Self-supervision

Saurabh Gupta

Solving a RL Problem

Better Reward Signals

Better Optimization

Convert into a Supervised Training Problem

Solve a Related but Supervision-rich Problem

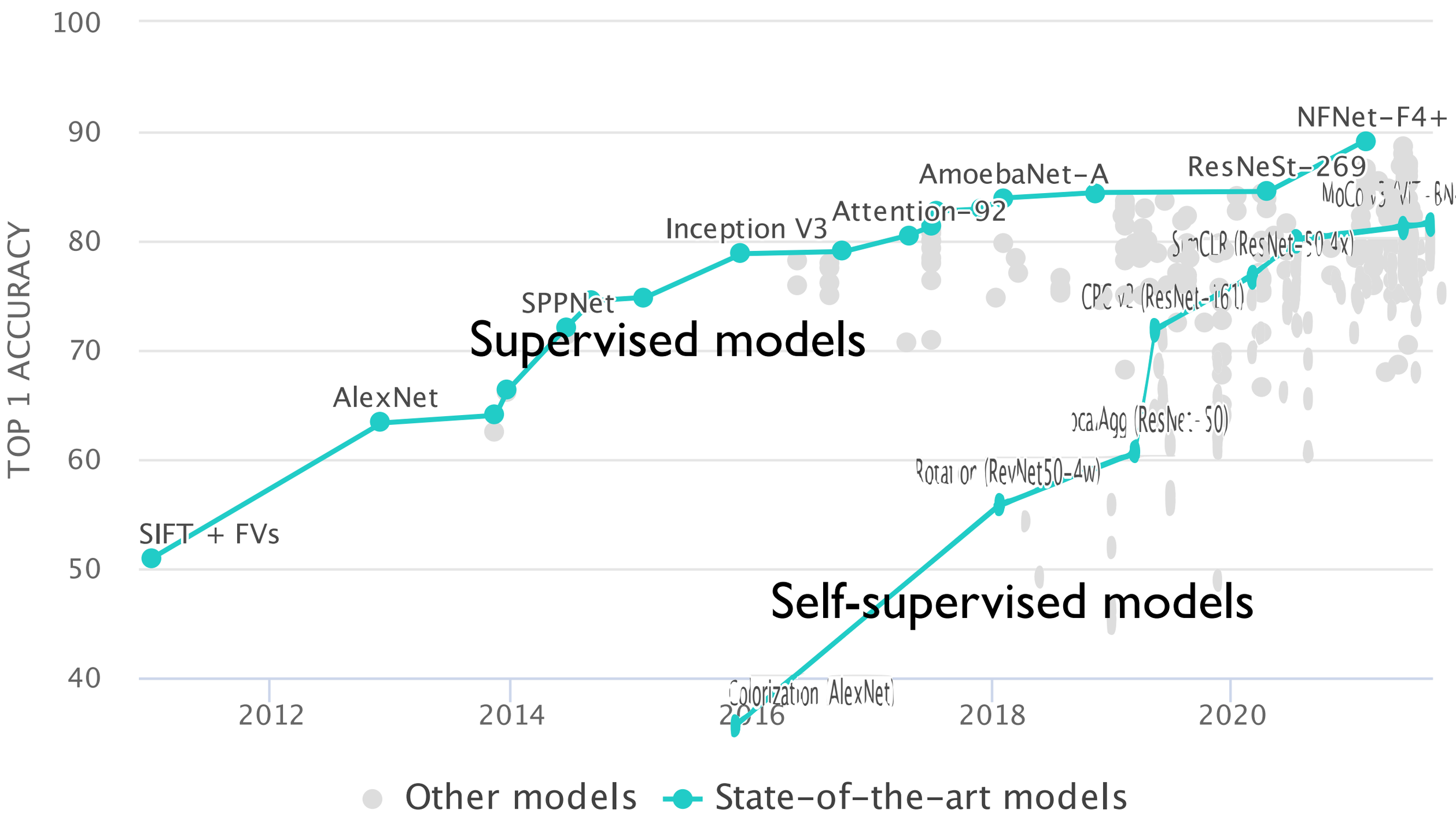
Build Models and Plan with Them

Model-free RL with sparse rewards

Known reward, known model. Model-based RL

In computer vision,

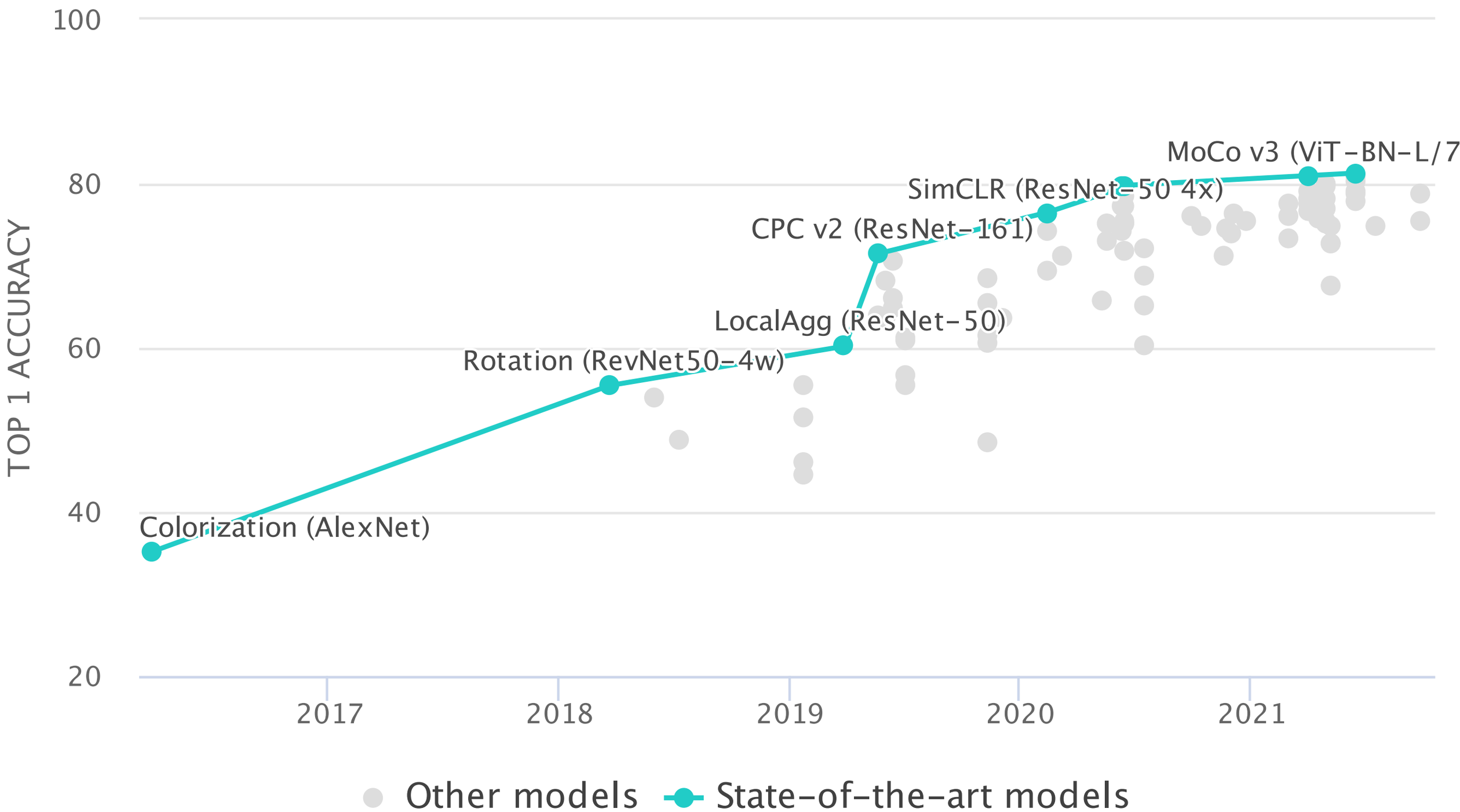
Solve a related but easy to supervise problem



