QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation

Dmitry Kalashnikov , Alex Irpan , Peter Pastor , Julian Ibarz , Alexander Herzog , Eric Jang , Deirdre Quillen, Ethan Holly , Mrinal Kalakrishnan , Vincent Vanhoucke , Sergey Levine

Topic: Q Value based RL

Presenter: Vismay Modi

Motivation

Grasping is a very common and necessary manipulation problem.

… Motivation …

In nature, grasping is a “dynamical process, interleaving sensing and control at every stage”

… Motivation …

It is very difficult, even for humans…

Definitely difficult for robots.

… Motivation

Prior work in 2 classes:

1. Open loop control:
    a. sense -> plan -> act
    b. No feedback handling
2. Close loop control:
    a. Manually dictated strategies
    b. Short-horizon reasoning

The paper:

QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation

Dmitry Kalashnikov[1], **Alex Irpan**[1], **Peter Pastor**[2], **Julian Ibarz**[1], **Alexander Herzog**[2], **Eric Jang**[1], **Deirdre Quillen**[3], **Ethan Holly**[1], **Mrinal Kalakrishnan**[2], **Vincent Vanhoucke**[1], **Sergey Levine**[1,3]
{dkalashnikov, alexirpan, julianibarz, ejang, eholly, vanhoucke, slevine}@google.com,
{peterpastor, alexherzog, kalakris}@x.team, {deirdrequillen}@berkeley.edu

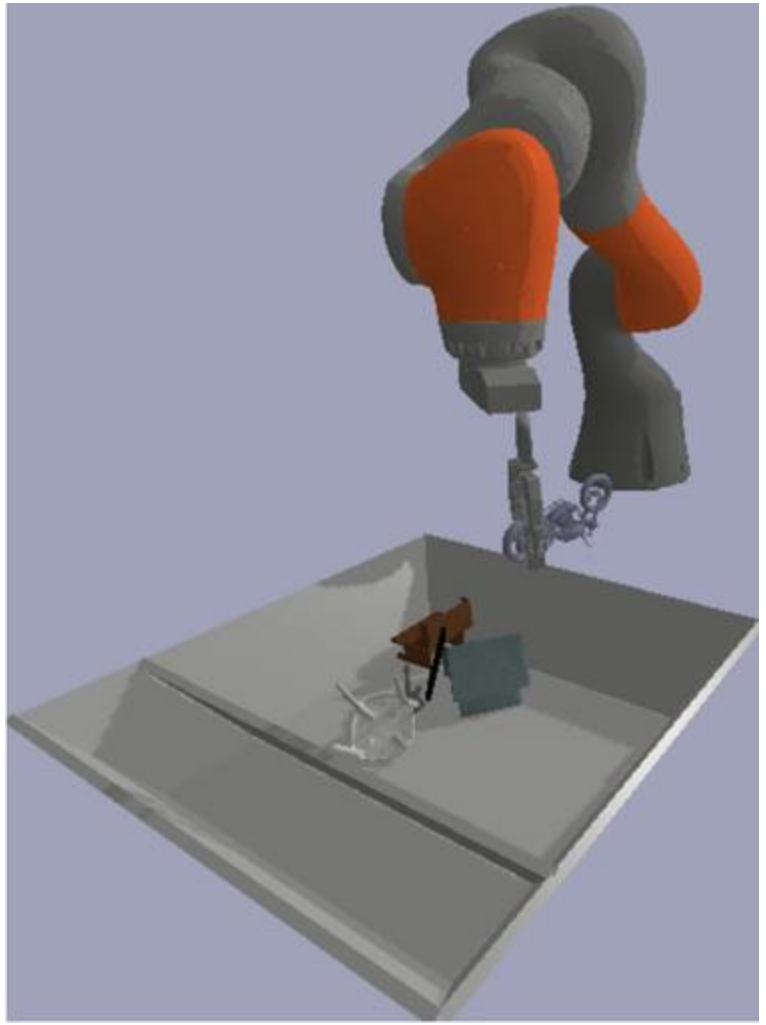
Abstract: In this paper, we study the problem of learning vision-based dynamic manipulation skills using a scalable reinforcement learning approach. We study this problem in the context of grasping, a longstanding challenge in robotic manipulation. In contrast to static learning behaviors that choose a grasp point and then execute the desired grasp, our method enables closed-loop vision-based control, whereby the robot continuously updates its grasp strategy based on the most recent observations to optimize long-horizon grasp success. To that end, we introduce QT-Opt, a scalable self-supervised vision-based reinforcement learning framework that can leverage over 580k real-world grasp attempts to train a deep neural network Q-function with over 1.2M parameters to perform closed-loop, real-world grasping that generalizes to 96% grasp success on unseen objects. Aside from attaining a very high success rate, our method exhibits behaviors that are quite distinct from more standard grasping systems: using only RGB vision-based perception from an over-the-shoulder camera, our method automatically learns regrasping strategies, probes objects to find the most effective grasps, learns to reposition objects and perform other non-prehensile pre-grasp manipulations, and responds dynamically to disturbances and perturbations.[4]

