Learning Object Agnostic Task Representations for Object Manipulation

(prepared for NeurIPS 2020)

Yunchu Zhang under Prof.Katerina Fragkiadaki

Outline

- ❑ **Motivation**
- ❑ Proposed Approach
- ❑ Progress & Next Steps

Motivation

Humans are good at reusing previous knowledge to transfer skills

Motivation

Efficiently learn *object-agnostic* task representations

Motivation

Efficiently learn *object-agnostic* task representations

Grasp an object from its rim

Grasp an object with its handle

Stir the contents of the container

Skills should be easily transferable within the same category of objects irrespective of shape and size

Motivation

Efficiently learn *object-agnostic* task representations

- Accelerates acquisition of new skills by reusing previous experience
- Reduces the number of training samples
- Skills are invariant to object shape and size; leads to better generalization

Challenges

Reinforcement Learning

- Task-specific RL can achieve impressively dexterous skills for a specific task [1][2][3]
- Does not generalize well to multiple tasks
- Model-free RL algorithms tend to be sample inefficient

Control Theory

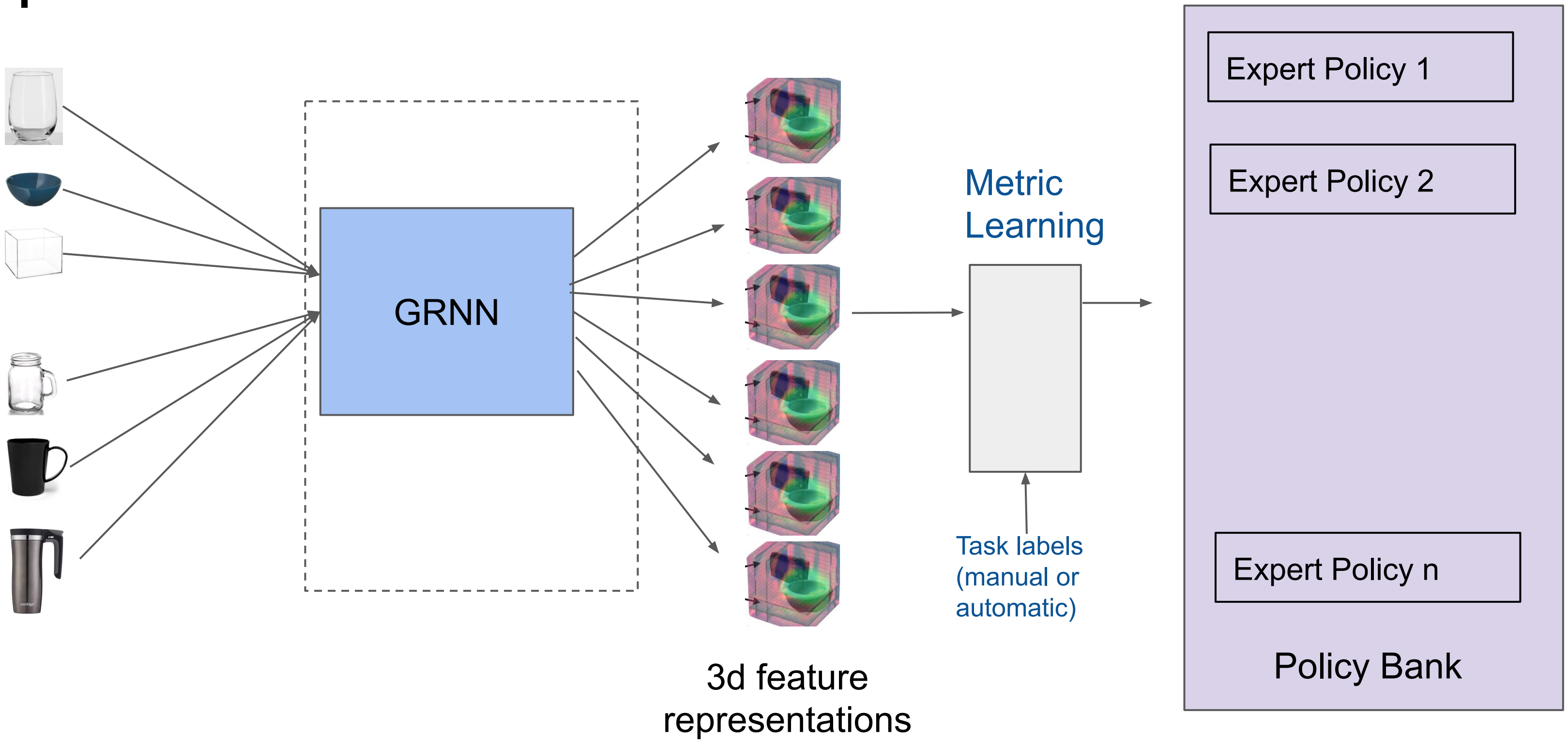
- Control methods achieve good performance on environments with known dynamics and object models [4][5]
- Requires significant human modelling effort
- Works in constrained environment

