CS 287 Fall 2019 - Lecture 17
Imitation Learning

Outline

- **Setup**
- Supervised learning
- Inverse optimal control
- Other key directions, example applications

Problem Setup & Overview

- Input:
  - State space, action space
  - Transition model
  - Demonstrations (samples from π^*)
    - Example: Cleaning robot
- Behavioral cloning
  - Estimation of π^*
- Inverse optimal control/RL
  - Estimation of R, and use to learn π^*

Outline

- Setup
- **Supervised learning**
- Inverse optimal control
- Other key directions, example applications

Behavioral Cloning

- Input:
  - State space, action space
  - Transition model
  - Demonstrations (samples from π^*)
    - (s0, a0), (s1, a1), (s2, a2), ...
- Learn mapping from (state, action) pairs to estimate π^*
  - Neural network, decision tree, SVM, etc.

Distributional Shift

- Common assumption is that training and test are iid
- However, $p_{\pi^*}(o_t) \neq p_{\pi_\theta}(o_t)$. Why?

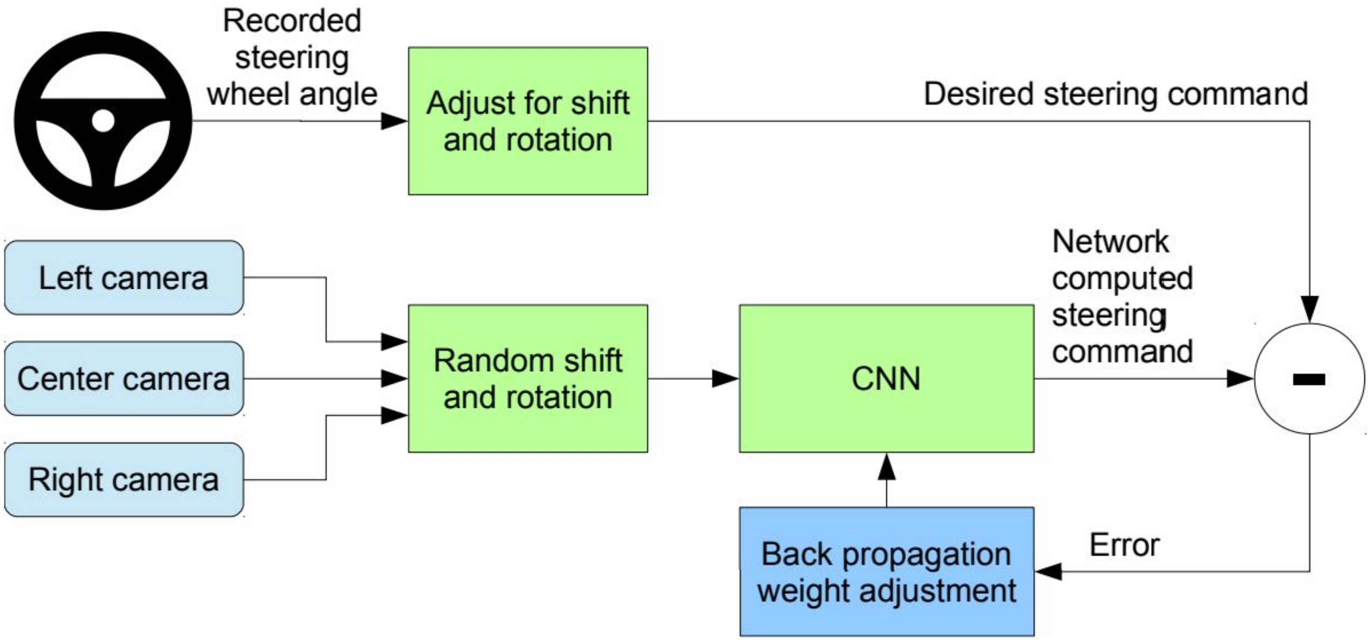
Image courtesy of Bertram Ruprecht/Getty Images

Distributional Shift

- Common assumption is that training and test are iid
- However, $p_{\pi^*}(o_t) \neq p_{\pi_\theta}(o_t)$. Why?

