Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization

Chelsea Finn, Sergey Levine, Pieter Abbeel Proceedings of The 33rd International Conference on Machine Learning, PMLR 48:49-58, 2016.

理論談話会#特別編 (2024/07/20)
M1 Furuhashi Fumihito

Summary

Main Challenge

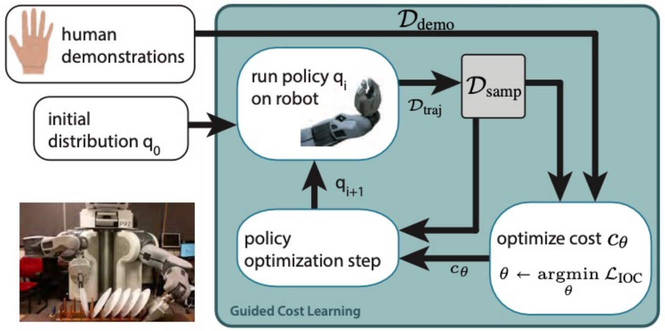
1. The need for **informative features** and **effective regularization** to impose structure on the cost.
2. The difficulty of learning the cost function under **unknown dynamics** for **high-dimensional continuous systems**.

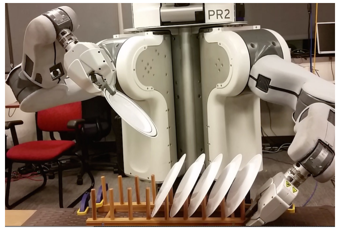
Contribution

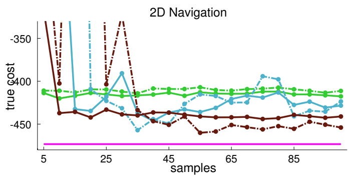
1. This paper presents an algorithm capable of learning **arbitrary nonlinear cost functions**, such as neural networks, without **meticulous feature engineering**.
2. This paper formulates **an efficient sample-based approximation** for MaxEnt IOC.

Validation

- Simulation tasks
- Real-world robotic manipulation problems







0. About this paper

Authors

Chelsea Finn

cbfinn at cs dot stanford dot edu

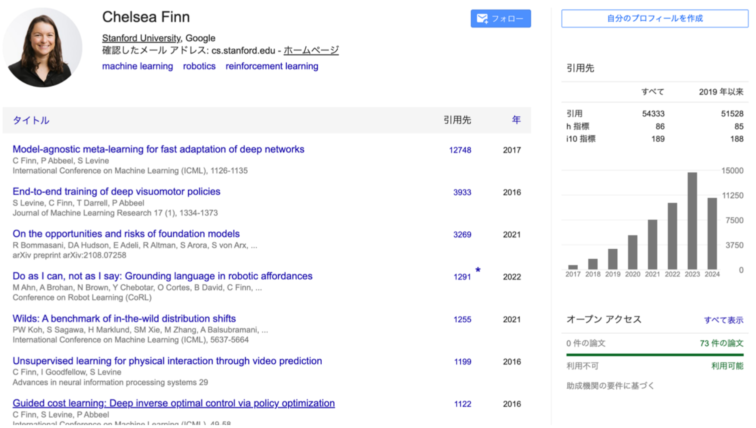
I am an Assistant Professor in Computer Science and Electrical Engineering at Stanford University and co-founder of Pi. My lab, IRIS, studies intelligence through robotic interaction at scale, and is affiliated with SAIL and the ML Group.

I am interested in the capability of robots and other agents to develop broadly intelligent behavior through learning and interaction.

Previously, I completed my Ph.D. in computer science at UC Berkeley and my B.S. in electrical engineering and computer science at MIT. I also spent time at Google as part of the Google Brain team.

Prospective students and post-docs, please see this page.

CV / Bio / PhD Thesis / Google Scholar / Twitter / IRIS Lab



0. About this paper

ICML

International Conference on Machine Learning

2 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

The **International Conference on Machine Learning** (**ICML**) is the leading international academic conference in machine learning. Along with NeurIPS and ICLR, it is one of the three primary conferences of high impact in machine learning and artificial intelligence research.[1] It is supported by the International Machine Learning Society (IMLS). Precise dates vary year to year, but paper submissions are generally due at the end of January, and the conference is generally held the following July. The first ICML was held 1980 in Pittsburgh.[2][3]

Locations [edit]

- ICML 2026 Seoul, South Korea
- ICML 2025 Vancouver, Canada
- ICML 2024 Vienna, Austria
- ICML 2023 Honolulu, United States
- ICML 2022 Baltimore, United States
- ICML 2021 Vienna, Austria (virtual conference)
- ICML 2020 Vienna, Austria (virtual conference)
- ICML 2019 Los Angeles, United States
- ICML 2018 Stockholm, Sweden
- ICML 2017 Sydney, Australia
- ICML 2016 New York City, United States
- ICML 2015 Lille, France

