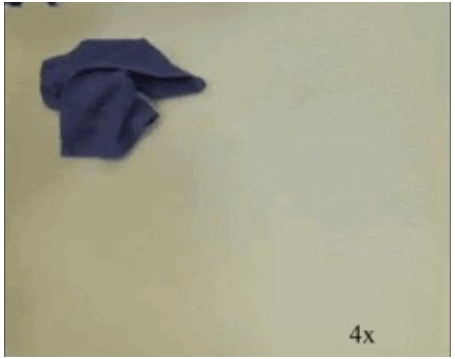
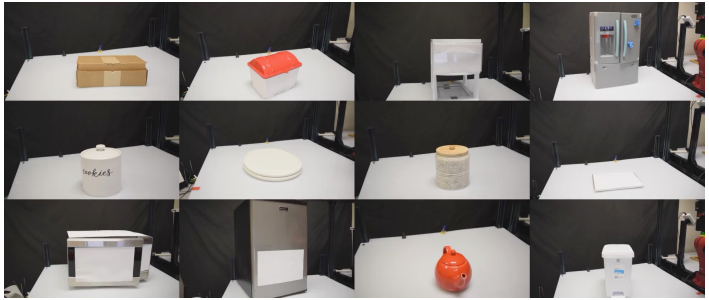
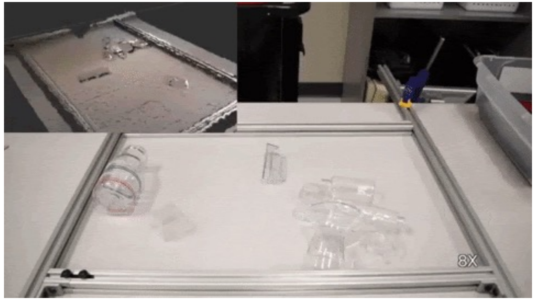
Relational Affordance Learning for Robot Manipulation

David Held
Carnegie Mellon University







“Vision Researchers”

“Robotics Researchers”

Sensor inputs → Perception System → State → Planning → Actions

Robot

“Vision Researchers” “Perceptual Robotics Researchers” “Robotics Researchers”

Sensor inputs → Perception System → State → Planning → Actions

Can we build a better system by thinking about these parts together?

Clarification: We are NOT <u>merging</u> the perception and planning!
We are just <u>thinking</u> about both parts together

"Vision Researchers" "Robotics Researchers"

"But why can't roboticists just use the output of a computer vision method?"

Detection Segmentation

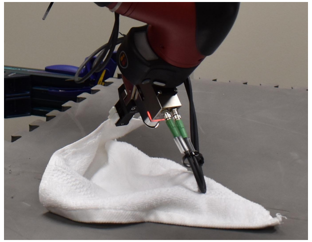
Computer vision: "Understand what is in an image"

“Vision Researchers”

“Robotics Researchers”

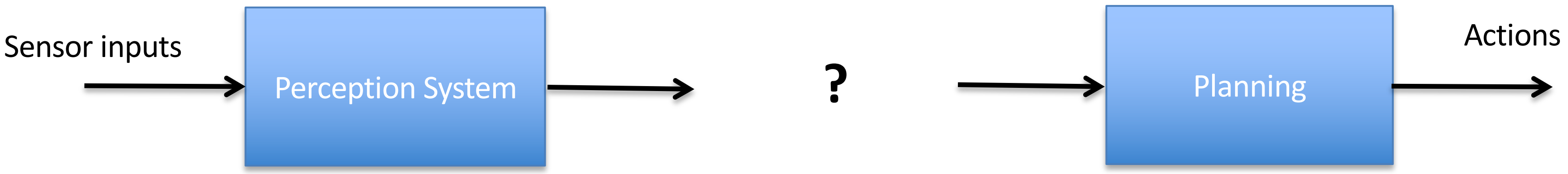
“But why can’t roboticists just use the output of a computer vision method?”

+



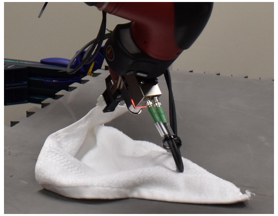
Robotics goal: Understand what will happen if a robot interacts with its environment

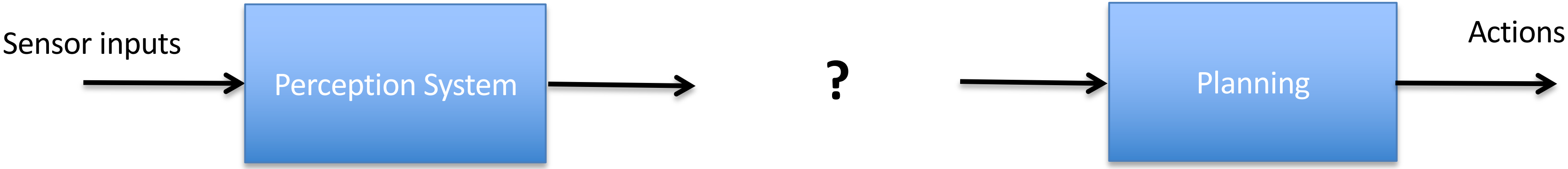
~~Computer vision: “Understand what is in an image”~~



How do we bridge the gap between perception and planning?

Representation:
- Can infer from observations / interaction
- Useful for planning
- Suitable for the task

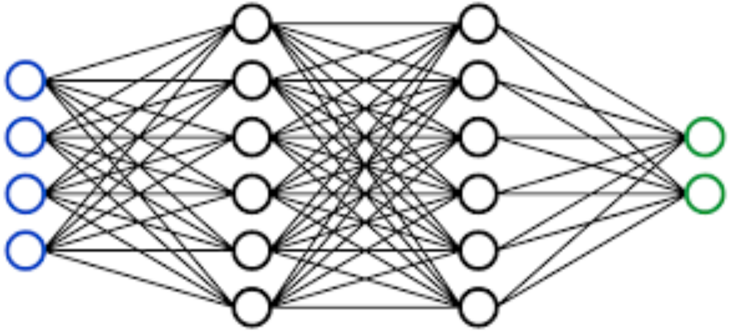




How should the robot represent this object…

… in order to plan how to achieve a task?

Does not generalize well to unseen objects or unseen configurations

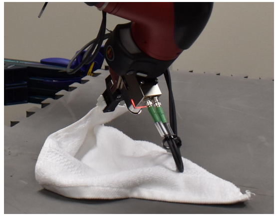


Black Box

?

Representation:

- Can infer from observations / interaction
- Useful for planning
- Suitable for the task

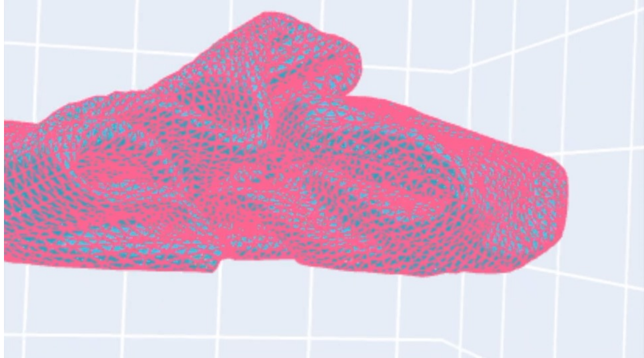


Sensor inputs → Perception System → ? → Planning → Actions

How should the robot represent this object…

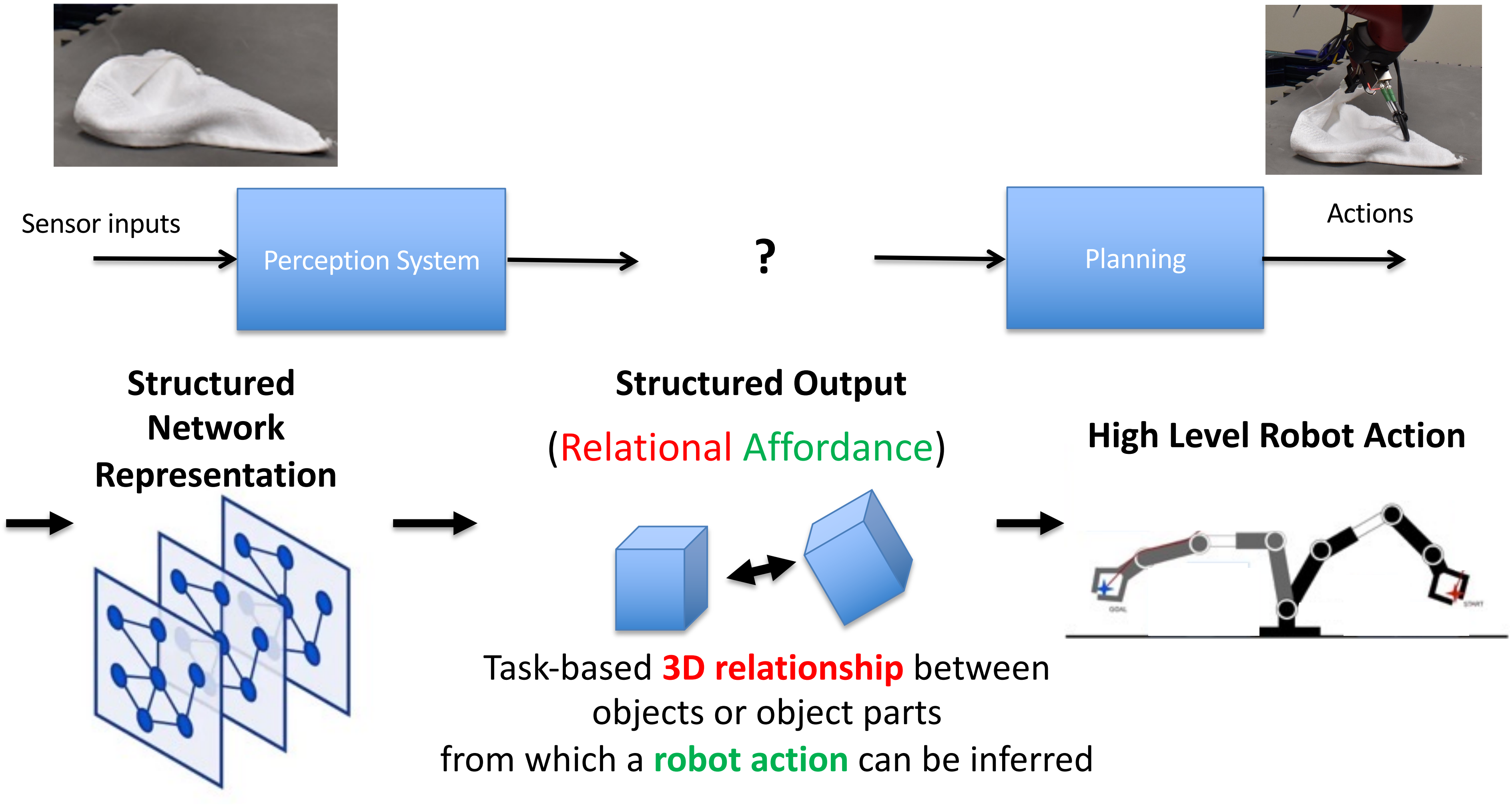
… in order to make decisions of how to achieve a task?

- **Difficult to infer from observations**
- **Slow for planning**



Full 3D State [RSS 2022]

?


Sensor inputs
Perception System
?
Planning
Actions
Structured Network Representation
Structured Output
(Relational Affordance)
High Level Robot Action
Task-based 3D relationship between
objects or object parts
from which a robot action can be inferred

Robot Planning Hierarchy

Task Planning

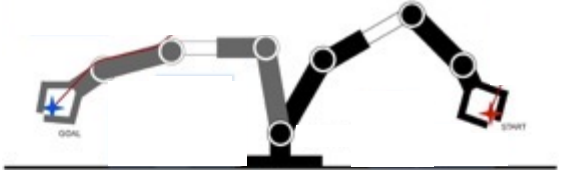
1. Pour the flour and milk to into a bowl
2. Crack the eggs and add to the bowl
3. Whisk the ingredients together
4. Pour the mixture into the pan

Output: Sequence of skills needed to complete a long horizon task

Skill

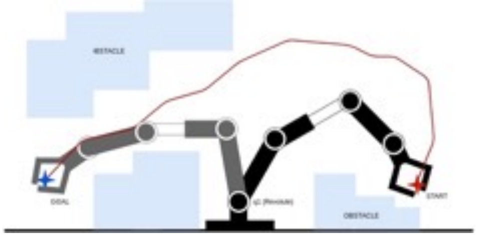
Folding skill

Relational Affordance Learning



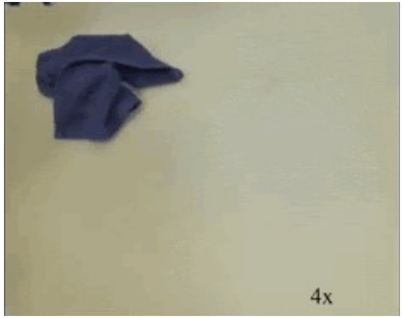
Output: Start + end points of a trajectory / sequence of subgoals

Motion Planning / RL + Low-level Control



Output: Robot Trajectory

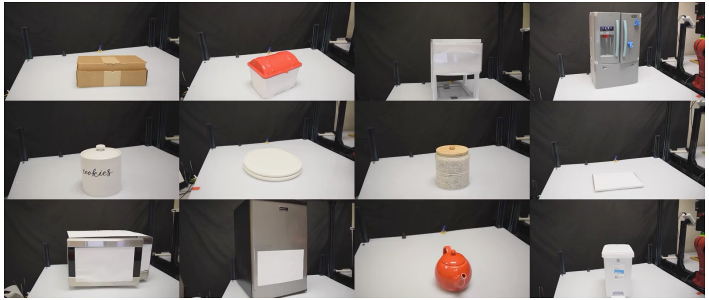
Relational Affordance Learning



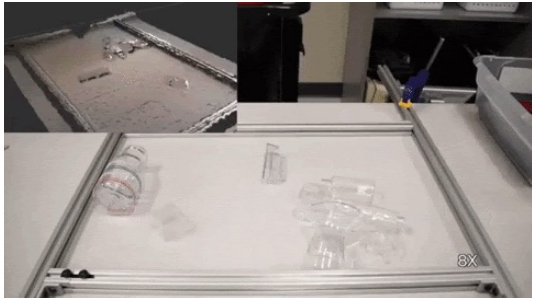
(CoRL 2021, RSS 2022)

(CoRL 2021)

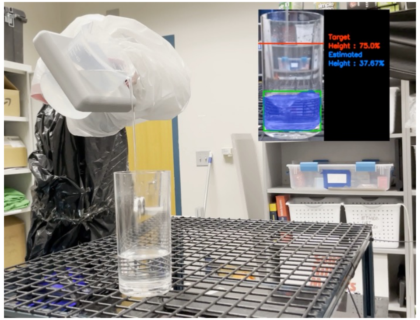
(CoRL 2022)



(RSS 2022 - Best Paper Finalist)



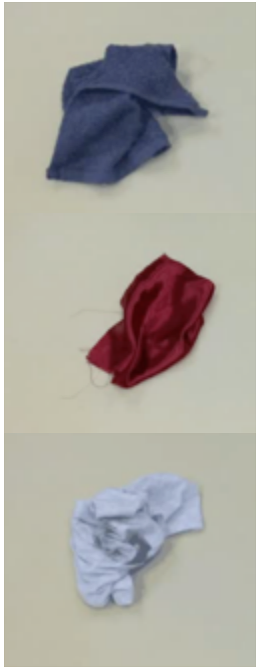
(ICRA 2020)



(ICRA 2022)

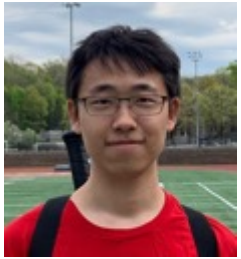
How can robots learn a dynamics model for complex manipulation tasks?

(CoRL 2021, RSS 2022)



Cloth Dynamics model

robot actions

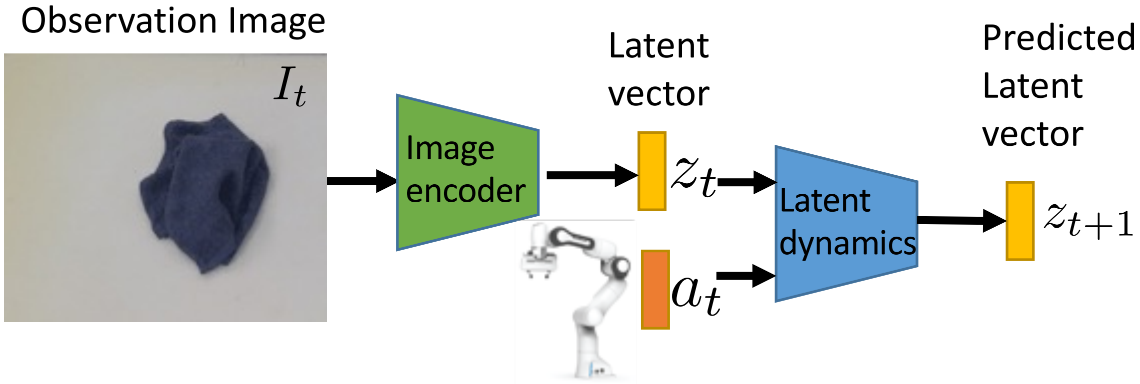


Xingyu Lin

Yufei Wang

Approach 1: Learn a latent vector dynamics model

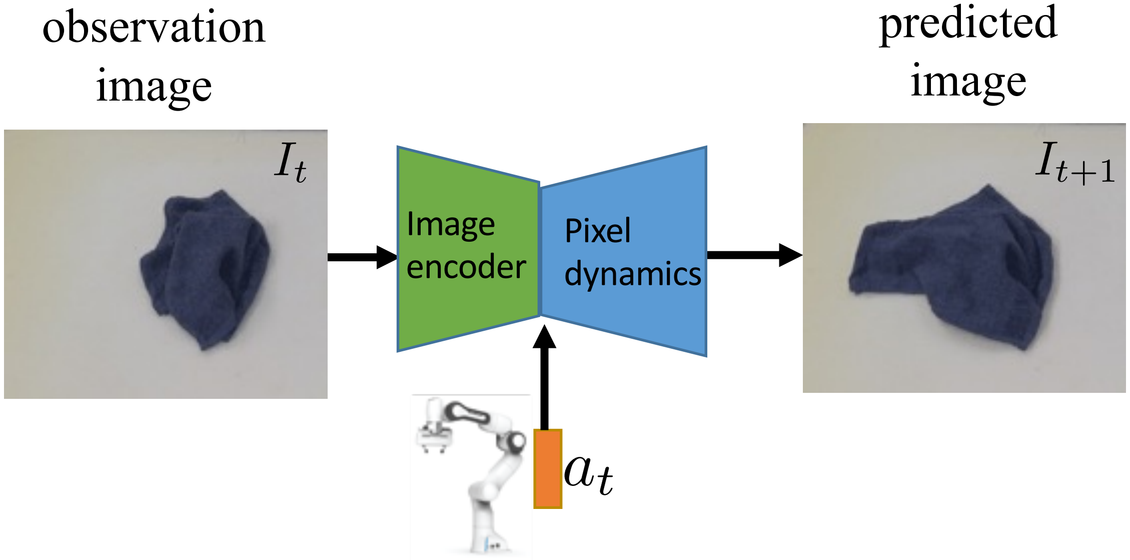
[Hafner, et al 2019, Yan et al. 2020]



Lacks environmental structure, making generalization difficult

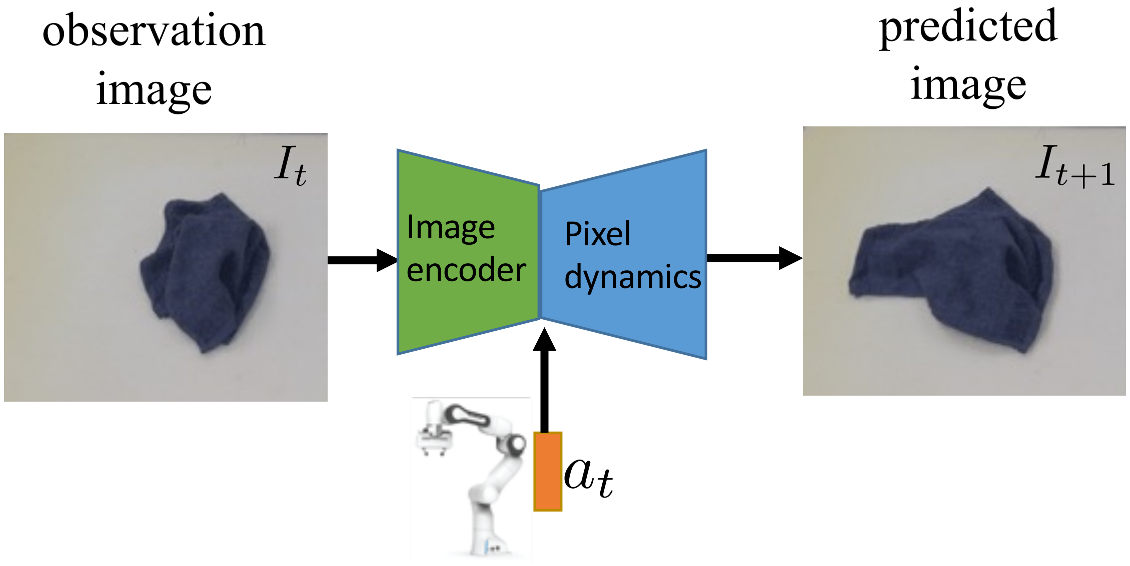
Approach 2: Learn a <u>pixel</u> dynamics model

[Finn et al. 2017, Hoque et al. 2020]

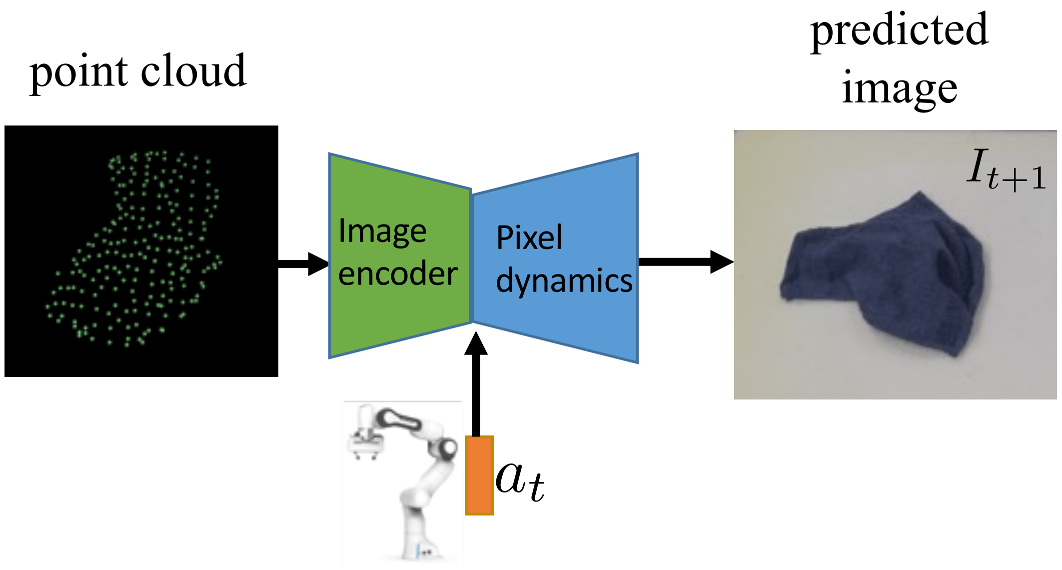


Pixel dynamics may not sufficiently capture the underlying physics of the cloth

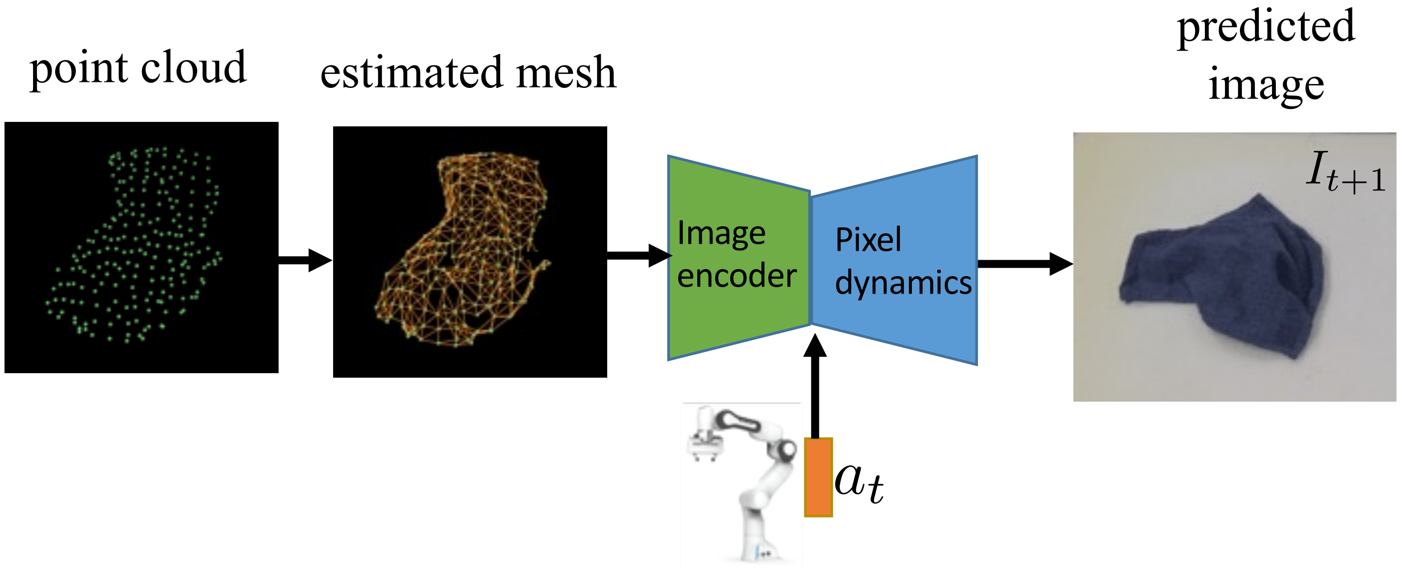
Our approach: Learn a **graph** dynamics model



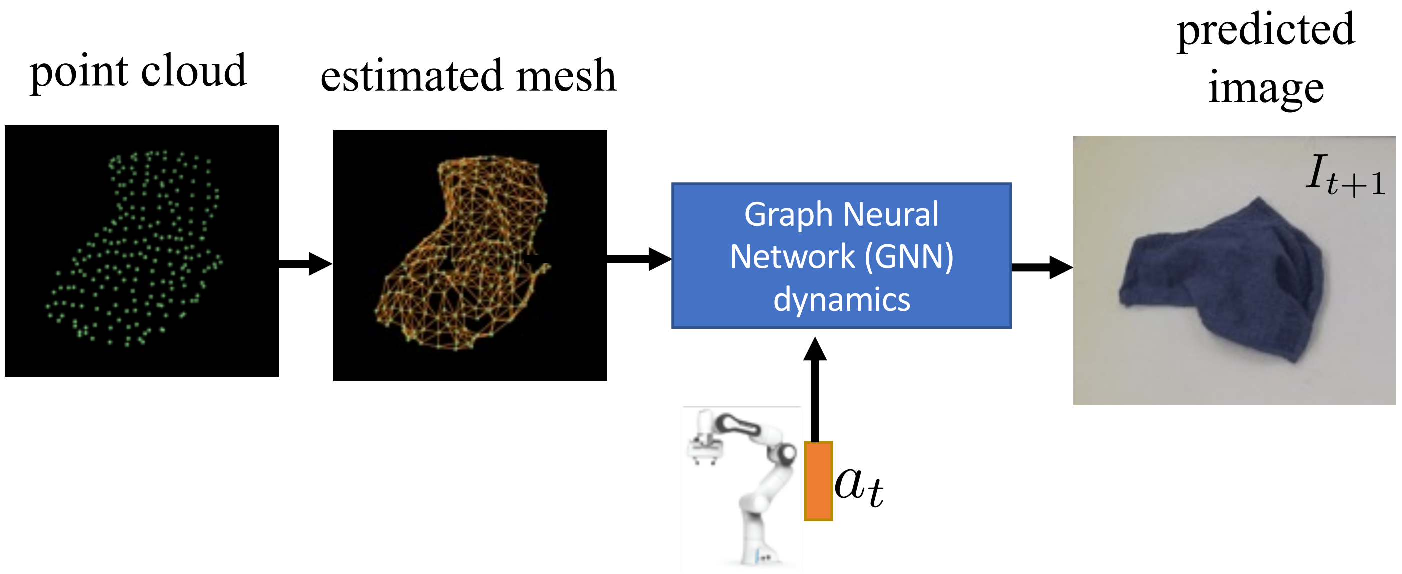
Our approach: Learn a **graph** dynamics model



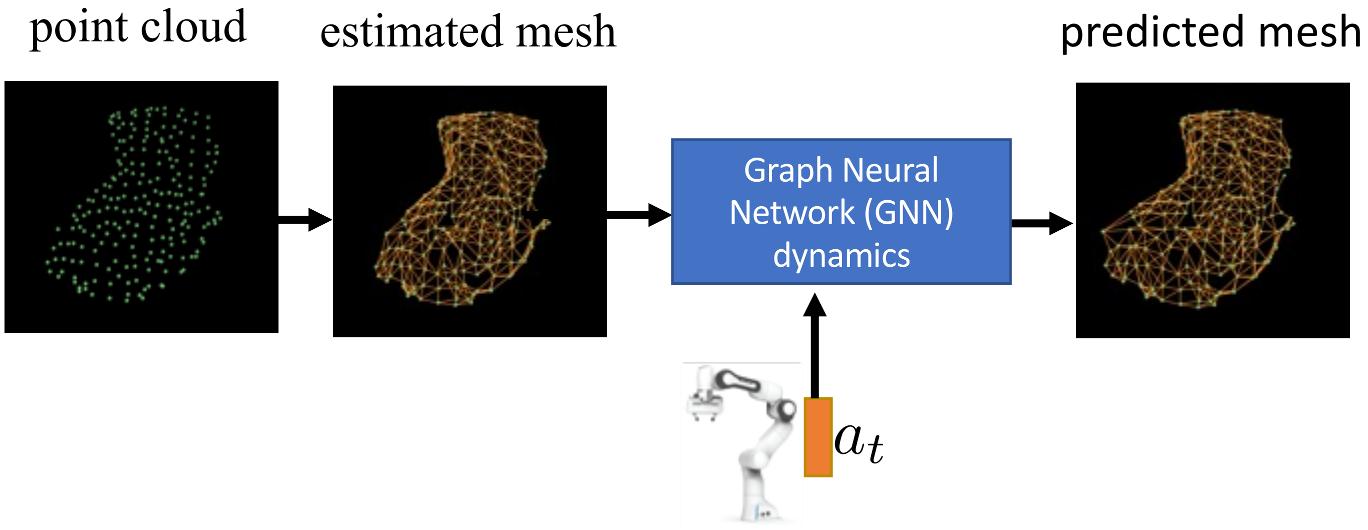
Our approach: Learn a <u>**graph**</u> dynamics model



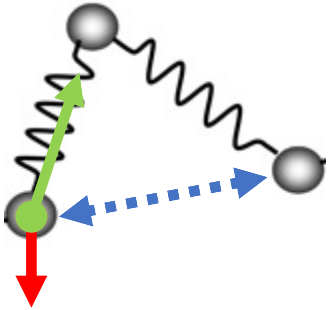
Our approach: Learn a **graph** dynamics model



Our approach: Learn a **graph** dynamics model



Mesh = Model of the cloth physics



Spring force

Gravity

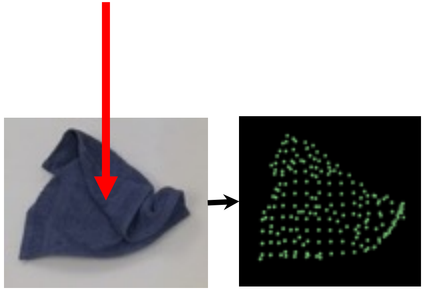
Collison force

Friction

Cloth can be modeled by a set of points connected by springs

Challenges

What is the state of the cloth underneath the surface?



RGBD observation

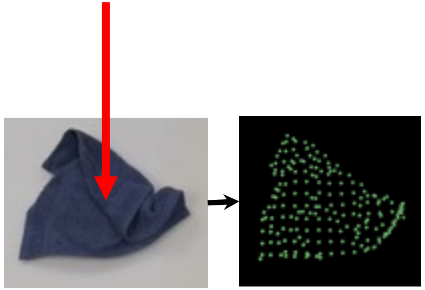
Voxelized Point cloud

How are these points connected on the underlying cloth mesh?

We cannot even see most of the points due to self-occlusions!

How to handle occlusions?

What is the state of the cloth underneath the surface?



RGBD observation

Voxelized Point cloud

Best solution:

- Estimate a distribution over the full configuration of the cloth
- Estimate uncertainty over the occluded regions

(RSS 2022 + Ongoing work)

This project:

Graph Dynamics

\+

Simple Approach for Occlusions

>

Pixel Dynamics

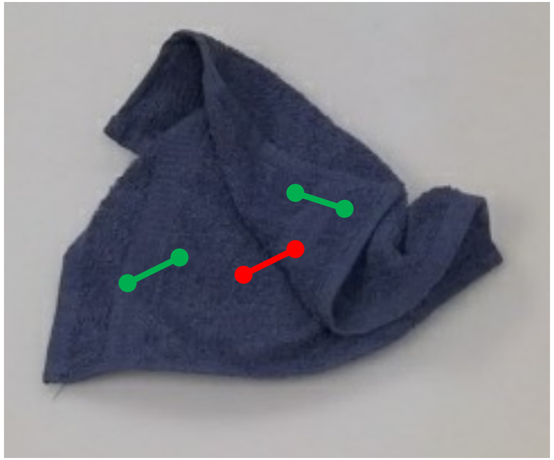
or

Latent Vector Dynamics

or

Model-free Policy

How to handle occlusions?



Graph Dynamics

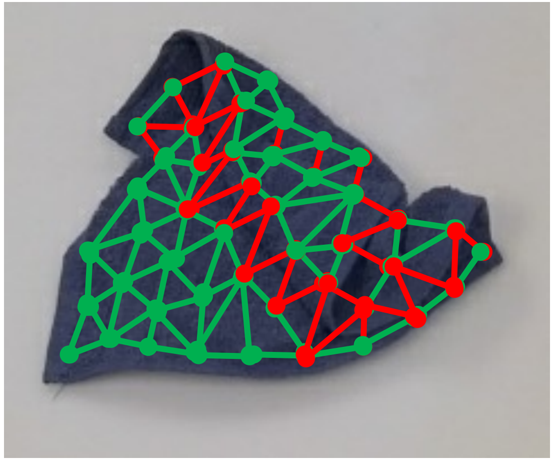
+

Simple Approach for Occlusions

Our solution:

1. Learn to estimate how the **visible points** are connected
2. Create a graph based on the estimated connectivity of the **visible points**

How to handle occlusions?



Visible Connectivity Graph

Graph Dynamics

+

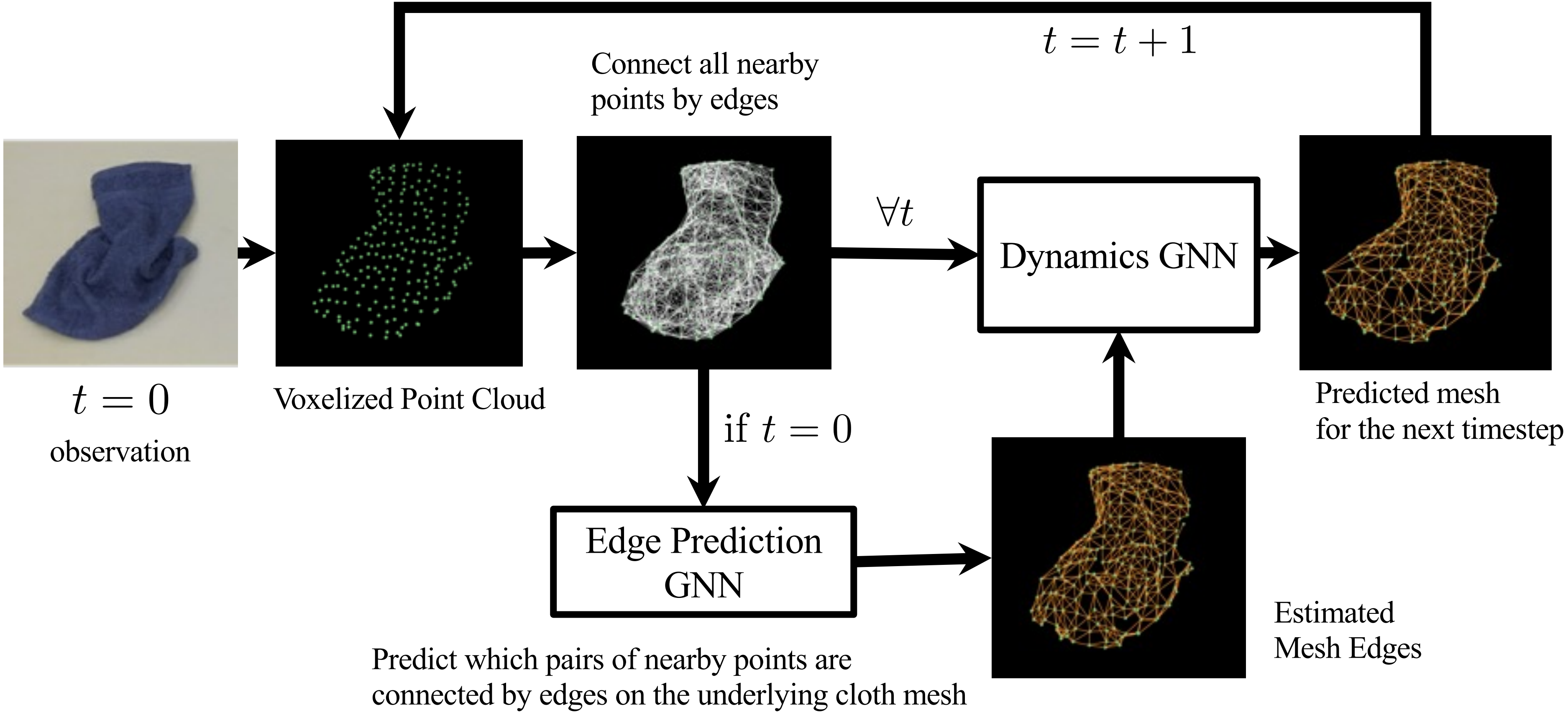
<u>Simple</u> Approach for Occlusions

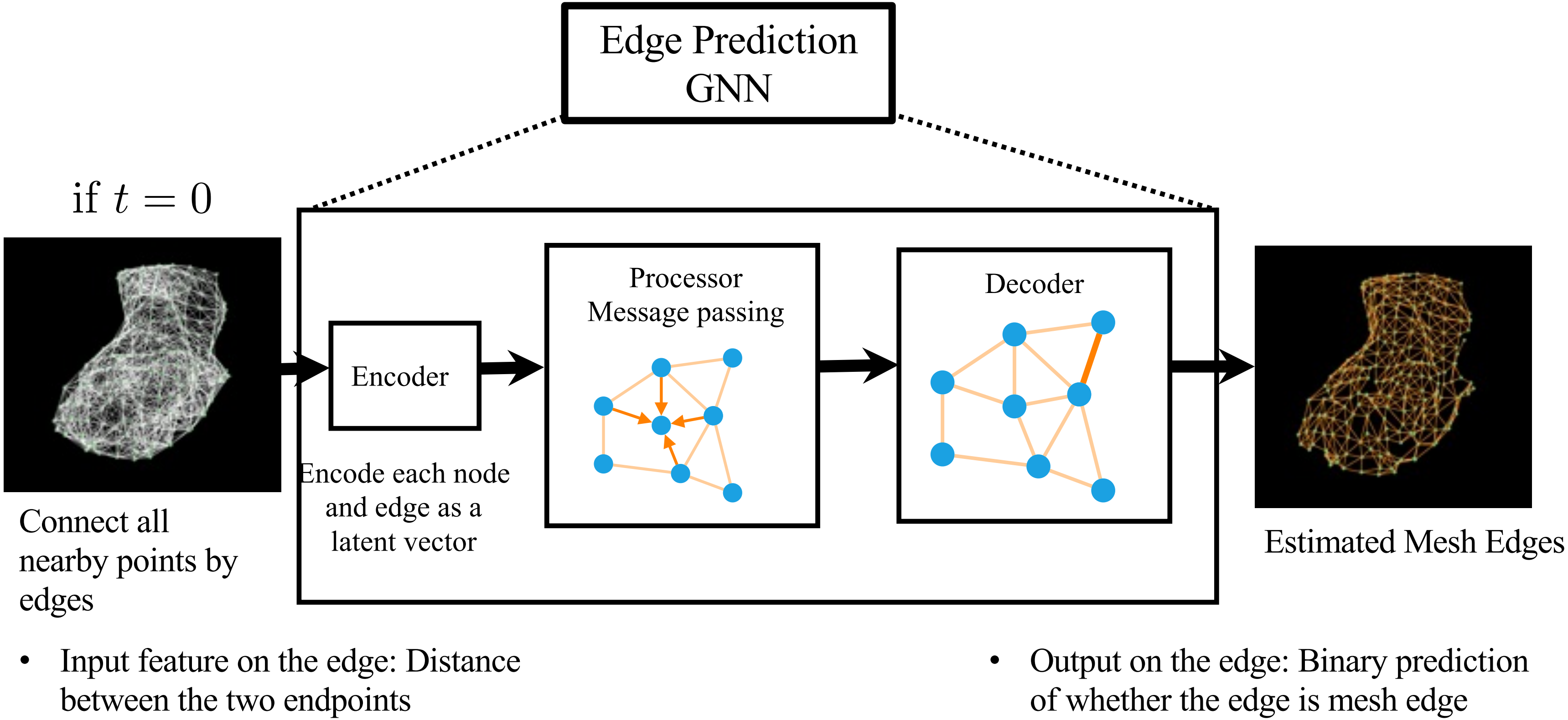
Our solution:

1. Learn to estimate how the **visible points** are connected
2. Create a graph based on the estimated connectivity of the **visible points**
3. Learn a dynamics model over this graph of **visible points**!