Source: https://paperswithcode.com/

In computer vision,

Solve a related but easy to supervise problem

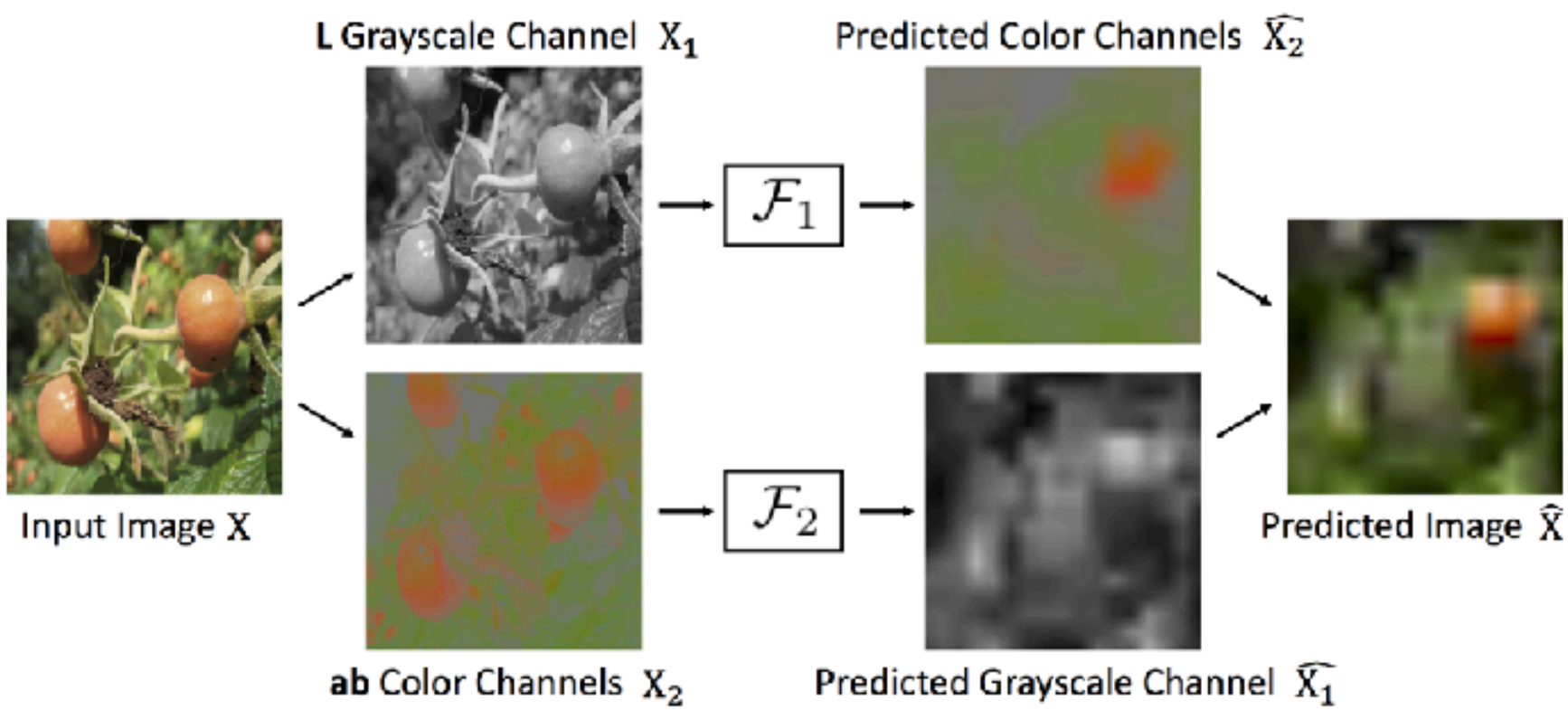


Source: https://paperswithcode.com/

In computer vision,

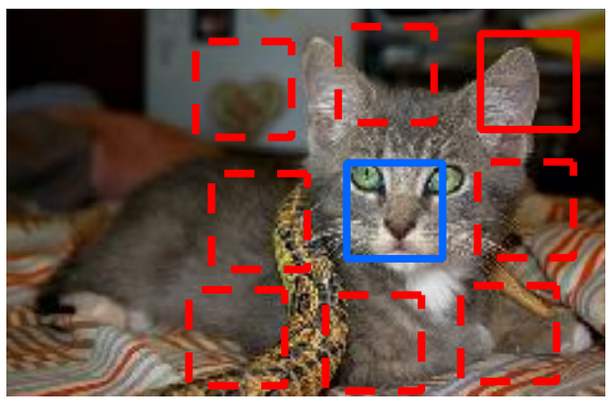
Solve a related but easy to supervise problem

- Collecting large-scale labeled datasets is a bottleneck
- Can we train models without relying on semantic supervision?
- Kind of. Supervise using *pretext tasks.*
  - *Pretext task:* easy to supervise, yet lead to good learning
- Eg: colorization:



In computer vision,

Solve a related but easy to supervise problem



Images

Videos

Across modalities

Solve a Related but Supervision-rich Problem

Design setups that can allow rapid data gathering

Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours

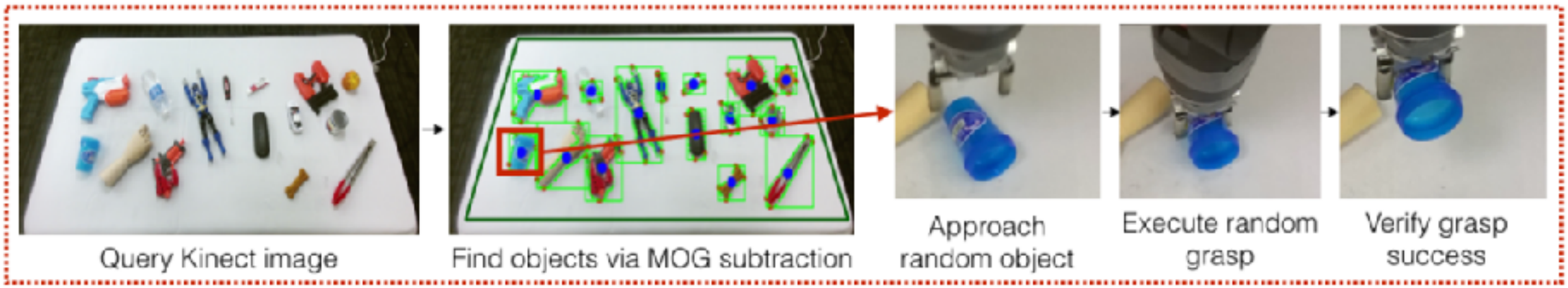
Lerrel Pinto and Abhinav Gupta
The Robotics Institute, Carnegie Mellon University
(lerrelp, abhinavg)@cs.cmu.edu

