Interactive Robot Fleet Learning from Heterogeneous Human Supervisors using Implicit Energy-Based Models

Gaurav Datta

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-82
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-82.html

May 10, 2024

Copyright © 2024, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Acknowledgement

First, I want to thank Professor Ken Goldberg for the opportunity to join AUTOLab as an undergraduate student, and for his guidance and enthusiasm for research throughout my time in the lab.

I'm very grateful for the mentorship of Ryan Hoque, without whom this thesis would not have been possible. I also want to thank Allie Gu for her contributions. Additionally, I'm thankful for many other collaborators with whom I've been able to do research, especially Max Fu, Raven Huang, Will Panitch, Jaimyn Drake, Lawrence Chen, Satvik Sharma, Kishore Srinivas, Ethan Qiu, Mallika Parulekar, and Yi Liu, and for the mentorship of Daniel Brown, Ellen Novoseller, and Wisdom Agboh.

Interactive Robot Fleet Learning from Heterogeneous Human Supervisors using Implicit Energy-Based Models

by

Gaurav Datta

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Chair
Professor Pieter Abbeel

Spring 2024

DocuSign Envelope ID: AEB935F6-C477-43B2-986D-29705D33DAD7

Interactive Robot Fleet Learning from Heterogeneous Human Supervisors using Implicit Energy-Based Models

by Gaurav Datta

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

DocuSigned by:

32B42AC2D1E3498...

Professor Ken Goldberg
Research Advisor

5/9/2024

(Date)

* * * * * * *

DocuSigned by:

Pieter Abbeel

F19D53BF3207486...

Professor Pieter Abbeel
Second Reader

5/10/2024

(Date)

Interactive Robot Fleet Learning from Heterogeneous Human Supervisors using Implicit Energy-Based Models

Copyright 2024
by
Gaurav Datta

Abstract

Interactive Robot Fleet Learning from Heterogeneous Human Supervisors using Implicit Energy-Based Models

by

Gaurav Datta

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Ken Goldberg, Chair

Imitation learning has been applied to a range of robotic tasks, but can struggle when robots encounter edge cases that are not represented in the training data (i.e., distribution shift). Interactive fleet learning (IFL) mitigates distribution shift by allowing robots to access remote human supervisors during task execution and learn from them over time, but different supervisors may demonstrate the task in different ways. Recent work proposes Implicit Behavior Cloning (IBC), which is able to represent multimodal demonstrations using energy-based models (EBMs). In this work, we propose Implicit Interactive Fleet Learning (IIFL), an algorithm that builds on IBC for interactive imitation learning from multiple heterogeneous human supervisors. A key insight in IIFL is a novel approach for uncertainty quantification in EBMs using Jeffreys divergence. While IIFL is more computationally expensive than explicit methods, results suggest that IIFL achieves a 2.8× higher success rate in simulation experiments and a 4.5× higher return on human effort in a physical block pushing task over (Explicit) IFL, IBC, and other baselines.

To my family,

Contents

Contents ii

List of Figures iv

List of Tables vi

1 Introduction 1

2 Preliminaries and Related Work 3
- 2.1 Imitation Learning 3
- 2.2 Interactive Imitation Learning 3
- 2.3 Robot Learning from Multimodal Data 4
- 2.4 Jeffreys Divergence 4

3 Problem Statement 6

4 Approach 8
- 4.1 Preliminaries: Implicit Models 8
- 4.2 Implicit Interactive Dataset Aggregation 9
- 4.3 Uncertainty Estimation for EBMs 9
- 4.4 Energy-Based Allocation 11

5 Experiments 12
- 5.1 Simulation Experiments: 2D Navigation 12
- 5.2 Simulation Experiments: IFL Benchmark 12
- 5.3 Physical Experiments: Pushing Block to Target Point amid Obstacle 16

6 Limitations 18

7 Conclusion 20

Bibliography 22

8 Appendix 27

8.1 Additional Details on Implicit Models 27
8.2 Uncertainty Estimation with Larger Ensembles 27
8.3 Additional Experimental Details 28

List of Figures

4.1 Consider a pair of isotropic Gaussian energy functions $E_1(s, a)$ and $E_2(s, a)$ in green and purple respectively, where each function is a negated Gaussian probability density function and E_1 adds a uniform offset of $Z = -100$ to all values (Left). Using numerical integration to directly compute the expectations in the Jeffreys divergence identity (Identity 1), at each state we calculate the distance between the implicit policies defined by the two energy functions (Right). As intuition suggests, the divergence peaks at the mean of each Gaussian (where one energy function is highest and the other is near zero) and approaches zero where the energy functions are the same (at the center and edges of the state space). Note the symmetric structure of the Jeffreys curve, which produces identical values regardless of the offset Z. 10

5.1 In the 2D navigation experiments, the robot must navigate from the blue X marker on the left to the green X marker on the right, where the robot can go either above or below the rectangular grey obstacle and continue through a section subject to upward wind forces (blue arrows) that shift commanded motions upward. **(A) Robot Trajectories:** After training on 100 demonstrations of the two paths around the obstacle, pure behavior cloning cannot make progress past the fork due to multimodal demonstrations, while Implicit Behavior Cloning cannot overcome the distribution shift due to wind in the $+y$ direction at execution time (denoted in light blue). IIFL reaches the goal by handling both multimodality and distribution shift. **(B) Implicit Interactive Fleet Learning Energy:** We display normalized IIFL energy distributions from representative states in the trajectory. Lower energy (darker) indicates a more optimal action, and the x and y axes are the 2D action deltas $\hat{a}$ that the robot can execute (which can be mapped directly onto the corresponding $1{\times}1$ cell in the maze). At the junction point, both upward and downward actions attain low energy; in a straight hallway, the rightmost actions attain low energy; in the windy area, actions toward the lower right corner (making progress toward the goal while fighting the wind) attain low energy. 13