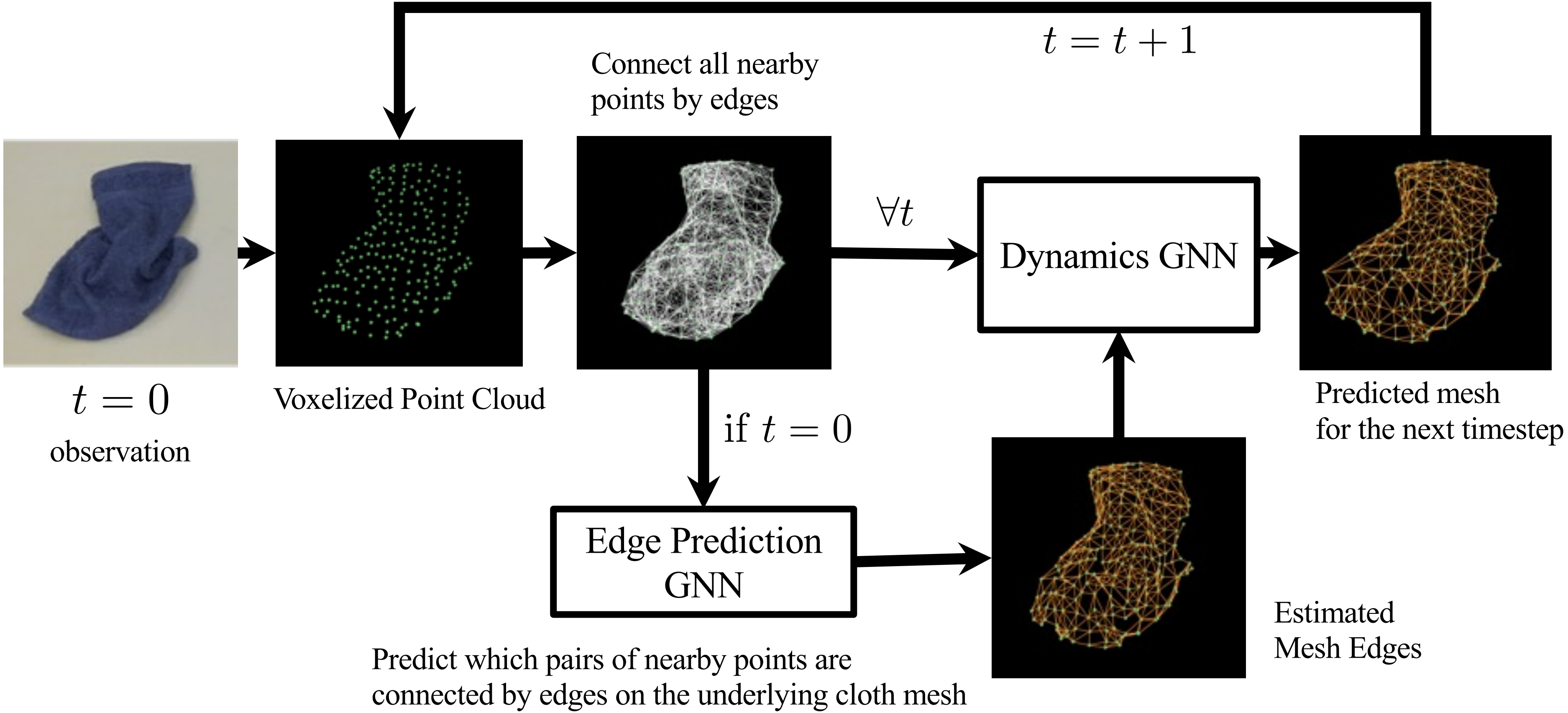
"Visible Connectivity Dynamics" (VCD)

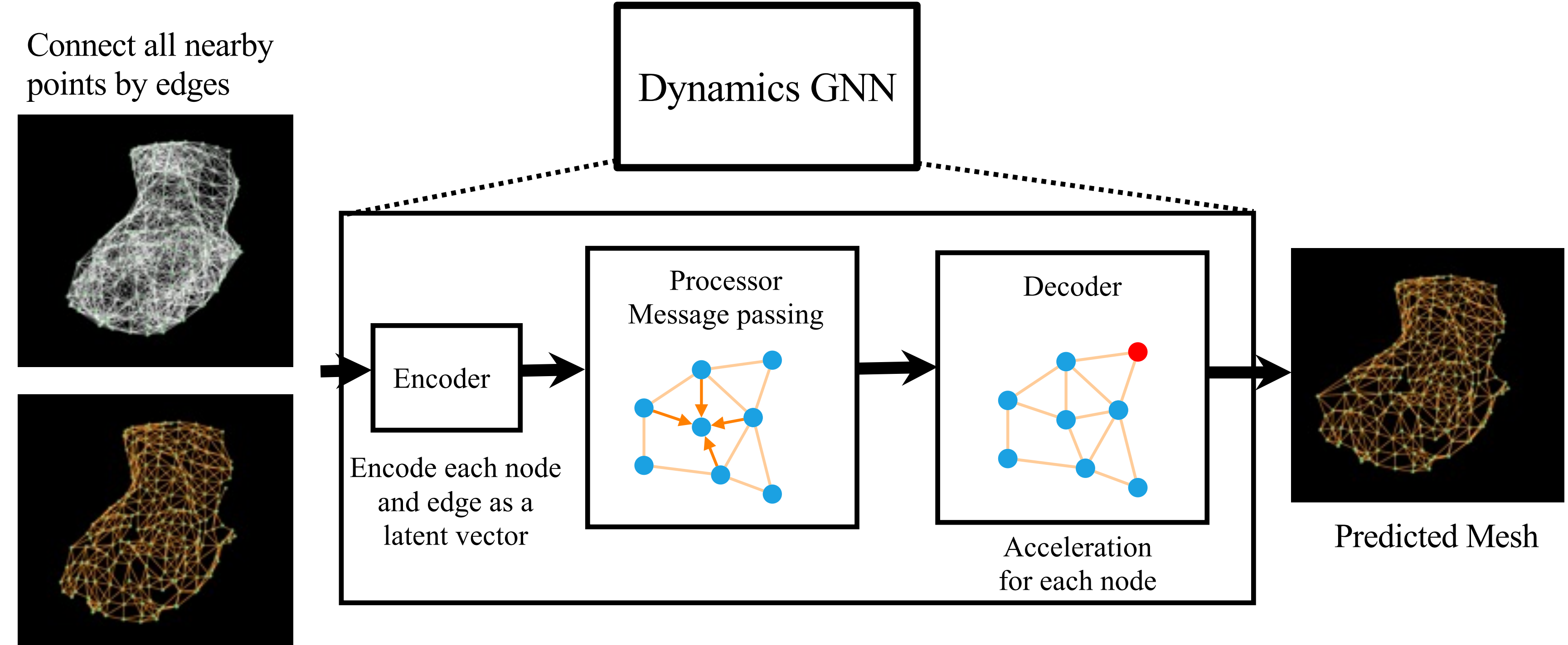
Overview - Learning Visible Connectivity Dynamics (VCD)




Edge Prediction
GNN
if $t = 0$
Connect all nearby points by edges
Encoder
Encode each node and edge as a latent vector
Processor
Message passing
Decoder
Estimated Mesh Edges
• Input feature on the edge: Distance between the two endpoints
• Output on the edge: Binary prediction of whether the edge is mesh edge

Overview - Learning Visible Connectivity Dynamics (VCD)





- Input node feature
  - Historical particle velocity
  - Indicator of whether the particle is picked
- Edge feature
  - Deviation of the endpoints' distance from its rest distance

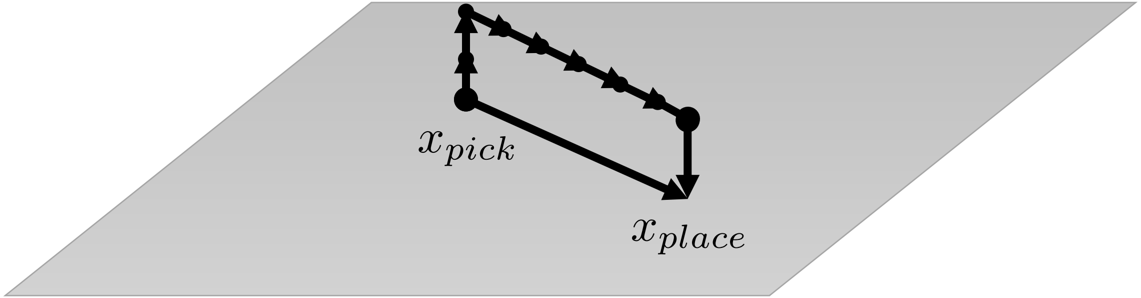
- Output: Acceleration on each node

Subdivide the actions

- Planning in high-level action space (x_{pick}, x_{place})
  - For each high-level action, decompose it into a sequence of waypoints

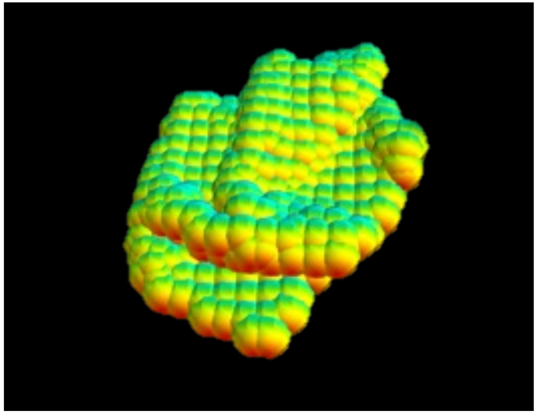
$$\Delta x_1, ..., \Delta x_H, \quad s.t.\ x_{pick} + \sum_{i=1}^{H} \Delta x_i = x_{place}$$

  - Use the dynamics model to predict the position of the cloth at each waypoint



Planning with VCD

- Reward for smoothing task: Covered area in the top down view of a set of spheres centered at each particle



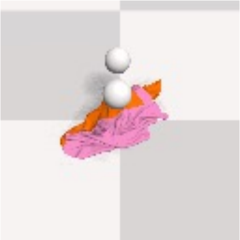
- Planning: “Simple Random Shooting”
  - Sample N pick-and-place actions (Horizon = 1)
  - Simulate the effect of each action using our mesh dynamics model (VCD)
  - Score each action according to the predicted reward
  - Choose the action with the highest predicted reward

This is one of the simplest planning methods

The point of this paper is on dynamics representations, not planning

Training

- Trained in SoftGym (FleX simulator)
- We train on 2000 random pick-and-place actions
  - Baselines are all trained on >100,000 pick-and-place actions
- We use a voxelized point cloud as input to the graph dynamics model
- So we can easily transfer from simulation to the real world!

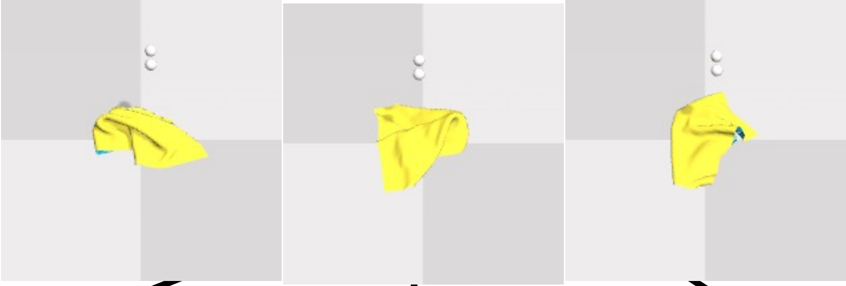


[SoftGym Lin et al. 20]

Voxelized point cloud

One mesh dynamics model (one set of weights)

Training:



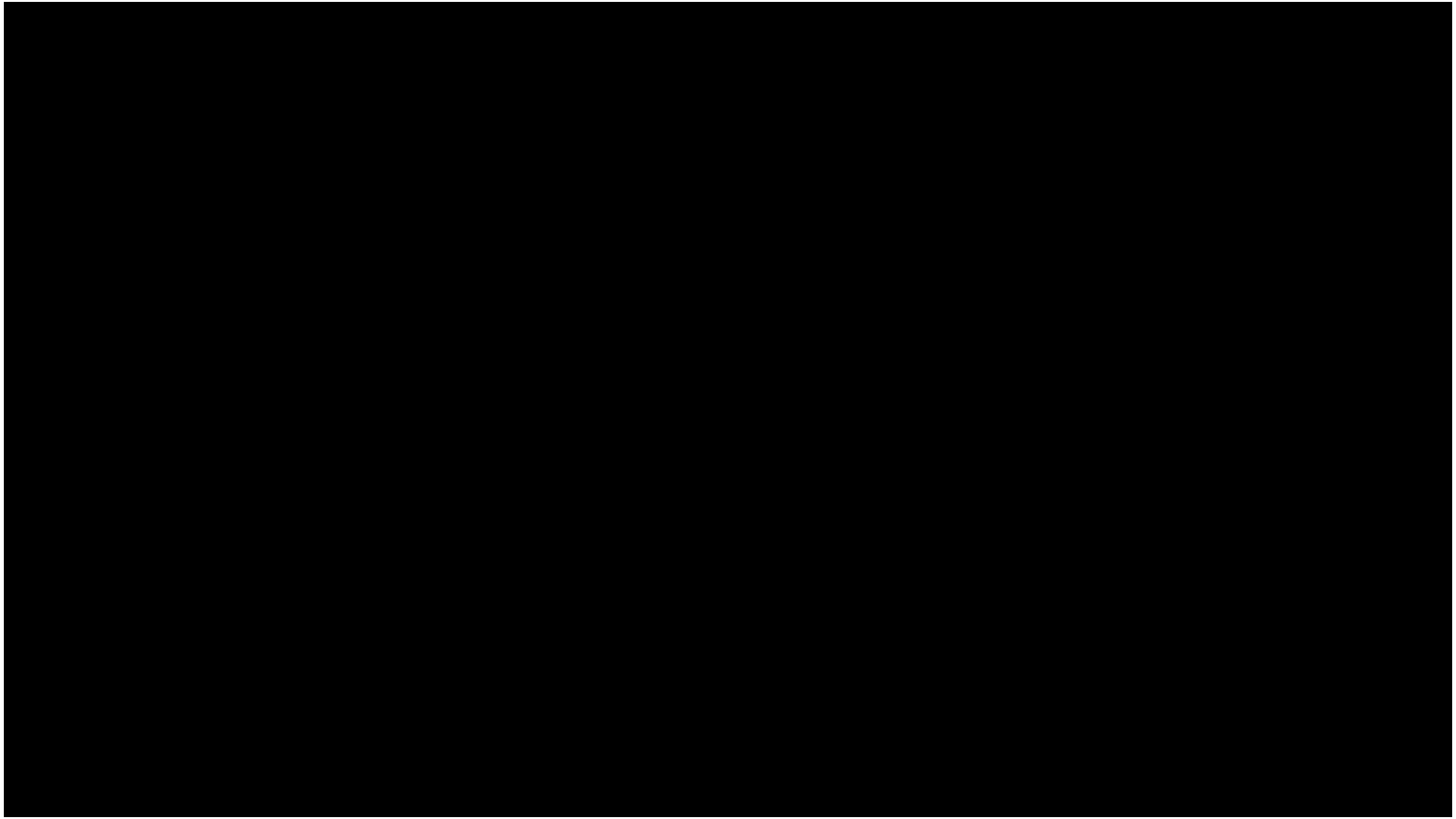
Evaluation:

cotton square

silk square

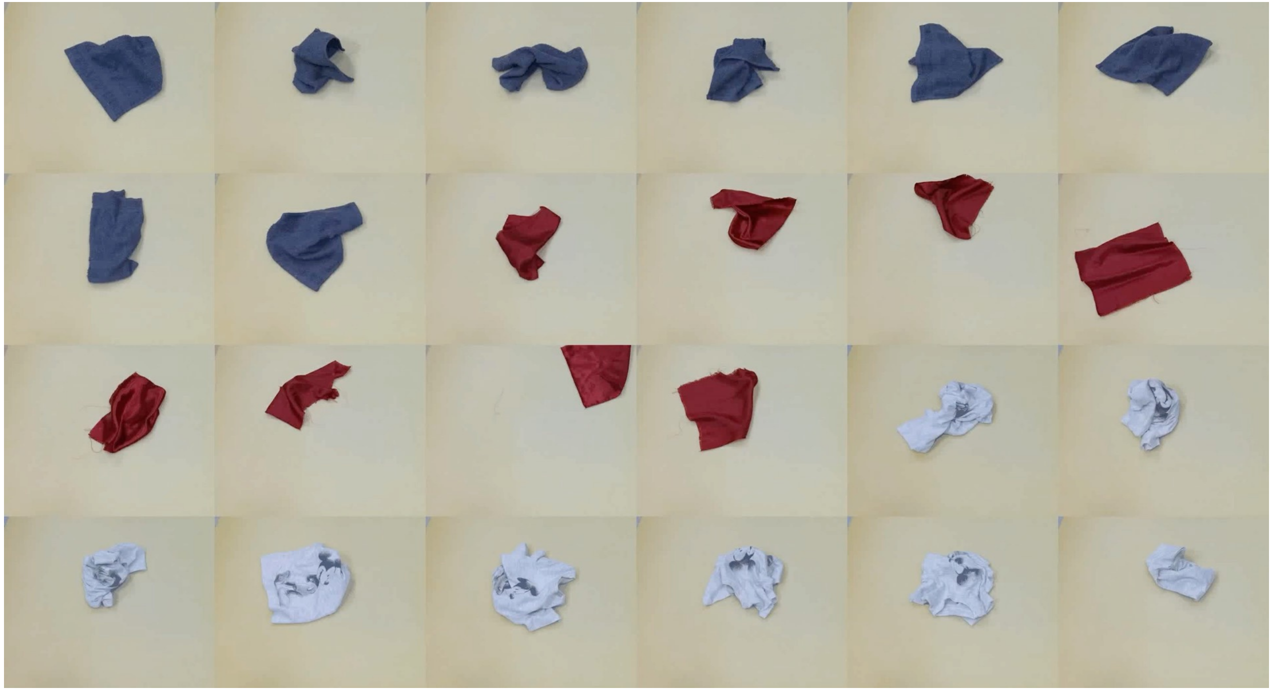
cotton t-shirt

Cloth smoothing



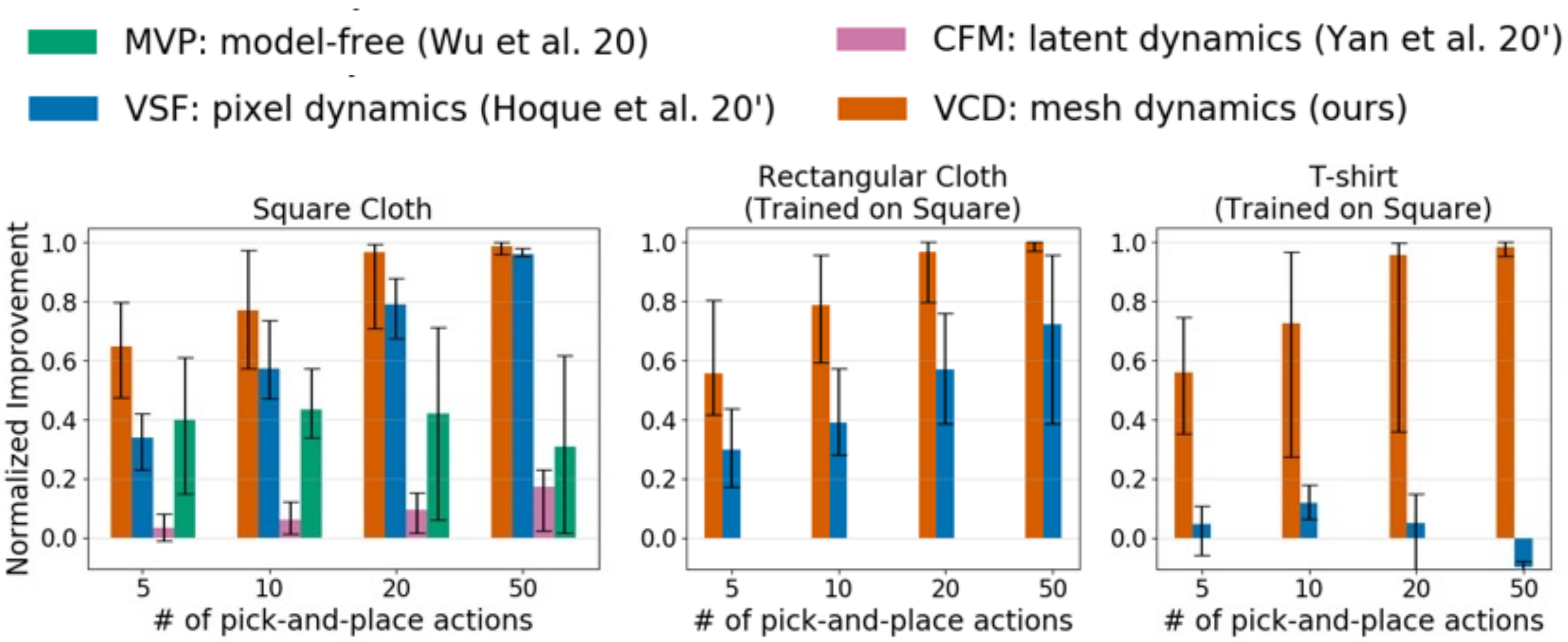
Videos also available at https://sites.google.com/view/vcd-cloth/

24 examples of cloth smoothing

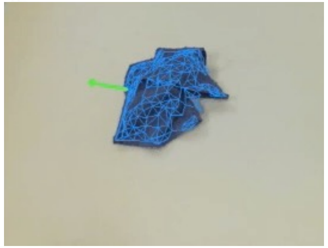


Videos also available at https://sites.google.com/view/vcd-cloth/

VCD outperforms all baselines under different number of actions



- Our method is the only one that generalizes to unseen cloth types
- We train on 2000 random pick-and-place actions
- Baselines are all trained on >100,000 pick-and-place actions (>50x more)



“Do we really need this structure? Can’t we just use a larger neural network and train for longer?”

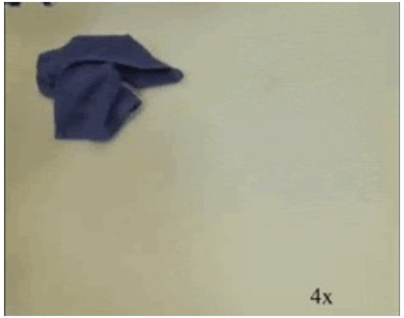
Archimedes: “Give me a lever long enough and a fulcrum on which to place it, and I shall move the world.”

RL-medes: “Give me a large enough neural network and enough computation, and I can memorize any training set!”

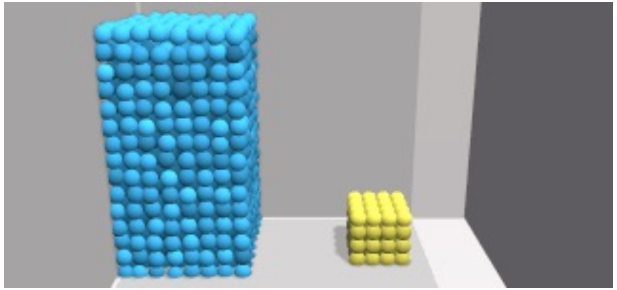
Yes: With a large enough neural network, a large enough dataset, and long enough training time, we can learn any function
(Theorem: Neural networks are universal function approximators)

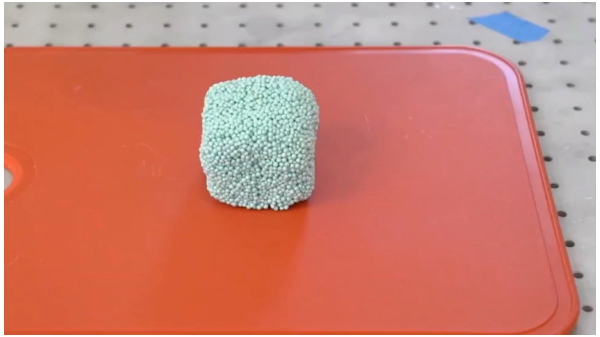
But:

1) We do not always have infinite data (baselines were trained on >50x data)
2) We do not have the capacity to store an infinitely large network
3) We do not have infinite time for training
(especially if we want to quickly teach robots new tasks)
4) This theorem says nothing about generalizing to new objects or new configurations
5) We can add structure to our network without sacrificing generalizability!

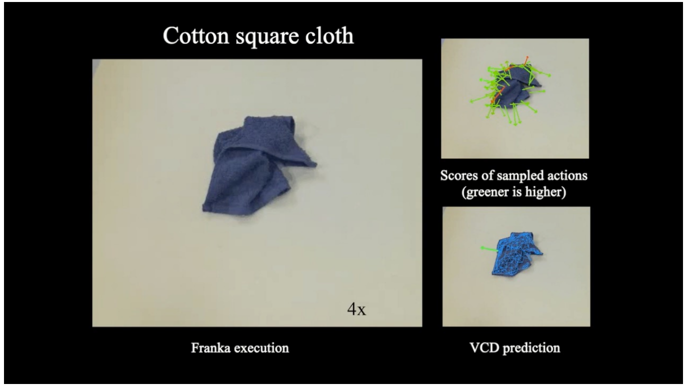


Graph Dynamics can be used for many object types





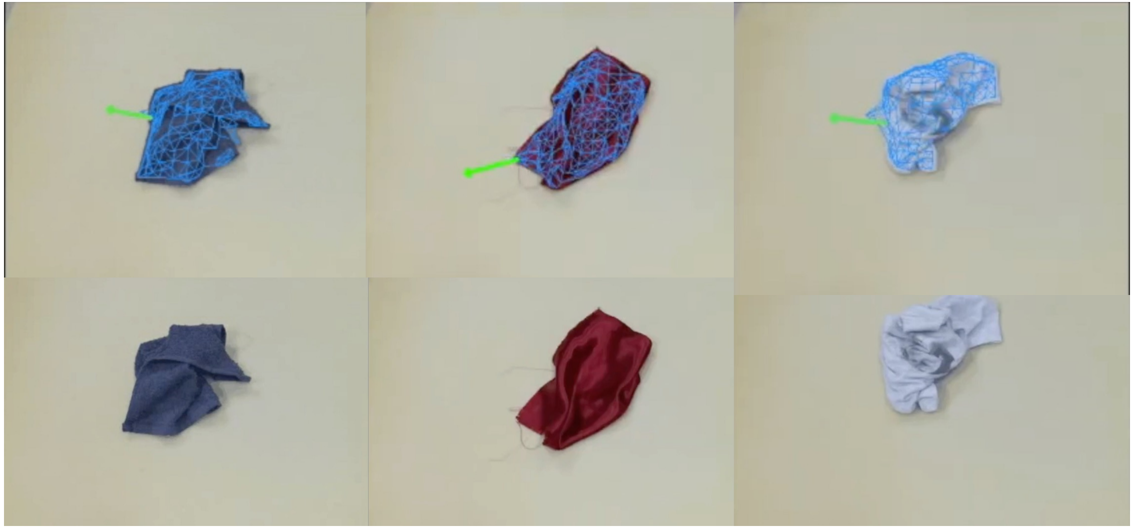
Li, Yunzhu, et al. "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids.” ICLR 2019.



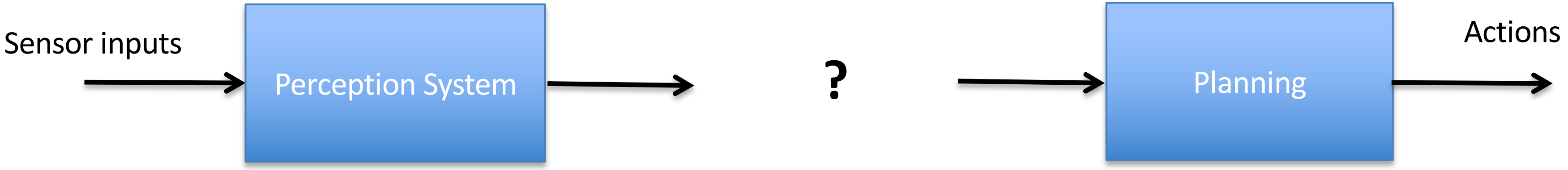
Lin, Xingyu, et al. "Learning visible connectivity dynamics for cloth smoothing." *CoRL* 2021.

Conclusions

- We estimate the connectivity of the visible part of the cloth
- We learn a mesh-based dynamics model based on the estimated visible connectivity
- We perform zero-shot sim2real transfer for cloth smoothing of different shapes and materials

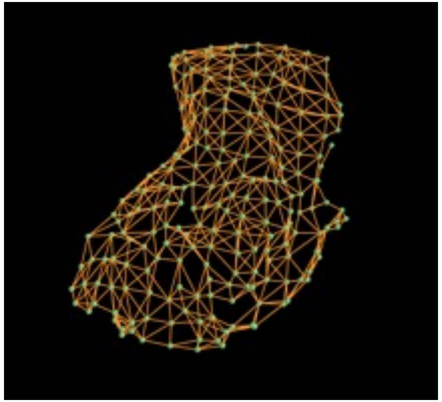


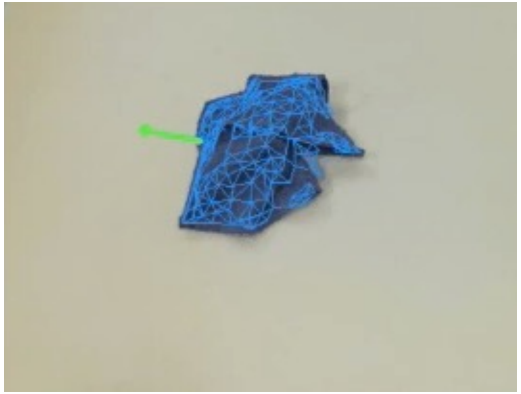
Relational Affordance Learning



Estimating the relationship between **nodes of the cloth mesh**

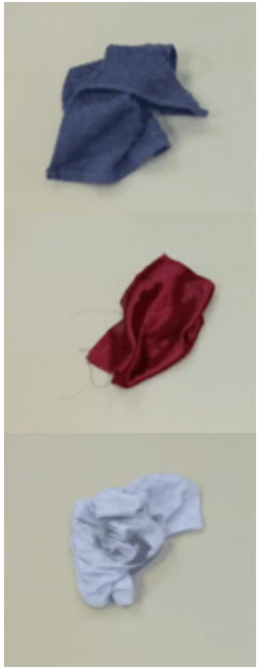
Planning over learned mesh dynamics





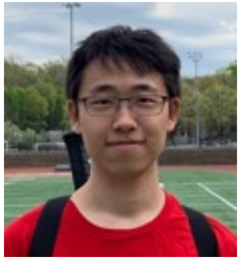
How can robots learn a dynamics model for non-rigid manipulation tasks?

(CoRL 2021, RSS 2022)



Cloth Dynamics model

robot actions



Xingyu Lin

Yufei Wang

How can robots learn dual arm manipulation policies for non-rigid objects?

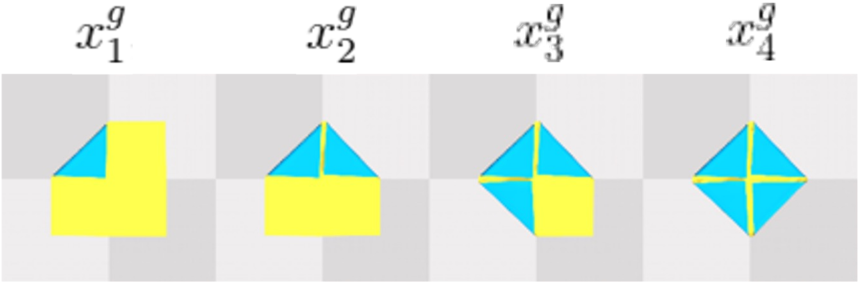
Flow-based Policy for Bimanual Goal-conditioned Cloth Flattening (CoRL 2021)

Thomas Weng

Sujay Bajrachaya

Problem formulation

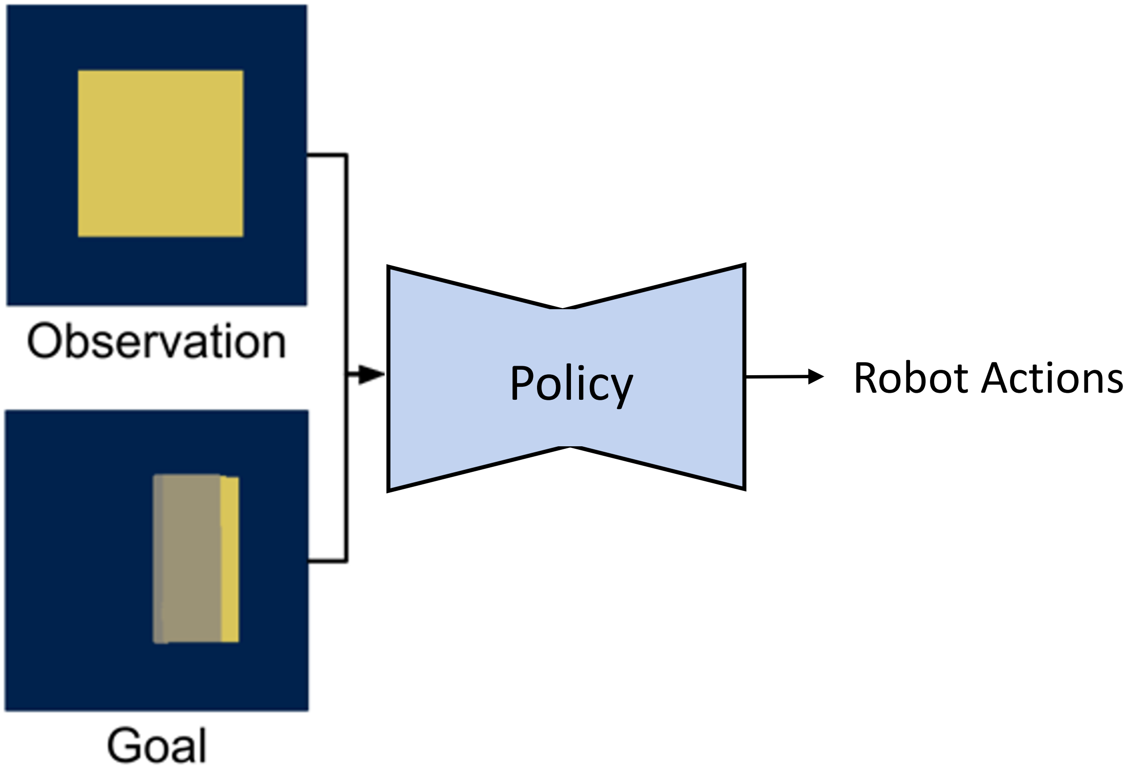
- Folding task defined by series of subgoals: $\mathcal{G} : \{x_1^g, x_2^g, \ldots, x_N^g\}$



- Goal conditioned policy takes current observation and subgoal

$$a_t = \pi(x_t, x_i^g)$$

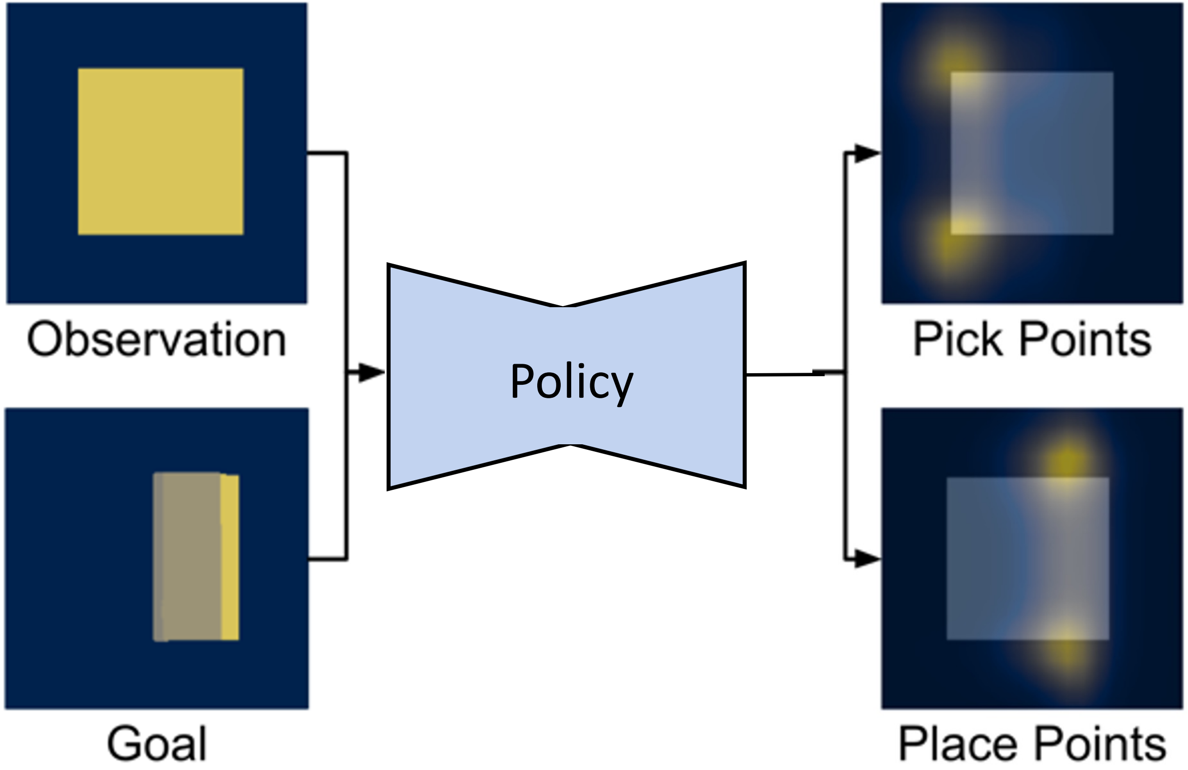
Previous Work:



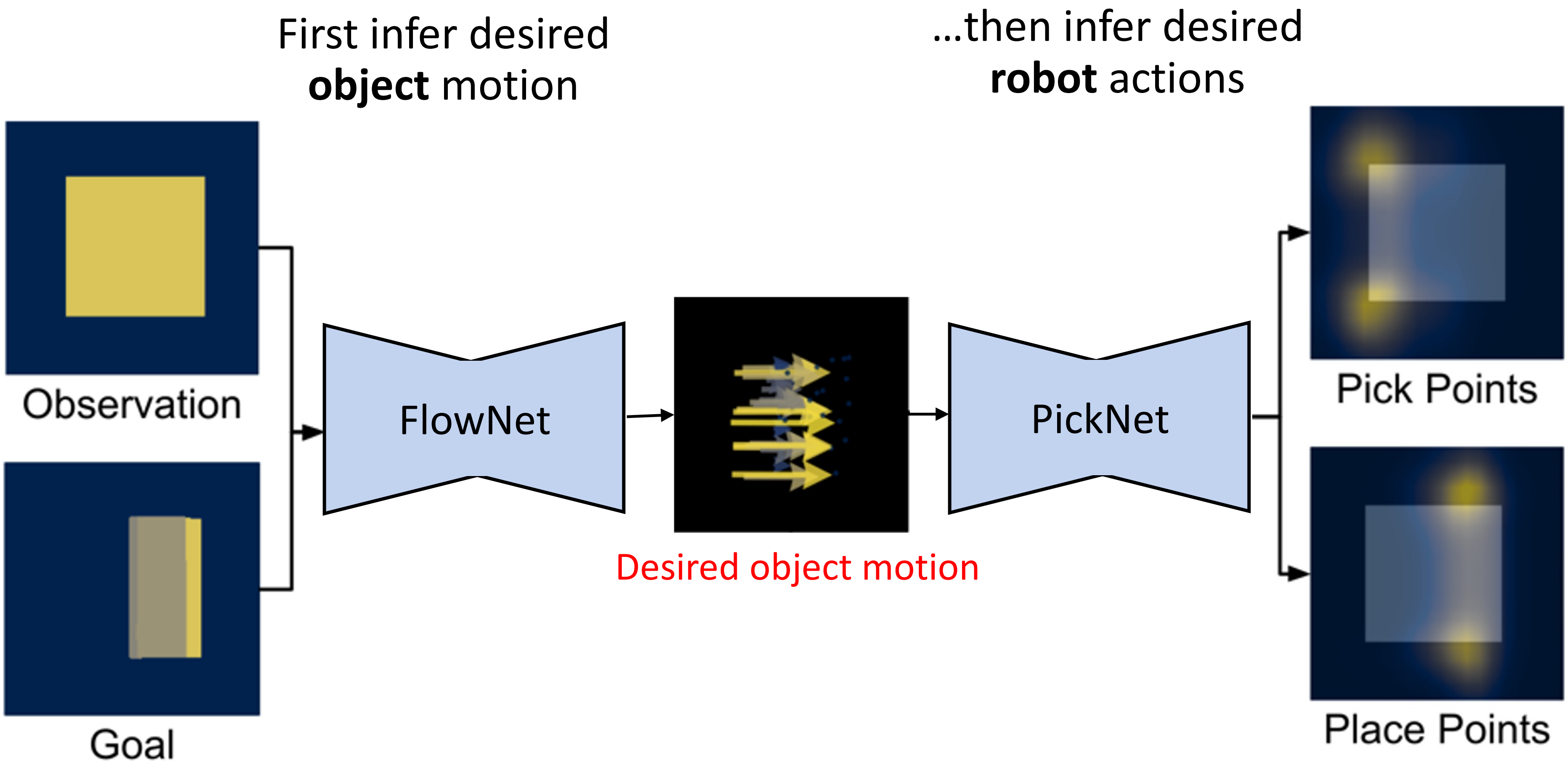
Previous Work:

Problem – required to jointly reason about:

- Relationship between the observation and the goal
- Where the robot needs to grasp the cloth to achieve that goal

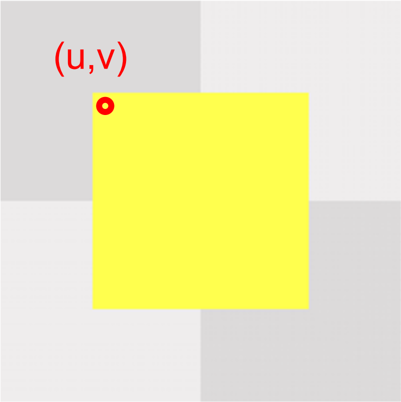


Our Main Insight:



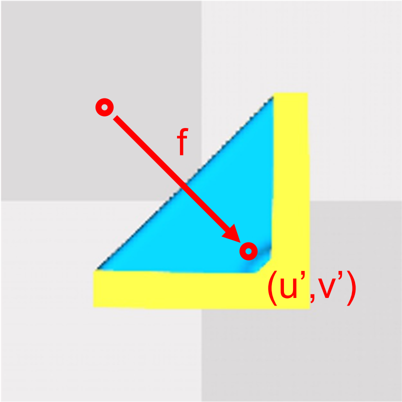
Estimating Correspondence between Observation and Goal ("Flow")

observation x_t

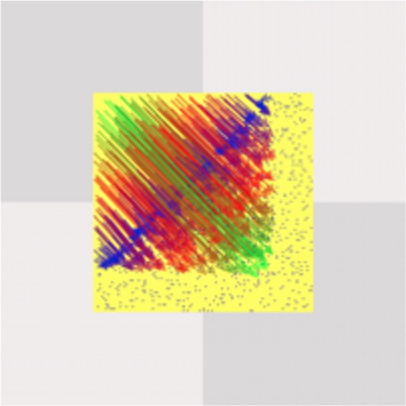


For each point in the observation

goal x^g

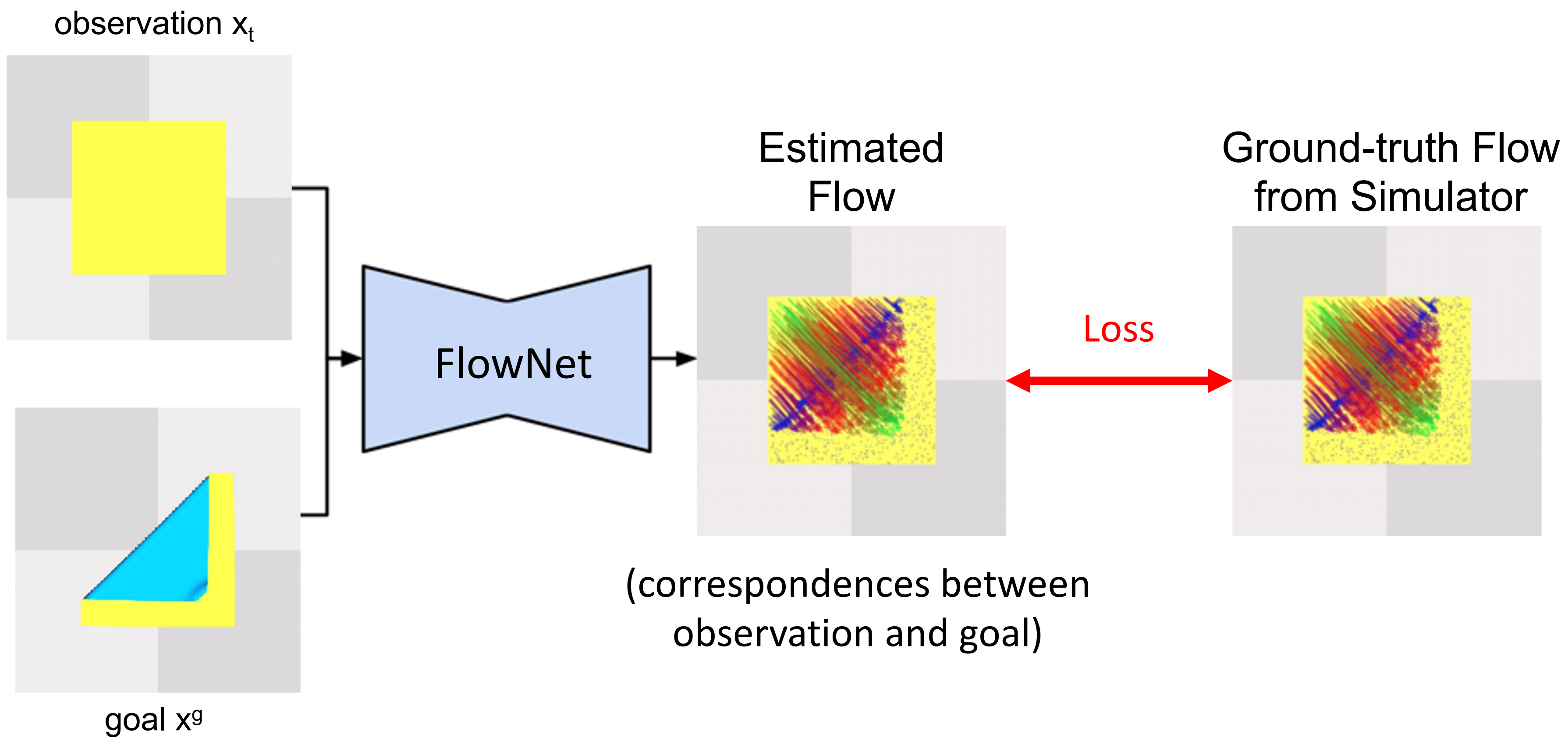


… we want to predict where did that point move to in the goal



Use simulator to find ground-truth correspondences

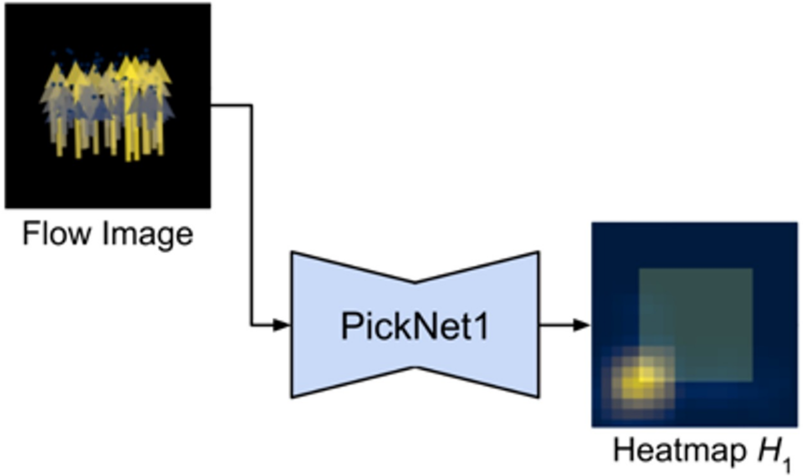
Estimating Correspondence between Observation and Goal



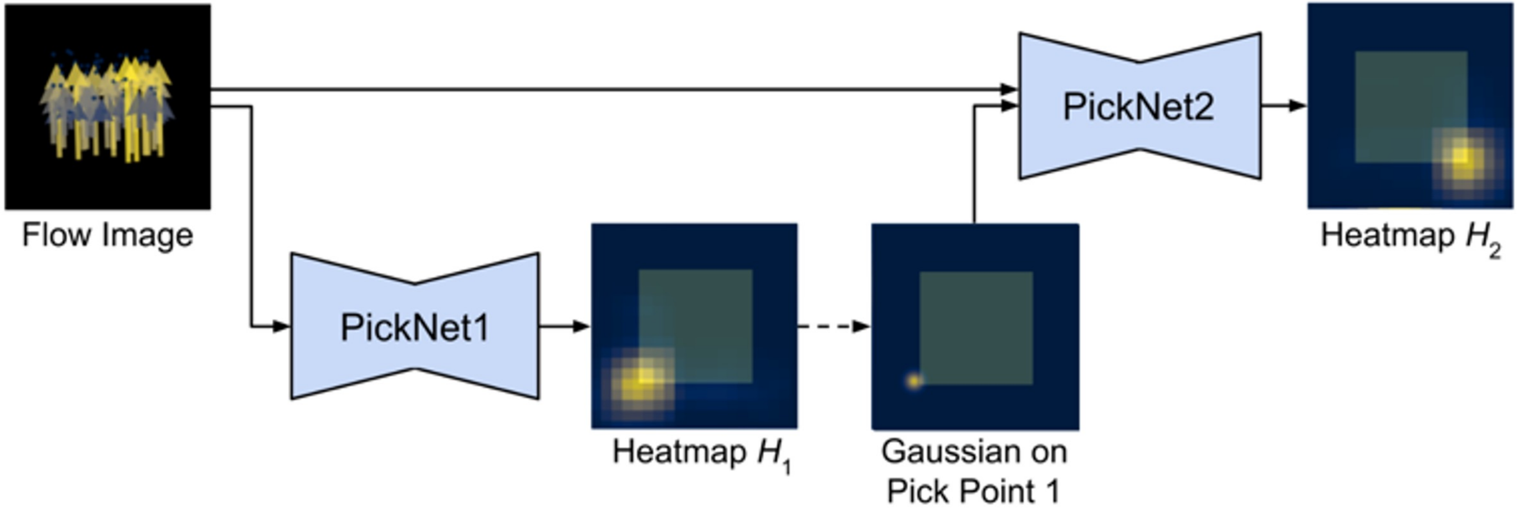
Learning to Predict Pick Points

Flow Image

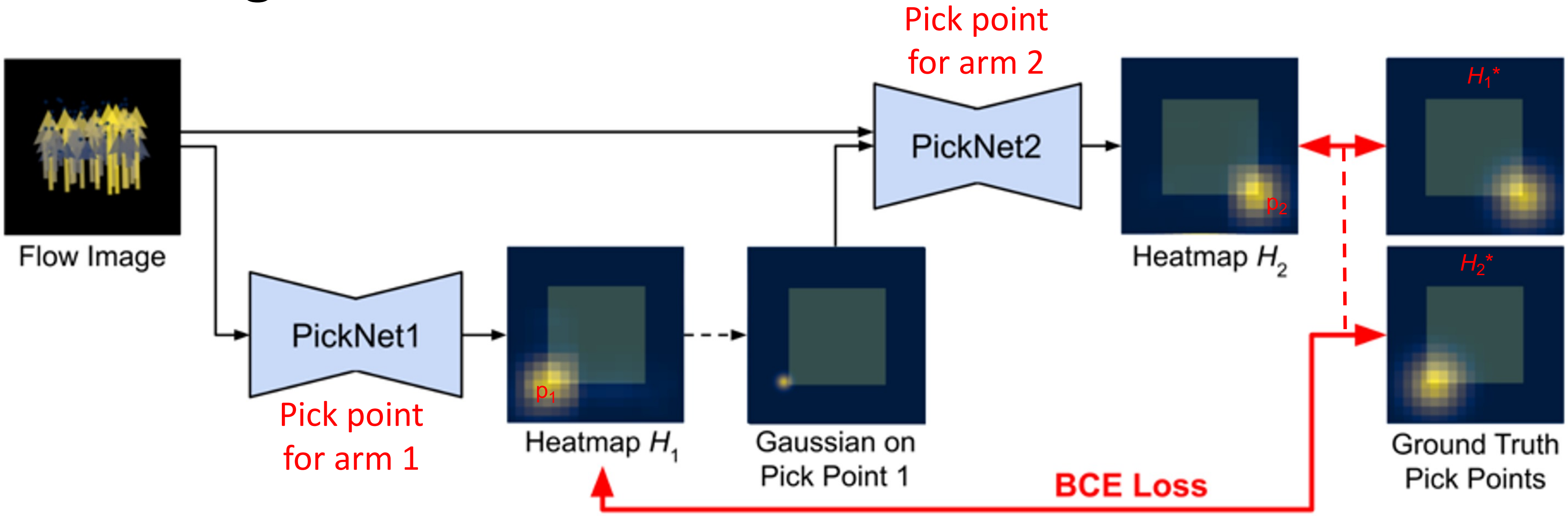
Learning to Predict Pick Points



Learning to Predict Pick Points



Learning to Predict Pick Points

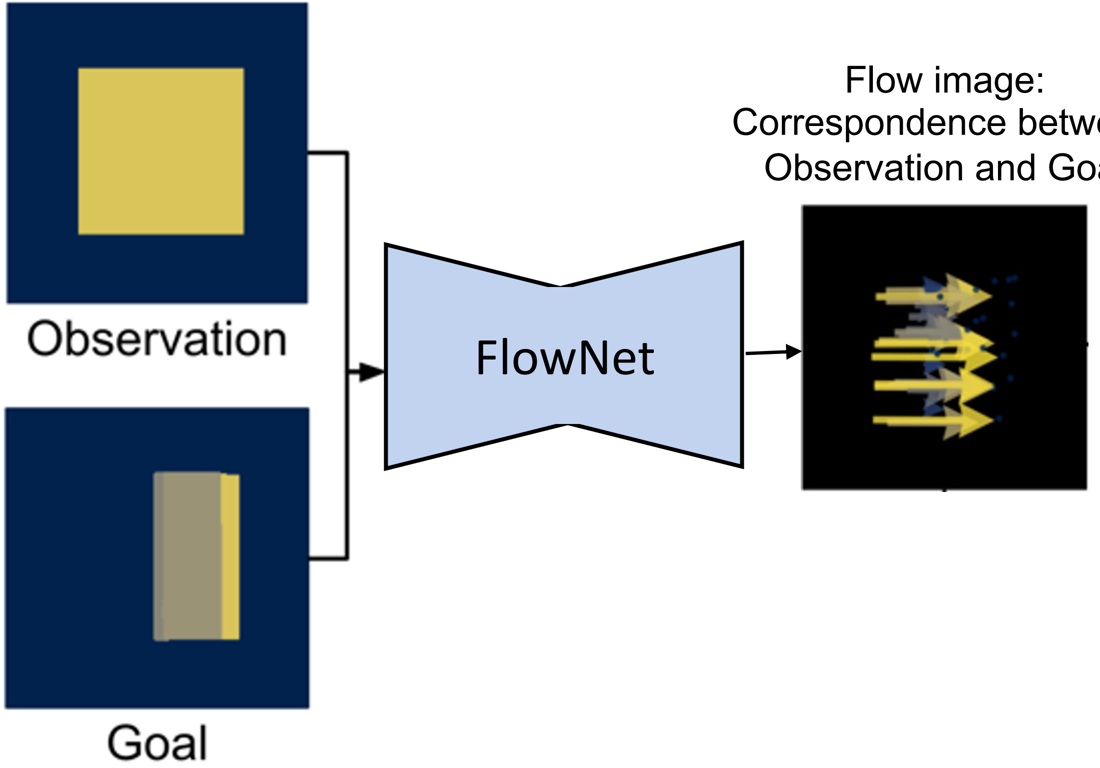


FabricFlowNet

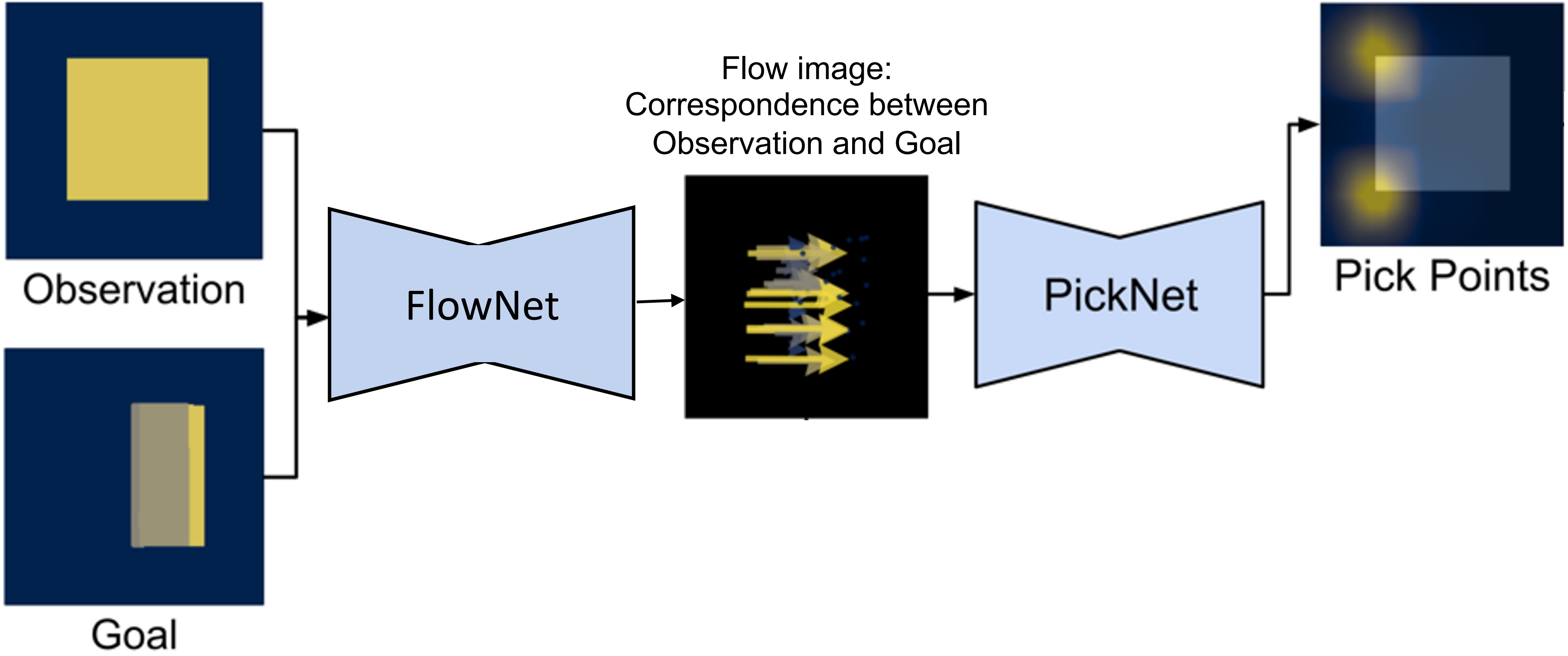
Observation

Goal

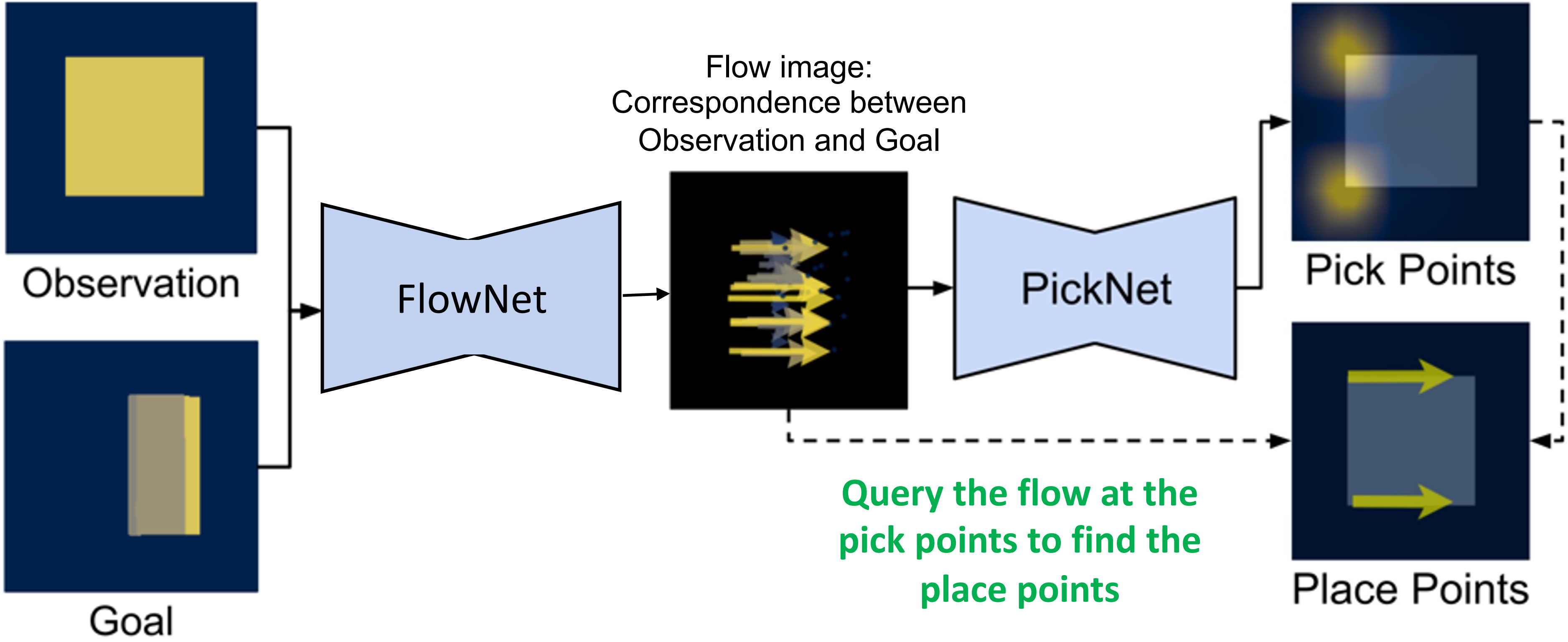
FabricFlowNet



FabricFlowNet



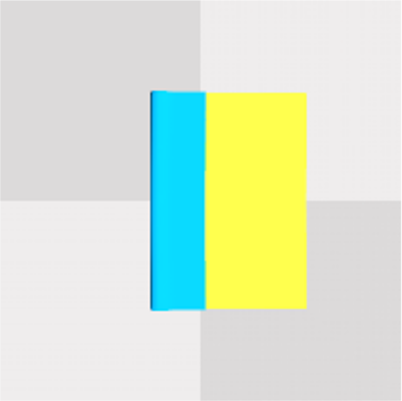
FabricFlowNet



Estimate the Place Points from Flow

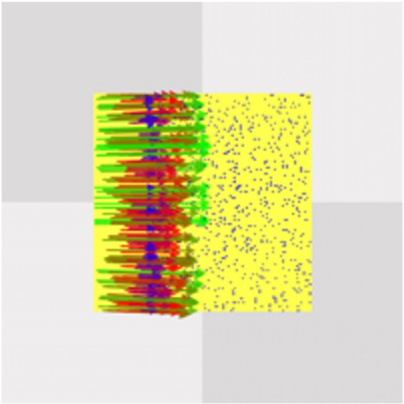
observation x_t

goal x^g_i

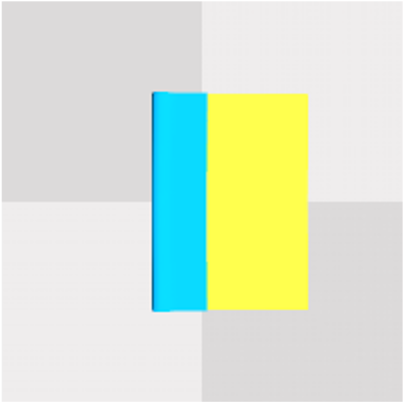


Estimate the Place Points from Flow

observation x_t
+ estimated flow

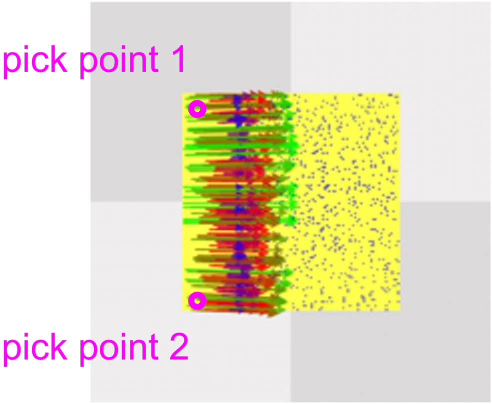


goal x^g

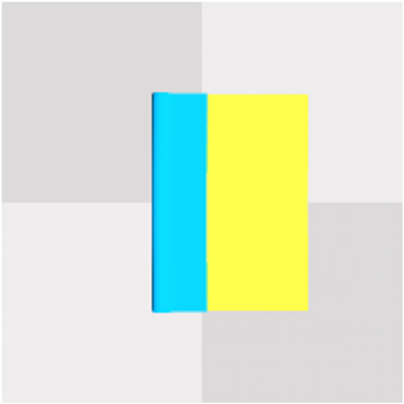


Estimate the Place Points from Flow

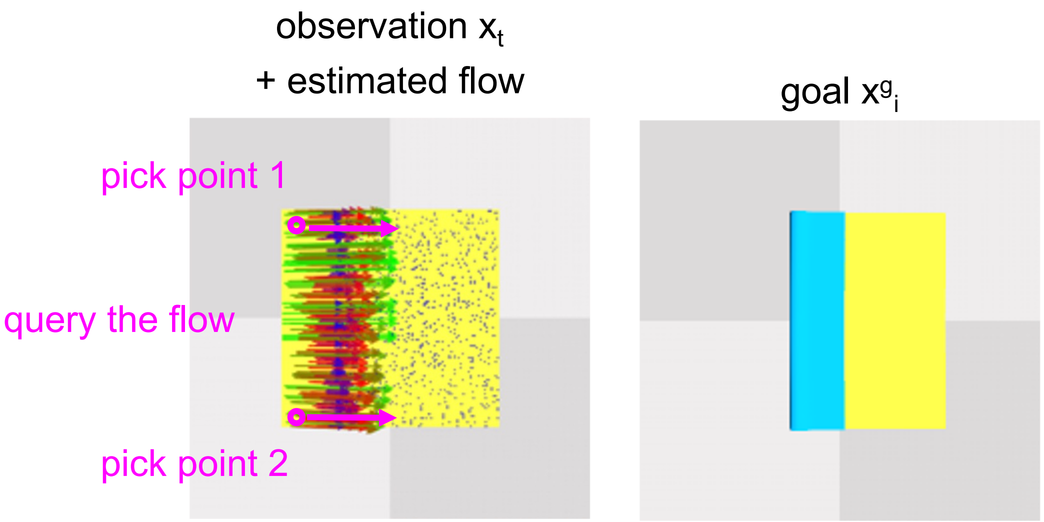
observation x_t
+ estimated flow



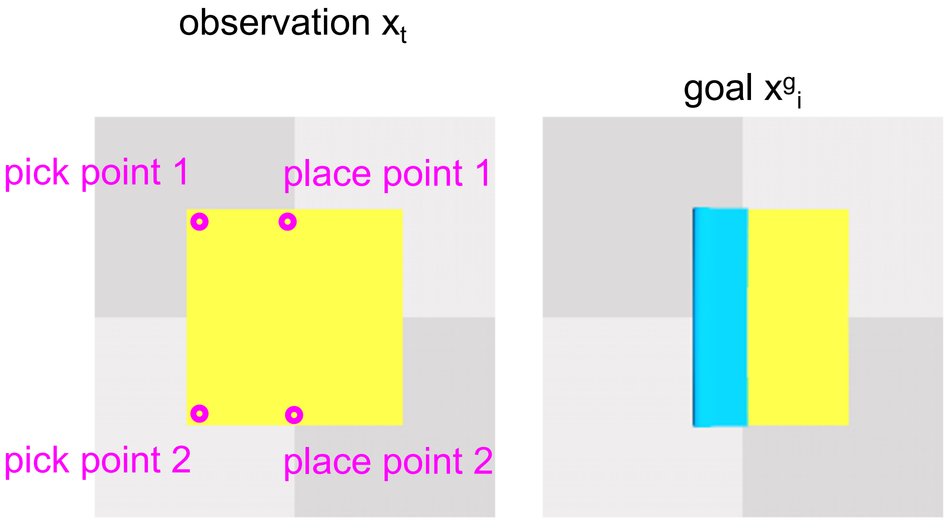
goal x^g



Estimate the Place Points from Flow

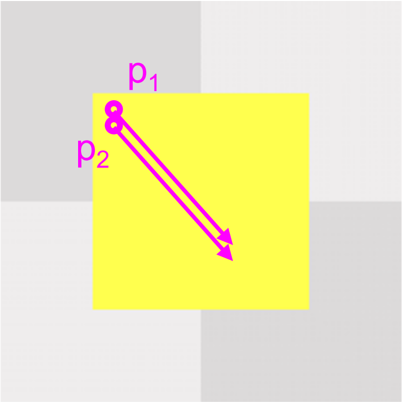


Estimate the Place Points from Flow



Switching between single and dual-arm actions

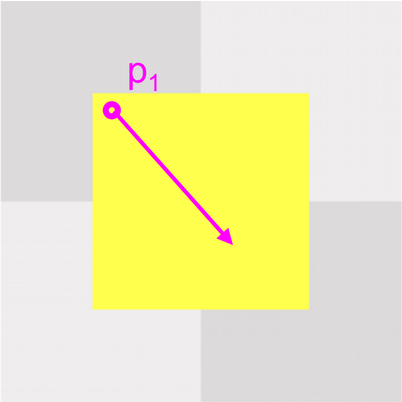
Two-arm prediction



Pick points
are within a
threshold

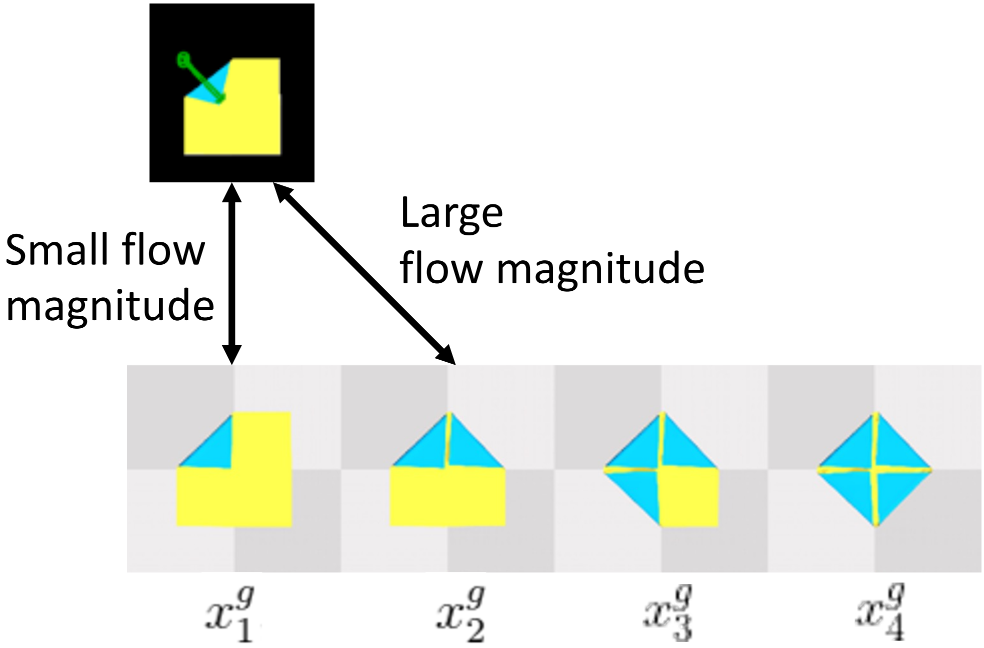
goal

New single-arm prediction



Merge the two pick
points to a single action

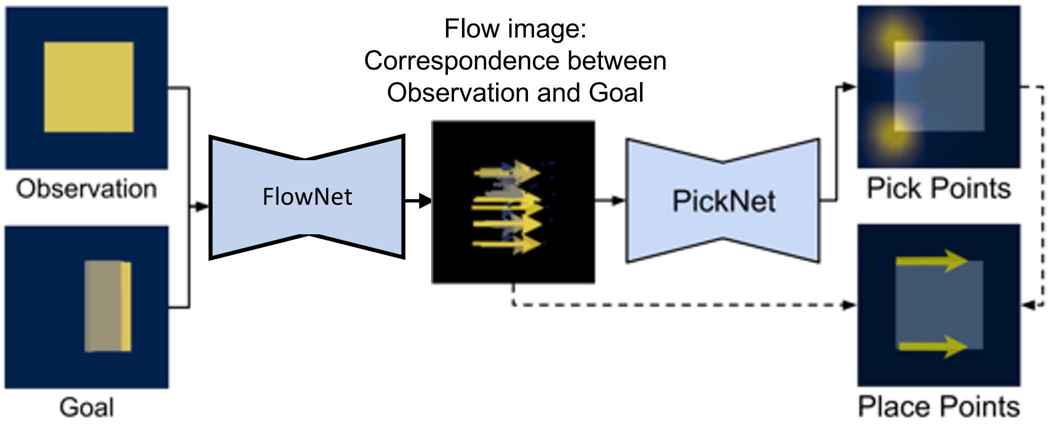
When to progress to the next subgoal?



Progress to the next subgoal when the average flow magnitude to the current subgoal is less than a threshold

Benefits of Flow

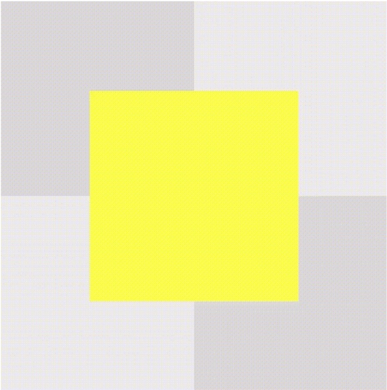
1. Explicit reasoning about relationship between observation and goal
2. Determine place points by querying the flow at the pick points
3. Use flow magnitude to decide when to progress to the next subgoal



Training

- Train in SoftGym and transfer to real world

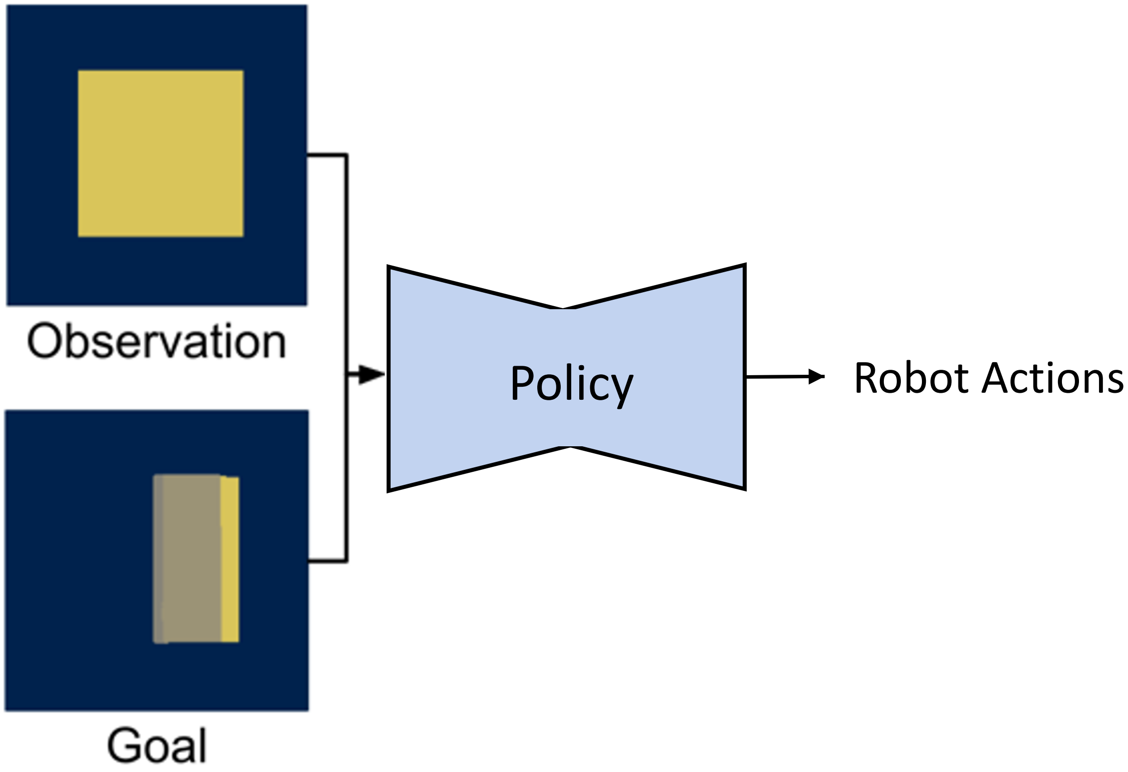
- Uses depth images as input so sim2real transfer is easy



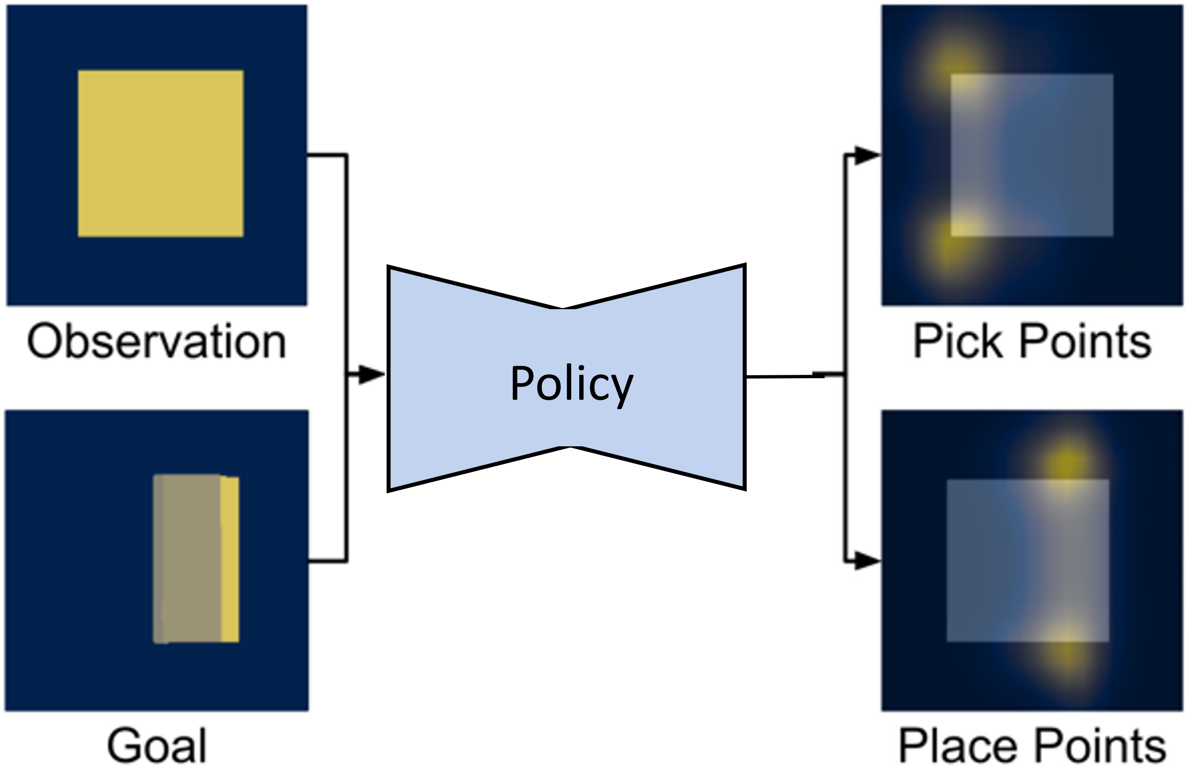
Collect 20K random pick and place actions

Depth image

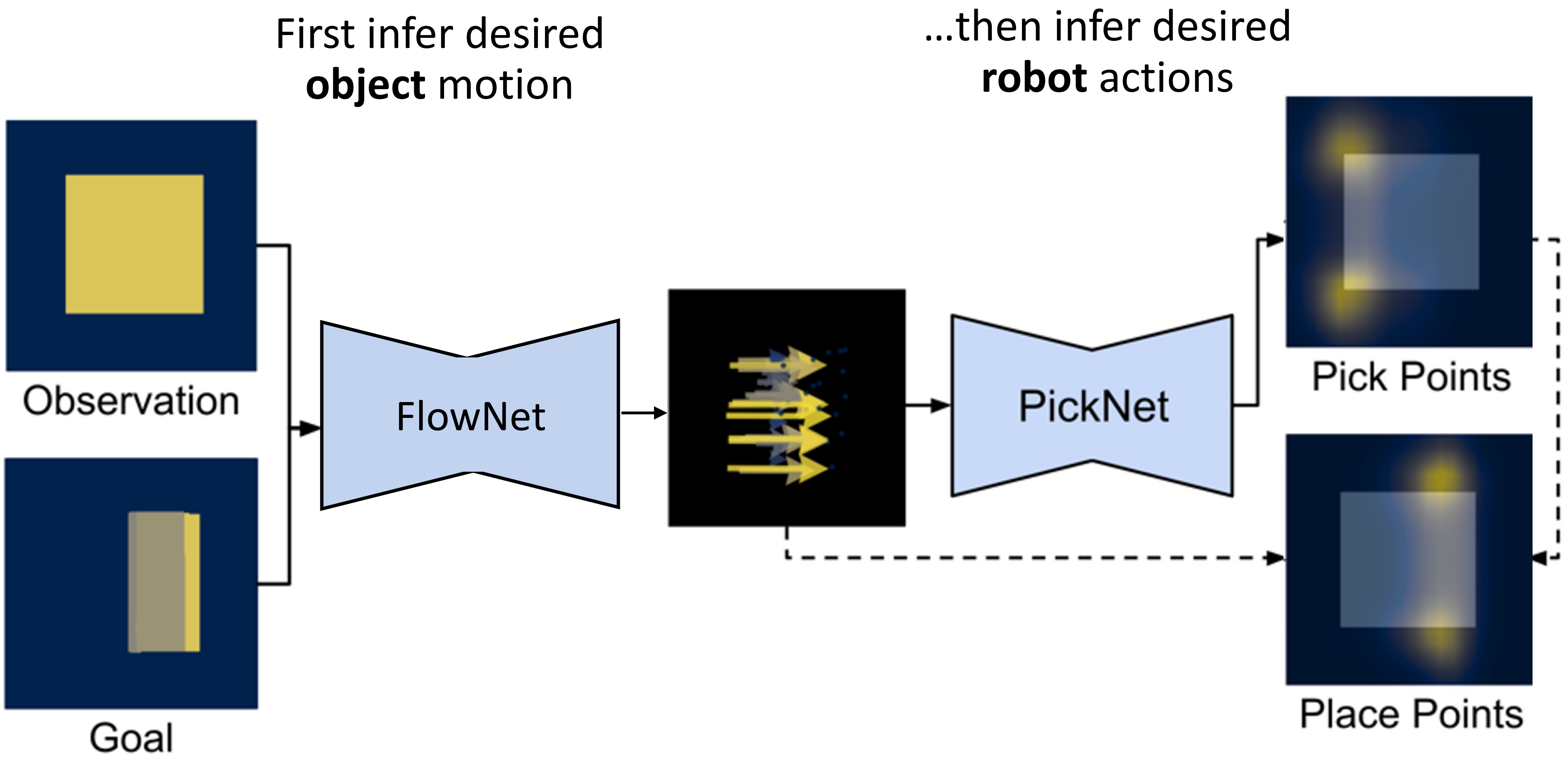
Previous Work:



Previous Work:



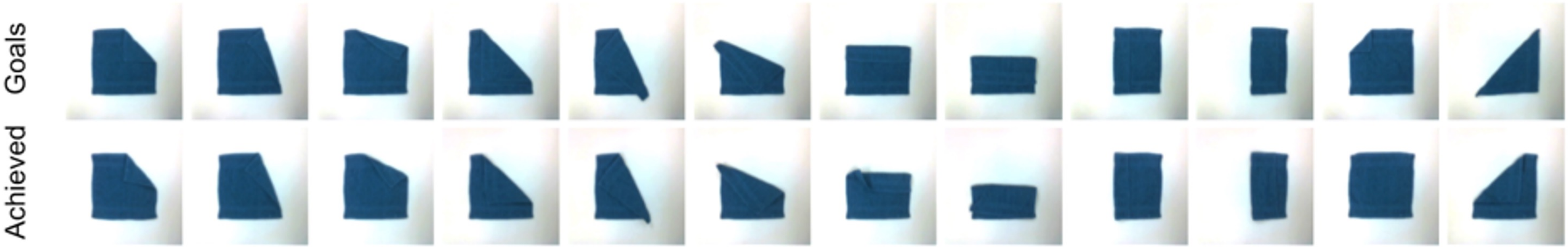
Our Main Idea:

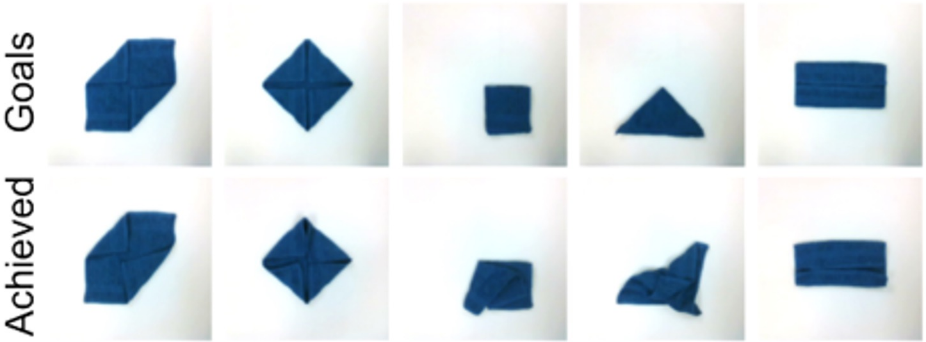


Real folding videos

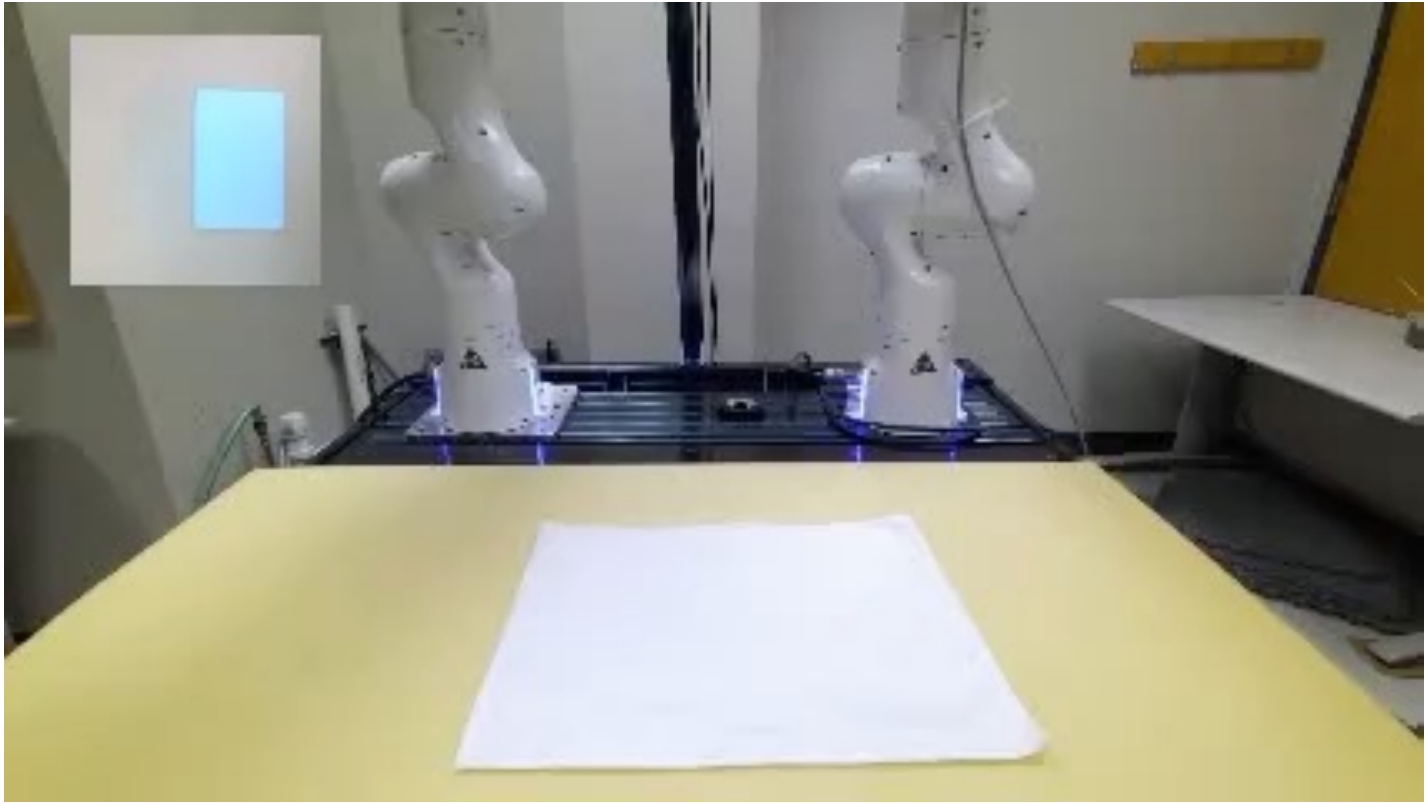
4x speed

Real World Results



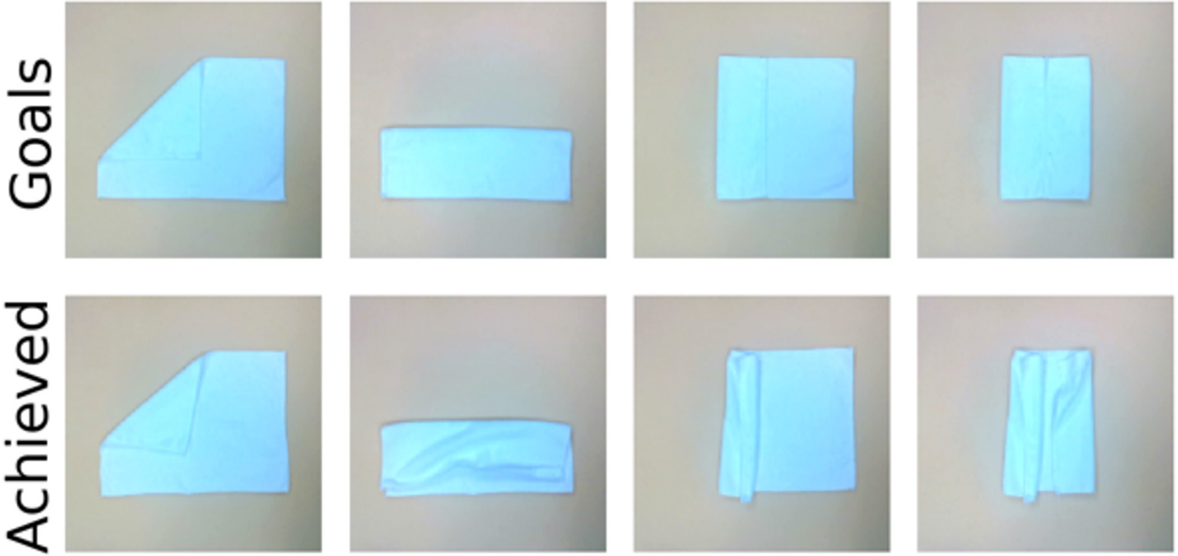


Zero shot generalization to white rectangular cloth



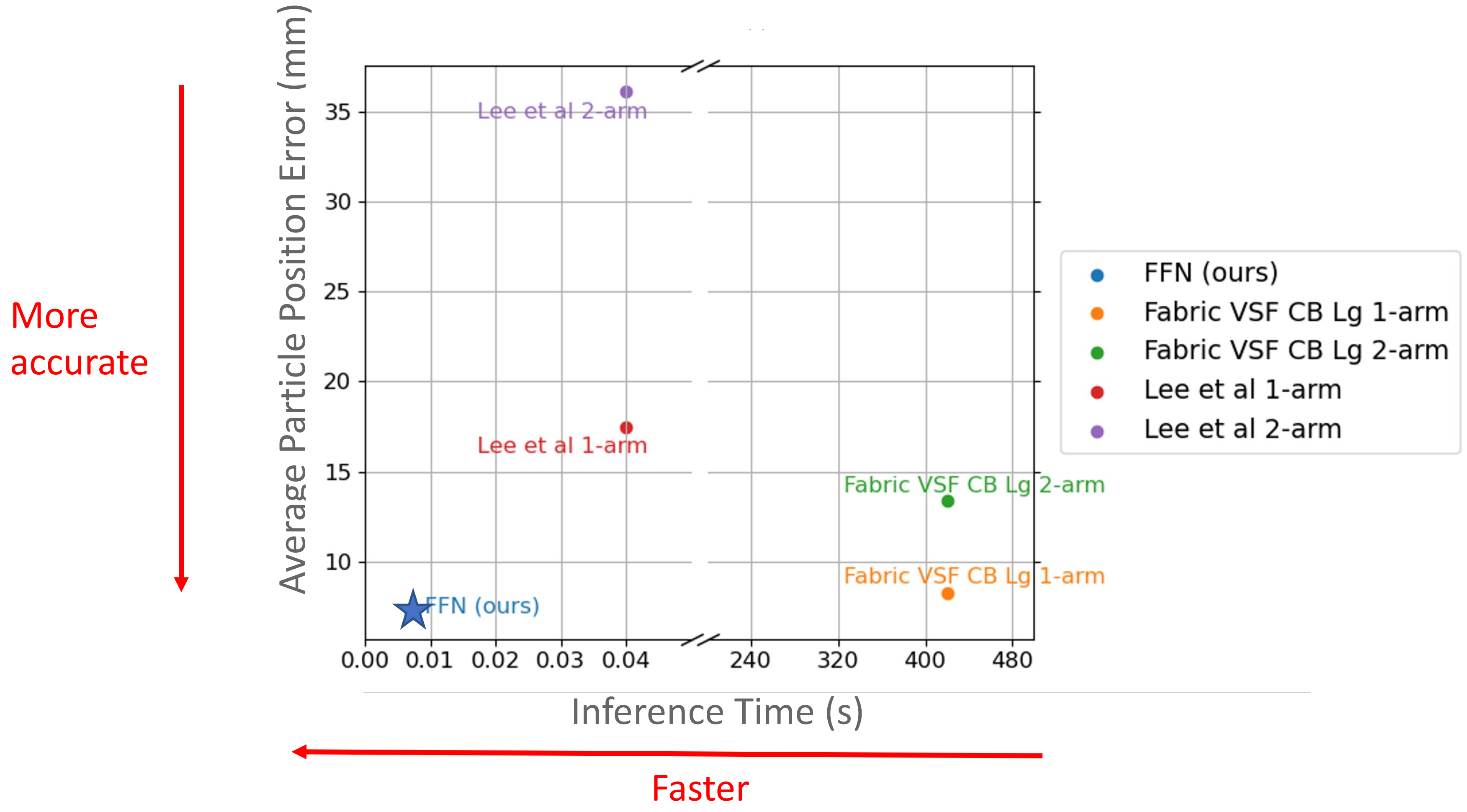
4x speed

Real World Generalization to Rectangle (trained on square)

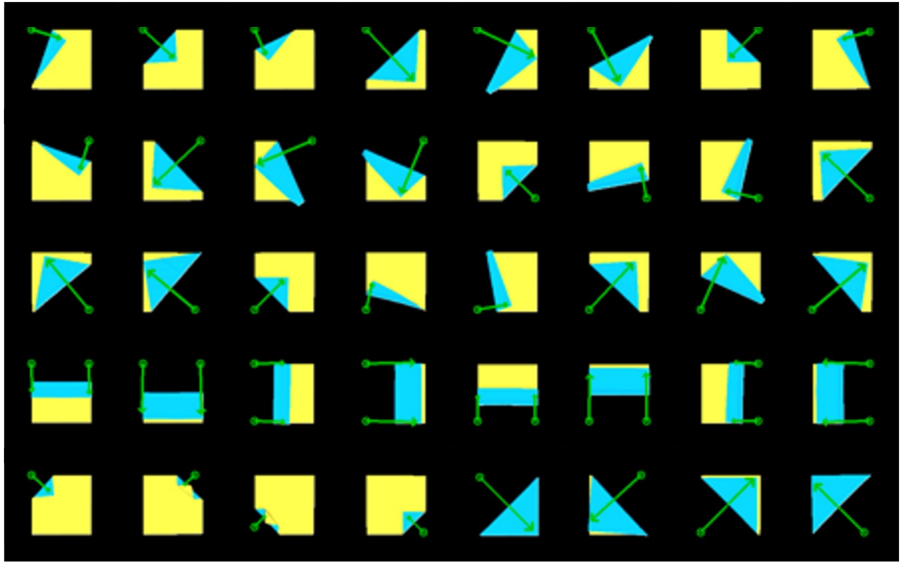


Real World Generalization to T-shirt (trained on square)

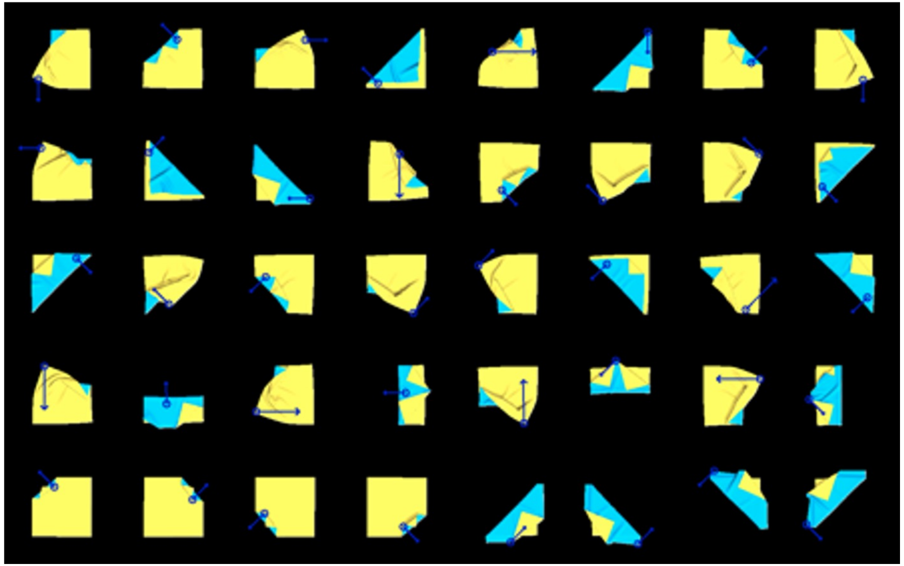
Folding results



Simulation results

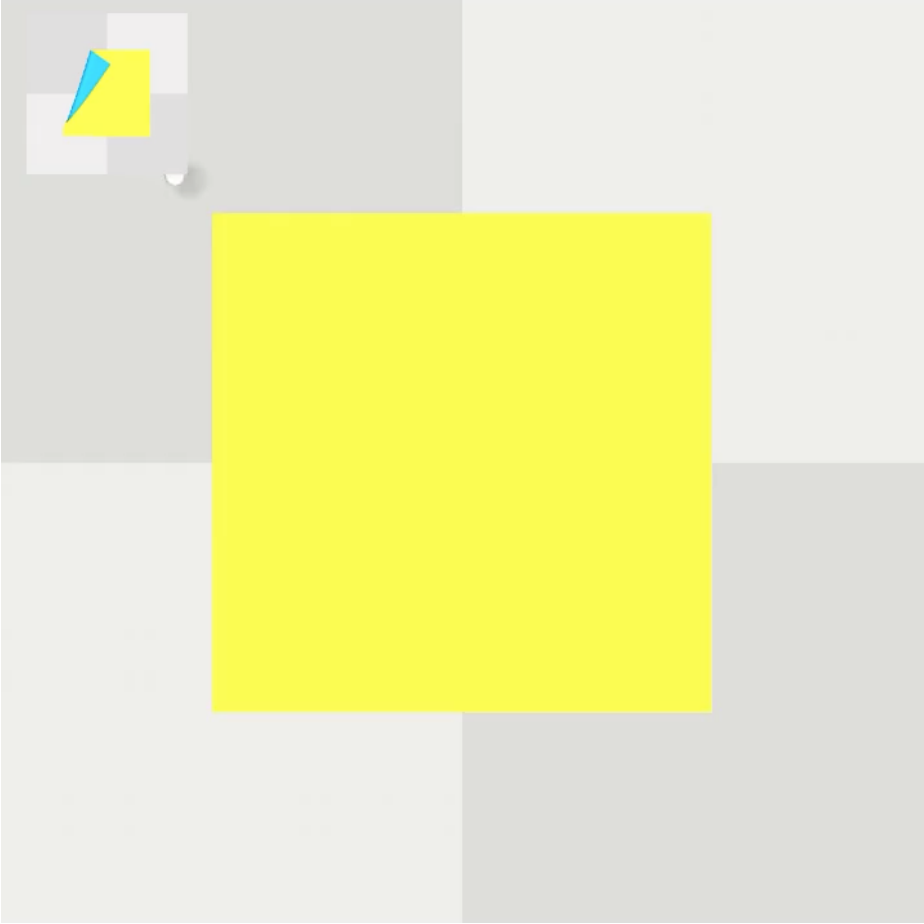


FFN (ours)



Lee et al.

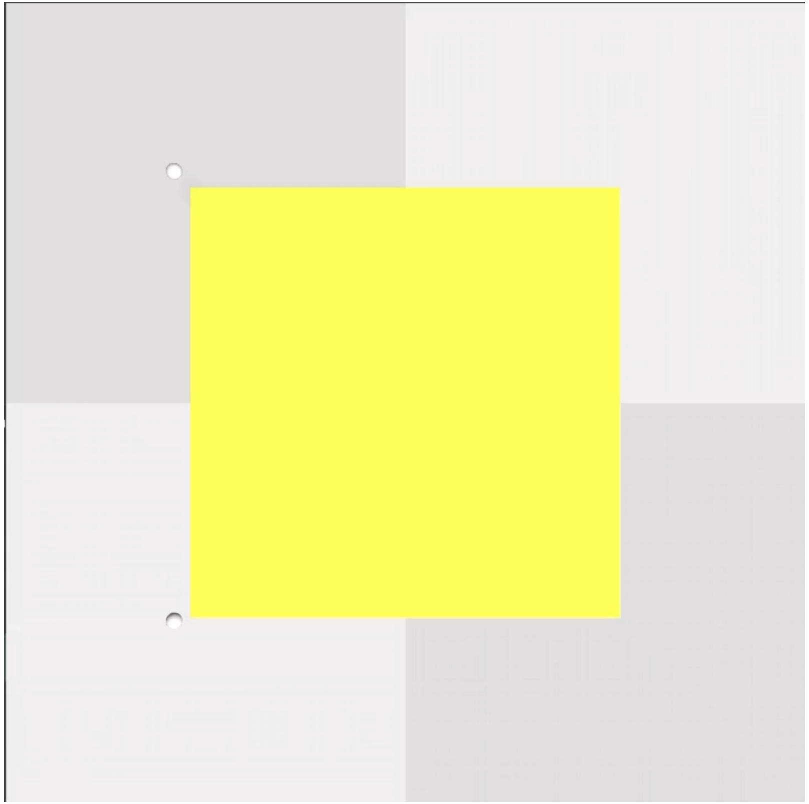
Simulation results



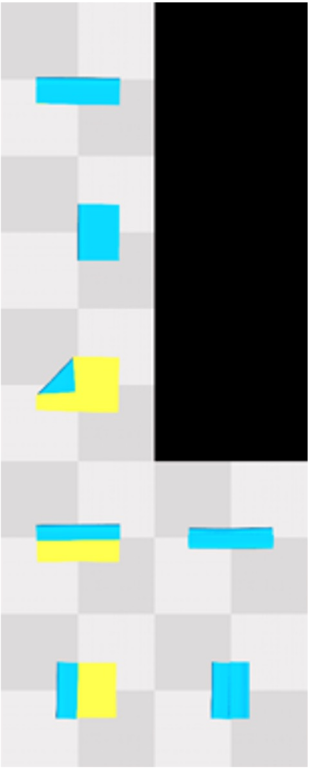
2x Speed

Iterative Corrective Actions

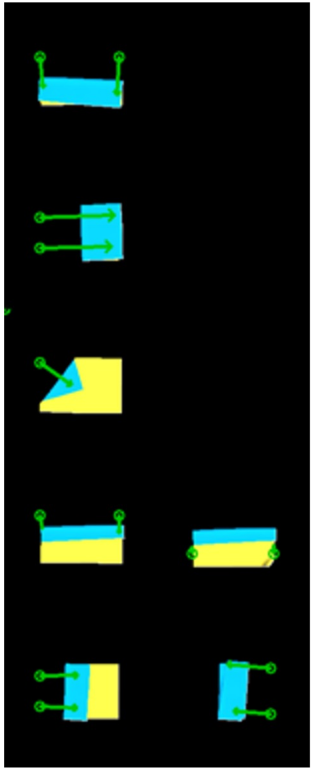
- Allow multiple actions per subgoal to correct errors
- Use flow magnitude to decide when to move to the next subgoal



Generalization to Rectangle cloth



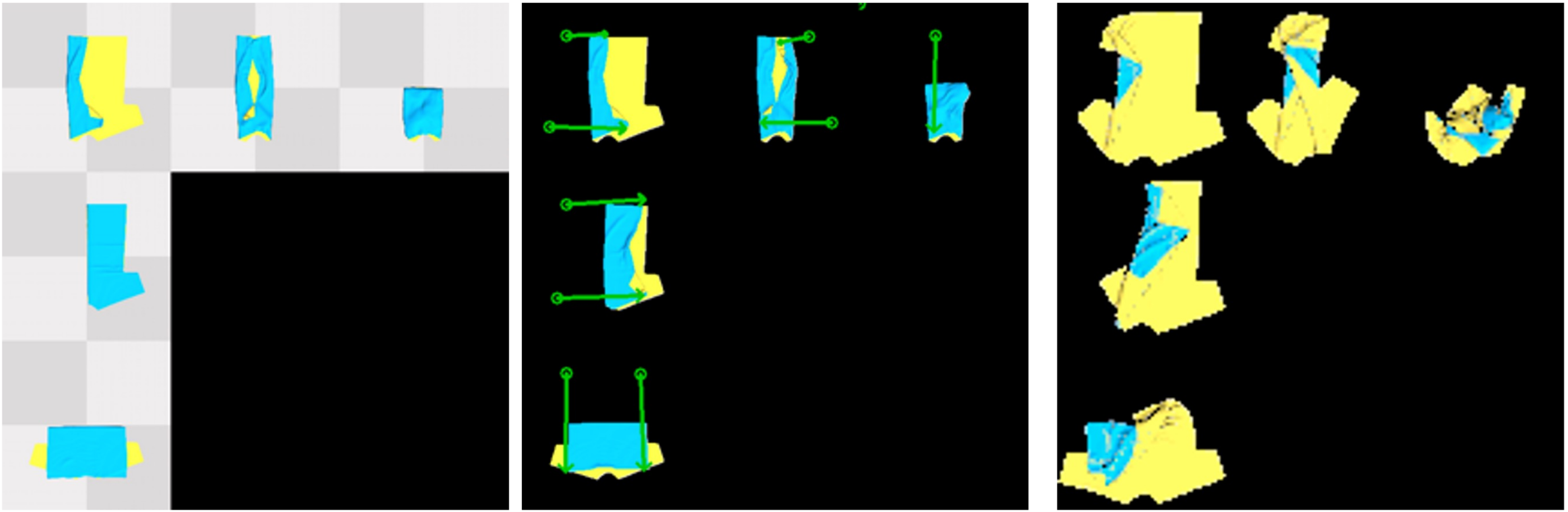
Goals



FFN (Ours)

Fabric-VSF

Generalization to T-shirt



Goals FFN (Ours) Fabric-VSF

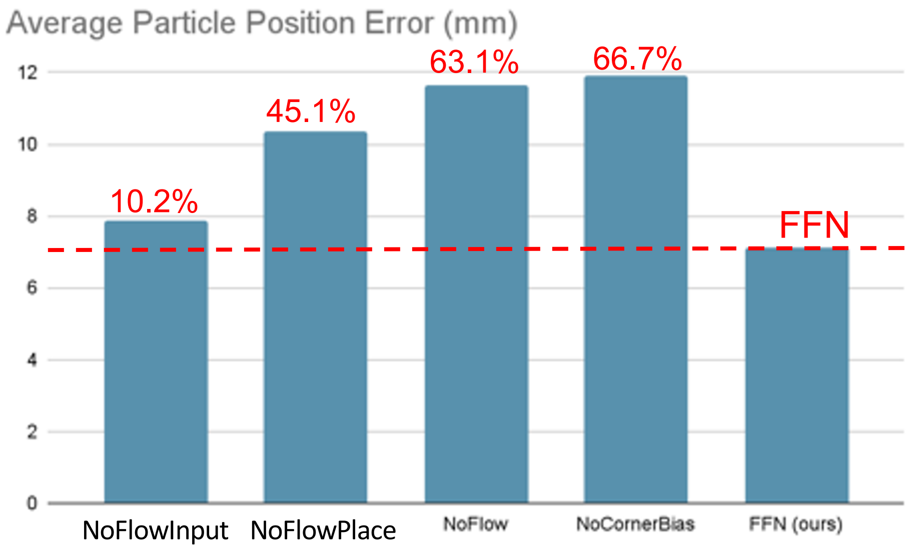
Ablations

NoFlowInput: use depth input instead of flow

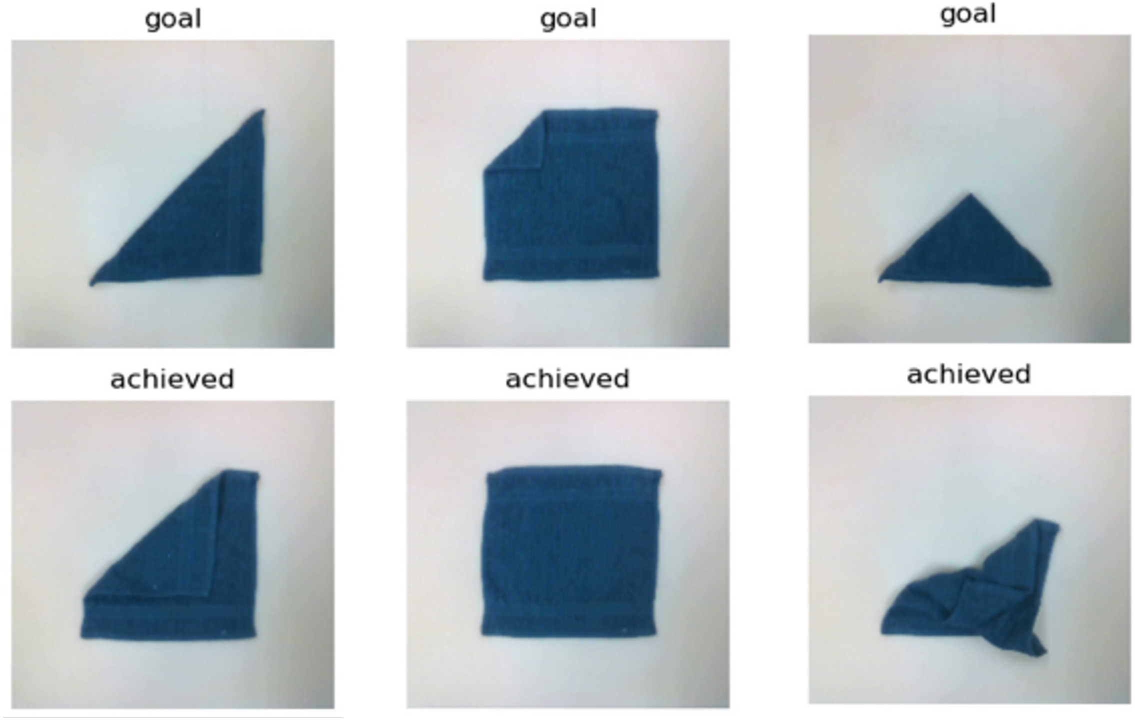
NoFlowPlace: predict place points instead of using flow

NoFlow: Combine depthIn and PredictPlace

NoCornerBias: only use uniformly random training data



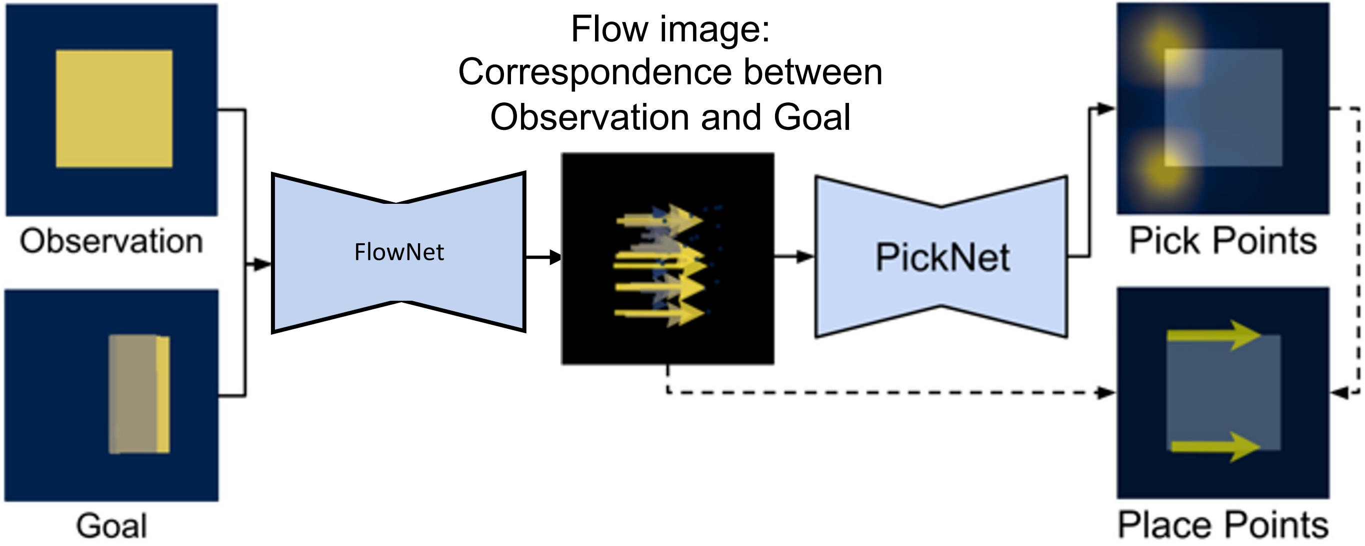
Failure cases



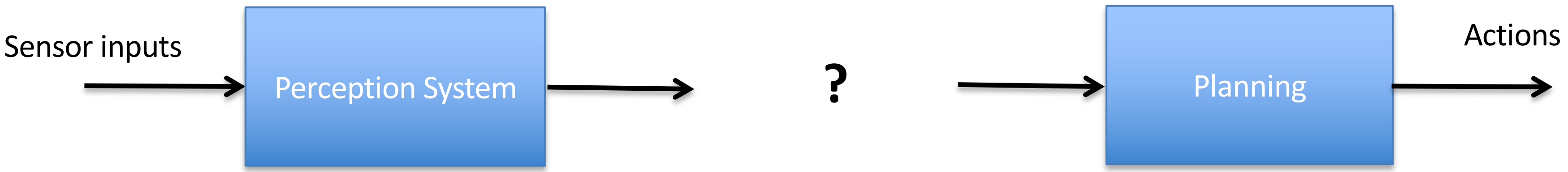
under-estimated flow unfolded multi-step

Conclusion

- Novel flow-based approach for bimanual goal-conditioned cloth manipulation
- Explicit reasoning about desired **object motion** then infer **robot motion**
- Successfully perform cloth folding in real world

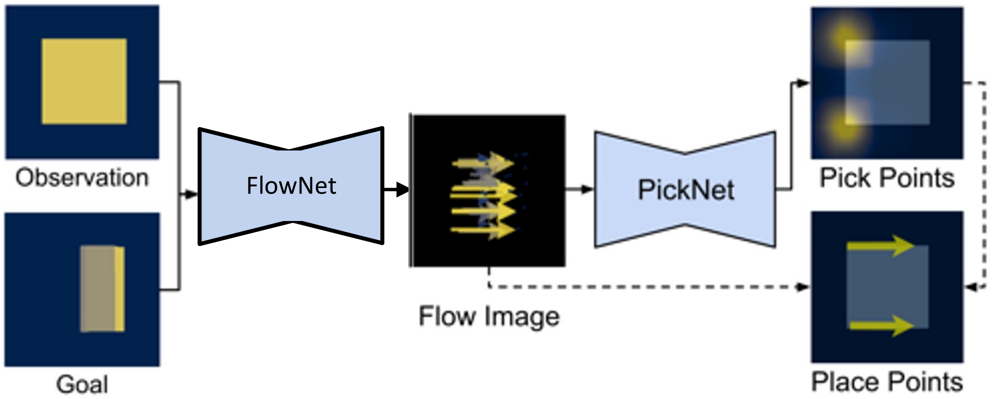


Relational Affordance Learning



Estimating the relationship between **the observation and the goal**

Inferring a pick-and-place robot action to **achieve this desired object motion**



Then use motion planning to perform the bimanual action without collisions

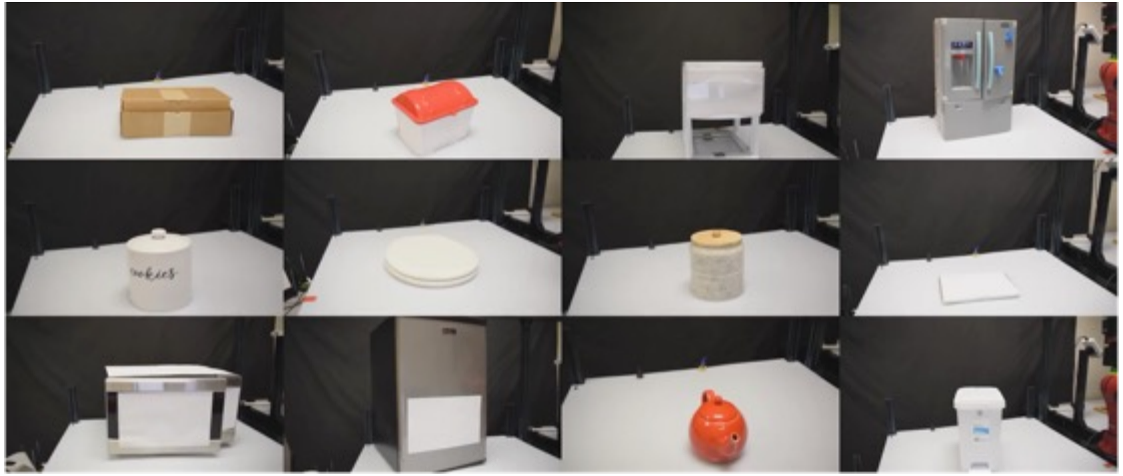
How can robots learn dual arm manipulation policies for non-rigid objects?

Flow-based Policy for Bimanual Goal-conditioned Cloth Flattening (CoRL 2021)

Thomas Weng

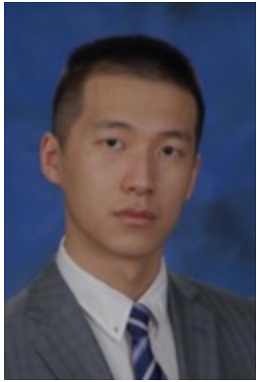
Sujay Bajrachaya

How can robots learn a policy to open any articulated object?