[1] S. Levine, C. Finn, T. Darrell, and P. Abbeel. “End-to-end training of deep visuomotor policies.” *IJMLR 2016*
[2] Gupta Abhishek et. al. “Learning Dexterous Manipulation for a Soft Robotic Hand from Human Demonstration.” *ICLR 2016*
[3] Rajeswaran, Aravind, et al. “Learning complex dexterous manipulation with deep reinforcement learning and demonstrations." *RSS 2018*
[4] Kaufmann, Elia, et al. “Beauty and the beast: Optimal methods meet learning for drone racing”. *ICRA 2019*
[5] Goh, Jonathan Y et al. "Toward automated vehicle control beyond the stability limits: drifting along a general path." *Journal of Dynamic Systems, Measurement, and Control* 2020.

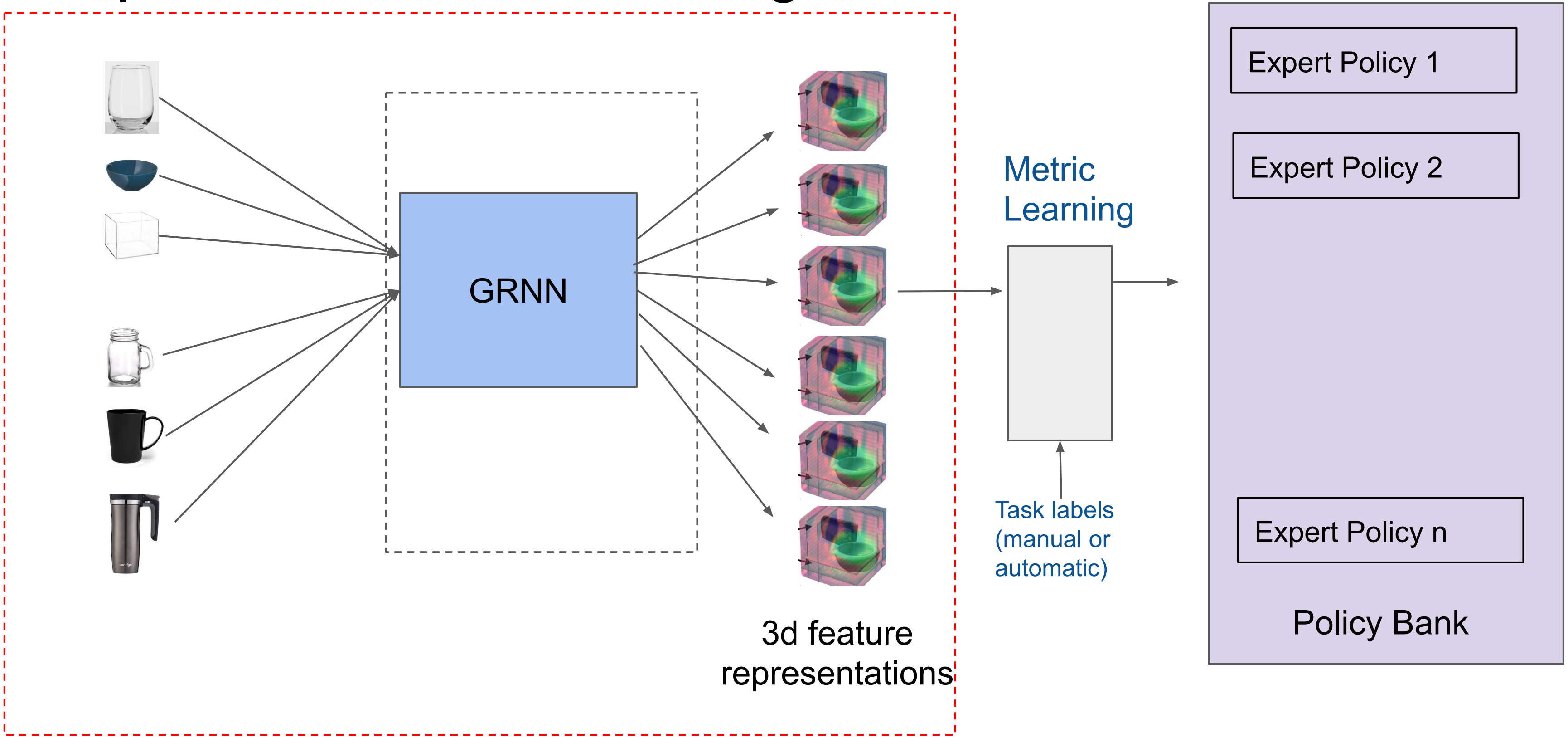
Outline

- ❏ Motivation
- ❏ **Proposed Model**
- ❏ Progress & Next Steps

Proposed Model



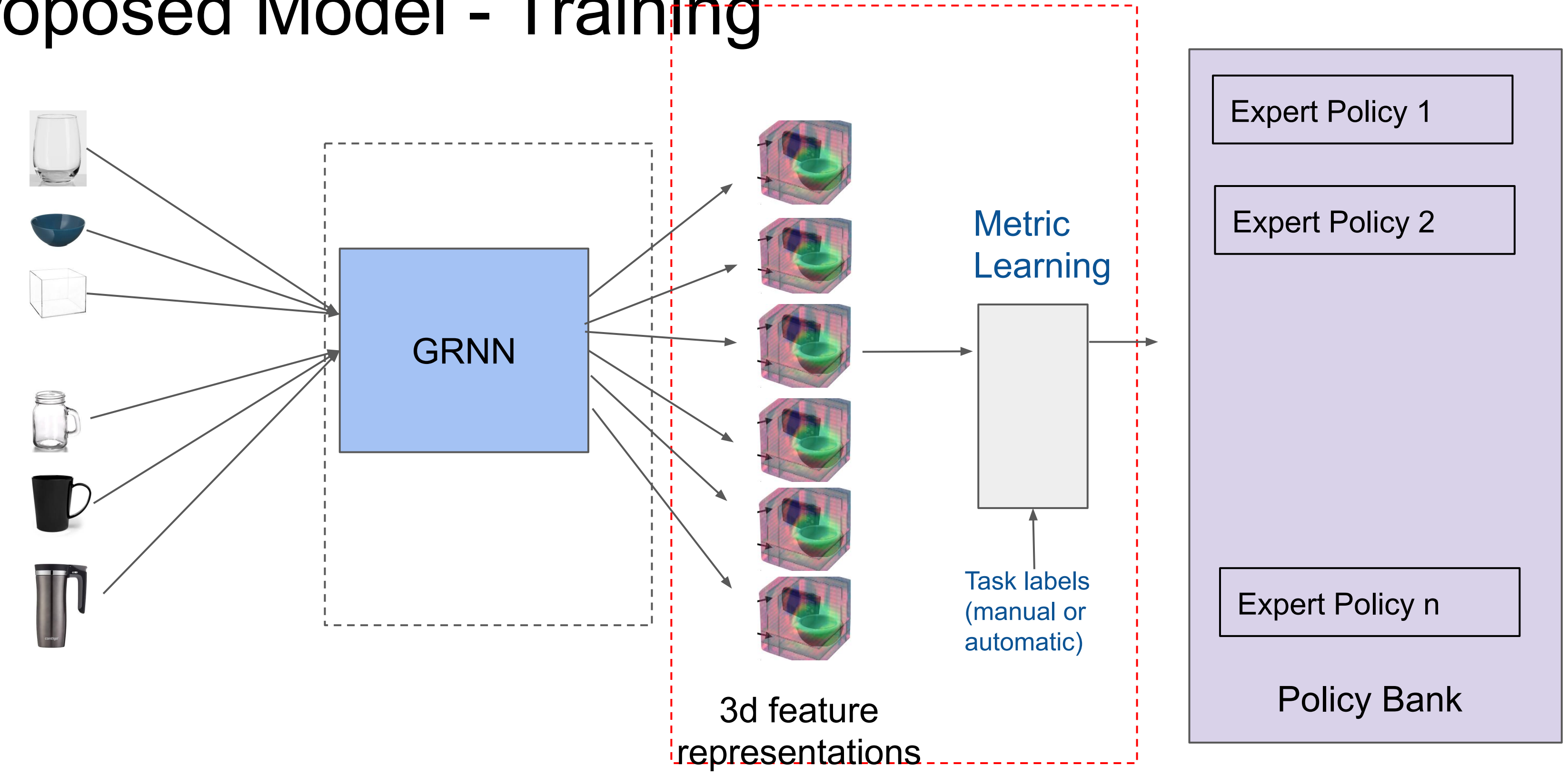
Proposed Model - Training



Proposed Model - Training 3d Feature maps

- Learn 3D feature maps using GRNN
  - Losses: View prediction and Occupancy
  - Data Collection:
    - Mujoco Fetch environment