5.2 IFL Benchmark simulation experiment results. Despite unimodal supervision, IIFL is competitive with or outperforms IFL and other baselines across 3 environments, suggesting benefits of implicit policies beyond robustness to multimodality. Shading represents ±1 standard deviation. 14
5.3 The scripted heterogeneous supervisors for the FrankaCubeStack Isaac Gym environment pick different faces of the cube for the same cube pose. 16
5.4 Physical experiment setup with 2 ABB YuMi robots for a total of 4 independent arms. 17

8.1 29
8.2 We plot the Jeffreys divergence estimates and the ground truth action discrepancies at the first 1000 states visited by a robot with a unimodal policy. Both variants of the Jeffreys divergence calculation are positively correlated with the $L2$ distance between the robot policy's and expert policy's actions. In the $n = 2$ case, the correlation coefficient is $r = 0.688$; in the $n = 5$ case, the correlation coefficient is $r = 0.804$, indicating that additional models can make the ensemble more predictive of when the agent will deviate from the expert (at the cost of increased computation time). 30

List of Tables

5.1 Execution results from the FrankaCubeStack Isaac Gym environment with 4 heterogeneous expert policies. IIFL significantly outperforms the baselines in average reward, task successes, and return on human effort. 14
5.2 Execution results from the FrankaCubeStack Isaac Gym environment with 2 heterogeneous supervisor policies (rather than 4). 15
5.3 Physical block pushing experiment results. IIFL outperforms all baselines in number of task successes and ROHE and explicit methods in hard resets. Implicit BC and IIFL incur similar amounts of hard resets. 17

6.1 Computation times for training, inference, and uncertainty estimation for IFL and IIFL. 19

8.1 Implicit model hyperparameters. 27
8.2 Simulation environment hyperparameters. 28

Acknowledgments

First, I want to thank Professor Ken Goldberg for the opportunity to join AUTOLab as an undergraduate student, and for his guidance and enthusiasm for research throughout my time in the lab. Research has been the most important activity I've done during my time at Berkeley.

The work presented in this thesis is not entirely my own. I'm very grateful for the mentorship of Ryan Hoque, without whom this thesis would not have been possible. Ryan taught me an incredible amount about the right way to go about doing robotics research, from looking for related work to designing experiments to presenting results, and was always ready to hear out my ideas and iterate on them. Ryan conceived the idea of combining implicit models with interactive imitation learning, was instrumental in setting up and running our experiments. I also want to thank Allie Gu for her contributions. Allie wrote and debugged a lot of code and trained many models that were crucial to the success of this project.

Additionally, I'm thankful for many other collaborators with whom I've been able to do research, especially Max Fu, Raven Huang, Will Panitch, Jaimyn Drake, Lawrence Chen, Satvik Sharma, Kishore Srinivas, Ethan Qiu, Mallika Parulekar, and Yi Liu, and for the mentorship of Daniel Brown, Ellen Novoseller, and Wisdom Agboh. Max, Raven, and Lawrence have been incredible graduate mentors to me as well. Many of these collaborators and others in the lab have become good friends, making research an even greater joy.

Chapter 1

Introduction

Imitation learning (IL), the paradigm of learning from human demonstrations and feedback, has been applied to diverse tasks such as autonomous driving [44, 46, 8], robot-assisted surgery [45, 29], and deformable object manipulation [50, 4, 23]. The most common IL algorithm is behavior cloning (BC) [46], where the robot policy is derived via supervised machine learning on an offline set of human task demonstrations. However, BC can suffer from distribution shift between the states visited by the human and those visited by the robot. Distribution shift may occur due to the compounding error of the robot policy which leads the robot to a state not present in the training data, or due to the "long tail" problem of states that are individually very unlikely and therefore are not all included in the training data., leading to incorrect behavior when they occur at test time. One family of solutions to distribution shift is interactive IL (IIL) algorithms including DAgger [47] and variants [24, 27, 37], which iteratively improve the robot policy with corrective human interventions during robot task execution. These algorithms are typically designed for the single-robot, single-human setting; *interactive fleet learning* (IFL) [21] extends IIL to multiple robots and multiple human supervisors. However, learning from multiple humans can be unreliable as the data is often multimodal.

Training data is *multimodal* when the same state is paired with multiple (correct) action labels: $\{(s, a_i), (s, a_j), \dots\}, a_i \neq a_j$. Almost all robot tasks such as grasping, navigation, motion planning, and manipulation can be performed in multiple equally correct ways; as a result, almost all demonstration data has some degree of multimodality. Multimodality is especially severe when learning from different human supervisors with varying preferences and proficiency, as they demonstrate the same task in different ways [36]. Multimodality can also occur in the demonstrations of one individual human who may make mistakes, become more proficient at the task over time, or execute a different valid action when subsequently encountering the same state [36, 40].

Florence et al. [15] propose Implicit Behavior Cloning (IBC), an IL algorithm that trains an energy-based model (EBM) [31] to represent state-action mappings *implicitly* rather than explicitly. While this makes model training and inference more computationally expensive (Chapter 6), implicit models can represent multiple actions for each state. This property allows them to handle both single-human multimodality and multi-human heterogeneity, as they are indistinguishable from a data-centric perspective. However, IBC suffers from the same distribution shift as (Explicit) BC.

In this thesis we combine implicit models with interactive fleet learning to facilitate interactive learning from multiple humans. See Figure 5.1 for intuition. As existing IFL algorithms rely on estimates of epistemic uncertainty like the output variance among an ensemble of networks, which are incompatible with implicit models (Chapter 4.3), we propose a new technique for estimating the epistemic uncertainty in EBMs using Jeffreys divergence [25].

This thesis makes the following contributions:

1. Implicit Interactive Fleet Learning (IIFL), the first IIL algorithm to use implicit policies,
2. a novel metric for estimating uncertainty in energy-based models,
3. simulation experiments with a fleet of 100 robots and 10 heterogeneous algorithmic supervisors, and
4. physical experiments with a fleet of 4 robots and 2 heterogeneous human supervisors.

