Do We Need 3D Representations for Robot Manipulation?

Yu Xiang

Assistant Professor

Intelligent Robotics and Vision Lab

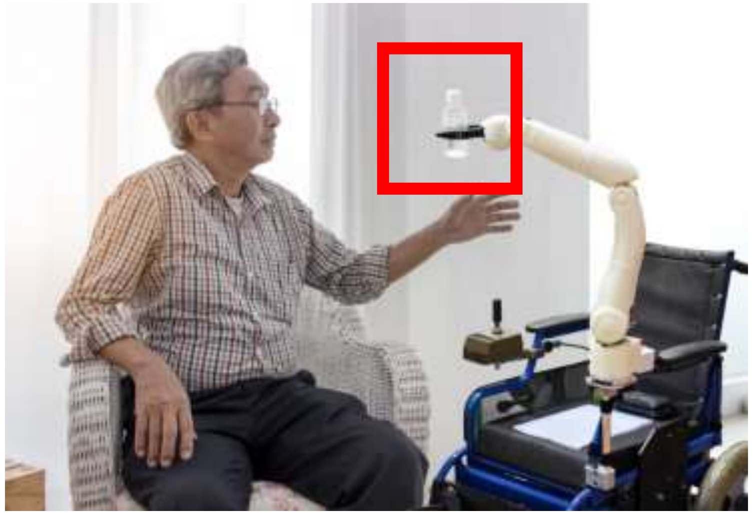
The University of Texas at Dallas

1st Workshop on 3D Visual Representations for Robot Manipulation.

5/17/2024

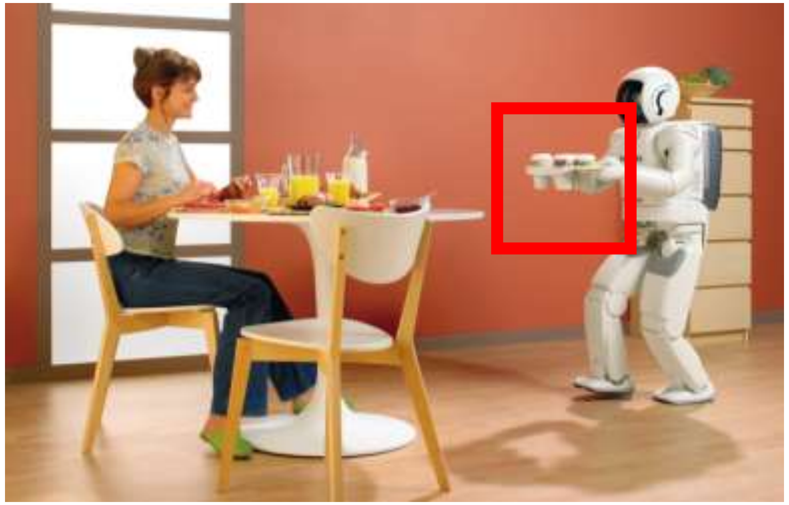
Future Intelligent Robots in Human Environments

Manipulation

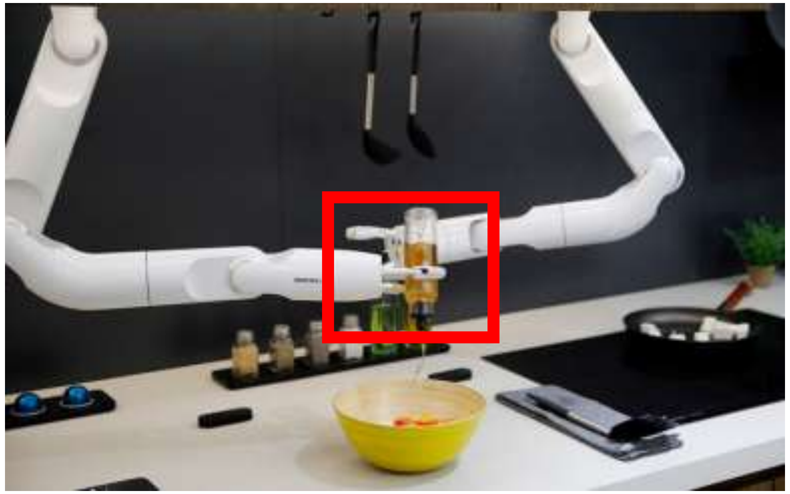


Senior Care

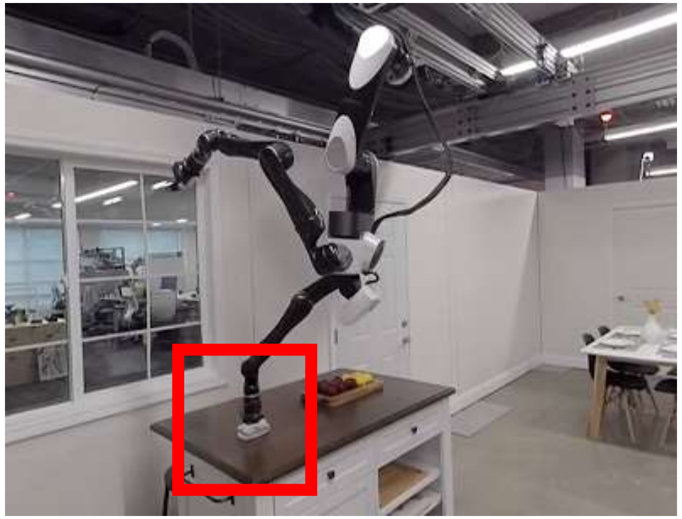
Assisting



Serving



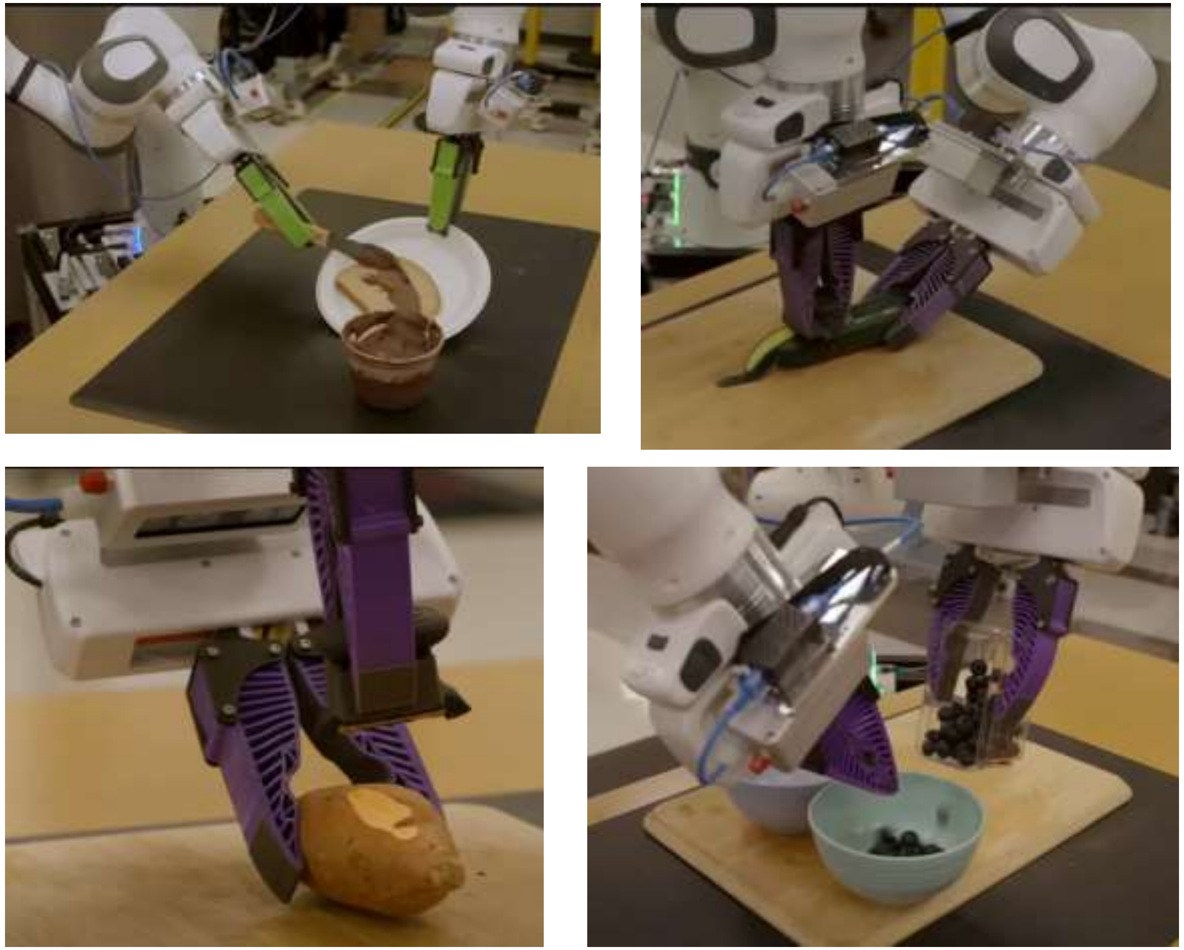
Cooking



Cleaning

Dish washing

Some Recent Breakthroughs



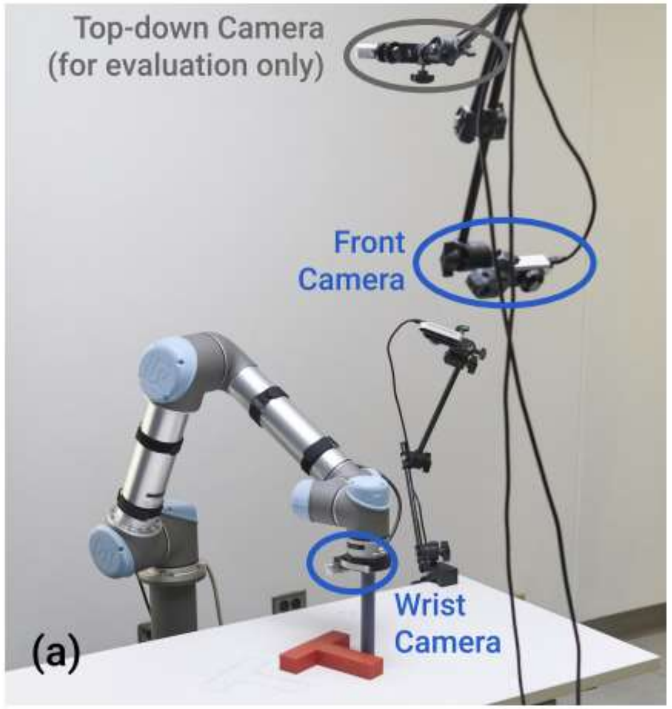
Diffusion Policy, Columbia & MIT & TRI
Cheng Chi, Shuran Song, et al.

https://diffusion-policy.cs.columbia.edu/

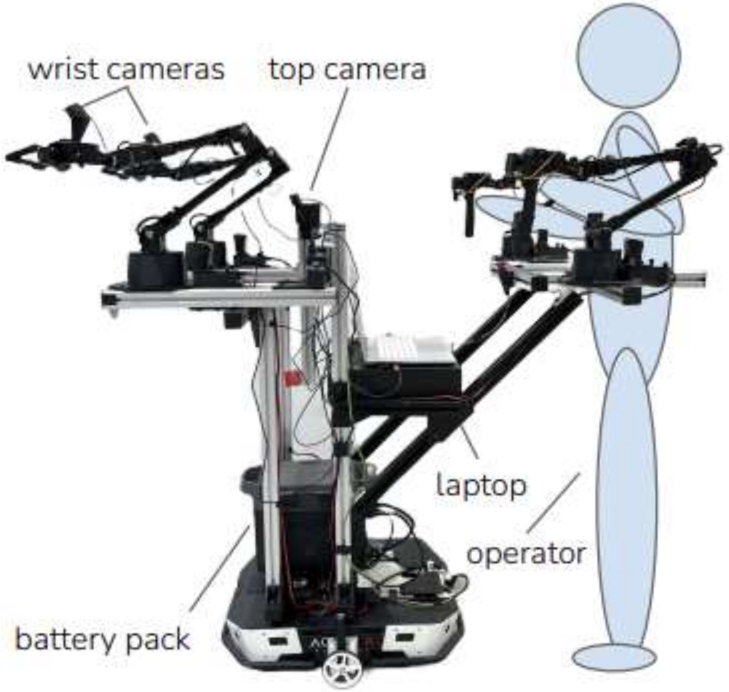
Mobile ALOHA, Stanford
Zipeng Fu, Tony Zhao, Chelsea Finn

https://mobile-aloha.github.io/

What is the Representation in these Models?



