Manipulation

Saurabh Gupta

Learning to Compose Hierarchical Object-Centric Controllers for Robotic Manipulation

Mohit Sharma[†1]
Robotics Institute
Carnegie Mellon University

Jacky Liang[†1]
Robotics Institute
Carnegie Mellon University

Jialiang Zhao[1]
Robotics Institute
Carnegie Mellon University

Alex LaGrassa[1]
Robotics Institute
Carnegie Mellon University

Oliver Kroemer[1]
Robotics Institute
Carnegie Mellon University

- Learns to compose object-centric controllers for contact-rich manipulation tasks

Controllers

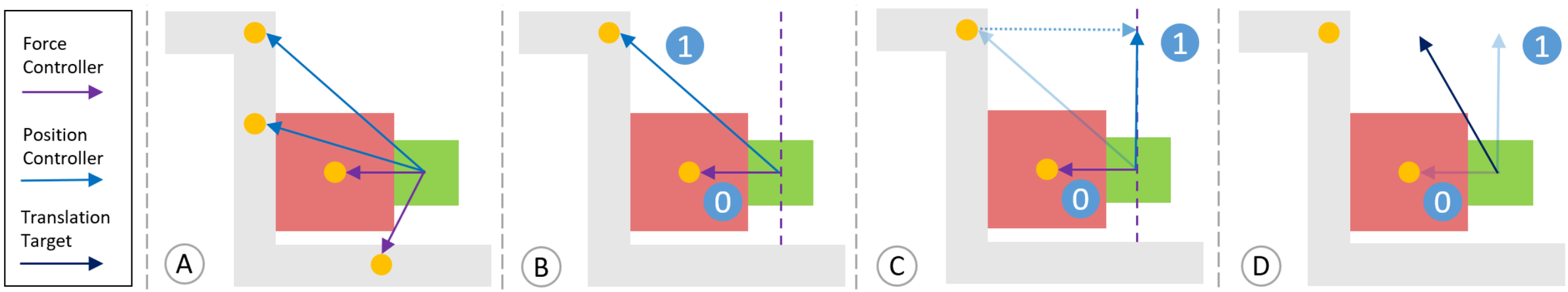
- End-effector controllers
  - Position controllers
  - Force controllers
  - Rotation controllers
  - Null controller
- Compute joint torques using task-space impedance control:
  - $\tau = J^\top (K_S \Delta + K_D \dot{\Delta})$

Controller Composition

- Priority order, apply lower priority controllers in the null-space of the higher-priority controllers

$$\Delta_x^0 = K_x \delta_x(x_d^0, u^0, x_c)$$
$$\Delta_x^1 = K_x \mathcal{N}([u^0]) \delta_x(x_d^1, u^1, x_c)$$
$$\Delta_x^2 = K_x \mathcal{N}([u^0, u^1]) \delta_x(x_d^2, u^2, x_c)$$

- $$\Delta_x = \sum_{i=0}^{2} \Delta_x^i$$

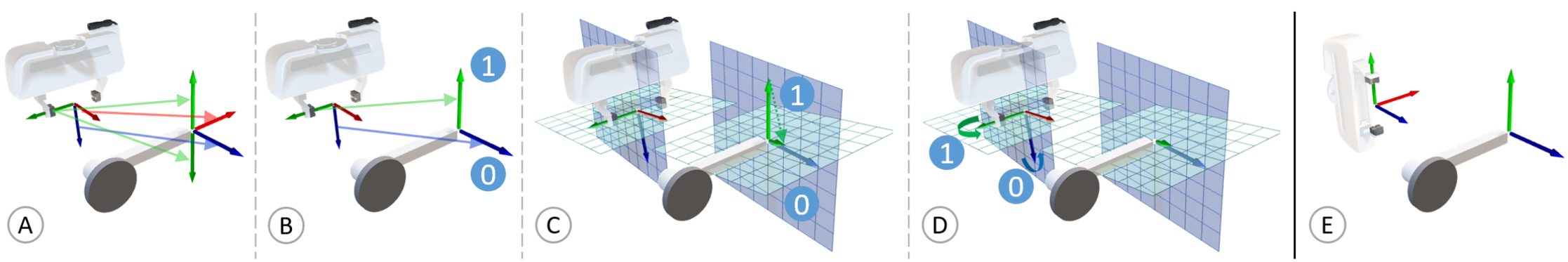
- 

Controller Composition (Rotation)

- Rotation controllers try to align a one axis to a given target axis
- Priority order, apply lower priority controllers in the null-space of the higher-priority controllers

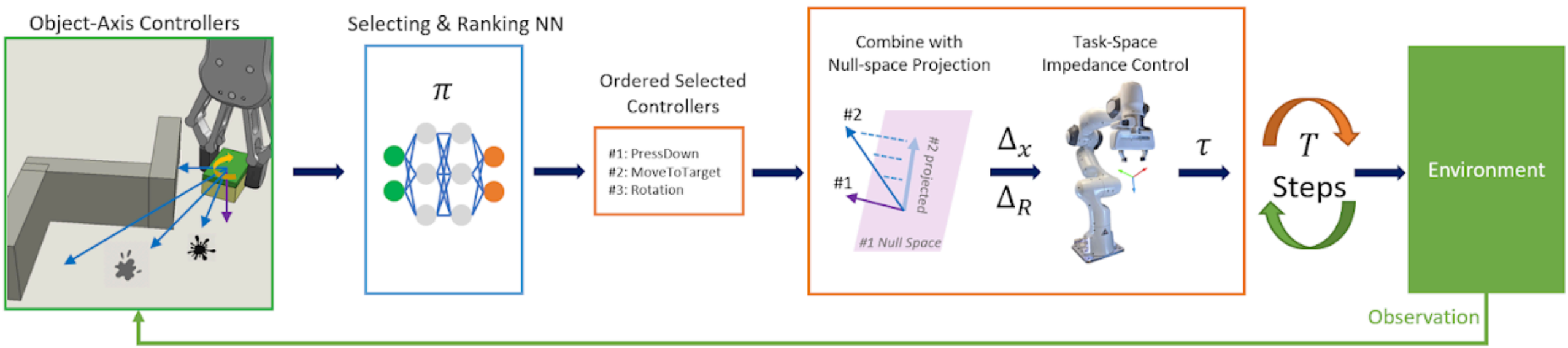
$$\Delta_R^0 = K_R \delta_R(r_d^0, u^0, R_c)$$
$$\Delta_R^1 = K_R \delta_R(\mathcal{N}([R_c u^0]) r_d^1, u^1, \mathcal{N}([R_c u^0]) R_c)$$
$$\Delta_R = \Delta_R^1 \circ \Delta_R^0$$