Open-source Python code is available at `https://github.com/BerkeleyAutomation/IIFL`.

This thesis is based on work that was presented at the 2023 Conference on Robot Learning in Atlanta, USA:

Gaurav Datta et al. "Iifl: Implicit interactive fleet learning from heterogeneous human supervisors". In: *Conference on Robot Learning*. PMLR. 2023, pp. 2340–2356

Chapter 2

Preliminaries and Related Work

2.1 Imitation Learning

Learning from an offline set of human task demonstrations is an intuitive and effective way to train a robot control policy [2, 3]. Popular approaches include behavior cloning (i.e., supervised learning) [46, 50, 44] and inverse reinforcement learning [1, 6, 3], which first infers a reward function from demonstrations and then trains a reinforcement learning agent with this reward. These demonstrations can be augmented with additional offline information such as pairwise preferences [7] and natural language [58]. Ho and Ermon [19] propose an alternative to inverse reinforcement learning using techniques from training generative adversarial networks [17], and Torabi, Warnell, and Stone [57] extend this to imitation from observation, where states are available but action labels are not. However, imitation learning from offline demonstration data can suffer from distribution shift [47], as compounding approximation errors and real-world data distributions (e.g., variable lighting in a warehouse) can lead the robot to visit states that were not visited by the human.

2.2 Interactive Imitation Learning

To mitigate distribution shift, Ross, Gordon, and Bagnell [47] propose dataset aggregation (DAgger), an IIL algorithm which collects online action labels on states visited by the robot during task execution and iteratively improves the robot policy. Since DAgger can request excessive queries to a human supervisor, several IIL algorithms seek to reduce human burden by intermittently ceding control to the human during robot execution based on some switching criteria [27, 24, 62]. Human-gated IIL [27, 54, 33] has the human decide when to take and cede control, while robot-gated IIL [22, 24, 37, 62] has the robot autonomously decide. Hoque et al. [21] propose Interactive Fleet Learning (IFL), which generalizes robot-gated IIL to multiple robots supervised by multiple humans. In this work, we consider the IFL setting.

Sun, Yang, and Mangharam [55] propose a method for interactive imitation learning from heterogeneous experts, but their method is not based on implicit policies and is limited to autonomous driving applications. Gandhi et al. [16] also interactively learn from multiple experts and propose actively soliciting the human supervisors to provide demonstrations that are compatible with the current data. However, this prevents the robot from learning alternative modes and requires the human supervisors to comply with suggestions, which may not occur due to human suboptimality, fatigue, or obstinacy [10].

2.3 Robot Learning from Multimodal Data

Learning from multimodal demonstrations is an active challenge in machine learning and robotics. A mixture density network [5] is a popular approach that fits a (typically Gaussian) mixture model to the data, but it requires setting a parameter for how many modes to fit, which may not be known a priori. When actions can be represented as pixels in an image (e.g., pick points), a Fully Convolutional Network [52] can be applied to learning pixelwise multimodality [23, 61]. Shafiullah et al. [51] propose Behavior Transformers, a technique that applies the multi-token prediction of Transformer neural networks [59] to imitation learning. Other Transformer-based policies report similar benefits for multimodal data [53, 26]; however, these approaches require action discretization to cast behavior prediction as next-token prediction. Chi et al. [9] introduce diffusion policies, an application of diffusion models [20] to imitation learning from multimodal data. Integrating diffusion policies with IFL is an exciting direction for future work.

Florence et al. [15] propose implicit behavior cloning, a technique that trains a conditional energy-based model [31] and is found to outperform (explicit) BC and mixture density networks in their experiments. As opposed to explicit models that take the form $\pi : \mathcal{S} \to \mathcal{A}$, implicit models take the form of a scalar-valued function $E : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$; the action is an input rather than an output of the model. To sample an action from the policy, instead of evaluating the explicit model $\hat{a} = \pi(s)$, the implicit model must perform optimization over E conditioned on state s:

$$\hat{a} = \arg\min_{a \in \mathcal{A}} E(s, a) \tag{2.1}$$

In this work, we combine IBC with IFL to mitigate the effects of both distribution shift and multimodality. To our knowledge, we are the first to extend implicit policies to interactive IL.

2.4 Jeffreys Divergence

The Jeffreys divergence [25] is a statistical measure of the distance between two probability distributions and is a symmetric version of the Kullback-Leibler (KL) divergence:

$$D_J(P \| Q) = D_{KL}(P \| Q) + D_{KL}(Q \| P).$$

The KL divergence is widely used in machine learning algorithms, most commonly in variational autoencoders [30] and generative adversarial networks [18]. It has also been used for dimensionality reduction [34], information bottlenecks [56], and policy gradient methods for reinforcement learning [49, 48]. The Jensen-Shannon divergence [32] is another symmetric KL divergence that sums the KL divergences of both distributions against the mixture of the two, but neither the Jensen-Shannon nor the asymmetric KL divergences have the structural properties that make Jeffreys divergence amenable to our setting (Chapter 4.3). Nielsen [39] derives a proposition similar to Identity 1 (Chapter 4.3) with Jeffreys divergence for exponential families but does not apply it to energy-based models. To our knowledge, IIFL is the first algorithm to use Jeffreys divergence for uncertainty estimation in energy-based models, exploiting its structural properties for fast computation that are not present in the asymmetric KL divergence or the Jensen-Shannon divergence.

