Introduction to inverse reinforcement learning (3/3)

Eiji Uchibe

Dept. of Brain Robot Interface

ATR Computational Neuroscience Labs.

Driver Route Modeling

- Modeling route selection strategies of 25 taxi drivers
- Fomulated as a deterministic MDP
  - 300,000 states (e.g., road segments)
  - 900,000 actions (e.g., transitions at intersection)
- We assume that the reward weight is independent of the goal state and, therefore, a single reward weight can be learned from many MDPs that differ only in the goal state

road network (Pittsburgh)

Ziebart, B.D., Maas, A., Bagnell, J.A., & Dey, A.K. (2009). Human Behavior Modeling with Maximum Entropy Inverse Optimal Control. *Proc. of AAAI Spring Symposium on Human Behavior Modeling*, 3931–3936.

Driver Route Modeling

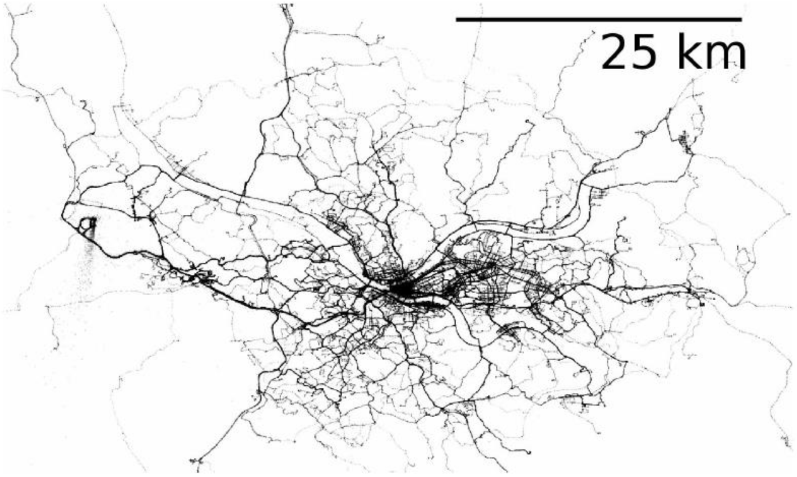
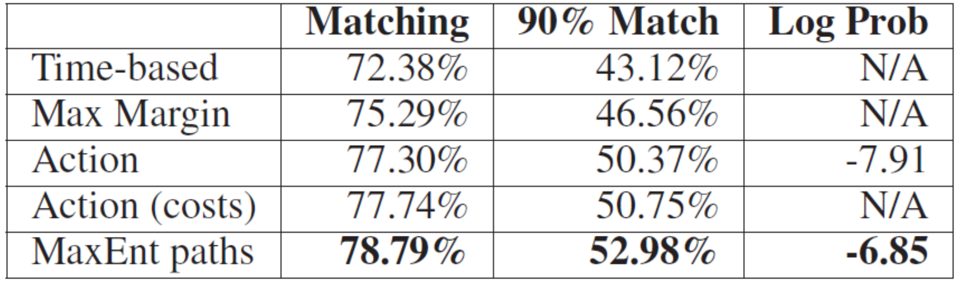
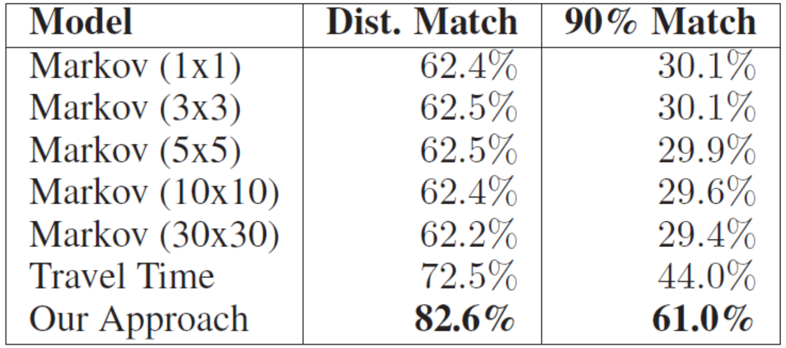
- 

Figure 2: The collected GPS datapoints



	Matching	90% Match	Log Prob
Time-based	72.38%	43.12%	N/A
Max Margin	75.29%	46.56%	N/A
Action	77.30%	50.37%	-7.91
Action (costs)	77.74%	50.75%	N/A
MaxEnt paths	**78.79%**	**52.98%**	**-6.85**

Table 1: Evaluation results for optimal estimated travel time route, max margin route, Boltzmann Q-value distributions (Action) and Maximum Entropy



Model	Dist. Match	90% Match
Markov (1x1)	62.4%	30.1%
Markov (3x3)	62.5%	30.1%
Markov (5x5)	62.5%	29.9%
Markov (10x10)	62.4%	29.6%
Markov (30x30)	62.2%	29.4%
Travel Time	72.5%	44.0%
Our Approach	**82.6%**	**61.0%**

Table 2: Evaluation results for Markov Model with various grid sizes, time-based model, and our umodel