Part of a series on
Machine learning and data mining

Paradigms [show]
Problems [show]
Supervised learning [show]
(**classification** · **regression**)
Clustering [show]
Dimensionality reduction [show]
Structured prediction [show]
Anomaly detection [show]
Artificial neural network [show]
Reinforcement learning [show]
Learning with humans [show]
Model diagnostics [show]
Mathematical foundations [show]
Machine-learning venues [hide]
ECML PKDD · NeurIPS · **ICML** · ICLR · IJCAI · ML · JMLR
Related articles [show]

V · T · E

1. Introduction

- **Reinforcement Learning Challenges**
  - Difficult to define a cost function that encodes the correct task and can be optimized effectively.
  - Cost shaping often used to solve complex real-world problems (Ng et al., 1999).
- **Inverse Optimal Control (IOC)**
  - IOC and inverse reinforcement learning (IRL) learn a cost function directly from expert demonstrations (Ng et al., 2000; Abbeel & Ng, 2004; Ziebart et al., 2008).
  - Challenges: Many costs induce the same behavior, and solving the forward problem (finding an optimal policy) in the inner loop of iterative cost optimization.
- **Proposed Approach**
  - Use expressive, nonlinear function approximators like neural networks to represent the cost.
  - Reduces the engineering burden and allows learning complex cost functions without hand-designed features.
- **Advantages**
  - Can handle unknown dynamics and high-dimensional systems.
  - Combines policy learning and cost learning, making it practical and efficient.
  - Achieves good global costs even for complex tasks.
- **Key Contributions**
  - Simultaneous policy and cost learning from demonstrations.
  - Guided cost learning algorithm based on policy optimization over a good region of the space.
  - Outperforms prior methods in simulated benchmarks and real-world tasks without manually designed cost functions.

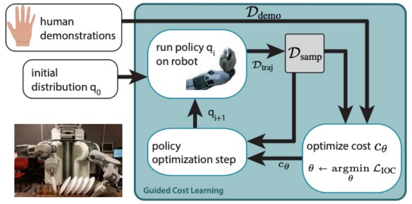


Figure 1. Right: Guided cost learning uses policy optimization to adaptively sample trajectories for estimating the IOC partition function. Bottom left: PR2 learning to gently place a dish in a plate rack.

2. Related Work

Issue #1 in IOC (or IRL)

The set of demonstrations is **not necessarily optimal**

- Maximum margin formulations
- Probabilistic models

2. Related Work

Issue #1 in IOC (or IRL)

The set of demonstrations is **not necessarily optimal**

- Maximum margin formulations
- **Probabilistic models**
    - **Maximum entropy IOC model**

There is still a great deal of ambiguity…

1. More detailed features
2. More powerful regularization

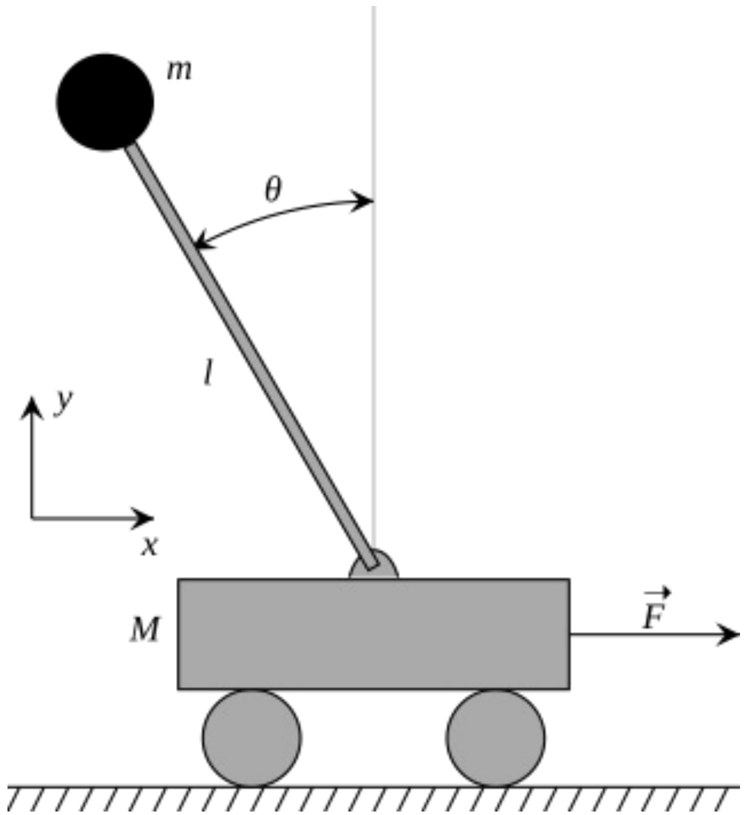
2. Related Work

Issue #2 in IOC (or IRL)

Necessity of solving a variant of the forward control problem

Solving the forward control problem

- Requires knowledge of the system dynamics to solve the problem
- This paper’s method is based on the principle of maximum entropy which can handle **unknown dynamics**



2. Related Work

Issue #2 in IOC (or IRL)

Necessity of solving a variant of the forward control problem

Solving the forward control problem

- Requires knowledge of the system dynamics to solve the problem
- This paper’s method is based on the principle of maximum entropy which can handle **unknown dynamics**