    - RGBD images and projection matrices from 50 cameras around the object

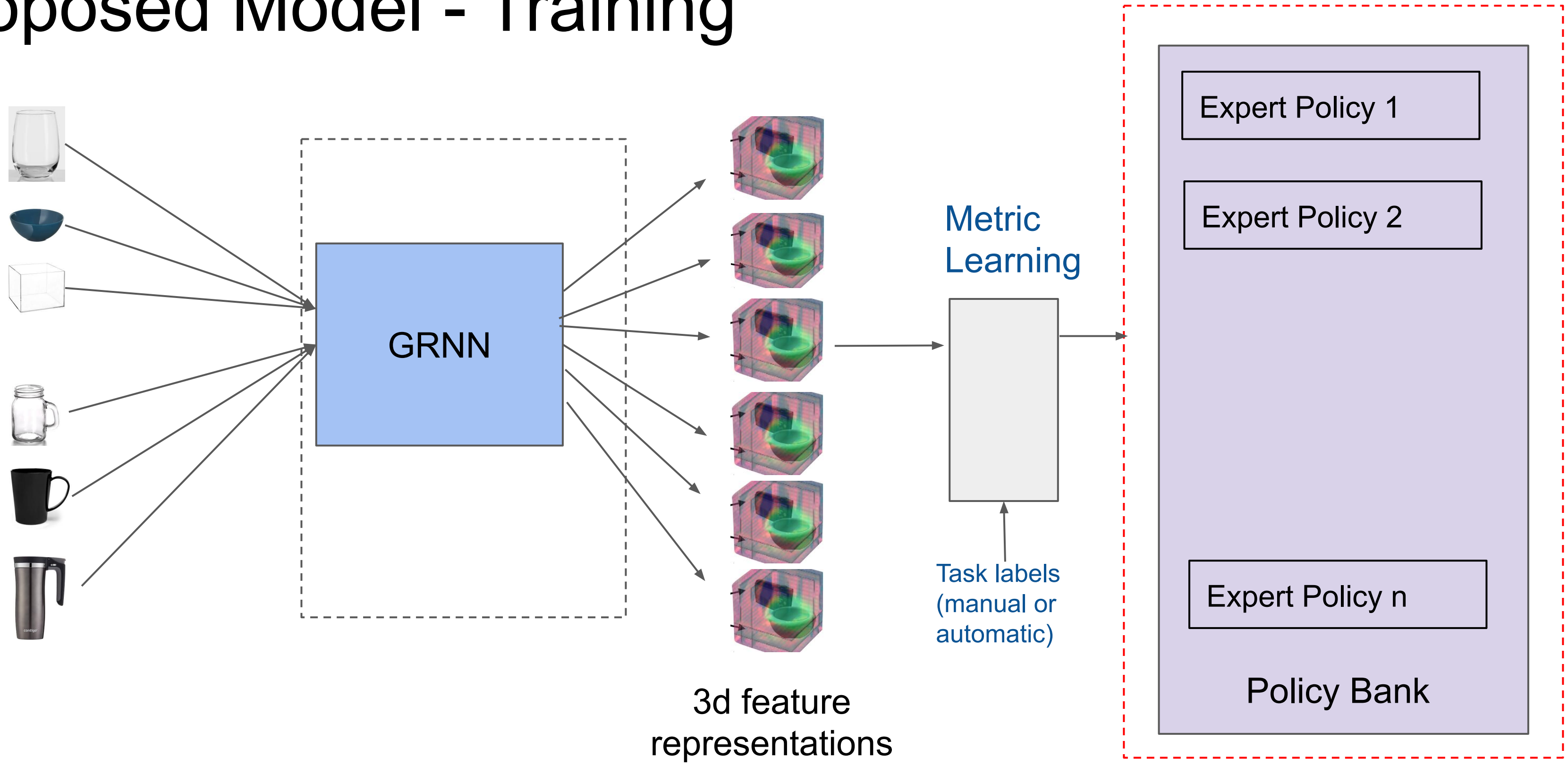
Proposed Model - Training



Proposed Model - Metric Learning

- Metric Learning with object-centric 3D feature maps as input
  - n-way classification loss $\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$
  - Learn feature mapping to bring objects belonging to same task together
- Class Representer
  - Naive: Compute the mean of all the object’s feature maps in a task
  - We assume presence of a trained expert/controller for every object class