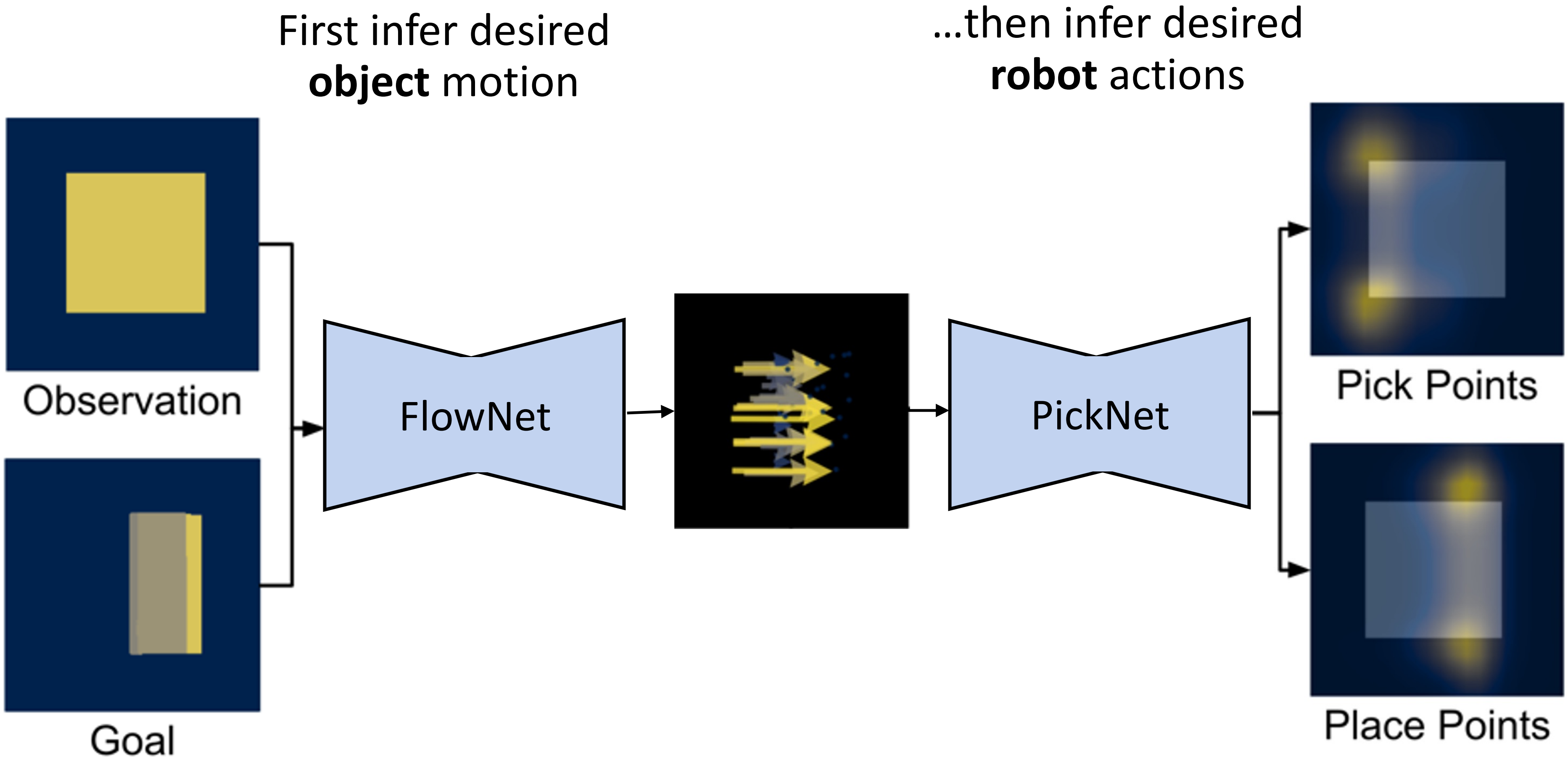


Articulated Object Manipulation
(RSS 2022 – Best Paper Finalist)

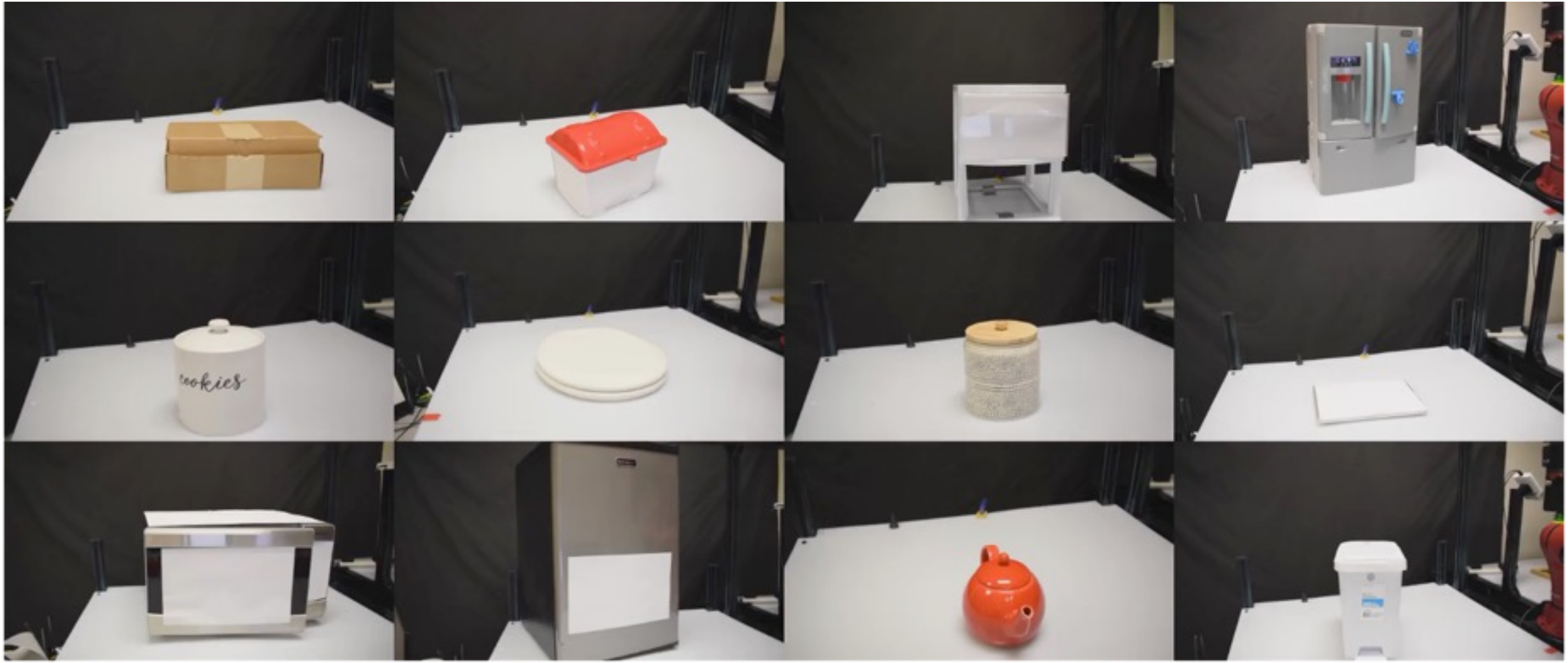
Ben Eisner



Harry Zhang

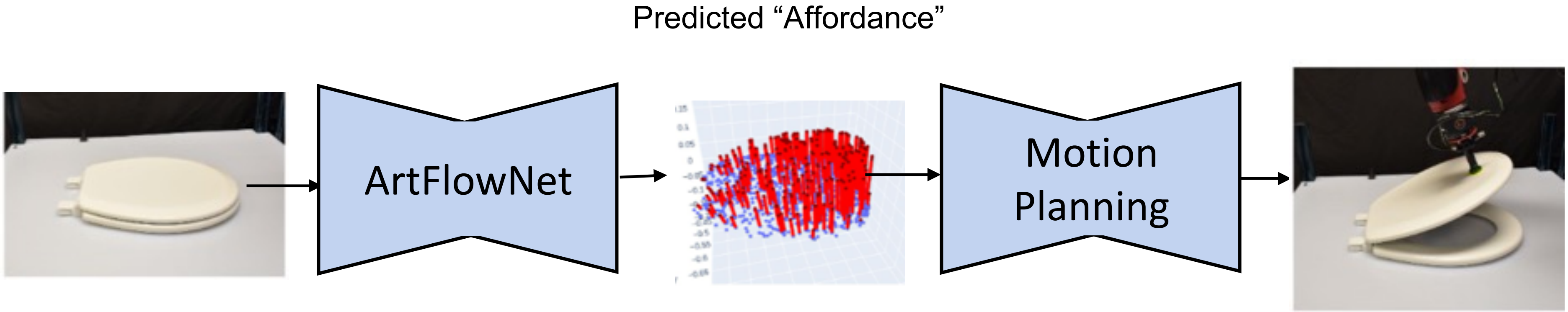

First infer desired
object motion
...then infer desired
robot actions
Observation
Goal
FlowNet
PickNet
Pick Points
Place Points

How can we learn to open unseen articulated objects?



First infer desired **object** motion

…then infer desired **robot** actions



One model trained across all training categories

Training Objects

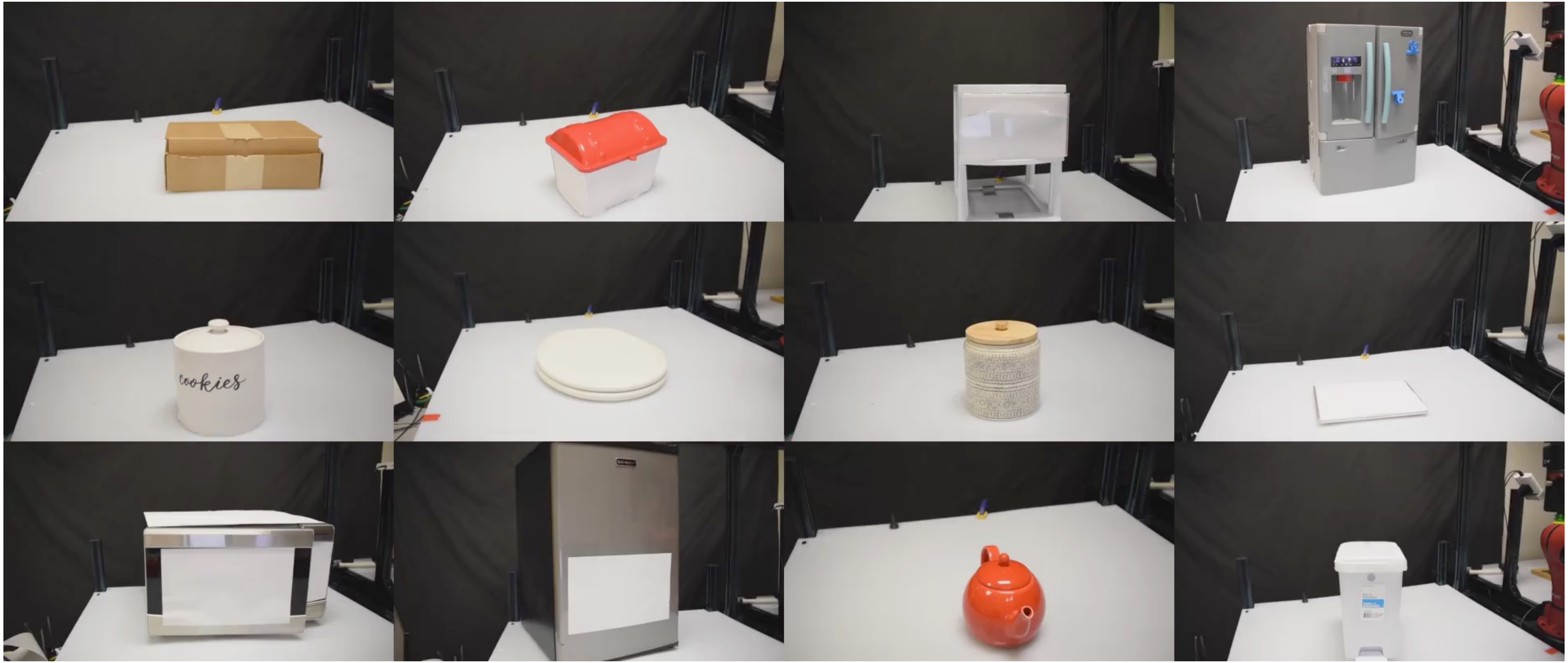
499 objects from 11 categories

Test Objects

Test on 15 objects in the real world

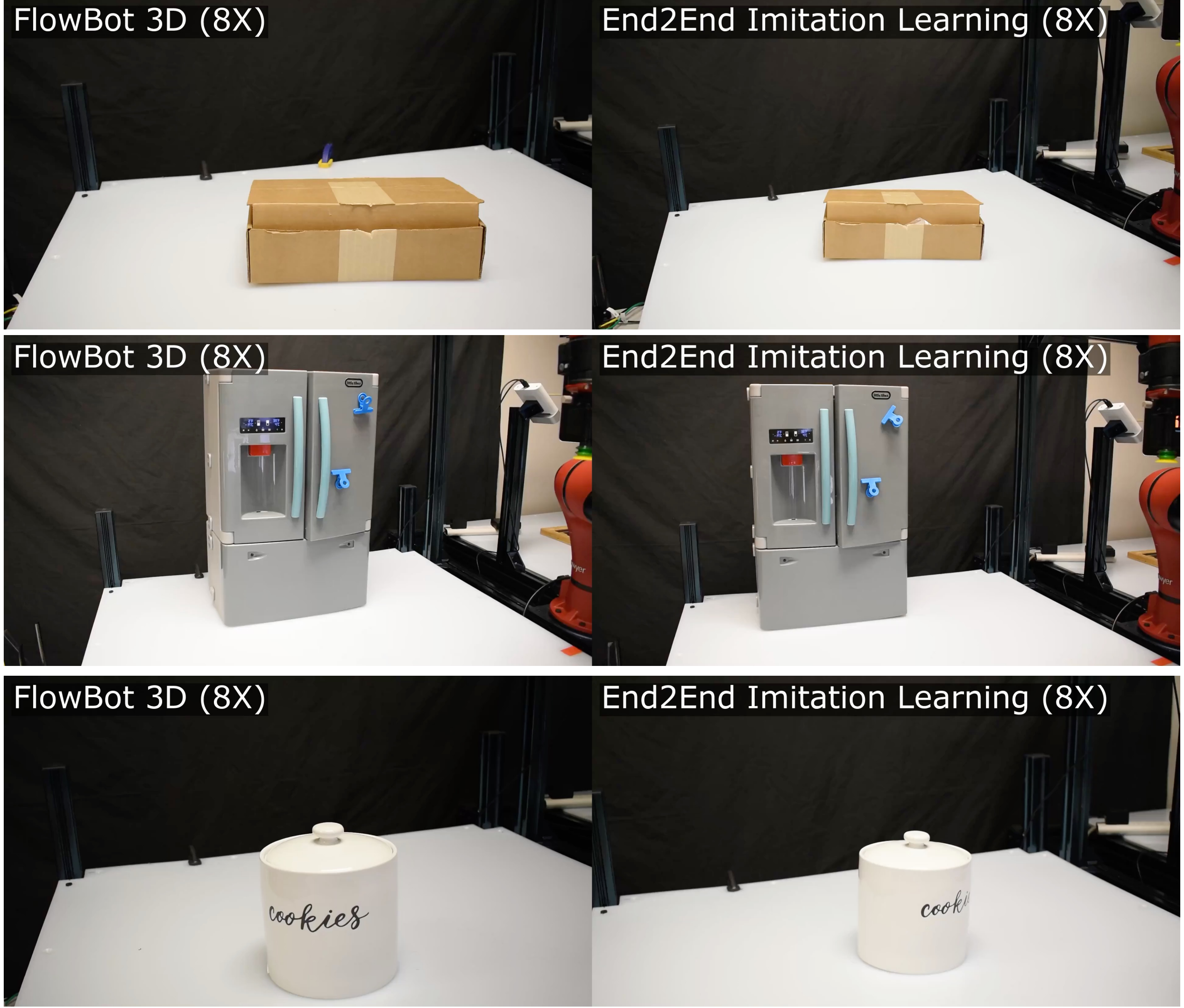
[1] SAPIEN; Xiang et. al. (2020)

Articulating Unseen Objects by Predicting Affordances



(5x)

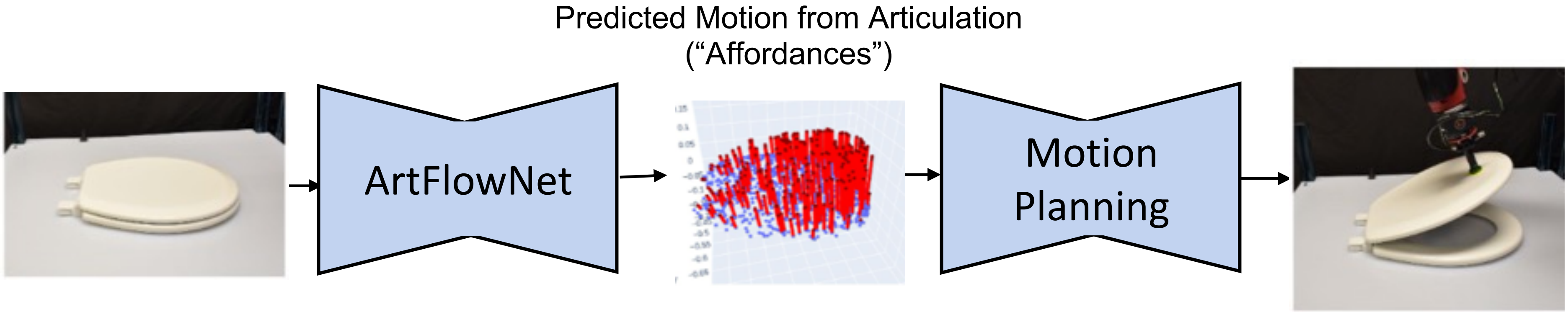
Flow predictions


FlowBot 3D (8X)
End2End Imitation Learning (8X)
FlowBot 3D (8X)
End2End Imitation Learning (8X)
FlowBot 3D (8X)
End2End Imitation Learning (8X)
cookies

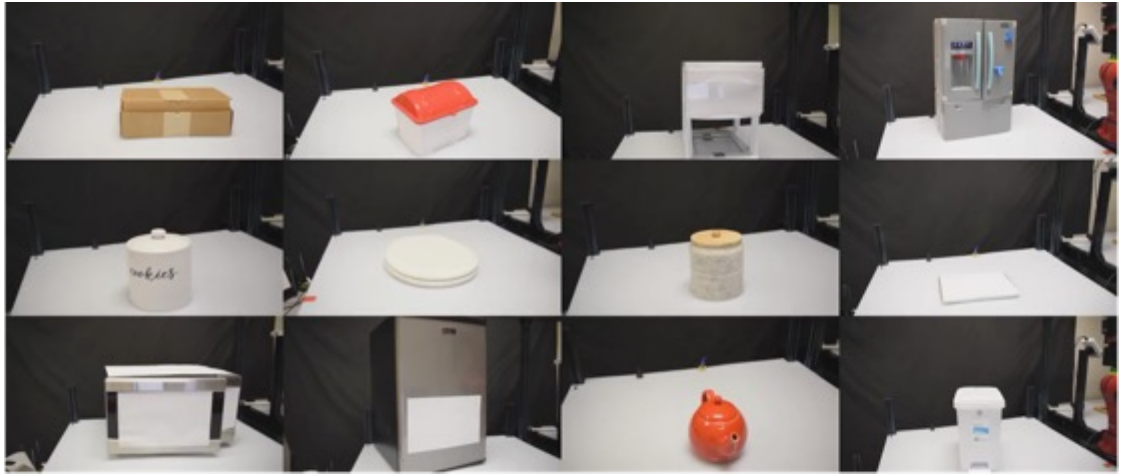
Our Main Insight:

First infer desired **object** motion

...then infer desired **robot** actions

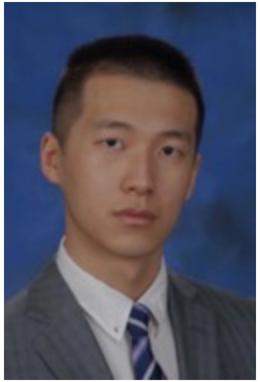


How can robots learn a policy to open any articulated object?



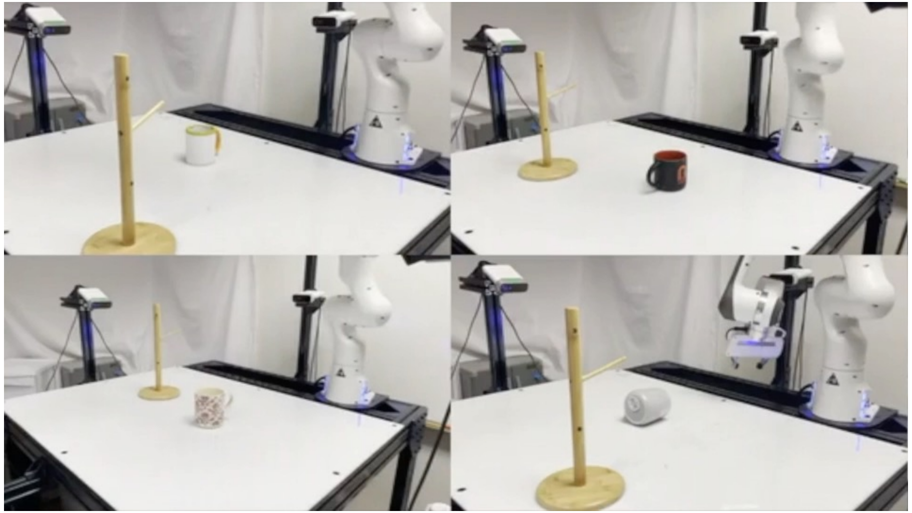
Articulated Object Manipulation
(RSS 2022 – Best Paper Finalist)

Ben Eisner



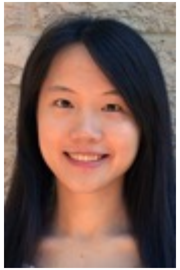
Harry Zhang

How can robots learn a task from just a few real-world demonstrations and generalize to new objects and new configurations?



Task-specific Relative Pose Estimation
(CoRL 2022)

Brian Okorn



Chuer Pan

Ben Eisner

Harry Zhang

Importance of Relative Transforms

Many robotic manipulation tasks are based on object pose relationships.

Importance of Relative Transforms

Many robotic manipulation tasks are based on object pose relationships.

Importance of Relative Transforms

Many robotic manipulation tasks are based on object pose relationships.

Importance of Relative Transforms

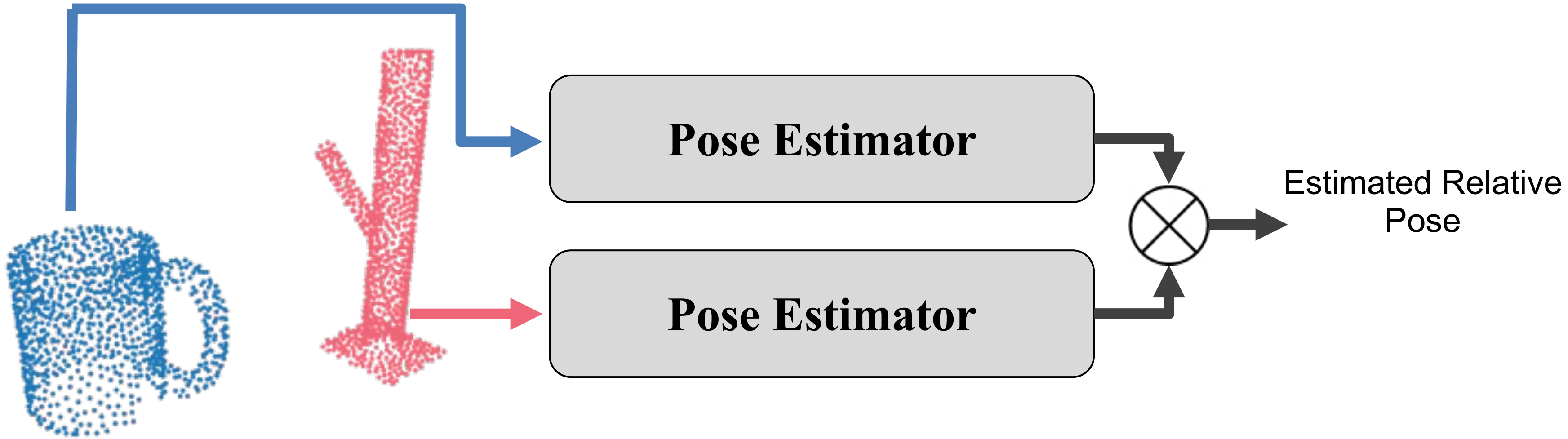
Many robotic manipulation tasks are based on object pose relationships.

Importance of Relative Transforms

Many robotic manipulation tasks are based on object pose relationships.

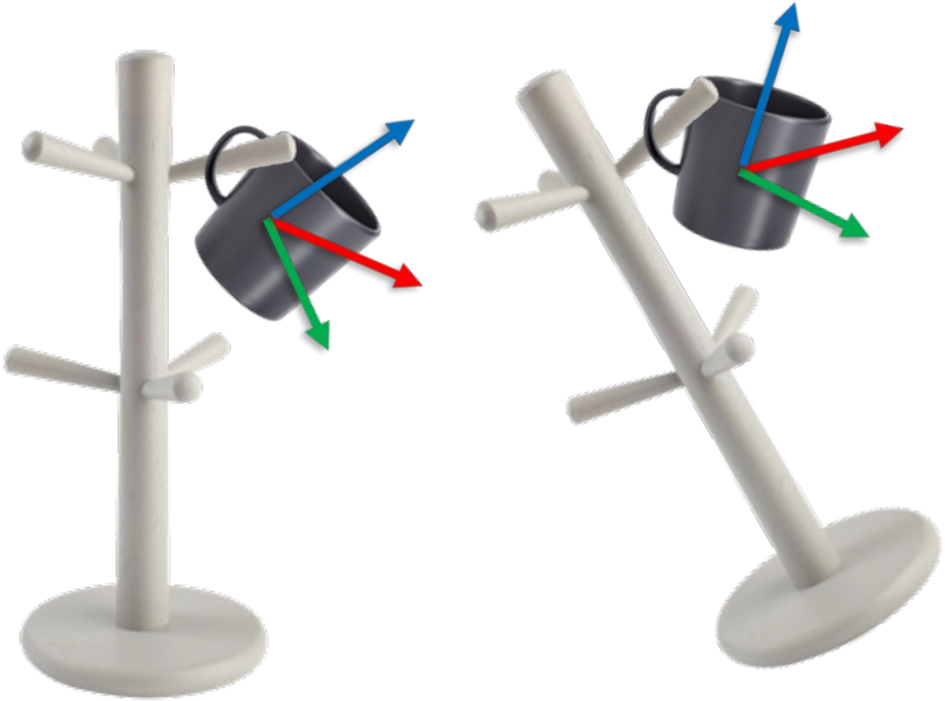
Equivariant relationship between a pair of objects

How do we estimate the relative pose?



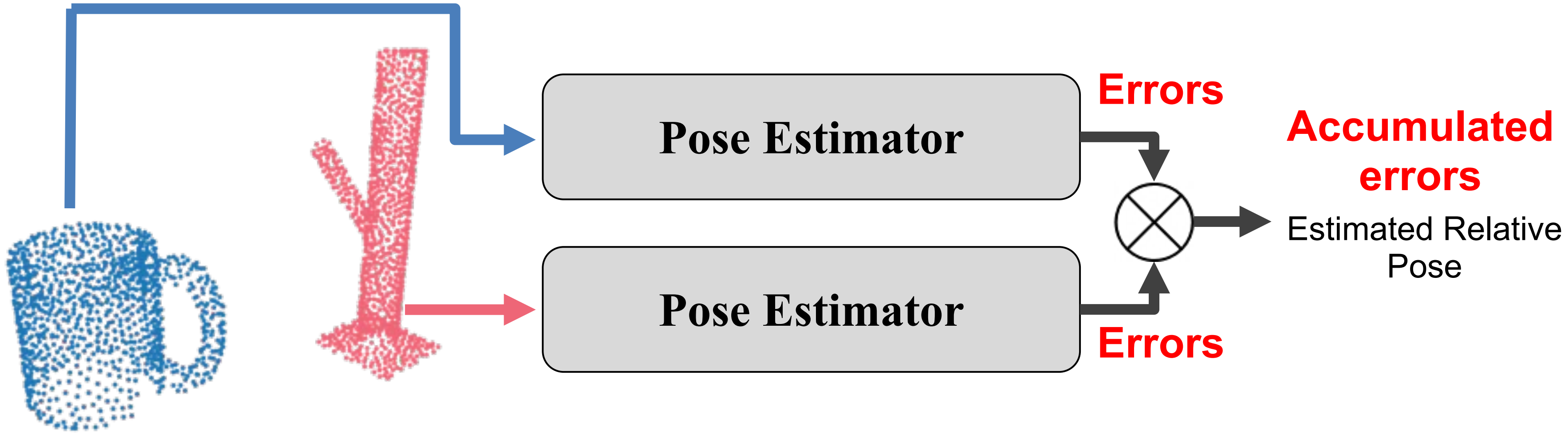
Pose estimators typically require object pose annotations

Our goal: Learn from a small number of demonstrations

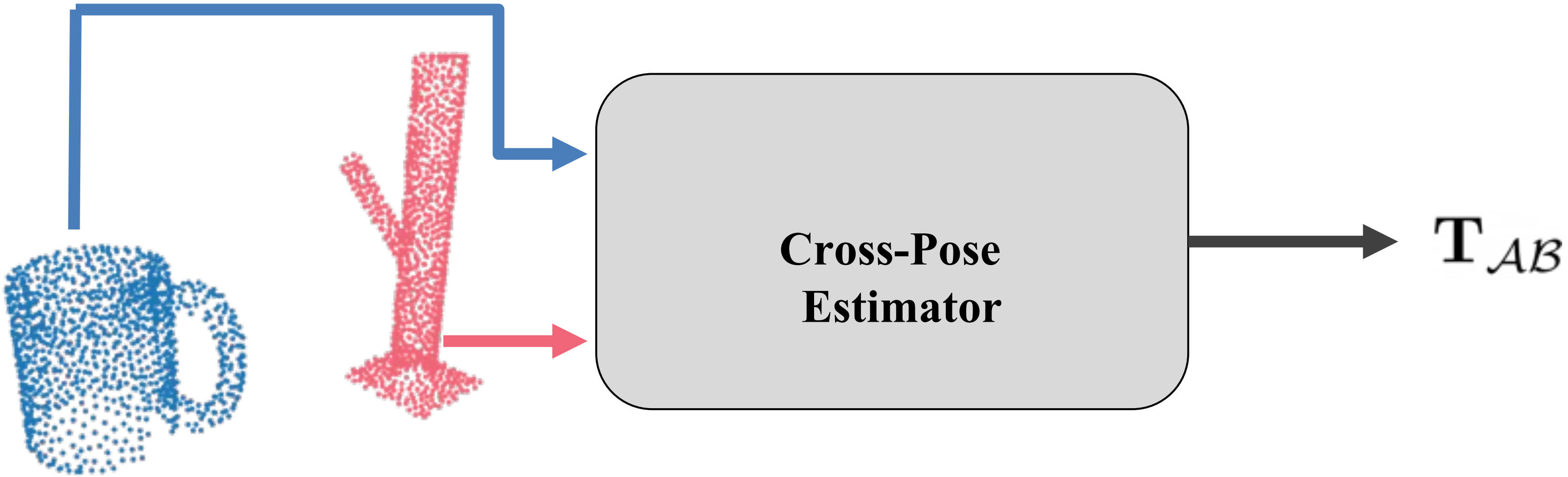


[1] Wang, He, et al. "Normalized object coordinate space for category-level 6d object pose and size estimation." *CVPR* 2019
[2] Weng, Yijia, et al. "Captra: Category-level pose tracking for rigid and articulated objects from point clouds." *ICCV* 2021.
[3] Thompson, Skye, Leslie Pack Kaelbling, and Tomas Lozano-Perez. "Shape-Based Transfer of Generic Skills." *ICRA 2021*.

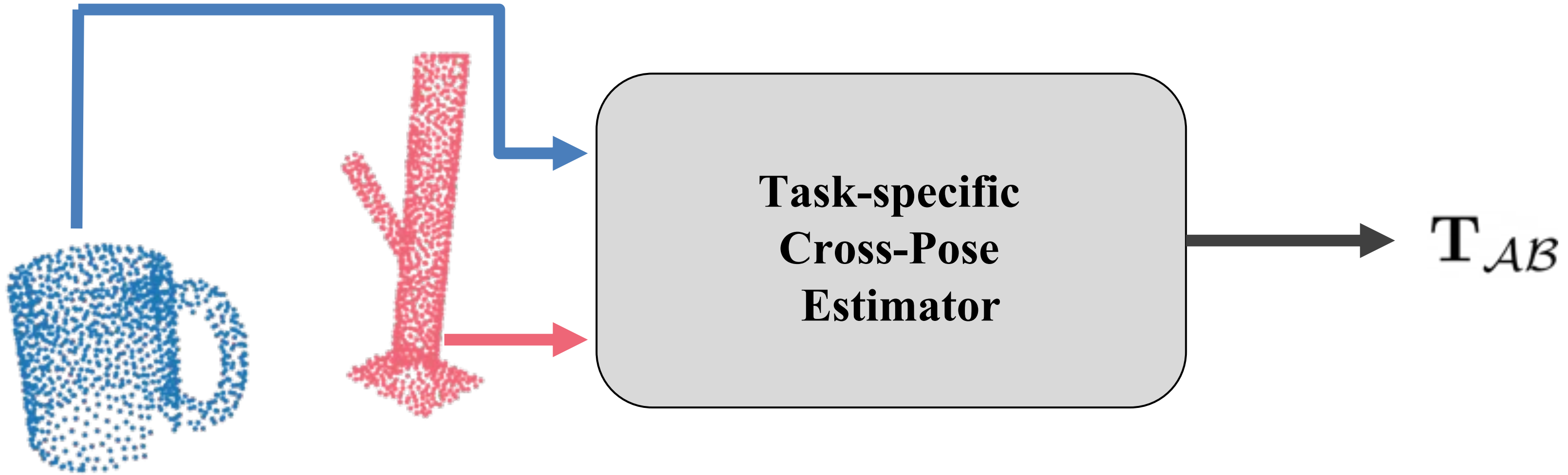
Another Issue: Error accumulation



Our approach:



Our approach:

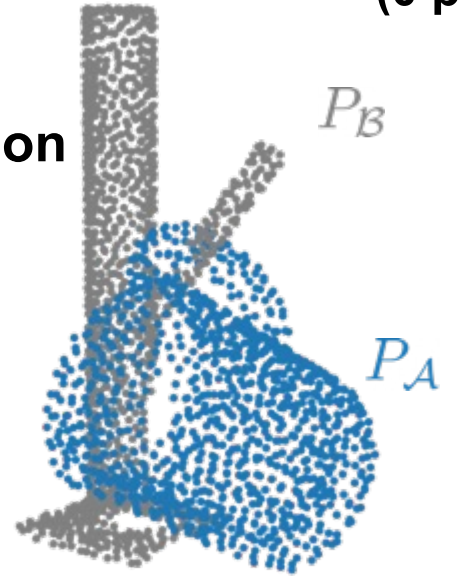


What transformation do we need to apply to object A to move it into the goal pose?

Cross-Pose: A function of **both** object point clouds: $f_\theta(P_{\mathcal{A}}, P_{\mathcal{B}}) = \mathbf{I}$

Cross-pose = Identity
(0 pose)

Goal Configuration

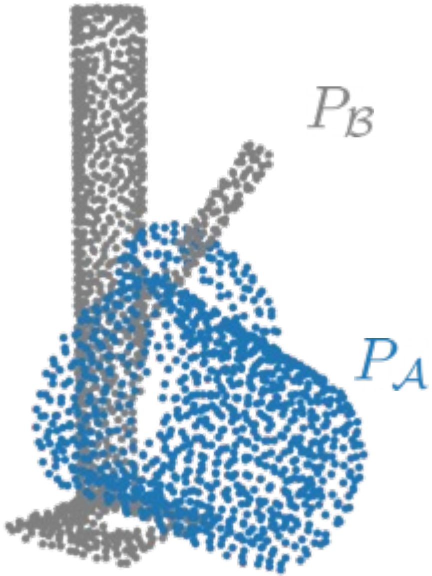


What transformation do we need to apply to object A to move it into the goal pose?

Cross-Pose: A function of **both** object point clouds: $f_\theta(P_\mathcal{A}, P_\mathcal{B}) = \mathbf{I}$

Cross-pose = Identity (0 pose)

Goal Configuration

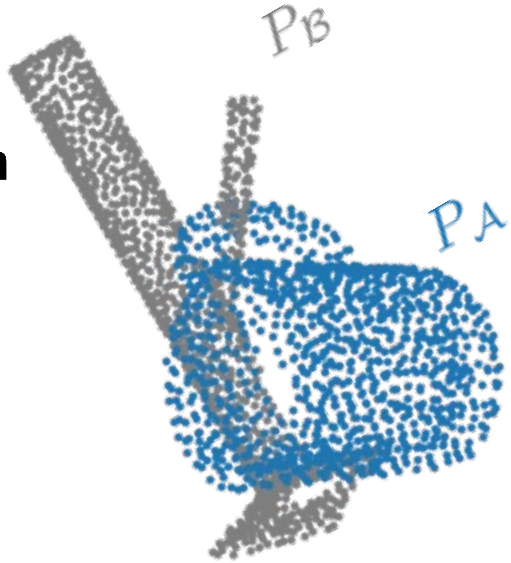


What transformation do we need to apply to object A to move it into the goal pose?

Cross-Pose: A function of **both** object point clouds: $f_\theta(P_\mathcal{A}, P_\mathcal{B}) = \mathbf{I}$

Cross-pose = Identity (0 pose)

Goal Configuration



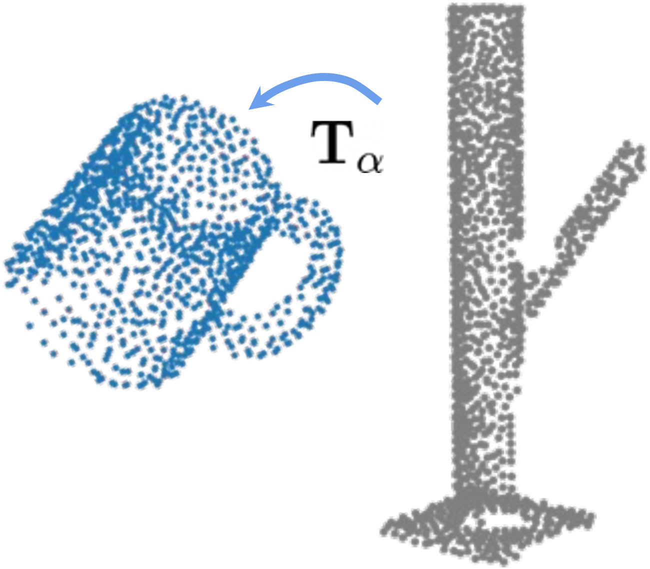
The goal is defined **relative** to the pose of object B

What transformation do we need to apply to object A to move it into the goal pose?

Cross-Pose: A function of **both** object point clouds: $f_\theta(P_\mathcal{A}, P_\mathcal{B}) = \mathbf{I}$

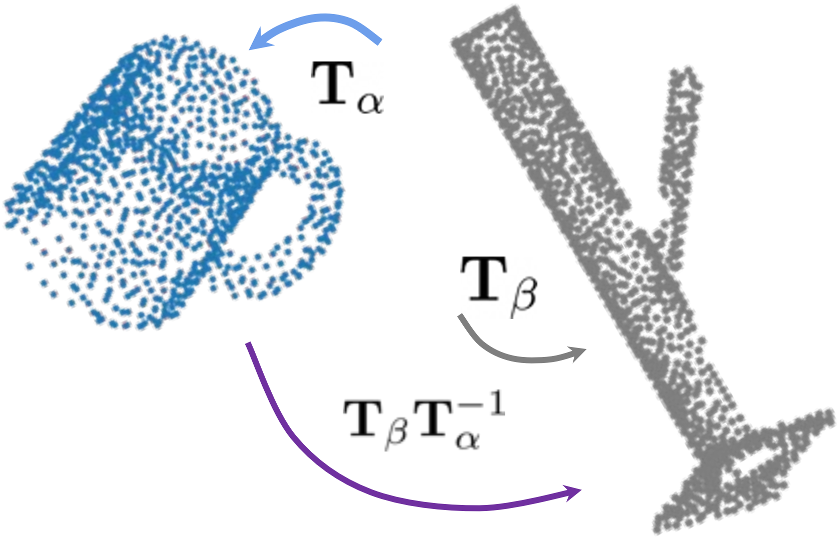
What transformation do we need to apply to object A to move it into the goal pose?

Cross-Pose: A function of **both** object point clouds: $f_\theta(P_{\mathcal{A}}, P_{\mathcal{B}}) = \mathbf{T}_\alpha^{-1}$



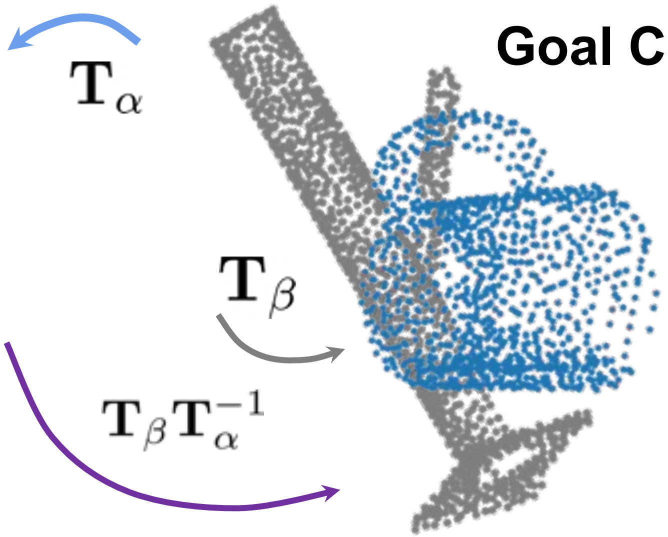
What transformation do we need to apply to object A to move it into the goal pose?

Cross-Pose: A function of **both** object point clouds: $f_\theta(P_{\mathcal{A}}, P_{\mathcal{B}}) = \mathbf{T}_\beta \mathbf{T}_\alpha^{-1}$



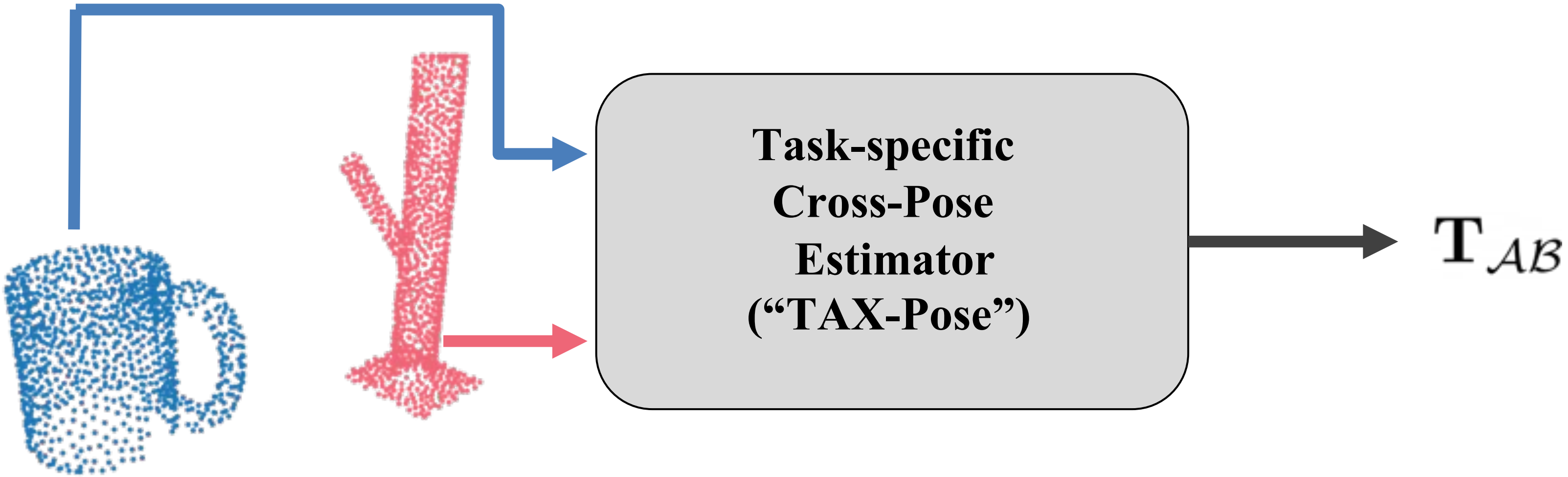
What transformation do we need to apply to object A to move it into the goal pose?

