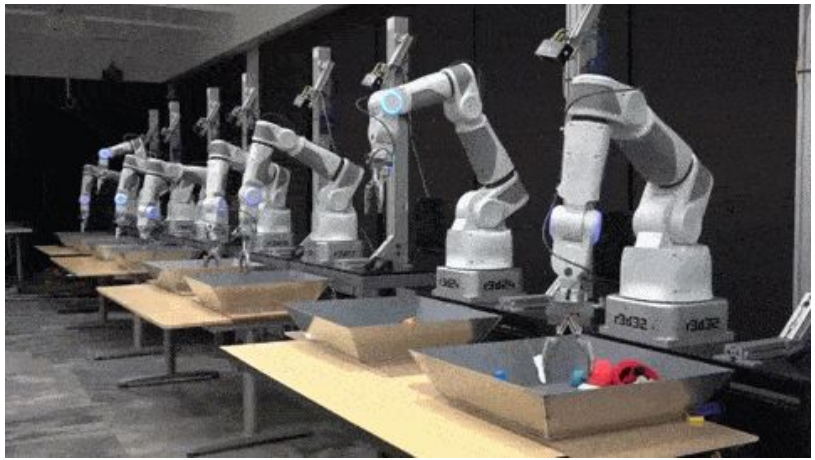
Reinforcement and Imitation Learning for Diverse Visuomotor Skills

Presenter: Joseph Muffoletto

10/28/21

Robotic Manipulation



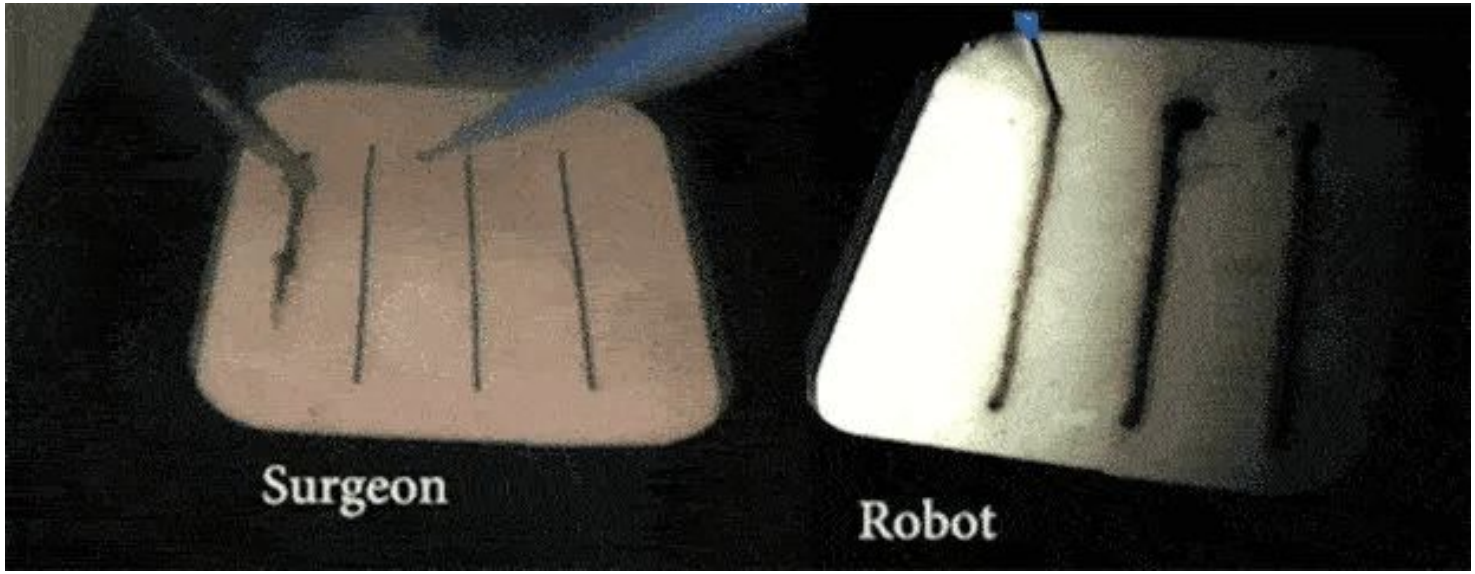
Manipulate the environment to complete **tasks**

Source: Google Has a Room Full of Robot Arms Learning Hand-Eye Coordination

Robotic Manipulation Applications

Source: Industrial Automation

Robotic Manipulation Applications



Source: [In Flesh-Cutting Task, Autonomous Robot Surgeon Beats Human Surgeons]

How do we achieve general robotic manipulation?

Deep reinforcement learning!

- End-to-End Training of Deep Visuomotor Policies (Levine et al.)
  - Trains a policy directly from proprioceptive and visual inputs
- Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection (Levine et al.)
  - Learn visuomotor skills by predicting grasp probabilities from images
- Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates (Gu et al.)
  - Uses DDPG and parallel training

But...

Drawbacks of Deep RL

Previously:

- **Generalize poorly** to complex environments
- **Dependent** on hand-crafted **task-specific rewards**
- Perform inefficient exploration in **high-dimensional** and **continuous** action spaces

What algorithm do we know that can help solve these issues?

How about Imitation Learning?

Benefits:

- Significantly easier **exploration** by leveraging human **demonstration**
- Recent advances such as **GAIL** require **less training data**

Downsides:

- Learning from human demonstrations can lead to **suboptimal policies**

How can we get the best of both Deep RL and Imitation Learning?

Hybrid Solution - Key Insight #1

We can combine **reinforcement** and **imitation** learning to simplify exploration while also exceeding human performance

Drawbacks of the Hybrid Solution

Still:

- **Generalize poorly** to complex environments
- ~~Dependent on hand-crafted **task-specific rewards**~~
- ~~Perform inefficient exploration in **high-dimensional** and **continuous** action spaces~~
- ~~Learning from human demonstrations can lead to suboptimal policies~~

How can we improve generalization?