Diffusion Policy, Columbia & MIT & TRI



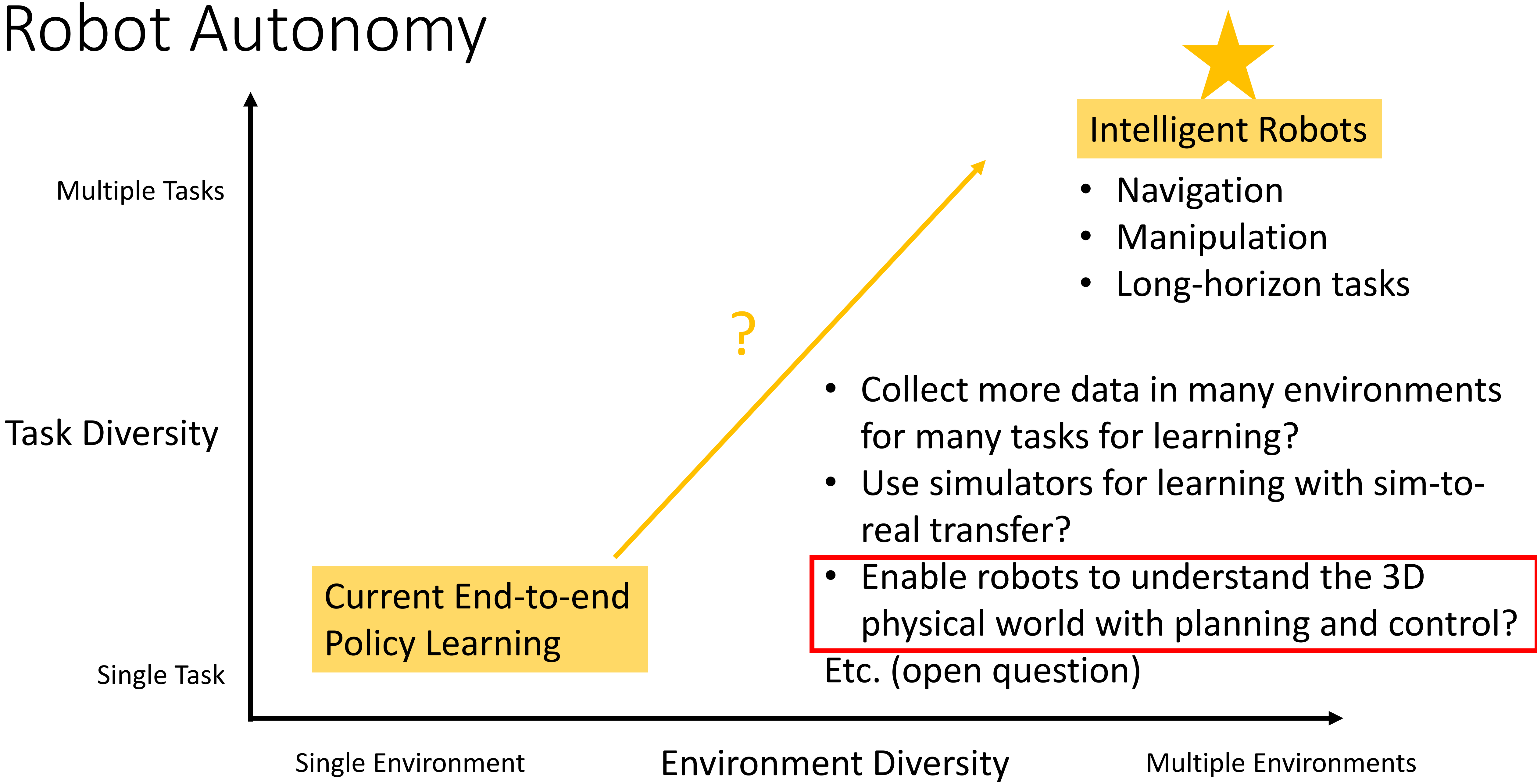
Mobile ALOHA, Stanford

Do we need 3D representations?

Camera images → Visual Encoder → Policy Network → Actions

End-to-end Learning from Images

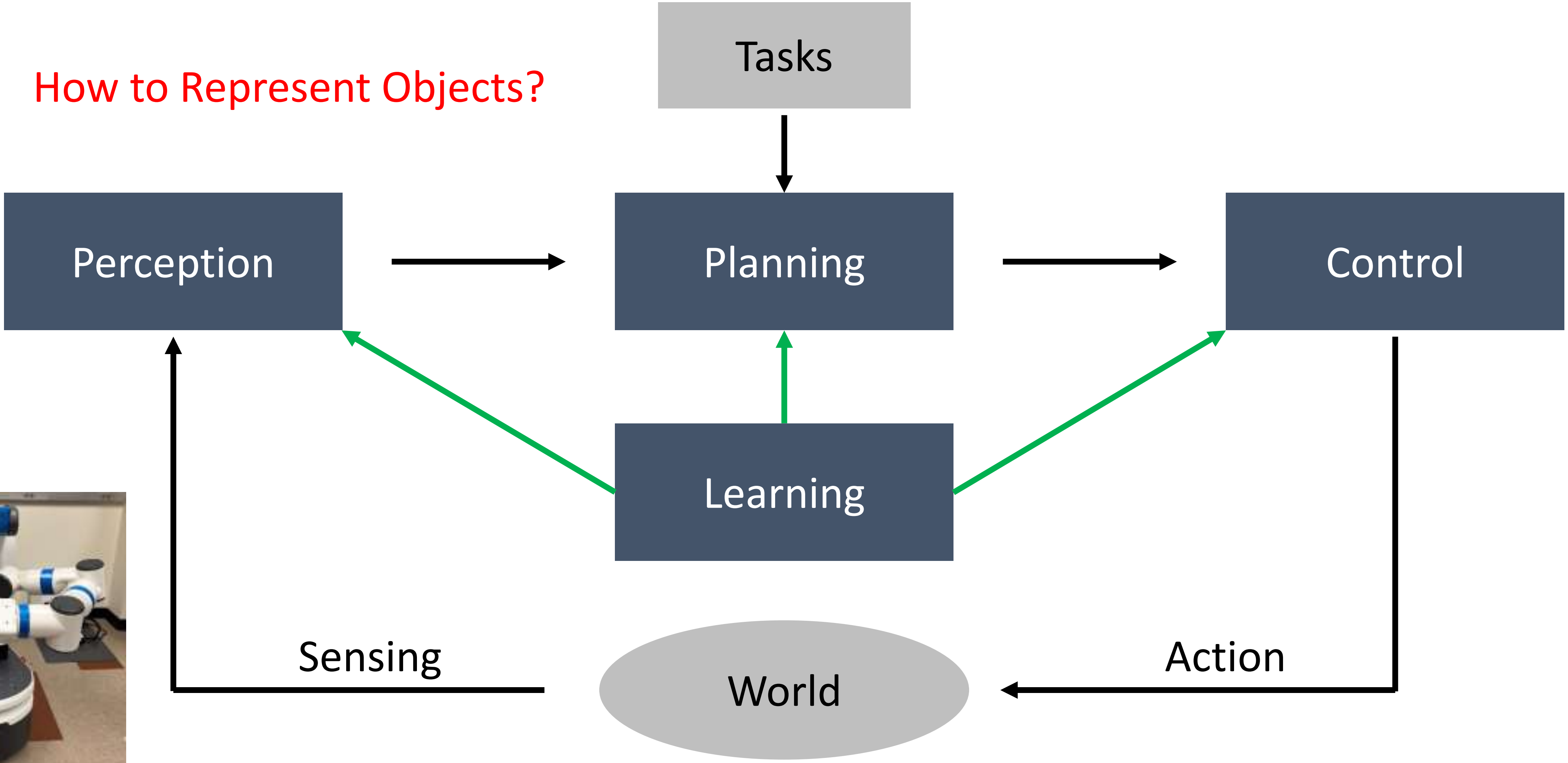
Robot Autonomy



The Perception, Planning and Control Loop

Good Old Fashioned Engineering (GOFE)

How to Represent Objects?



How to Represent Objects?

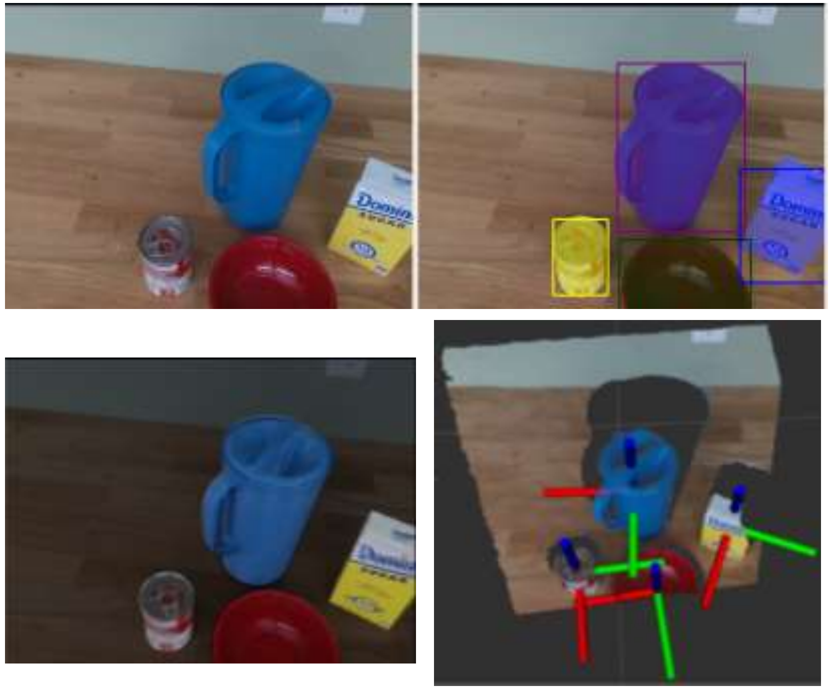
- 3D CAD models (Model-based)

- Point clouds (Model-free)