Proposed Model - Training



Proposed Model - Training

Policy Compression

Expert Policy 1

Proposed Model - Training

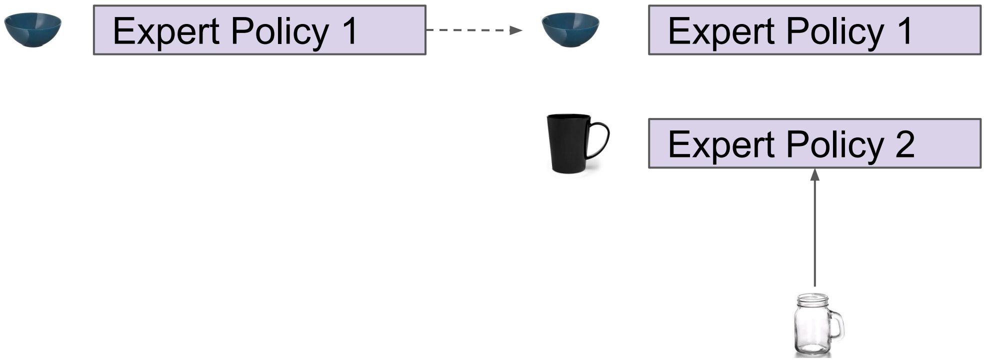
Policy Compression

Expert Policy 1

Check performance with different object orientations

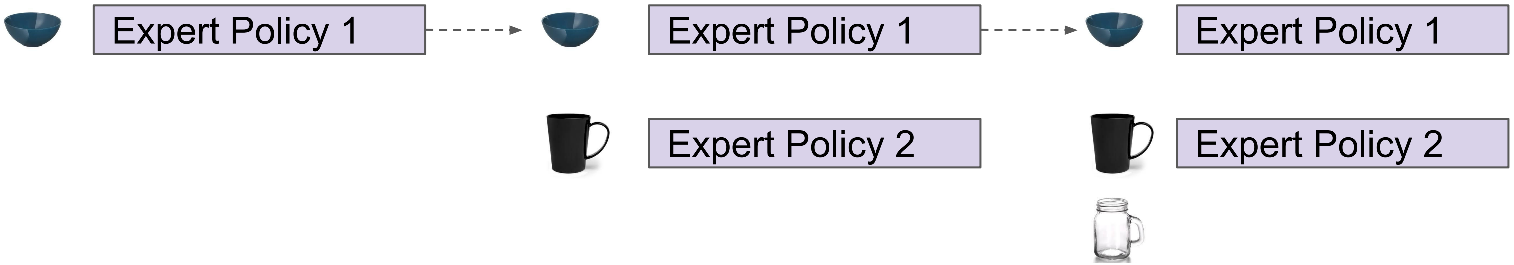
Proposed Model - Training

Policy Compression



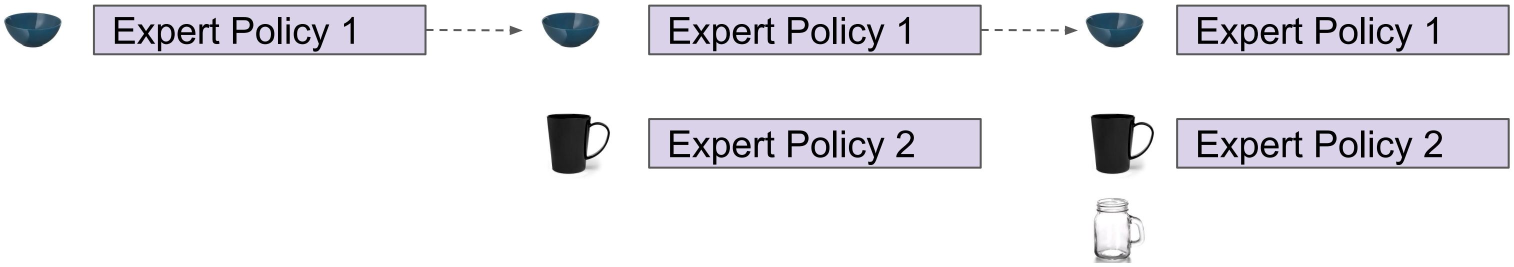
Proposed Model - Training