Cross-Pose: A function of **both** object point clouds: $f_\theta(P_\mathcal{A}, P_\mathcal{B}) = \mathbf{T}_\beta \mathbf{T}_\alpha^{-1}$

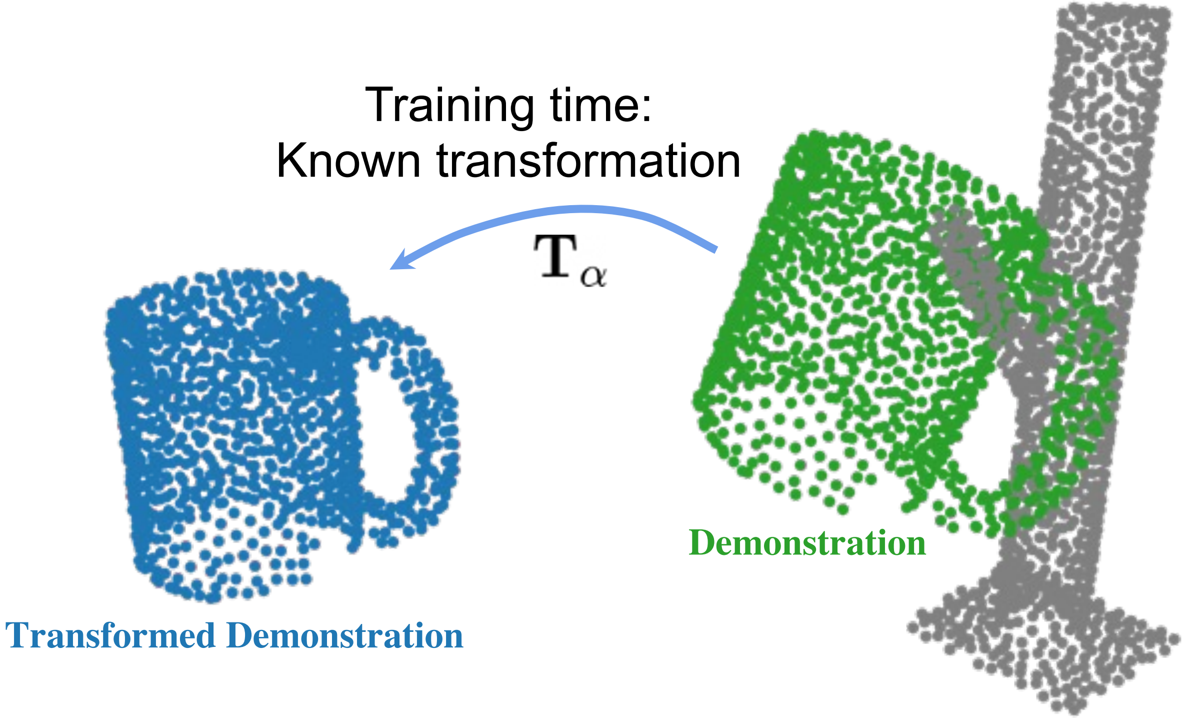


Goal Configuration

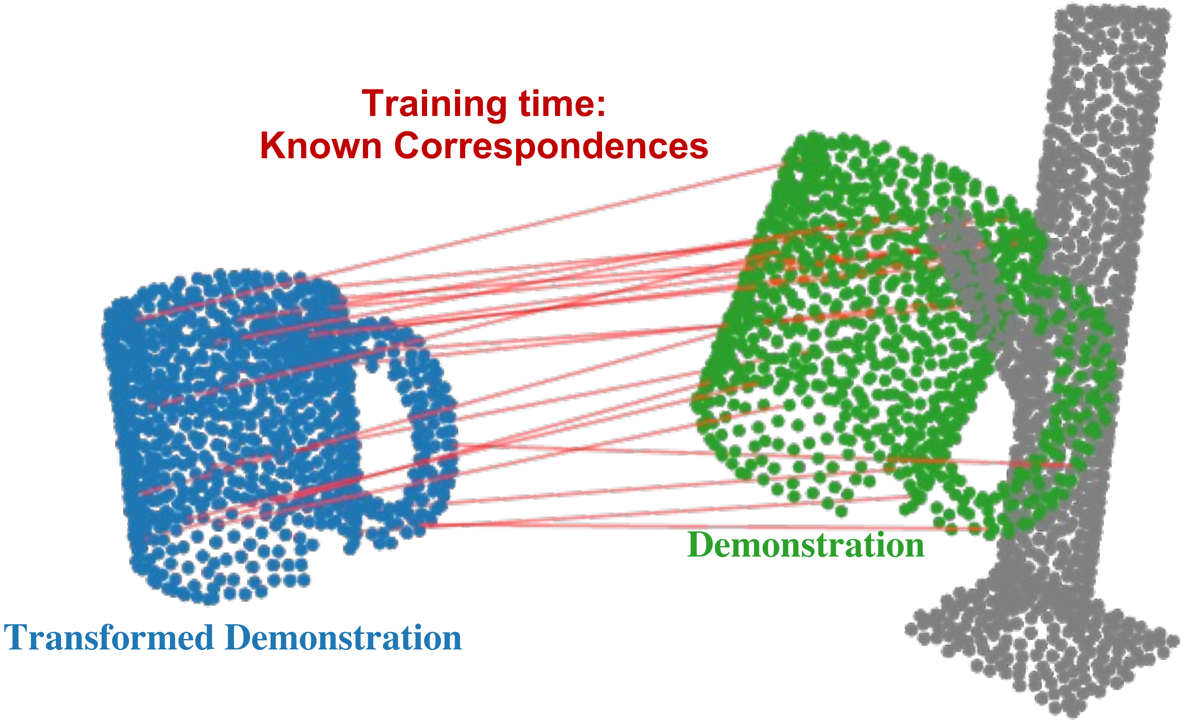
Our approach:



Training TAX-Pose:



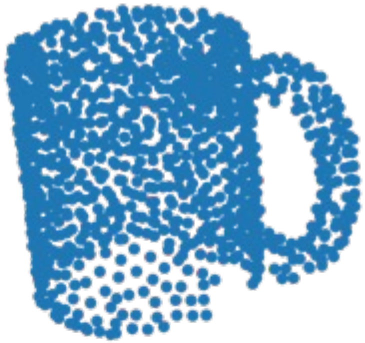
Training TAX-Pose:

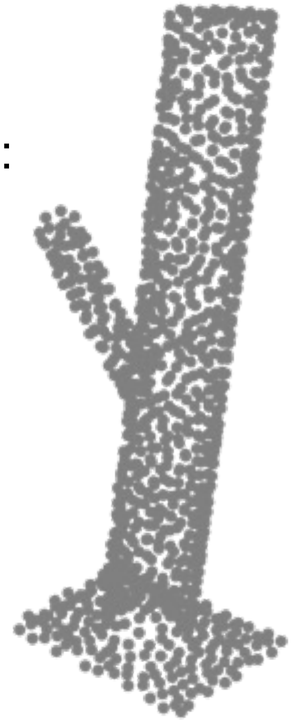


Training TAX-Pose

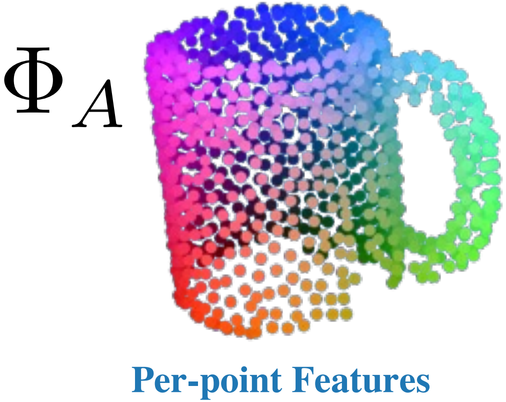
Want to estimate transformation:

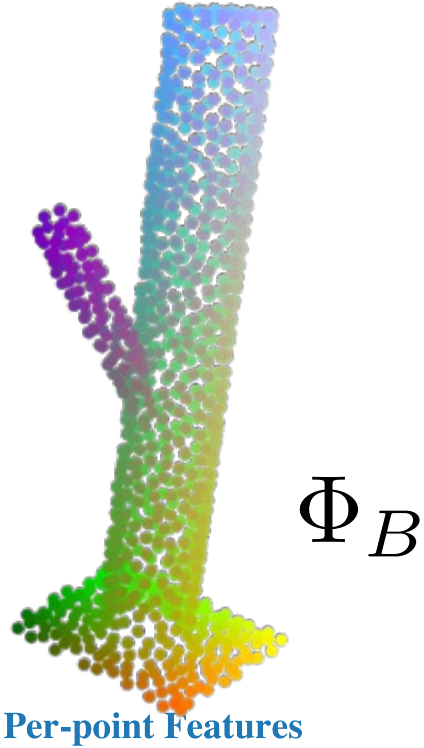
$$\mathbf{T}_\alpha^{-1}$$





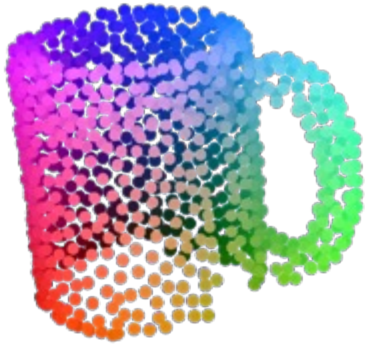
Training TAX-Pose:





Cross-object Correspondences

Φ_A



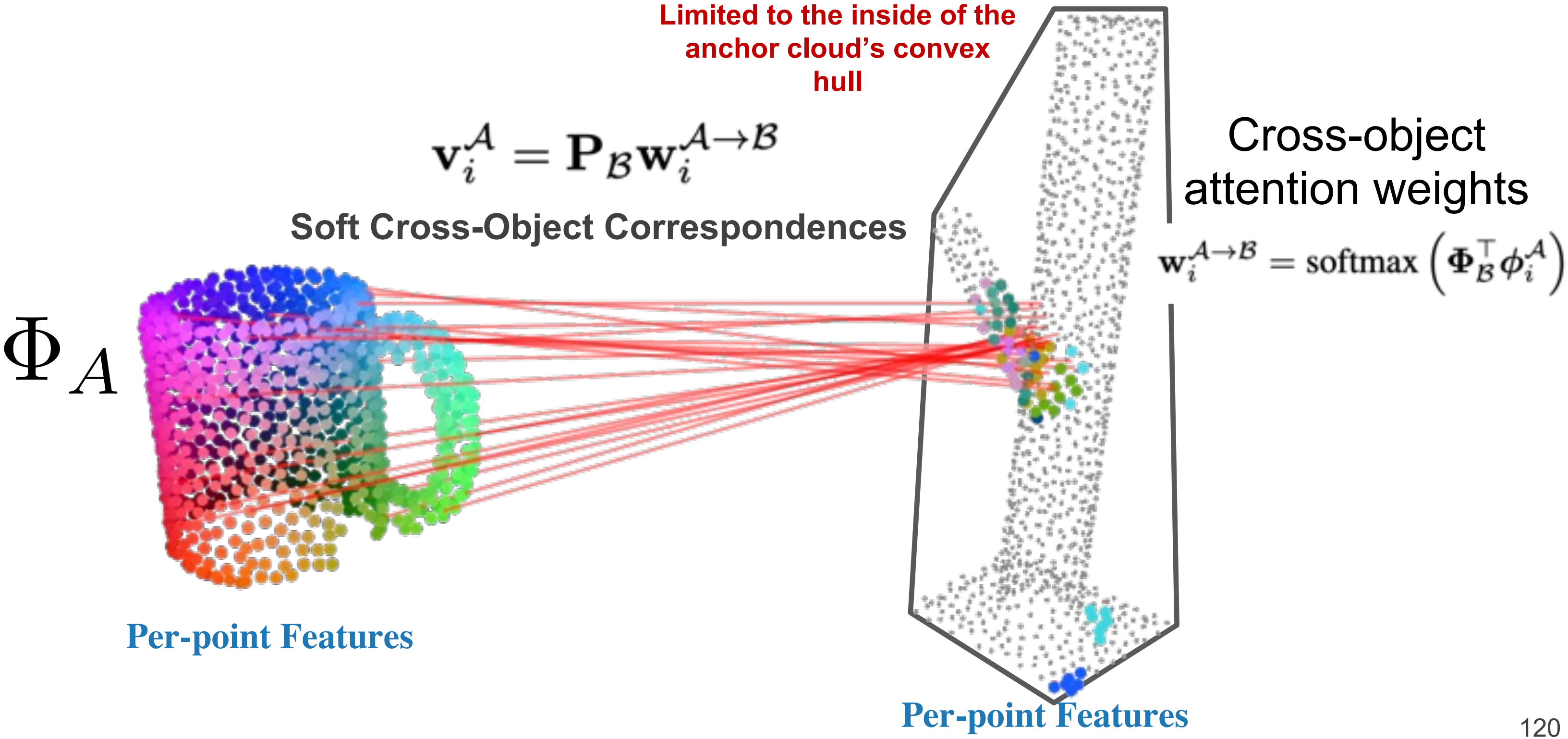
Per-point Features

Cross-object attention weights

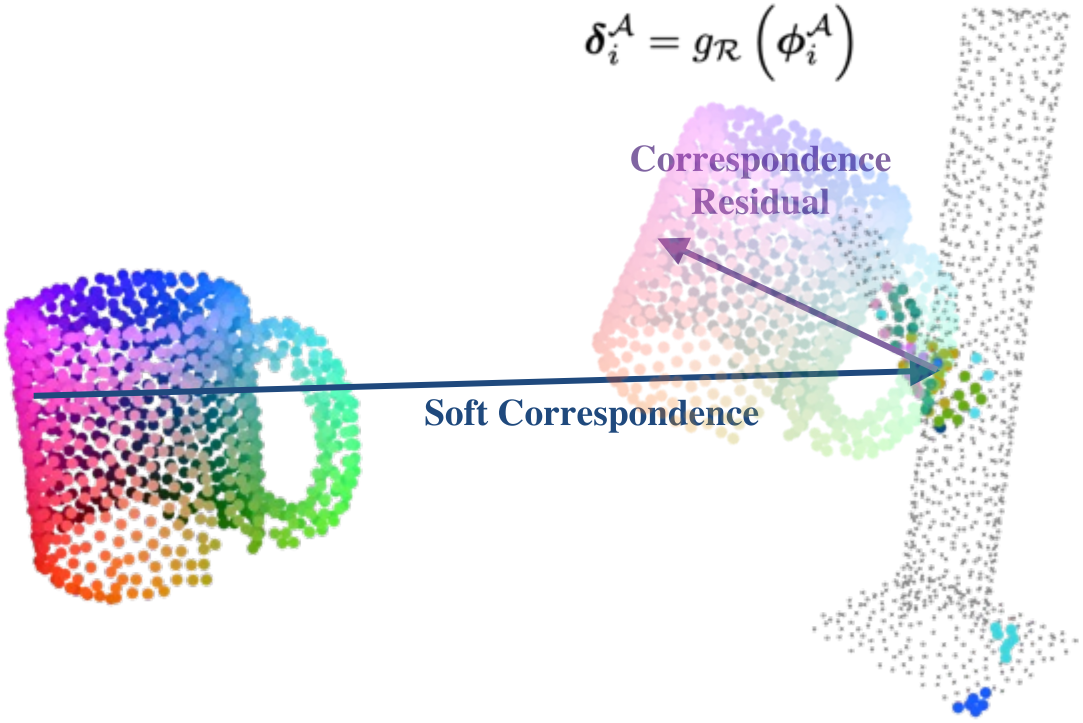
$$\mathbf{w}_i^{\mathcal{A}\to\mathcal{B}} = \mathrm{softmax}\left(\mathbf{\Phi}_{\mathcal{B}}^{\top}\phi_i^{\mathcal{A}}\right)$$

Φ_B

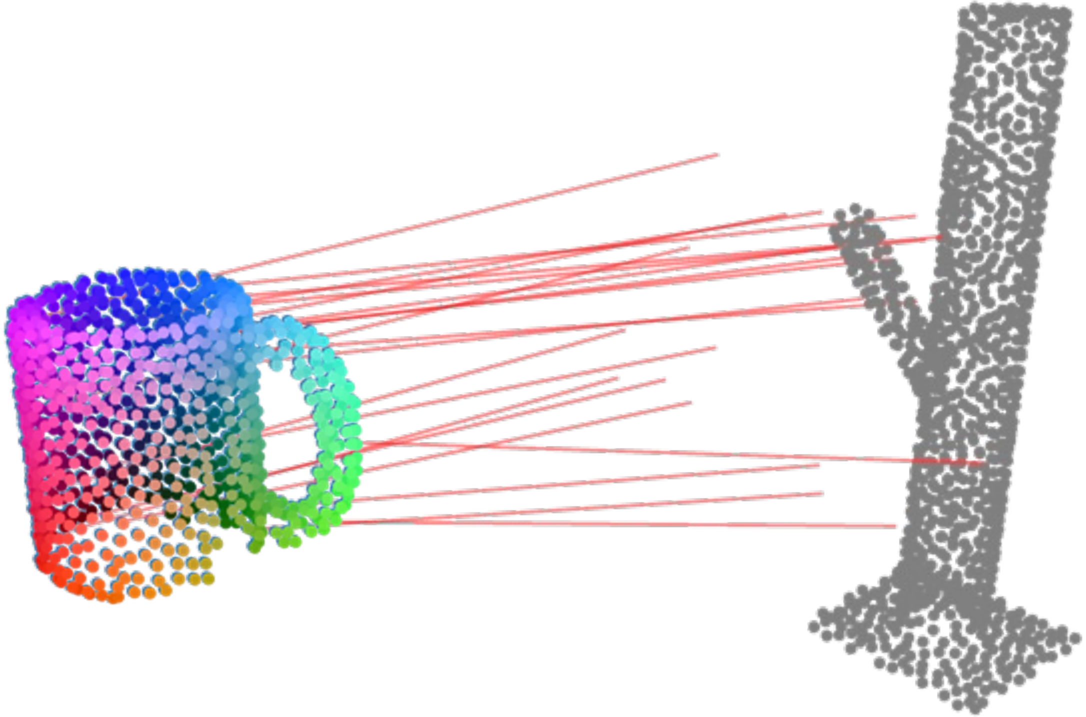
Cross-object Correspondences



Cross-object Correspondences

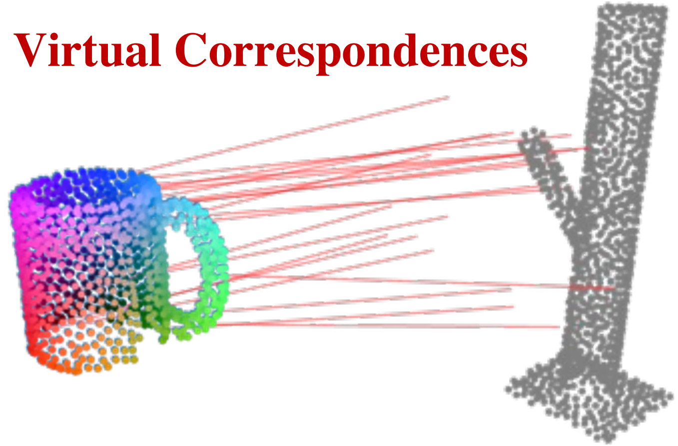


Virtual Correspondencs



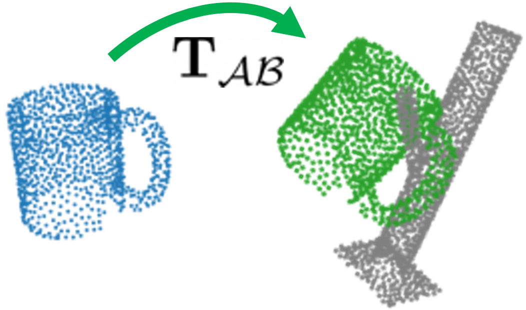
The points cannot move independently!

Virtual Correspondences



Differentiable SVD

Rigid Transformation to the goal pose (Cross-pose)



Least squares optimization:

$$\min_{\mathbf{T}_{\mathcal{AB}}} \sum_{i=1}^{N_{\mathcal{A}}} ||\mathbf{T}_{\mathcal{AB}}\, \mathbf{p}_i^{\mathcal{A}} - \tilde{\mathbf{v}}_i^{\mathcal{A}}||_2^2$$

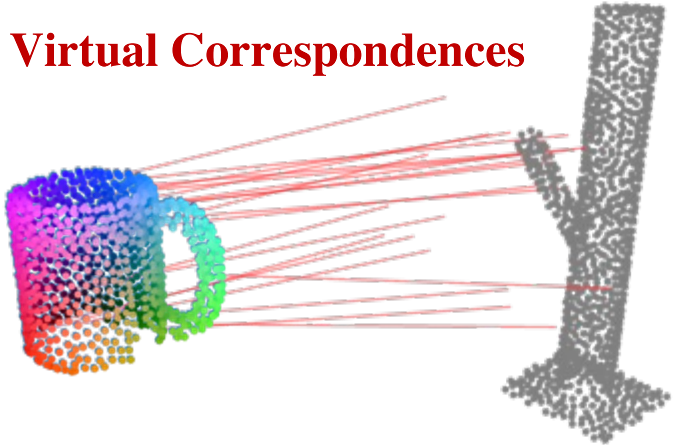
Rigid Transformation

Point

Virtual Correspondences

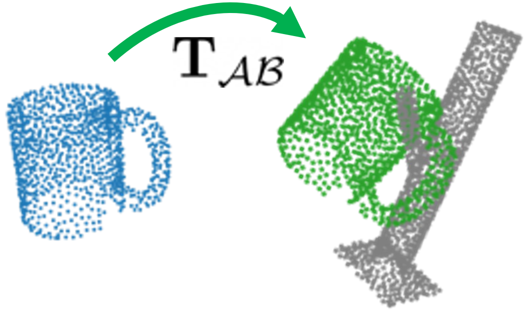
The points cannot move independently!

Virtual Correspondences



Rigid Transformation to the goal pose (Cross-pose)

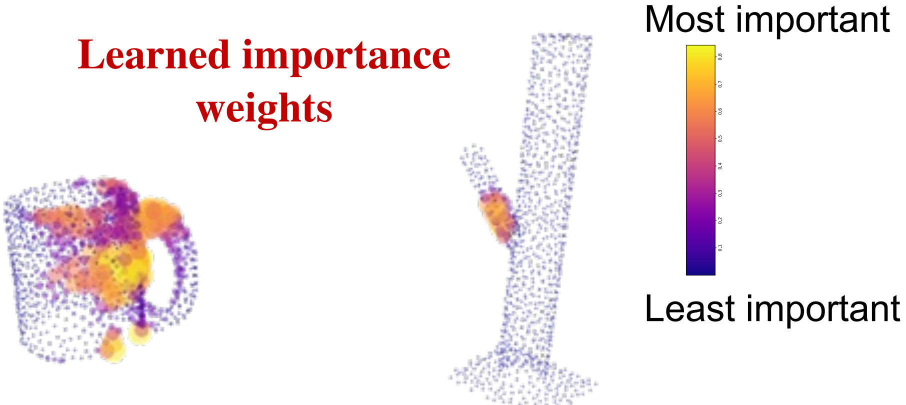
Differentiable SVD



Least squares optimization:

$$\min_{\mathbf{T}_{\mathcal{AB}}} \sum_{i=1}^{N_{\mathcal{A}}} ||\mathbf{T}_{\mathcal{AB}}\,\mathbf{p}_i^{\mathcal{A}} - \tilde{\mathbf{v}}_i^{\mathcal{A}}||_2^2$$

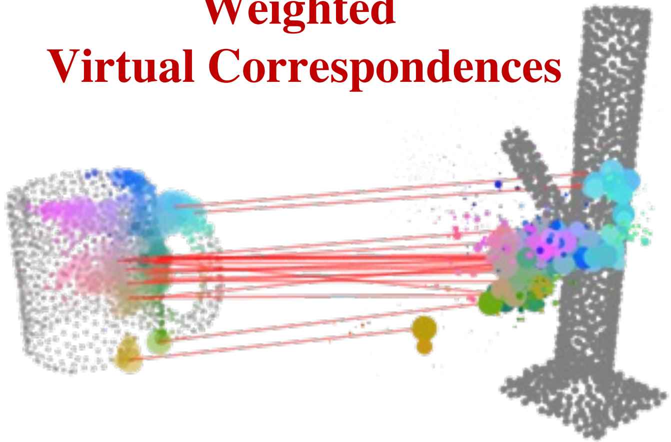
Learned importance weights



Not All Correspondences are Equally Useful

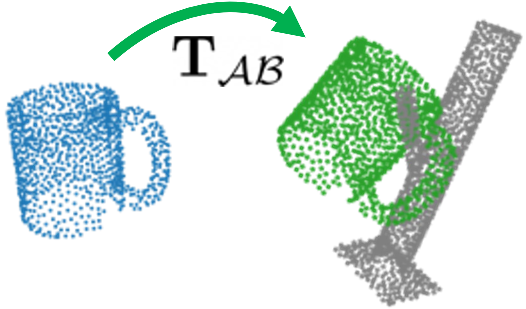
The points cannot move independently!

Weighted Virtual Correspondences



Differentiable SVD

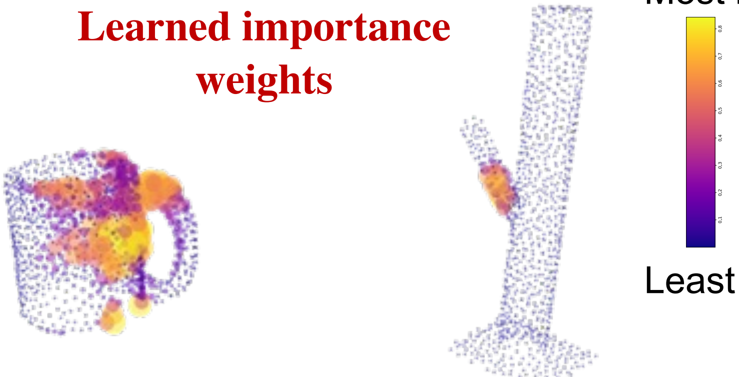
Rigid Transformation to the goal pose (Cross-pose)



Least squares optimization:

$$\min_{\mathbf{T}_{\mathcal{AB}}} \sum_{i=1}^{N_{\mathcal{A}}} ||\mathbf{T}_{\mathcal{AB}}\, \mathbf{p}_i^{\mathcal{A}} - \tilde{\mathbf{v}}_i^{\mathcal{A}}||_2^2$$

Learned importance weights



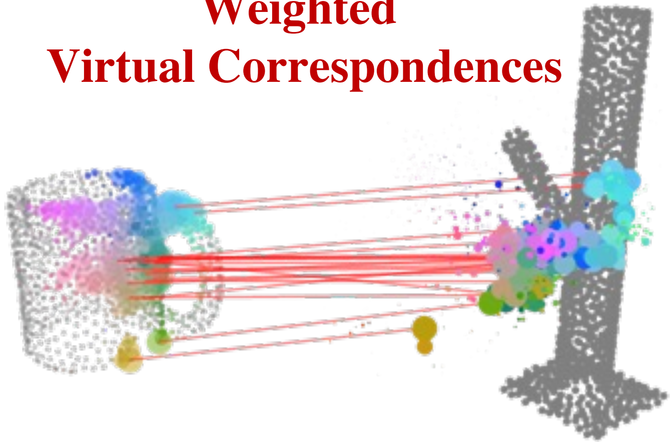
Most important

Least important

Not All Correspondences are Equally Useful

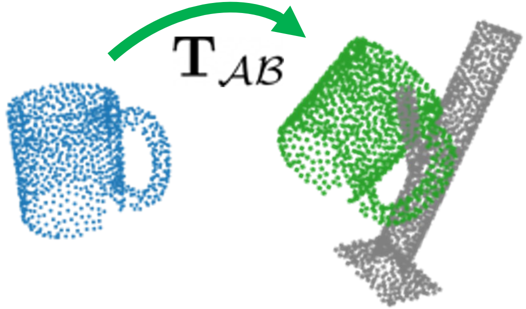
The points cannot move independently!

Weighted Virtual Correspondences



Weighted Differentiable SVD

Rigid Transformation to the goal pose (Cross-pose)

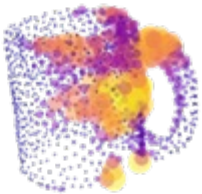


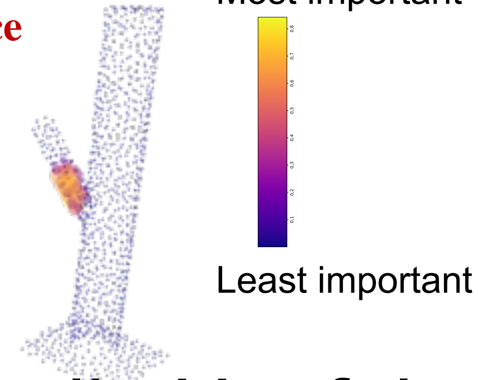
Weighted least squares optimization:

$$\min_{\mathbf{T}_{\mathcal{AB}}} \sum_{i=1}^{N_{\mathcal{A}}} \alpha_i^{\mathcal{A}} ||\mathbf{T}_{\mathcal{AB}}\, \mathbf{p}_i^{\mathcal{A}} - \tilde{\mathbf{v}}_i^{\mathcal{A}}||_2^2$$

Learned importance weight

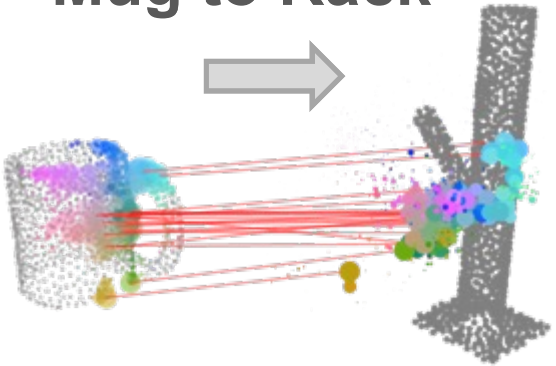
Learned importance weights



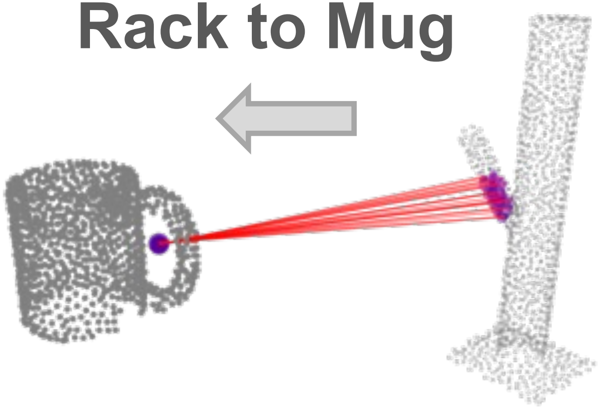


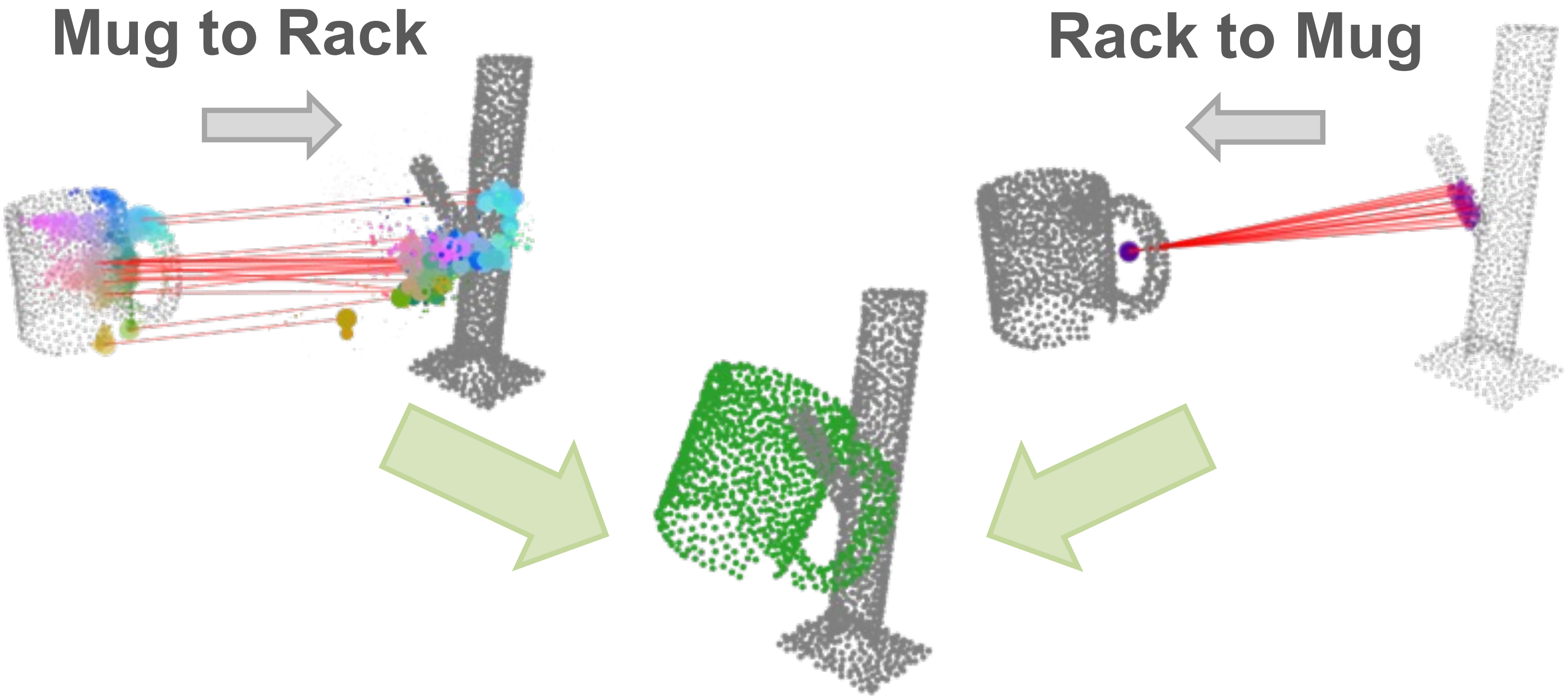
Not All Correspondences are Equally Useful

Mug to Rack



Rack to Mug





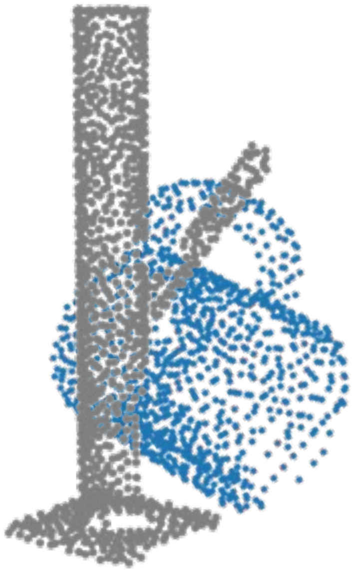
Bi-directional weighted least squares optimization:

$$\min_{\mathbf{T}_{\mathcal{AB}}} \sum_{i=1}^{N_{\mathcal{A}}} \alpha_i^{\mathcal{A}} ||\mathbf{T}_{\mathcal{AB}}\, \mathbf{p}_i^{\mathcal{A}} - \tilde{\mathbf{v}}_i^{\mathcal{A}}||_2^2 + \sum_{i=1}^{N_{\mathcal{B}}} \alpha_i^{\mathcal{B}} ||\mathbf{T}_{\mathcal{AB}}^{-1}\, \mathbf{p}_i^{\mathcal{B}} - \tilde{\mathbf{v}}_i^{\mathcal{B}}||_2^2$$

Mug to Rack　　Rack to Mug

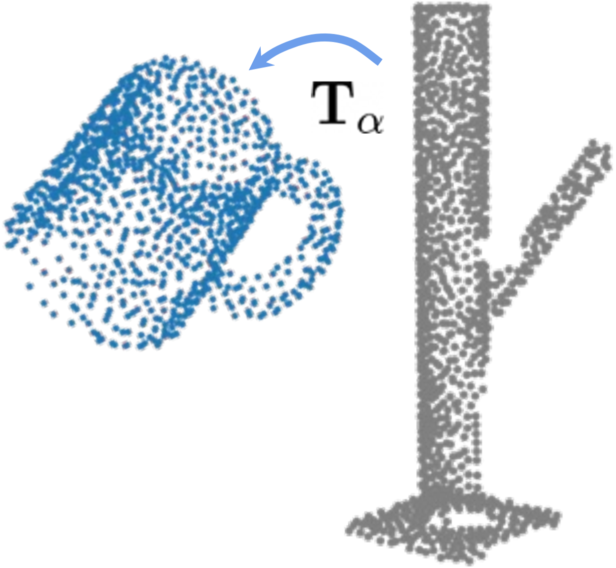
Training TAX-Pose

Estimated as a function of **both** object point clouds: $f_\theta(P_{\mathcal{A}}, P_{\mathcal{B}}) = \mathbf{I}$



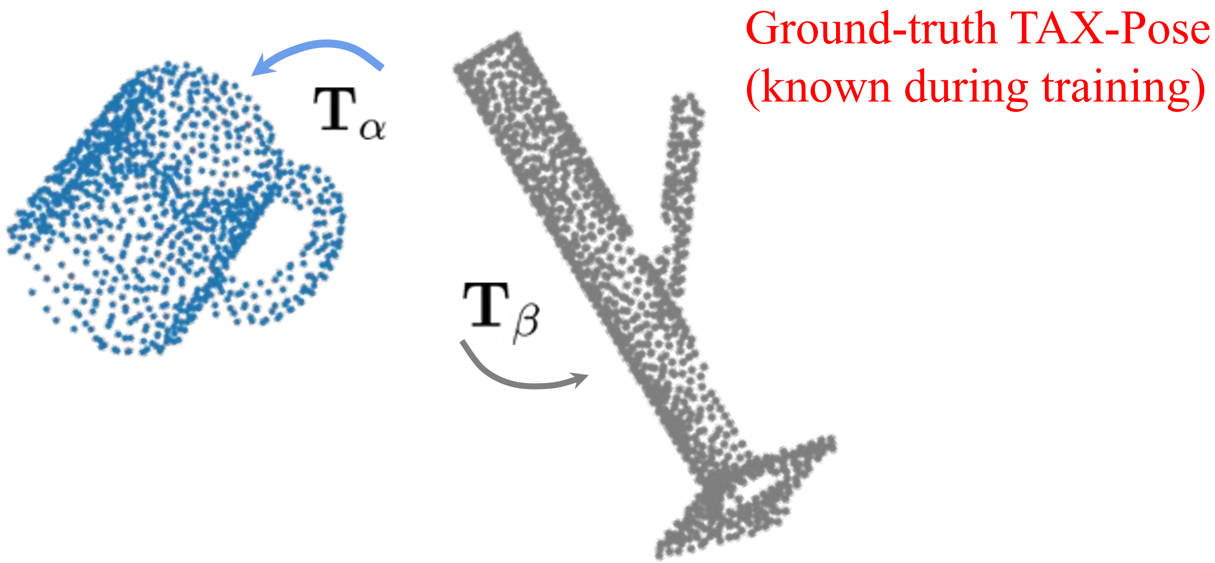
Training TAX-Pose

Estimated as a function of **both** object point clouds: $f_\theta(P_\mathcal{A}, P_\mathcal{B}) = \mathbf{T}_\alpha^{-1}$

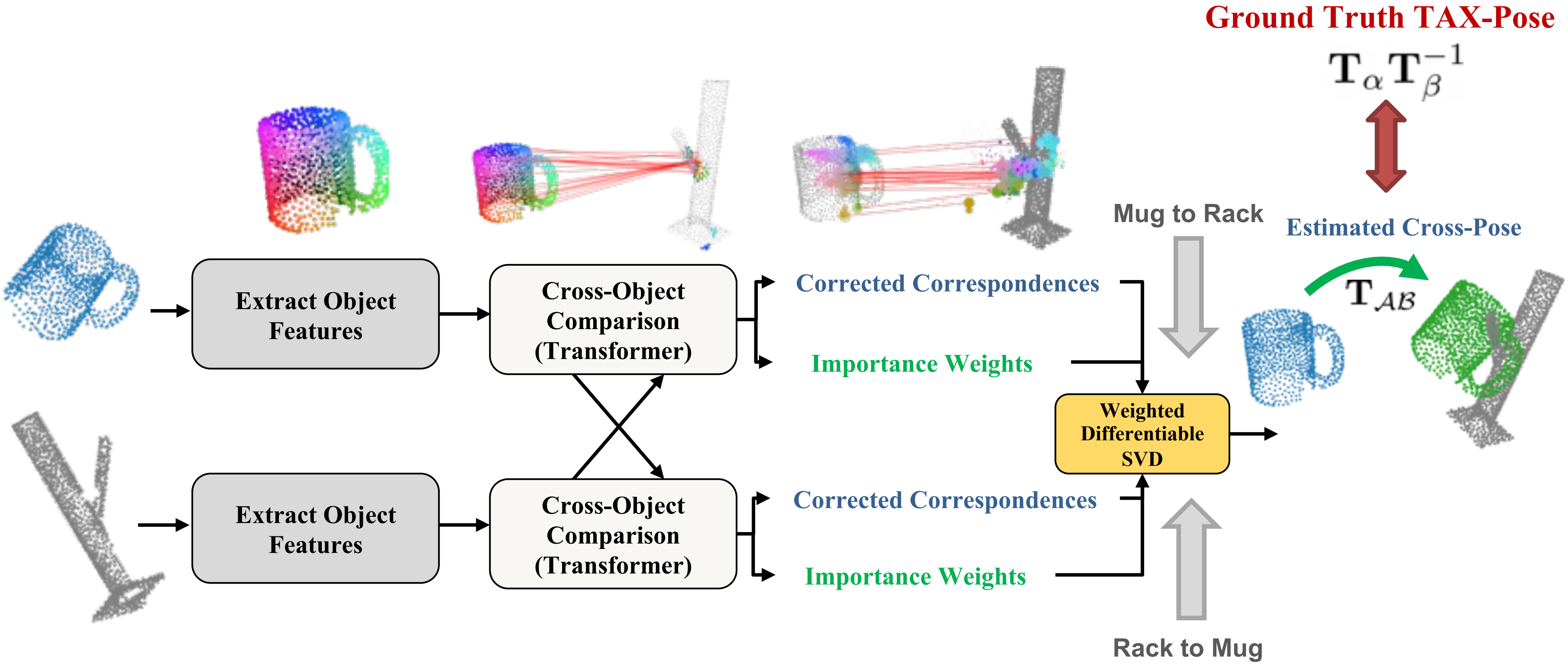


Training TAX-Pose

Estimated as a function of **both** object point clouds: $f_\theta(P_\mathcal{A}, P_\mathcal{B}) = \mathbf{T}_\beta \mathbf{T}_\alpha^{-1}$