Problem Setup

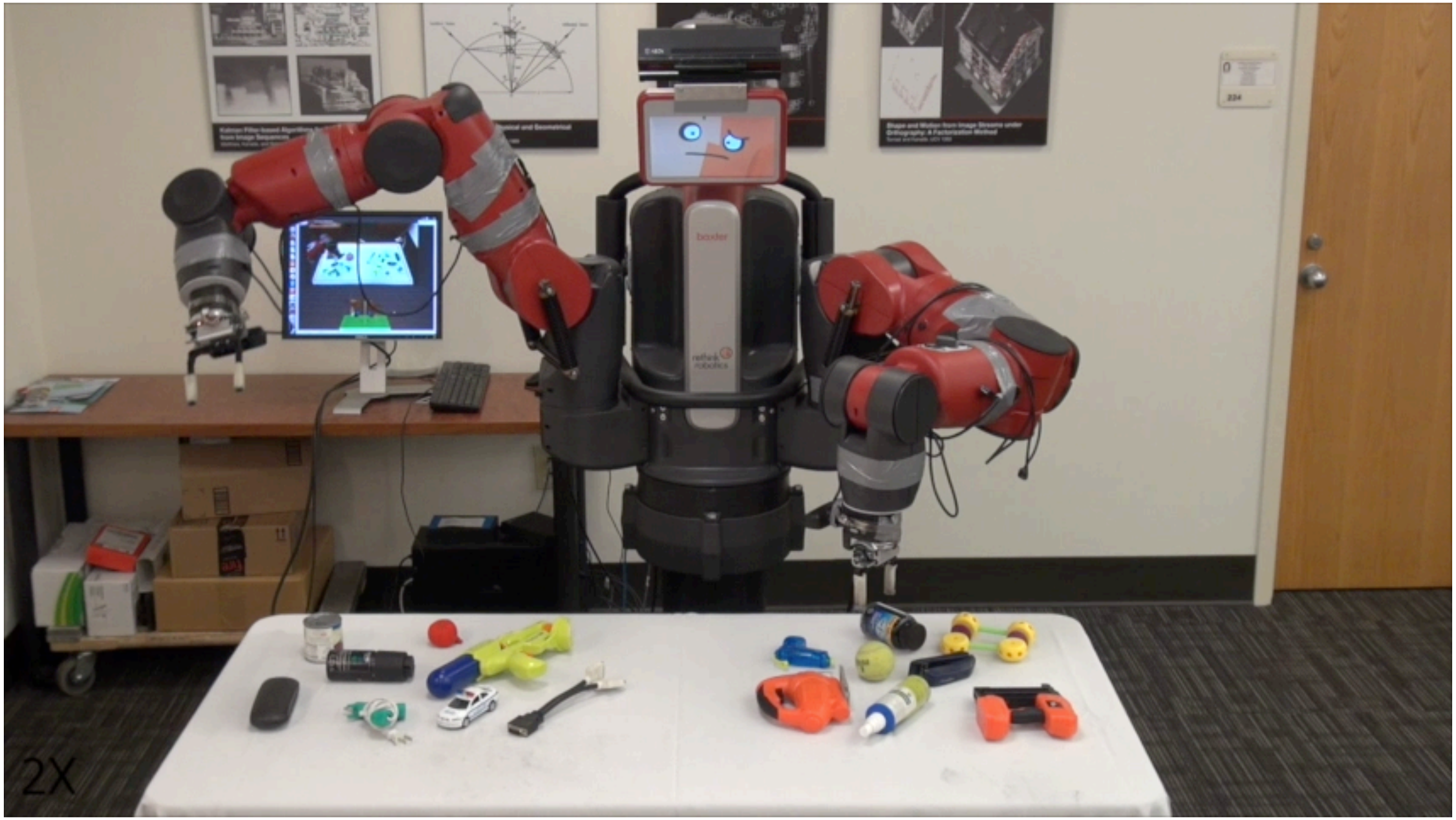
- Grasp objects on a table-top
- Known robot, unknown objects
- Planar grasping, parameterized by (x, y, θ).
- Observation space?
- Action space?

Data collection

- Autonomous data collection:
  - force closure on gripper indicates grasp success / failure.
- Collect data over 700 hours.

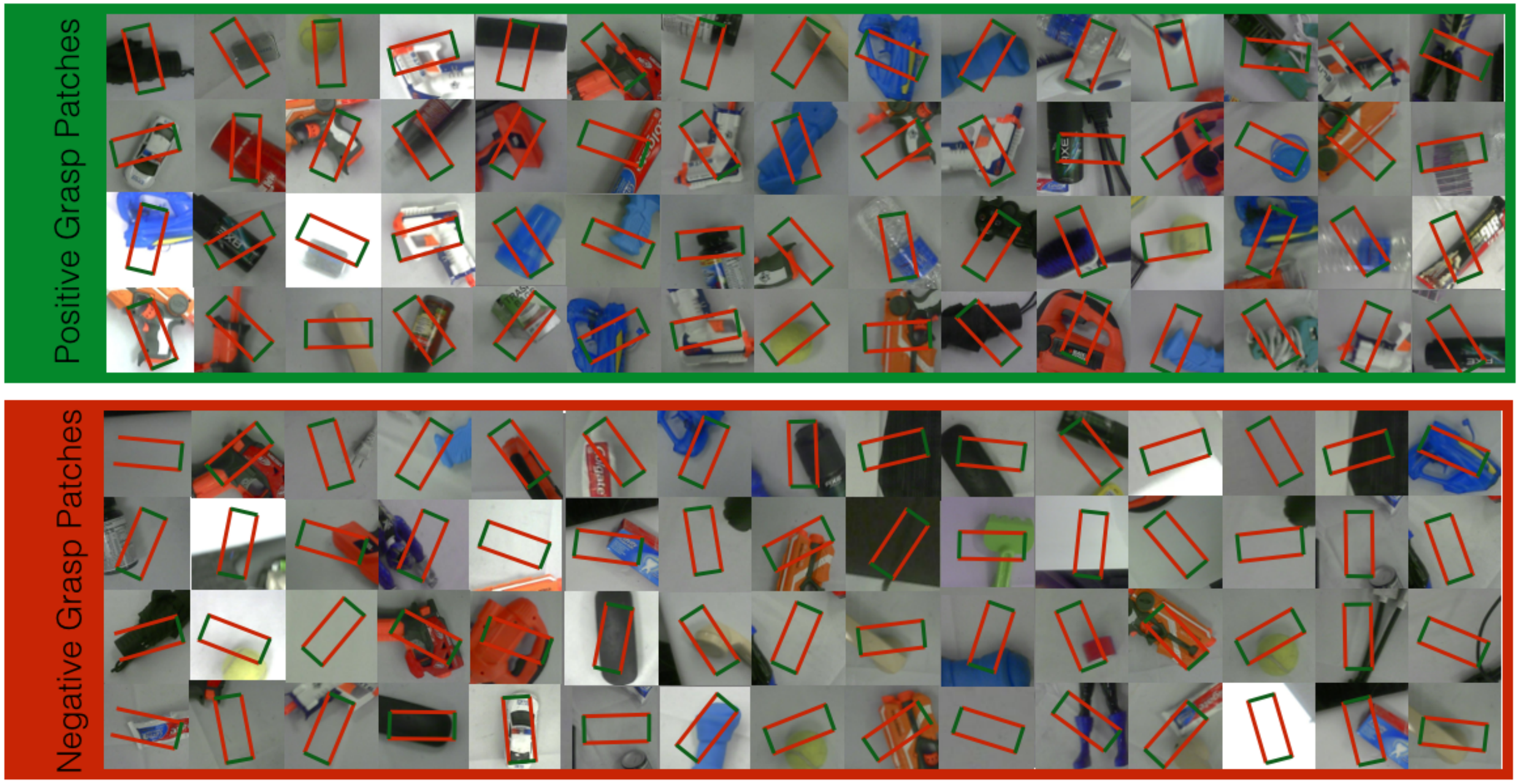


Autonomous data collection and labeling



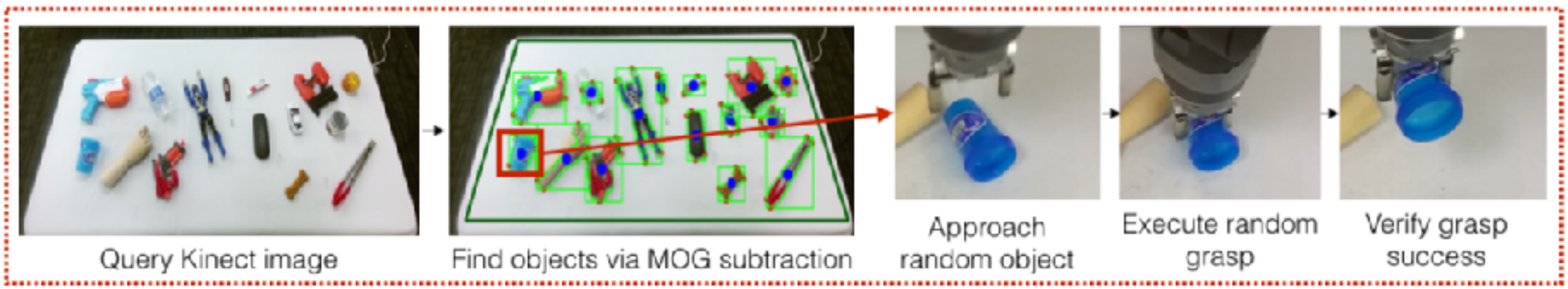
L. Pinto et al. **Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours.** ICRA 2016