Comparing with other sample base methods…

- Adapts the sampling distribution using policy optimization
- This adaptation is crucial for obtaining good results

2. Related Work

Summary

This paper’s method combines key features for effective algorithms

Manages high-dimensional, complex systems Applicable to real torque-controlled robotic arms	**Learns complex, expressive cost functions** Utilizes neural networks
Eliminates the need for hand-engineering of cost features	**Handles unknown dynamics** Crucial for real-world robotic tasks

3. Preliminaries and Overview

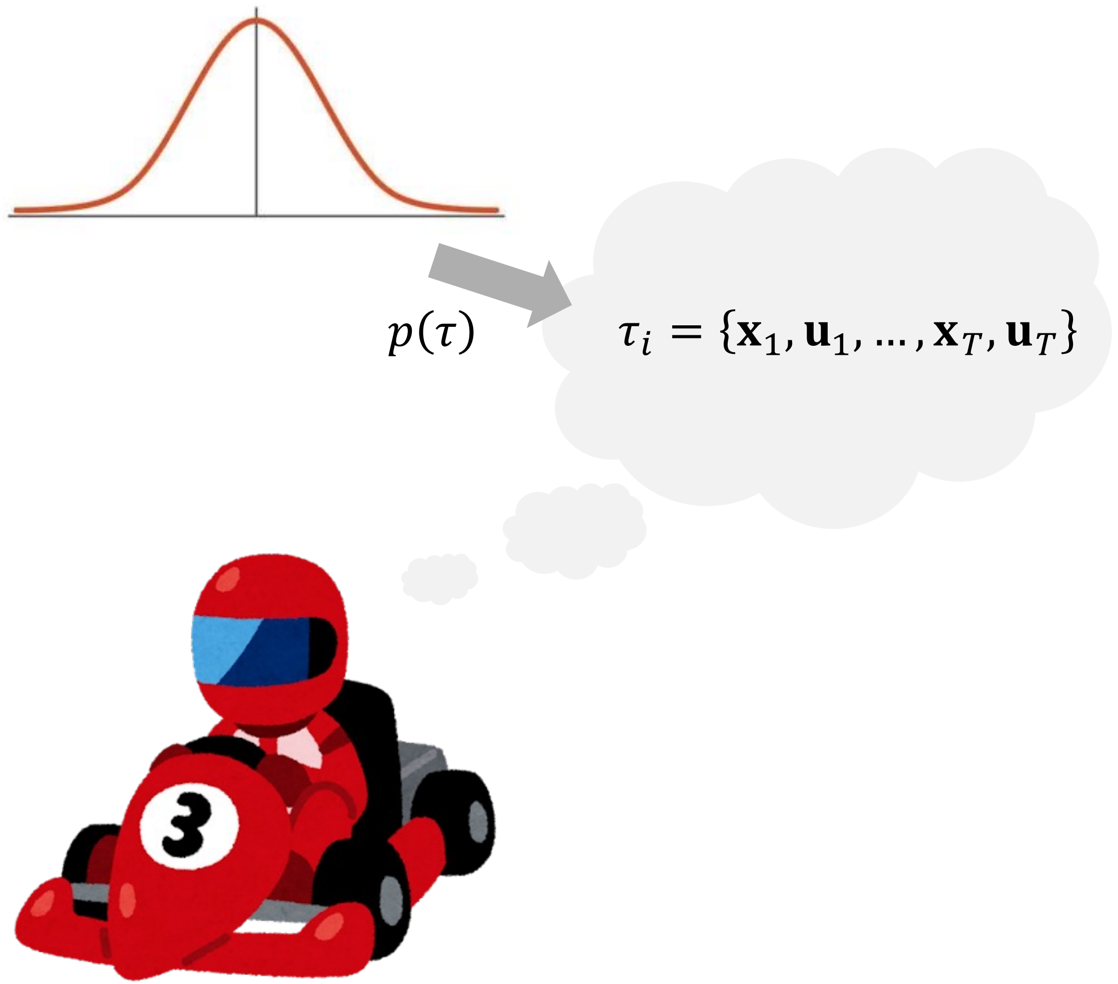
Probabilistic Max-Ent IOC

(Ziebart et al., 2008)

- Assumes that experts act probabilistically and nearly optimally with respect to an unknown cost function
- Assumes that the expert samples the demonstrated trajectory $\{\tau_i\}$ from distribution

$$p(\tau) = \frac{1}{Z}\exp\left(-c_\theta(\tau)\right)$$

  - $\tau = \{\mathbf{x}_1, \mathbf{u}_1, \dots, \mathbf{x}_T, \mathbf{u}_T\}$
    Trajectory sample of expert demonstrations
  - $-c_\theta(\tau) = \sum_t c_\theta(\mathbf{x}_t, \mathbf{u}_t)$
    Unknown cost function characterized by parameters θ
  - $\mathbf{x}_t, \mathbf{u}_t$
    State/Input at time t



3. Preliminaries and Overview

Probabilistic Max-Ent IOC

(Ziebart et al., 2008)

