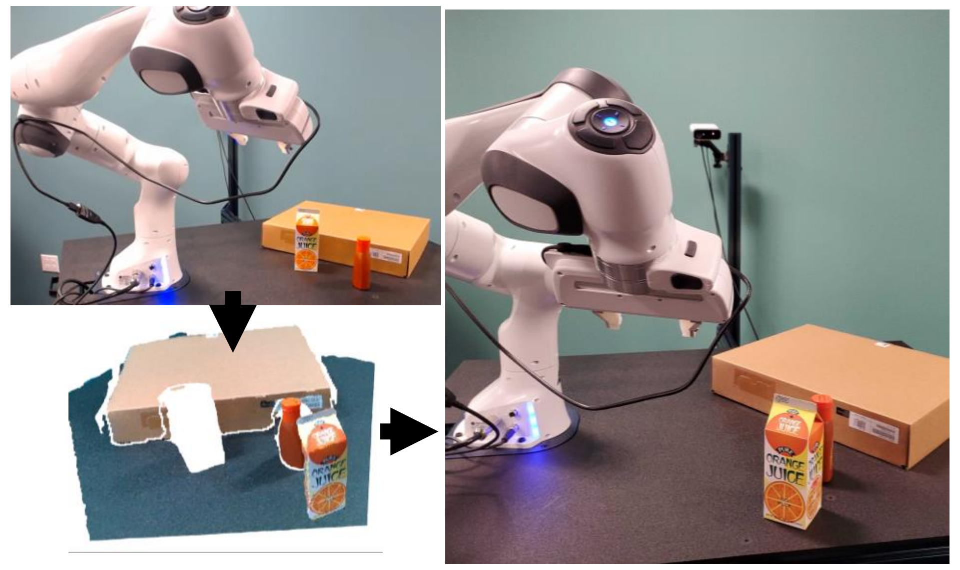
Predicting Stable Configurations for Semantic Placement of Novel Objects

Chris Paxton, Chris Xie, Tucker Hermans, Dieter Fox

Introduction

Semantic placement is an important task for autonomous robots. This includes both motion planning, placement planning, and satisfying high-level directives. Our goal is to:

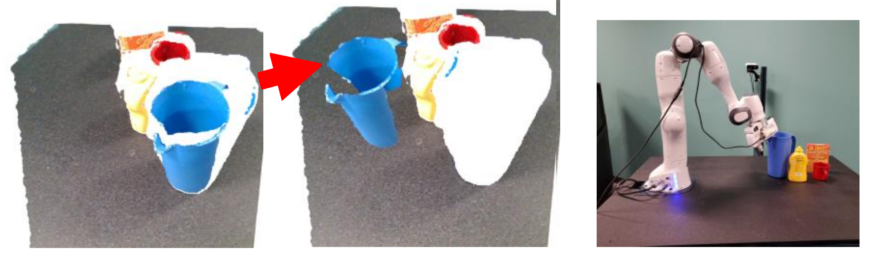
- Learn models that capture spatial relationships between different unknown objects
- Determine which placements are reasonable, to handle widely varied scene geometry
- Integrate into a sampling-based planning framework for real-world execution



Our method works on a wide range of objects, using unseen-object instance segmentation and grasping models to identify what can be moved.

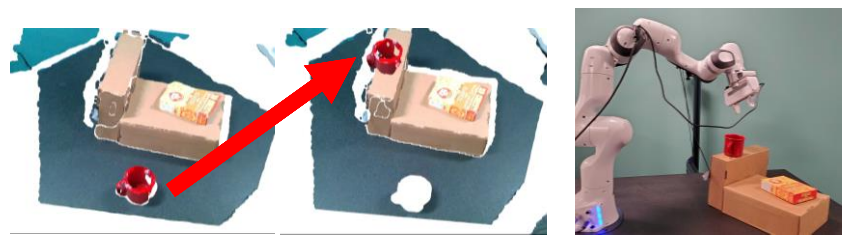
"Place the mug in front of the blue pitcher."

"Place the pitcher to the left of the mustard."