Collected Data



L. Pinto et al. **Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours.** ICRA 2016

Data collection

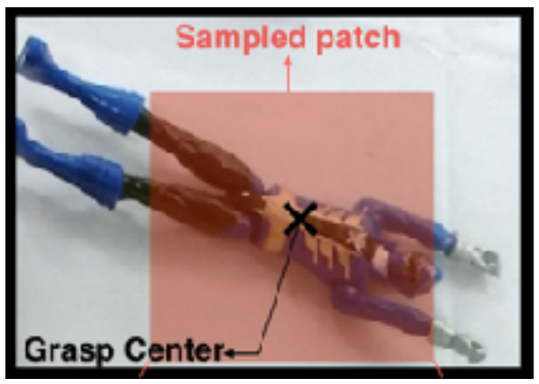
- Autonomous data collection:
  - force closure on gripper indicates grasp success / failure.
- Collect data over 700 hours.
- 

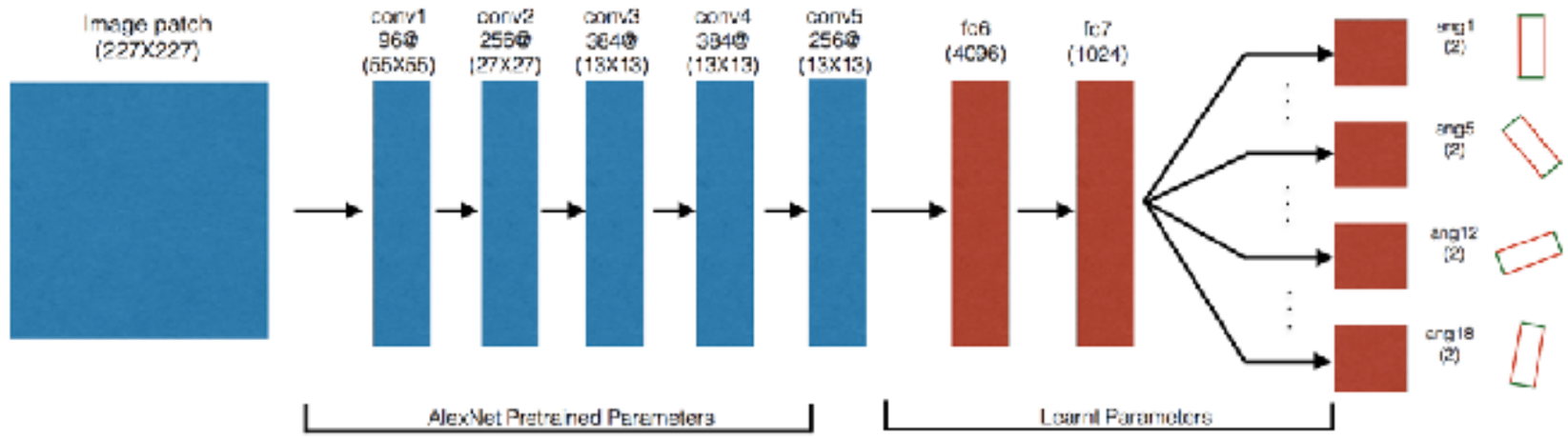
- Iterative data collection
-

Data Collection Type	Positive	Negative	Total	Grasp Rate
Random Trials	3,245	37,042	40,287	8.05%
Multi-Staged	2,807	4,500	7,307	38.41%
Test Set	214	2,759	2,973	7.19%
	6,266	**44,301**	**50,567**	

Training

- Train neural network for prediction grasp success rate





-

- $$L_B = \sum_{i=1}^{B} \sum_{j=1}^{N=18} \delta(j, \theta_i) \cdot \text{softmax}(A_{ji}, l_i)$$

Testing

- Off-line evaluation
  - measure graspability prediction on novel objects
- On-robot Grasping:
  - Grasp objects
  - Clearing clutter

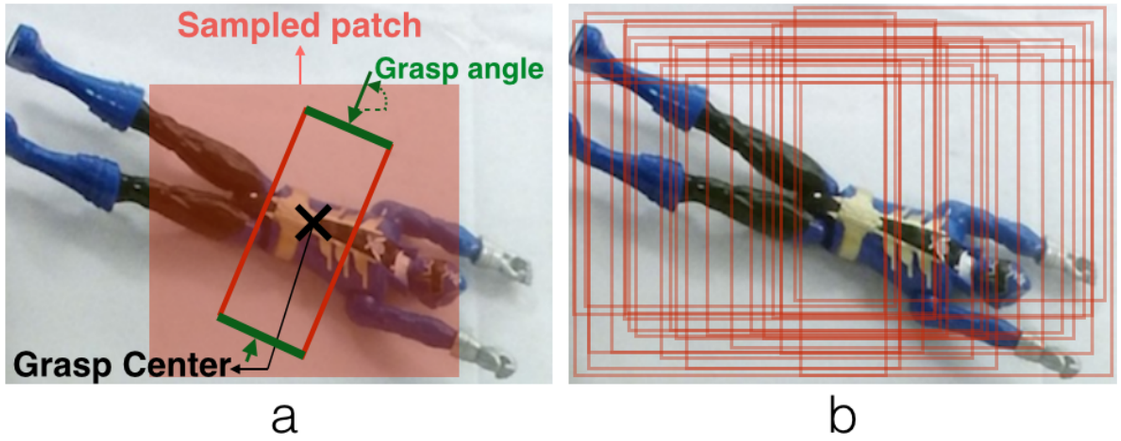


Fig. 3. (a) We use 1.5 times the gripper size image patch to predict the grasp-ability of a location and the angle at which it can be grasped. Visualization for showing the grasp location and the angle of gripper for grasping is derived from [8]. (b) At test time we sample patches at different positions and choose the top graspable location and corresponding gripper angle.

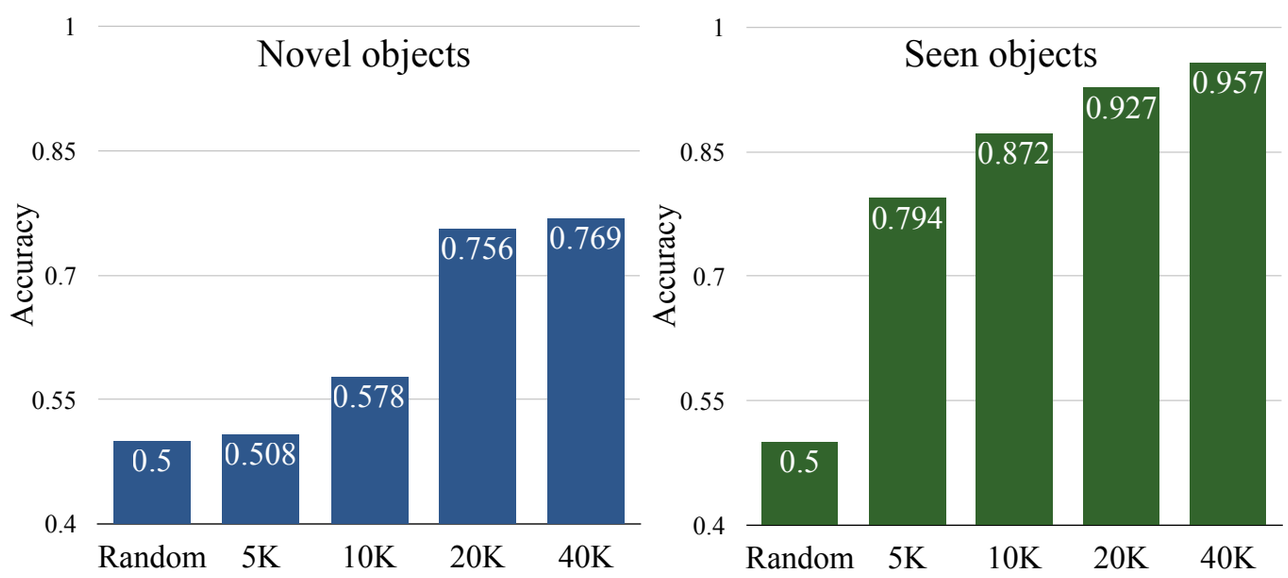
- ●

Off-line Testing

- Comparison to heuristic-methods

	Heuristic			Learning based			
	Min eigenvalue	Eigenvalue limit	Optimistic param. select	kNN	SVM	Deep Net (ours)	Deep Net + Multi-stage (ours)
Accuracy	0.534	0.599	0.621	0.694	0.733	0.769	**0.795**