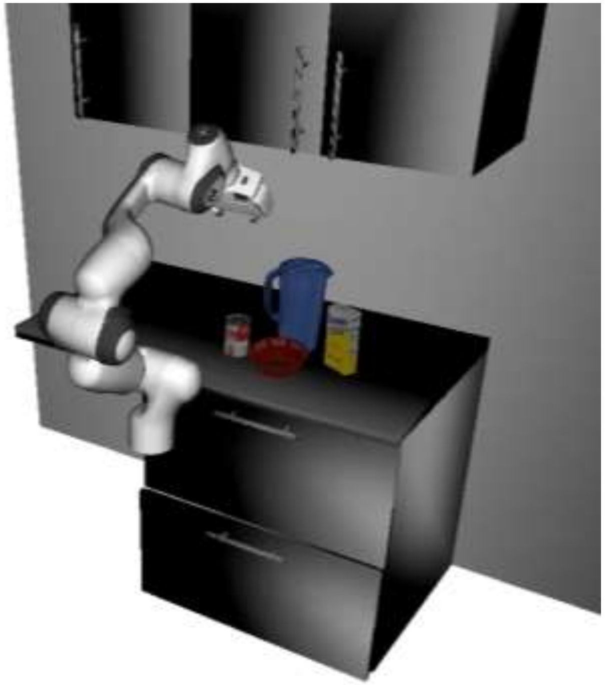
Using 3D Object Models

Perception → Planning → Control

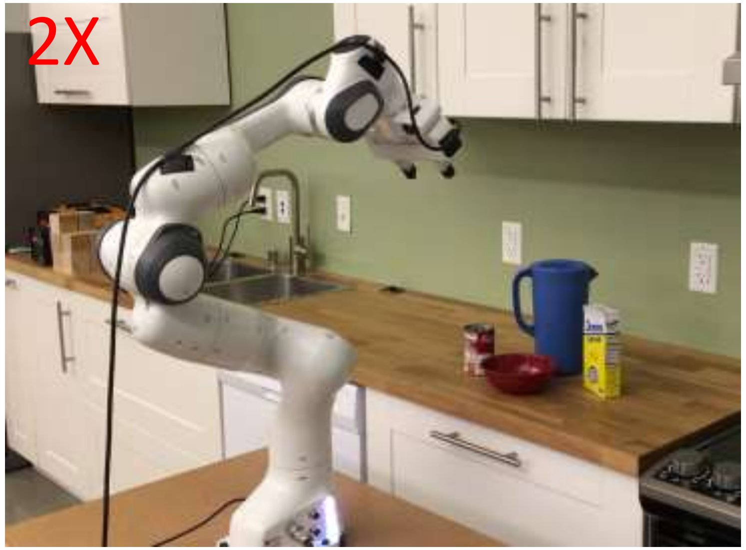
6D object pose estimation



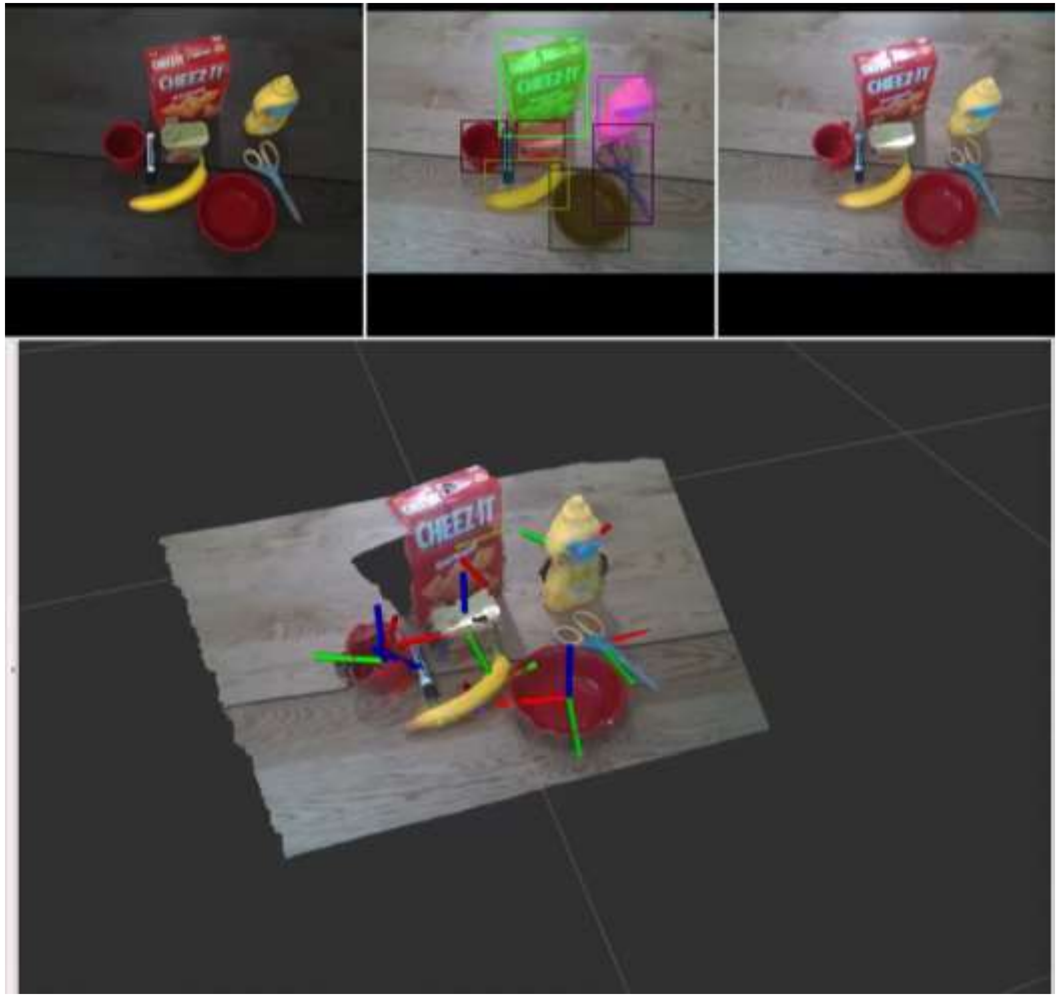
Grasp planning and motion planning



Manipulation trajectory following



6D Object Pose Estimation



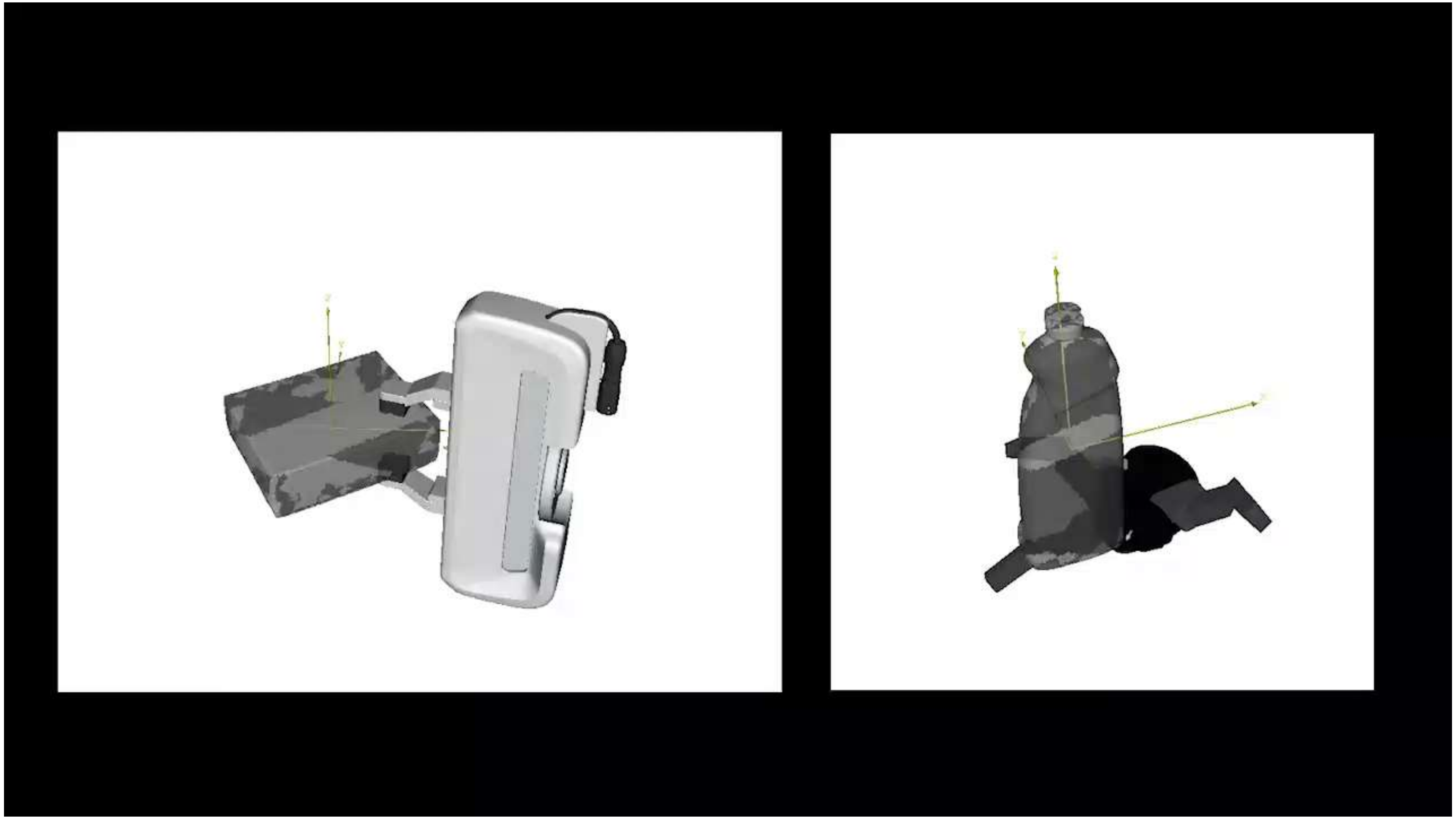
FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects

Bowen Wen, Wei Yang, Jan Kautz, Stan Birchfield

NVIDIA

- PoseCNN, RSS’17
- DeepIM, ECCV’18
- DOPE, CoRL’18
- PoseRBPF, RSS’19, T-TO’21
- Self-supervised 6D Pose, ICRA’20
- LatentFusion, CVPR’20

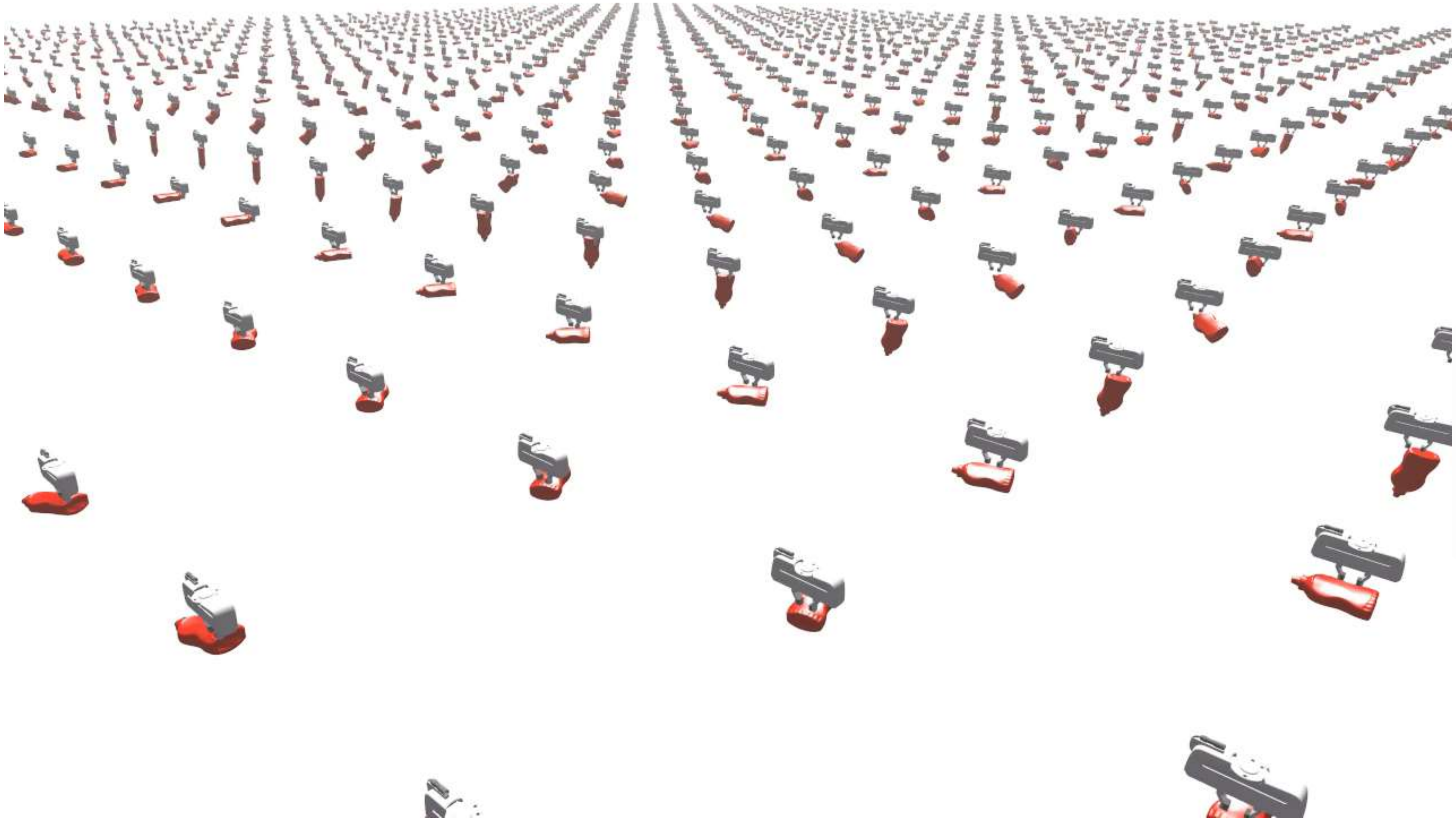
Grasp Planning: GraspIt!



GraspIt! https://graspit-simulator.github.io/

Andrew Miller and Peter K. Allen. "Graspit!: A Versatile Simulator for Robotic Grasping". IEEE Robotics and Automation Magazine, V. 11, No.4, Dec. 2004, pp. 110-122.

