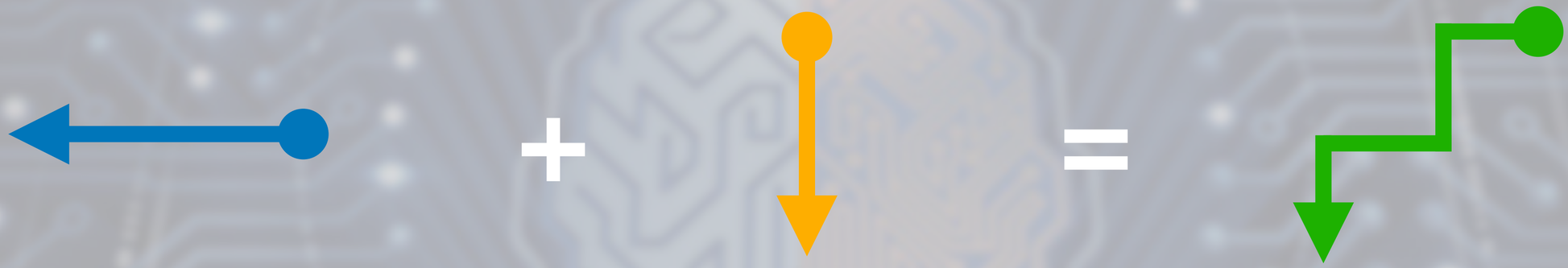
+

Policy B:

=

Combined Policy:





Reward A

Reward B

Sum of Rewards

Q_Σ: Upper Boundary

Lemma 1: Let Q_1^* and Q_2^* be the soft Q-function of the optimal policies corresponding to reward functions r_1 and r_2, and define $Q_\Sigma \triangleq \frac{1}{2}(Q_1^* + Q_2^*)$. Then the optimal soft Q-function of the combined reward $r_\mathcal{C} \triangleq \frac{1}{2}(r_1 + r_2)$ satisfies

$$Q_\Sigma(\mathbf{s},\mathbf{a}) \geq Q_\mathcal{C}^*(\mathbf{s},\mathbf{a}) \geq Q_\Sigma(\mathbf{s},\mathbf{a}) - C^*(\mathbf{s},\mathbf{a}), \quad \forall \mathbf{s} \in \mathcal{S}, \forall \mathbf{a} \in \mathcal{A}, \tag{7}$$

where C^* is the fixed point of (8)

$$C(\mathbf{s},\mathbf{a}) \leftarrow \gamma\, \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[\mathcal{D}_{\frac{1}{2}} \left(\pi_1^*(\cdot|\mathbf{s}') \,\|\, \pi_2^*(\cdot|\mathbf{s}') \right) + \max_{\mathbf{a}' \in \mathcal{A}} C(\mathbf{s}',\mathbf{a}') \right],$$

and $\mathcal{D}_{\frac{1}{2}}(\cdot \,\|\, \cdot)$ is the Rényi divergence of order $1/2$.

Proof: See Appendix A. ■

$Q_C^{\pi_\Sigma}$: Upper Boundary

Theorem 1: With the definitions in Lemma 1, the value of π_Σ satisfies

$$Q_C^{\pi_\Sigma}(\mathbf{s}, \mathbf{a}) \geq Q_C^*(\mathbf{s}, \mathbf{a}) - D^*(\mathbf{s}, \mathbf{a}), \tag{9}$$

where $D^*(\mathbf{s}, \mathbf{a})$ is the fixed point of (10)

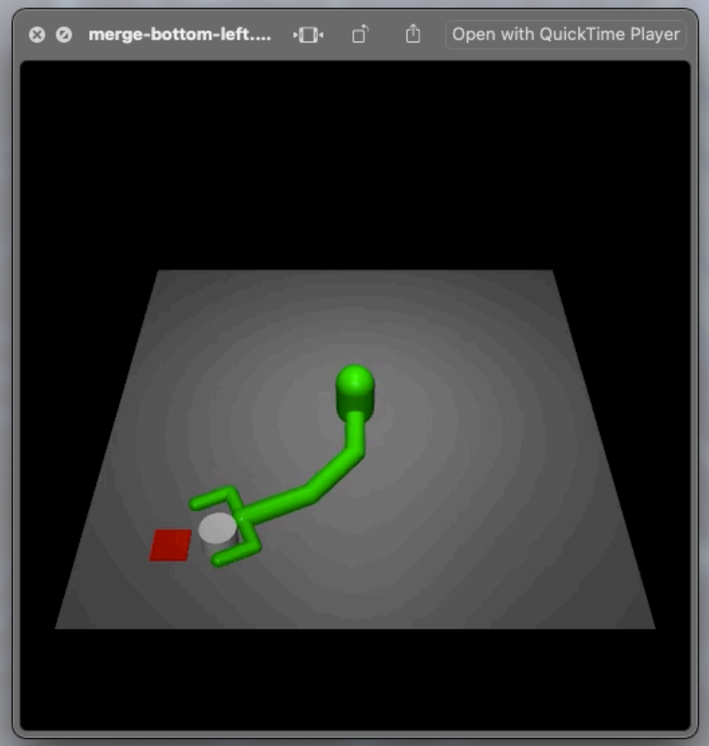
$$D(\mathbf{s}, \mathbf{a}) \leftarrow \gamma \, \mathbb{E}_{\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})} \left[\mathbb{E}_{\mathbf{a}' \sim \pi_\Sigma(\mathbf{a}'|\mathbf{s}')} \left[C^*(\mathbf{s}', \mathbf{a}') + D(\mathbf{s}', \mathbf{a}') \right] \right].$$