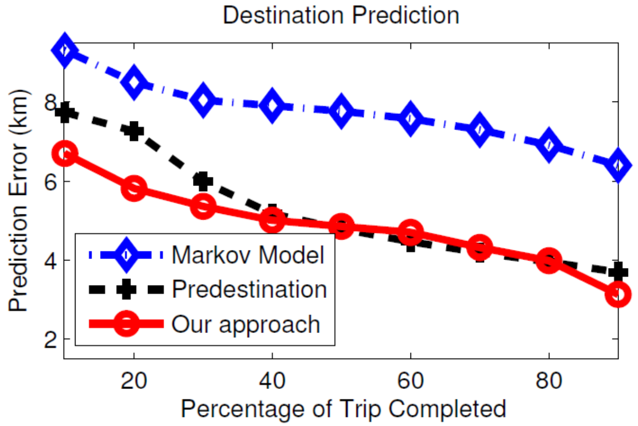


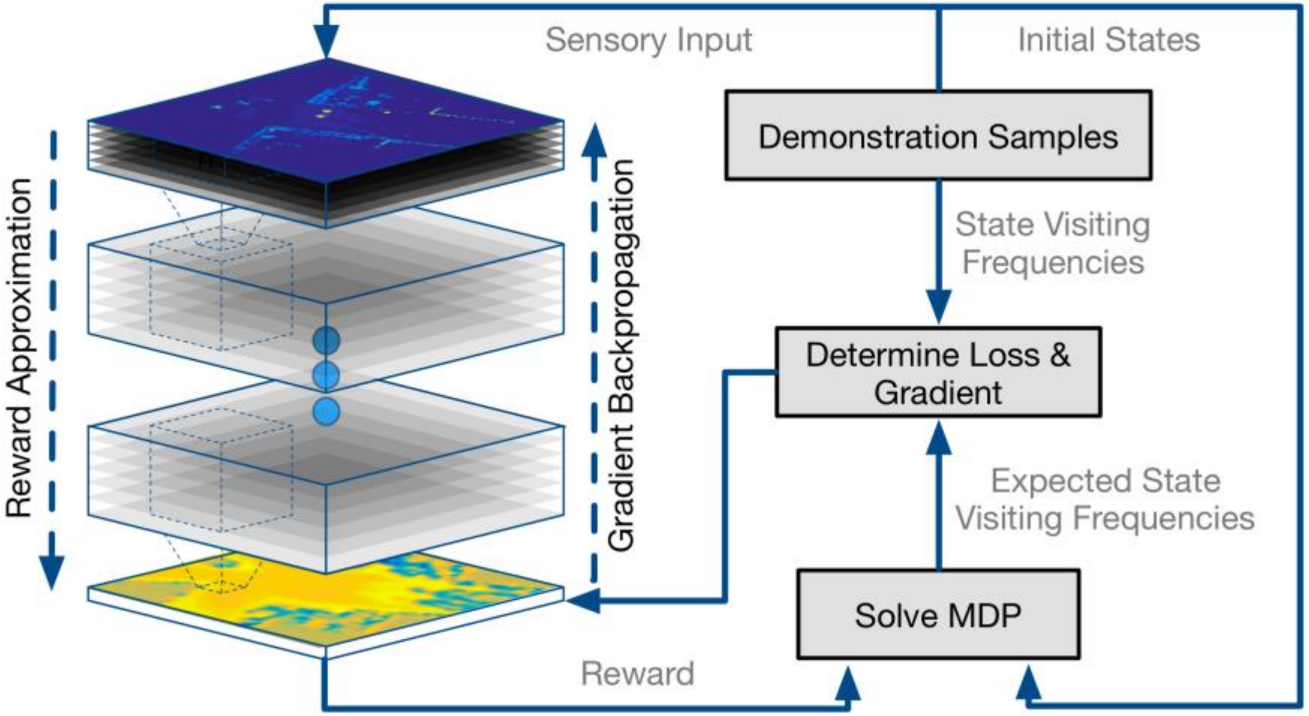
Figure 3: The best Markov Model, Predestination, and our approach's prediction errors

Ziebart, B.D., Maas, A., Bagnell, J.A., & Dey, A.K. (2009). Human Behavior Modeling with Maximum Entropy Inverse Optimal Control. *Proc. of AAAI Spring Symposium on Human Behavior Modeling*, 3931–3936.

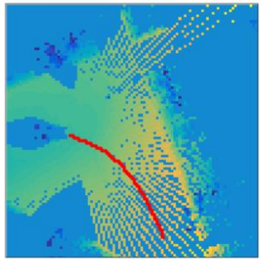
Prediction of pedestrians's behavior

Ziebart, B.D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J.A., Hebert, M., Dey, A.K., & Srinivasa, S. (2009). Planning-based predictions for pedestrians. *Proc. of IEEE/RSJ IROS.*

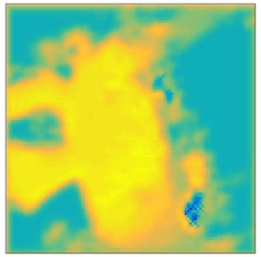
Maximum Entropy Deep Inverse Reinforcement Learning (MEDIRL): Nonlinear MaxEnt-IRL