-
-

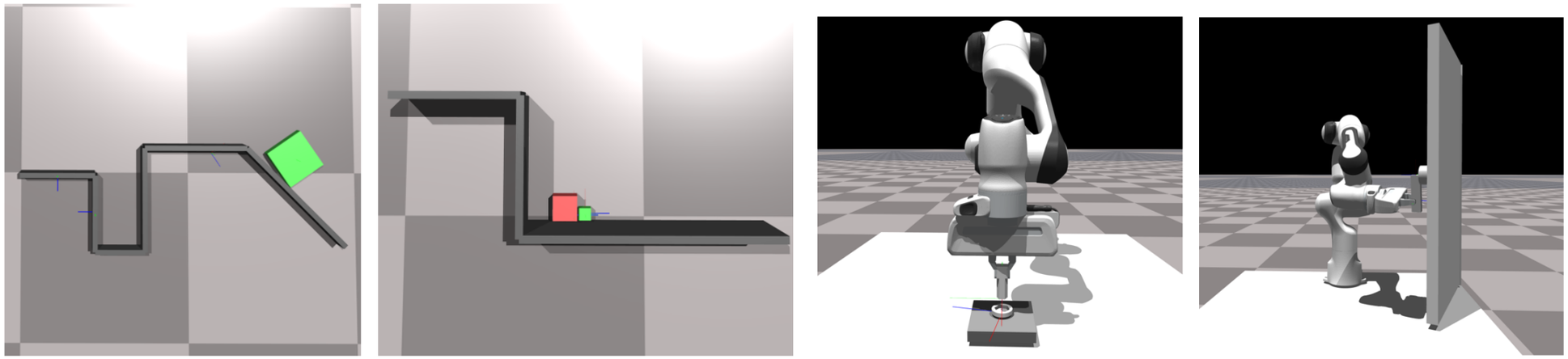


Learn a policy to compose basic controllers for solving a task

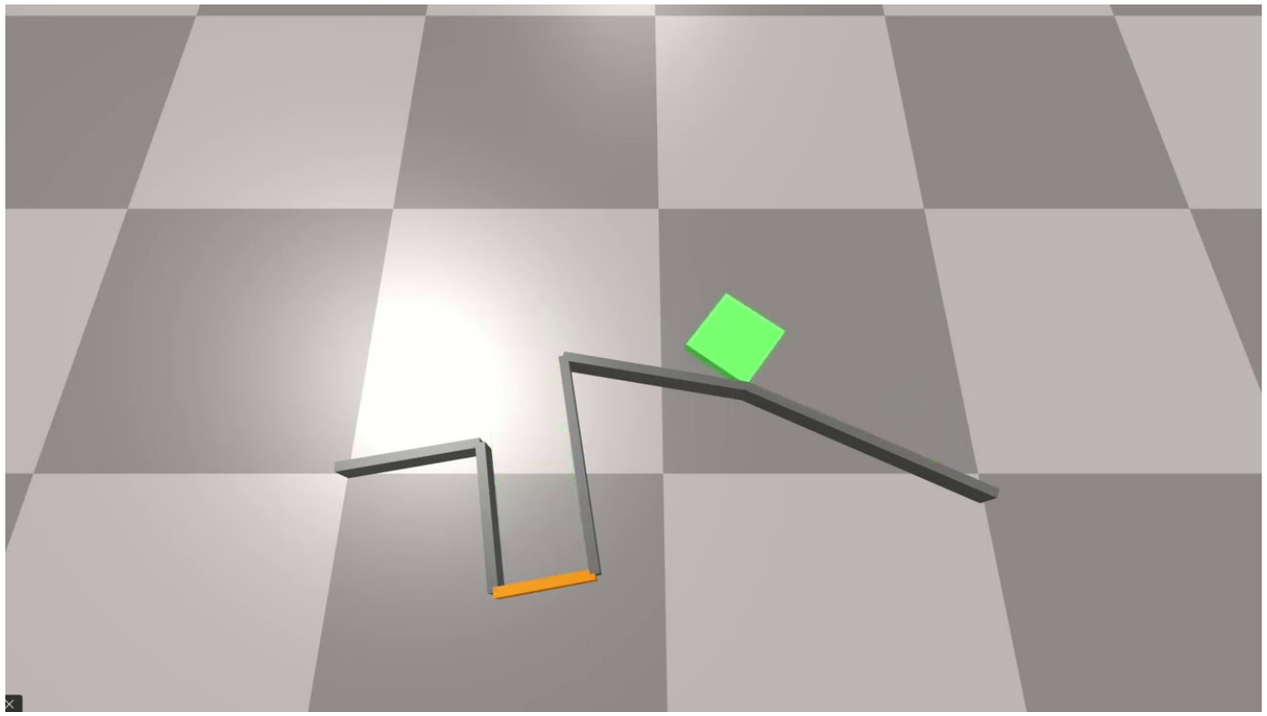
Overall architecture

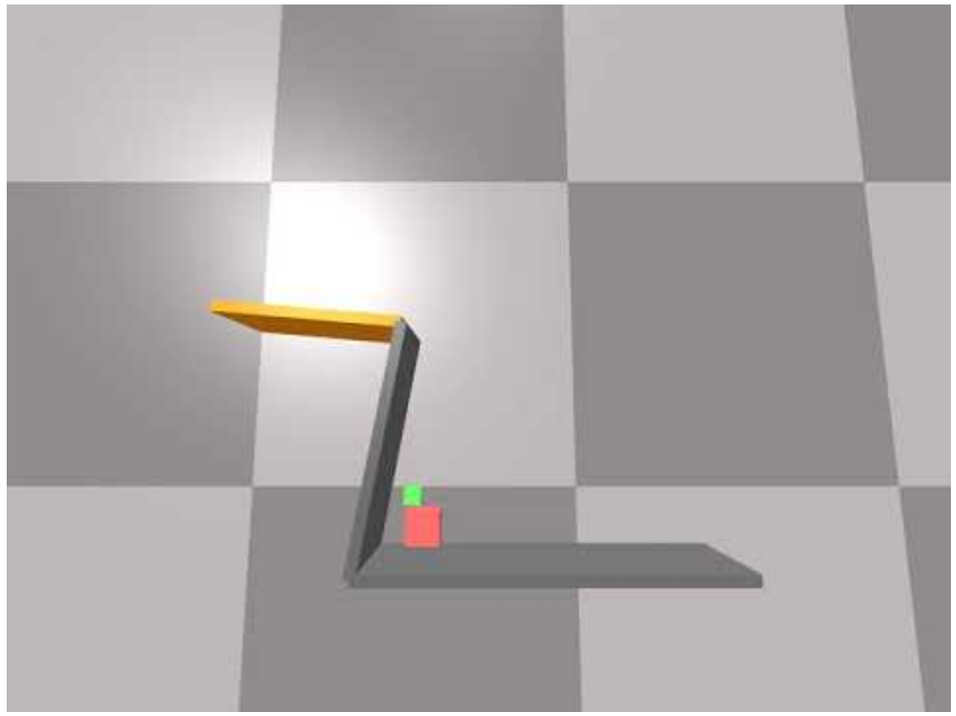


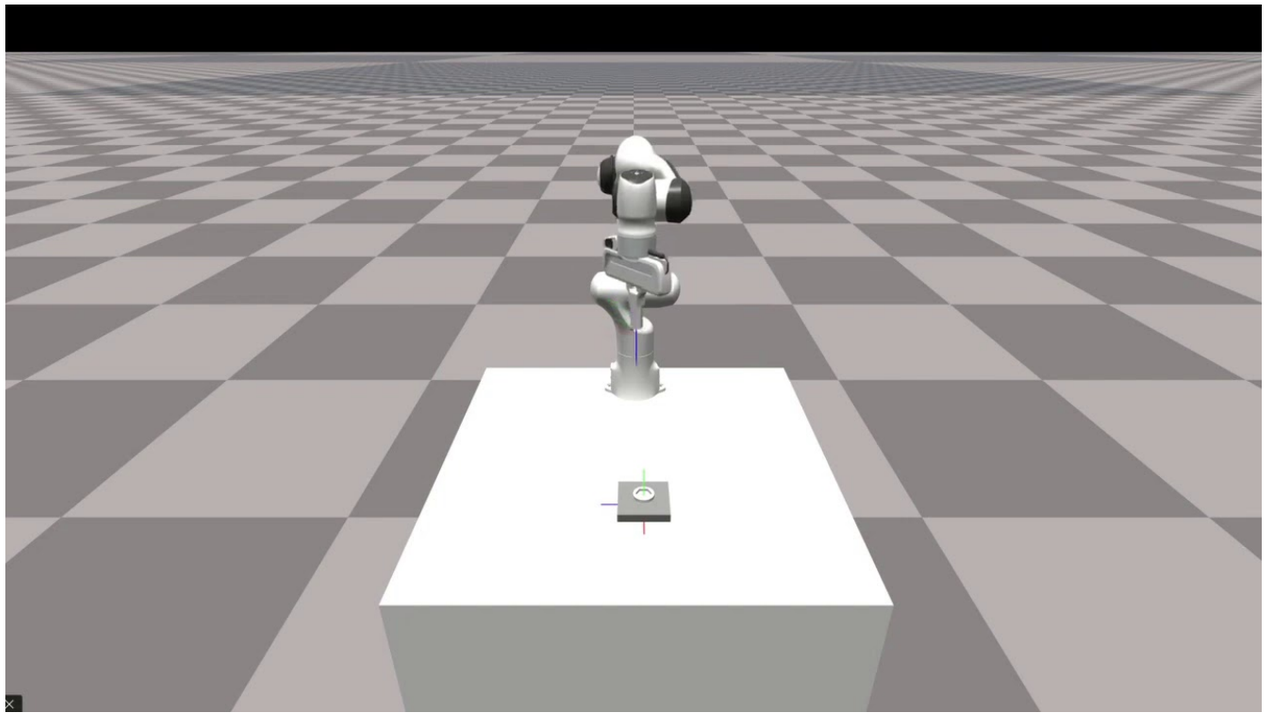