Grasp Planning: A Physics-based Approach



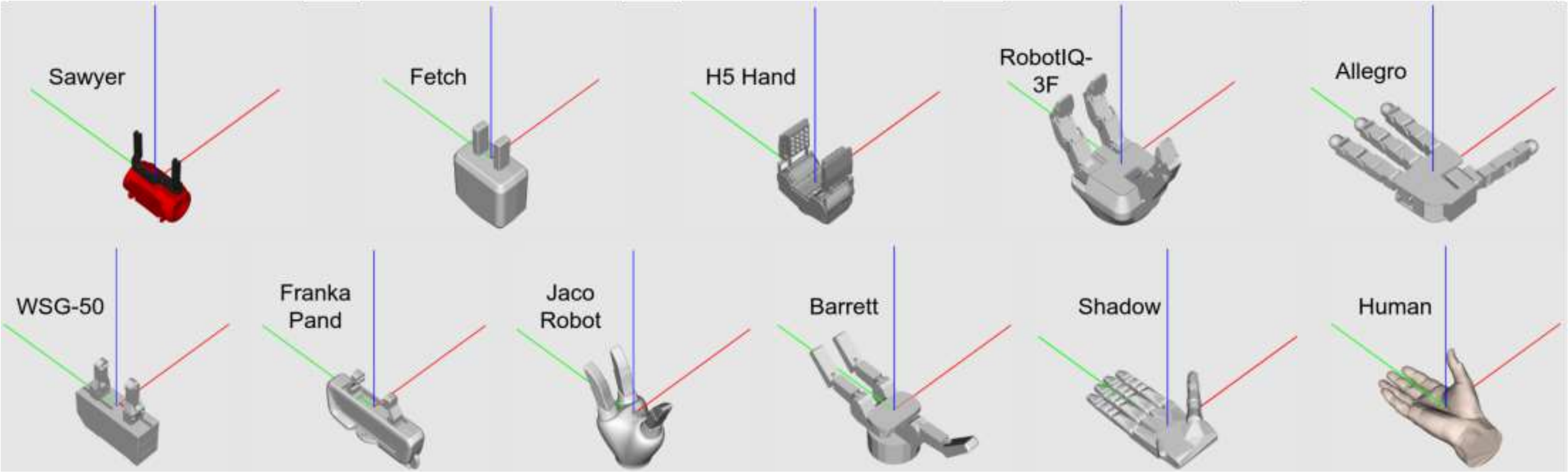
Eppner-Mousavian-Fox, ISRR’19

MultiGripperGrasp

- A large-scale dataset for robotic grasping
  - 11 grippers, 345 objects, 30M grasps

MultiGripperGrasp: A Dataset for Robotic Grasping from Parallel Jaw Grippers to Dexterous Hands
Luis Felipe Casas Murrilo*, Ninad Khargonkar*, Balakrishnan Prabhakaran, Yu Xiang (*equal contribution)
In arXiv, 2024.

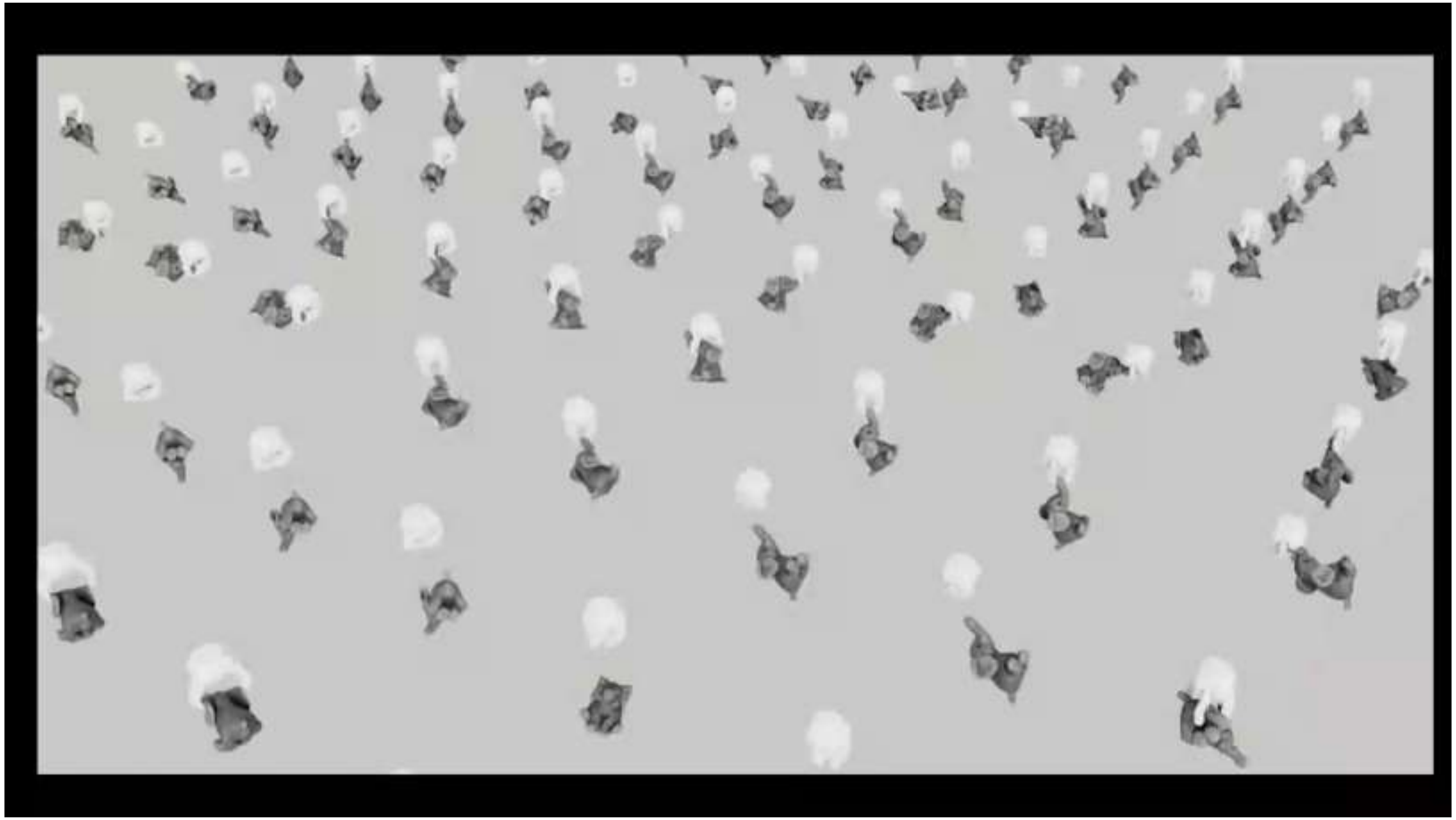
MultiGripperGrasp



- 11 grippers (aligned with palm directions)
  - 2-finger grippers: Fetch, Franka Panda, WSG50, Sawyer, H5 Hand
  - 3-finger grippers: Barrett, Robotiq-3F, Jaco Robot
  - 4-finger grippers: Allegro
  - 5 finger grippers: Shadow, Human Hand

MultiGripperGrasp

- Generate initial grasps using GraspIt!
- Ranking grasps in Isaac Sim

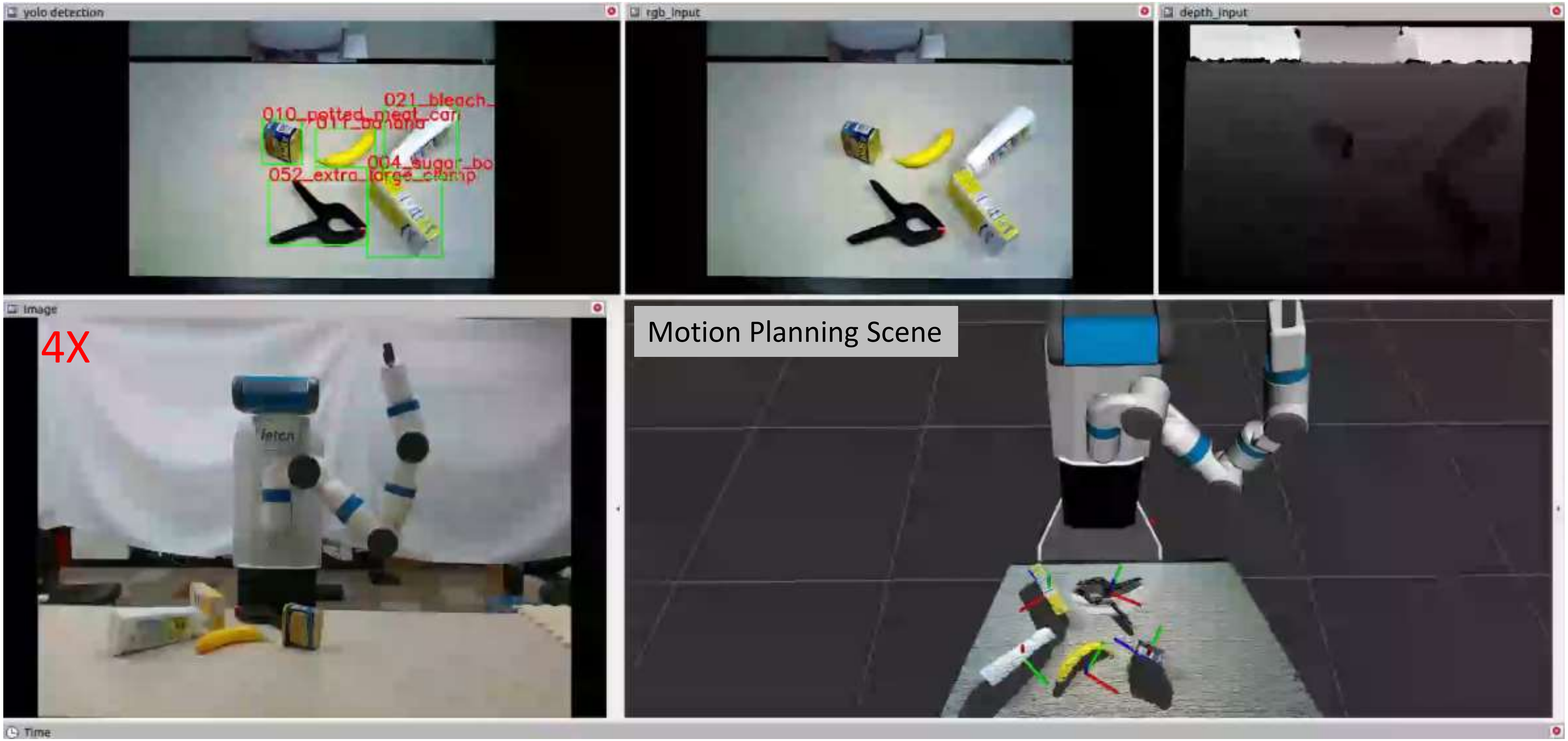


MultiGripperGrasp

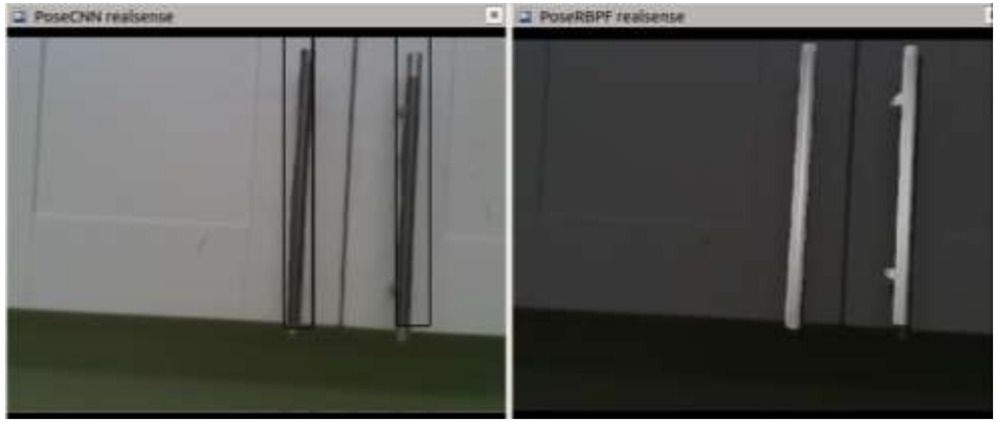
- Grasp Transfer in Isaac Sim

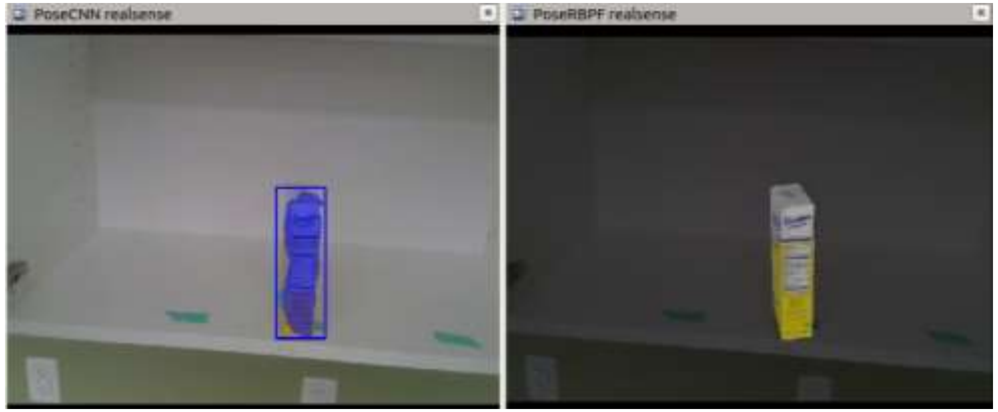
https://irvlutd.github.io/MultiGripperGrasp/