Keywords: grasping, reinforcement learning, deep learning

Contribution…

QT-Opt solves the problem of:

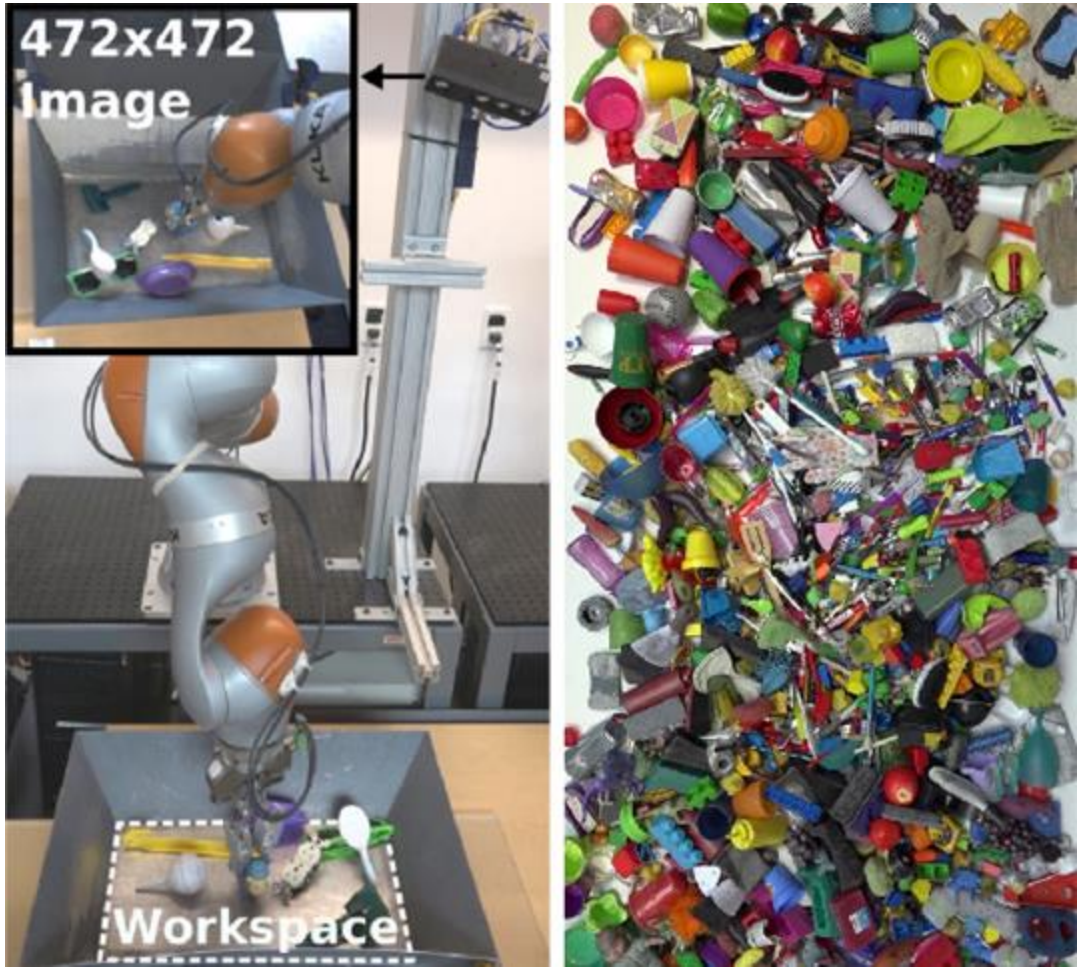
Grasping previously unseen objects while continuously updating its strategy