Experiments

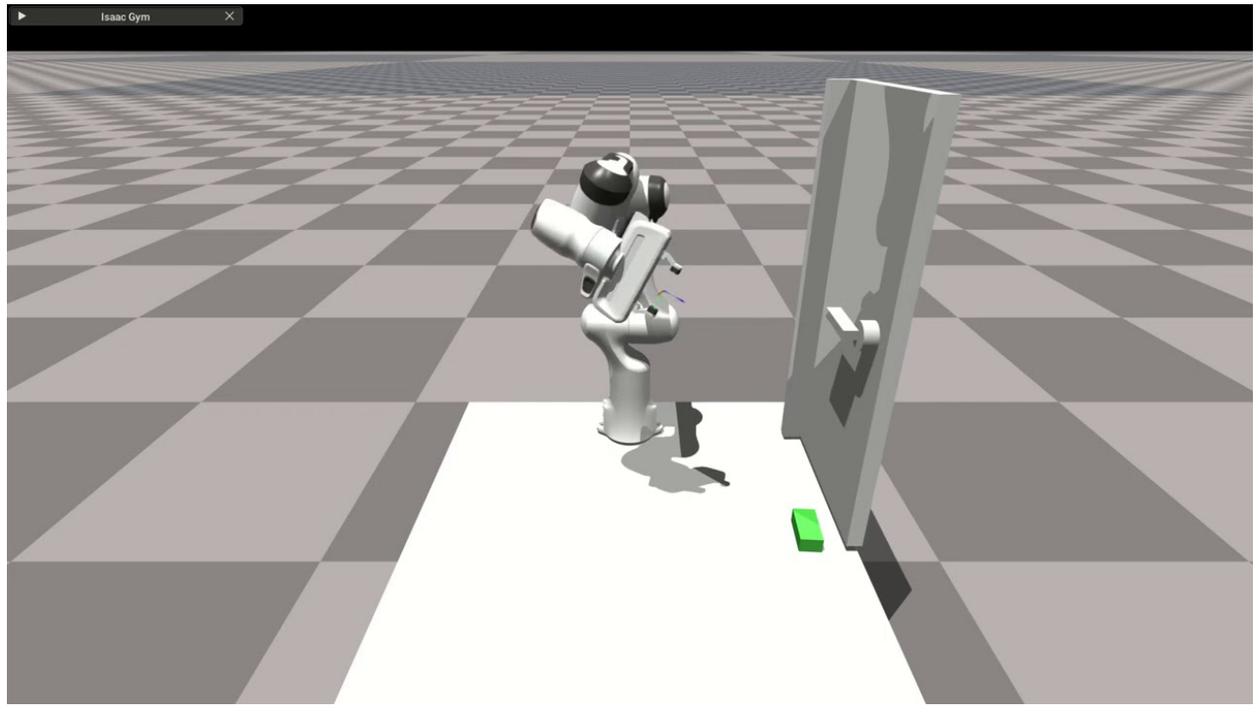
- Tasks
  - 
- Baselines:
  - EE control, 1-Controller at a time
- Ablations:
  - Discrete action composition
  - Continuous priority scores
  - Expanded MDP (Multi 1hot, Single 1 hot)