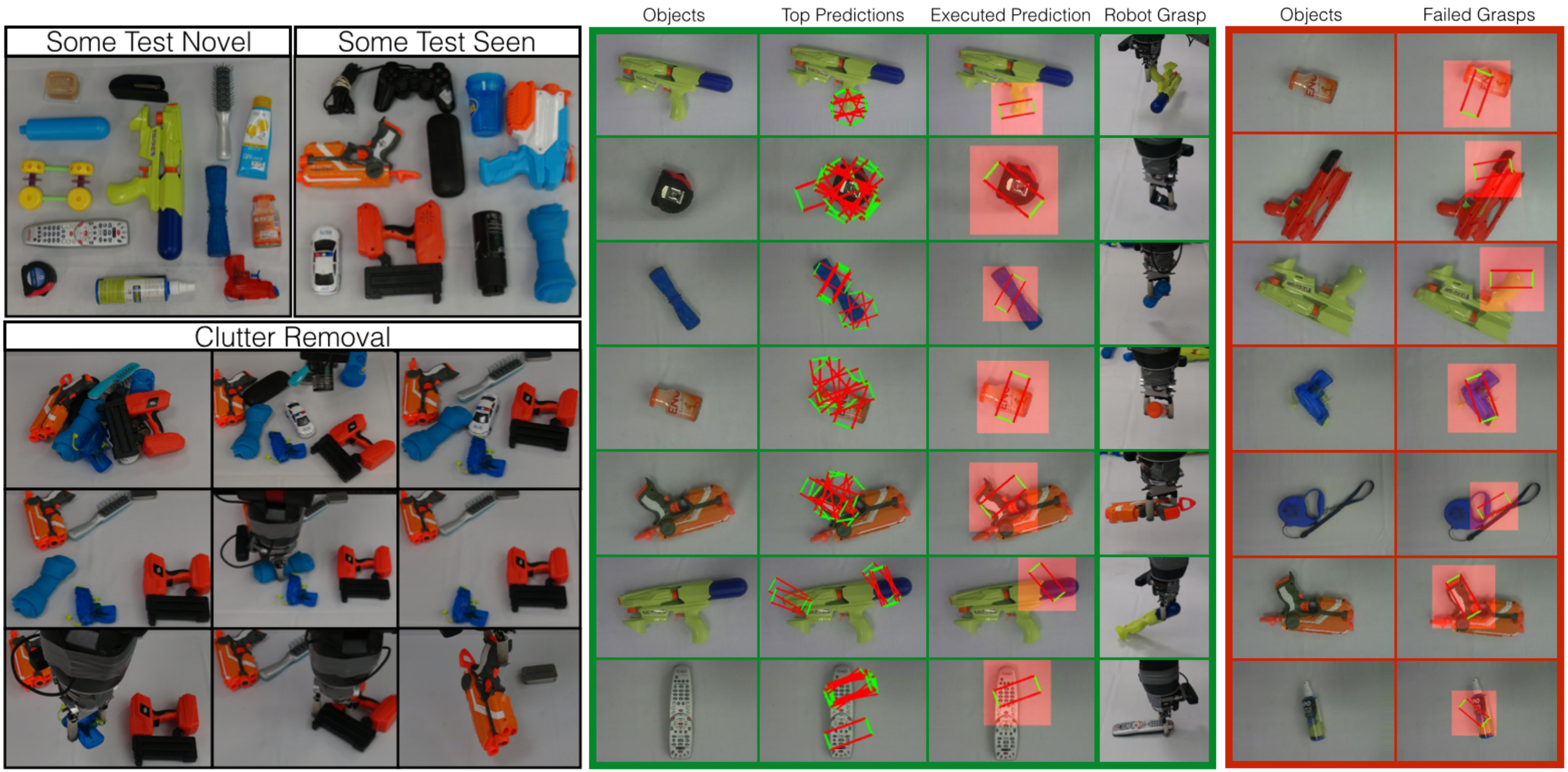
- Effect of amount of training data:



- Multi-stage data-gathering helps: accuracy goes up from 76.9 to 79.5.

On-line Testing

- Grasping rate for novel objects: 66%
- Grasping rate for seen objects: 73%
- Clutter removal: 10 object clutter, 26 interactions to clear clutter

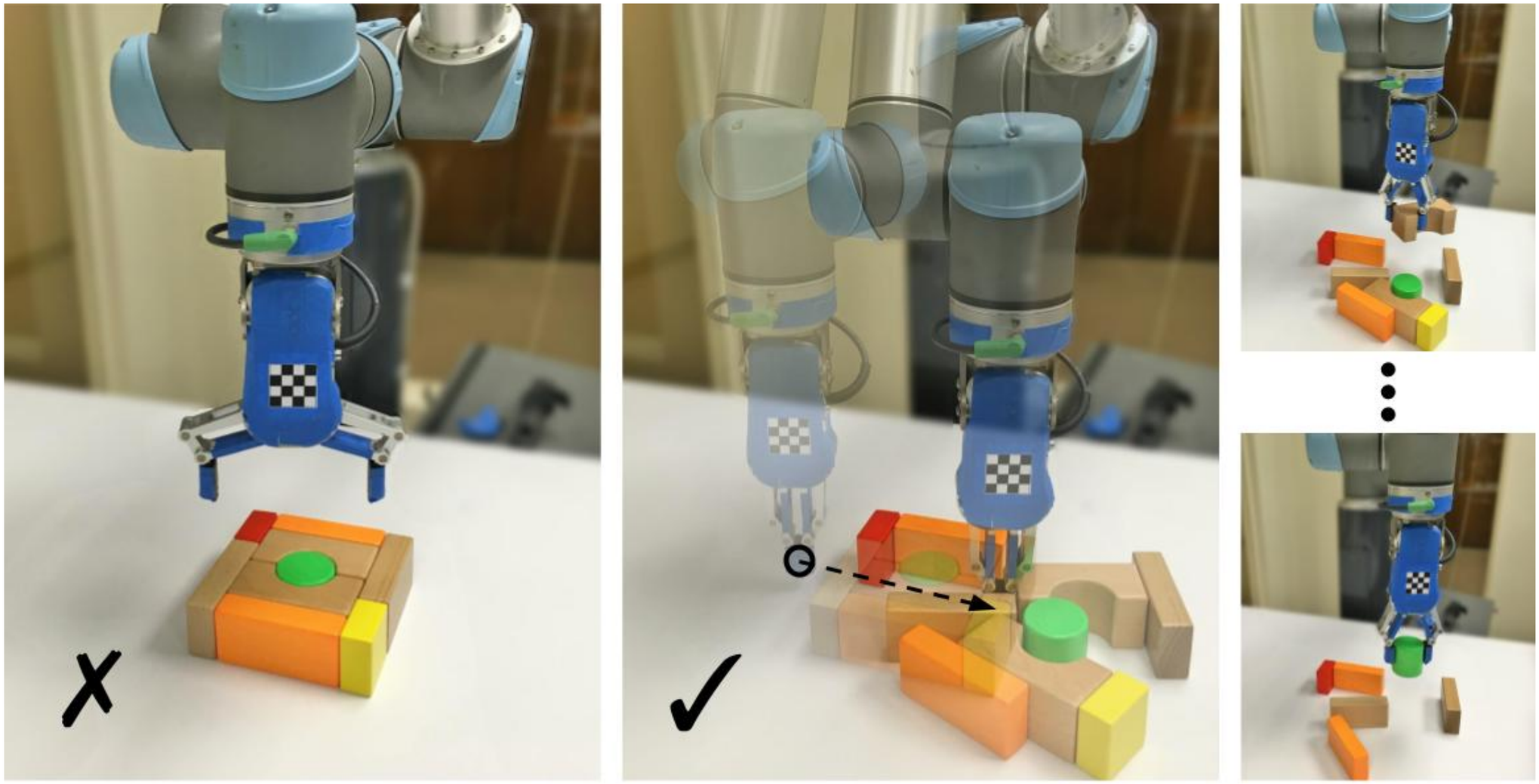


Learning Synergies between Pushing and Grasping with Self-supervised Deep Reinforcement Learning