Chapter 3

Problem Statement

We consider the interactive fleet learning (IFL) setting proposed by Hoque et al. [21], in which a fleet of N robots operate in parallel independent Markov Decision Processes (MDPs) $\{\mathcal{M}_i\}_{i=1}^N$ that are identical apart from their initial state distributions. Each MDP is a tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma, p_i^0)$, with set of states $\mathcal{S}$, set of actions $\mathcal{A}$, transition dynamics $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, \infty)$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, discount factor $\gamma \in (0, 1)$, and initial state distribution $p_i^0 : \mathcal{S} \to [0, \infty)$. The MDPs are equipped with an indicator function $c : \mathcal{S} \to \{0, 1\}$ which takes value 1 when the robot is violating a constraint, i.e. it is in a fault state and cannot make further progress. The N identical MDPs may be vectorized into a single MDP $\left(\mathcal{S}^N, \mathcal{A}^N, \tilde{p}, \tilde{r}, \gamma, \tilde{p}^0\right)$, where for $\mathbf{s} = (s_1, \ldots, s_N) \in \mathcal{S}^{\mathcal{N}}$ and $\mathbf{a} = (a_1, \ldots, a_N) \in \mathcal{A}^N$, $\tilde{p}(\mathbf{s}^{t+1}|\mathbf{s}^t, \mathbf{a}^t) = \prod_{i=1}^N p(s_i^{t+1}|s_i^t, a_i^t)$, $\tilde{r}(\mathbf{s}, \mathbf{a}) = \sum_{i=1}^N r(s_i, a_i)$, and $\tilde{p}^0(\mathbf{s}) = \prod_{i=1}^N p_i^0(s_i)$.

The robots can query a set of $M < N$ human supervisors with action space $\mathcal{A}_H = \mathcal{A} \cup \{R\}$, where $a \in \mathcal{A}$ is teleoperation in the action space of the robots and R is a "hard reset" that physically resets a robot in a failure state (e.g., a delivery robot tipped over on its side). Like Hoque et al. [21], we assume that (1) the robots share policy $\pi_{\theta_t} : \mathcal{S} \to \mathcal{A}$, (2) the MDP timesteps are synchronous across robots, and (3) each human can only help one robot at a time. However, unlike the original IFL formulation [21], we do *not* assume that the human supervisors are homogeneous; instead, each human i may have a unique policy $\pi_H^i : \mathcal{S} \to \mathcal{A}_H$. Furthermore, each π_H^i may itself be nondeterministic and multimodal, but is assumed to be optimal or nearly optimal.

An IFL supervisor allocation algorithm is a policy ω that determines the assignment $\boldsymbol{\alpha}^t$ of humans to robots at time t, with no more than one human per robot and one robot per human at a time:

$$\omega : (\mathbf{s}^t, \pi_{\theta_t}, \cdot) \mapsto \boldsymbol{\alpha}^t \in \{0, 1\}^{N \times M} \quad \text{s.t.} \quad \sum_{j=1}^M \boldsymbol{\alpha}_{ij}^t \leq 1 \text{ and } \sum_{i=1}^N \boldsymbol{\alpha}_{ij}^t \leq 1 \quad \forall i, j. \tag{3.1}$$

Here, $\mathbf{s}^t$ are the current states of each of the robots, $\boldsymbol{\alpha}^t$ is an $N \times M$ binary matrix that indicates which robot will receive assistance from which human at the current timestep t, and π_{θ_t} is the shared robot control policy at time t.

The allocation policy ω in IFL must be autonomously determined with robot-gated criteria [24, 37] rather than human-gated criteria [27, 54, 33] in order to scale to large ratios of N to M. The IFL objective is to find an ω that maximizes return on human effort (ROHE), defined as the average performance of the robot fleet normalized by the amount of human effort required [21]:

$$\max_{\omega \in \Omega} \mathbb{E}_{\tau \sim p_{\omega,\theta_0}(\tau)} \left[\frac{M}{N} \cdot \frac{\sum_{t=0}^{T} \bar{r}(\mathbf{s}^t, \mathbf{a}^t)}{1 + \sum_{t=0}^{T} \|\omega(\mathbf{s}^t, \pi_{\theta_t}, \boldsymbol{\alpha}^{t-1}, \mathbf{x}^t)\|_F^2} \right] \tag{3.2}$$

where $\|\cdot\|_F$ is the Frobenius norm, T is the amount of time the fleet operates (rather than an individual episode horizon), and θ_0 are the initial parameters of π_{θ_t}.

Chapter 4

Approach

4.1 Preliminaries: Implicit Models

We build on Implicit Behavior Cloning [15]. IBC seeks to learn a conditional energy-based model $E : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, where $E(s, a)$ is the scalar "energy" for action a conditioned on state s. Lower energy indicates a higher correspondence between s and a. The energy function defines a multimodal probability distribution π of action a conditioned on state s:

$$\pi(a|s) = \frac{e^{-E(s,a)}}{Z(s)} \tag{4.1}$$

where $Z(s)$ is a normalization factor known as the "partition function." In practice, we estimate E with a learned neural network function approximator E_θ parameterized by θ.

Implicit BC trains an energy-based model E_θ on samples $\{s_i, a_i\}$ collected from the expert policies π_H. After generating a set of counter-examples $\{\tilde{a}_i^j\}$ for each s_i, Implicit BC minimizes the following InfoNCE [42] loss function:

$$\mathcal{L} = \sum_{i=1}^{N} -\log \hat{p}_\theta(a_i|s_i, \{\tilde{a}_i^j\}), \quad \hat{p}_\theta(a_i|s_i, \{\tilde{a}_i^j\}) := \frac{e^{-E_\theta(s_i,a_i)}}{e^{-E_\theta(s_i,a_i)} + \sum_j e^{-E_\theta(s_i,\tilde{a}_i^j)}}. \tag{4.2}$$

This loss is equivalent to the negative log likelihood of the training data, where the partition function $Z(s)$ is estimated with the counter-examples. Florence et al. [15] propose three techniques for generating these counter-examples $\{\tilde{a}_i^j\}$ and performing inference over the learned model E_θ; we choose gradient-based Langevin sampling [60] with an additional gradient penalty loss for training in this work as Florence et al. [15] demonstrate that it scales with action dimensionality better than the alternate methods. This is a Markov Chain Monte Carlo (MCMC) method with stochastic gradient Langevin dynamics. More details are available in Appendix B.3 of Florence et al. [15].