エキスパート
データ



推定された報酬

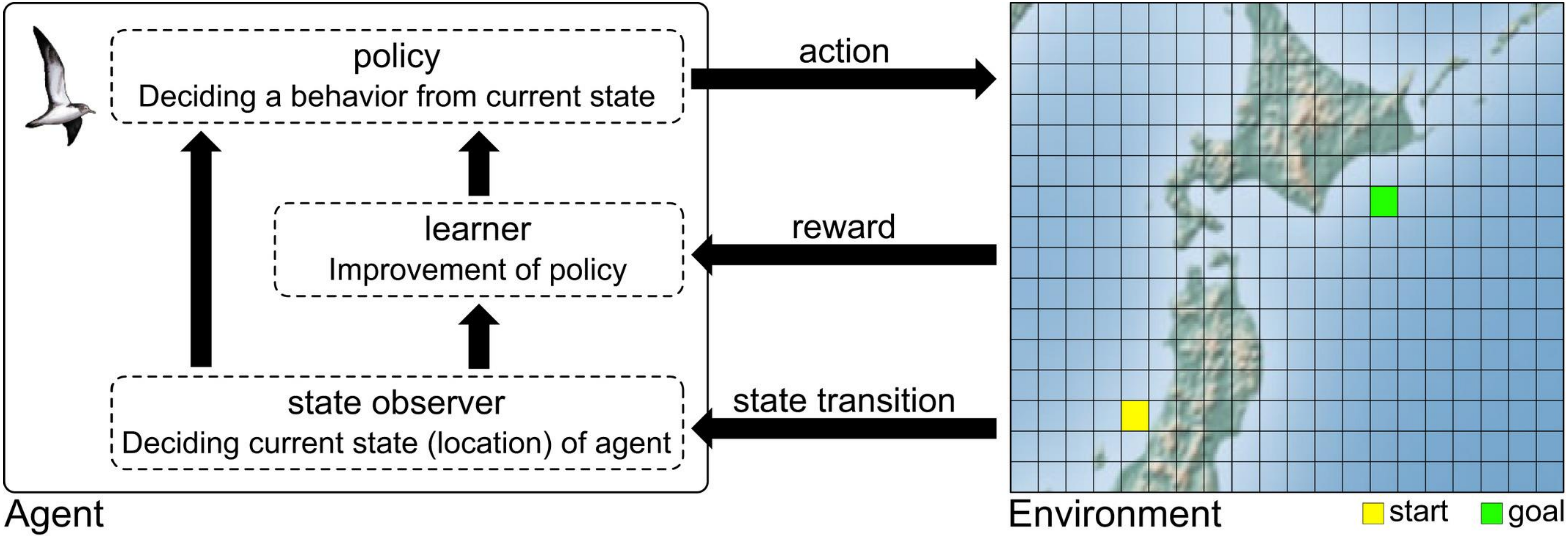


Wulfmeier, M., Wang, D.Z., & Posner, I. (2016). Watch This : Scalable Cost-Function Learning for Path Planning in Urban Environments. *Proc. of IEEE/RSJ IROS*.

Wulfmeier, M., Rao, D., Wang, D.Z., Ondruska, P., & Posner, I. (2017). Large-scale cost function learning for path planning using deep inverse reinforcement learning. *International Journal of Robotics Research*, vol. 36, no. 10: 1073–1087.

Prediction of the most likely route

- Formulate as discrete-state and discrete-action MDP
  - The original state is given by the position (x_t, y_t) and the elapsed time z_t

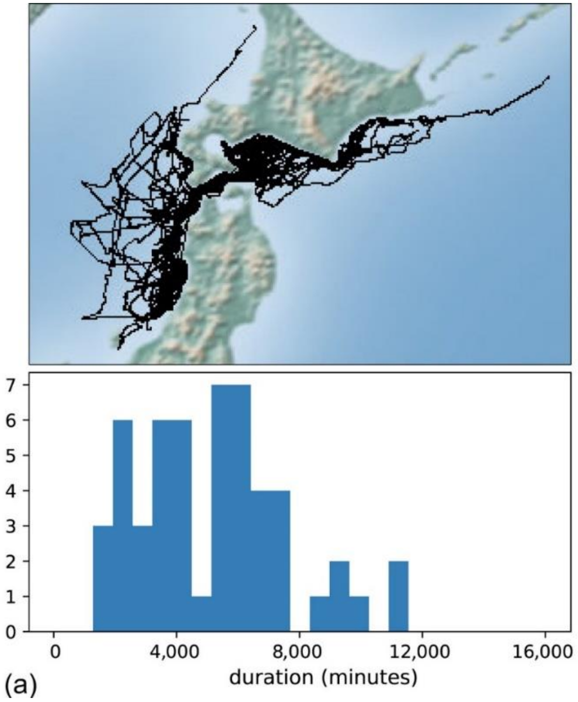


Hirakawa, T., Yamashita, T., Tamaki, T., Fujiyoshi, H., Umezu, Y., Takeuchi, I., Matsumoto, S., and Yoda, K. (2018). Can AI predict animal movements? Filling gaps in animal trajectories using inverse reinforcement learning. Ecosphere.