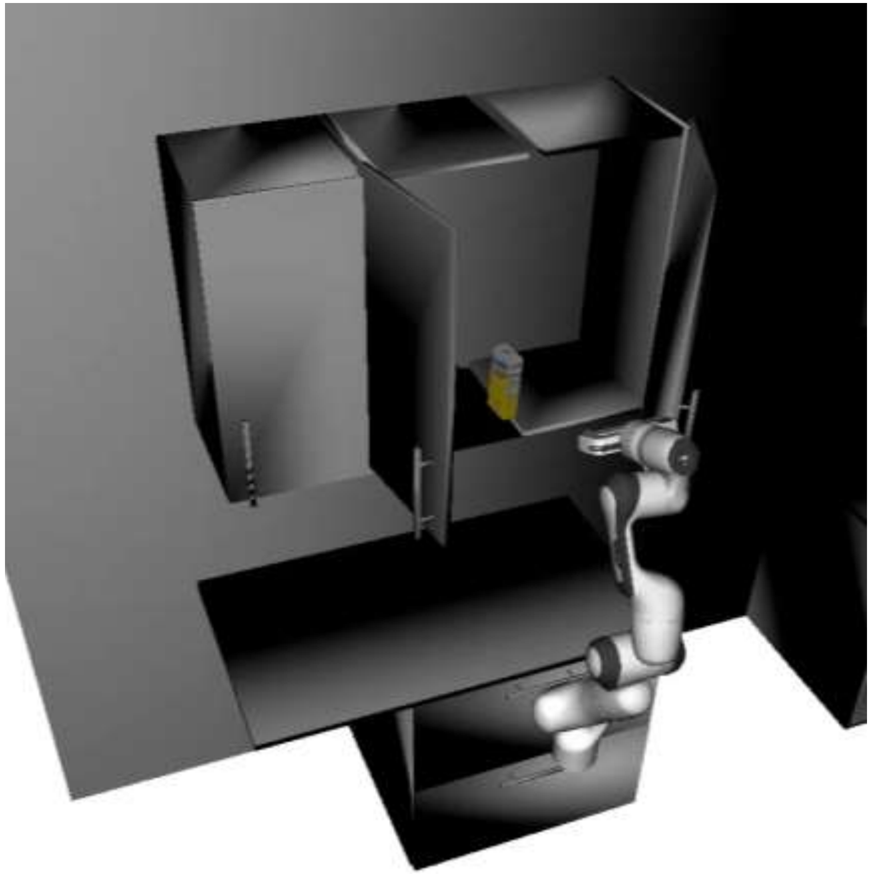
Motion Planning



The Open Motion Planning Library in MoveIt https://ompl.kavrakilab.org/index.html


PoseCNN realsense
PoseRBPF realsense


PoseCNN realsense
PoseRBPF realsense

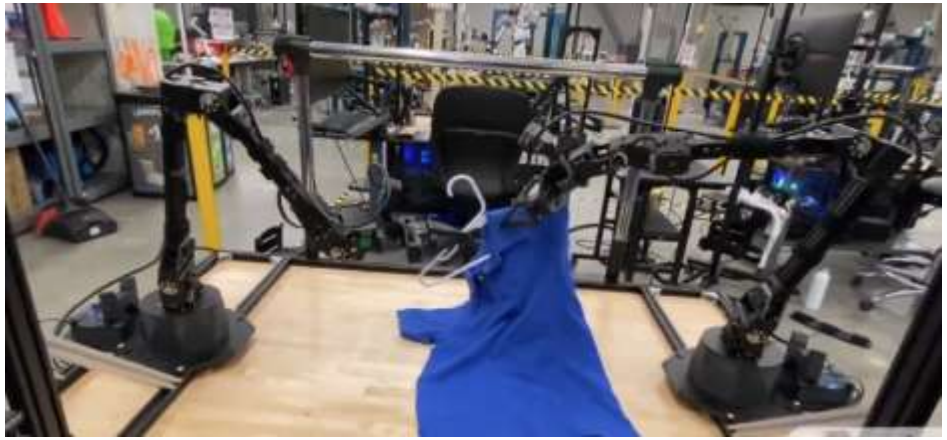



4X

Using 3D Object Models

- Pros
  - Encodes appearance, 3D shape, affordance, physical properties for perception, planning and simulation

- Cons
  - We cannot build 3D models for all objects

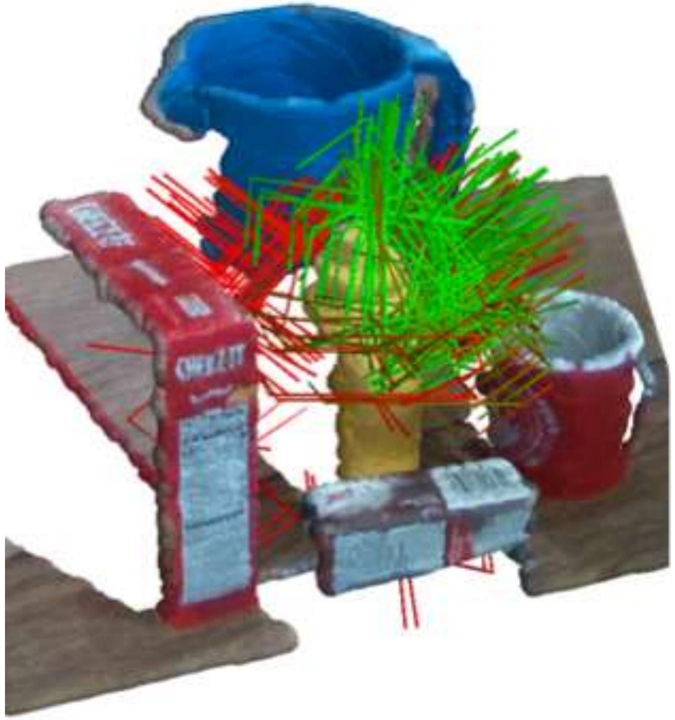


ALOHA Unleashed
Google DeepMind

Using 3D Point Clouds

Perception → Planning → Control

object instance segmentation



Grasp planning from point clouds

Control to reach grasp

Figure Credit: Murali-Mousavian-Eppner-Paxton-Fox, ICRA’20

Segmenting Unseen Objects

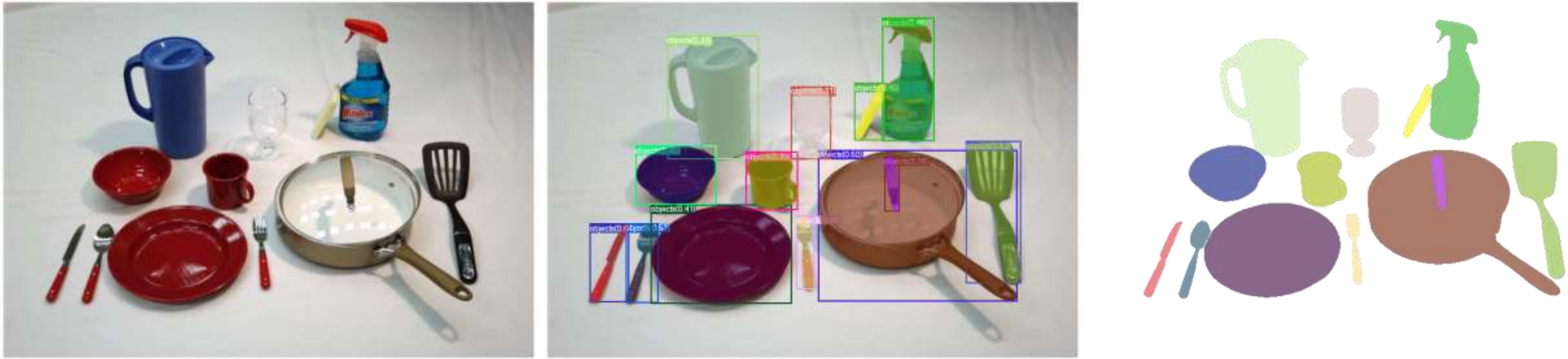
Input Image

Output Label

Xie-Xiang-Mousavian-Fox, CoRL’19, T-RO’21, CoRL’21
Xiang-Xie-Mousavian-Fox, CoRL’20
Lu-Khargonkar-Xu-Averill-Palanisamy-Hang-Guo-Ruozzi-Xiang, RSS’23
Lu-Chen-Ruozzi-Xiang, ICRA’24
Qian-Lu-Ren-Wang-Khargonkar-Xiang-Hang, ICRA’24

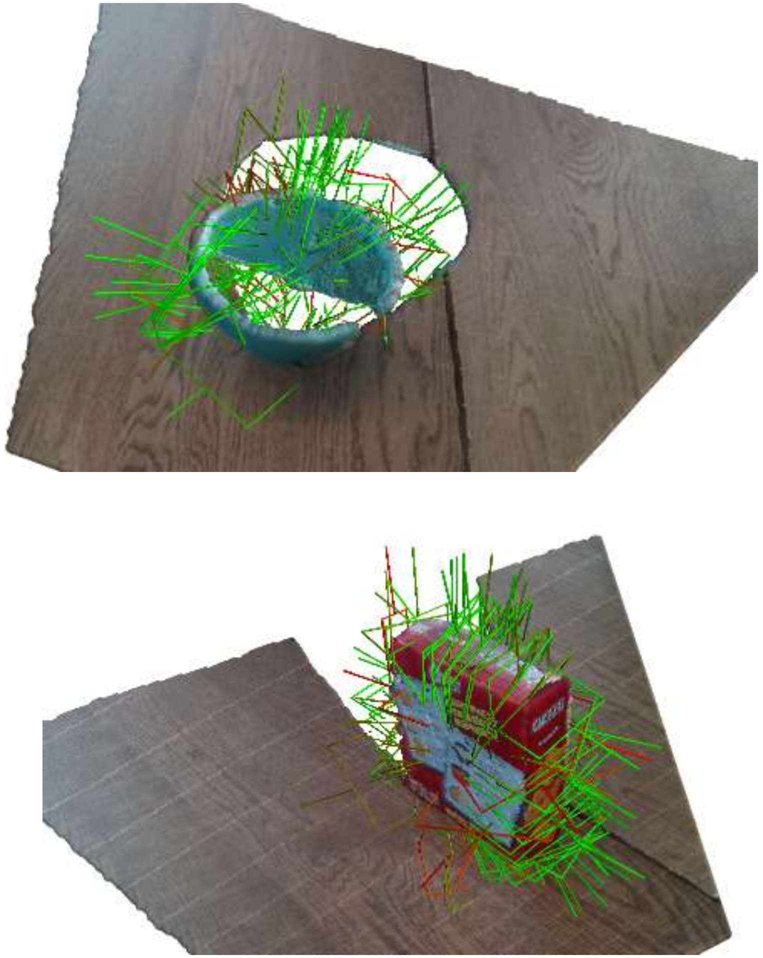
Leveraging Large Models from the Vision Community

- Gounding Dino (object detection)
- SAM (object segmentation)



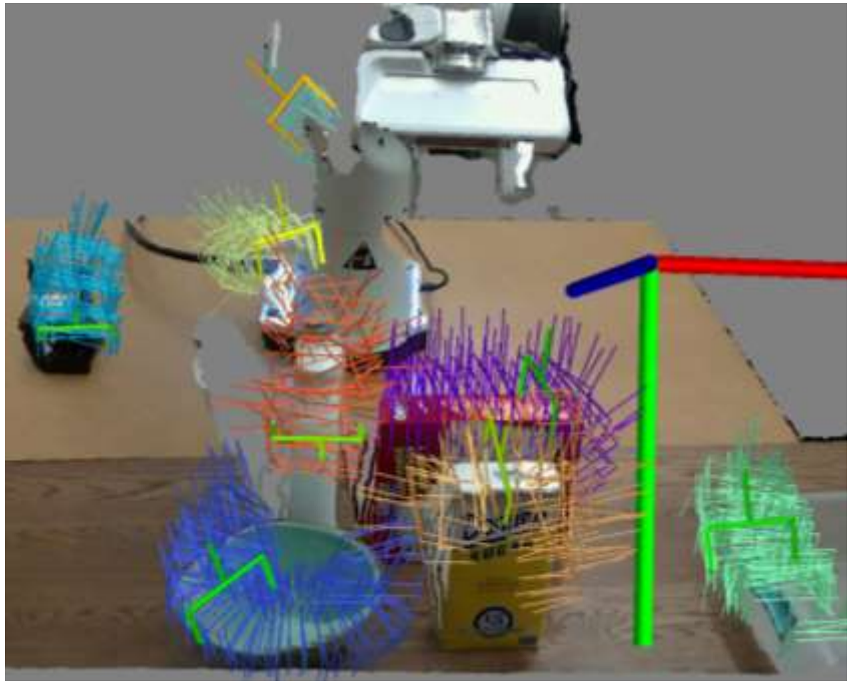
- Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. Liu et al., 2023
- Segment Anything. Kirillov et al., 2023