4.2 Implicit Interactive Dataset Aggregation

Behavior cloning is prone to distribution shift due to compounding approximation errors [47], and any data-driven robot policy may encounter edge cases during execution that are not represented in the training data [21]. We extend IBC to interactive imitation learning using dataset aggregation of online human data, and iteratively update the shared robot policy with the aggregate dataset at a fixed interval $1 \leq \hat{t} \leq T$ via supervised learning, as in DAgger [47] and variants [27, 21]:

$$\begin{cases} D^{t+1} \leftarrow D^t \cup D^t_H, \text{ where } D^t_H := \{(s^t_i, \pi^j_H(s^t_i)) : \pi^j_H(s^t_i) \neq R \text{ and } \sum_{j=1}^{M} \boldsymbol{\alpha}^t_{ij} = 1\} \\ \pi_{\theta_t} \leftarrow \arg\min_\theta \mathcal{L}(\pi_\theta, D^t), \texttt{ if } t \equiv 0 \pmod{\hat{t}} \end{cases}$$

where $\pi^j_H(s^t_i)$ is the teleoperation action from human j for robot i at time t, and $\boldsymbol{\alpha}^t_{ij}$ is the assignment of human j to robot i at time t, as in Equation 3.1. Ross, Gordon, and Bagnell [47] show that such a policy incurs approximation error that is linear in the time horizon rather than quadratic, as in behavior cloning.

4.3 Uncertainty Estimation for EBMs

While prior work computes the output variance among a bootstrapped ensemble of neural networks to estimate epistemic uncertainty [11, 37, 24], this approach is not applicable to implicit policies because multimodality results in a false positive: different ensemble members may select equally optimal actions from different modes, resulting in high variance despite high certainty. Furthermore, training and inference in EBMs are much more computationally expensive than in explicit models (Chapter 6), making ensembles of 5+ models impractical. Finally, inference in implicit models is nondeterministic, creating an additional source of variance that is not due to uncertainty.

The notion of ensemble disagreement can still be applied to EBMs by considering the action *distributions* at a given state rather than the single predicted actions. At states within the distribution of the human data, a bootstrapped EBM will predict action distributions that are concentrated around the human actions. However, outside of the human data distribution, the models have no reference behavior to imitate, and will likely predict different conditional action distributions due to random initialization, stochastic optimization, and bootstrapping. Accordingly, we propose bootstrapping 2 implicit policies and calculating the Jeffreys divergence D_J [25] between them as a measure of how their conditional action distributions differ at a given state. Jeffreys divergence in this setting has two key properties: (1) it is symmetric, which is useful as neither bootstrapped policy is more correct than the other, and (2) it is computationally tractable for EBMs as it does not require estimating the partition function $Z(s)$ (Equation 4.1). To show (2), we derive the following novel identity:

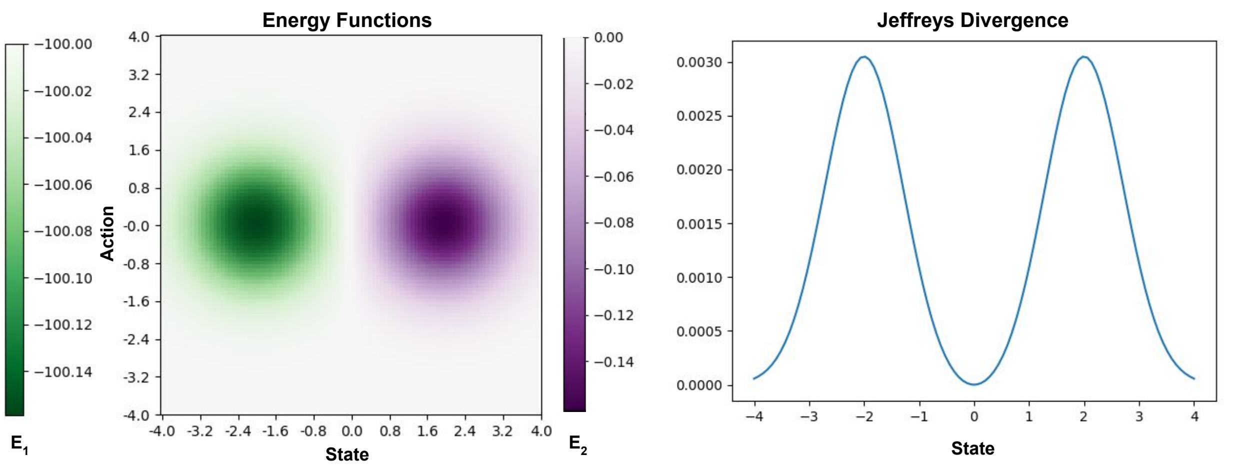


Figure 4.1: Consider a pair of isotropic Gaussian energy functions $E_1(s, a)$ and $E_2(s, a)$ in green and purple respectively, where each function is a negated Gaussian probability density function and E_1 adds a uniform offset of $Z = -100$ to all values (Left). Using numerical integration to directly compute the expectations in the Jeffreys divergence identity (Identity 1), at each state we calculate the distance between the implicit policies defined by the two energy functions (Right). As intuition suggests, the divergence peaks at the mean of each Gaussian (where one energy function is highest and the other is near zero) and approaches zero where the energy functions are the same (at the center and edges of the state space). Note the symmetric structure of the Jeffreys curve, which produces identical values regardless of the offset Z.