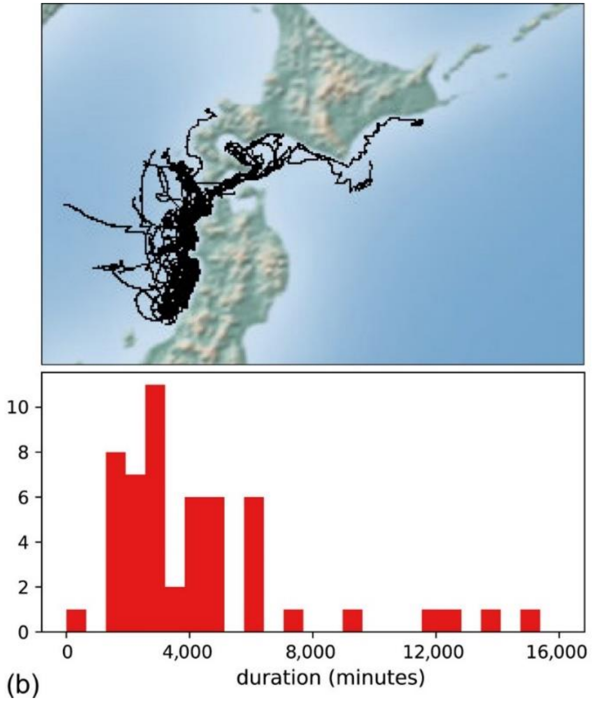
Data

- 106 trajectories (53 males and 53 females)

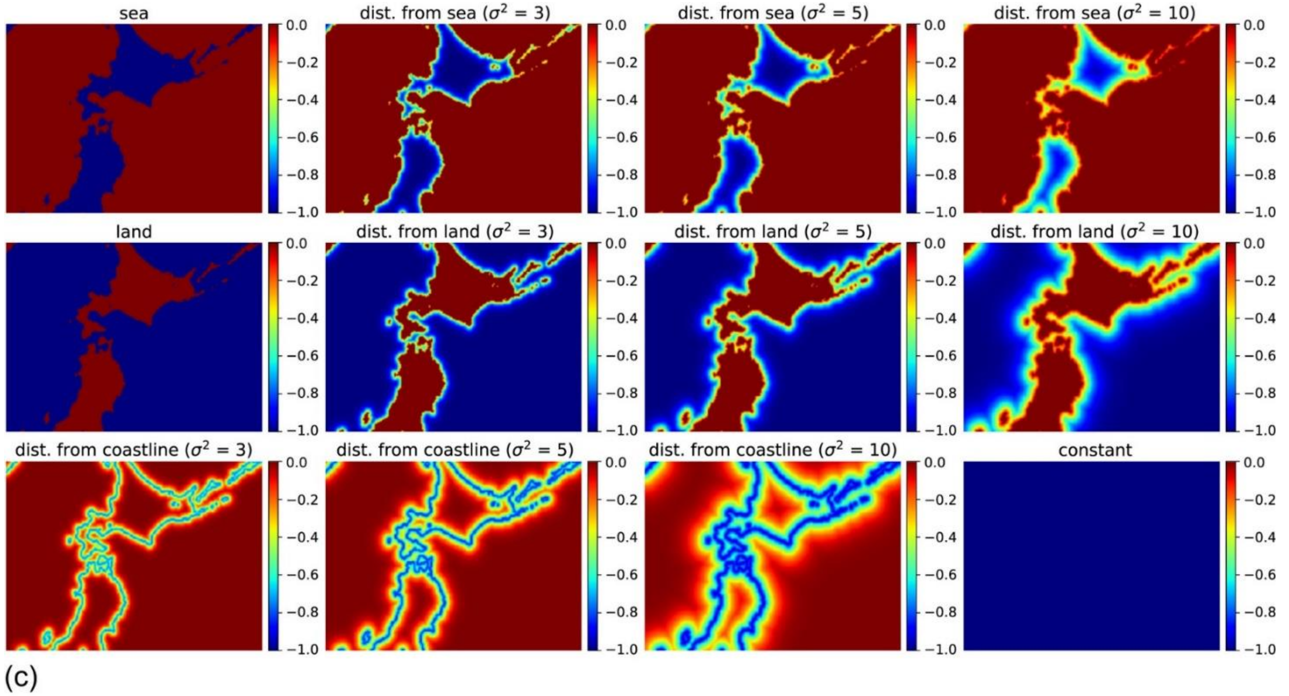
Male shearwater trajectories and the histogram of trajectory duration