… Contributions…

Grasping is hard because it requires robots to have:

- Vision
- Strategy
- Control
- Knowledge of the object
- Knowledge of the environment



… Contributions …

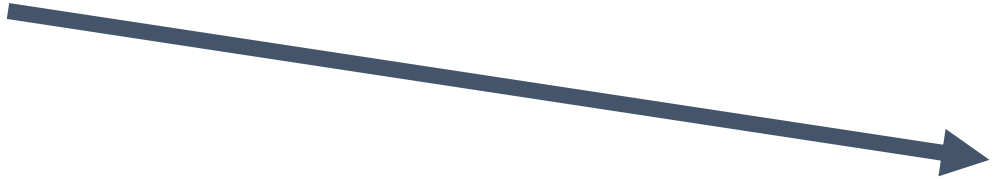
Prior works limitations:

- Strategy doesn't look at feedback (open-loop) [5,6,7,8]
- Requires supervision
- Requires multiple cameras or depth sensing [7,25,29]

… Contributions

Key insights of QT-opt

- Closed loop control
  - Dynamic strategies
  - Handles changes in the environment
- Self supervised, vision based training
- Long-horizon planning using RL
- Uses a single, over-the-shoulder, RGB camera for vision



96% success on grasping previously unseen objects

General background and definitions

Behavior policy: How the robot behaves

Update policy: How the robot learns the optimal policy

On-policy learning: The robot learns and behaves using the same policy

Off-policy learning (Q-learning): The robot learns using a different policy than behaves

Algorithm (background)...

Q. Whats the goal of RL?

A. “Figure out a policy that selects actions which maximize total expected rewards”

Q. How do you do that?

A. Solve for an optimal Q(s, a) function, (value function) which determines the expected rewards received for an action (a) taken in state (s)

Q. And once you have a good Q function?

A. Recover the policy $\pi(\mathbf{s}) = \arg\bar{\max}_{\mathbf{a}} Q_{\bar{\theta}_1}(\mathbf{s}, \mathbf{a})$

Algorithm (background)…

Q. What is a good Q function?

A. In this paper, the authors minimize the bellman error

Target value: gets updated as a better target value is found

Metric to measure distance

Current Q: Get this to match target

$$\mathcal{E}(\theta) = \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim p(\mathbf{s},\mathbf{a},\mathbf{s}')}\left[D\left(Q_\theta(\mathbf{s},\mathbf{a}), Q_T(\mathbf{s},\mathbf{a},\mathbf{s}')\right)\right], \quad (1)$$

Network weights

State = image

Robot's action

Q. And once you have a good Q function?

A. Recover the policy $\pi(\mathbf{s}) = \arg\max_{\mathbf{a}} Q_{\bar{\theta}_1}(\mathbf{s},\mathbf{a})$

… Algorithm …

Q. What if Q is non-convex (not easily maximizable)?

A. This is where “QT-opt” is special.

A derivative-free stochastic optimization is done to maximize Q (using the cross-entropy method).