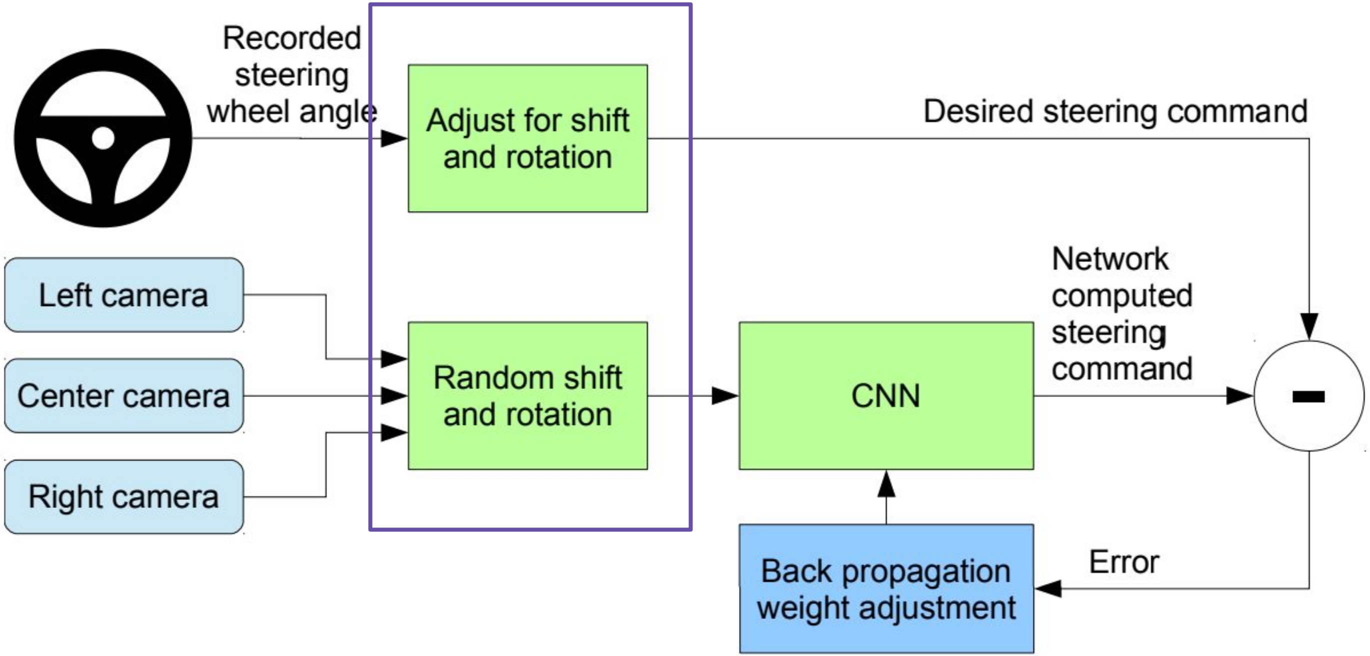
Image courtesy of Bertram Ruprecht/Getty Images

Example: DAVE-2



End to End Learning for Self-Driving Cars, Bojarski et al. 2016

Example: DAVE-2



End to End Learning for Self-Driving Cars, Bojarski et al. 2016

Example: DAVE-2

Example: DAgger

Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.

- Query expert for labels on $p_{\pi_i}(o_t)$
- Train on aggregated dataset

- Theoretical guarantees
- Expensive, not always possible

A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning, Ross et al. 2011

Example: DAgger

Image courtesy of Bertram Ruprecht/Getty Images

Example: DAgger

Image courtesy of Bertram Ruprecht/Getty Images

Example: DAgger

Learning Monocular Reactive UAV Control in Cluttered Natural Environments, Ross et al. 2013

Outline

- Setup
- Supervised learning
- **Inverse optimal control**
- Other key directions, example applications

Can we do better with the expert data?

- Behavioral Cloning mimics the expert, no notion of *intention*
  - Expert suboptimality
  - Different embodiments
  - Robustness
- Effectively finding out *what* the teacher is trying to do, can potentially enable the agent to do *better* than the demonstrator

Inverse Optimal Control

- Input:
  - State space, action space
  - Transition model
  - Demonstrations (samples from π^*)
    - (s0, a0), (s1, a1), (s2, a2), ...
- Learn reward function R(s,a)
- Use the reward function to learn π^*

Some simplifying assumptions

- We assume a linear reward function on featurized state

$$\text{Let } R(s) = w^\top \phi(s), \text{ where } w \in \mathbb{R}^n, \text{ and } \phi : S \to \mathbb{R}^n.$$

- The value function w.r.t. a particular reward function and policy is then:

$$\begin{aligned}
\mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi] &= \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t w^\top \phi(s_t) | \pi] \\
&= w^\top \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t) | \pi] \\
&= w^\top \mu(\pi)
\end{aligned}$$

$\mu(\pi)$ → *'feature expectations'*

Feature Matching

- The value of the optimal policy w.r.t. the ‘true’ reward function is greater than the value of any other policy (by definition)

$$\mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*] \geq \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi] \qquad \forall \pi$$

- Plugging in from previous slide, we want to

$$\text{Find } w^* \text{ such that } {w^*}^\top \mu(\pi^*) \geq {w^*}^\top \mu(\pi) \qquad \forall \pi$$

Feature Matching

- For a policy to be guaranteed to perform as well as the expert policy, it suffices that the feature expectations 'match' Concretely,