Female shearwater trajectories and the histogram of trajectory duration



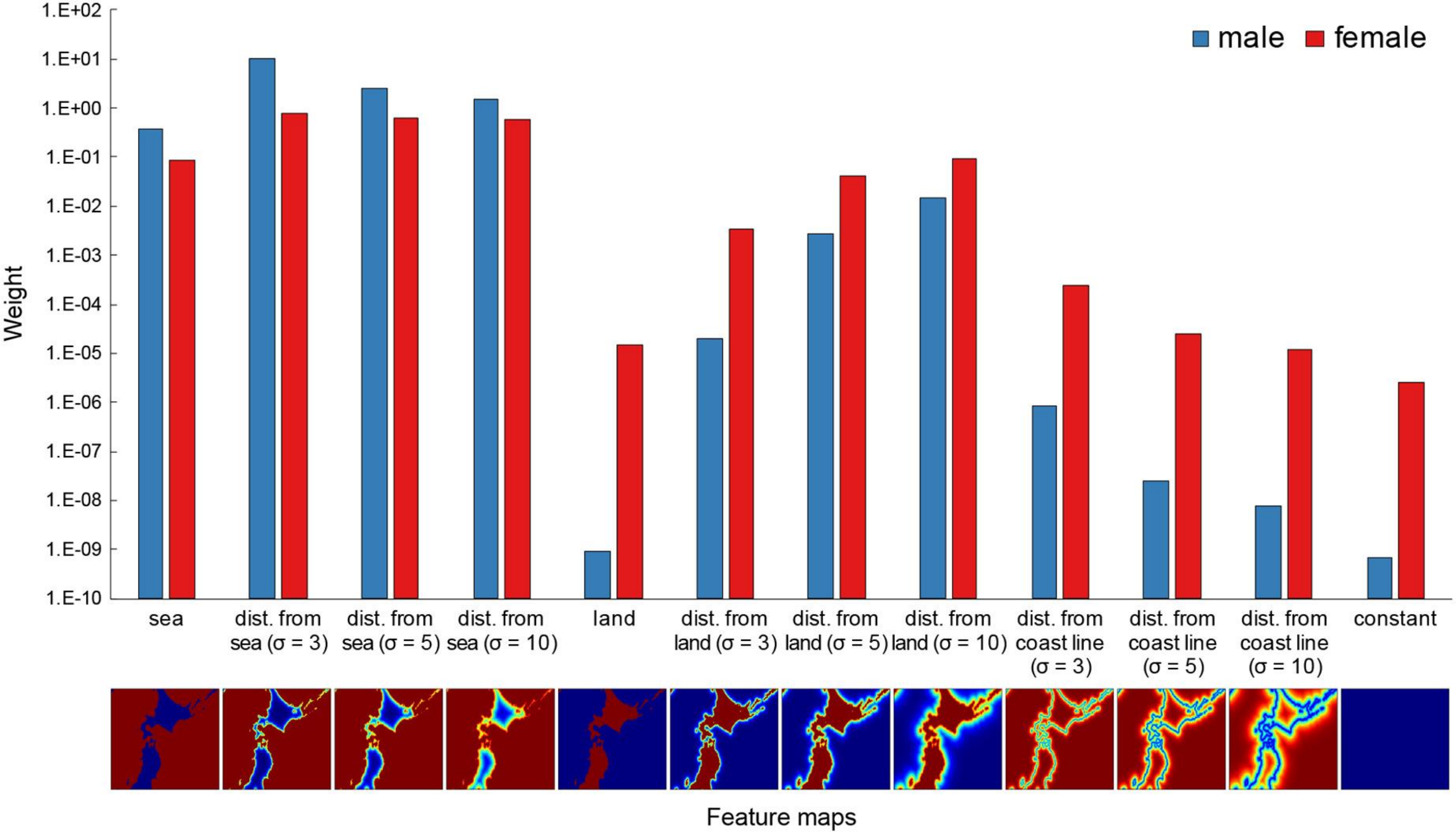
reward features



sea
dist. from sea ($\sigma^2 = 3$)
dist. from sea ($\sigma^2 = 5$)
dist. from sea ($\sigma^2 = 10$)
land
dist. from land ($\sigma^2 = 3$)
dist. from land ($\sigma^2 = 5$)
dist. from land ($\sigma^2 = 10$)
dist. from coastline ($\sigma^2 = 3$)
dist. from coastline ($\sigma^2 = 5$)
dist. from coastline ($\sigma^2 = 10$)
constant
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
(c)

Hirakawa, T., Yamashita, T., Tamaki, T., Fujiyoshi, H., Umezu, Y., Takeuchi, I., Matsumoto, S., and Yoda, K. (2018). Can AI predict animal movements? Filling gaps in animal trajectories using inverse reinforcement learning. Ecosphere.