Key Insight #2

We can improve **generalization** by increasing the **diversity** of the training conditions

Problem and Method

Problem

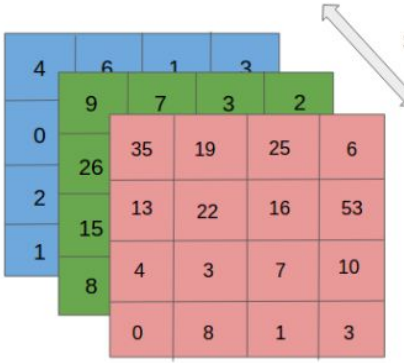
Given **proprioceptive** and **RGB** inputs, generate a **policy** to solve a robotic manipulation task

24-dimensional proprioceptive feature vector

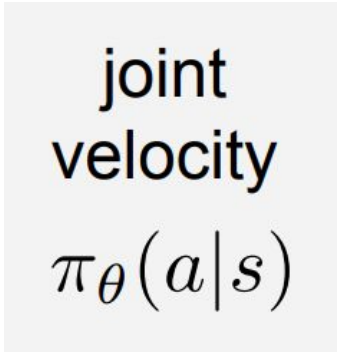
Positions (12-d) and velocities (6-d) of the six arm joints

Positions (6-d) of the three fingers

+



64x64x3 pixel observations



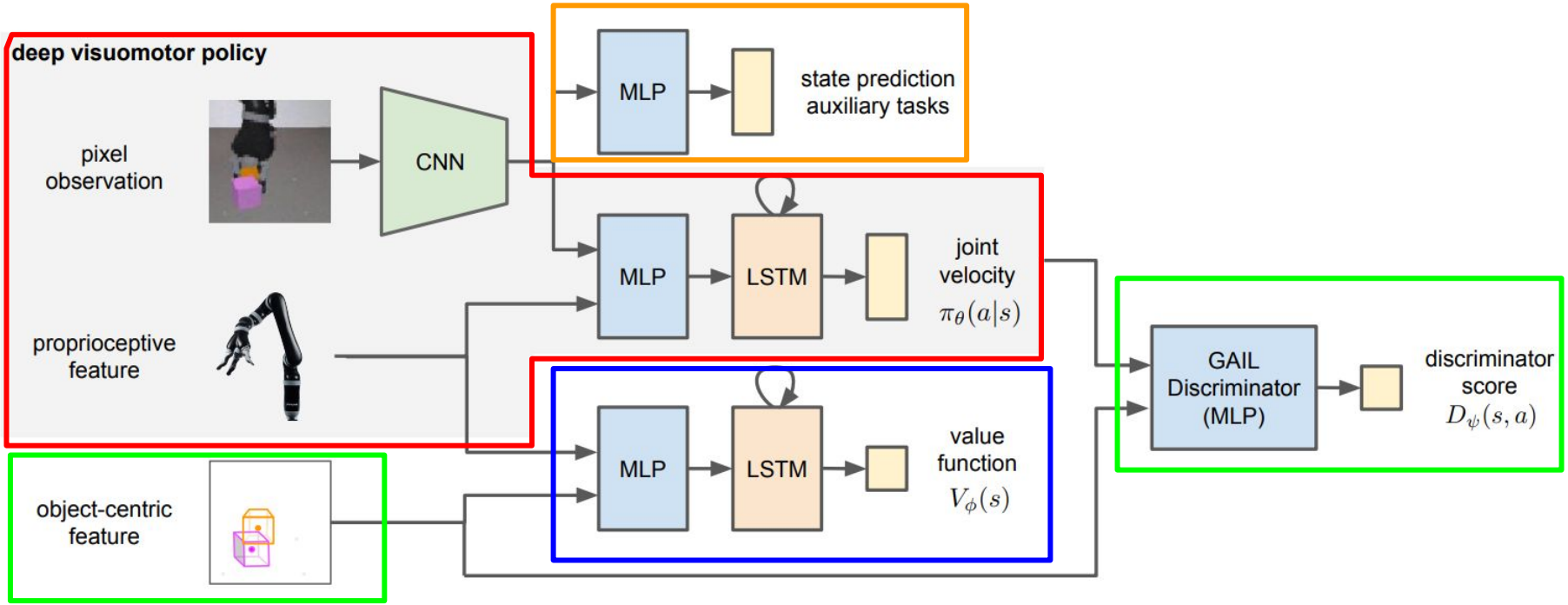
9-dimensional velocities

Velocities (6-d) of the six arm joints

Velocities (3-d) of the three fingers

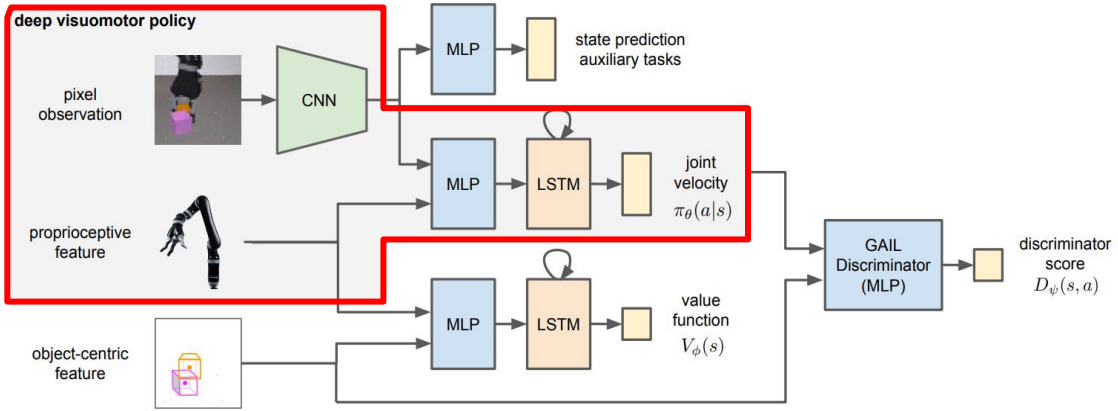
Source: A Comprehensive Guide to Convolutional Neural Networks