Proof: See Appendix B. ■

Evaluation

Simulation

MuJoCo physics engine



Real-World

Sawyer robotic manipulator

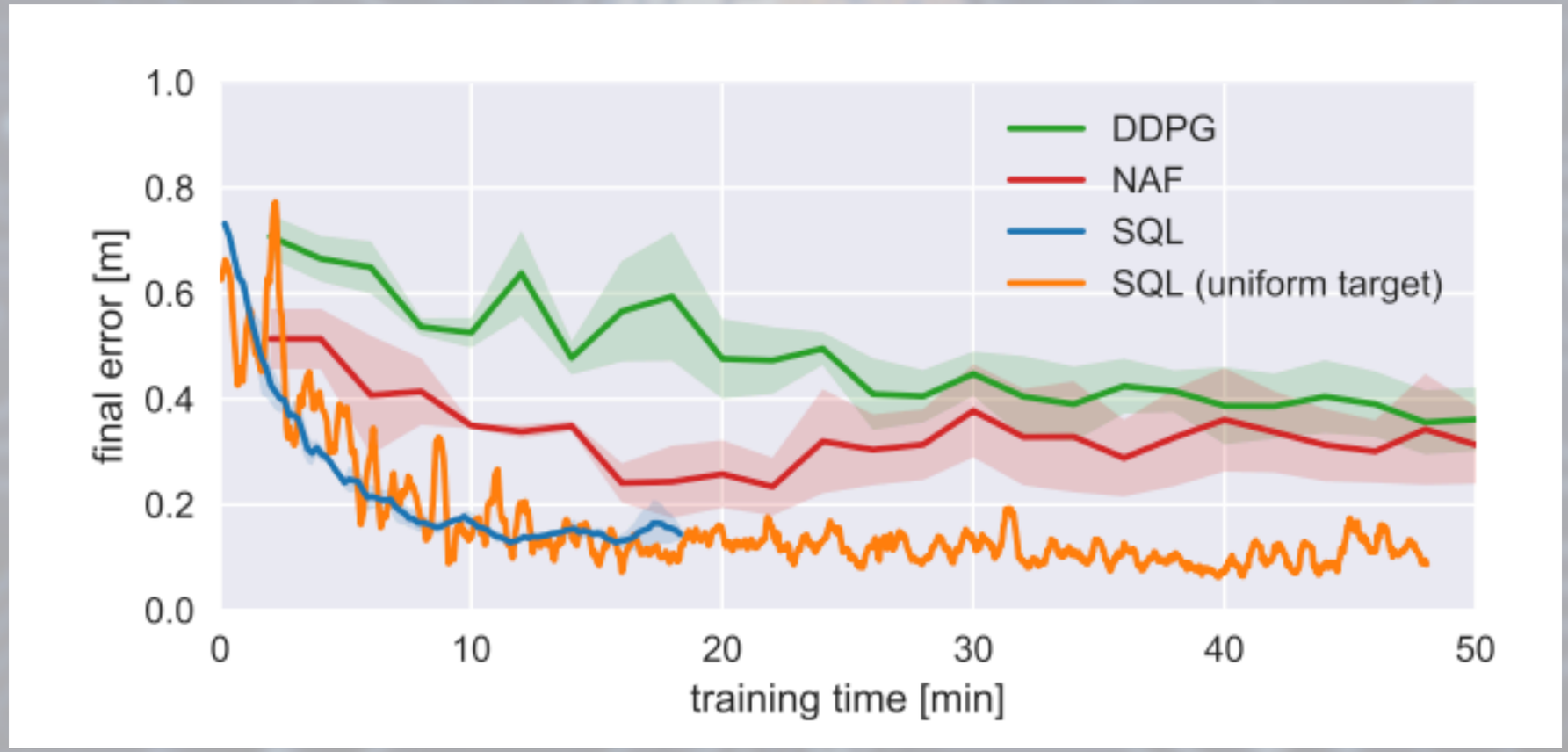

Baselines: 1. NAF; 2. DDPG

Tasks

1. Pushing
2. Reaching
3. Lego Block Stacking
4. Stacking + Avoiding

Pushing

Reaching


1.0
0.8
0.6
0.4
0.2
0.0
final error [m]
0
10
20
30
40
50
training time [min]
DDPG
NAF
SQL
SQL (uniform target)

Stacking

Stacking + Avoiding

Conclusion

Soft Q-Learning