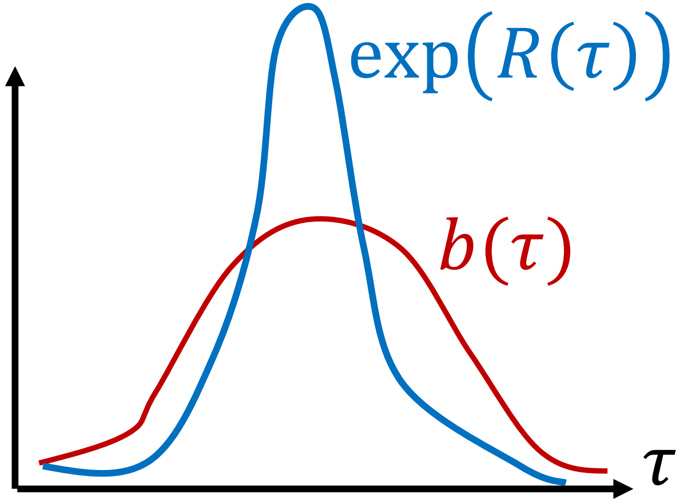
Estimated reward weights



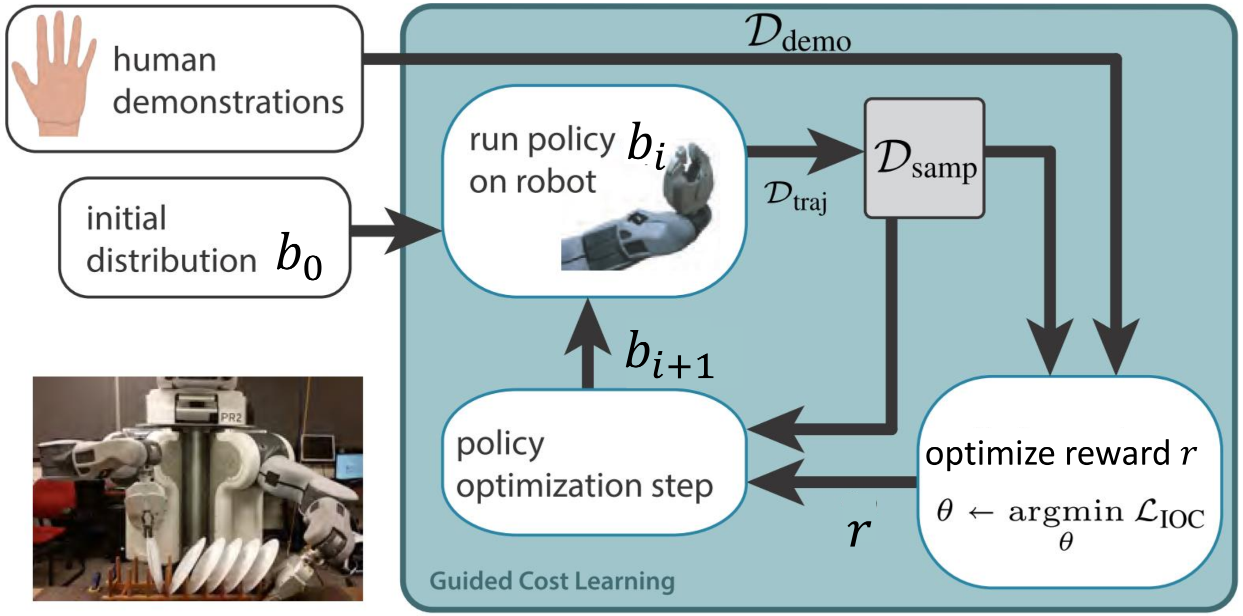
Recap: Maximum Entropy Inverse RL

- Double-loop training process is time-consuming
  - Initialize the reward function
  - Value or policy iteration with the estimated reward function
  - Collect a set of trajectories by the optimal policy
  - Update the reward function
- Simplify the inner loop
  - Initialize a behavior policy
  - Collect a set of trajectories by the behavior policy
  - Update the reward function
  - Update the behavior policy with the estimated reward function

Guided Cost Learning



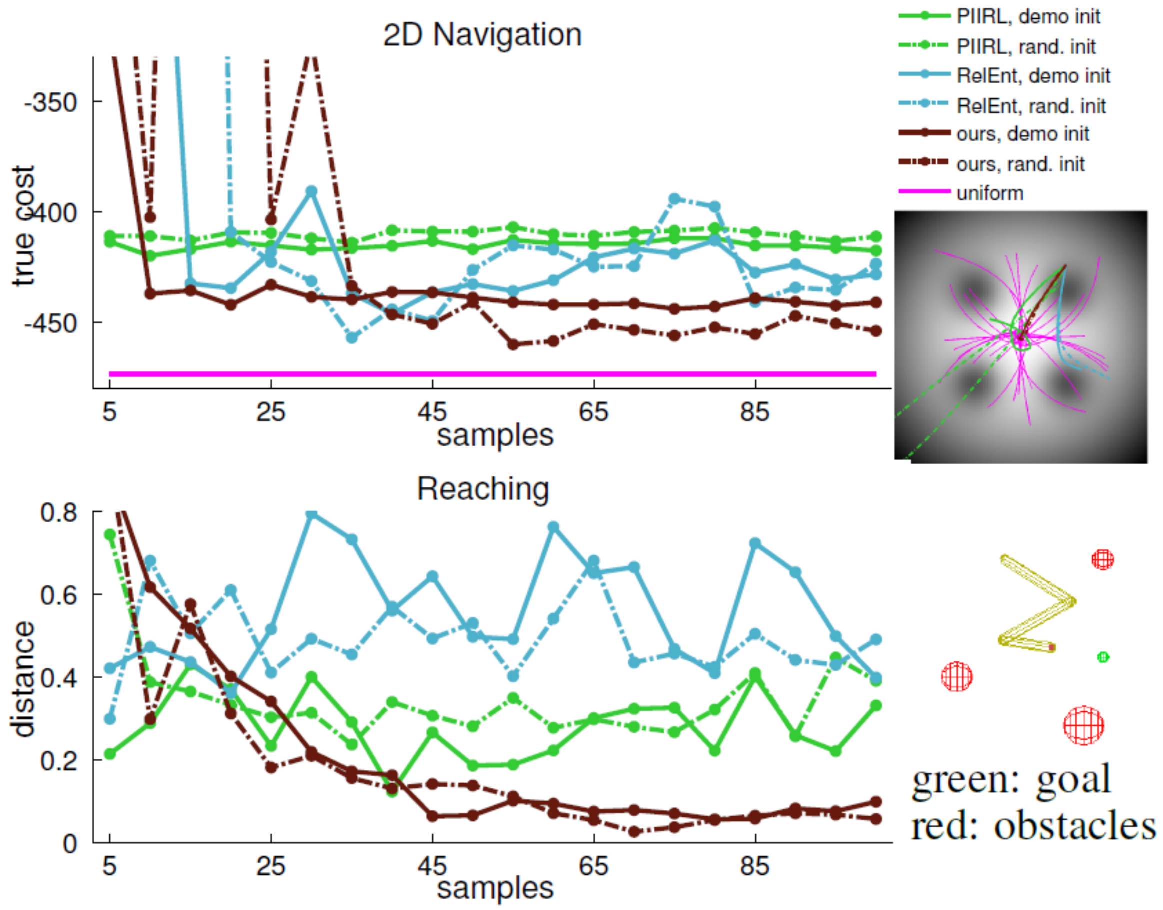
- What is the best $b(\tau)$? ➡ $b(\tau) \propto \exp(R(\tau))$
- Designing $b(\tau)$ is quite difficult because $R(\tau)$ is unknown
- Instead, we can adaptively refine $b(\tau)$ to generate more samples in those regions of the trajectory space that are good according to the current reward function
- Iteration of reward estimation and policy improvement