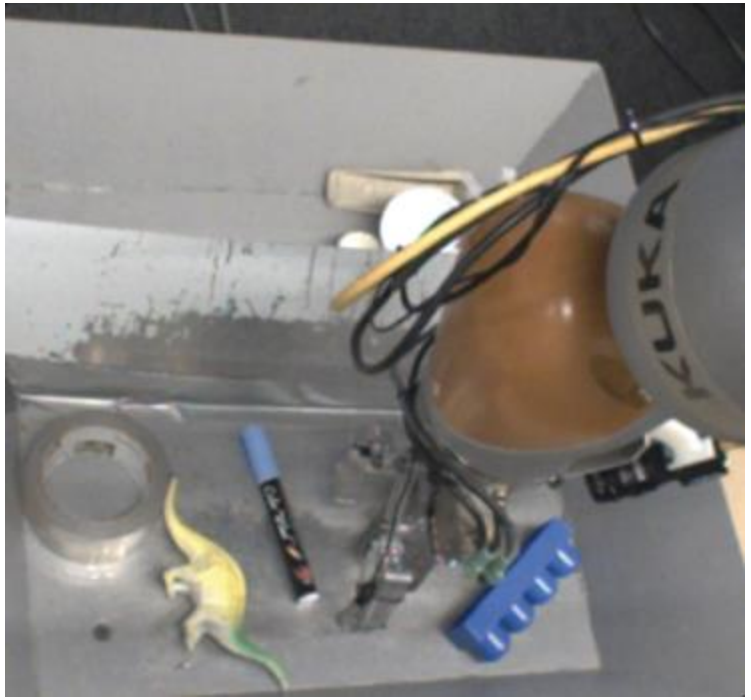
End “Q & A”

… Algorithm …

Distributed Async QT-Opt

Training over 7 robots using a distributed async implementation of QT-Opt allowed collecting 580k grasps over several weeks.