- Assumes that experts act probabilistically and nearly optimally with respect to an unknown cost function
- Assumes that the expert samples the demonstrated trajectory $\{\tau_i\}$ from distribution

$$p(\tau) = \frac{1}{Z}\exp\big(-c_\theta(\tau)\big)$$

  - $\tau = \{\mathbf{x}_1, \mathbf{u}_1, \dots, \mathbf{x}_T, \mathbf{u}_T\}$
    Trajectory sample of expert demonstrations
  - $-c_\theta(\tau) = \sum_t c_\theta(\mathbf{x}_t, \mathbf{u}_t)$
    Unknown cost function characterized by parameters θ
  - $\mathbf{x}_t, \mathbf{u}_t$
    State/Input at time t

Challenges and Solutions

- Calculating the partition function Z is difficult
  - Ziebart (2008) first calculated Z exactly using dynamic programming
  - Laplace Approximation (Levine & Koltun, 2012)
  - Value Function Approximation (Huang & Kitani, 2014)
  - **Sampling** (Boularias et al., 2011)

Significance

- Can perform IOC even with **unknown system dynamics**!
  Crucial for robotics interacting with objects of unknown physical properties

4. Guided Cost Learning

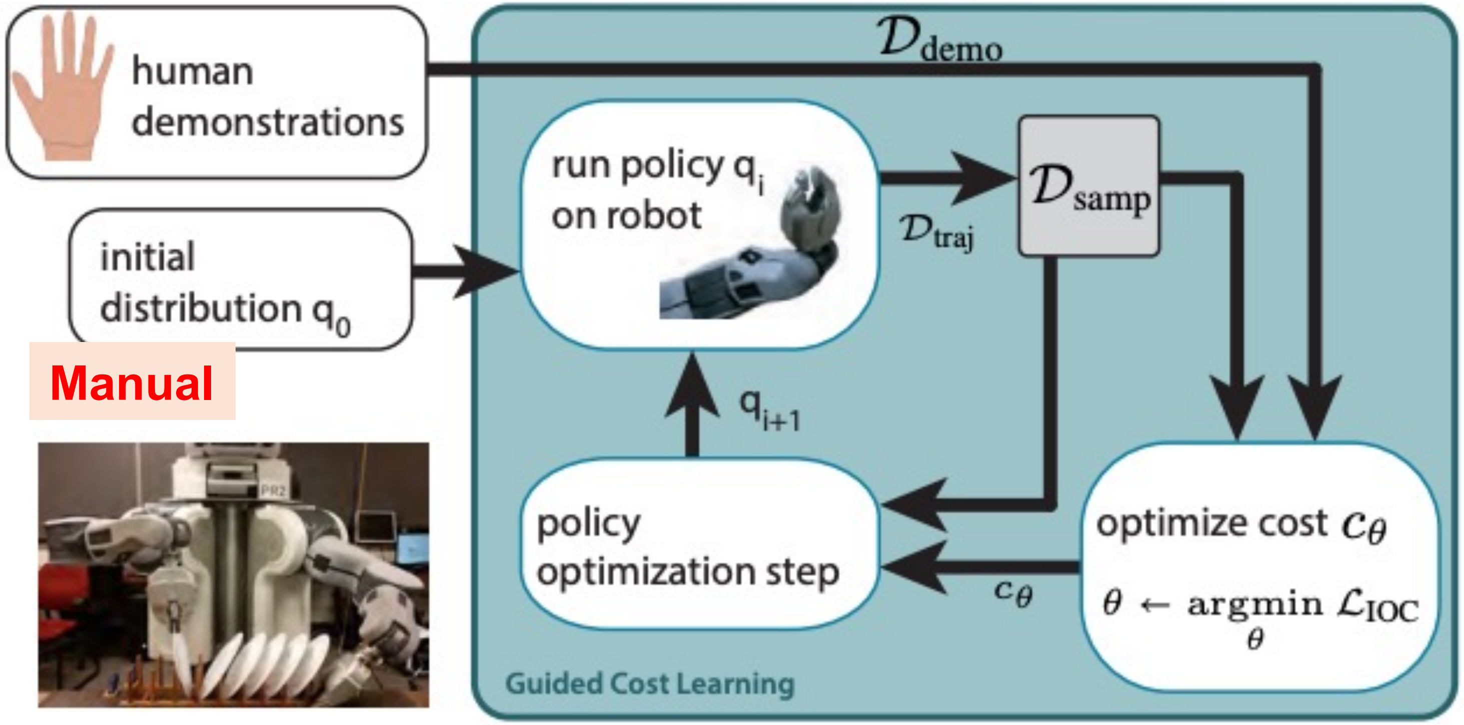


Figure 1. Right: Guided cost learning uses policy optimization to adaptively sample trajectories for estimating the IOC partition function. Bottom left: PR2 learning to gently place a dish in a plate rack.