If $\|\mu(\pi) - \mu(\pi^*)\|_1 \leq \epsilon$, then $|w^T\mu(\pi) - w^T\mu(\pi^*)| \leq \epsilon \quad \forall w, \|w\|_\infty \leq 1$

- Justification :

$$
\begin{aligned}
|\mathrm{E}[\textstyle\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*] - \mathrm{E}[\textstyle\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi]| &= |w^T\mu(\pi) - w^T\mu(\pi^*)| \leq \epsilon \\
&\leq \|w\|_\infty \|\mu(\pi) - \mu(\pi^*)\|_1 \\
&\leq 1 \cdot \epsilon = \epsilon
\end{aligned}
$$

Apprenticeship Learning via Inverse Reinforcement Learning, Abbeel and Ng 2004

Apprenticeship Learning via IRL [Abbeel & Ng 2004]

1. Let $R(s) = w^\top \phi(s)$, where $w \in \mathbb{R}^n$, and $\phi : S \to \mathbb{R}^n$.
2. Initialize some policy π_0
3. Iterate for i = 1, 2, 3……
   - **Guess the reward** : Find a reward function such that the demonstrator policy *maximally* outperforms all previously found policies
   - Find **optimal control policy** π_i, for the current reward function
     - If expert suboptimal, pick best policy in a mixture
   - Exit if $\gamma \leq \epsilon/2$

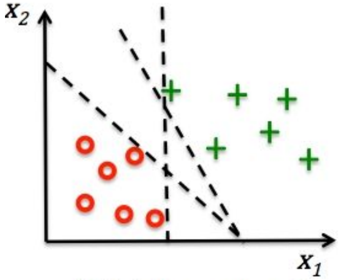
Apprenticeship Learning via Inverse Reinforcement Learning, Abbeel and Ng 2004

A Note on Reward Functions

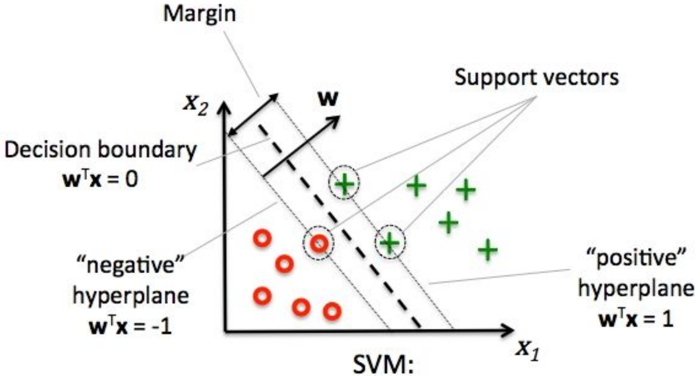
- How do we “guess the reward”?
- Initial IRL formulation by [Ng and Russell, 2000]
  - Degeneracy: “the existence of a large set of reward functions for which the observed policy is optimal”
- How do we resolve ambiguity?

Max Margin Formulation

- Recall standard classification problem



Which hyperplane?



SVM:
Maximize the margin

- *Similar* idea here:
  - Maximally separate the policy induced by our learned reward function from suboptimal policies
  - Formally we can write:

$$\max_{\gamma, w : \|w\|_2 \leq 1} \gamma$$

$$s.t. \quad w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + \gamma \quad \forall \pi \in \{\pi_0, \pi_1, \ldots, \pi_{i-1}\}$$

Apprenticeship Learning via Inverse Reinforcement Learning, Abbeel and Ng 2004

Apprenticeship Learning via IRL [Abbeel & Ng 2004]

1. $\text{Let } R(s) = w^\top \phi(s), \text{ where } w \in \mathbb{R}^n, \text{ and } \phi : S \to \mathbb{R}^n.$
2. Initialize some policy π_0
3. Iterate for i = 1, 2, 3……
   - **Guess the reward** : Find a reward function such that the demonstrator policy *maximally* outperforms all previously found policies

$$\max_{\gamma, w : \|w\|_2 \leq 1} \gamma$$
$$s.t. \quad w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + \gamma \quad \forall \pi \in \{\pi_0, \pi_1, \ldots, \pi_{i-1}\}$$

   - Find **optimal control policy** π_i, for the current reward function
     - If expert suboptimal, pick best policy in a mixture
   - Exit if $\gamma \leq \epsilon/2$

Apprenticeship Learning via Inverse Reinforcement Learning, Abbeel and Ng 2004

Challenges

- Max-margin is one way to break ties, still not guaranteed to capture demonstrator's 'true' objective
- Hard to optimize (constrained optimization) with more expressive reward functions
  - e.g. neural networks
- Expert suboptimality?
  - Add slack variables
    - Analogous to soft-margin SVM
    - See *Maximum Margin Planning,* Ratliff et al. 2006