Andy Zeng[1,2], Shuran Song[1,2], Stefan Welker[2], Johnny Lee[2], Alberto Rodriguez[3], Thomas Funkhouser[1,2]

[1]Princeton University [2]Google [3]Massachusetts Institute of Technology

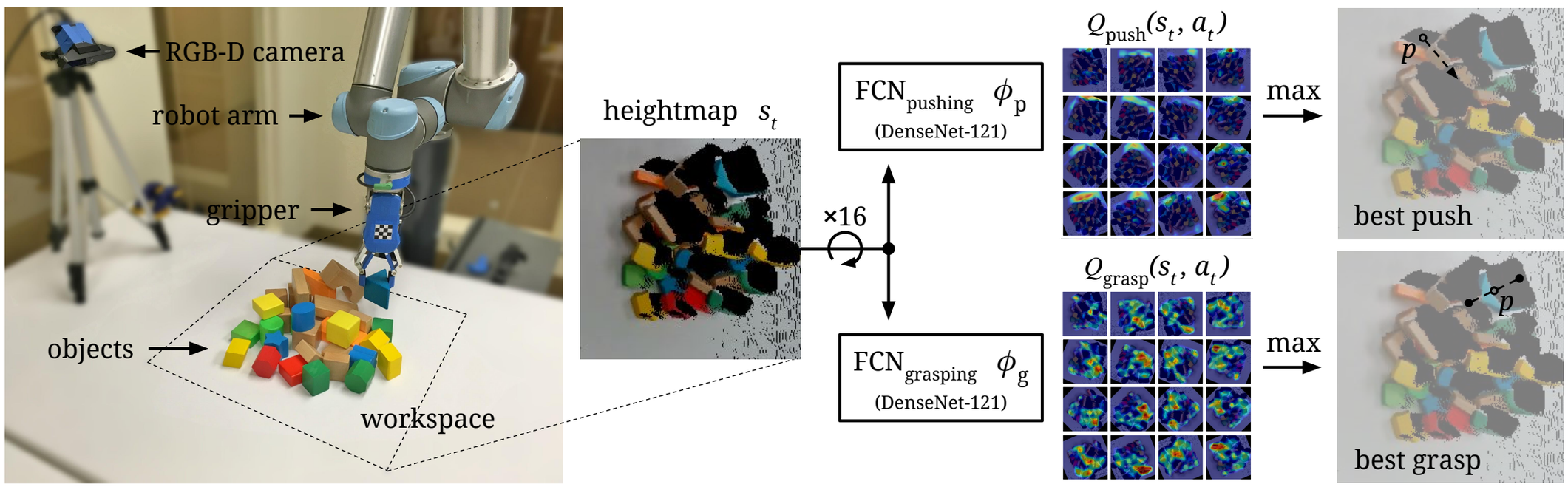
http://vpg.cs.princeton.edu



Summary

- Learning of pushing motion that is useful for grasping
- Clever parameterization of Q-functions
- Thorough evaluation of design choices in simulation

Q-functions

- Action space?
- Observation space?
- 

- Rewards:
  - R_g +1 is successfully grasped
  - R_p +0.5 if push caused change in scene (SSD)
  - gamma = 0.5

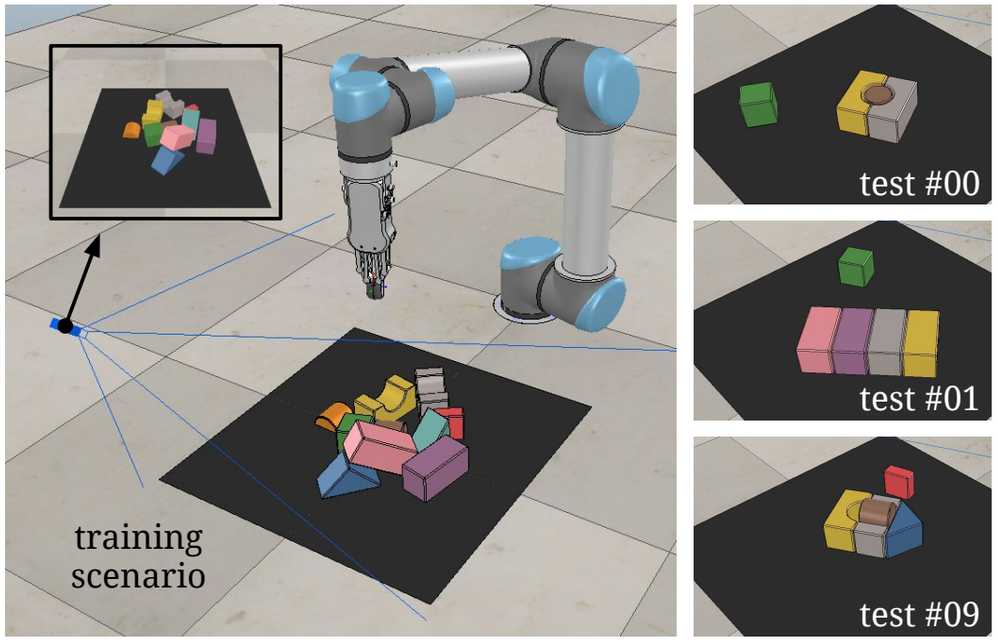
Simulation Results

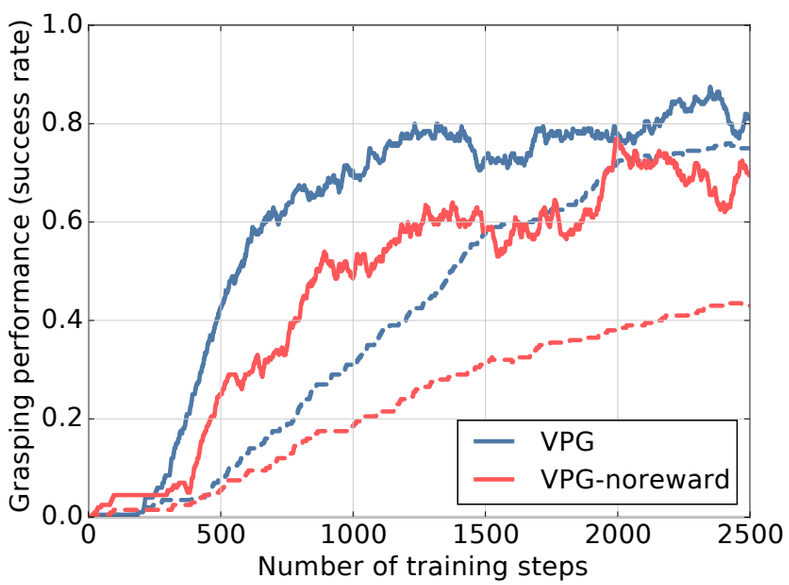
- Grasping random arrangements

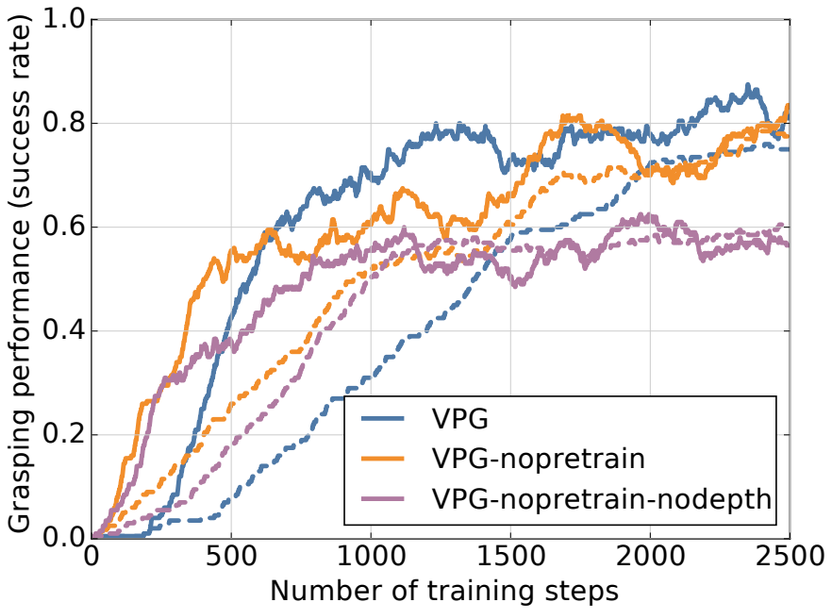
Method	Completion	Grasp Success	Action Efficiency
Grasping-only [8]	90.9	55.8	55.8
P+G Reactive	54.5	59.4	47.7
VPG	**100.0**	**67.7**	**60.9**