Policy Compression



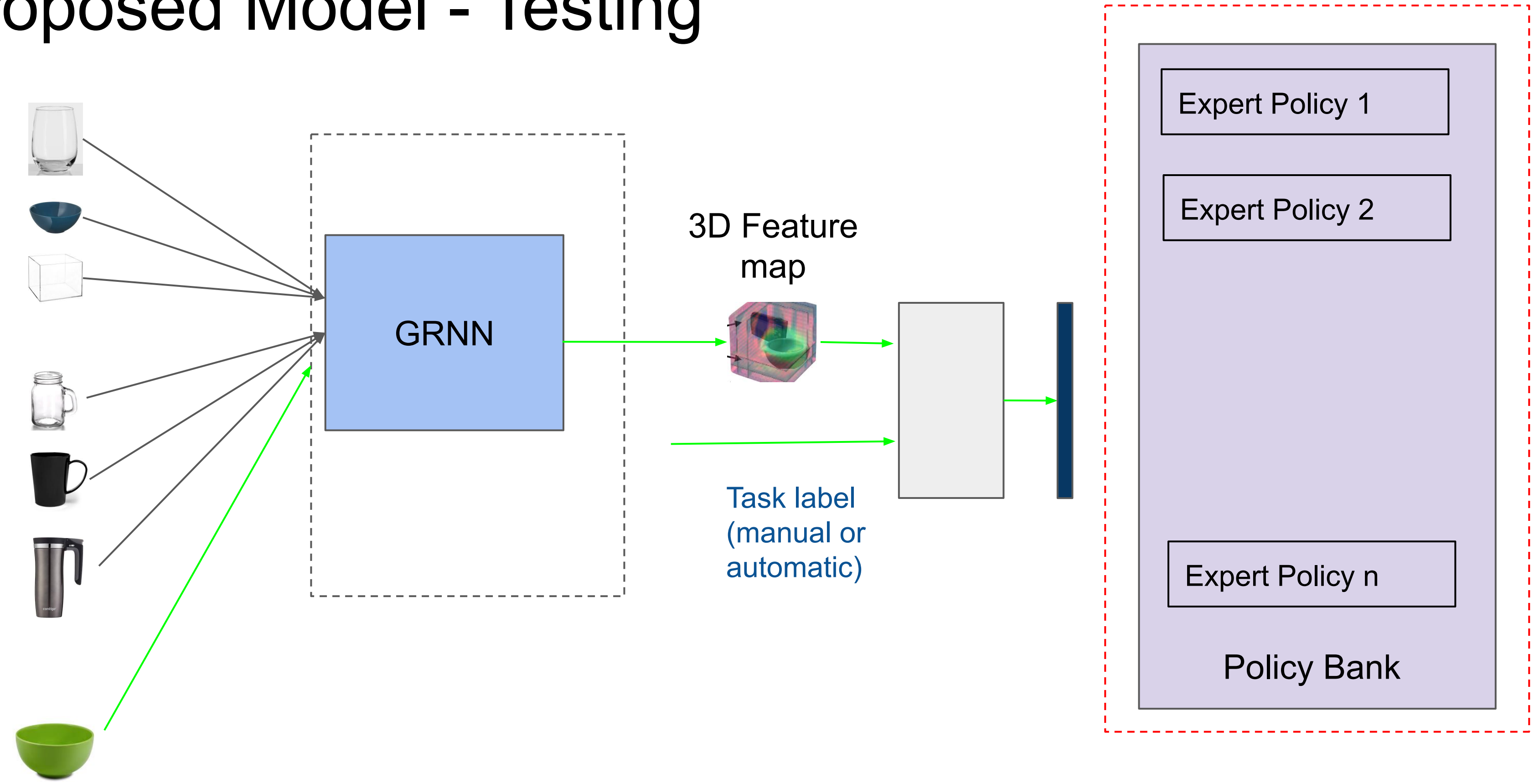
Proposed Model - Training

Policy Compression



- Amortized training time is expected to reduce with introduction of every new object
- Expert policy is trained with DDPG + HER
- Observation space: gripper position, object position, object relative to gripper, object rotation
- Action space: dx, dy, dz, gripper open/close