Identity 1. *Let E_1 and E_2 be two energy-based models that respectively define distributions π_1 and π_2 according to Equation 4.1. Then,*

$$D_J\left(\pi_1(\cdot|s) \| \pi_2(\cdot|s)\right) = \mathbb{E}_{a \sim \pi_1(\cdot|s)}\left[E_2(s,a) - E_1(s,a)\right] + \mathbb{E}_{a \sim \pi_2(\cdot|s)}\left[E_1(s,a) - E_2(s,a)\right].$$

Proof. The proof follows from applying the definition of Jeffreys divergence to EBMs:

$$\begin{aligned}
D_J\left(\pi_1(\cdot|s) \| \pi_2(\cdot|s)\right) &= D_{KL}\left(\pi_1(\cdot|s) \| \pi_2(\cdot|s)\right) + D_{KL}\left(\pi_2(\cdot|s) \| \pi_1(\cdot|s)\right) \\
&= \mathbb{E}_{a \sim \pi_1(\cdot|s)}\left[\log \frac{\pi_1(a|s)}{\pi_2(a|s)}\right] + \mathbb{E}_{a \sim \pi_2(\cdot|s)}\left[\log \frac{\pi_2(a|s)}{\pi_1(a|s)}\right] \\
&= \mathbb{E}_{a \sim \pi_1(\cdot|s)}\left[E_2(s,a) - E_1(s,a)\right] - \log Z_1(s) + \log Z_2(s) \\
&\quad + \mathbb{E}_{a \sim \pi_2(\cdot|s)}\left[E_1(s,a) - E_2(s,a)\right] - \log Z_2(s) + \log Z_1(s) \\
&= \mathbb{E}_{a \sim \pi_1(\cdot|s)}\left[E_2(s,a) - E_1(s,a)\right] + \mathbb{E}_{a \sim \pi_2(\cdot|s)}\left[E_1(s,a) - E_2(s,a)\right].
\end{aligned}$$

□

Crucially, the intractable partition functions do not appear in the expression due to the symmetry of Jeffreys divergence. We estimate the expectations in Identity 1 using Langevin sampling. Note that this method is not limited to the interactive IL setting and may have broad applications for any algorithms or systems that use energy-based models. To provide more intuition on this identity, we plot the Jeffreys divergence for a pair of isotropic Gaussian energy functions in Figure 4.1. In Appendix 8.2, we consider how this method may be generalized to a greater number of models in Appendix 8.2.

4.4 Energy-Based Allocation

To extend IBC to the IFL setting, we synthesize the Jeffreys uncertainty estimate with Fleet-DAgger [21]. Fleet-DAgger allocates robots according to a priority function $\hat{p} : (s, \pi_{\theta_t}) \rightarrow [0, \infty)$, where a higher value indicates a higher priority robot. Specifically, we set the Fleet-DAgger priority function to prioritize robots with high uncertainty as quantified by our Jeffreys divergence estimate, followed by robots that require a hard reset R (i.e. they are in a state s with $c(s) = 1$). This produces a supervisor allocation policy ω with Fleet-EnsembleDAgger, the U.C. (Uncertainty-Constraint) allocation scheme proposed by Hoque et al. [21]. We refer to the composite approach as IIFL.

Chapter 5

Experiments

5.1 Simulation Experiments: 2D Navigation

To evaluate the correctness of our implementation and provide visual intuition, we first run experiments in a 2D pointbot navigation environment. See Figure 5.1 for the maze environment, representative trajectories, and energy distribution plots. We consider discrete 2D states $s = (x, y) \in \mathbb{N}^2$ (the Cartesian pose of the robot) and continuous 2D actions $a = (\Delta x, \Delta y) \in [-1, 1]^2$ (relative changes in Cartesian pose). The maze has a fixed start and goal location and consists of a forked path around a large obstacle followed by a long corridor. An algorithmic supervisor provides 100 demonstrations of the task, randomly choosing to go upward or downward at the fork with equal probability. Since a model can simply overfit to the demonstrations in this low-dimensional environment, to induce distribution shift we add "wind" at execution time to a segment of the right corridor with magnitude 0.75 in the $+y$ direction.

In 100 trials, (explicit) BC achieves a 0% success rate, IBC achieves a 0% success rate, and IIFL achieves a 100% autonomous success rate (i.e., robot-only trajectories without human interventions, after interactive training). In Figure 5.1 we observe that BC cannot pass the fork due to averaging the two modes to zero. Meanwhile, IBC is not robust to the distribution shift: once the wind pushes the robot to the top of the corridor, it does not know how to return to the center. We also observe that the IIFL energy distributions in Figure 5.1(B) reflect the desired behavior in accordance with intuition.

5.2 Simulation Experiments: IFL Benchmark

Environments: Evaluating IIFL in simulation is uniquely challenging as it requires all of the following, precluding the use of most existing benchmarks in similar papers: (1) efficient simulation of large robot fleets, (2) simulation of multiple algorithmic humans, (3) interactive human control, and (4) heterogeneous human control, which is difficult to specify in joint space. To accommodate these requirements, following prior work [21] we evaluate with Isaac