The proposed solution - Architecture



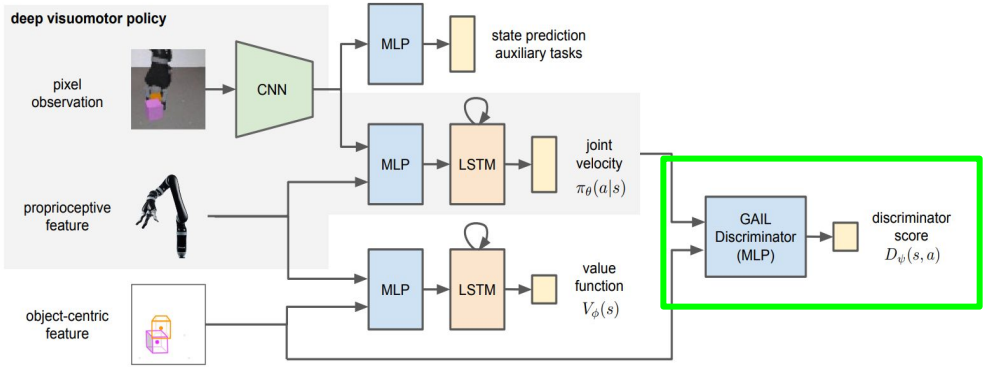
Deep Visuomotor Policy

- Encodes **pixel** inputs using a **CNN**
- Encodes **proprioceptive** features using an **MLP**
- Run the concatenated features through an **LSTM** to produce **joint velocities**



Now, what is this “GAIL Discriminator” module?

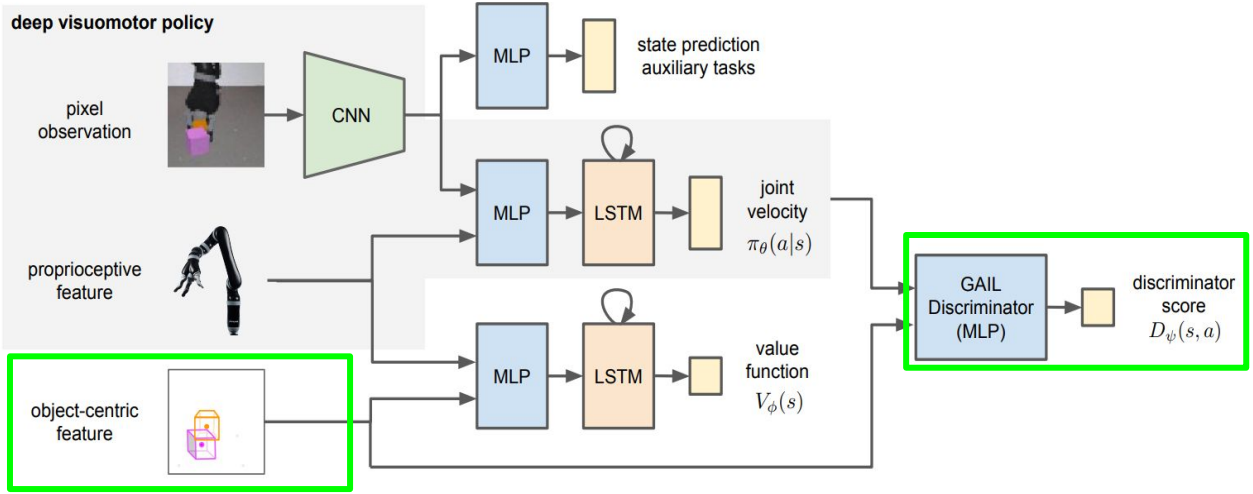
- GAIL: provably equivalent to **Inverse RL**, except that it's much **cheaper** to train
  - Train **discriminator** to discriminate between the **expert** and the **apprentice**
  - Train apprentice **policy** to fool the discriminator into thinking it is the **expert**



$$\min_{\theta} \max_{\psi} \mathbb{E}_{\pi_E}[\log D_\psi(s,a)] + \mathbb{E}_{\pi_\theta}[\log(1 - D_\psi(s,a))]$$

Object-Centric Discriminator

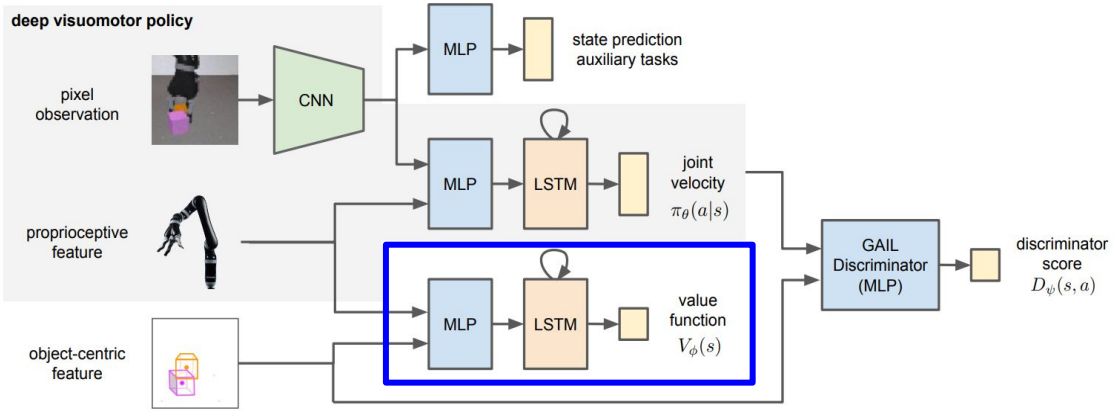
Problem: Discriminator performs **poorly** when trained on **proprioceptive features** (gets distracted by frivolous information)