Max Entropy IRL

- Addressing ambiguity and expert suboptimality by modeling in a probabilistic framework
- Employs the principle of maximum entropy (Jaynes, 1957)
  - Pick the “least committed” distribution subject to constraints

- Assume linear reward function and known dynamics, modeling $p(\tau) \propto e^{-c(\tau)}$ is modeling the objective of the expert as:

$$\min_{\pi} \mathbb{E}_{\pi}[c_{\theta}(\tau)] - \mathcal{H}(\pi)$$

Maximum Entropy Inverse Reinforcement Learning, Ziebart et al. 2008

Max Entropy IRL

1. Initialize θ, gather demonstrations $\mathcal{D}$
2. Solve for optimal policy $\pi(a \mid s)$ w.r.t c_θ
3. Solve for state visitation frequencies $p(s \mid \theta, T)$
4. Compute gradient

$$\nabla_\theta \mathcal{L} = \frac{1}{M} \sum_{\tau_d \in \mathcal{D}} \mathbf{f}_{\tau_\mathbf{d}} + \sum_s p(s \mid \theta, T) \mathbf{f}_\mathbf{s}$$

5. Update θ with one gradient step using $\nabla_\theta \mathcal{L}$

Maximum Entropy Inverse Reinforcement Learning, Ziebart et al. 2008 *slide adapted from C. Finn*

Outline

- Setup
- Supervised learning
- Inverse optimal control
- **Other key directions, example applications**

Imitation via Consumer-Grade VR Teleoperation

- Motivation:
    - There are existing control interfaces for driving cars/piloting drones. What about robotic manipulation?
    - Kinesthetic teaching introduces visual obstruction (problem if depend on vision)
    - How else can we provide demonstrations?

- Highlights
    - Developed cost-effective, consumer-grade VR teleoperation system
    - Single neural network architecture that performs all tasks from vision
    - Behavior cloning loss augmented with *auxiliary loss* making it goal-oriented
        - Source of self-supervision, incorporating some concepts from IRL

Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation, Zhang et al. 2018

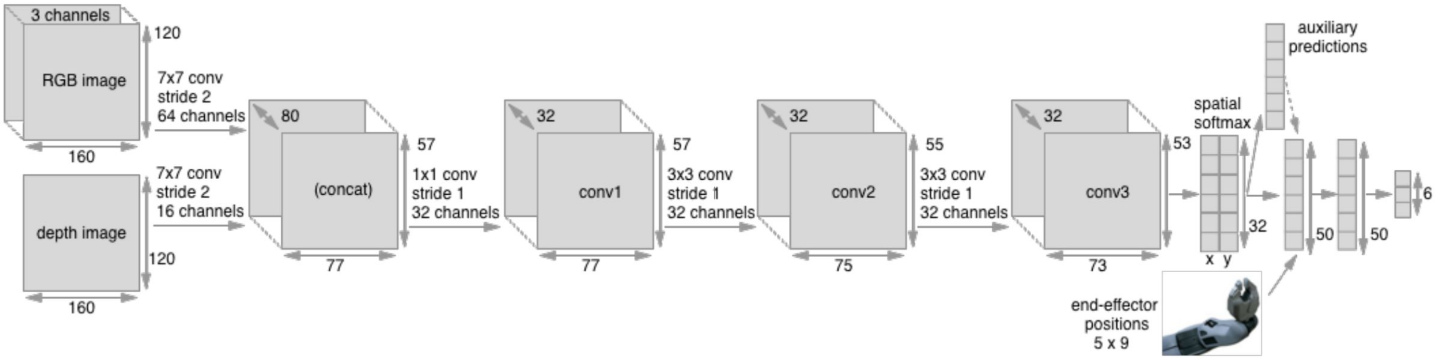
Imitation via Consumer-Grade VR Teleoperation

task: grasp-and-place		
number of demonstrations	success rates (with)	success rates (without)
109	96%	80%
55	53%	26%
11	28%	20%

efficacy of auxiliary loss

Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation, Zhang et al. 2018

Imitation via Consumer-Grade VR Teleoperation



Inputs to the policy include :