Tasks









Results

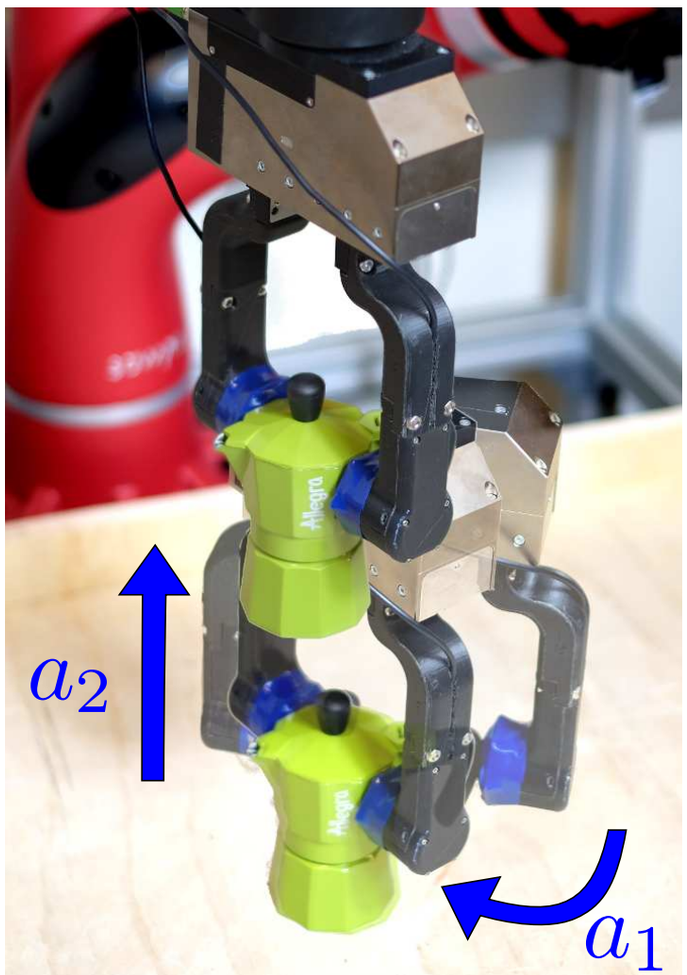
Task	Variation	EE-Space	1-Ctrlr	3-Priority	3-Combo	3-Exp-Single	3-Exp-Multi
Block Fit	Train	0.87 (0.213)	0.778 (0.38)	0.936 (0.032)	0.294 (0.18)	0.998 (0.002)	**1.00 (0.0)**
	Test-Small	0.87 (0.10)	0.916 (0.14)	**0.99 (0.001)**	0.184 (0.12)	**0.99 (0.001)**	**0.99 (0.01)**
	Test-Large	0.371 (0.246)	0.396 (0.423)	0.877 (0.141)	0.165 (0.23)	**0.974 (0.048)**	0.953 (0.087)
Block Push	Train	0.966 (0.046)	0.594 (0.087)	0.548 (0.129)	0.0 (0.0)	0.974 (0.025)	**0.978 (0.022)**
	Test-Small	0.912 (0.045)	0.577 (0.193)	0.396 (0.041)	0.0 (0.0)	0.945 (0.045)	**0.960 (0.030)**
	Test-Large	0.518 (0.185)	0.152 (0.137)	0.376 (0.032)	0.0 (0.0)	0.751 (0.103)	**0.788 (0.132)**

Task	Variation	EE-Space	1-Ctrlr	3-Priority	3-Combo	3-Exp-Single	3-Exp-Multi
Hex-Screw	Train	0.002 (0.002)	0.183 (0.303)	0.960 (0.048)	0.774 0.194)	**0.984 (0.01)**	0.980 (0.016)
	Test-Small	0.00 (0.00)	0.13 (0.072)	0.62 (0.045)	0.429 (0.430)	0.963 (0.01)	**0.966 (0.015)**
	Test-Large	0.00 (0.00)	0.026 (0.025)	0.633 (0.081)	0.34 (0.057)	**0.936 (0.028)**	**0.936 (0.035)**
	Real-World	n/a	0.0	0.5	0.0	**0.9**	0.6
Door-Open	Train	0.002 (0.006)	0.947 (0.021)	0.982 (0.007)	0.984 (0.013)	**0.987 (0.009)**	0.984 (0.015)
	Test-Small	0.066 (0.063)	0.922 (0.043)	0.965 (0.046)	0.975 (0.011)	**0.997 (0.006)**	0.992 (0.015)
	Test-Large	0.000 (0.001)	0.936 (0.032)	0.983 (0.006)	0.985 (0.007)	**0.996 (0.005)**	0.994 (0.013)
	Real-World	n/a	0.0	**1.0**	0.9	**1.0**	**1.0**

- Videos: https://sites.google.com/view/compositional-object-control/videos

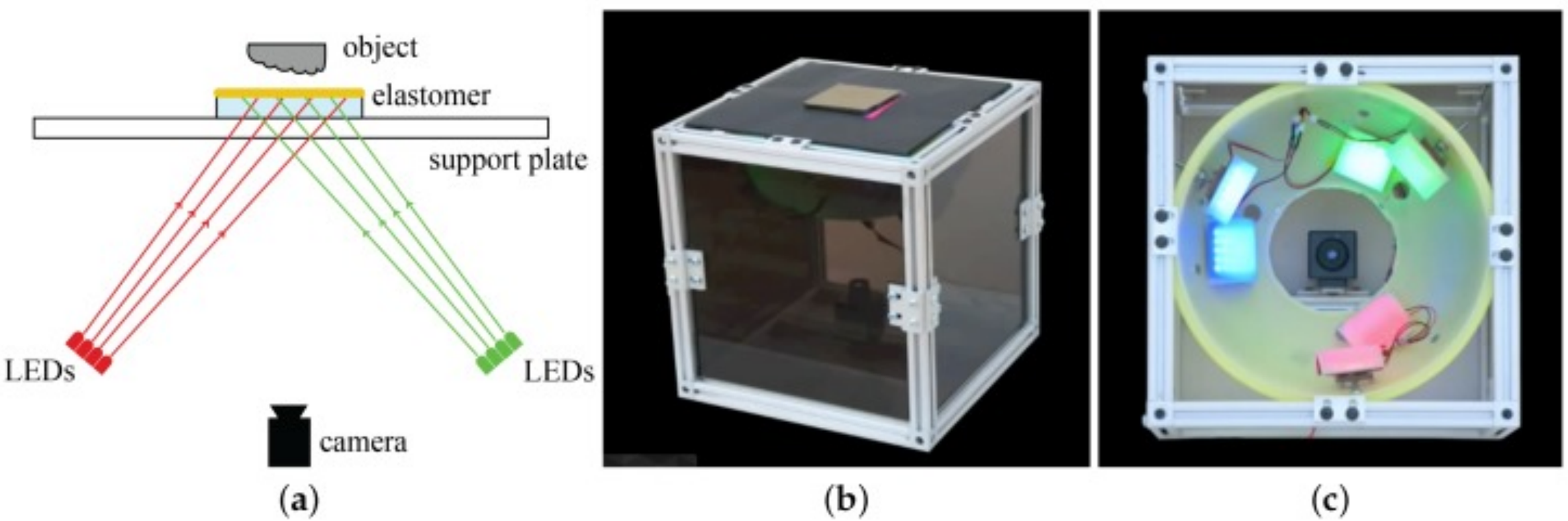
More Than a Feeling: Learning to Grasp and Regrasp using Vision and Touch

Roberto Calandra[1], Andrew Owens[1], Dinesh Jayaraman[1], Justin Lin[1], Wenzhen Yuan[2], Jitendra Malik[1], Edward H. Adelson[2], and Sergey Levine[1]