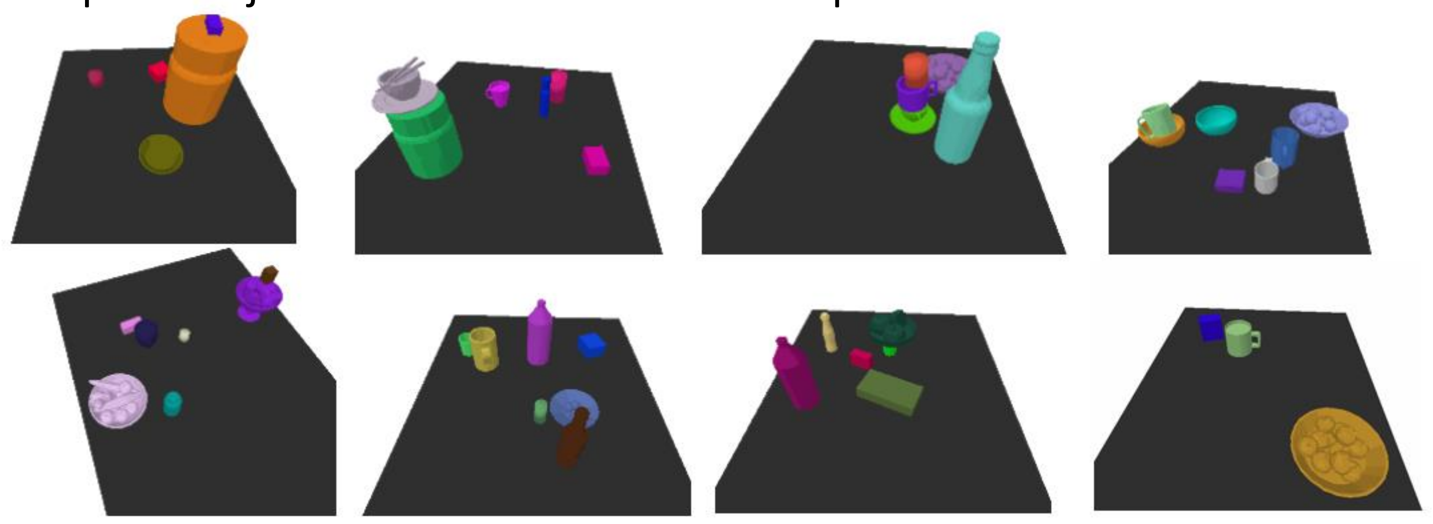


"Place the mustard behind the cookie box."

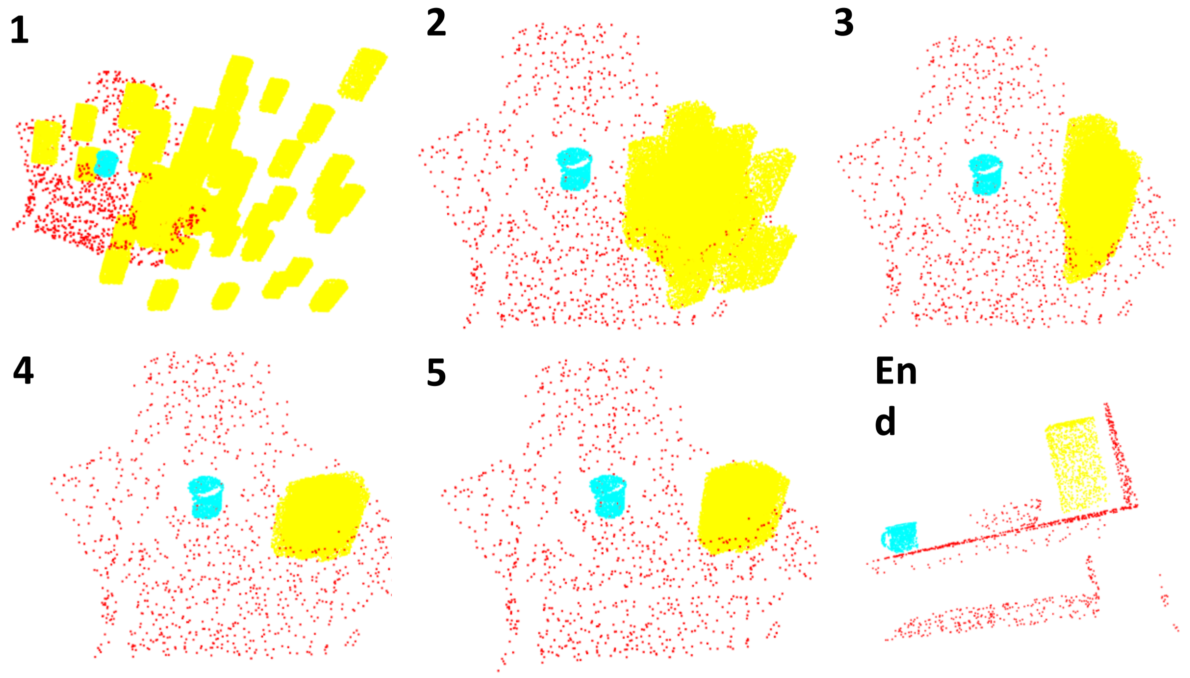
"Place the mug on top of the box."