- Raw image observation
- End-effector position

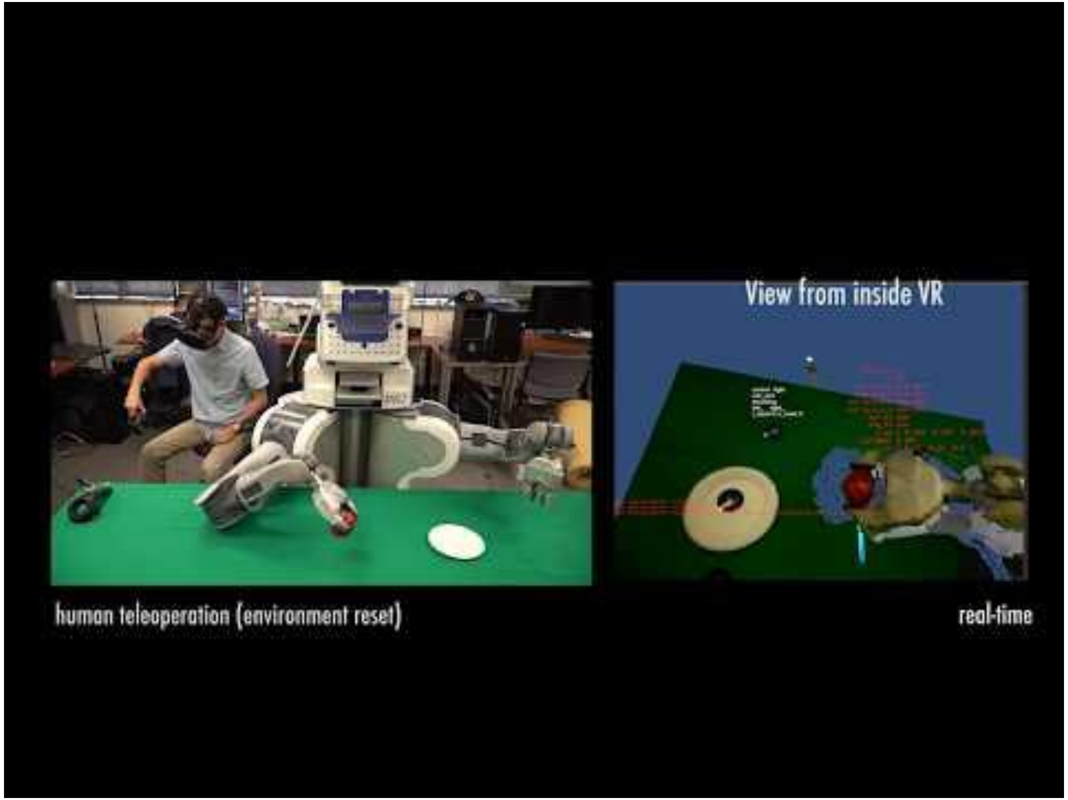
For each auxiliary task (a), the loss is given by:

$$\mathcal{L}_{aux}^{(a)} = ||\mathrm{NN}(f_t; \theta_{aux}^{(a)}) - s_t^{(a)}||_2^2$$

Predict current pose and final pose -> accelerates learning

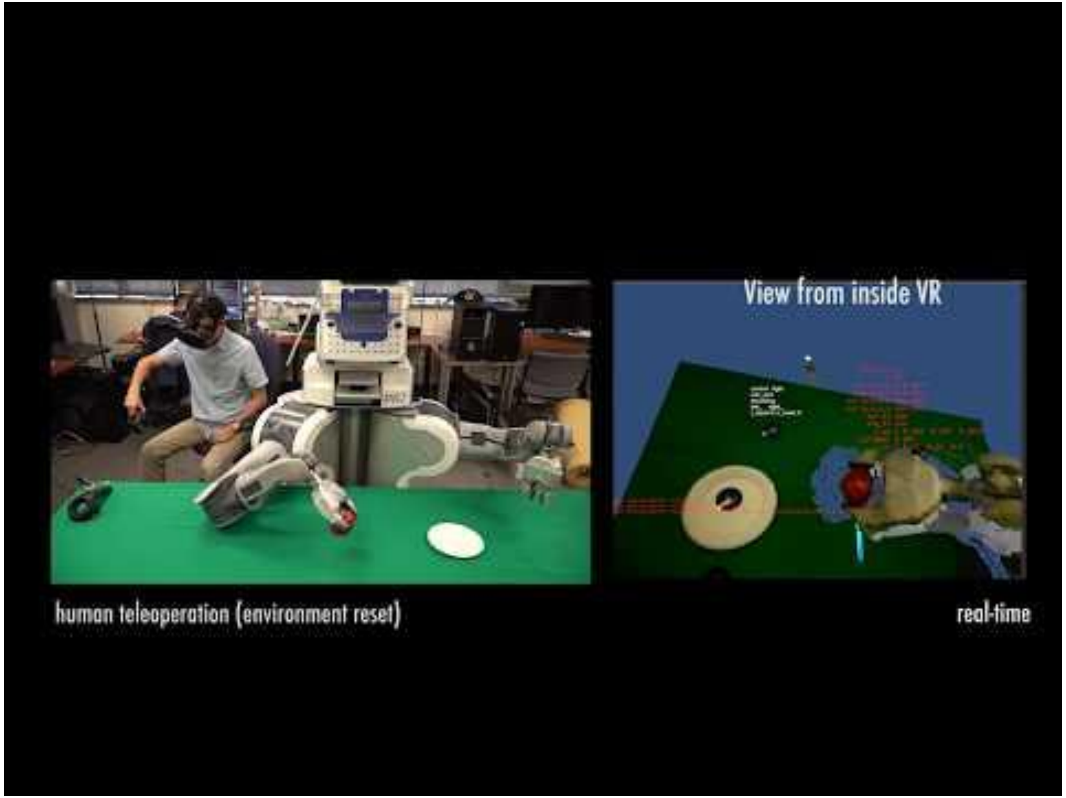
*Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation***,** Zhang et al. 2018