human demonstrations
$\mathcal{D}_{\text{demo}}$
run policy b_i on robot
$\mathcal{D}_{\text{traj}}$
$\mathcal{D}_{\text{samp}}$
initial distribution b_0
b_{i+1}
policy optimization step
r
optimize reward r
$\theta \leftarrow \underset{\theta}{\text{argmin}}\, \mathcal{L}_{\text{IOC}}$
Guided Cost Learning
PR2

Finn, C., Levine, S., & Abbeel, P. (2016). Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization. *Proc. of ICML*, 49–58. ICML.

Experimental results

- Updating the sampling distribution is important
  - Note: the vertical axis represents the total cost (smaller is better)



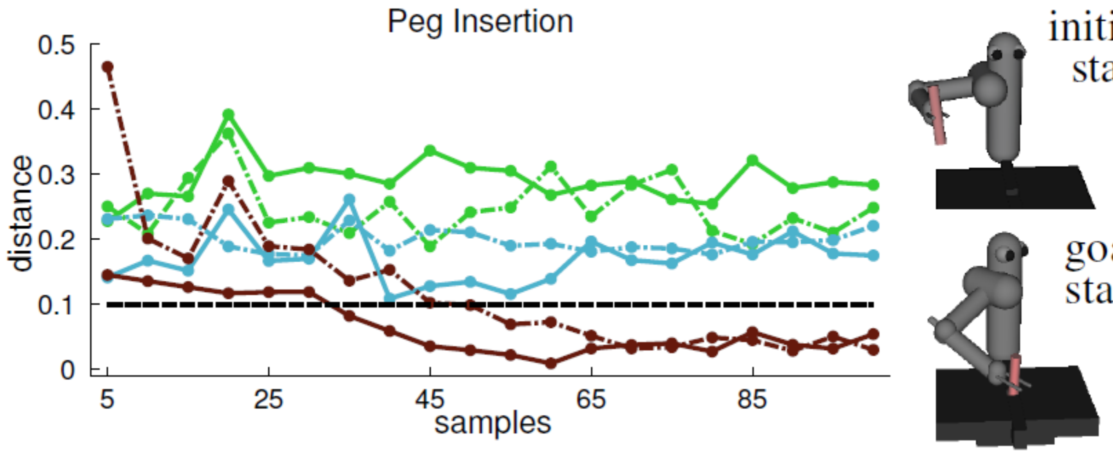
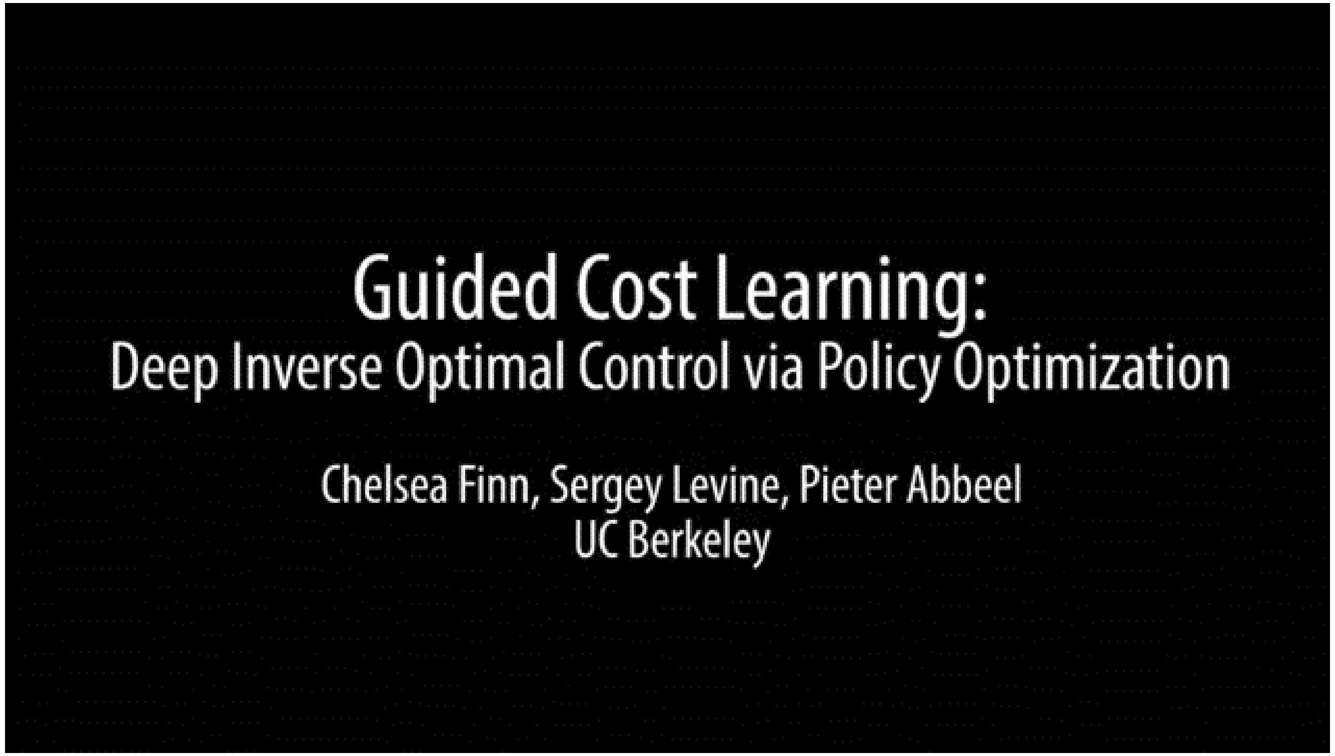