… Algorithm…



Dynamic Vision Based Grasping

- Monocular over the shoulder,

  RGB camera, 472x472 pixels

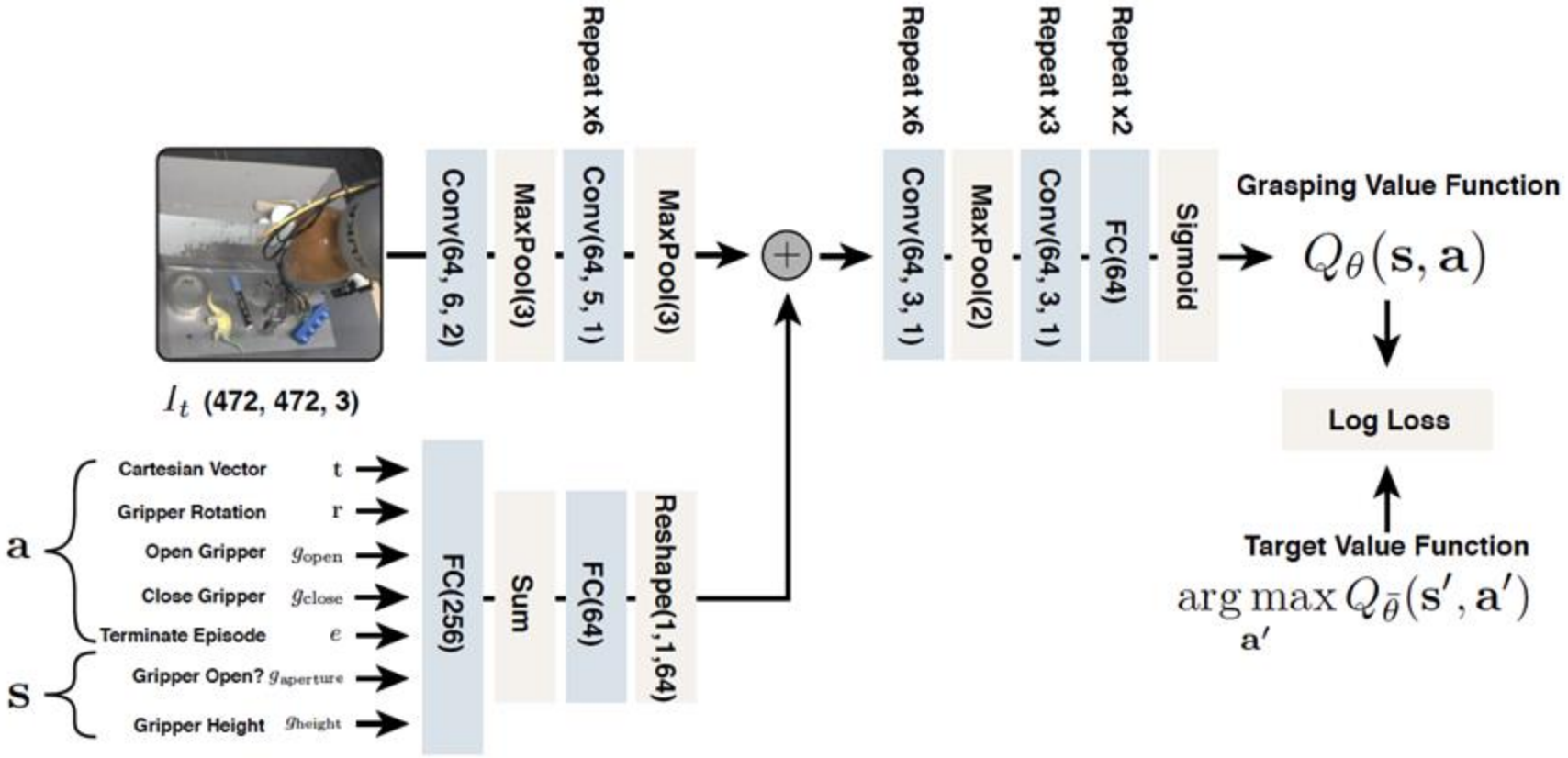
MDP for grasping

- **State (s)** = {camera image, gripper open/closed, gripper height}
- **Action (a)** = {gripper position, gripper angle, gripper open/closed, termination}
- **Reward function** = 1 for holding an object above a certain height, -0.5 for each time step taken

… Algorithm

Q-function representation

- Large conv net with 1.2M parameters



Experimental Results

State	Termination action	Intermediate reward	Discount factor	Perf. at 300K steps	Perf. at 1M steps
Image+gripper status+height	No	-0.05	0.9	75%	95%
	No	0	0.9	68%	92%
	No	0	0.7	50%	90%
Image only	No	-0.05	0.9	25%	81%
Image+gripper status+height	Yes	-0.05	0.9	67%	94%

Table 7: Simulation studies for tuning grasping task parameters

Discussion

- Small time penalty (-0.5) increases performances
- No Termination action speeds up the learning rate, but the authors argue having a termination action would make the MDP implementation simpler
- Otherwise, results are pretty self evident