Instead: Use **object-centric features** such as absolute and relative positions of the objects

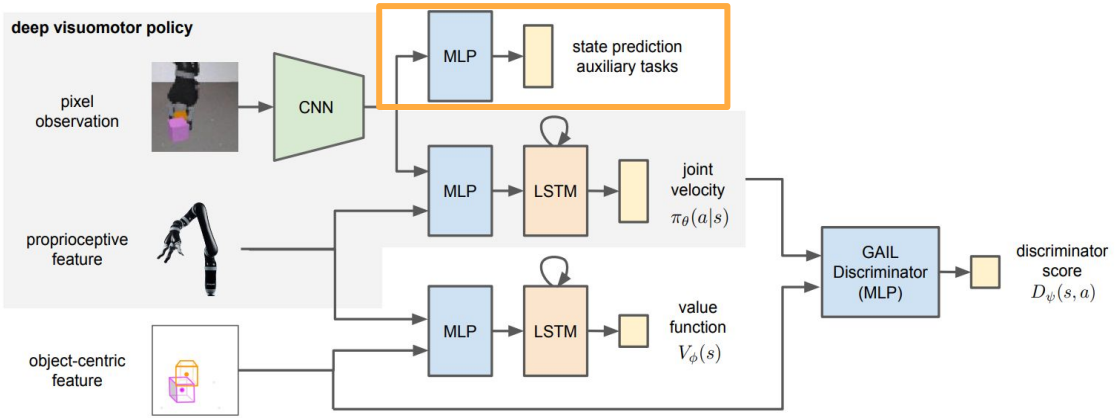
Value Function

- **Proximal Policy Optimization** uses a **value function** during training
- Beneficial to train on a **different modality** than the **policy** (increases stability)



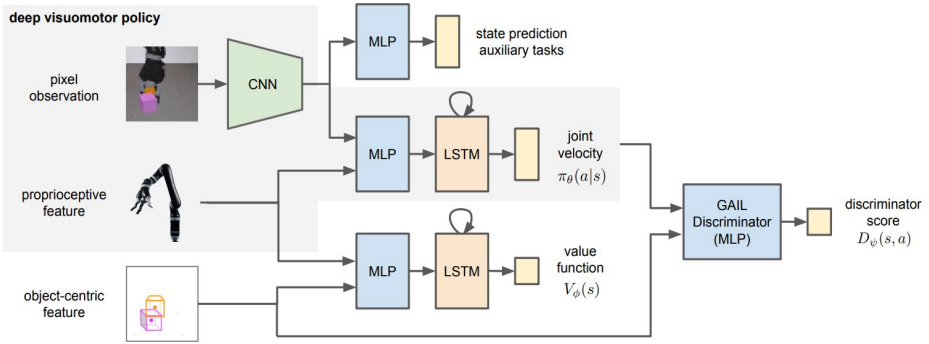
State Prediction Auxiliary Task

- **Goal: Accelerate** the **training** of the **CNN**
- **How**: Solve an auxiliary task -> Train an MLP to **predict** the locations of **objects** from the camera observation



Training

- Trained **End-to-end** using Proximal Policy Optimization (**PPO**)
- Uses **domain randomization** to increase training diversity
- Maximizes a **hybrid reward**
- Uses **demonstration** as **curriculum**



$$r(s_t, a_t) = \lambda r_{gail}(s_t, a_t) + (1-\lambda) r_{task}(s_t, a_t) \;\; \lambda \in [0, 1]$$

Training with the hybrid reward!

$$r(s_t, a_t) = \lambda r_{gail}(s_t, a_t) + (1-\lambda) r_{task}(s_t, a_t) \quad \lambda \in [0, 1]$$

Intuitively...

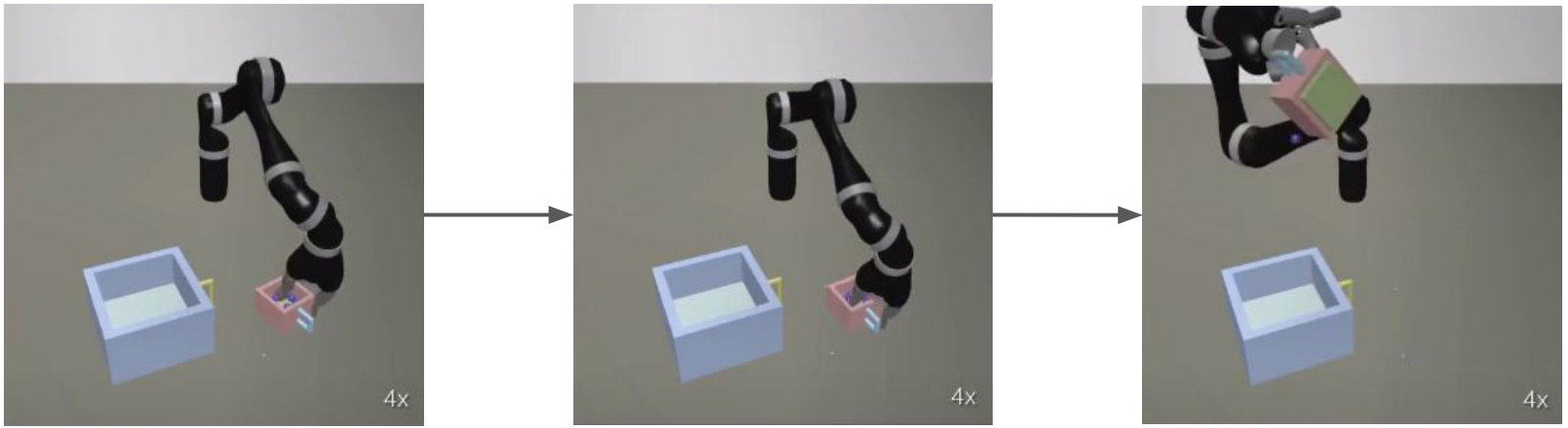
If we wanted **pure reinforcement learning** -> **λ** = 0

If we wanted **pure GAIL** -> **λ** = 1

For a balanced approach -> **λ** = .5

Demonstration as Curriculum?