Figure 2. Comparison to prior work on simulated 2D navigation, reaching, and peg insertion tasks. Reported performance is averaged over 4 runs of IOC on 4 different initial conditions . For peg insertion, the depth of the hole is 0.1m, marked as a dashed line. Distances larger than this amount failed to insert the peg.

Real robot experiments

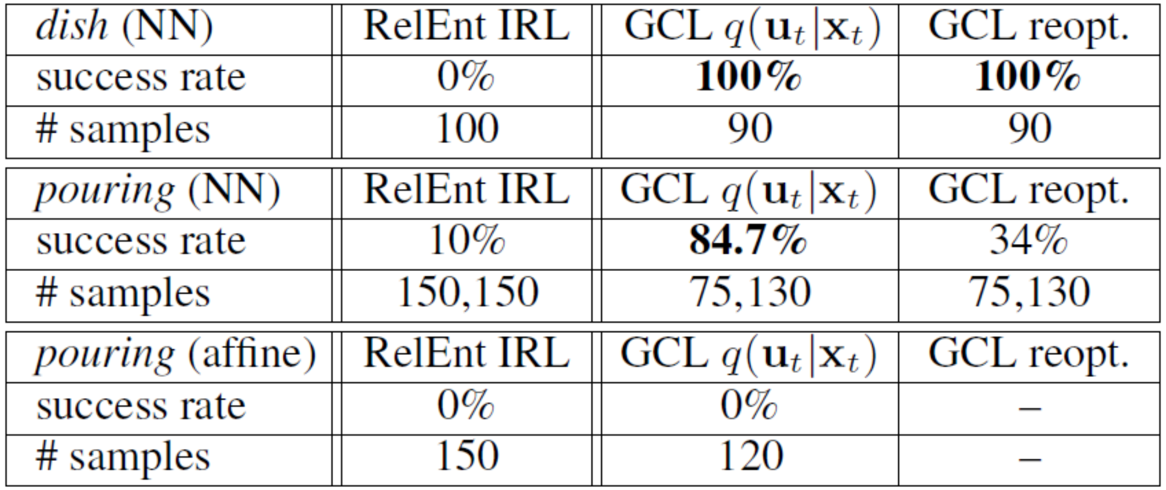


Guided Cost Learning:

Deep Inverse Optimal Control via Policy Optimization

Chelsea Finn, Sergey Levine, Pieter Abbeel

UC Berkeley



dish (NN)	RelEnt IRL	GCL $q(\mathbf{u}_t \mid \mathbf{x}_t)$	GCL reopt.
success rate	0%	**100%**	**100%**
# samples	100	90	90
pouring (NN)	RelEnt IRL	GCL $q(\mathbf{u}_t \mid \mathbf{x}_t)$	GCL reopt.
success rate	10%	**84.7%**	34%
# samples	150,150	75,130	75,130
pouring (affine)	RelEnt IRL	GCL $q(\mathbf{u}_t \mid \mathbf{x}_t)$	GCL reopt.
success rate	0%	0%	–
# samples	150	120	–

Table 1. Performance of guided cost learning (GCL) and relative entropy (RelEnt) IRL on placing a dish into a rack and pouring almonds into a cup. Sample counts are for IOC, omitting those for optimizing the learned cost. An affine cost is insufficient for representing the pouring task, thus motivating using a neural network cost (NN). The pouring task with a neural network cost is evaluated for two positions of the target cup; average performance is reported.

Finn, C., Levine, S., & Abbeel, P. (2016). Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization. *Proc. of ICML*, 49–58. ICML.

Conclusion

- This lecture introduced (deep) inverse reinforcement learning algorithms
- To solve the ill-posed problem, maximum entropy inverse reinforcement learning uses the maximum entropy principle to pick up the distribution
- Guided Cost Learning generated the baseline trajectories using the learned policy with the estimated reward