4-1. Sample-Based Approach to Maximum Entropy IOC

Sample-Based Approach to Max-Ent IOC

- The partition function $Z = \int \exp\left(c_\theta(\tau)\right) d\tau$ is estimated using a background distribution $q(\tau)$
  - Prior methods:
    - A linear representation for the cost function to simplify the cost learning problem (e.g., Boularias et al., 2011)
  - This paper:
    - Generalizes and uses a **non-linear** parameterized cost function
  - The negative log-likelihood of $p(\tau)$ is given by

$$\mathcal{L}_{IOC}(\theta) = \frac{1}{N} \sum_{\tau_i \in D_{demo}} c_\theta(\tau_i) + \log Z$$

$$\mathcal{L}_{IOC}(\theta) \approx \frac{1}{N} \sum_{\tau_i \in D_{demo}} c_\theta(\tau_i) + \log\left[\frac{1}{M} \sum_{\tau_j \in D_{samp}} \frac{\exp\left(-c_\theta(\tau_j)\right)}{q(\tau_j)}\right]$$

- D_{demo}: Set of N demonstrated trajectories
- D_{samp}: Set of M trajectories sampled from the background distribution
- q: Often manually chosen as the demonstration distribution or a uniform distribution

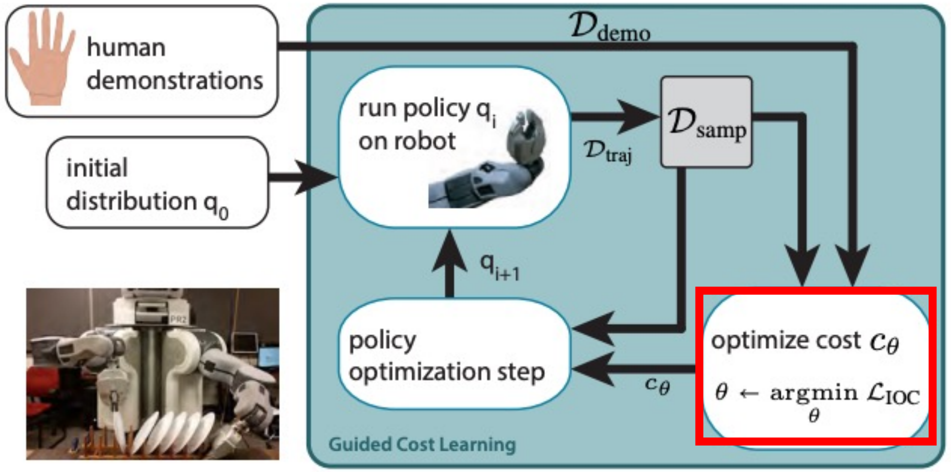


Figure 1. Right: Guided cost learning uses policy optimization to adaptively sample trajectories for estimating the IOC partition function. Bottom left: PR2 learning to gently place a dish in a plate rack.

4-1. Sample-Based Approach to Maximum Entropy IOC

Sample-Based Approach to Max-Ent IOC

- The partition function $Z = \int \exp\left(c_\theta(\tau)\right) d\tau$ is estimated using a background distribution $q(\tau)$
  - Prior methods:
    - A linear representation for the cost function to simplify the cost learning problem (e.g., Boularias et al., 2011)
  - This paper:
    - Generalizes and uses a **non-linear** parameterized cost function
  - The negative log-likelihood of $p(\tau)$ is given by

$$\mathcal{L}_{IOC}(\theta) = \frac{1}{N} \sum_{\tau_i \in D_{demo}} c_\theta(\tau_i) + \log Z$$

$$\mathcal{L}_{IOC}(\theta) \approx \frac{1}{N} \sum_{\tau_i \in D_{demo}} c_\theta(\tau_i) + \log \left[\frac{1}{M} \sum_{\tau_j \in D_{samp}} \frac{\exp\left(-c_\theta(\tau_j)\right)}{q(\tau_j)} \right]$$

    - D_{demo}: Set of N demonstrated trajectories
    - D_{samp}: Set of M trajectories sampled from the background distribution
    - q: Often manually chosen as the demonstration distribution or a uniform distribution

- To find the gradient of this objective function with respect to θ, define $w_j = \frac{\exp\left(-c_\theta(\tau_j)\right)}{q(\tau_j)}$ (so that $Z = \sum_j w_j$)
- The gradient is

$$\frac{d\mathcal{L}_{IOC}}{d\theta} = \frac{1}{N} \sum_{\tau_i \in D_{demo}} \frac{dc_\theta}{d\theta}(\tau_i) - \frac{1}{Z} \sum_{\tau_j \in D_{samp}} \frac{dc_\theta}{d\theta}(\tau_j)$$

- If the cost function is approximated by a neural network:
  - Backpropagate $\frac{1}{N}$ for $\tau_i \in D_{demo}$
  - Backpropagate $-\frac{w_j}{Z}$ for $\tau_j \in D_{samp}$

4-2. Adaptive Sampling via Policy Optimization

- Choosing the Background Sample Distribution $q(\tau)$for Estimating $\mathcal{L}_{IOC}$ Is Crucial for the Success of Sample-Based IOC Algorithms
- The optimal importance sampling distribution to estimate the partition function $Z = \int \exp(c_\theta(\tau))\, d\tau$ is $q(\tau) \propto |\exp(-c_\theta(\tau))| = \exp(-c_\theta(\tau))$
  - However, designing a single background distribution $q(\tau)$ is **difficult** when the cost function c_θ is unknown
  - Instead, adaptively improving $q(\tau)$ using the current cost function $c_\theta\ (\tau)$ generates more samples in specific regions of the trajectory space
  - To Achieve This
    - IOC Optimization
      - Find the cost function that maximizes the likelihood of the demonstrated trajectories
    - Policy Optimization
      - Improve the trajectory background distribution $q(\tau)$ with respect to the current cost
  - **Alternate between these two optimizations**
  - Since policy optimization can handle **unknown system dynamics**, adopt the method by Levine & Abbeel (2014), which iteratively fits time-varying linear dynamics using samples from the system dynamics.