Progression of mug pouring task



Reach the mug | Grasp the mug | Pour the mug

With probability 1 - ϵ , start the training episode in one of these states!
Otherwise, start in a random initial state

Experiments and Results

Environment Setup

Kinova Jaco arm

- 9 DoF, 6 arm joints and 3 fingers
- Controlled by 9-D continuous veloicities, [-1,1] at 20Hz

Setup

- Diverse set of objects on a tabletop
- Simulated environment generated using MuJoCo
- Images generated by a Kinect camera

Demonstrations

- Collected using a SpaceNavigator 3D motion controller
- Collected 30 episodes of demonstration for each task (<30 minutes per task)

Tasks

1. **Block lifting** : grasp and lift a randomized block
2. **Block stacking** : stack one block on top of the other block
3. **Clearing blocks on table** : two blocks on table, efficiently remove them
4. **Clearing table with a storage box** : toy car and box on table, efficiently remove them
5. **Pouring liquid** : pour simulated liquid from one mug to another
6. **Order fulfillment** : toy planes, toy cars, red box, green box -> planes into green box and cars into red box

All were trained and tested in sim, 1 and 2 were tested in real life

Evaluation Metrics

Baselines:

- Pure RL, **λ** = 0
- Pure GAIL, **λ** = 1
- RL without demonstration curriculum

Recall:

$$r(s_t, a_t) = \lambda r_{gail}(s_t, a_t) + (1-\lambda) r_{task}(s_t, a_t) \;\; \lambda \in [0, 1]$$

Metrics:

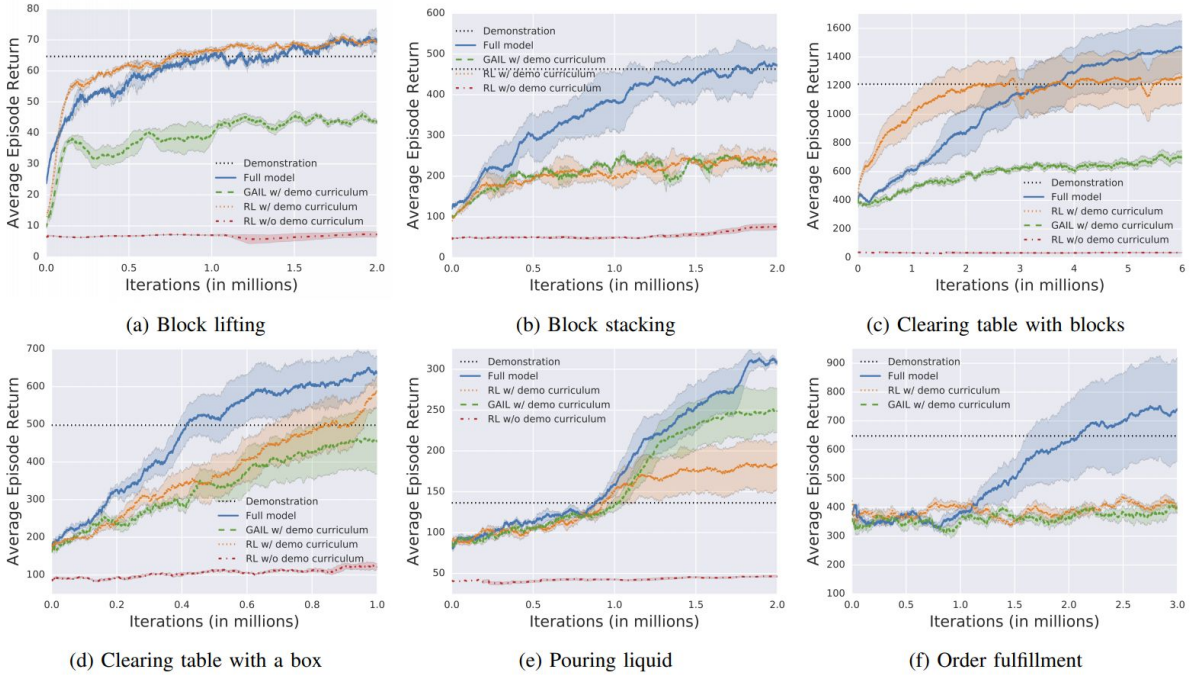
- Average episode return vs. number of training iterations

Hypotheses:

- **Demo curriculum** should greatly increase **training** speeds
- **Full model** should exceed pure GAIL and pure RL in **training time** and **end performance**