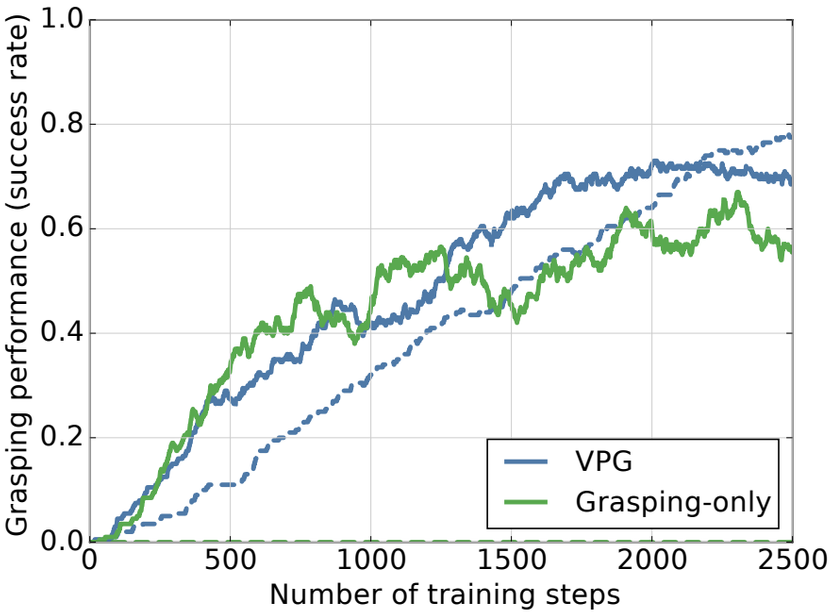
- Grasping adversarial arrangements

Method	Completion	Grasp Success	Action Efficiency
Grasping-only [8]	40.6	51.7	51.7
P+G Reactive	48.2	59.0	46.4
VPG	**82.7**	**77.2**	**60.1**



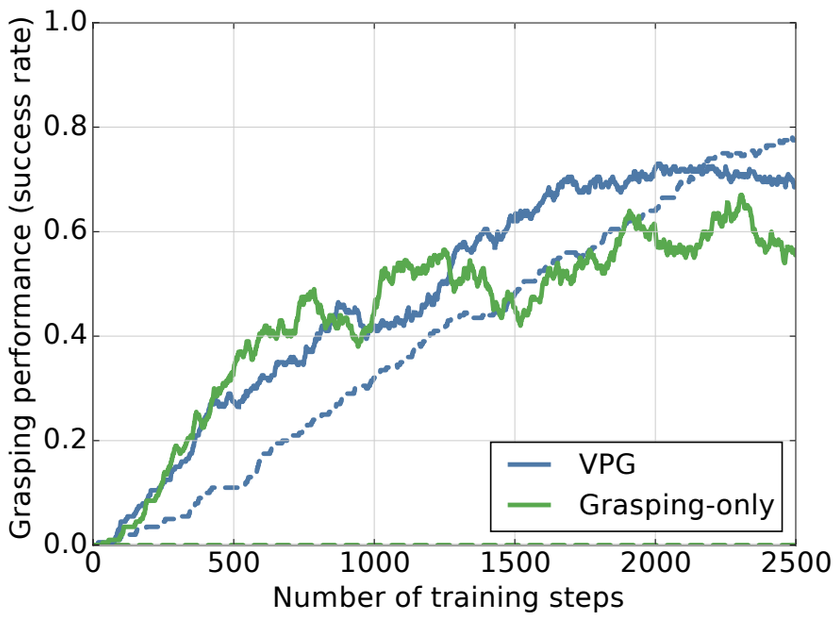


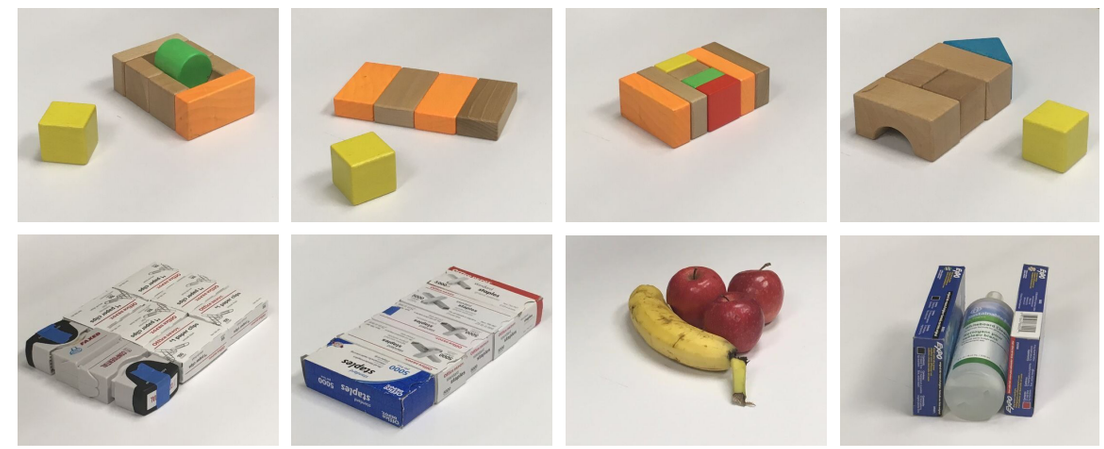




Method	Completion	Grasp Success	Action Efficiency
VPG-myopic	79.1	74.3	53.7
VPG	**82.7**	**77.2**	**60.1**

Real World Results





Method	Completion	Grasp Success	Action Efficiency
Grasping-only [8]	42.9	43.5	43.5
VPG	**71.4**	**83.3**	**69.0**