Grasp Planning with Point Clouds



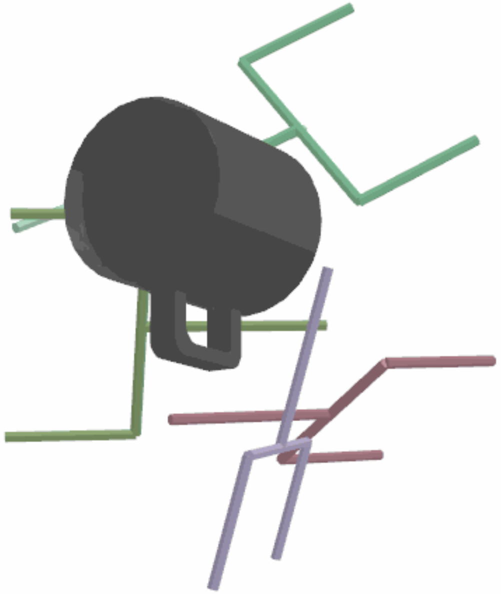
6D GraspNet

6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. Mousavian et al., ICCV'19



Contact-GraspNet

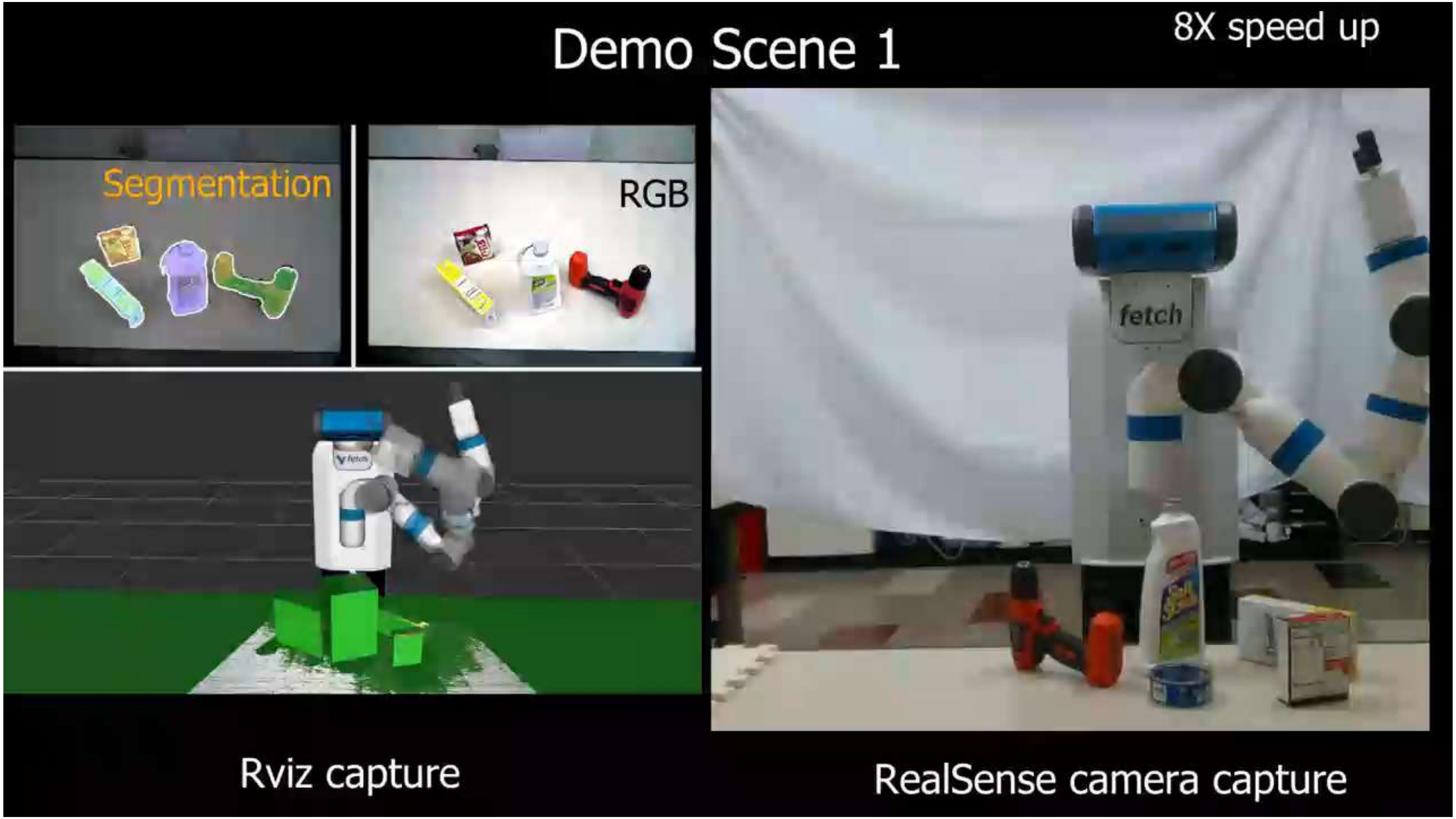
Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes. Sundermeyer, et al., ICRA'21



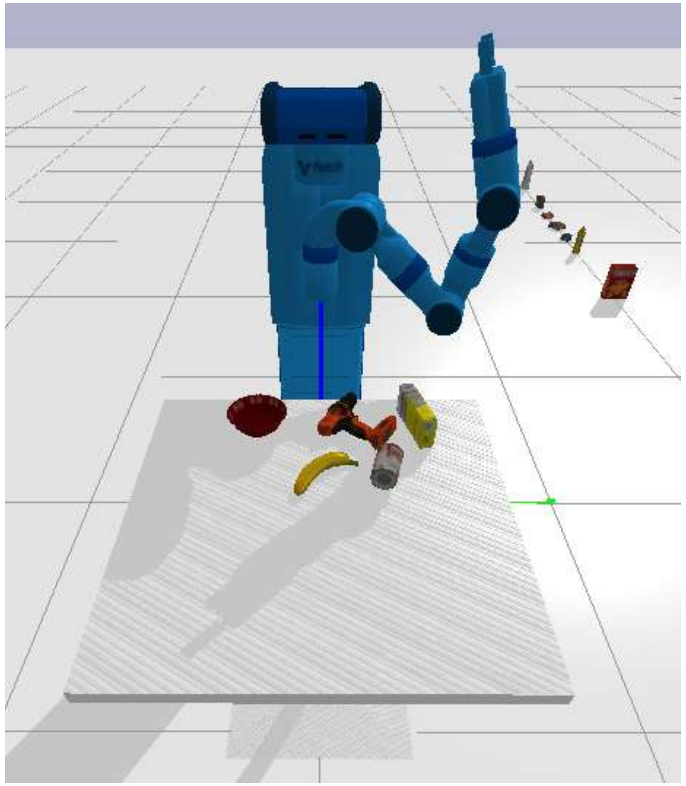
SE(3)-DiffusionFields

SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. Urain et al., 2023

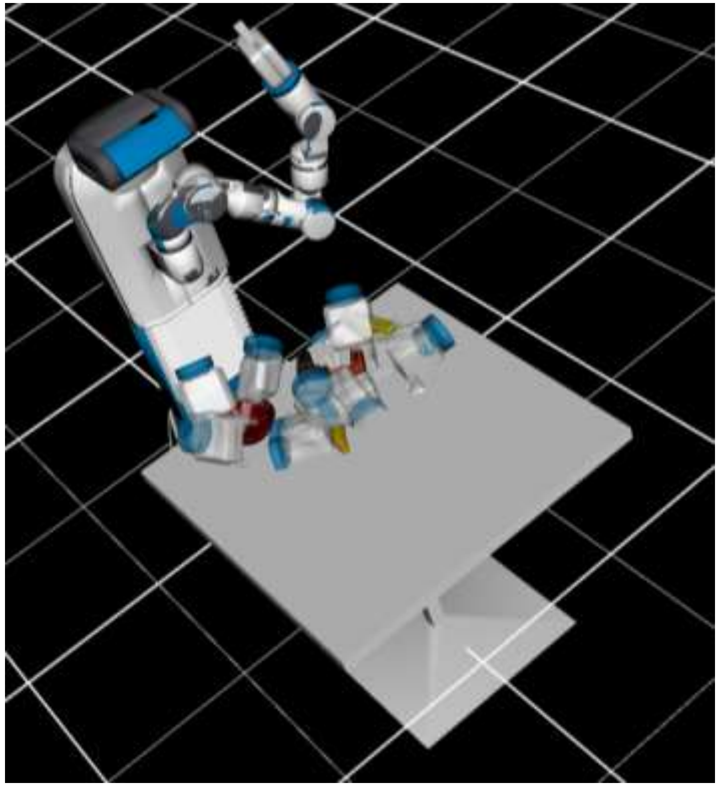
Model-free Grasping Example



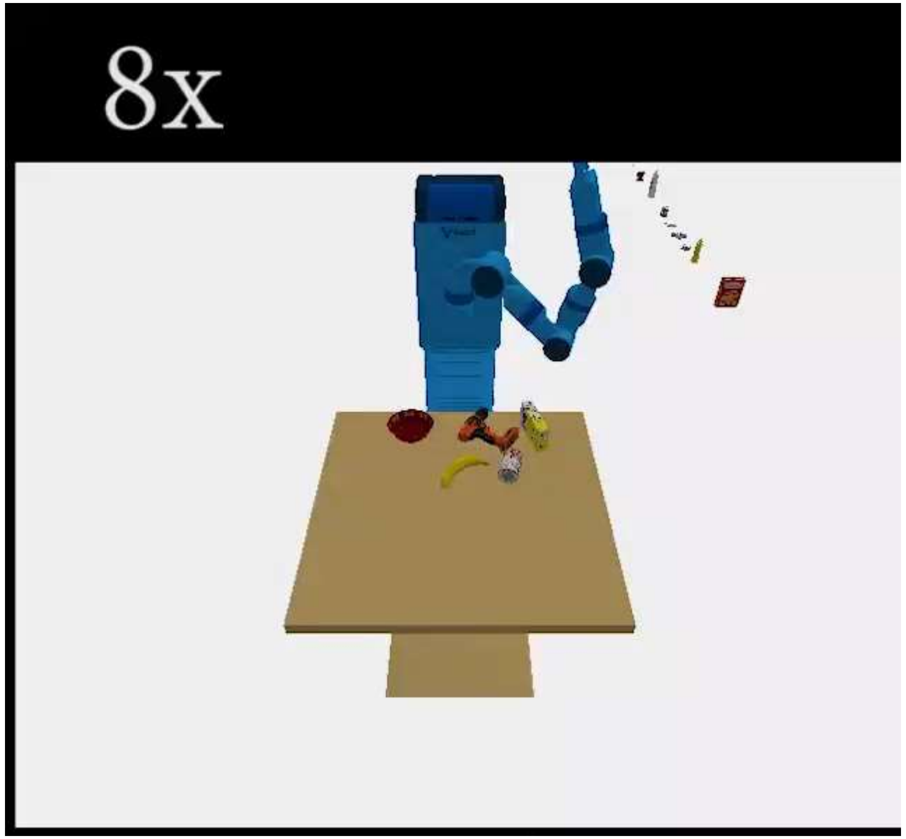
Grasping Trajectory Optimization with Point Clouds