Demonstration as curriculum greatly increases training speed

RED line is the only w/out curriculum



(a) Block lifting (b) Block stacking (c) Clearing table with blocks

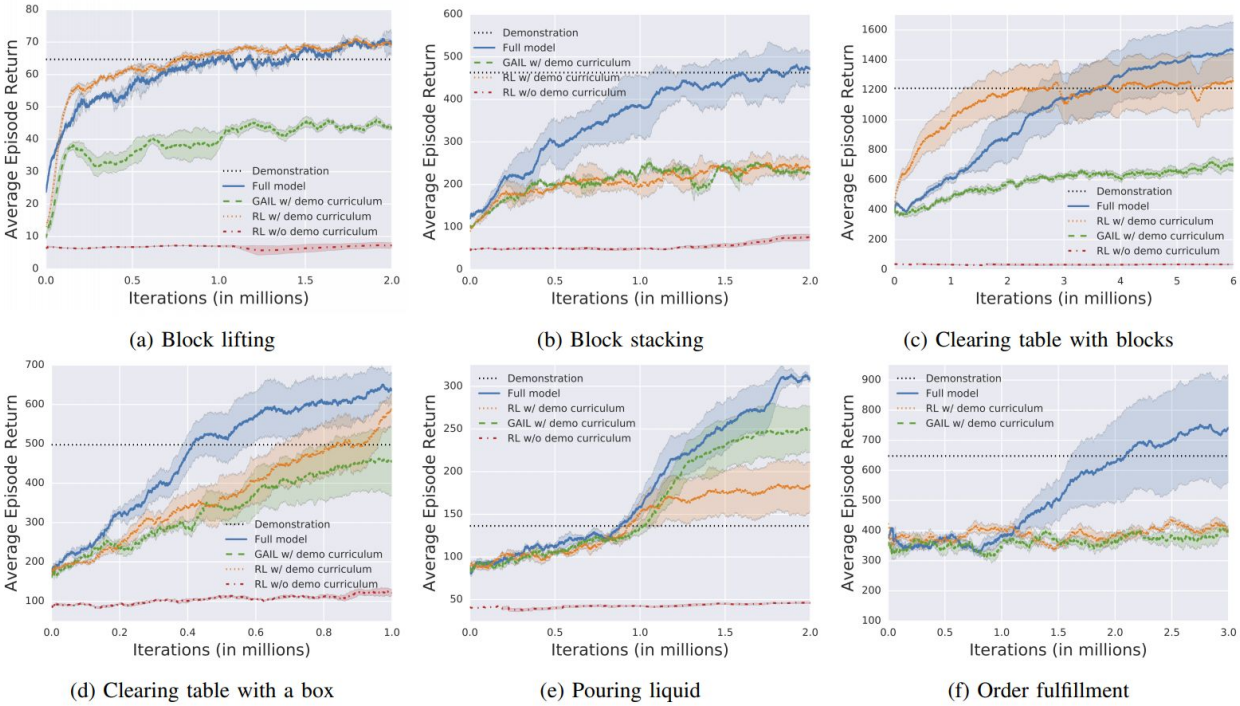
(d) Clearing table with a box (e) Pouring liquid (f) Order fulfillment

The hybrid/full model almost always outperforms baselines

BLUE: Full model

GREEN: Pure GAIL

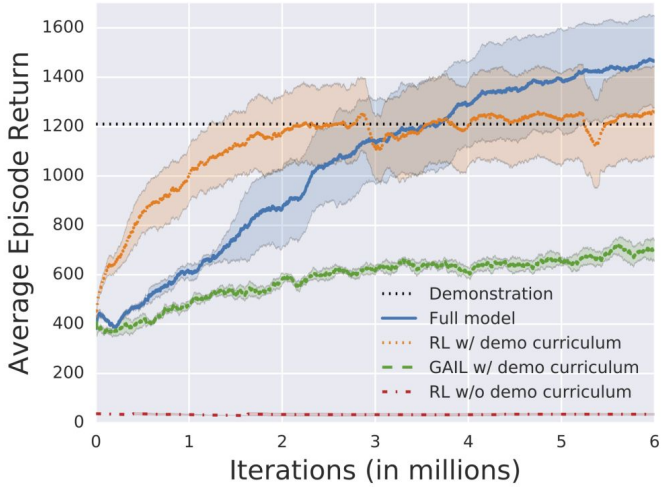
ORANGE: Pure RL



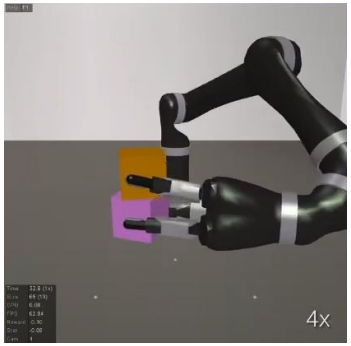
(a) Block lifting (b) Block stacking (c) Clearing table with blocks

(d) Clearing table with a box (e) Pouring liquid (f) Order fulfillment

Pure RL trains faster in the clearing blocks task?



(c) Clearing table with blocks



Suboptimal human demonstration

Optimal learned policy

Pure RL trains faster in the clearing blocks task?

Key Insight: Pure RL did not have to overcome the suboptimal human demonstration and thus trained faster initially

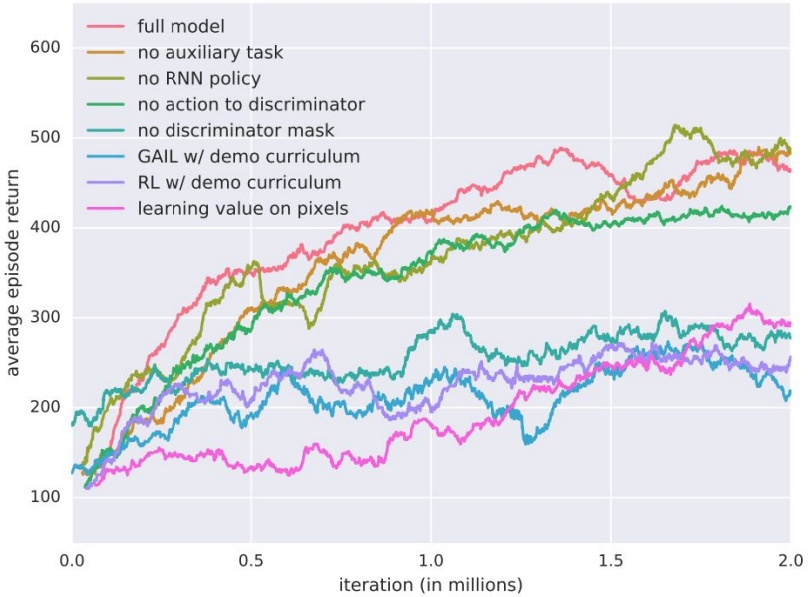
Ablation Experiments

Requirements:

- GAIL + RL
- Discriminator Mask (object-centric discriminator)
- Value function learned on proprioceptive features, not on pixels

Optional:

- Auxiliary Task for CNN training
- LSTM module in core
- Action to discriminator

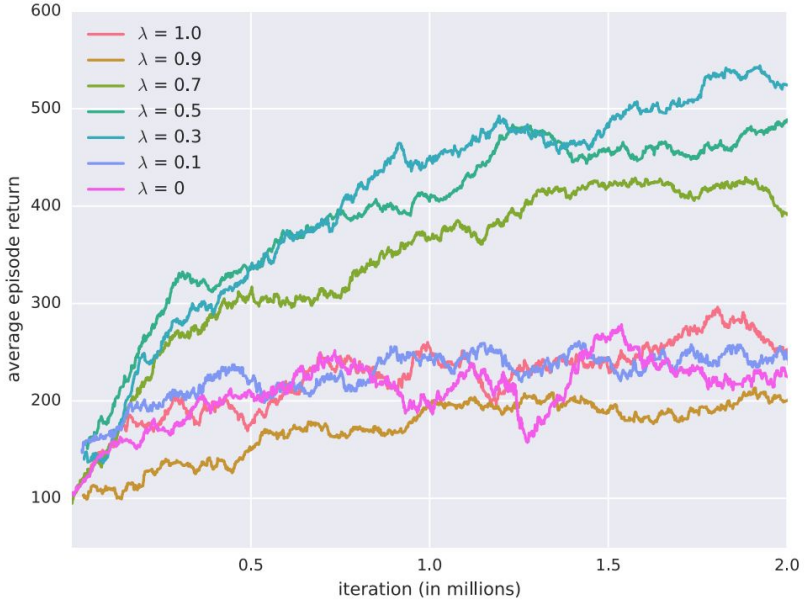


(a) Ablation study of model components

Tuning the hybrid reward hyperparameter

Takeaways:

- ❖ Neither extreme works well (pure RL vs pure GAIL)
- ❖ Workable range : [.3,.7]
- ❖ In general, an RL favored reward responds better than an imitation favored reward to increased training



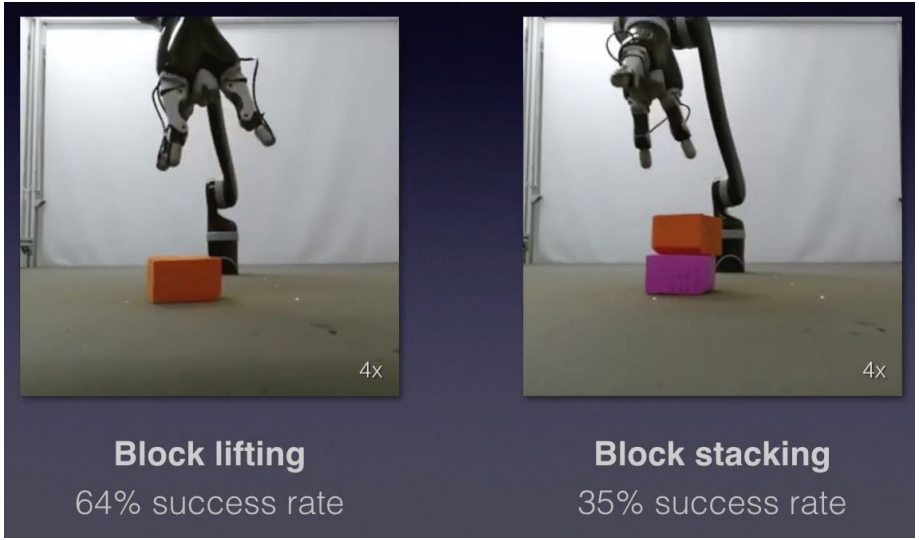
(b) Model sensitivity to λ values

Sim2Real - close, but not yet

Lifting has a **64%** success rate

Stacking has a **35%** success rate

A good starting point



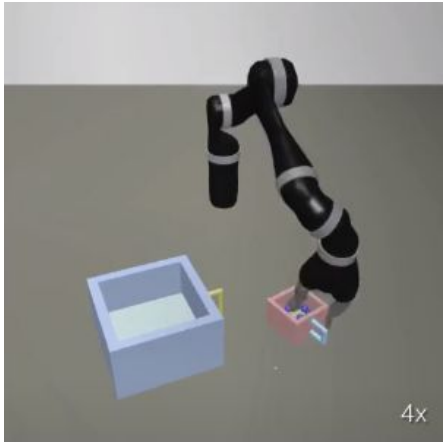
Discussion and Wrap-up

What works

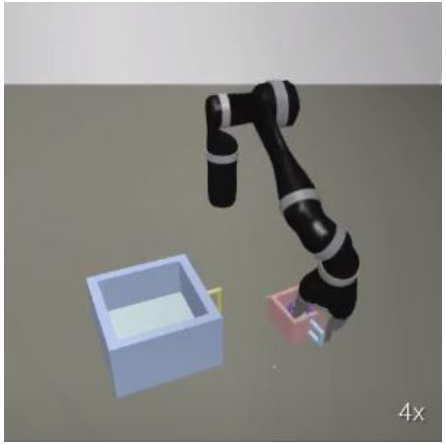
- Reinforcement and imitation learning make up for each others flaws
- The hybrid method reduces training time and increases end performance relative to baselines
- Initializing training episodes to intermediate states can simplify long-horizon tasks
- Object-centric features provide the correct signals to a GAIL discriminator
- In PPO, the value function and policy should be learned on different modalities