Imitation via Consumer-Grade VR Teleoperation



*Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation***,** Zhang et al. 2018

Imitation via Consumer-Grade VR Teleoperation



*Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation***,** Zhang et al. 2018

Learning from a Single Demonstration

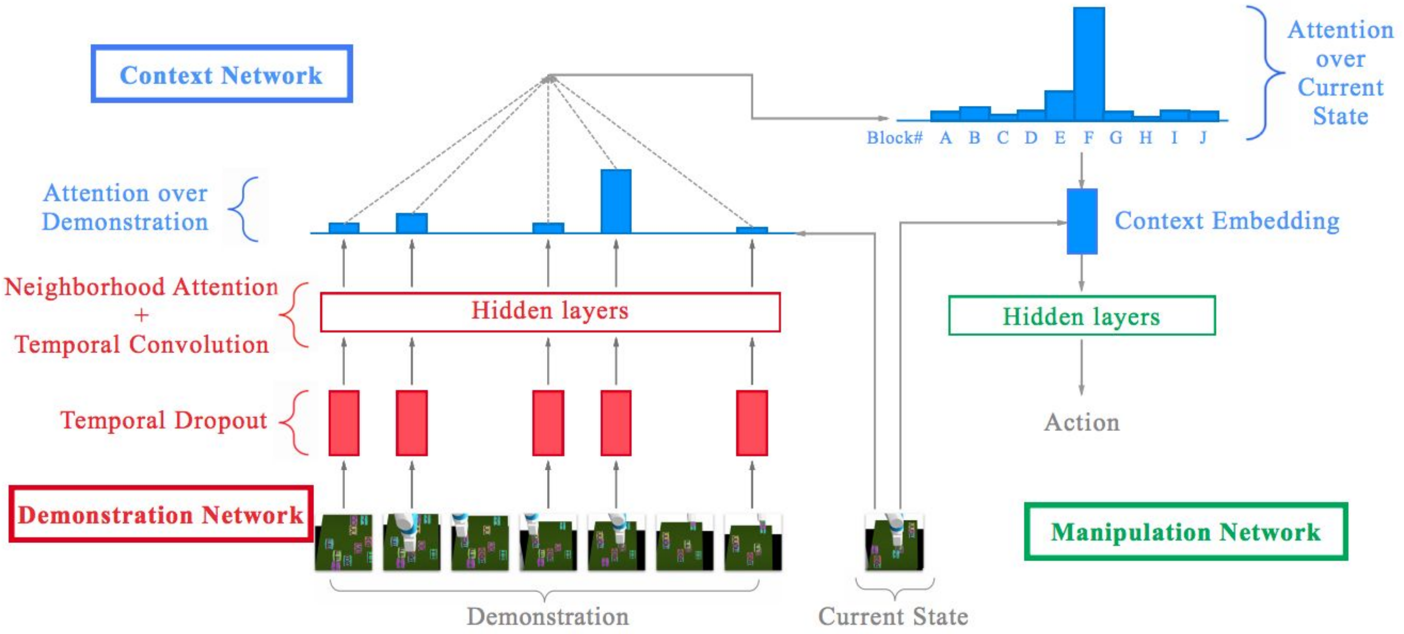
- Motivation:
    - Ideally learn a task from just a few demonstrations and generalize to arbitrary instantiations of the task
    - If we can build a tower of blocks, we should be able to build any configuration of blocks if shown an example

- Highlights:
    - Meta-learning approach trained on pairs of demonstrations
    - A key contribution is the proposed architecture consisting of demonstration, context, and manipulation networks
    - Use of *soft attention* allows the model to generalize to conditions and tasks unseen in the training data