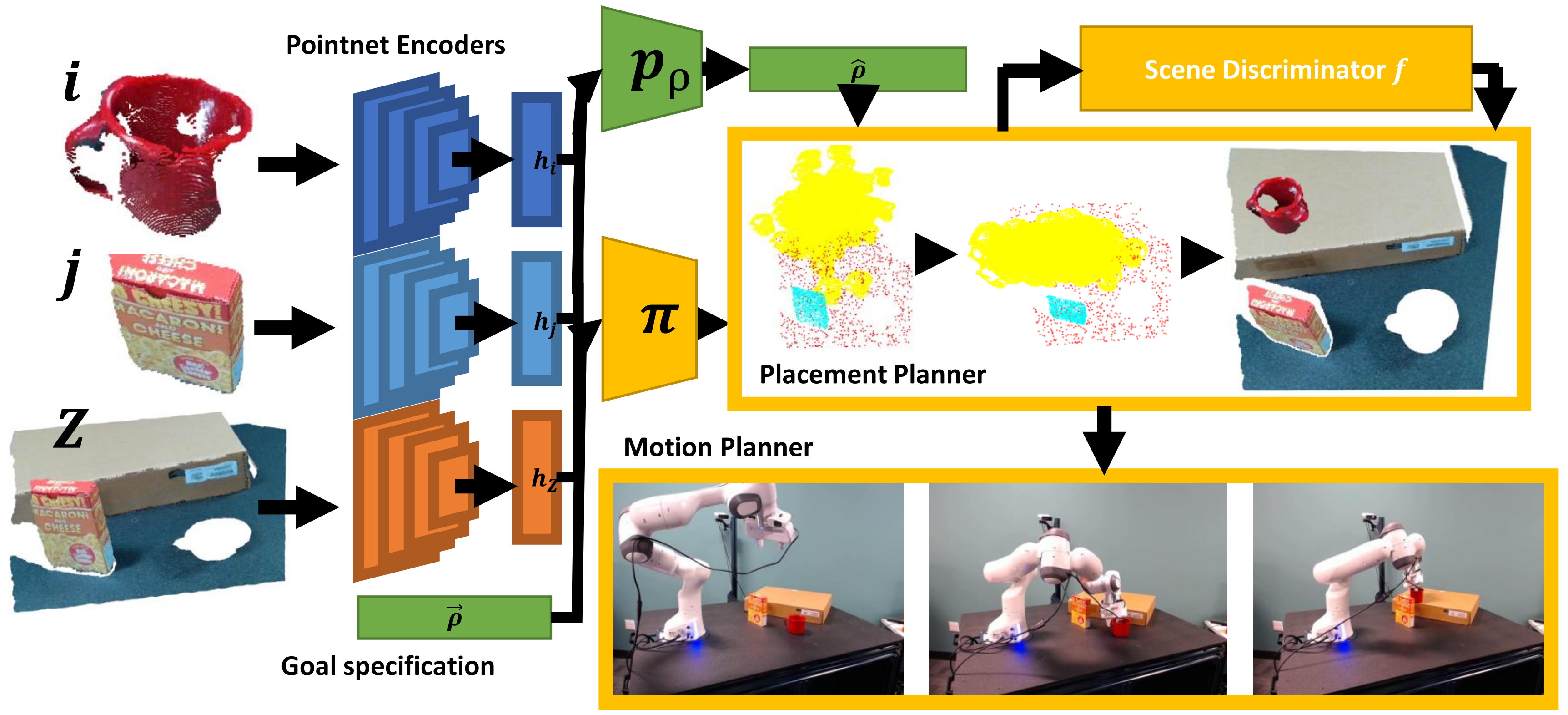
Dataset: our data is generated in simulation. We create scenes in PyBullet and use physics to place objects and let them settle on top of one another.