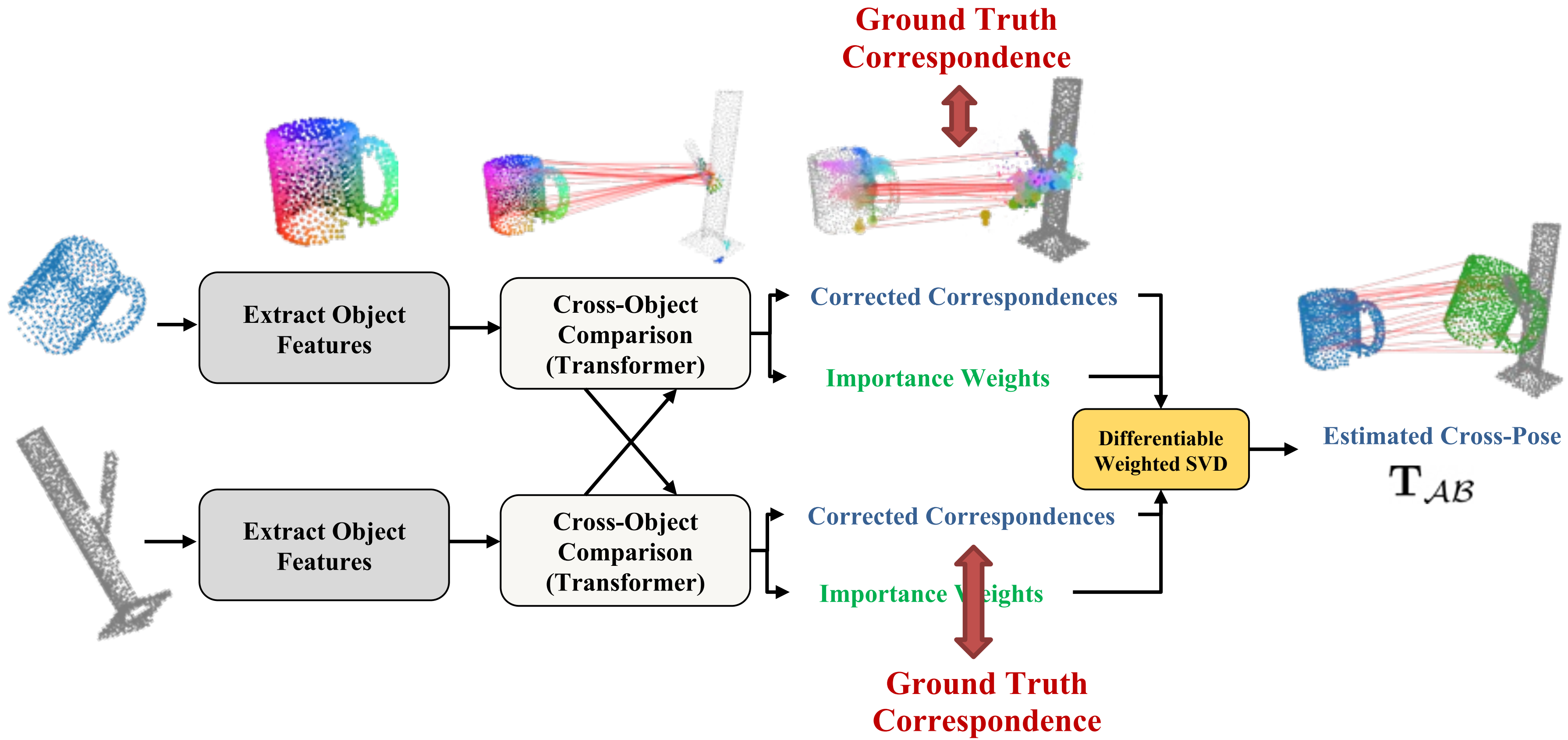


Training TAX-Pose

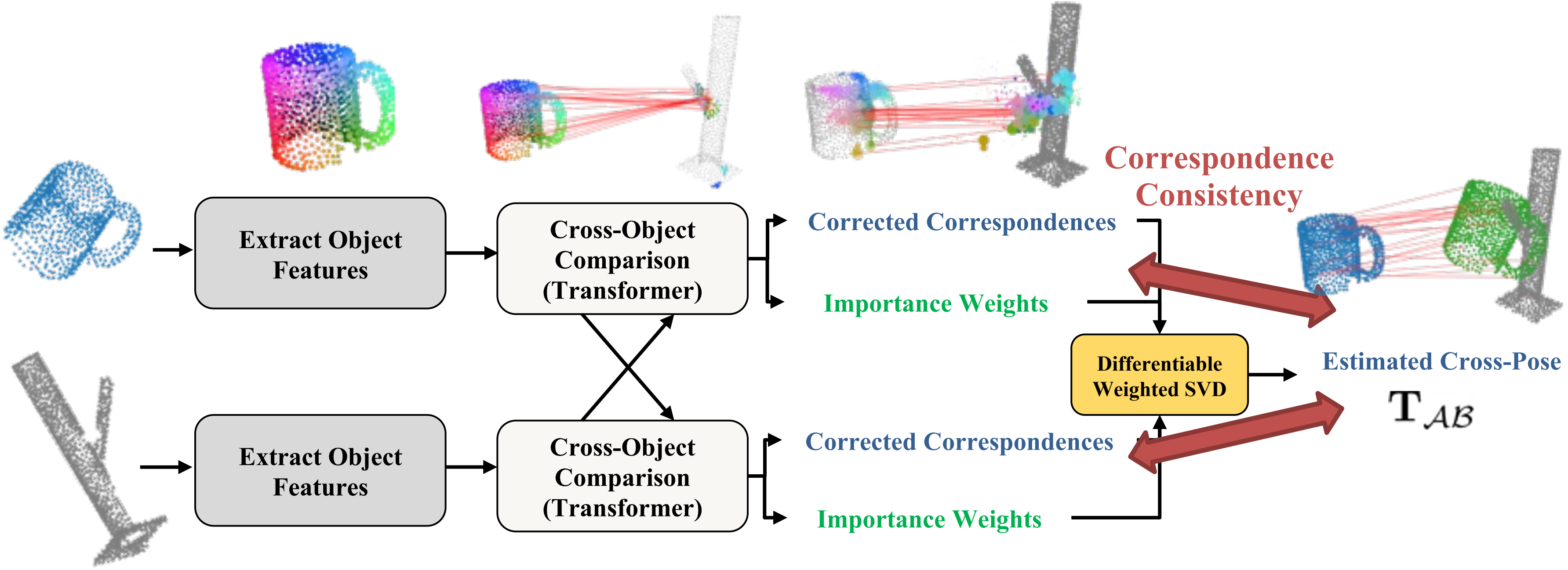


Motion planning to achieve this desired cross-pose

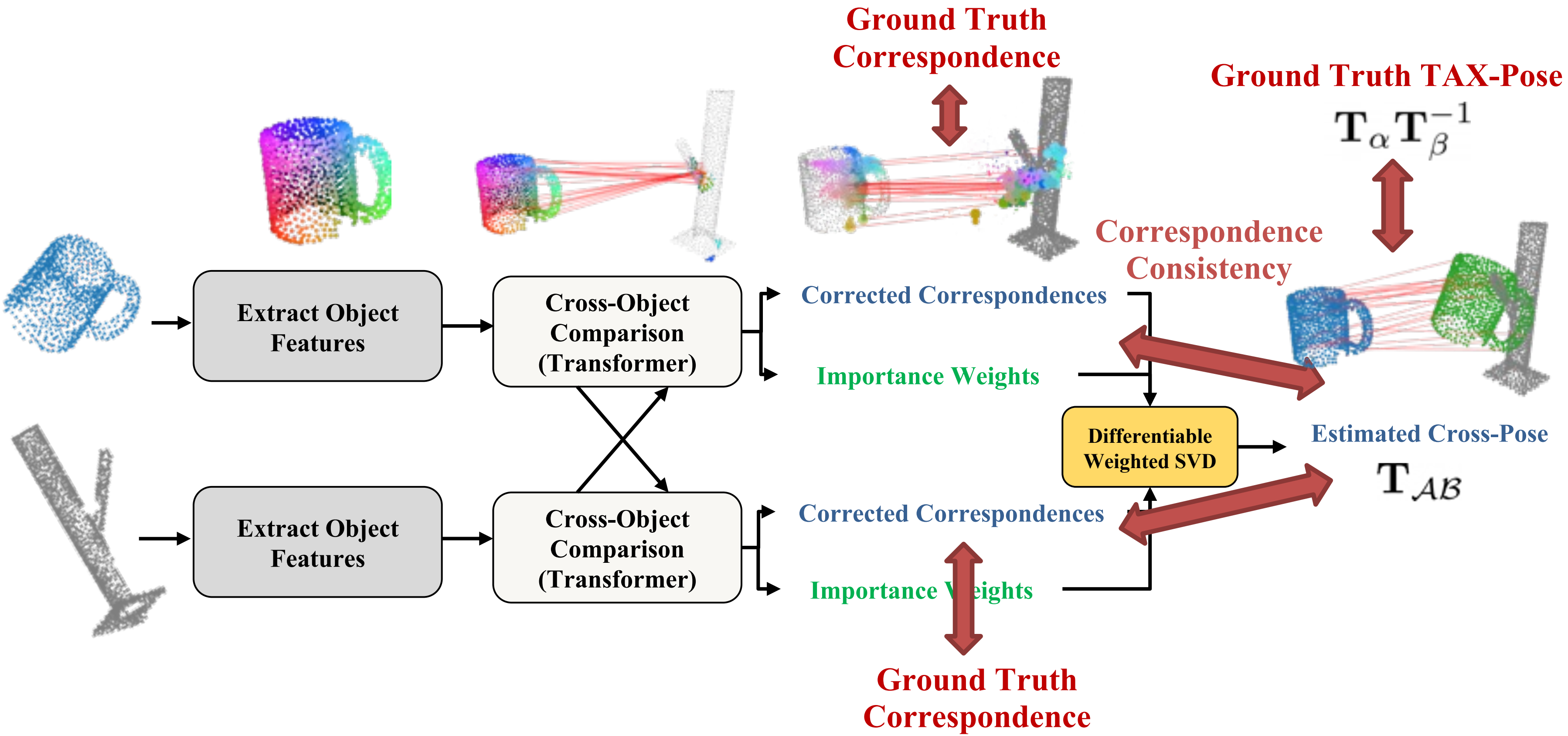
Training TAX-Pose



Training TAX-Pose



Training TAX-Pose

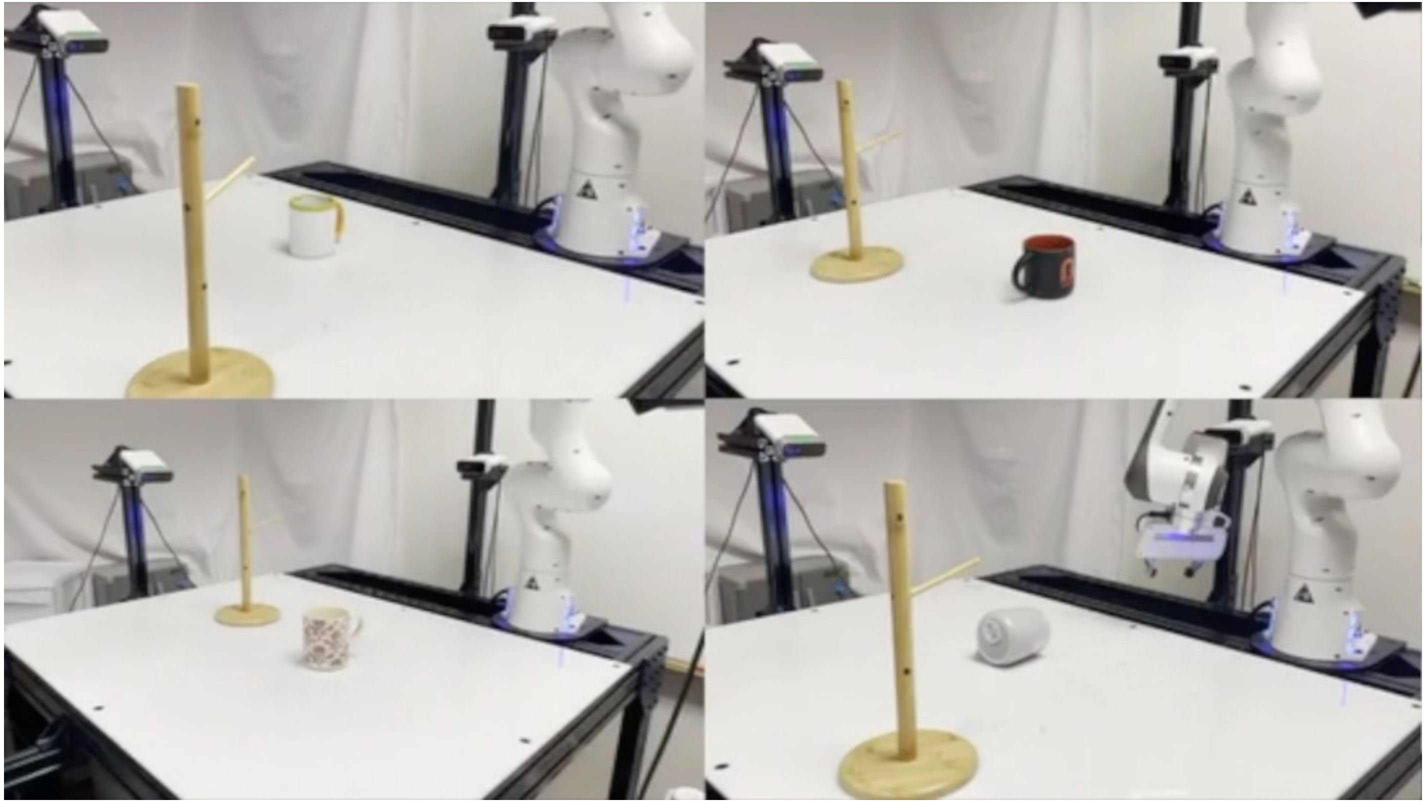


Category-level generalization

Given only **10 demos in the real world** from training mugs

Complete the task with **unseen** object instances

Mug Hanging Results



4x

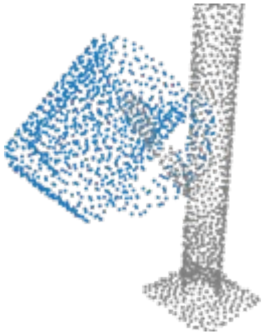
Our method generalizes to unseen mug instances and poses using only a few demonstrations.

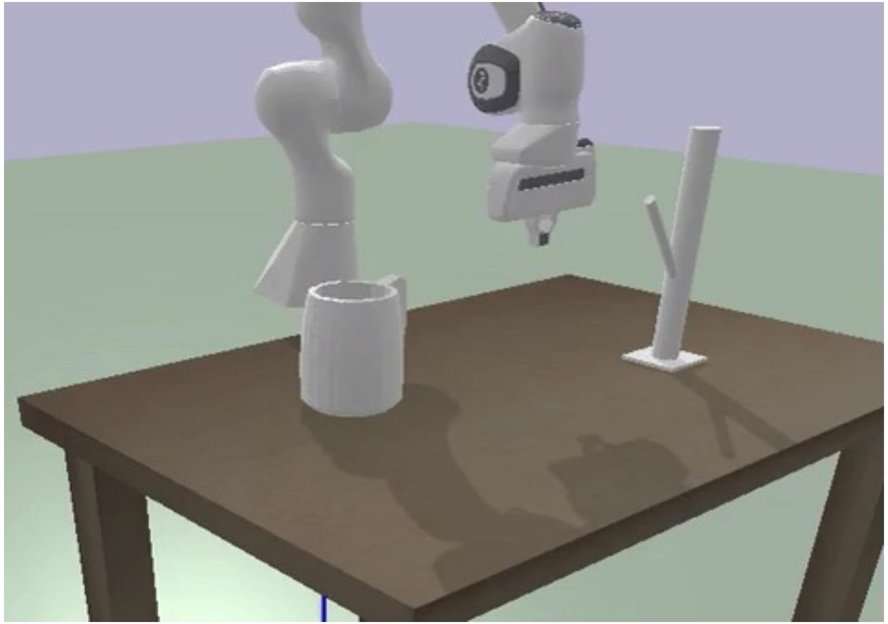
Baselines all require a fixed anchor in a known pose

Mug Hanging Task

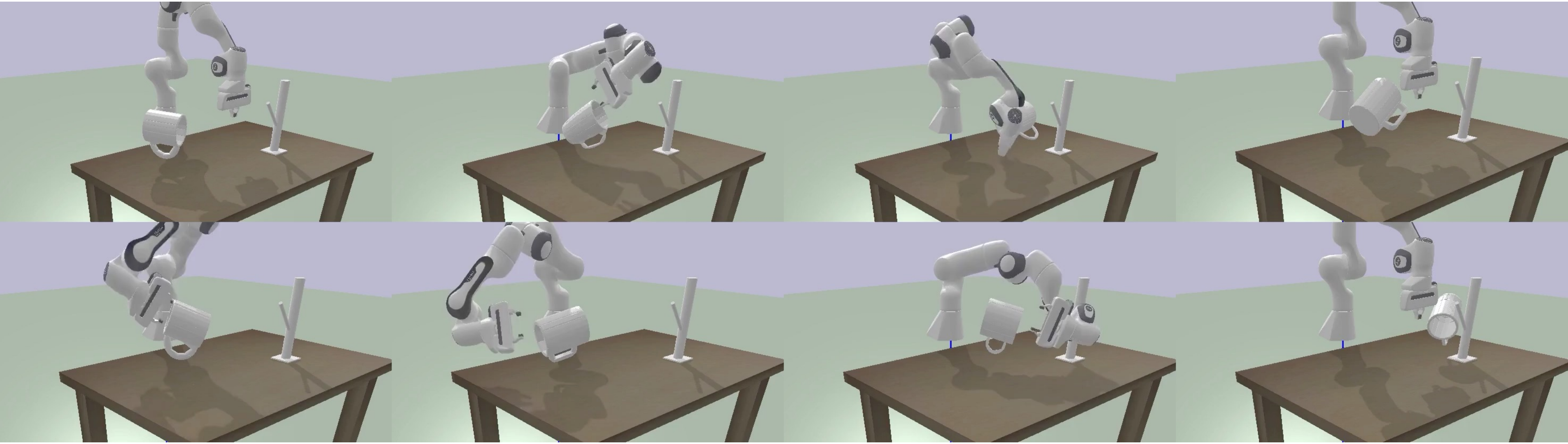
1) Cross-pose from **Gripper** to **Mug**

2) Cross-pose from **Mug** to **Rack**



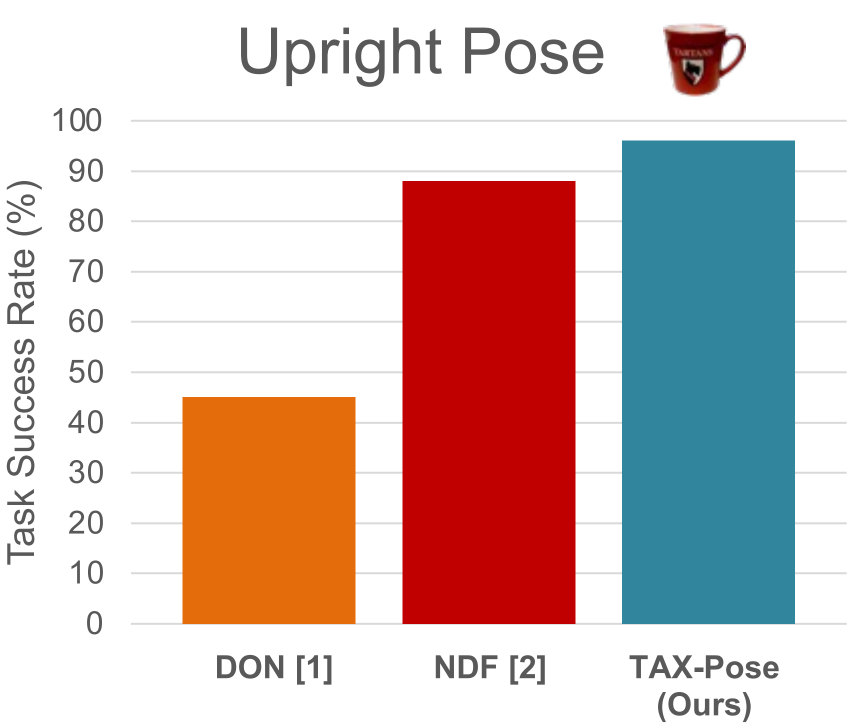


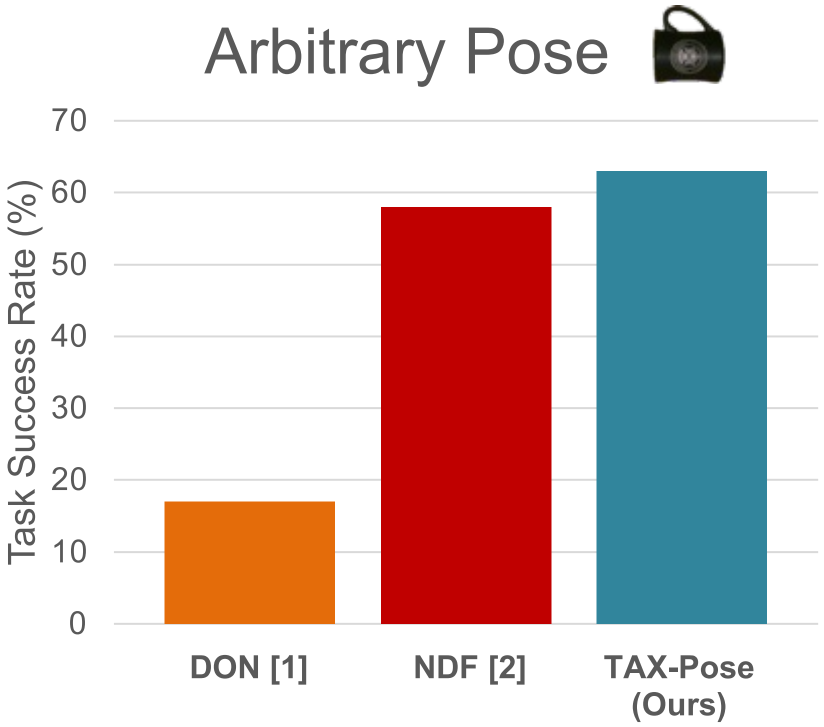
Mug Hanging Results



Our method generalizes to unseen mug instances and poses using only a few demonstrations.

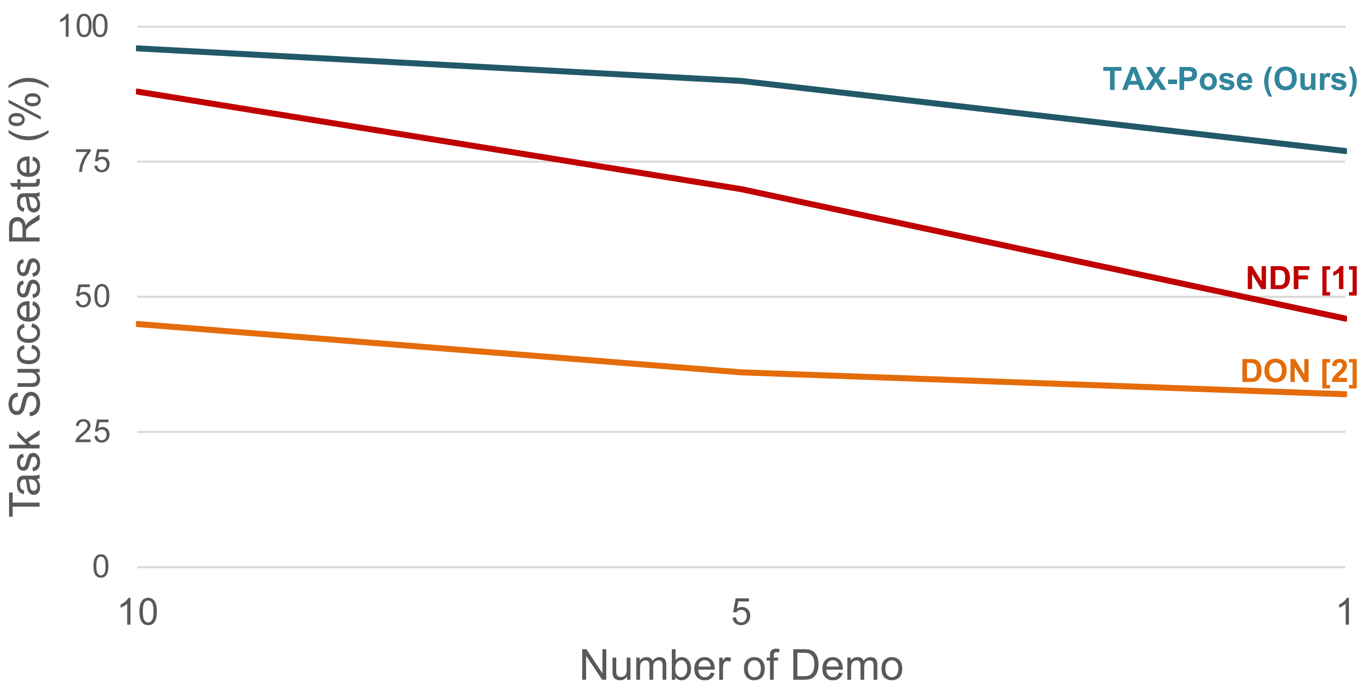
Mug Hanging Results





[1] Florence, Peter R., Lucas Manuelli, and Russ Tedrake. "Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation." *CORL 2018*
[2] Simeonov, Anthony, et al. "Neural descriptor fields: Se (3)-equivariant object representations for manipulation." I*CRA 2022*

Number of Demonstrations

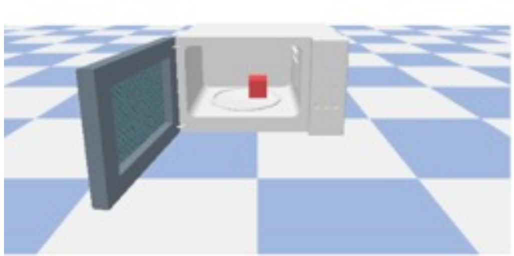


[1] Florence, Peter R., Lucas Manuelli, and Russ Tedrake. "Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation." *CORL 2018*
[2] Simeonov, Anthony, et al. "Neural descriptor fields: Se (3)-equivariant object representations for manipulation." I*CRA 2022*

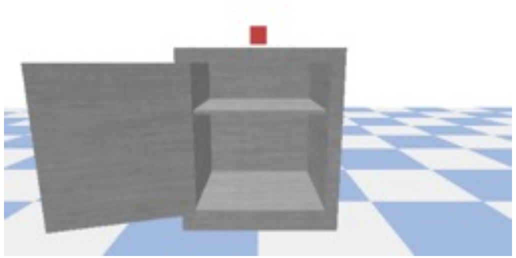
Objects Placement Task

Trained on **4** semantic goals (input to the network)

Inside



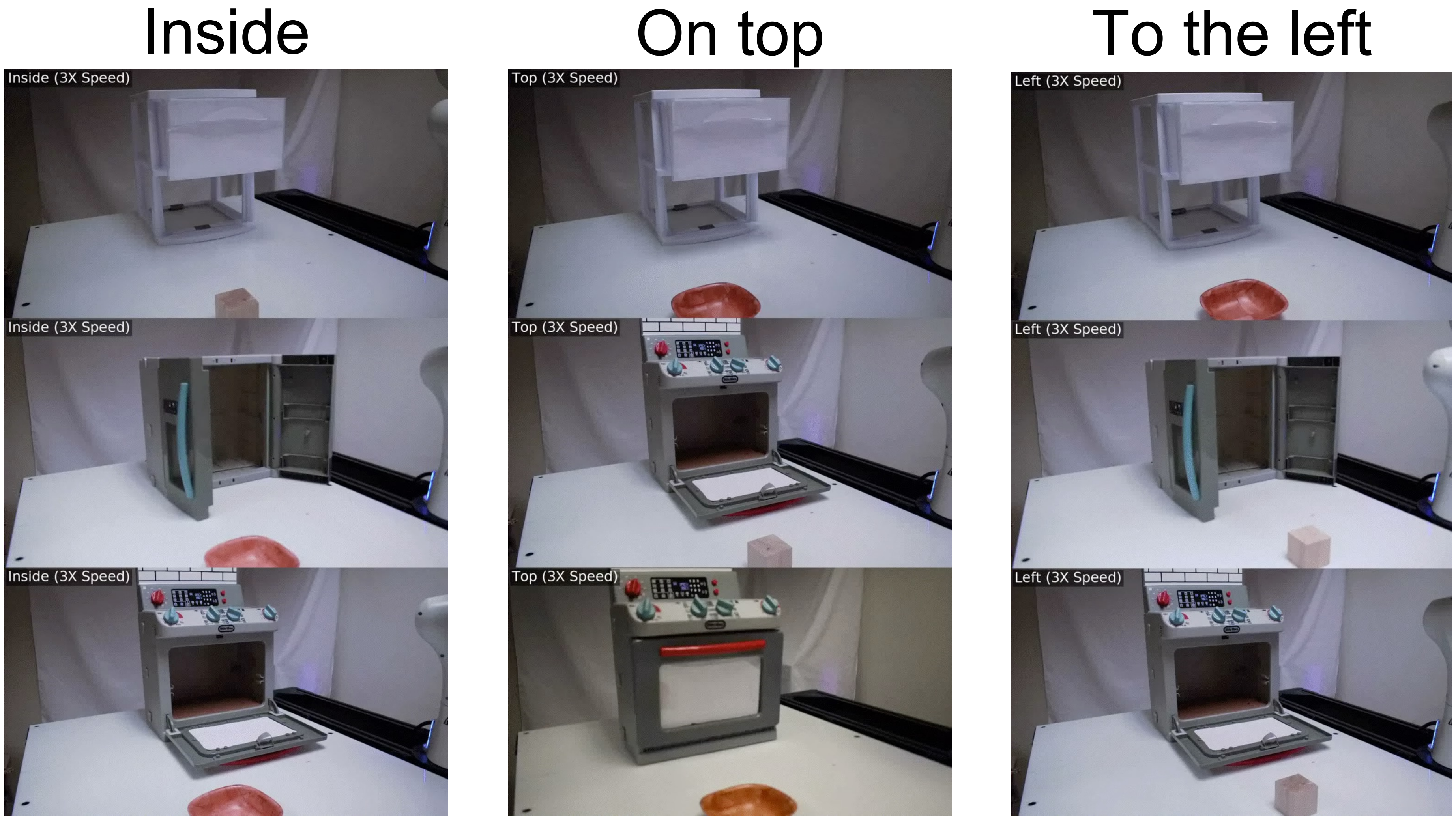
On Top



To the Left / To the Right

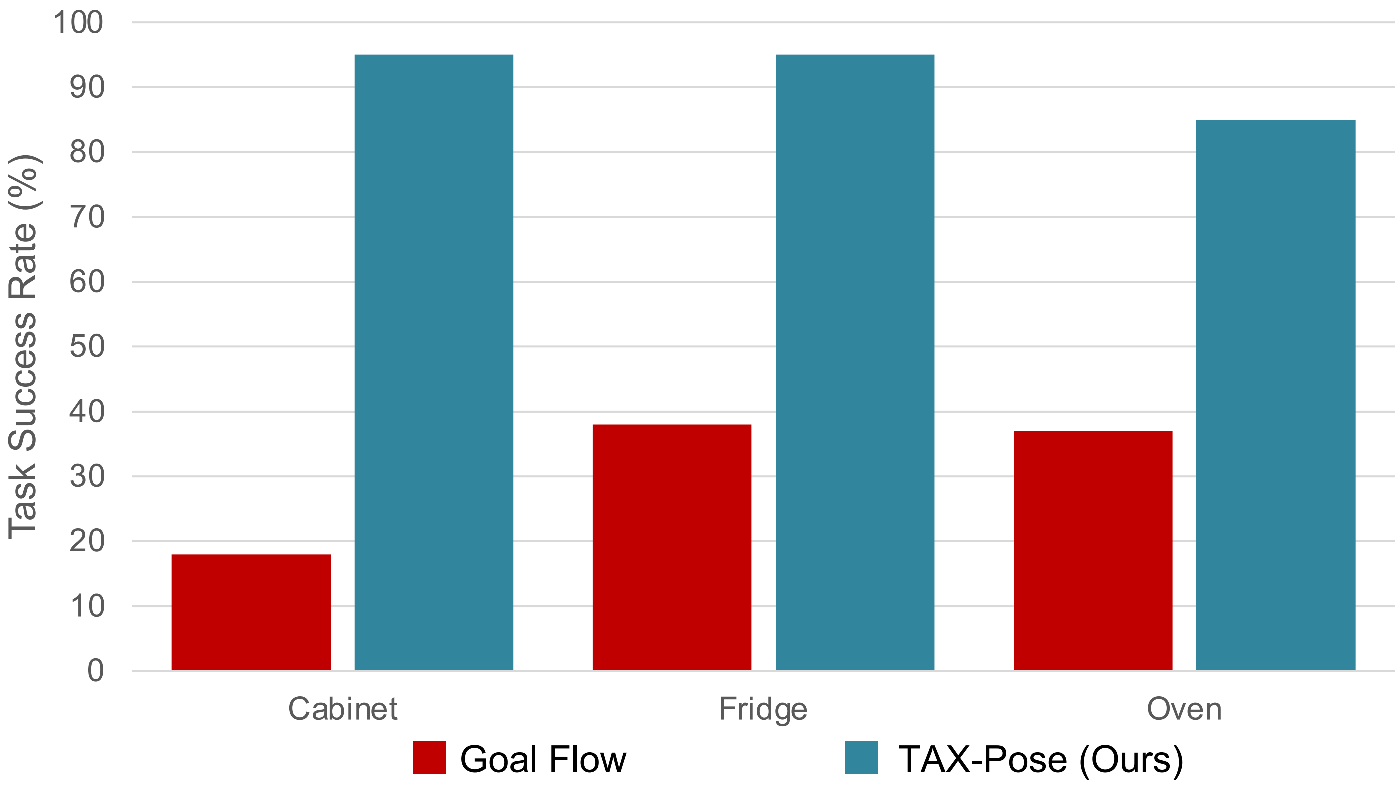
Trained **a single model** on PartNet Mobility [1] across **8** object classes in simulation

[1] Xiang, Fanbo, et al. "Sapien: A simulated part-based interactive environment." *CVPR 2020*



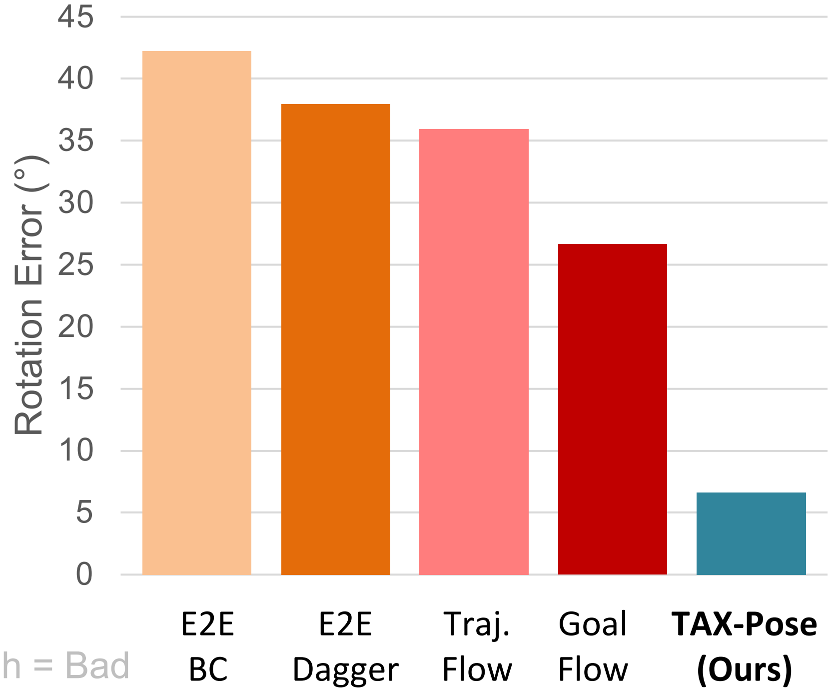
Inside
On top
To the left
Inside (3X Speed)
Top (3X Speed)
Left (3X Speed)
Inside (3X Speed)
Top (3X Speed)
Left (3X Speed)
Inside (3X Speed)
Top (3X Speed)
Left (3X Speed)

Object Placement: Real-world

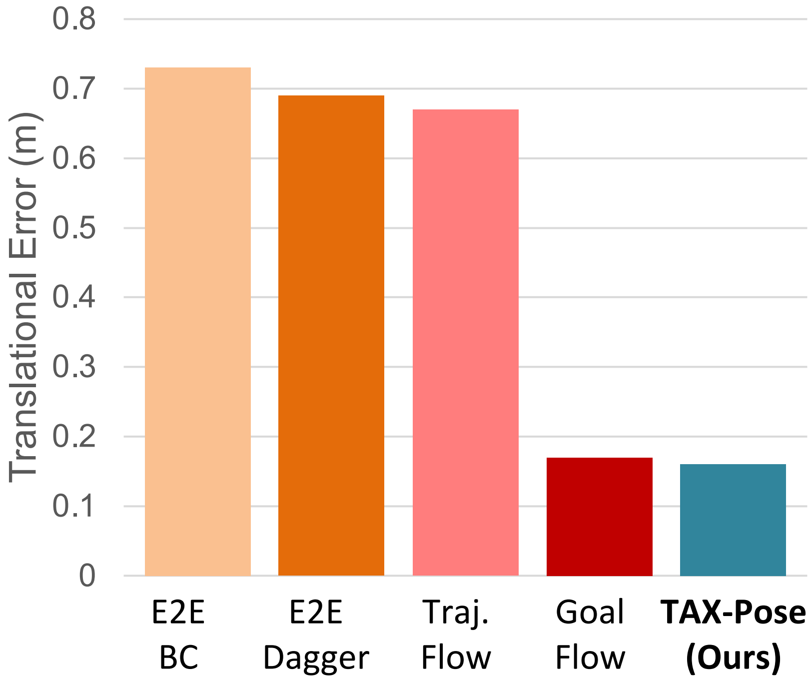


Object Placement: Simulation

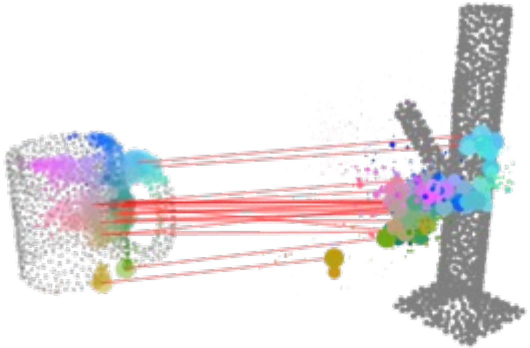
(Averaged over all objects)

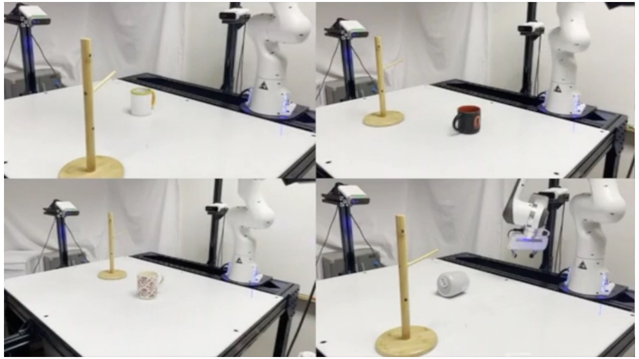


High = Bad



Take Aways



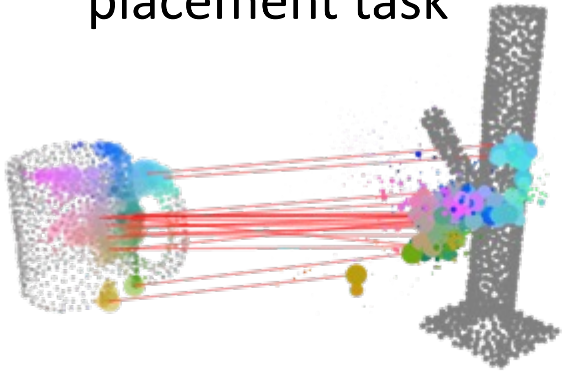


- Estimate the “cross-pose” needed to complete the relative placement task
- Using cross-object correspondences allows the network to learn about object interactions
- Learned importance weights let the network focus on the parts that matter
- Differentiable Weighted SVD to convert to a rigid transform

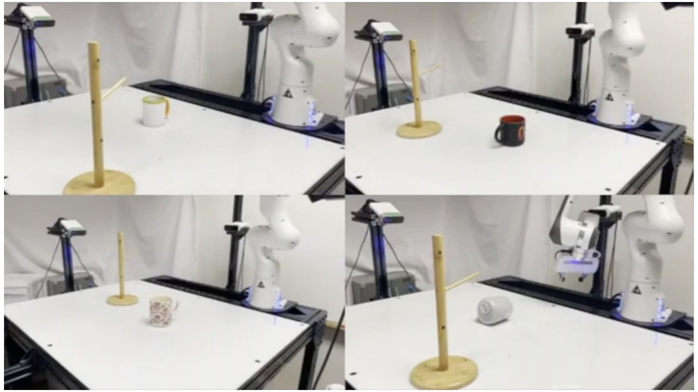
Relational Affordance Learning

Sensor inputs → Perception System → ? → Planning → Actions

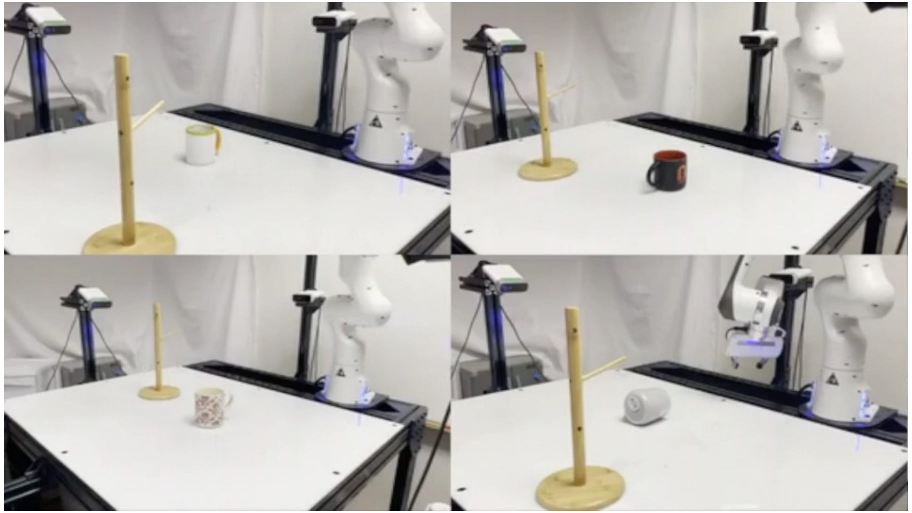
Estimating the **cross-object relationship** for the relative placement task



Motion planning to achieve this desired cross-pose

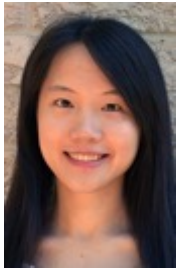


How can robots learn a task from just a few real-world demonstrations and generalize to new objects and new configurations?