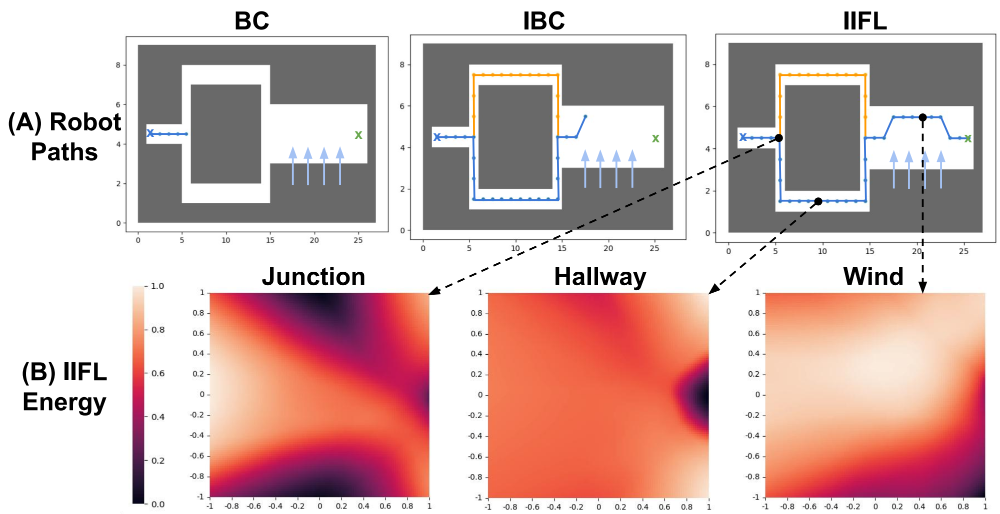


Figure 5.1: In the 2D navigation experiments, the robot must navigate from the blue X marker on the left to the green X marker on the right, where the robot can go either above or below the rectangular grey obstacle and continue through a section subject to upward wind forces (blue arrows) that shift commanded motions upward. **(A) Robot Trajectories:** After training on 100 demonstrations of the two paths around the obstacle, pure behavior cloning cannot make progress past the fork due to multimodal demonstrations, while Implicit Behavior Cloning cannot overcome the distribution shift due to wind in the $+y$ direction at execution time (denoted in light blue). IIFL reaches the goal by handling both multimodality and distribution shift. **(B) Implicit Interactive Fleet Learning Energy:** We display normalized IIFL energy distributions from representative states in the trajectory. Lower energy (darker) indicates a more optimal action, and the x and y axes are the 2D action deltas $\hat{a}$ that the robot can execute (which can be mapped directly onto the corresponding $1{\times}1$ cell in the maze). At the junction point, both upward and downward actions attain low energy; in a straight hallway, the rightmost actions attain low energy; in the windy area, actions toward the lower right corner (making progress toward the goal while fighting the wind) attain low energy.

Gym [35] and the IFL Benchmark [21]. We separate these experiments into two domains: (1) homogeneous human control in 3 environments (Ball Balance, Ant, Anymal) to compare with prior IFL algorithms that assume unimodal supervision; (2) heterogeneous human control in FrankaCubeStack, the only Isaac Gym environment with Cartesian space control. More details are available in Appendix 8.3.

Metrics: Following prior work [21], we measure the total successful task completions across the fleet and the total number of hard resets. For interactive algorithms, we also measure the return on human effort (Equation 3.2) where reward is a sparse $r \in \{0, 1\}$ for task completion. Task execution is deemed successful if the robot completes its trajectory without a hard reset and reaches 95% of expert human reward.

Baselines: We compare IIFL to the following baselines: (explicit) BC, IBC, (explicit) IFL (specifically, Fleet-EnsembleDAgger [21]), and IIFL-Random (IIFL-R), which is an ab-

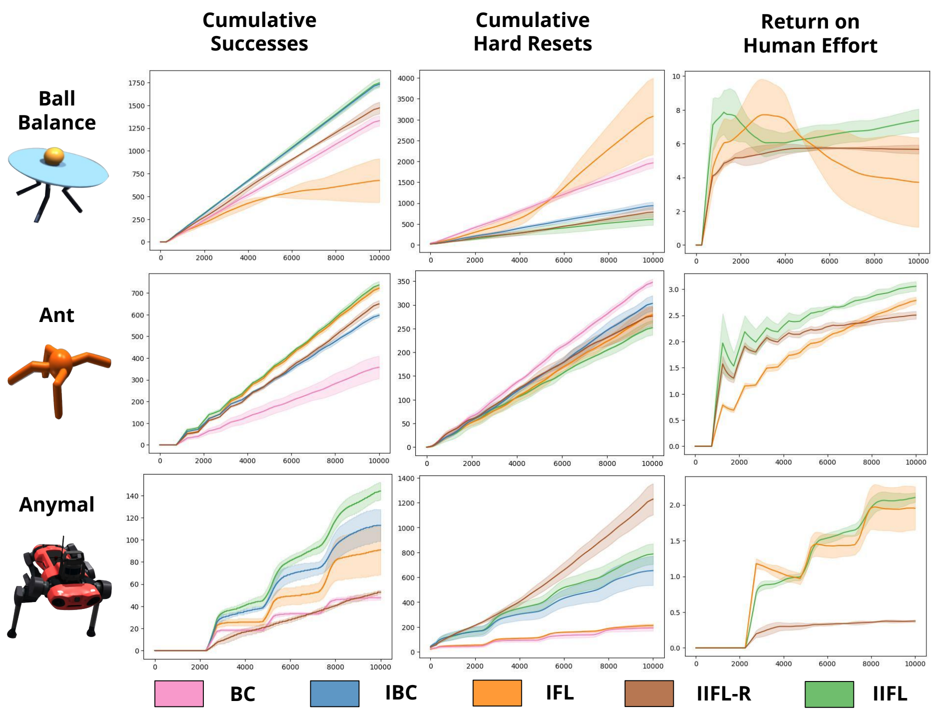


Figure 5.2: IFL Benchmark simulation experiment results. Despite unimodal supervision, IIFL is competitive with or outperforms IFL and other baselines across 3 environments, suggesting benefits of implicit policies beyond robustness to multimodality. Shading represents ± 1 standard deviation.

Algorithm	Avg. Reward	Task Successes	ROHE
BC	29.27 ± 14.05	0.3 ± 0.5	N/A
IBC	24.96 ± 0.83	0.0 ± 0.0	N/A
IFL	230.39 ± 53.41	7.0 ± 2.2	2.30 ± 0.53
IIFL-R	166.24 ± 28.63	0.0 ± 0.0	1.66 ± 0.29
IIFL	$\mathbf{784.26 \pm 122.41}$	$\mathbf{26.7 \pm 4.5}$	$\mathbf{7.84 \pm 1.22}$

Table 5.1: Execution results from the FrankaCubeStack Isaac Gym environment with 4 heterogeneous expert policies. IIFL significantly outperforms the baselines in average reward, task successes, and return on human effort.

lation of IIFL that allocates humans to robots randomly instead of using the Jeffreys uncertainty estimate. Human supervisors for BC and IBC perform only hard resets (i.e., no teleoperation) during execution.