Limitations - Task-Specific Rewards

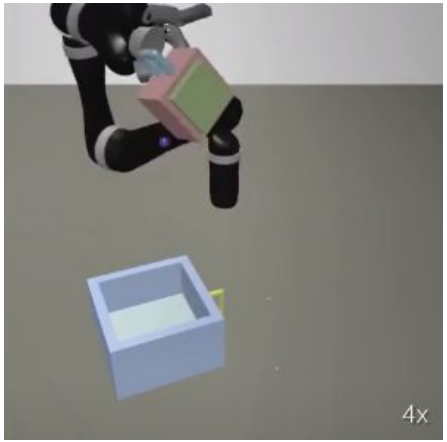
Progression of mug pouring task



Reach the mug



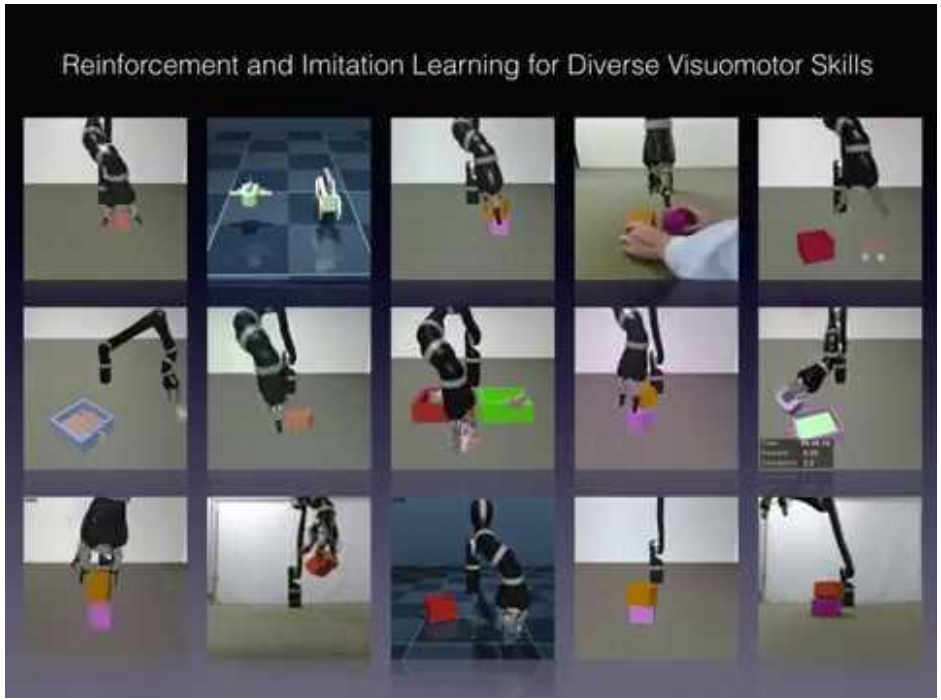
Grasp the mug



Pour the mug

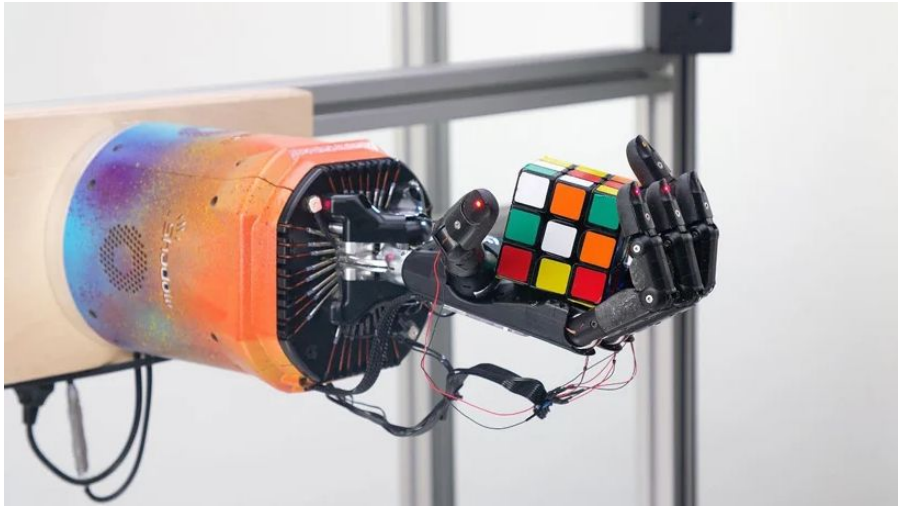
While not entirely dependent, the proposed method still requires the user to break down the task and create an adequate reward function

Limitations - “Expert” Demonstration



Future Work

- Better sim2real performance
- More complex tasks
- More degrees of freedom



Source: extremely dextrous robot arm uses AI to solve rubik's cube one-handed.

Extended Readings

- Andrychowicz, OpenAI: Marcin, et al. "Learning dexterous in-hand manipulation." The International Journal of Robotics Research 39.1 (2020): 3-20.
- Ibarz, Borja, et al. "Reward learning from human preferences and demonstrations in Atari." arXiv preprint arXiv:1811.06521 (2018).
- Kostrikov, Ilya, et al. "Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning." arXiv preprint arXiv:1809.02925 (2018).
- Pfeiffer, Mark, et al. "Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations." IEEE Robotics and Automation Letters 3.4 (2018): 4423-4430.

Summary

- **Problem:** How can we achieve diverse visuomotor robotic manipulation skills?
- **Importance**: Crucial step towards long-horizon, real life, and high-dimensional robotic manipulation
- **Prior limitation**: Inefficient policy searches
- **Key insight**: We can combine reinforcement and imitation learning to simplify exploration while also exceeding human performance
- **Benefits of insight**:
- Train faster, solve harder problems, and exceed demonstrations

Questions?

Thank you!