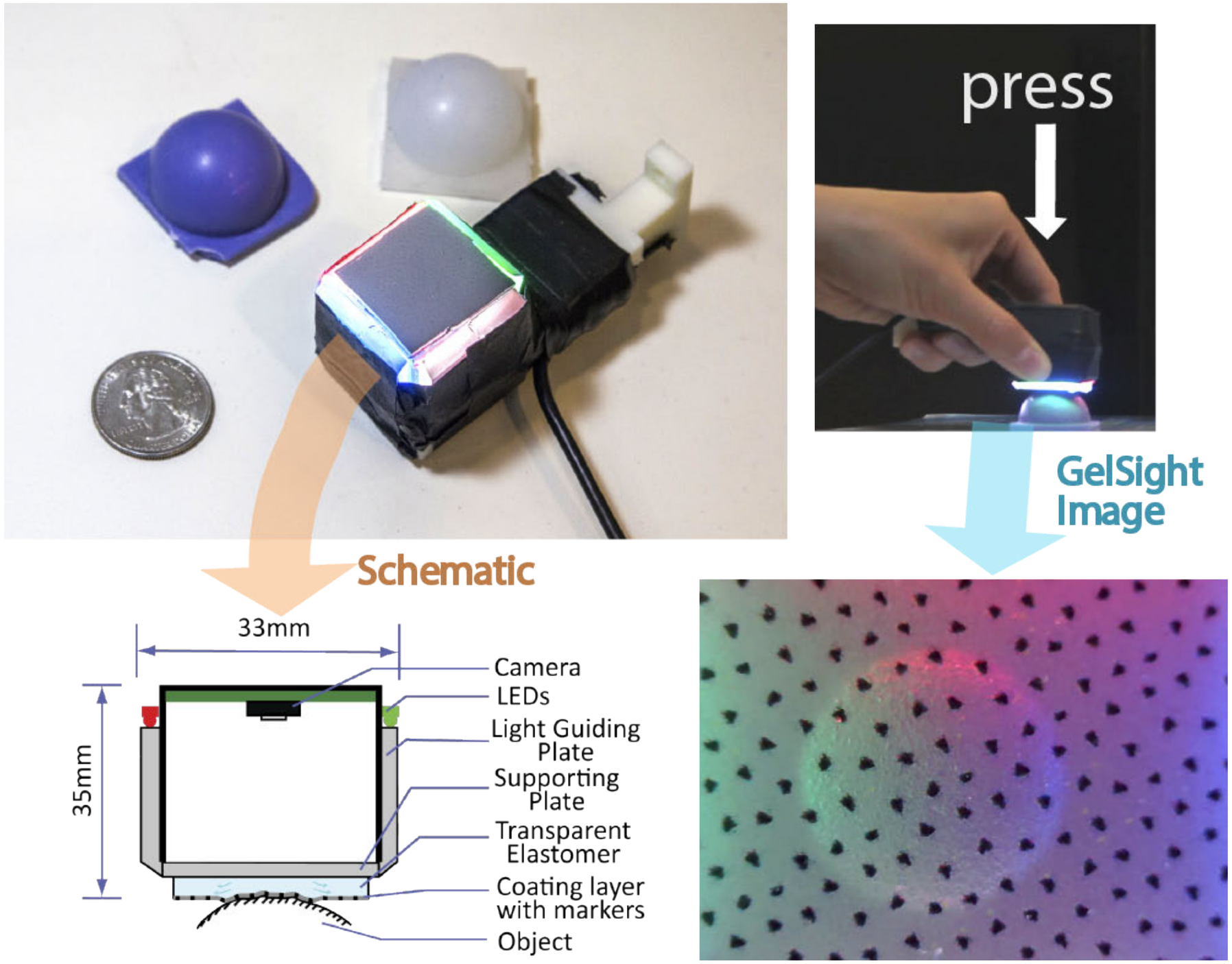
- Use haptic feedback to determine how to re-grasp for better grasping outcomes