More results

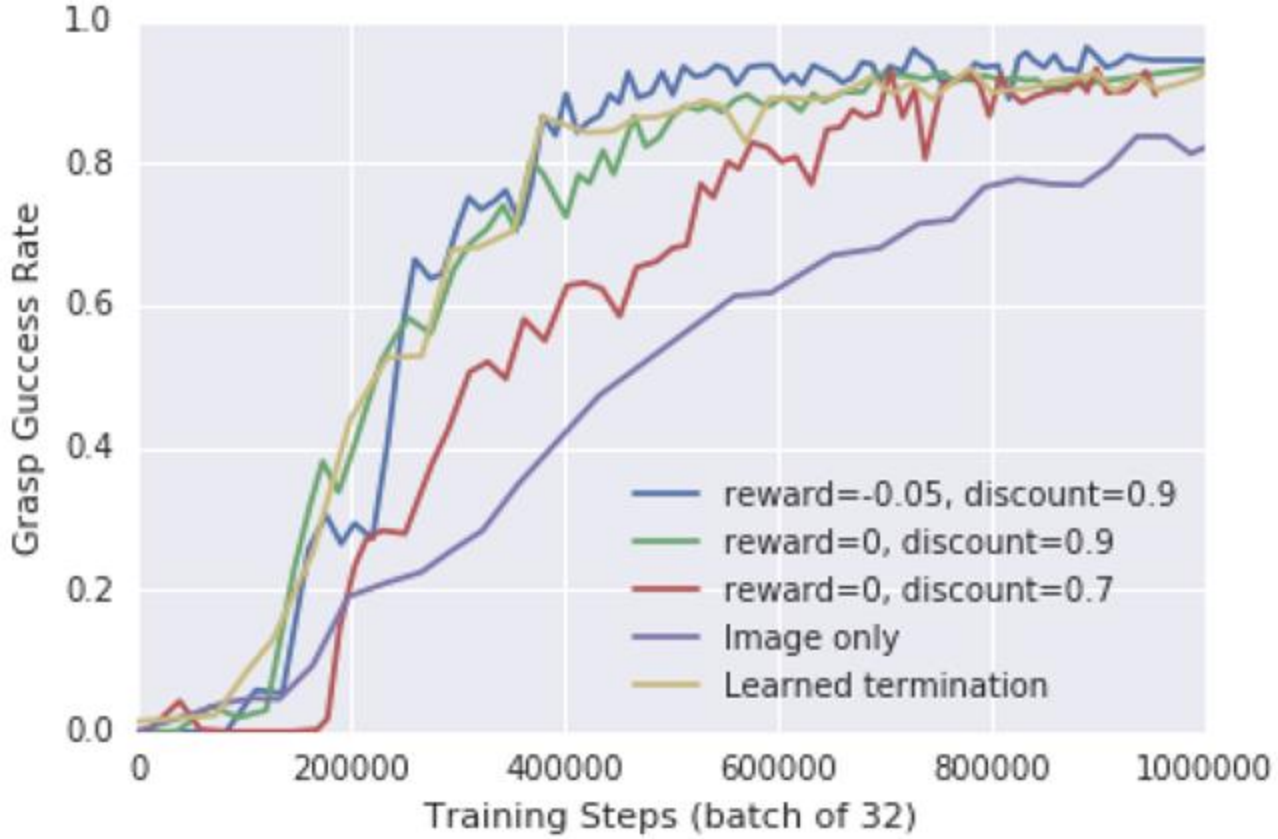


Figure 10: Performance graphs of simulation studies for tuning grasping task parameters

Discussion

- The rates of learning differed quite a bit for the different parameters, surprisingly.

- Maybe some kind of hybrid approach would accelerate training?
  - Use the fastest learning parameters for each interval

Critique

Algorithm critique

- Initial policies were hand-scripted to bootstrap data collection
- We don’t know much about the effectiveness of the gradient-free stochastic optimization method to find an optimal Q.
  - Ex. Why terminate CEM after 2 iterations? Why is M,N = 6,64 good values?

Issues

Time discretization issues

- Time step is too large
- The motion of the robot is very unnatural
- Too few time steps allowed (only 20)

Limitations

Implementation limitations

- Immense undertaking with 7 robots and weeks of experimentation
- Not easily replicable

QT-Opt Contributions (Recap)

- Attempts to solve grasping for previously unseen objects
- Uses only one shoulder mounted RGB camera for vision
- Closed-loop control
  - Dynamic grasping strategies
  - Handles external changes
- Distributional Asynch QT-Opt
  - (Mostly) self supervised
  - Off-policy learning
    - Allows learning on massive set of data (580k graps)
    - Distributed across 7 robot arms
  - Derivative-free stochastic optimization method to maximize Q
- **Results:** Near ±90% success rate at grasping

Follow up Q2T-Opt

Quantile QT-Opt for Risk-Aware Vision-Based Robotic Grasping

Cristian Bodnar[1], Adrian Li[2], Karol Hausman[3], Peter Pastor[2], Mrinal Kalakrishnan[2]