Task-specific Relative Pose Estimation
(CoRL 2022)

Brian Okorn



Chuer Pan

Ben Eisner

Harry Zhang

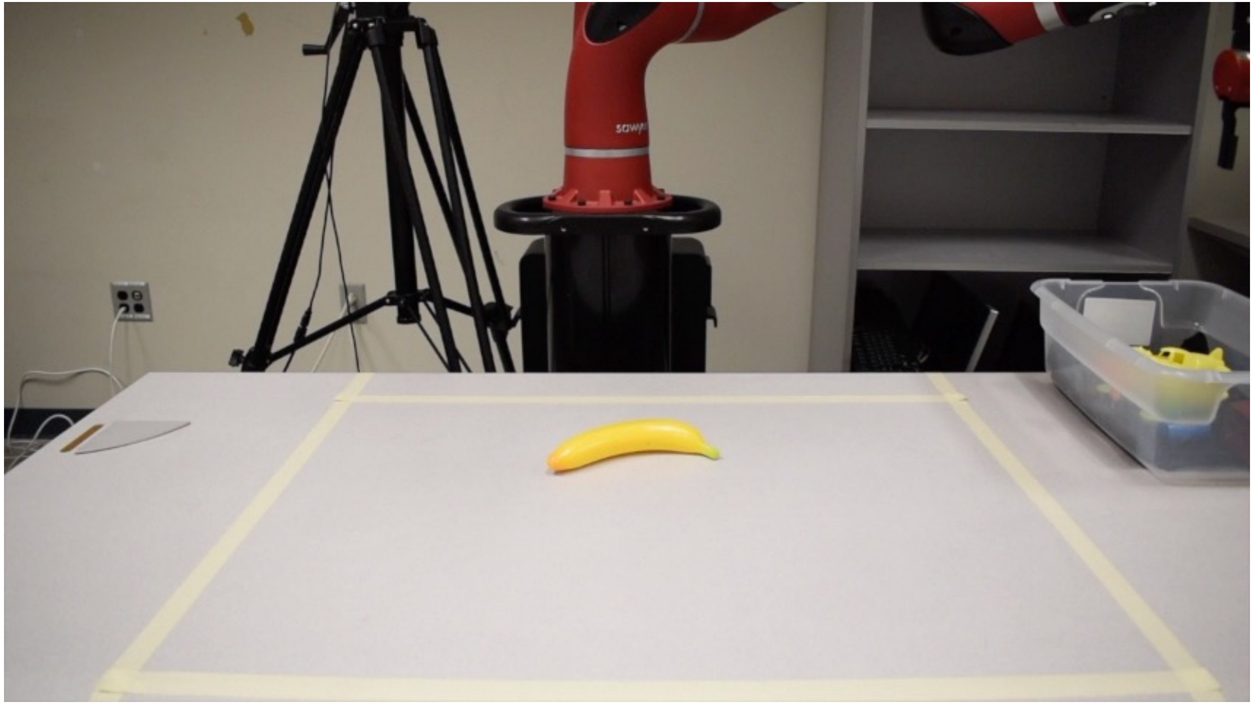
Perceptual Robot Learning

Grasping Transparent and Reflective Objects
(ICRA 2020)

Thomas
Weng

Imperfect Perception System → ? → Planning / Control

State-of-the-art grasping methods work well on opaque objects

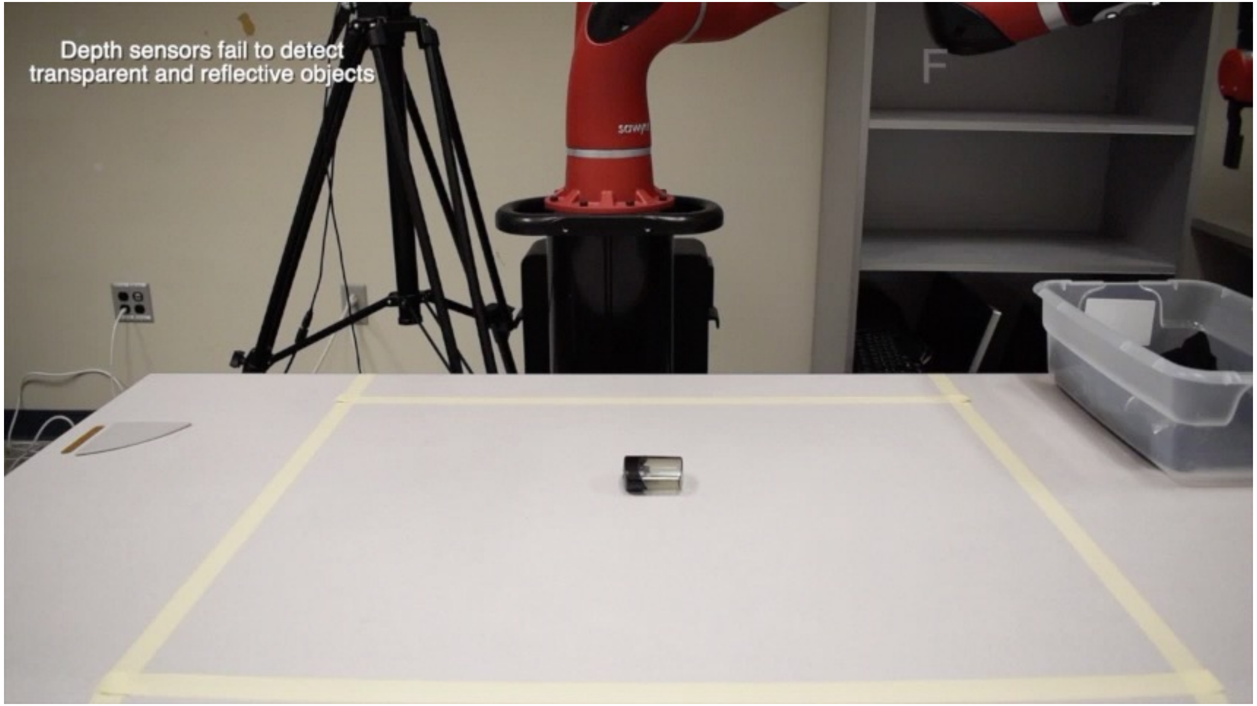


1. Mahler, Jeffrey, et al. "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics." Robotics: Science and Systems (2017).
2. Morrison, Douglas et al. "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach." *Robotics: Science and Systems* (2018).

This is because they use depth images to infer object geometry

But depth sensors have difficulty estimating the depth of transparent and specular objects

State-of-the-art grasping methods fail on transparent and reflective objects

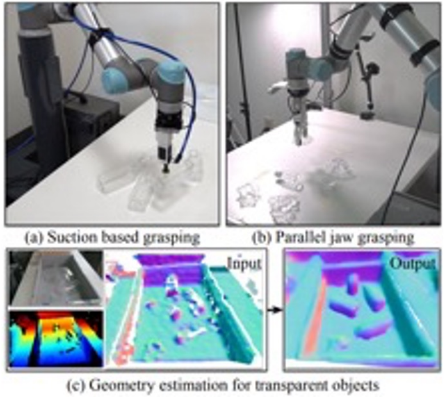


Previous methods:

~800k real-world grasp attempts

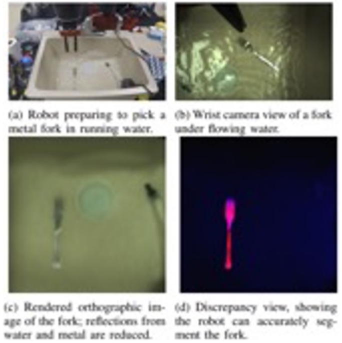
[Levine et al. 2018]

High-fidelity simulator



[Sajjan et al. 2019]

~140 viewpoints per test grasp



[Oberlin et al. 2018]

Poor generalization

Slow

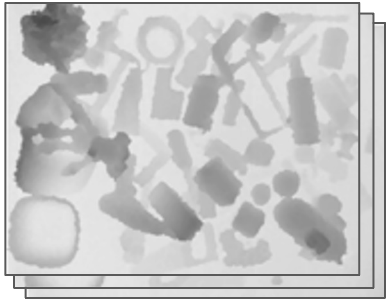
1. Saxena, Ashutosh et al. "Robotic grasping of novel objects using vision." The International Journal of Robotics Research 27.2 (2008): 157-173.
2. Oberlin, John, and Stefanie Tellex. "Time-Lapse Light Field Photography for Perceiving Transparent and Reflective Objects." *Robotics: Science and Systems* (2018).
3. Levine, Sergey, et al. "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection." *The International Journal of Robotics Research* 37.4-5 (2018): 421-436.

Our method does not require:

- Any real grasp attempts
- Human labeling
- Specialized hardware
- Simulation of transparent objects
- Multiple viewpoints

Our method requires only:

A dataset of 250 paired RGB-D images of opaque objects

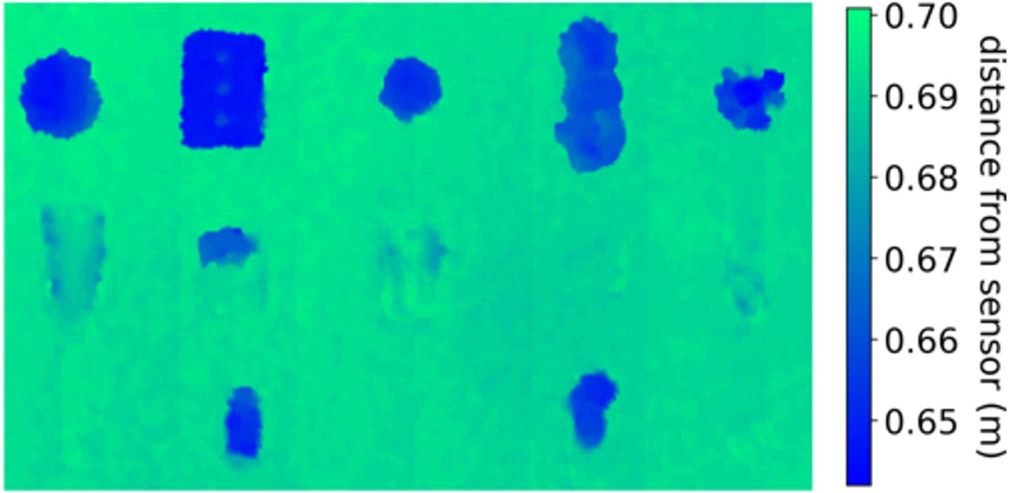


A depth-based network that predicts a grasp score for different possible grasps of opaque objects

1. Satish, Vishal, et al. "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks." *IEEE Robotics and Automation Letters* 4.2 (2019): 1357-1364.

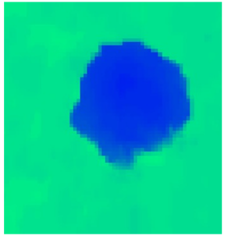
Insight: RGB is a better modality than depth for perceiving transparent and specular objects

RGB image



Depth image

To train, we input images of opaque objects

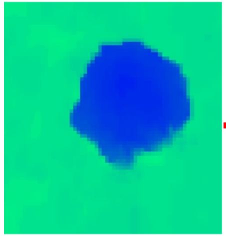


Depth Image

RGB Image

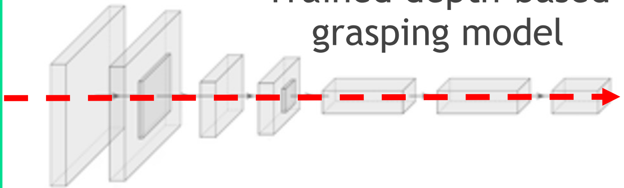
1. Gupta, Saurabh et al. "Cross modal distillation for supervision transfer." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016).
2. Satish, Vishal, et al. "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks." *IEEE Robotics and Automation Letters* 4.2 (2019): 1357-1364.

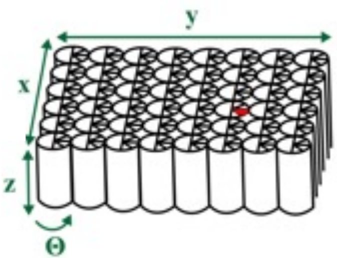
We use the pre-trained depth grasping network to output a dense set of grasp scores



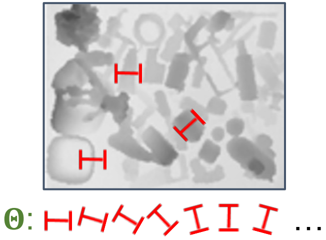
Depth Image

Trained depth-based grasping model





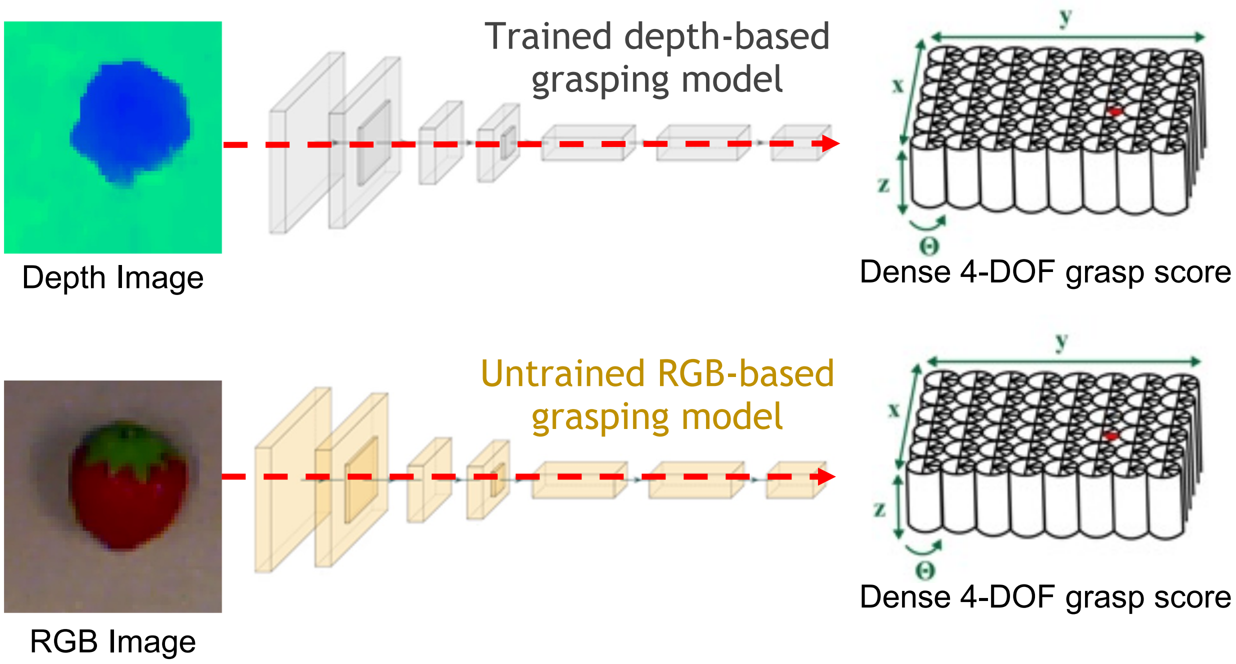
Dense 4-DOF grasp score



RGB Image

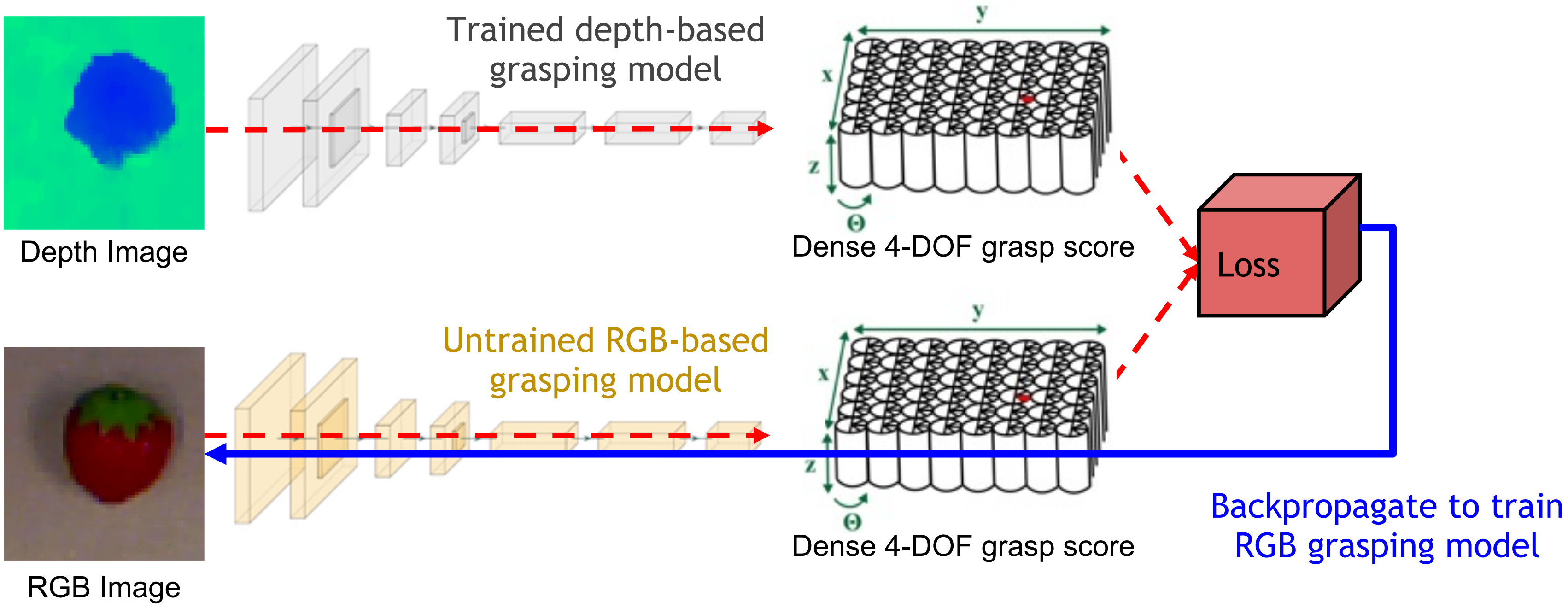
1. Gupta, Saurabh et al. "Cross modal distillation for supervision transfer." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016).
2. Satish, Vishal, et al. "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks." *IEEE Robotics and Automation Letters* 4.2 (2019): 1357-1364.

To train an RGB grasping network…



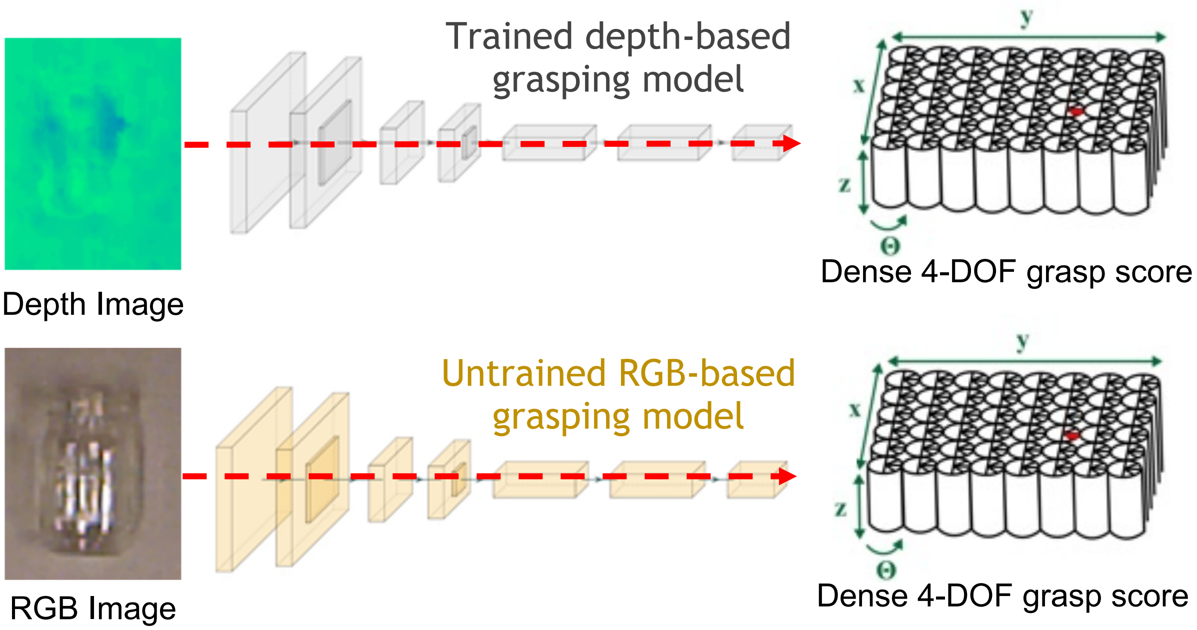
1. Gupta, Saurabh et al. "Cross modal distillation for supervision transfer." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016).
2. Satish, Vishal, et al. "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks." *IEEE Robotics and Automation Letters* 4.2 (2019): 1357-1364.

We supervise the RGB network to match the grasping scores of the pre-trained depth network



1. Gupta, Saurabh et al. "Cross modal distillation for supervision transfer." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016).
2. Satish, Vishal, et al. "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks." *IEEE Robotics and Automation Letters* 4.2 (2019): 1357-1364.

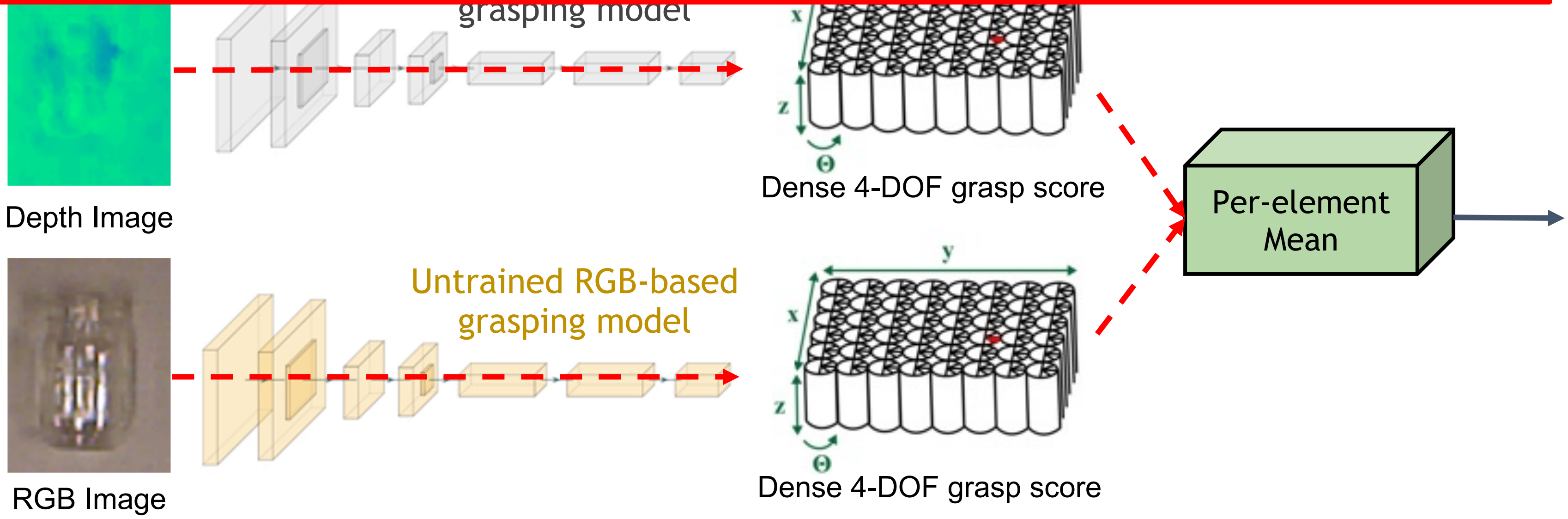
At test time, we input a transparent or specular object into both networks



1. Gupta, Saurabh et al. "Cross modal distillation for supervision transfer." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016).
2. Satish, Vishal, et al. "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks." *IEEE Robotics and Automation Letters* 4.2 (2019): 1357-1364.

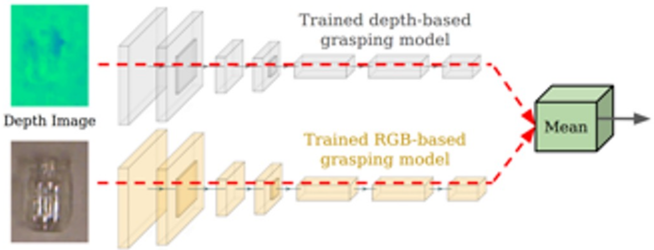
… and average the output (very simple!)

We originally planned to try a more complicated method, but this simple one worked as well as we could have hoped!



1. Gupta, Saurabh et al. "Cross modal distillation for supervision transfer." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016).
2. Satish, Vishal, et al. "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks." *IEEE Robotics and Automation Letters* 4.2 (2019): 1357-1364.

Why does our method work?



1. The depth-based grasping network outputs a reasonable grasp prediction for opaque objects during training
2. The RGB grasping network learns to imitate the depth network on opaque objects
3. The RGB network is able to generalize this training to transparent and reflective objects, which appear somewhat similar to the opaque objects that were used for training (when viewed in RGB)
4. Combining the output of both RGB and depth networks may help our method to be robust to changes in visual texture like background or lighting variations.

1. Gupta, Saurabh et al. "Cross modal distillation for supervision transfer." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016).
2. Satish, Vishal, et al. "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks." *IEEE Robotics and Automation Letters* 4.2 (2019): 1357-1364.

Train objects:

Opaque

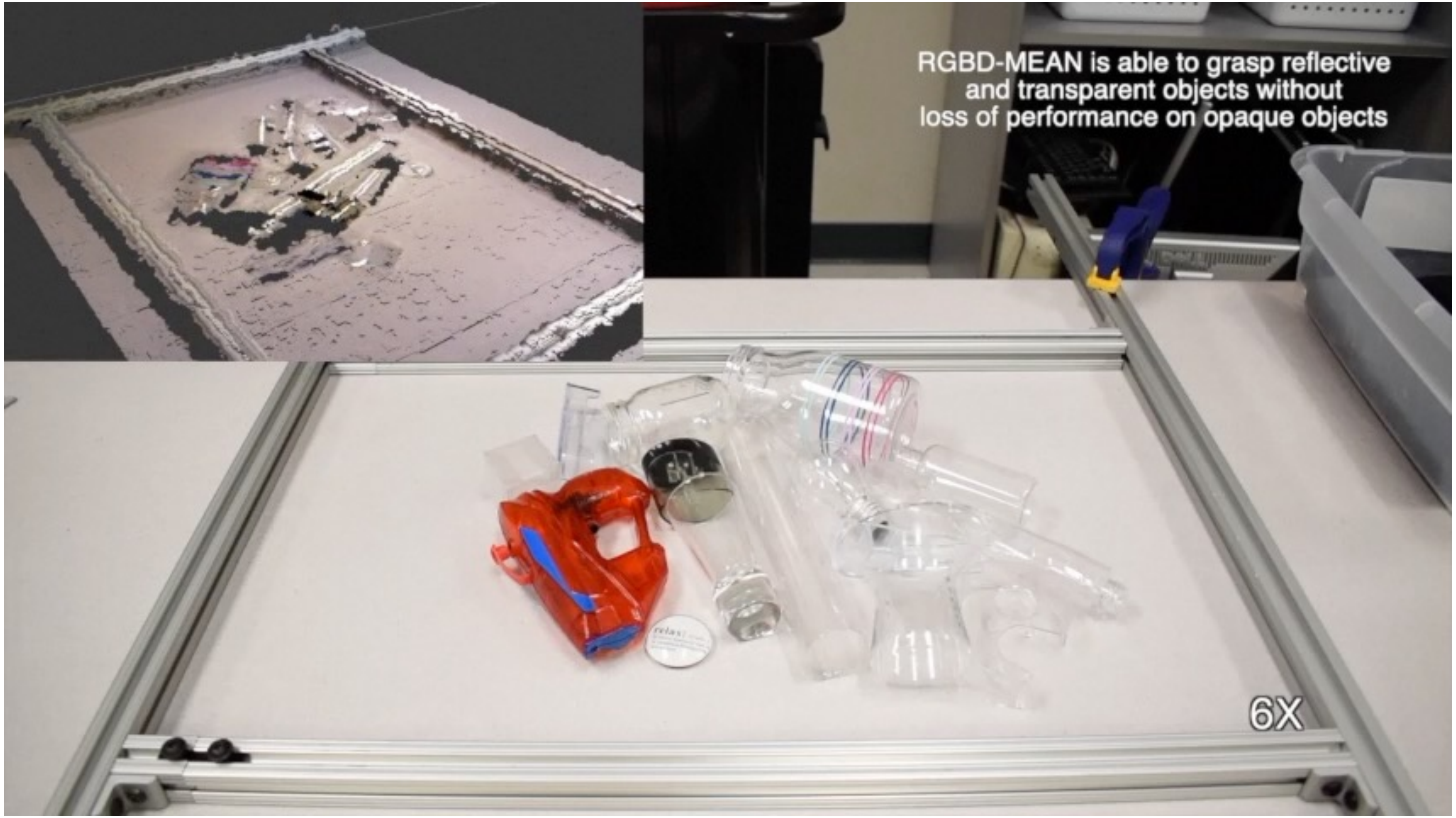
Test objects:

Opaque

Transparent

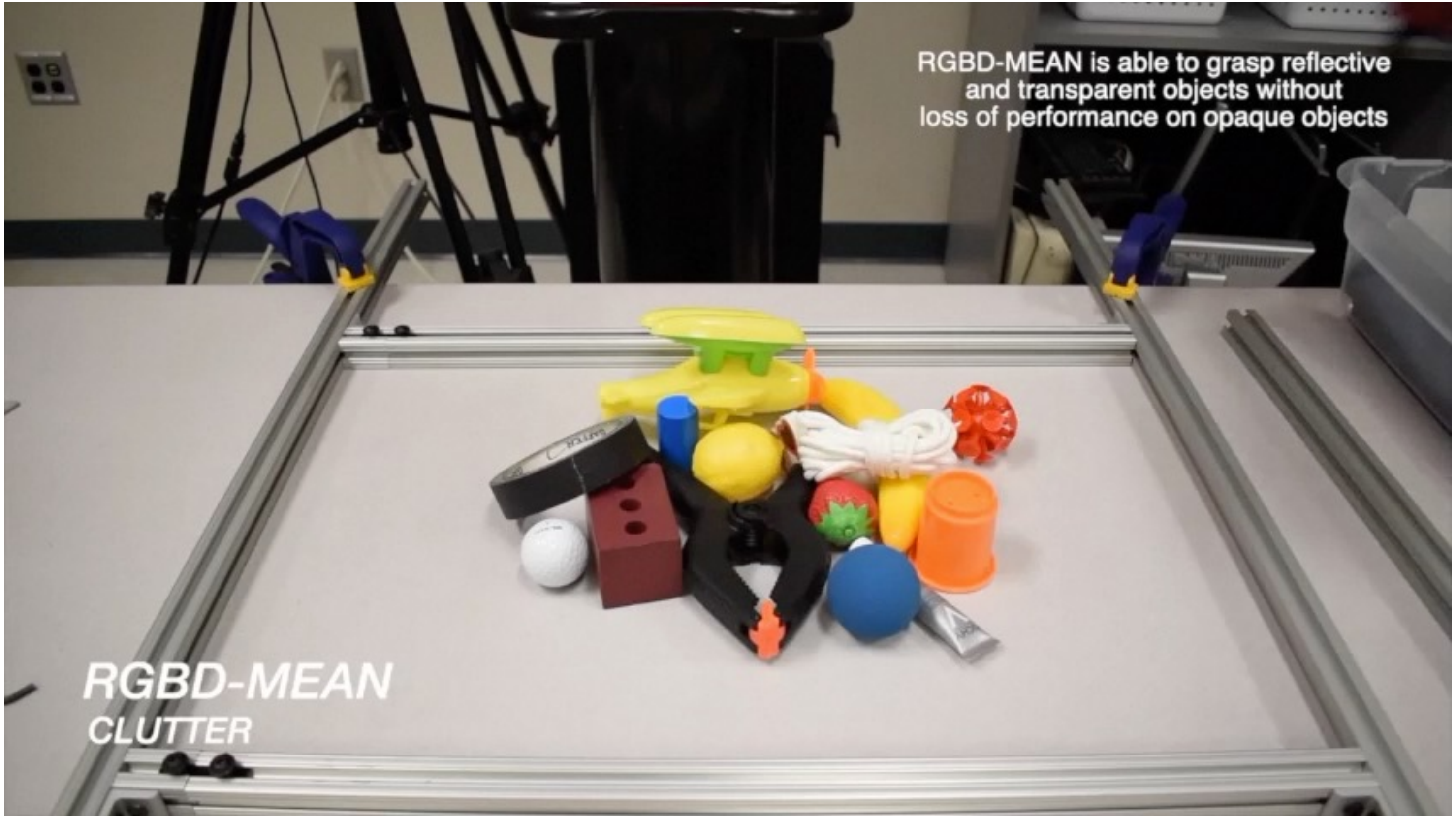
Specular

Our method is able to grasp transparent objects

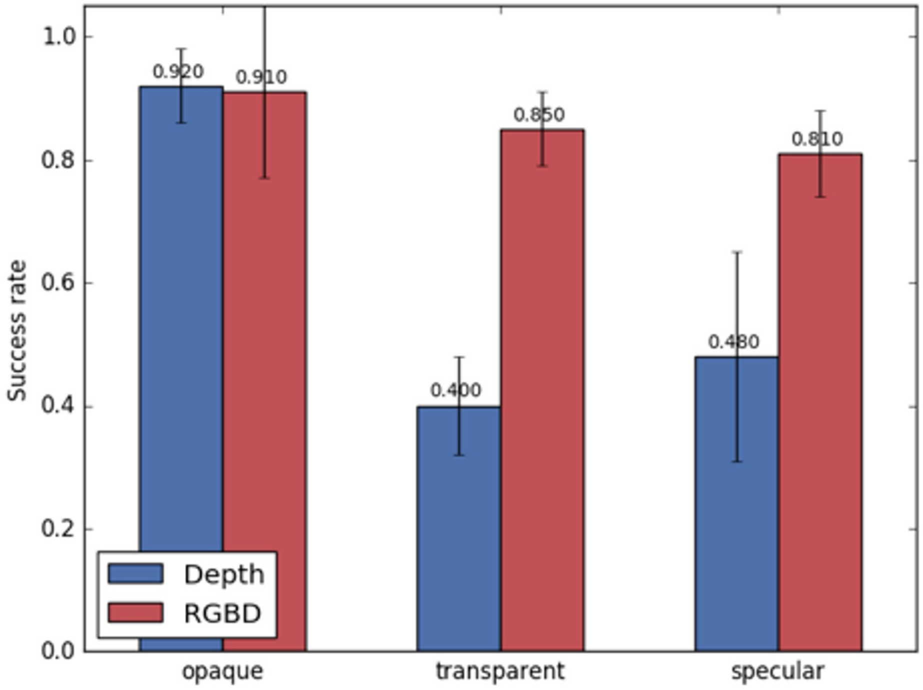


Our method is able to grasp reflective objects

Our method is able to grasp opaque objects

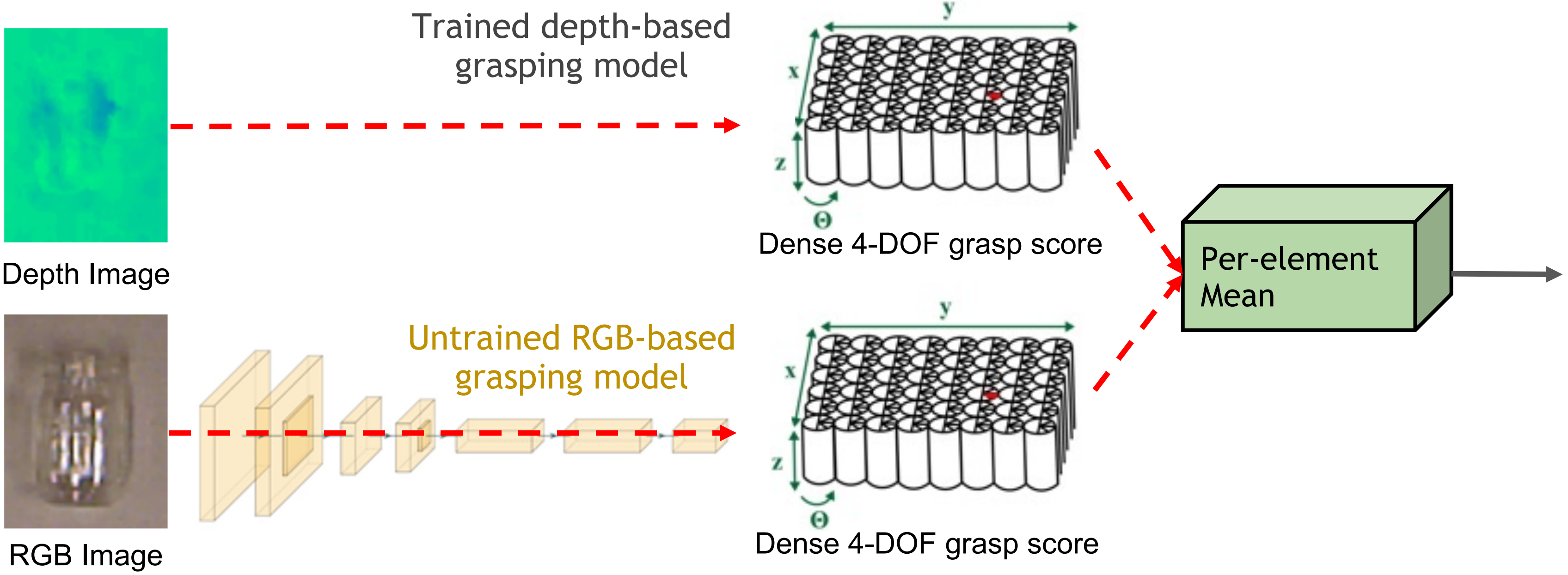


Our method outperforms the depth-only network for transparent and specular objects



Summary

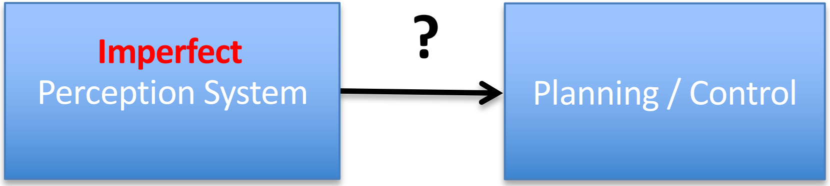
- Train RGB network to imitate depth network on **opaque** objects
- Combine depth and RGB networks for grasping **opaque, transparent, and reflective objects**



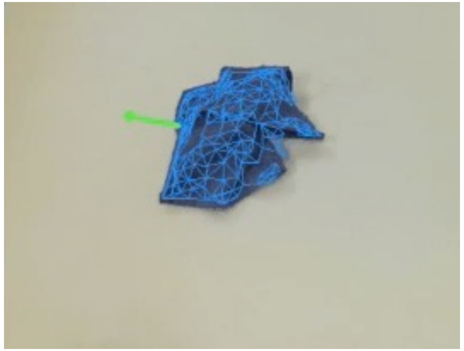
Perceptual Robot Learning

Grasping Transparent and Reflective Objects
(ICRA 2020)

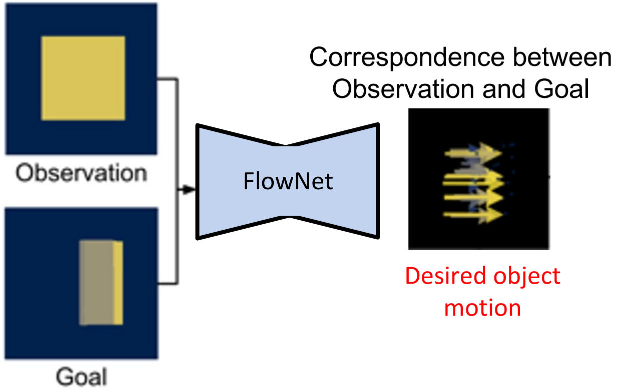
Thomas
Weng



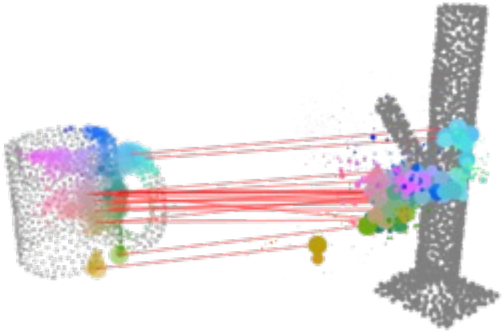
How do we bridge the gap between perception and planning?



Use a graph to reason about the relationship between object parts



Reason explicitly about the relationship between the current state and the goal



Reason explicitly about the cross-object relationship for relative placement tasks

Reasoning about **relationships**