(a) Task Space



(b) Grasp Planning

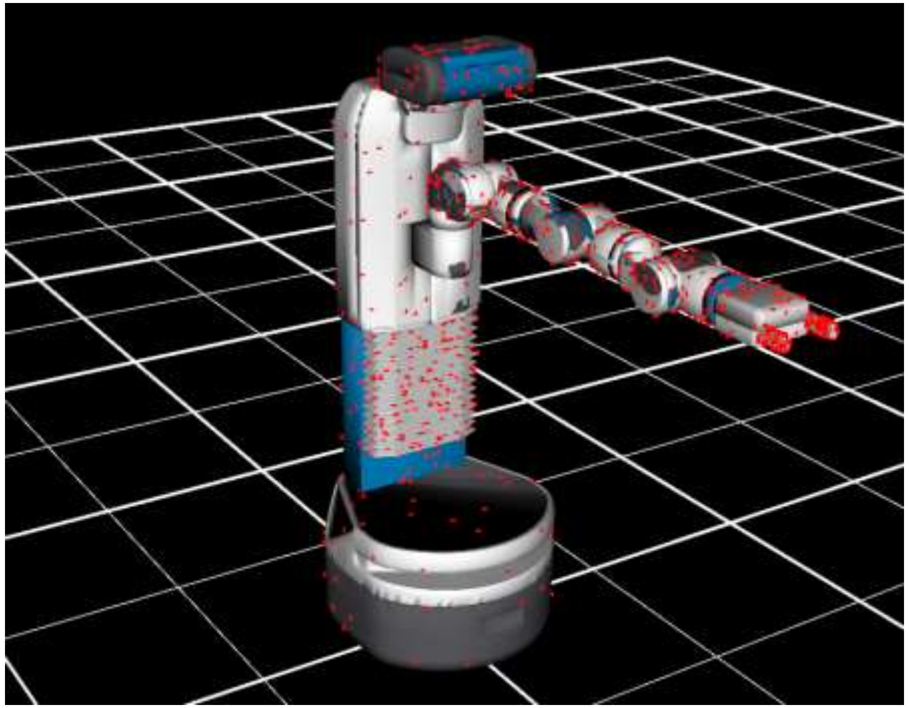


(c) Grasp Trajectory Optimization

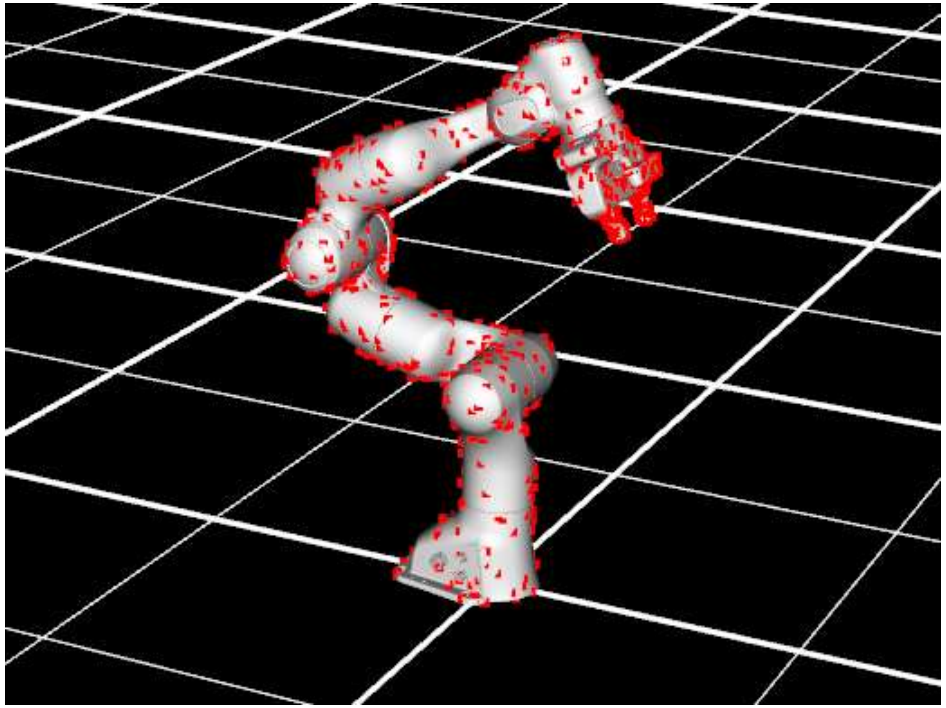
Grasping Trajectory Optimization with Point Clouds. Yu Xiang, Sai Haneesh Allu, Rohith Peddi, Tyler Summers, Vibhav Gogate
In arXiv, 2024.

Grasping Trajectory Optimization with Point Clouds

- Represent robots as point clouds (can be used for any robot)



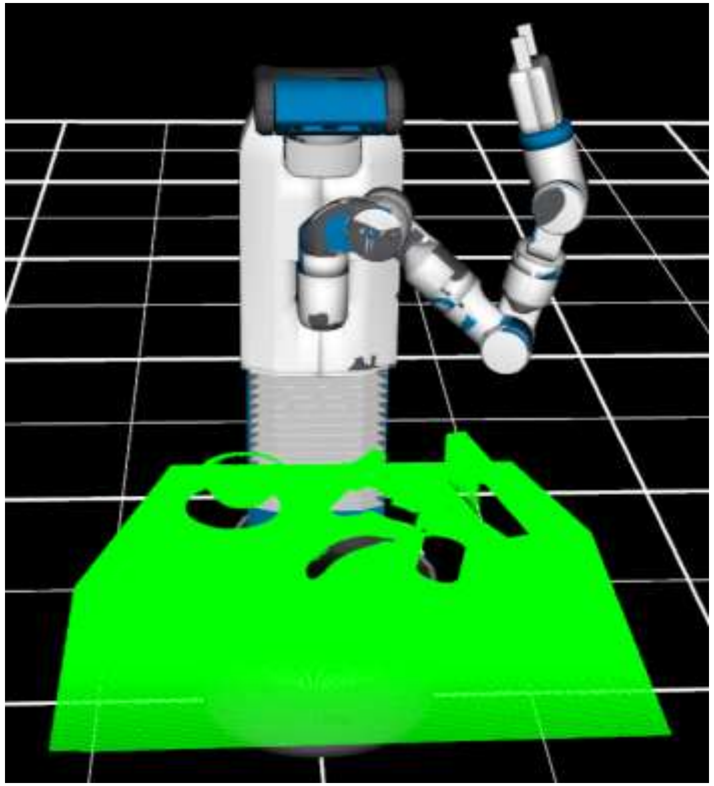
(a) A Fetch Mobile Manipulator



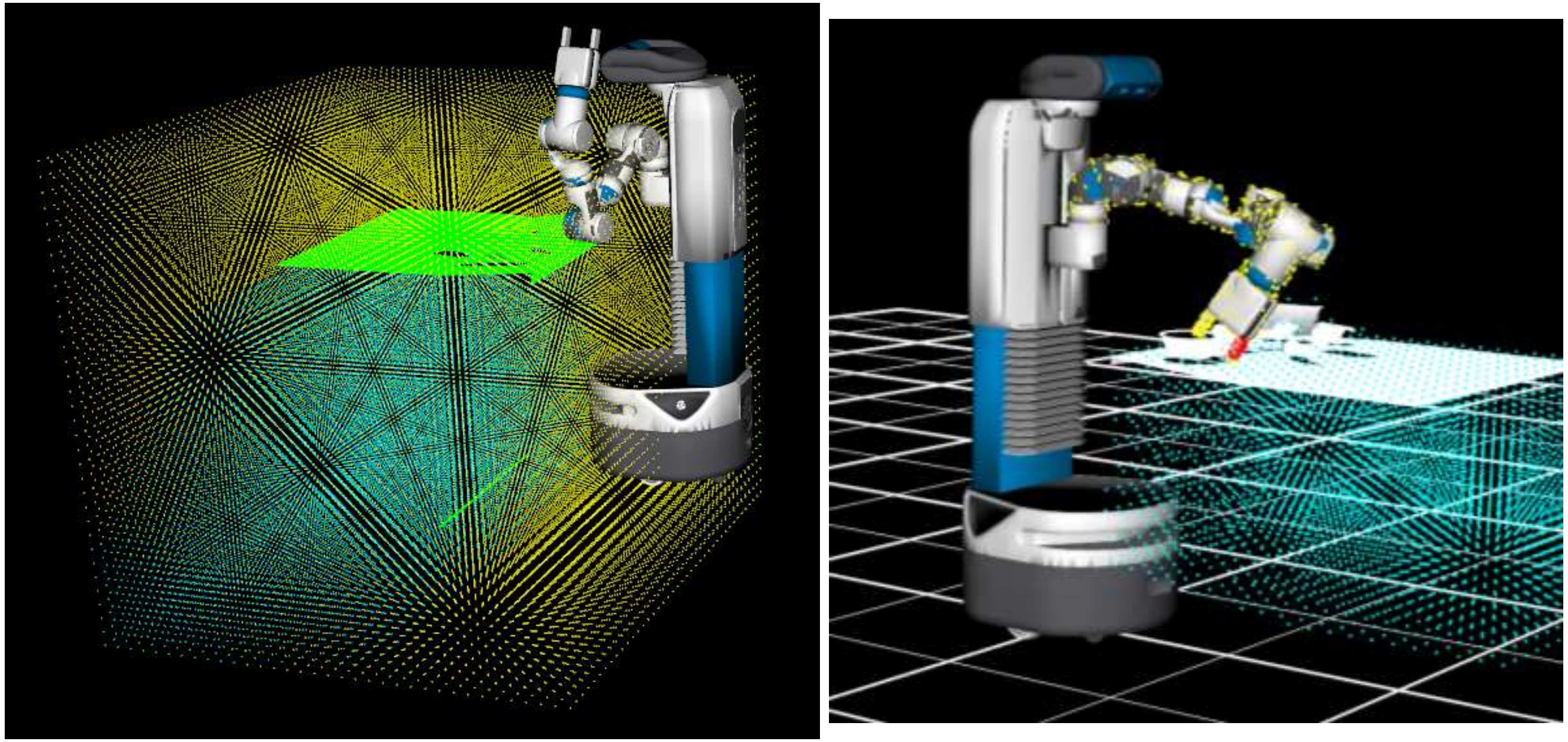
(b) A Franka Panda Arm

Grasping Trajectory Optimization with Point Clouds

- Represent task spaces as point clouds (can be used for any task)
- Build signed distance fields using point clouds for collision avoidance



(a) 3D Scene Points from a Depth Image



(b) Signed Distance Field of the Task Space

Grasping Trajectory Optimization with Point Clouds

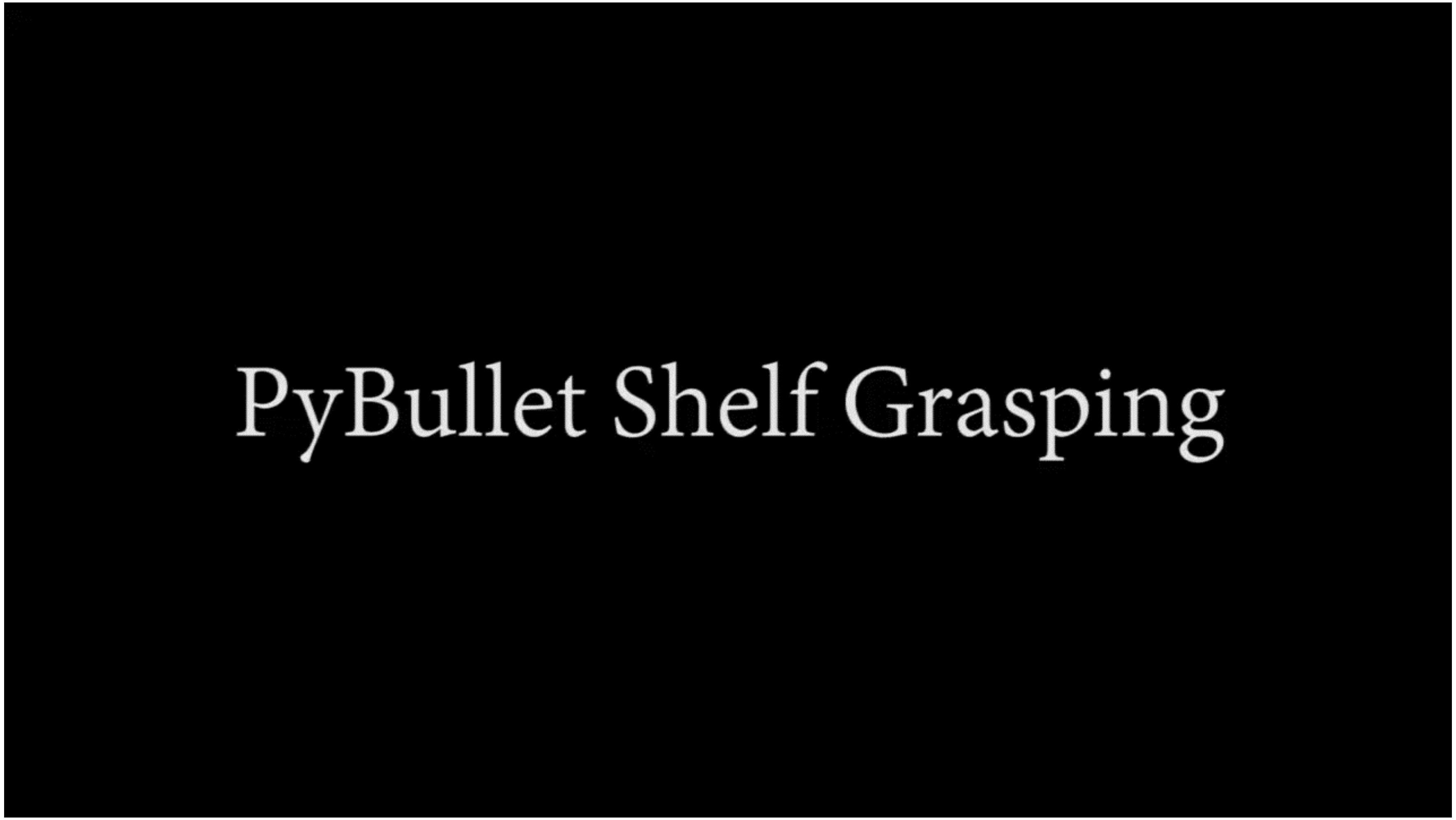
- Solve a trajectory with joint positions and joint velocities

$$\mathcal{Q} = (\mathbf{q}_1, \ldots, \mathbf{q}_T) \qquad \dot{\mathcal{Q}} = (\dot{\mathbf{q}}_1, \ldots, \dot{\mathbf{q}}_T)$$

$$\begin{aligned}
\arg\min_{\mathcal{Q},\dot{\mathcal{Q}}} \quad & \Big(\min_{i=1}^{K} \left(c_{\text{goal}}(\mathbf{T}(\mathbf{q}_T), \mathbf{T}_i) + c_{\text{standoff}}(\mathbf{T}(\mathbf{q}_{T-\delta}), \mathbf{T}_i \mathbf{T}_\Delta) \right) \\
& \quad + \lambda_1 \textstyle\sum_{t=1}^{T} c_{\text{collision}}(\mathbf{q}_t) + \lambda_2 \sum_{t=1}^{T} \|\dot{\mathbf{q}}_t\|^2 \Big) \\
\text{s.t.,} \quad & \mathbf{q}_1 = \mathbf{q}_0 \\
& \dot{\mathbf{q}}_1 = \mathbf{0}, \dot{\mathbf{q}}_T = \mathbf{0} \\
& \mathbf{q}_{t+1} = \mathbf{q}_t + \dot{\mathbf{q}}_t dt, t = 1, \ldots, T-1 \\
& \mathbf{q}_l \leq \mathbf{q}_t \leq \mathbf{q}_u, t = 1, \ldots, T \\
& \dot{\mathbf{q}}_l \leq \dot{\mathbf{q}}_t \leq \dot{\mathbf{q}}_u, t = 1, \ldots, T,
\end{aligned}$$