-

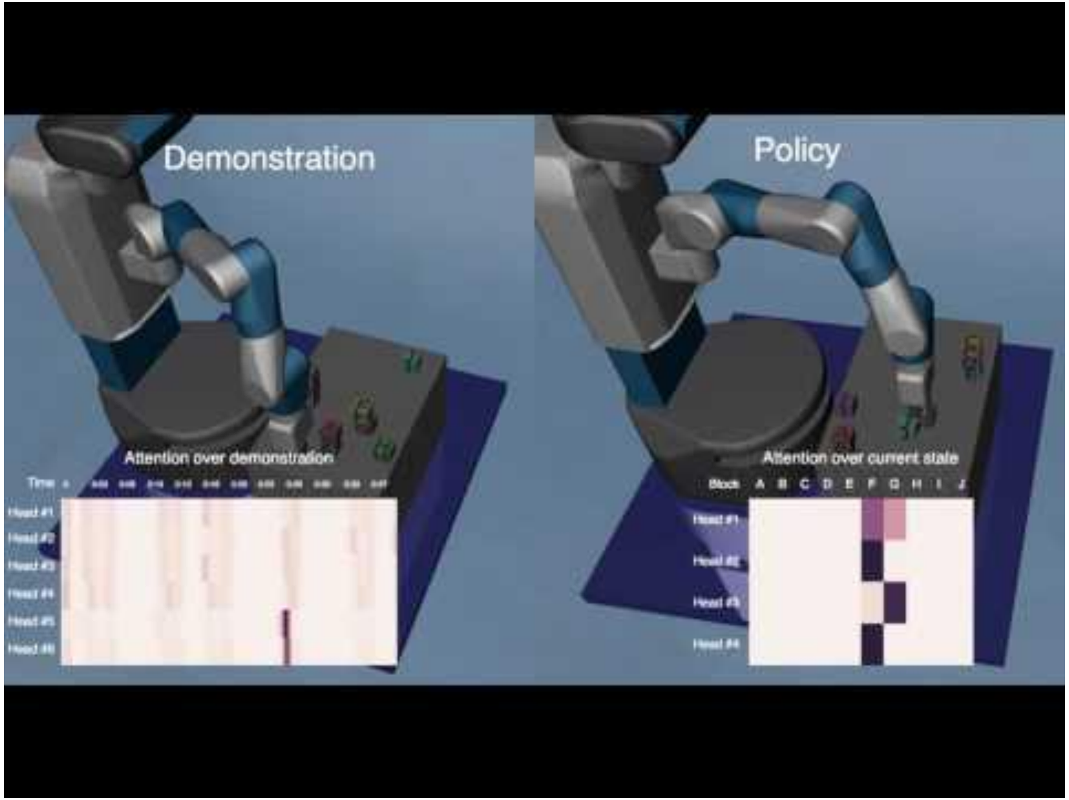
One-Shot Imitation Learning, Duan et al. 2017