***Abstract*— The distributional perspective on reinforcement learning (RL) has given rise to a series of successful Q-learning algorithms, resulting in state-of-the-art performance in arcade game environments. However, it has not yet been analyzed how these findings from a discrete setting translate to complex practical applications characterized by noisy, high dimensional and continuous state-action spaces. In this work, we propose Quantile QT-Opt (Q2-Opt), a distributional variant of the recently introduced distributed Q-learning algorithm [11] for continuous domains, and examine its behaviour in a series of simulated and real vision-based robotic grasping tasks. The absence of an actor in Q2-Opt allows us to directly draw a parallel to the previous discrete experiments in the literature without the additional complexities induced by an actor-critic architecture. We demonstrate that Q2-Opt achieves a superior vision-based object grasping success rate, while also being more sample efficient. The distributional formulation also allows us to experiment with various risk-distortion metrics that give us an indication of how robots can concretely manage risk in practice using a Deep RL control policy. As an additional contribution, we perform experiments on offline datasets and compare them with the latest findings from discrete settings. Surprisingly, we find that there is a discrepancy between our results and the previous batch RL findings from the literature obtained on arcade game environments.**

Despite the improvements distributional Q-learning algorithms demonstrated in the discrete arcade environments, it is yet to be examined how these findings translate to practical, real-world applications. Intuitively, the advantageous properties of distributional Q-learning approaches should be particularly beneficial in a robotic setting. The value distributions can have a significant qualitative impact in robotic tasks, usually characterized by highly-stochastic and continuous state-action spaces. Additionally, performing safe control in the face of uncertainty is one of the biggest impediments to deploying robots in the real world, an impediment that RL methods have not yet tackled. In contrast, a distributional approach can allow robots to learn an RL policy that appropriately quantifies risks for the task of interest.

However, given the brittle nature of deep RL algorithms and their often counter-intuitive behaviour [8], it is not entirely clear if these intuitions would hold in practice. Therefore, we believe that an empirical analysis of distributional Q-learning algorithms in real robotic applications would shed light on their benefits and scalability, and provide essential insight for the robot learning community.

In this paper, we aim to address this need and perform a

Q2T-Opt

- **Quantile Q-learning:** Value function returns a distribution, not a single scalar value

- **Mapping a risk metric to the value distribution**
  - New policies can be risk-aware or risk-seeking
  - Improves the safety of doing RL

- Additionally, Q2T-Opt introduces a way to train the robot off-line

Q2T-Opt Result vs. QT-Opt

Model	Mean Success	Success Std	Median Success
QT-Opt	0.903	0.005	0.903
Q2R-Opt (Ours)	0.923	0.006	0.924
Q2F-Opt (Ours)	**0.928**	**0.001**	**0.928**

TABLE I: Final sim success rate statistics. The proposed distributional methods achieve higher success rate.

Discussion

- Q2T-Opt outperforms QT-Opt
- Standard deviation for Q2T-Opt is much smaller than QT-Opt