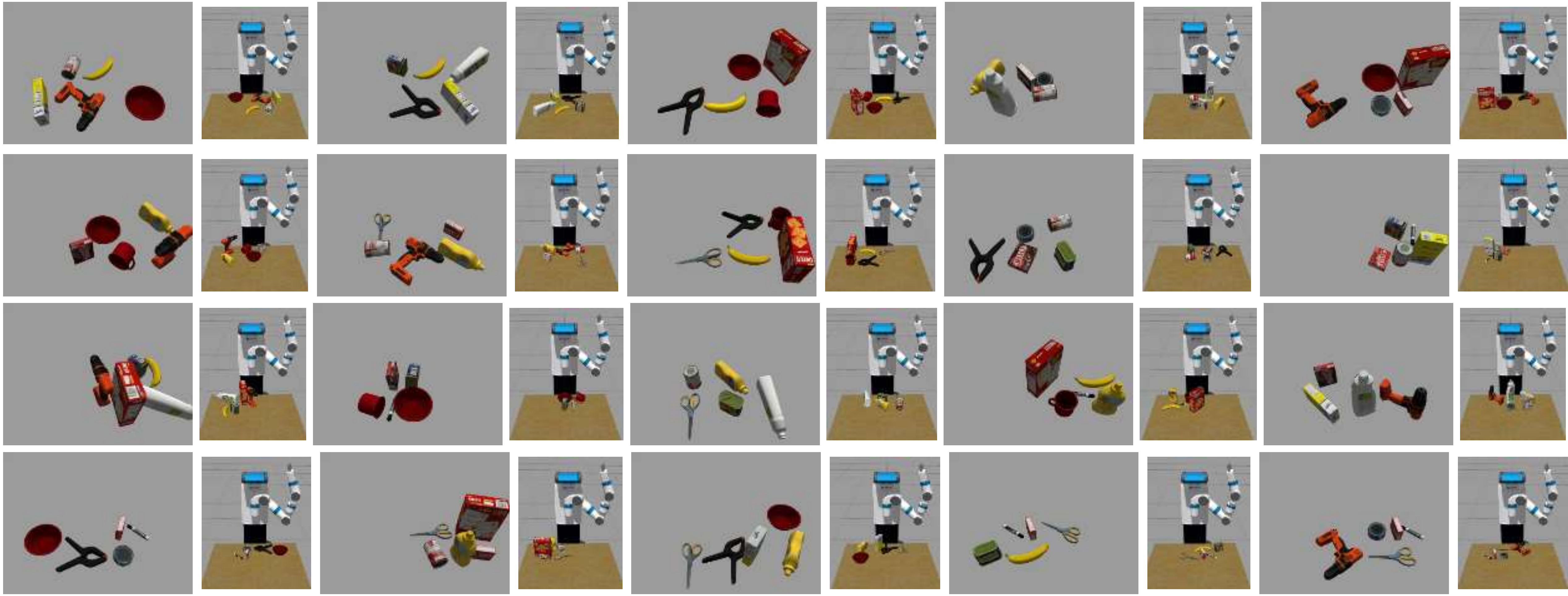
Grasping Trajectory Optimization with Point Clouds

- Simulation results



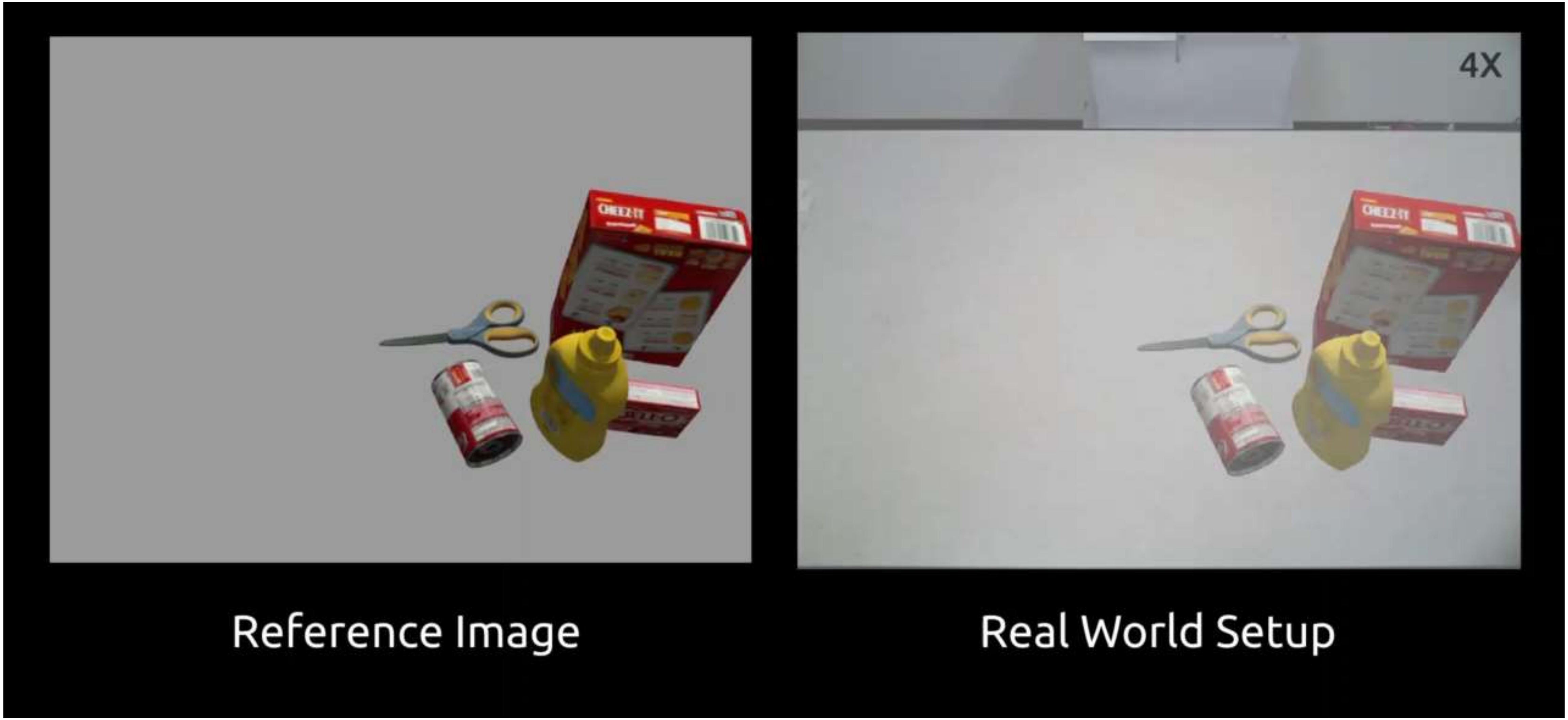
SceneReplica Benchmark

20 Scenes



SceneReplica, ICRA'24: https://irvlutd.github.io/SceneReplica/

Real-World Scene Setup



SceneReplica Benchmark

Method #	Perception	Grasp Planning	Motion Planning	Control	Ordering	Pick-and-Place Success	Grasping Success
			Model-based Grasping				
1	PoseRBPF [21]	GraspIt! [22] + Top-down	OMPL [24]	MoveIt	Near-to-far	58 / 100	64 / 100
1	PoseRBPF [21]	GraspIt! [22] + Top-down	OMPL [24]	MoveIt	Fixed	59 / 100	59 / 100
2	PoseCNN [19]	GraspIt! [22] + Top-down	OMPL [24]	MoveIt	Near-to-far	47 / 100	48 / 100
2	PoseCNN [19]	GraspIt! [22] + Top-down	OMPL [24]	MoveIt	Fixed	40 / 100	45 / 100
3	GDRNPP [34], [36]	GraspIt! [22] + Top-down	OMPL [24]	MoveIt	Near-to-far	**66 / 100**	69 / 100
3	GDRNPP [34], [36]	GraspIt! [22] + Top-down	OMPL [24]	MoveIt	Fixed	62 / 100	64 / 100
			Model-free Grasping				
4	UCN [26]	GraspNet [28] + Top-down	OMPL [24]	MoveIt	Near-to-far	43 / 100	46 / 100
4	UCN [26]	GraspNet [28] + Top-down	OMPL [24]	MoveIt	Fixed	37 / 100	40 / 100
5	UCN [26]	Contact-graspnet [29] + Top-down	OMPL [24]	MoveIt	Near-to-far	60 / 100	63 / 100
5	UCN [26]	Contact-graspnet [29] + Top-down	OMPL [24]	MoveIt	Fixed	60 / 100	64 / 100
6	MSMFormer [27]	GraspNet [28] + Top-down	OMPL [24]	MoveIt	Near-to-far	38 / 100	41 / 100
6	MSMFormer [27]	GraspNet [28] + Top-down	OMPL [24]	MoveIt	Fixed	36 / 100	41 / 100
7	MSMFormer [27]	Contact-graspnet [29] + Top-down	OMPL [24]	MoveIt	Near-to-far	57 / 100	65 / 100
7	MSMFormer [27]	Contact-graspnet [29] + Top-down	OMPL [24]	MoveIt	Fixed	61 / 100	**70 / 100**
8	MSMFormer [27]	Top-down	OMPL [24]	MoveIt	Fixed	56 / 100	59 / 100
			End-to-end Learning-based Grasping				
9	Dex-Net 2.0 [37] (Top-Down Grasping)		OMPL [24]	MoveIt	Algorithmic	43 /100	51 / 100
			Ground truth pose-based Grasping				
10	Ground truth object pose	GraspIt! [22] + Top-down	OMPL [24]	MoveIt	Near-to-far	78 / 100	82 / 100
11	Ground truth object pose	GraspIt! [22] + Top-down	OMPL [24]	MoveIt	Fixed	78 / 100	87 / 100

Grasping Trajectory Optimization with Point Clouds

- Real world experiments

Method #	Perception	Grasp Planning	Motion Planning	Control
			Model-free Grasping	
1	MSMFormer [33]	Contact-graspnet [29] + Top-down	OMPL [34]	MoveIt
2	MSMFormer [33]	Contact-graspnet [29] + Top-down	**GTO (Ours)**	MoveIt

Ordering	Pick-and-Place Success	Grasping Success
Near-to-far	57 / 100	65 / 100
Near-to-far	**65 / 100**	**71 / 100**

Grasping Trajectory Optimization with Point Clouds. Yu Xiang, Sai Haneesh Allu, Rohith Peddi, Tyler Summers, Vibhav Gogate
In arXiv, 2024.

Using 3D Point Clouds

- Pros
  - No need to build 3D models
  - Direct sensor input from RGB-D cameras
  - Encode appearance and 3D geometry

- Cons
  - It is difficult to capture depth for certain objects (flat, thin, transparent, metal)
  - Planning from partial observations

Do We Need 3D Representations for Robot Manipulation?

- Depends on your goals
- Goal: A repetitive task in a specific environment
  - Probably no need
- Goal: enabling any robot to do any task in any environment
  - We need to think about generalization
  - 3D Understanding might help generalization in robots, tasks and environments
  - 3D representation + policy learning (e.g., 3D diffusion policy)

Thank you!

SONY DARPA