Algorithm 1 Guided cost learning

1: Initialize $q_k(\tau)$ as either a random initial controller or from demonstrations
2: **for** iteration $i = 1$ to I **do**
3: Generate samples $\mathcal{D}_{\text{traj}}$ from $q_k(\tau)$
4: Append samples: $\mathcal{D}_{\text{samp}} \leftarrow \mathcal{D}_{\text{samp}} \cup \mathcal{D}_{\text{traj}}$
5: Use $\mathcal{D}_{\text{samp}}$ to update cost c_θ using Algorithm 2
6: Update $q_k(\tau)$ using $\mathcal{D}_{\text{traj}}$ and the method from (Levine & Abbeel, 2014) to obtain $q_{k+1}(\tau)$
7: **end for**
8: **return** optimized cost parameters θ and trajectory distribution $q(\tau)$

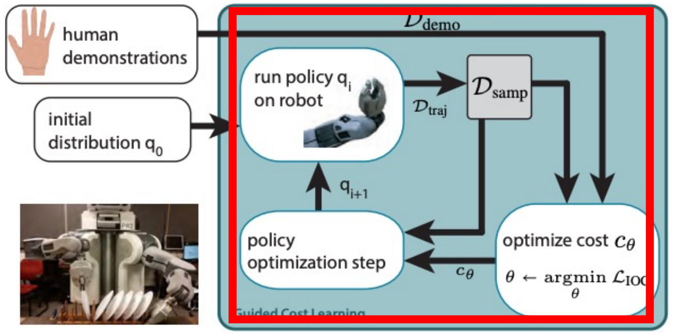


Figure 1. Right: Guided cost learning uses policy optimization to adaptively sample trajectories for estimating the IOC partition function. Bottom left: PR2 learning to gently place a dish in a plate rack.

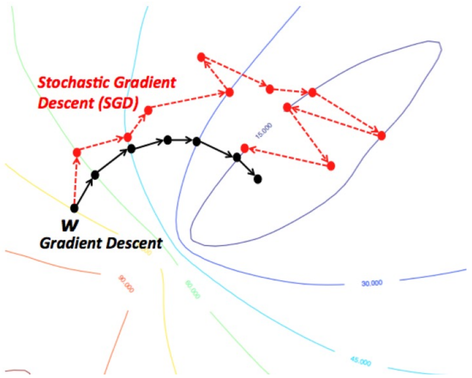
4-3. Cost Optimization and Importance Weights

Optimizing the IOC Objective Function

- The IOC objective function can be optimized using standard nonlinear optimization methods and the gradient $\frac{d\mathcal{L}_{\text{IOC}}}{d\theta}$
- For neural networks, **stochastic gradient methods** can be used
- It is straightforward if the objective function is factored over samples, but the partition function here is not
- In this paper, the objective function can be optimized by **sampling subsets** of samples from demonstrations and the background distribution in each iteration

Algorithm 1 Guided cost learning

Initialize $q_k(\tau)$ as either a random initial controller or from demonstrations
for iteration $i = 1$ to I **do**
  Generate samples $\mathcal{D}_{\text{traj}}$ from $q_k(\tau)$
  Append samples: $\mathcal{D}_{\text{samp}} \leftarrow \mathcal{D}_{\text{samp}} \cup \mathcal{D}_{\text{traj}}$
  $\boxed{\text{Use } \mathcal{D}_{\text{samp}} \text{ to update cost } c_\theta \text{ using Algorithm 2}}$
  Update $q_k(\tau)$ using $\mathcal{D}_{\text{traj}}$ and the method from (Levine & Abbeel, 2014) to obtain $q_{k+1}(\tau)$
end for
return optimized cost parameters θ and trajectory distribution $q(\tau)$



Algorithm 2 Nonlinear IOC with stochastic gradients

for iteration $k = 1$ to K **do**
  Sample demonstration batch $\hat{\mathcal{D}}_{\text{demo}} \subset \mathcal{D}_{\text{demo}}$
  Sample background batch $\hat{\mathcal{D}}_{\text{samp}} \subset \mathcal{D}_{\text{samp}}$
  Append demonstration batch to background batch: $\hat{\mathcal{D}}_{\text{samp}} \leftarrow \hat{\mathcal{D}}_{\text{demo}} \cup \hat{\mathcal{D}}_{\text{samp}}$
  Estimate $\frac{d\mathcal{L}_{\text{IOC}}}{d\theta}(\theta)$ using $\hat{\mathcal{D}}_{\text{demo}}$ and $\hat{\mathcal{D}}_{\text{samp}}$
  Update parameters θ using gradient $\frac{d\mathcal{L}_{\text{IOC}}}{d\theta}(\theta)$
end for
return optimized cost parameters θ