Q2T-Opt vs QT-Opt Learn rate

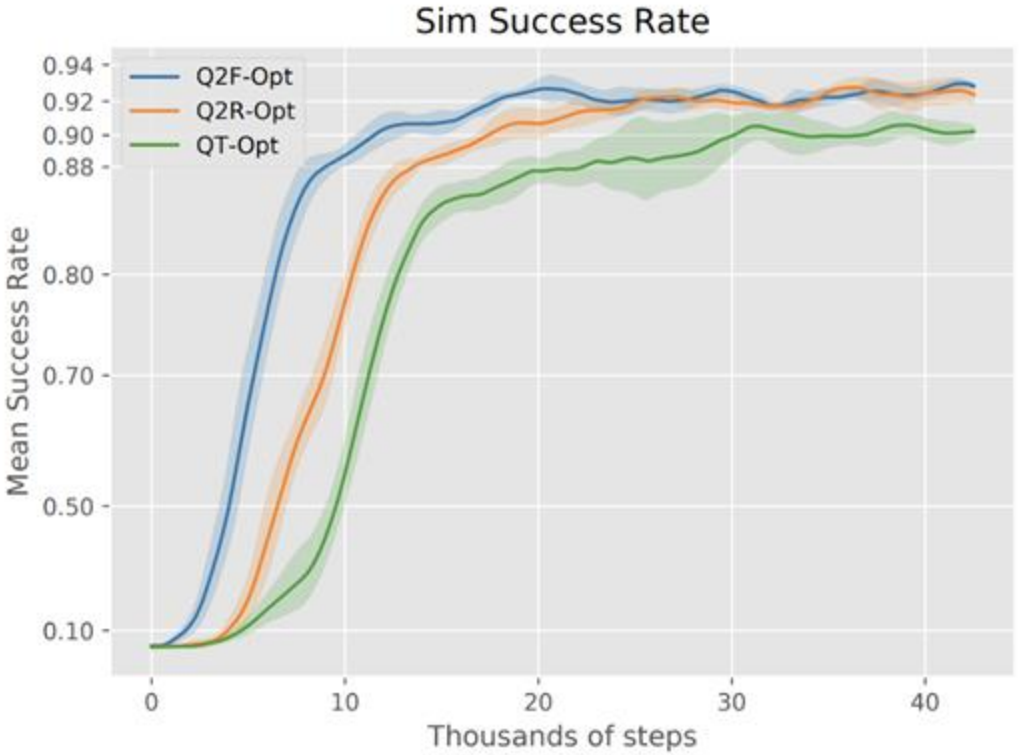


Fig. 3: Sim success rate as a function of the global step. The distributional methods achieve higher grasp success rates in a lower number of global steps.

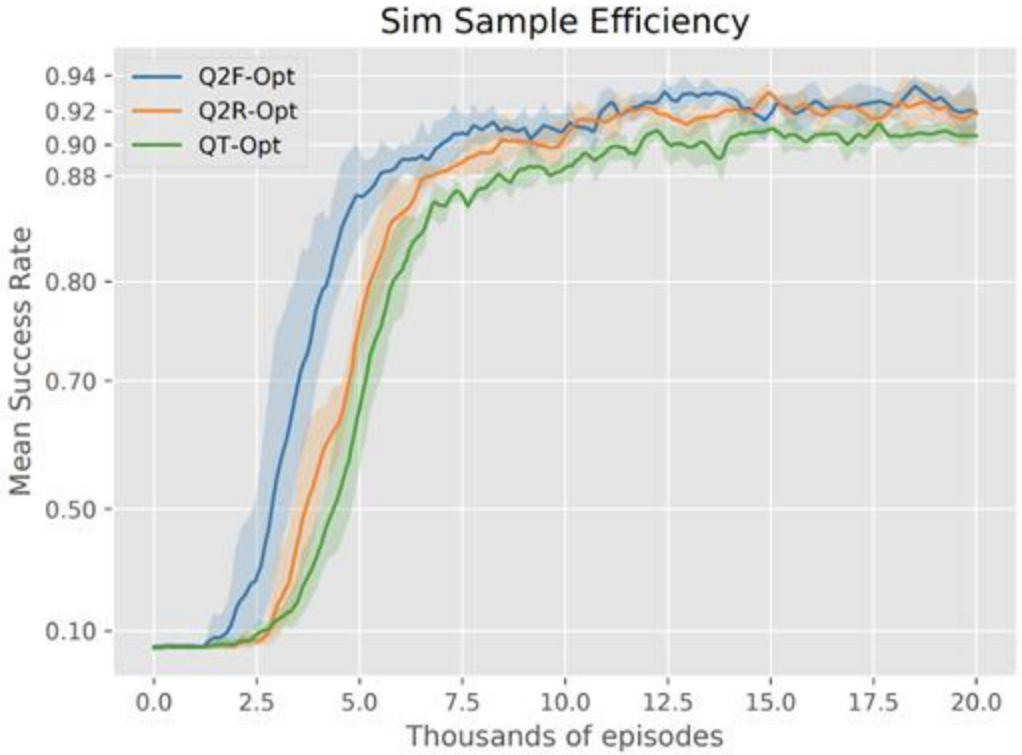


Fig. 4: Sim success rate as a function of the number of generated environment episodes. The distributional methods are significantly more sample efficient than QT-Opt.

Discussion

- Q2T-Opt Learns faster than QT-Opt, even at the later stages of learning

Thanks!

Videos here:
https://sites.google.com/view/qtopt