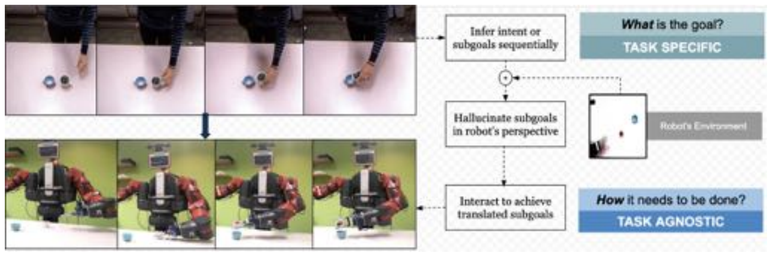
Third-Person Visual Imitation Learning via Decoupled Hierarchical Control

Pratyusha Sharma, Deepak Pathak, Abhinav Gupta

Carnegie Mellon University
The Robotics Institute

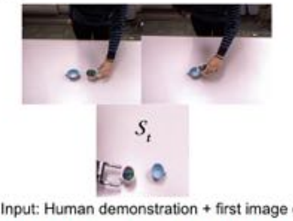


Why is it hard?

- Inferring useful information in the video
- Handling domain shift
- Every *major* part of the sequence needs to be executed correctly - Ex: For pouring, it needs to reach the cup before twisting its hand
- The manipulation is challenging. (6D, novel objects and positioning, no force feedback)

Issue

Scenario 1:



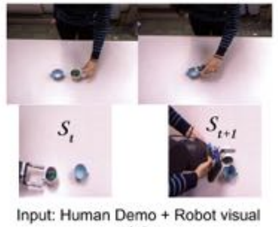
Sequentially predict the states of the robot arm

Output

Input: Human demonstration + first image of object

Issue: Not closed loop. No understanding of how the positions of the objects placed in front of the robot change with time!

Scenario 2:



Sequentially predict the states of the robot arm

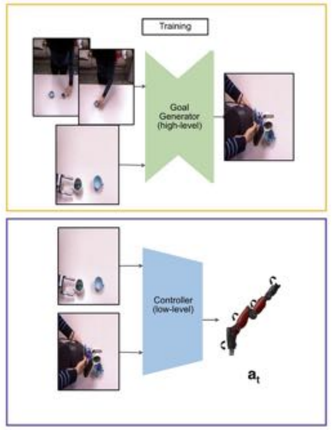
Output

Input: Human Demo + Robot visual state

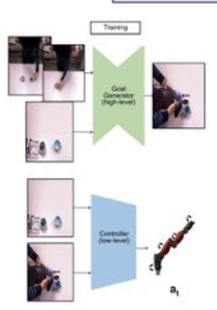
*How do we force it to use task information from Human demonstration **alone** but condition its action on current observable state?*

*We want to build a model that can **infer the intent from the Human Demonstration** of a task and **act in the Robot's current environment** to then accomplish the task.*

*We decouple the task of **Goal Inference** from **Local Control***

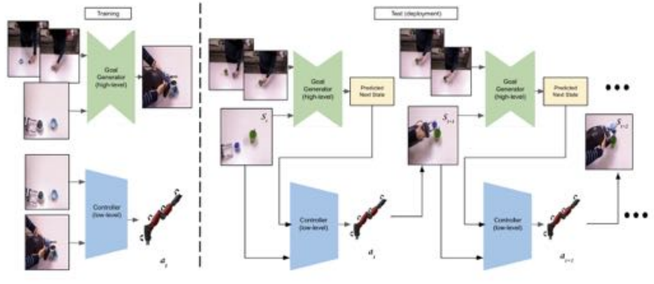


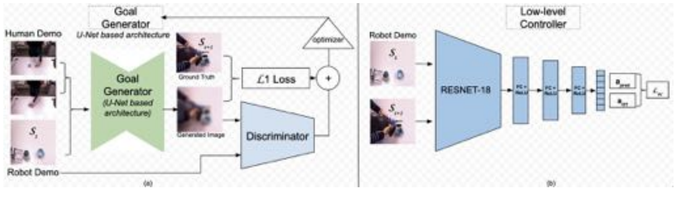
Goal Generator: Given human demo and present visual state of the robot we hallucinate the next step



Inverse Model : Use the hallucinated prediction with the current visual state to predict the action!

Approach - Train Vs Test





Results

Method	L2	L1	PSNR	SSIM
L1 only	60.24	72.57	2.92	0.15
L2 only	76.44	75.94	3.02	0.14
Cycle GAN [31]	99.15	118.67	2.37	0.11
Goal-Gen(Ours)	**39.98**	**52.37**	**3.95**	**0.18**

Table 1: Goal-Generator generalization to novel objects and locations. Our goal generator outperforms the other approaches, both qualitatively and quantitatively, across different loss metrics. The models are evaluated on the pouring test set.

Method	RMSE (mean)	RMSE (stderr)
End to End (all) [25]	14.7	2.3
End to End (single) [25]	8.9	1.7
DAML (single) [28]	11.84	2.1
Ours (all)	**14.4**	**2.2**
Ours (single)	**8.1**	**1.6**

Table 2: Inverse model generalization to novel objects and locations. This table contains models trained on all tasks of the MIME dataset (all) and just the task of pouring (single). The models are evaluated on the common test set of pouring

Method	Train (15 Tasks) Mean	Train (15 Tasks) Stderr	Test (5 Tasks) Mean	Test (5 Tasks) Stderr
End to End [25]	23.63	1.06	24.83	1.56
DAML [28]	35.90	1.56	36.45	1.55
Inv. Model (Ours)	**18.05**	**0.76**	**16.90**	**1.04**

Table 3: Generalization of the Inverse-Model to New Tasks. Our inverse model is trained on 15 tasks of the MIME dataset. It is evaluated on a held-out set from training tasks as well as 5 novel tasks where it significantly outperforms the baselines.

Method	Pouring Reaches	Pouring Pours	Placing Reaches	Placing Drops
End to End [25]	20%	8%	20%	10%
DAML [28]	25%	15%	20%	10%
Hierarchy (Ours)	**75%**	**60%**	**70%**	**50%**

Table 4: Joint evaluation of our hierarchical decoupled controllers. Our approach outperforms the other baselines on the tasks of pouring and placing in a box with a significant margin, however, it is still much far from perfect completion of the task.