Gelsight as a haptic sensor



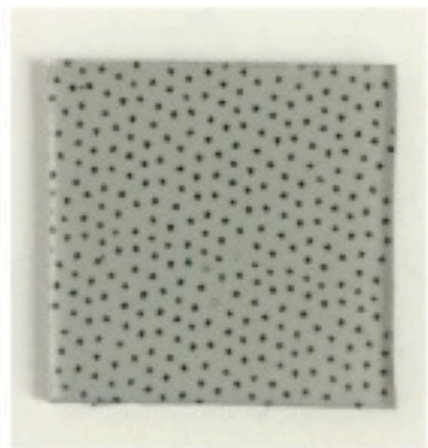
Ted Adelson's lab at MIT

Gelsight as a haptic sensor

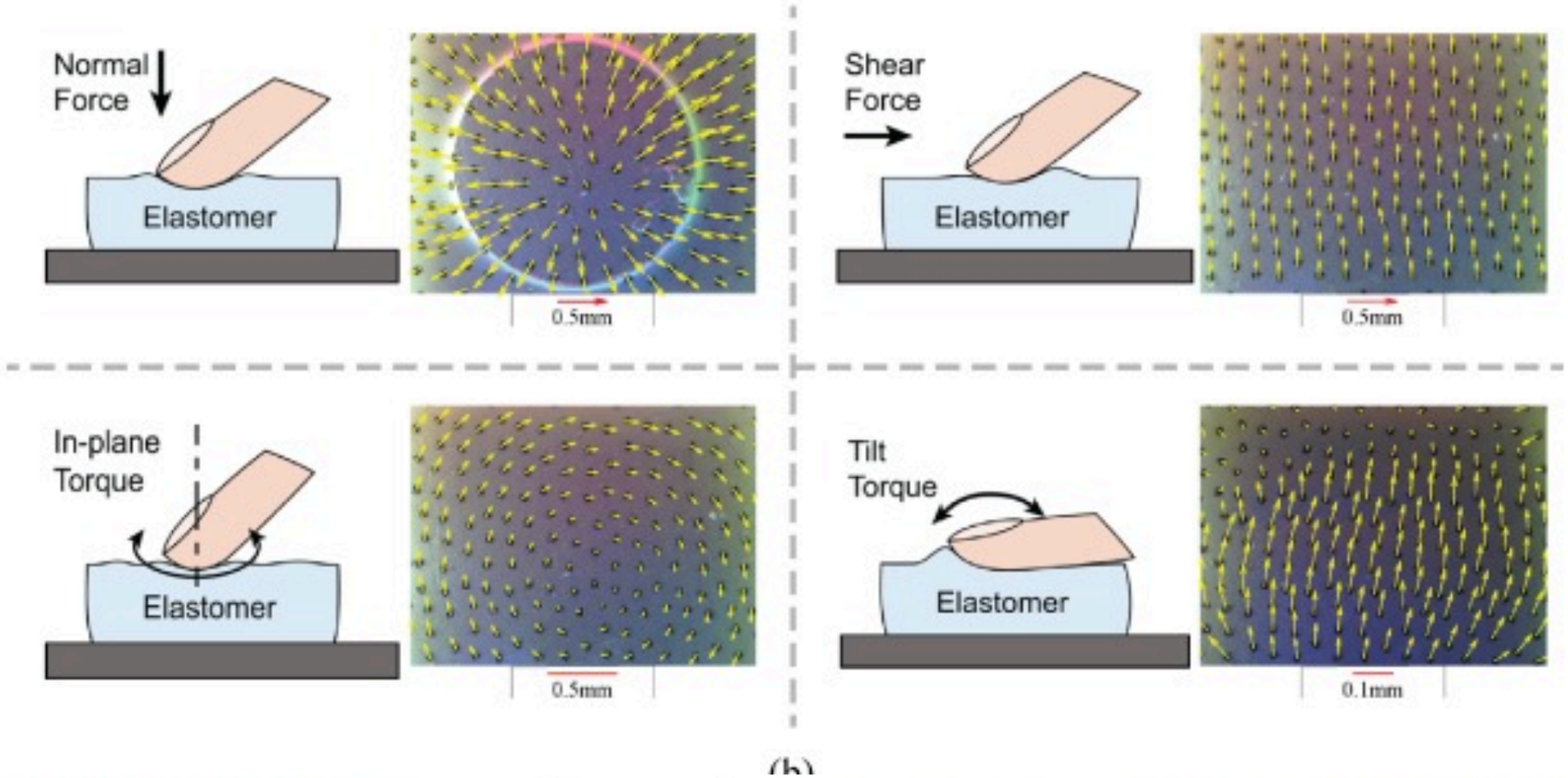


Yuan et al. Estimating Object Hardness with a GelSight Touch Sensor

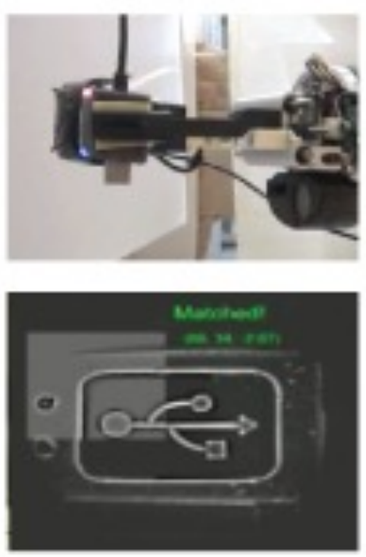
Gelsight as a haptic sensor



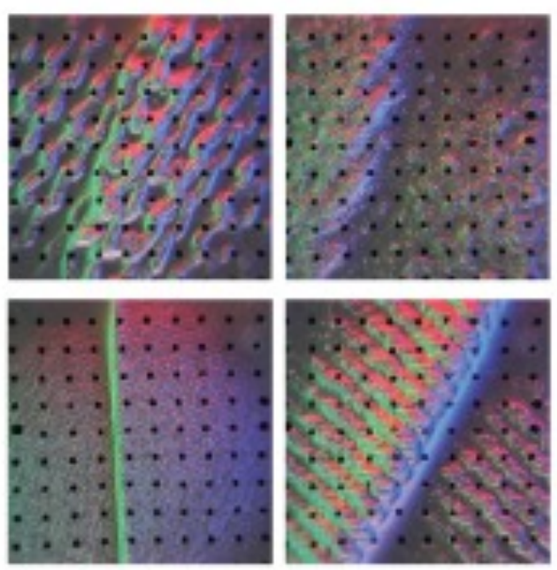
(a)



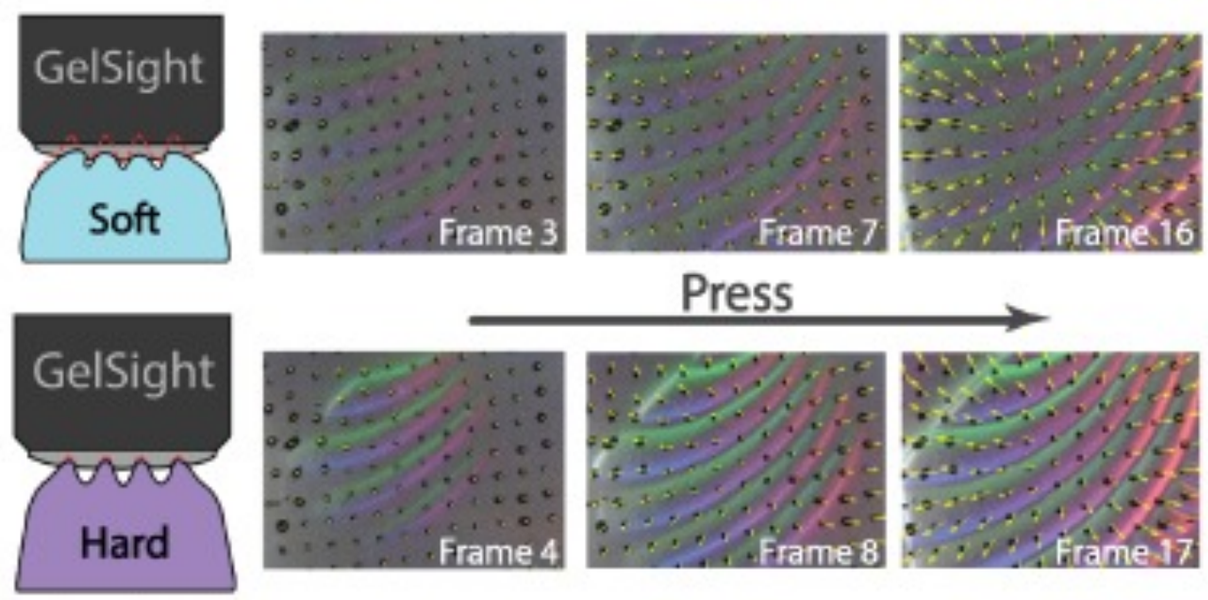
(b)



(a)



(b)