Planning from Point Clouds



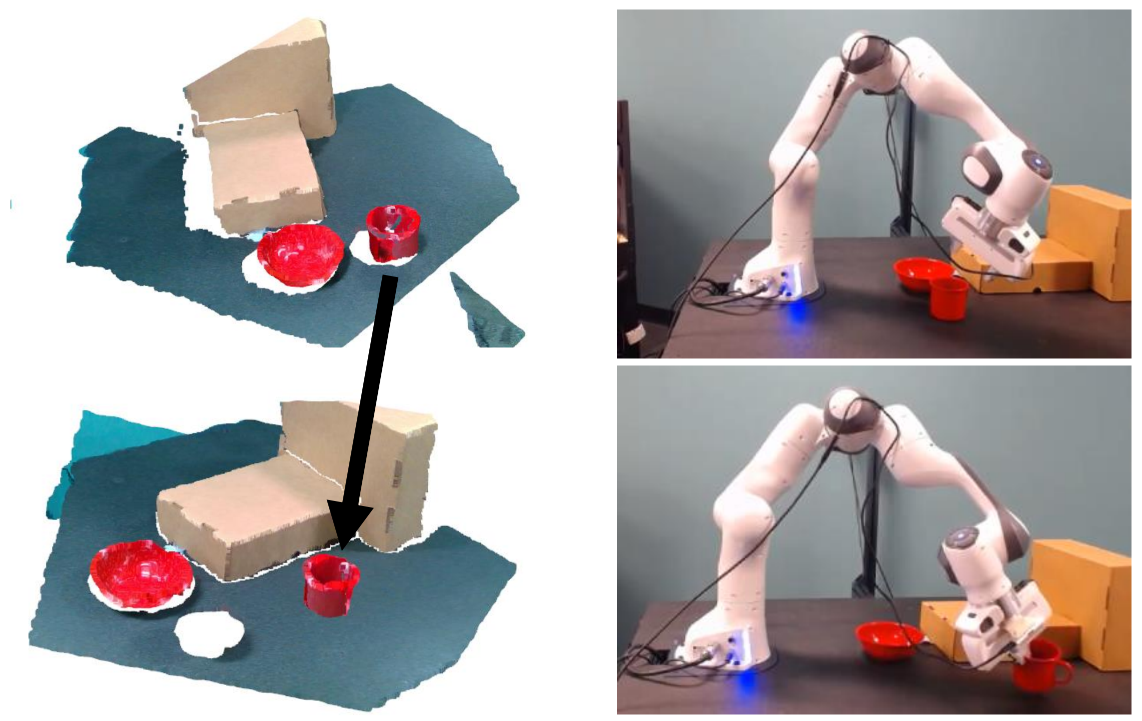
Our planner samples poses from a learned Mixture Density Network distribution, then refines them to find a stable pose that matches learned predicates. *Above: object point cloud positions being refined over time.*



Sampling-Based Planning Algorithm

- Take segmented point clouds for two different objects: A and B, scene S, predicate goal p
- Get initial distribution from $\pi(p, A, B, S)$
- Loop for N iterations:
  - Sample a set of positions from distribution
  - Transpose point clouds to these positions
  - Evaluate predicate classifiers and discriminator
  - Solve kinematics and score solutions; refit dis
- Take best result, execute motion plan

Example: place mug right of box



Experiments

- Performed experiments both in the real world and in simulation.
- Tested predicate models against a reasonable baseline implementation from the point clouds of the objects; F1 scores for individual predicates are comparable or better. Overall, we see a large increase when matching all predicates.
- Adding the discriminator makes a large difference in simulation, and discriminator scores are correlated with stability.
- Real-world experiments allowed us to pick and place objects in a variety of novel scenes.

Below: experiments done in simulation showing usefulness of discriminator-based approach.

Variant	Predicate Found	Realistic Found	Successful	Stable Pose
Full, $\lambda = 100$	93	**87**	**84**	**71**
Full, $\lambda = 1$	**94**	81	78	67
No disc	100	3	3	25
Mean only	96	27	27	39
MDN prior	99	6	6	42

Conclusions

- We described a system for semantic placement of unknown objects in unknown scenes.
- We learn (1) a predicate classifier, (2) a scene discriminator to identify "realistic" configurations, and (3) a Mixture Density Network prior with which we initialize our planning process.
- Our system is trained purely on simulation data, and transfers from sim to real.
- We showed experiments on a variety of scenes and queries.