Proposed Model - Testing



Outline

- ❑ Motivation
- ❑ Proposed Method
- ❑ **Progress & Next Steps**

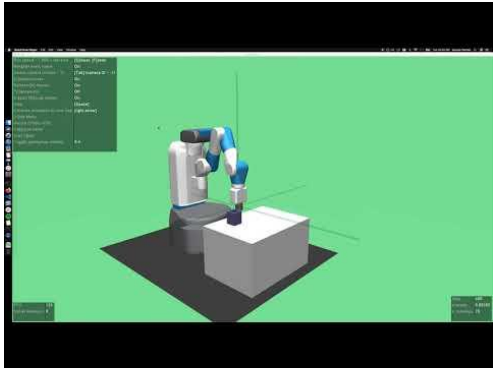
Progress

- Trained experts for grasping different cups from the rim
- Wrote a general purpose controller which can go to any given position and attain a particular orientation
- Trained 3D feature tensor for different types of cups (44 cups)
- Tested naive policy compression module (tested on 5 cups - 2 clusters)

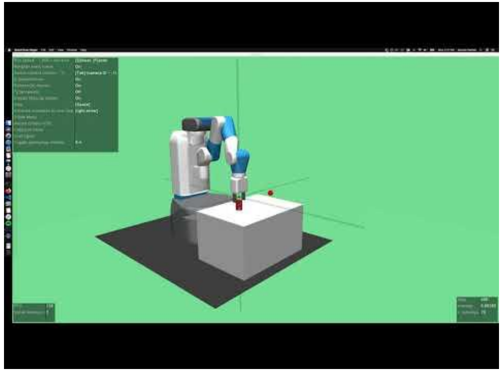
Next Steps:

- Use the 3D object tensor and do metric learning
- Check if the a similar object is mapped to correct cluster
- Use the policy of that cluster and check if we get good enough success

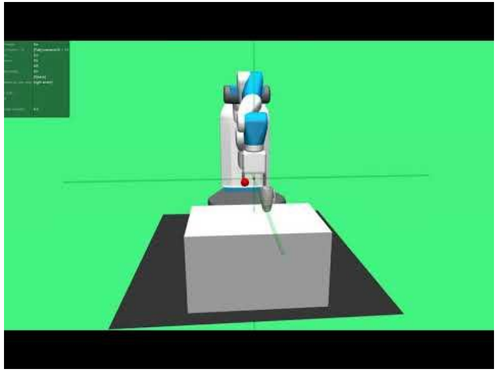
General Purpose Controller Demo



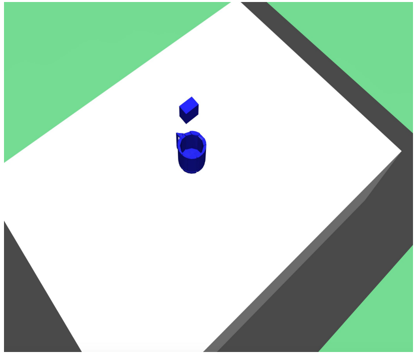
Pen in cup controller



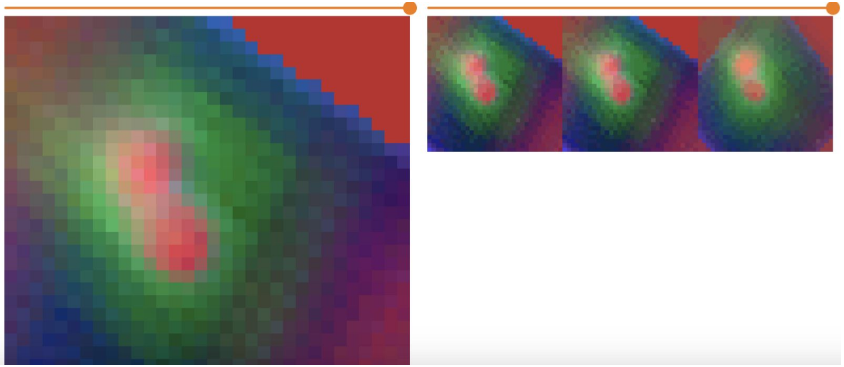
Grasping convex object controller



Pouring controller



RGB Image



3D feature Map