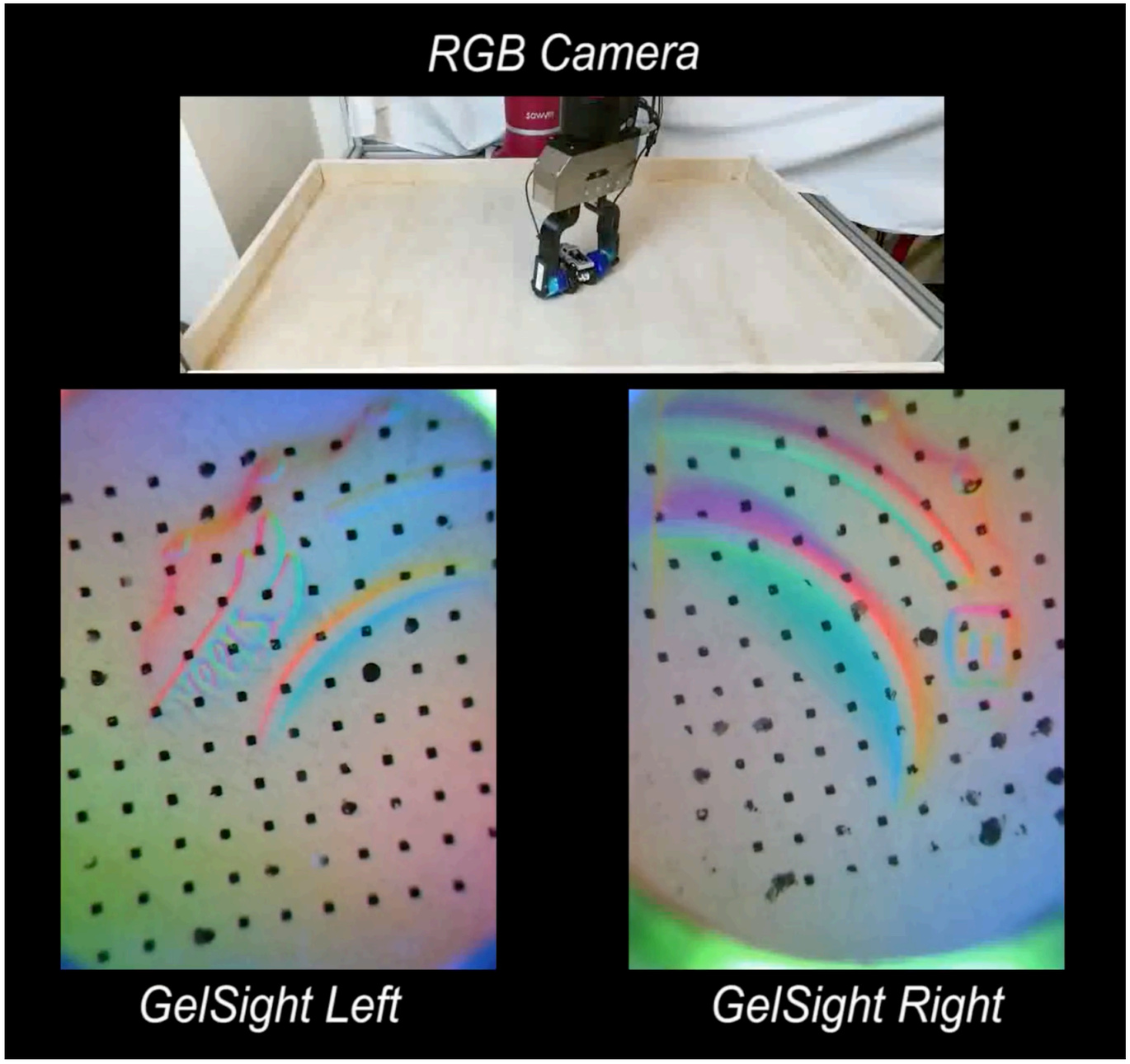


(c)

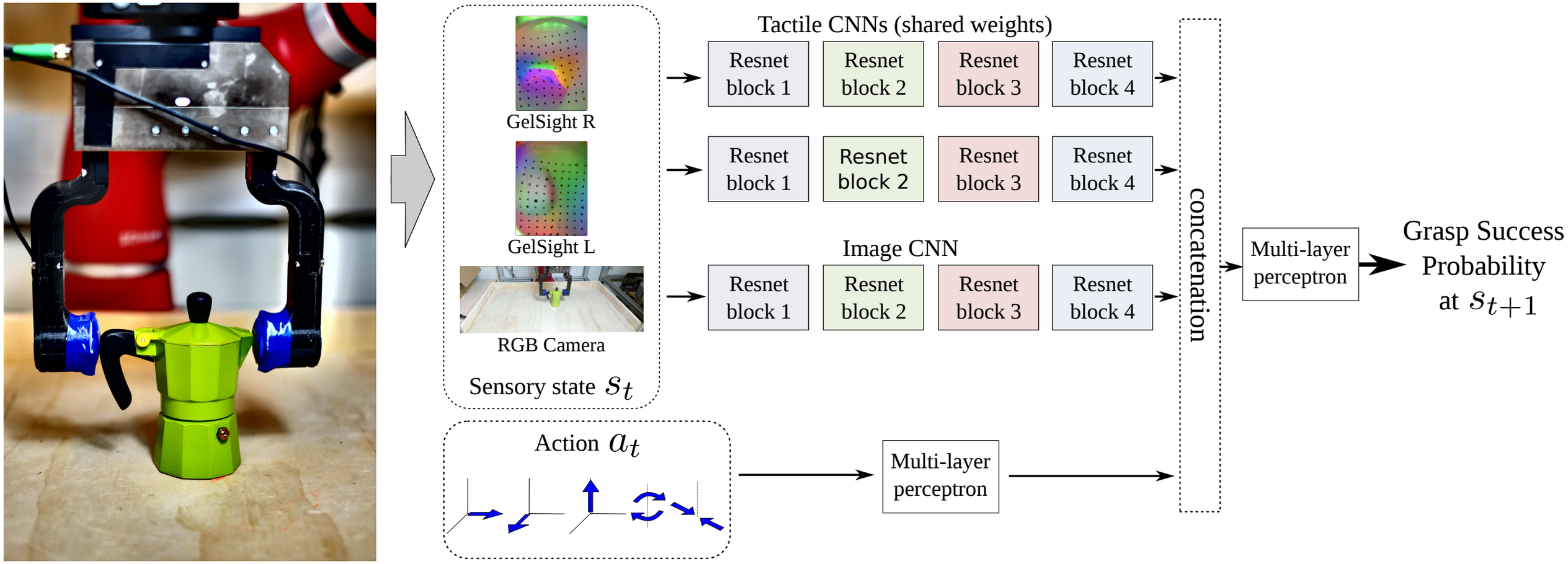
Ted Adelson's lab at MIT

Robotic Setup

- 7-DoF Sawyer Arm + Weiss WSG-50 Parallel Gripper
- Two GelSight Sensors
- Microsoft Kinect2 Sensor



Overall Architecture



Data Collection

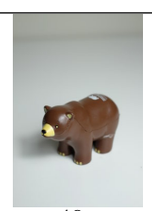
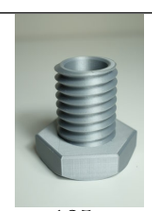
- Estimate Starting Position
- Set end-effector coordinates and height
- Set orientation and randomize force
- Lift and wait for 4s

Results

Table I: K-fold (K=3) cross-validation accuracy of the different models trained with 18,070 data points.