4-3. Cost Optimization and Importance Weights

Optimizing the IOC Objective Function

- The IOC objective function can be optimized using standard nonlinear optimization methods and the gradient $\frac{d\mathcal{L}_{\mathrm{IOC}}}{d\theta}$
- For neural networks, **stochastic gradient methods** can be used
- It is straightforward if the objective function is factored over samples, but the partition function here is not
- In this paper, the objective function can be optimized by **sampling subsets** of samples from demonstrations and the background distribution in each iteration

Importance Sampling for Partition Function Estimation

- Importance sampling is required for estimating the partition function
- Previous works (Kalakrishnan et al., 2013; Aghasadeghi & Bretl, 2011) suggest dropping importance weights, but this generates inconsistent likelihood estimates and poor cost functions
- To evaluate importance weights, construct a composite distribution as samples are drawn from **multiple distributions**
- When samples are drawn from k distribution $q_1(\tau), \dots, q_k(\tau)$, a consistent estimate of the expectation of function $f(\tau)$ under a uniform distribution is:

$$E[f(\tau)] \approx \frac{1}{M}\sum_{\tau_j} \frac{1}{\sum_k q_k(\tau_j)} f(\tau_j)$$

Accordingly, the importance weight is:

$$z_j = \left[\frac{1}{\sum_k q_k(\tau_j)}\right]^{-1}$$

Objective Function:

$$\mathcal{L}_{IOC}(\theta) = \frac{1}{N}\sum_{\tau_i \in D_{demo}} c_\theta(\tau_i) + \log\left[\frac{1}{M}\sum_{\tau_j \in D_{samp}} z_j \exp\left(-c_\theta(\tau_j)\right)\right]$$

4-4. Learning Costs and Controllers

Algorithm Capabilities

- Produces both a cost function $c_\theta(\mathbf{x}_t, \mathbf{u}_t)$ and a controller $q(\mathbf{u}_t|\mathbf{x}_t)$.
- Can execute desired behaviors directly using the generated controller.

Contrast with Previous Methods

- Unlike many previous IOC and IRL methods, our approach simultaneously learns a cost and optimizes the policy for new task instances without demonstrations.

Advantages

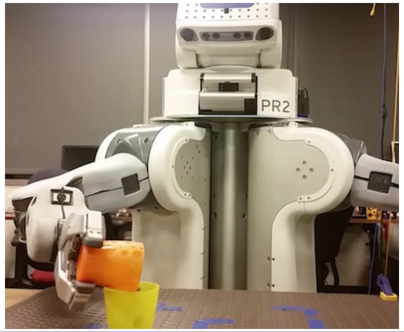
- Uses knowledge that demonstrations are near-optimal under some unknown cost function.
- Similar to recent IOC work by direct loss minimization (Doerr et al., 2015).

Policy Optimization

- Learned cost function can optimize policies for new task instances without additional cost learning.
- In challenging tasks, continuous policy learning with IOC outperforms using a single learned cost.

Hypothesis

- Training on new task instances provides better cost function and reduces overfitting.
- Demonstrations cover limited task variations; new samples improve task execution understanding.



Changes positions of a cup

5. Representation and Regularization

- **Expressiveness**
  - Affine cost functions lack sufficient expressiveness (Section 6.2).
  - Neural network parameterizations are useful for learning visual representations from **raw image pixels**
  - Uses an **unsupervised visual feature learning method** (Finn et al., 2016) to learn cost functions dependent on visual input
- **Challenges of Nonlinear Cost Functions**
  - Introduce significant **model complexity.**
  - Requires **regularization** to mitigate overfitting.
- **Existing Regularization Methods**
  - Penalize the l_1 or l_2 norm of the cost parameters (Ziebart, 2010; Kalakrishnan et al., 2013).
  - Insufficient for high-dimensional nonlinear cost functions.

Proposed Regularization Methods:

- **Local Change Rate Regularization (General)**
  - Encourages the cost of demo and sample trajectories to change at a constant rate.
  - Reduces high-frequency fluctuations indicative of overfitting and promotes cost redistribution.
  - Formula:

$$g_{lcr}(\tau) = \sum_{x_t \in \tau} [(c_\theta(x_{t+1}) - c_\theta(x_t)) - (c_\theta(x_t) - c_\theta(x_{t-1}))]^2$$

- **Monotonicity Regularization (Local)**
  - Tailored for one-shot episodic tasks.
  - Uses squared hinge loss to ensure cost of demo trajectories decreases monotonically over time.
  - Assumes tasks progress monotonically towards goals on a potentially nonlinear manifold.
  - Formula:

$$g_{mono}(\tau) = \sum_{x_t \in \tau} [\max(0, (c_\theta(x_t) - c_\theta(x_{t-1}) - 1)]^2$$