Learning from a Single Demonstration



One-Shot Imitation Learning, Duan et al. 2017

Learning from a Single Demonstration



One-Shot Imitation Learning, Duan et al. 2017

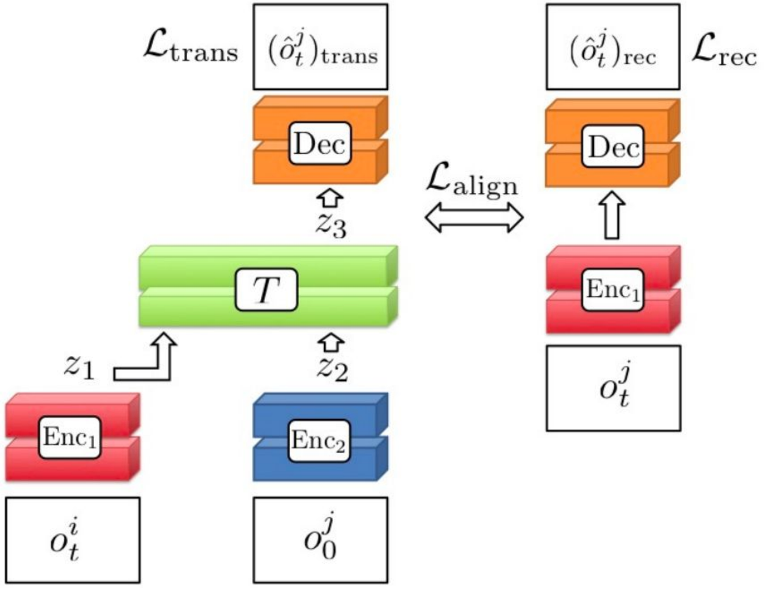
Third-Person Imitation Learning

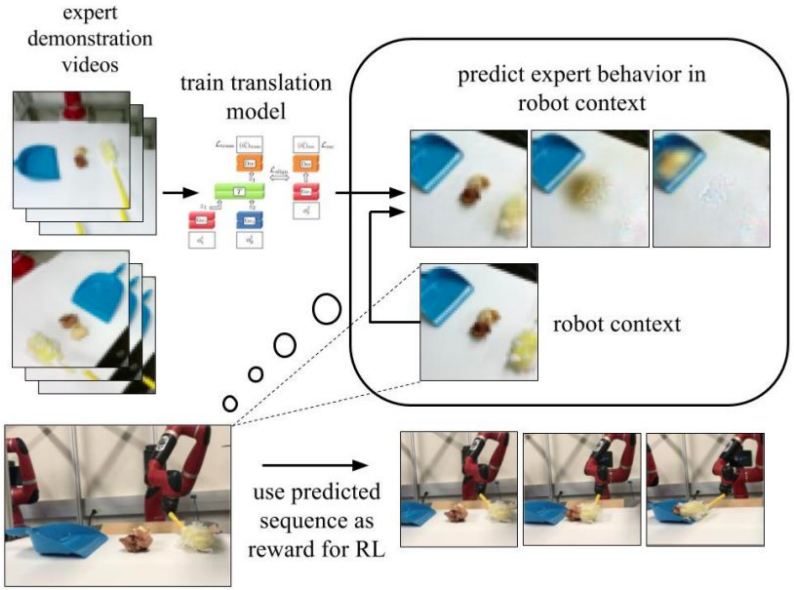
- Motivation:
  - Stringent assumptions that we have access to observations and actions which are consistent with the robot’s (first-person)
  - We should be able to imitate by observing behavior “compensating for changes in viewpoint, surroundings, object positions/types, and other factors” which constitute different *contexts*

- Highlights
  - Learn a context-aware translation model on multiple demonstrations taken in different contexts
  - When faced with a new context, translate demonstrations and use RL to follow the trajectory of translated features

Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation, Liu et al. 2018

Third-Person Imitation Learning





Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation, Liu et al. 2018

Third-Person Imitation Learning

Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation, Liu et al. 2018

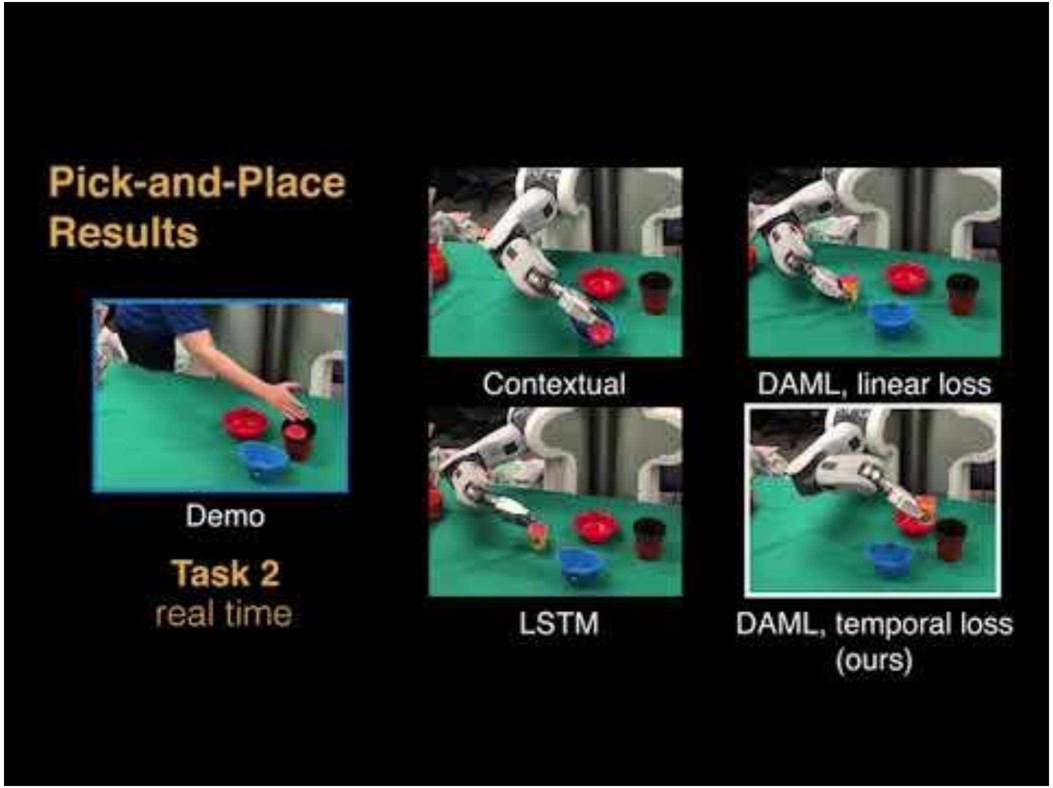
Third-Person, One-Shot Imitation Learning

- Motivation:
  - We, as humans, can imitate others by observing a single demonstration
  - Imitation by observing humans is enticing, but it is difficult to resolve differences in morphology (previous work we saw circumvented this challenge by using tools)

- Highlights:
  - Instead of manual correspondence + pose detection to overcome differences (maybe this isn't even possible), take a data-driven approach and *infer* the goal
  - Build a rich prior on structurally similar tasks during *meta-training* to be able to infer a policy given a human demo
  - Uses temporal convolutions to integrate temporal information in demonstration

One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning, Yu et al. 2018

Third-Person, One-Shot Imitation Learning



One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning, Yu et al. 2018