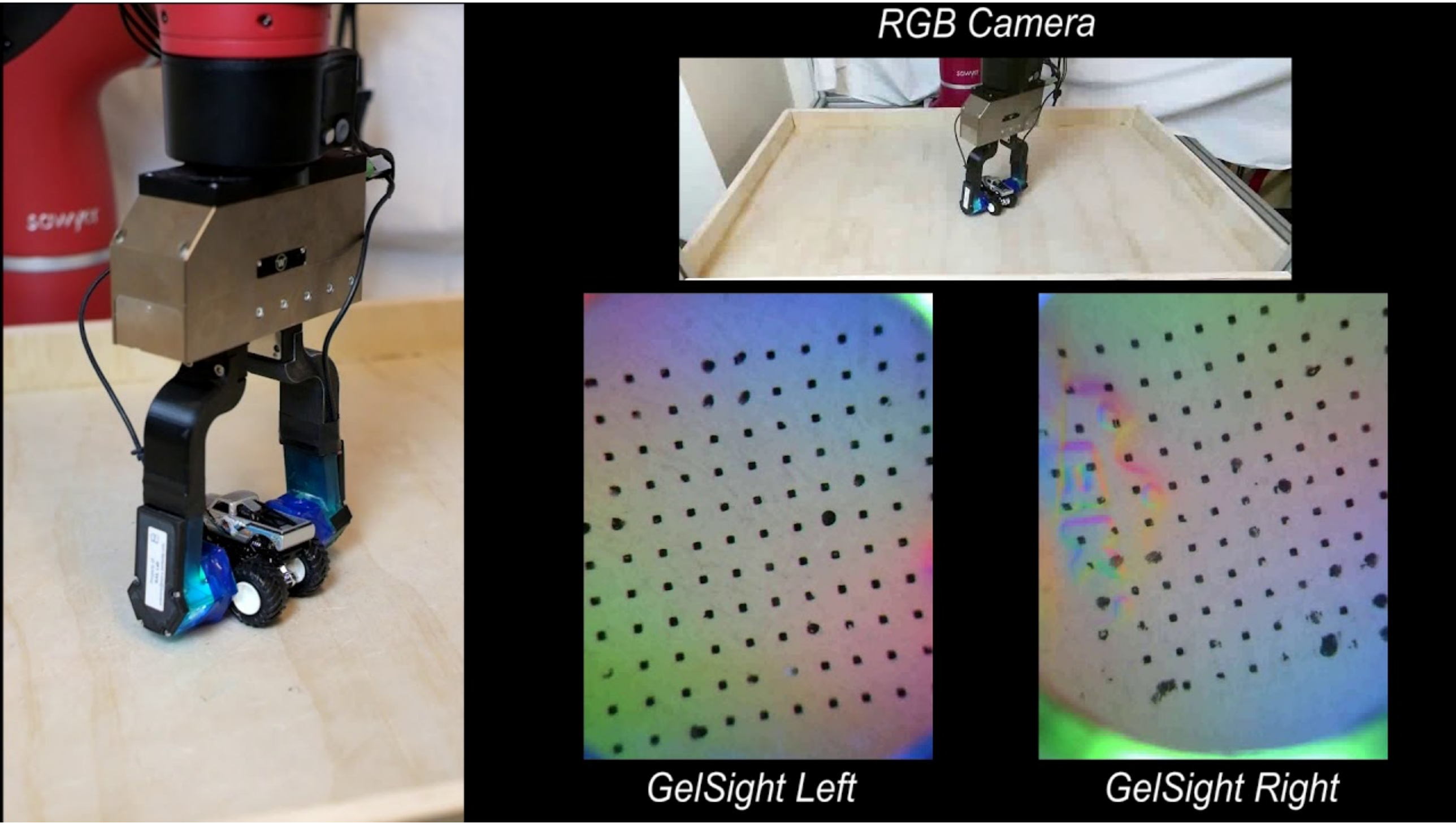
Model	Accuracy (mean ± std. err.)
Chance	$62.80\% \pm 0.85\%$
Vision (+ action)	$73.03\% \pm 0.24\%$
Tactile (+ action)	$79.34\% \pm 0.66\%$
Tactile + Vision (+ action)	$\mathbf{80.28\%} \pm \mathbf{0.68\%}$
Tactile + Vision (no action)	$76.43\% \pm 0.42\%$

Table II: Detailed grasping results using different policies for the "Easy" and "Hard" test objects.

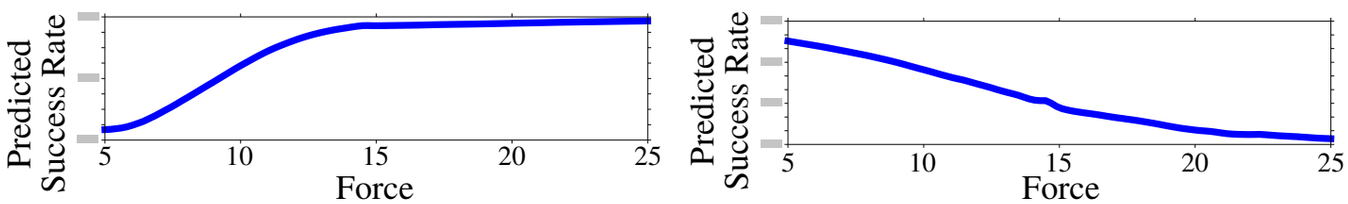
"Easy" set	Objects / Methods												Average grasp success
		215g	160g	40g	125g	125g	65g	135g	30g	380g	140g	10g	
		% grasp success (# success / # trials)											
	Vision only	76% (38/50)	70% (7/10)	60% (6/10)	50% (5/10)	50% (5/10)	90% (9/10)	40% (4/10)	60% (6/10)	**90% (9/10)**	10% (1/10)	**100% (10/10)**	63.2%
	Tactile + Vision	**95% (95/100)**	**100% (10/10)**	**100% (10/10)**	**100% (10/10)**	90% (9/10)	**100% (10/10)**	**90% (9/10)**	**100% (10/10)**	80% (8/10)	**90% (9/10)**	90% (9/10)	**94.0%**
	Cylinder fitting	90% (18/20)	90% (18/20)	80% (16/20)	55% (11/20)	**100% (20/20)**	**100% (20/20)**	**90% (18/20)**	75% (15/20)	35% (7/20)	20% (4/20)	**100% (20/20)**	75.9%

"Hard" set	Objects / Methods												Average grasp success
		230g	120g	195g	50g	70g	85g	38g	165g	65g	340g	110g	
		% grasp success (# success / # trials)											
	Vision only	60% (6/10)	80% (8/10)	30% (3/10)	30% (3/10)	80% (8/10)	40% (4/10)	60% (6/10)	50% (5/10)	50% (5/10)	50% (5/10)	20% (2/10)	50%
	Tactile + Vision	80 % (8/10)	**100% (10/10)**	**50% (5/10)**	80% (8/10)	**90% (9/10)**	**70% (7/10)**	**100% (10/10)**	40% (4/10)	**60% (6/10)**	**80% (8/10)**	60% (6/10)	**73.6%**
	Cylinder fitting	**95% (19/20)**	**100% (20/20)**	35% (7/20)	**100% (20/20)**	**90% (18/20)**	15% (3/20)	90% (18/20)	**85% (17/20)**	15% (3/20)	15% (3/20)	**95% (19/20)**	66.8%

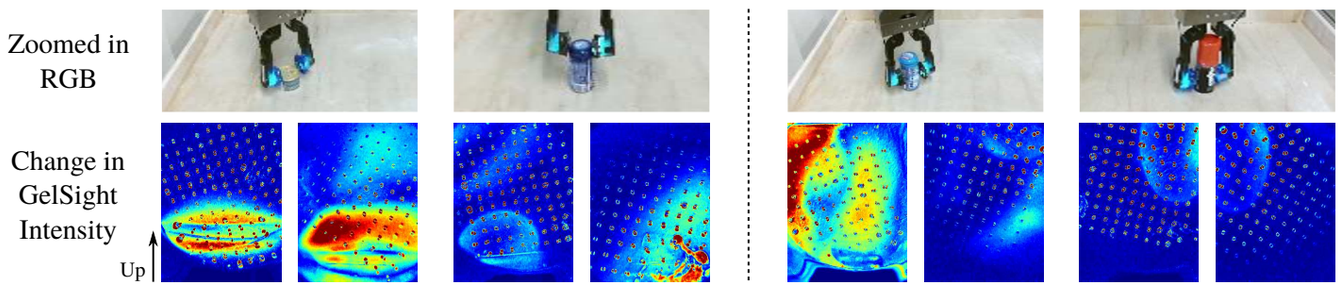
Overall Architecture



Results (Analysis)

- 

(a) Stable grasp (b) Unstable grasp

- 

(a) Improvement from downward motion (b) No improvement

Thank you