6-1. Simulated Comparisons

- **Tasks**
  - **2D Navigation**
  - **3-Link Arm**
  - **3D Peg Insertion**
- **Methodology**
  - Compared guided cost learning with prior sample-based methods on task performance and sample complexity.
  - Used MuJoCo physics simulator for experiments.
  - Sampled from different initializations and regularizations (detailed in Appendix E).
- **Sampling Methods**
  - Used suboptimal samples for estimating the partition function.
  - Samples obtained either by a baseline random controller or by fitting a linear-Gaussian controller to demonstrations.
- **Key Findings**
  - More complex cost function required for precise tasks like peg insertion.
  - Demonstrations and additional samples provided **better learning for complex tasks.**
  - Prior methods required additional samples, but did not improve performance with more samples from the same distribution.
  - Proposed method effectively **handled complex, high-dimensional tasks.**

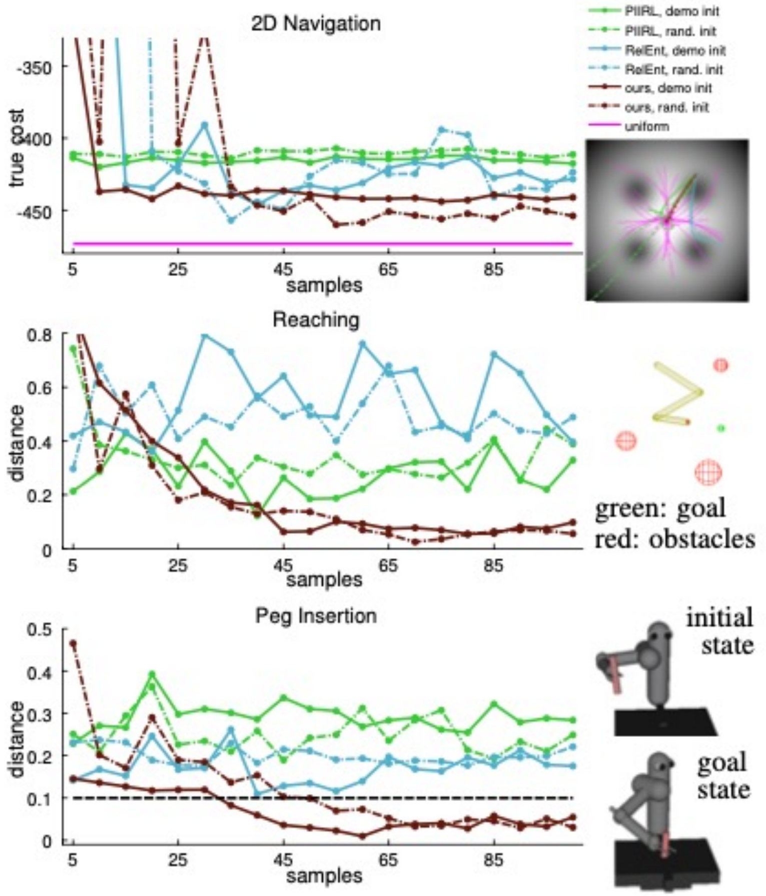


Figure 2. Comparison to prior work on simulated 2D navigation, reaching, and peg insertion tasks. Reported performance is averaged over 4 runs of IOC on 4 different initial conditions . For peg insertion, the depth of the hole is 0.1m, marked as a dashed line. Distances larger than this amount failed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

6-2. Real-world robotics

- **Tasks**
  - Placing a Plate into a Dish Rack
  - Pouring Almonds from One Cup to Another
- **Results**
  - **Dish Rack Task**
    - Neural network-based method achieved 100% success rate.
    - Relative entropy IRL failed (0% success).
  - **Pouring Task**
    - Neural network method had an 84.7% success rate; affine cost function failed.
    - Neural network method required fewer samples than relative entropy IRL.
  - **Generalizability**
    - Learned cost used to optimize policies for new positions successfully.
    - Demonstrates the need for rich function approximators in complex domains.

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

- **Insights**
  - Learned policies succeeded even when **cost functions were local and too specific**.
  - Indicates potential for **further exploration** of training on different novel instances to improve generalizability.

7. Discussion

Main Challenge

1. The need for **informative features** and **effective regularization** to impose structure on the cost.
2. The difficulty of learning the cost function under **unknown dynamics** for **high-dimensional continuous systems**.

Contribution

1. This paper presents an algorithm capable of learning **arbitrary nonlinear cost functions**, such as neural networks, without **meticulous feature engineering**.
2. This paper formulates **an efficient sample-based approximation** for MaxEnt IOC.

Validation

Future Work

- Extend approach to learn cost functions directly from natural images.
- Introduce regularization methods developed for domain adaptation in computer vision (Tzeng et al., 2015).
- Encode prior knowledge that demonstrations have similar visual features to samples.

所感

- 強化学習の事前知識があればやっていることは単純…?
  - 分配関数の定式化が難しい
- 手法の評価について、さまざまなパターンで実験されているのは信頼感がおけていいのではないか