Solve a Related but Supervision-rich Problem

S. Levine et al. **Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection.** ISER 2017.

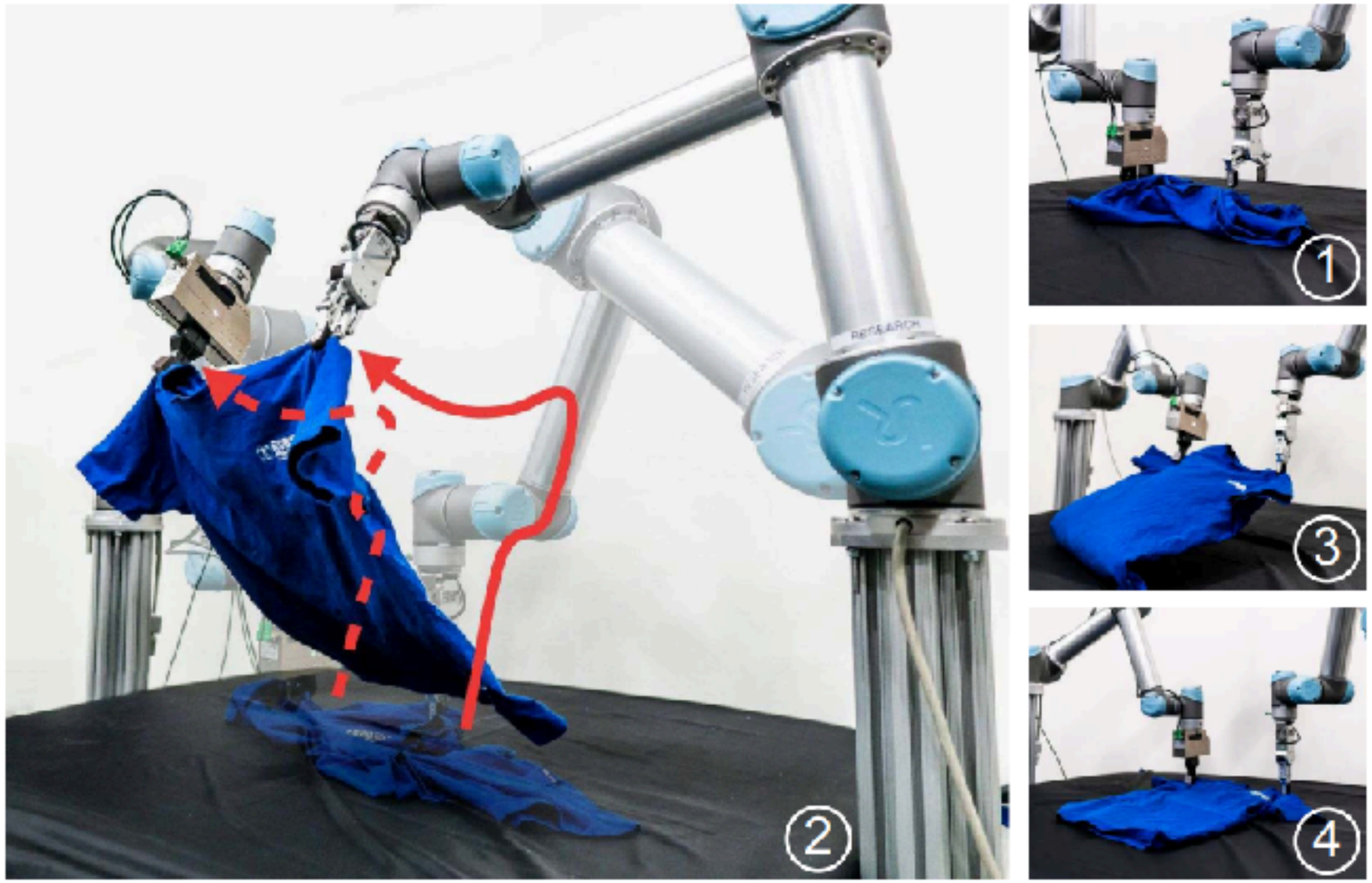
Solve a Related but Supervision-rich Problem

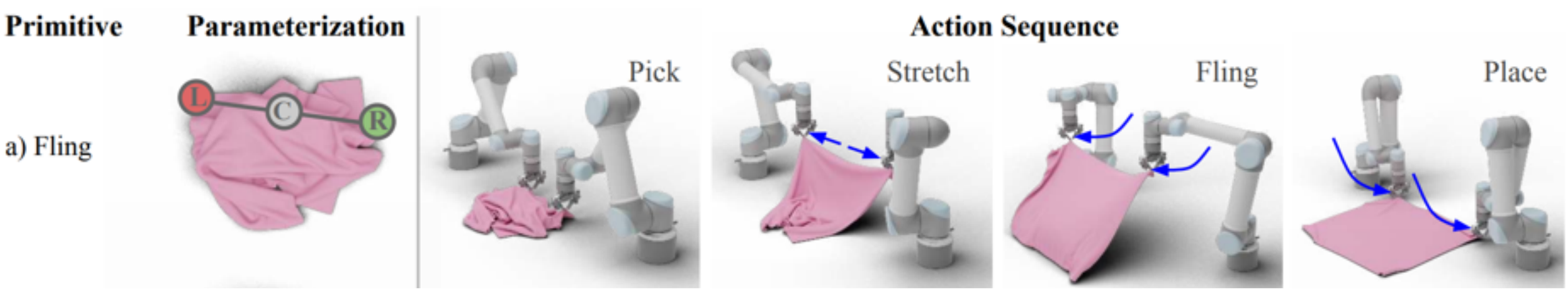
S. Levine et al. **Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection.** ISER 2017.

Solve a Related but Supervision-rich Problem

S. Levine et al. **Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection.** ISER 2017.

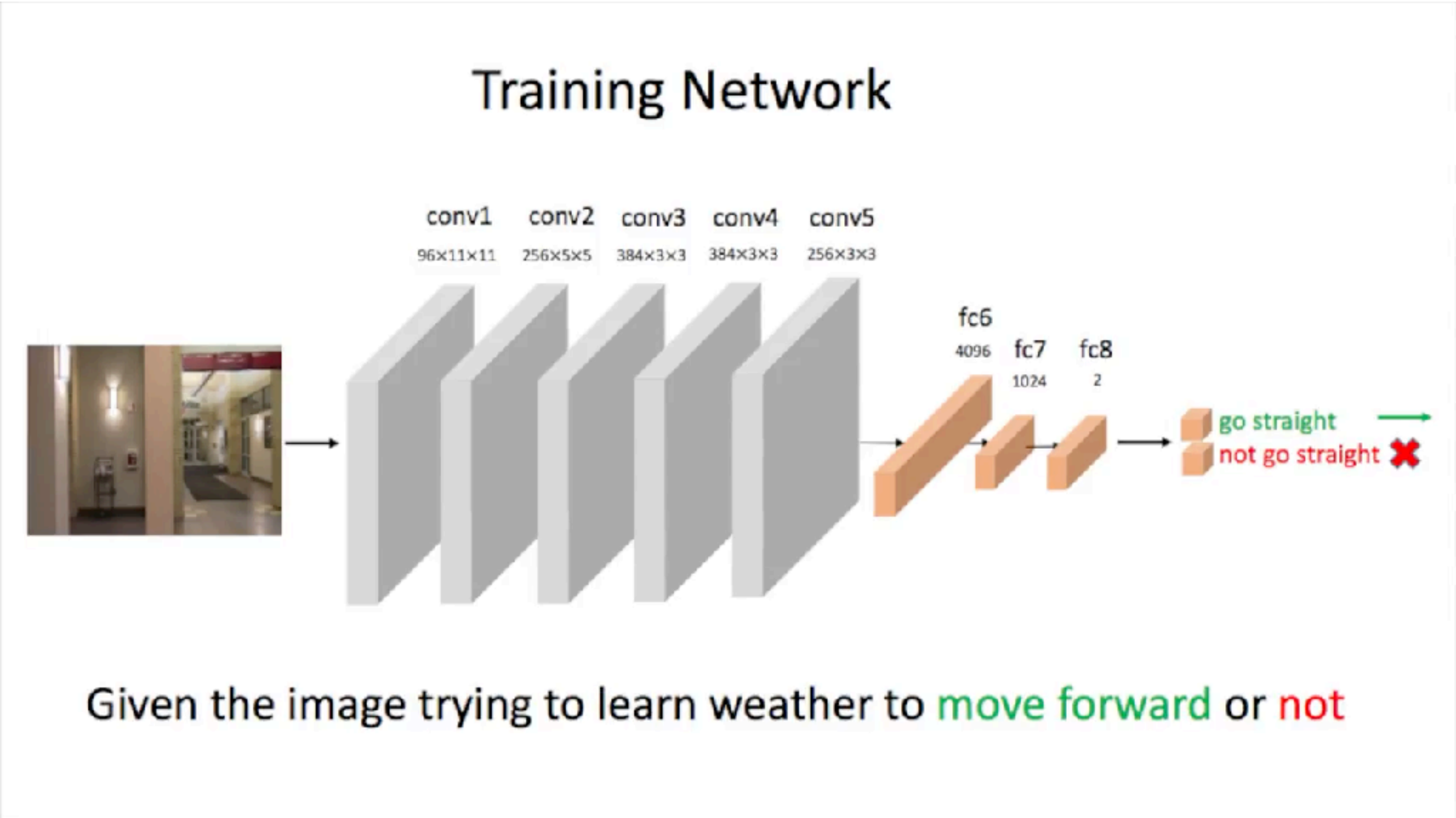
FlingBot





Ha and Song **FlingBot: The Unreasonable Effectiveness of Dynamic Manipulation for Cloth Unfolding.** CoRL 2021.

Solve a Related but Supervision-rich Problem



D. Gandhi et al. **Learning to Fly by Crashing.** IROS 2017.

Solve a Related but Supervision-rich Problem

D. Gandhi et al. **Learning to Fly by Crashing.** IROS 2017.

Solve a Related but Supervision-rich Problem

Design setups that can allow rapid data gathering

Discussion

- A) Why may we want to use learning for grasping?

- B) What are the limitations of the techniques that we discussed today? In what scenarios can / can't we use them?

Thank you