Experimental Setup: We run experiments with a fleet of $N = 100$ robots and $M = 10$ algorithmic supervisors, where the supervisors are reinforcement learning agents trained with Isaac Gym's reference implementation of PPO [48]. All training runs have hard reset time $t_R = 5$ timesteps, minimum intervention time $t_T = 5$ timesteps, and fleet operation time $T = 10,000$ timesteps [21], and are averaged over 3 random seeds. The initial robot policy π_{θ_0}

Algorithm	Avg. Reward	Task Successes	ROHE
BC	23.45 ± 0.99	0.0 ± 0.0	N/A
IBC	30.32 ± 2.78	0.0 ± 0.0	N/A
IFL	307.87 ± 118.59	9.3 ± 4.7	3.08 ± 1.19
IIFL-R	244.98 ± 32.58	0.0 ± 0.0	2.45 ± 0.33
IIFL	$\mathbf{604.17 \pm 263.06}$	$\mathbf{17.7 \pm 11.1}$	$\mathbf{6.04 \pm 2.63}$

Table 5.2: Execution results from the FrankaCubeStack Isaac Gym environment with 2 heterogeneous supervisor policies (rather than 4).

for all algorithms is initialized with behavior cloning on 10 full task demonstrations. While IFL trains at every timestep following prior work [21], the implicit interactive algorithms train at intervals of 1000 timesteps with an equivalent total amount of gradient steps for increased stability of EBM training.

FrankaCubeStack, in which a Franka arm grasps a cube and stacks it on another, has several differences from the other 3 environments. First, since it allows Cartesian space control, we can script 4 *heterogeneous* supervisor policies with grasps corresponding to each face of the cube; the $M = 10$ supervisors are split into 4 groups, each of which has a unique policy. Second, due to the difficulty of scripting interactive experts, the online interventions take place at execution-time (i.e., the robot policy is frozen). Third, since there is no notion of catastrophic failure in the cube stacking environment, we do not report hard resets as there are none.

Using known pose information and Cartesian space control, the supervisor policy does the following, where Cube A is to be stacked on Cube B: (1) move the end effector to a position above Cube A; (2) rotate into a pre-grasp pose; (3) descend to Cube A; (4) lift Cube A; (5) translate to a position above Cube B; (6) place Cube A on Cube B; and (7) release the gripper. Heterogeneity is concentrated in Step 2: while one supervisor rotates to an angle $\theta \in [0, \frac{\pi}{2}]$ that corresponds to a pair of antipodal faces of the cube, the others rotate to $\theta - \pi, \theta - \frac{\pi}{2}$, and $\theta + \frac{\pi}{2}$. See Figure 5.3 for intuition. We also consider an experiment with only 2 hetereogeneous policies (θ and $\theta - \frac{\pi}{2}$).

Results: The results are shown in Figure 5.2 and Table 5.1. In the homogeneous control experiments, we observe that IIFL rivals or outperforms all baselines across all metrics, with the exception of hard resets in the Anymal environment. We hypothesize that the latter results from learning more "aggressive" locomotion that makes greater progress on average but is more prone to failure. These results suggest that implicit policies may have desirable properties over explicit policies such as improved data efficiency and generalization even when multimodality is *not* present in the data, as suggested by prior work [15]. The severity of distribution shift due to compounding approximation error [47] in the homogeneous experiments roughly corresponds to the performance gap between BC and IFL (or IBC and IIFL). Surprisingly, (explicit) IFL underperforms BC in Ball Balance; we hypothesize that this is due to its frequent policy updates on a shifting low-dimensional data distribution. In the FrankaCubeStack environment (with 4 hetereogeneous policies), IIFL significantly

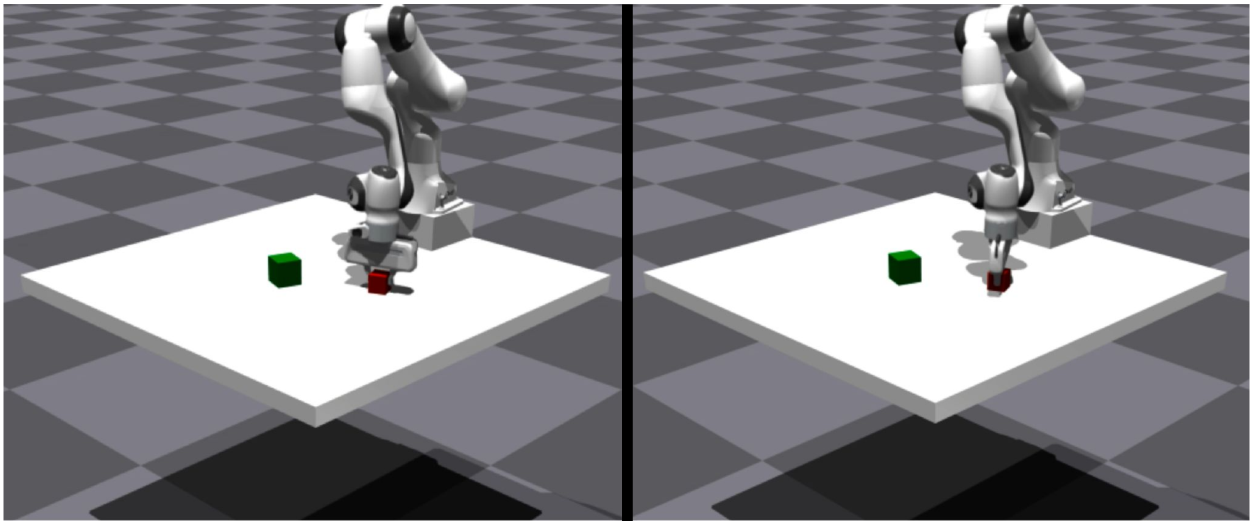


Figure 5.3: The scripted heterogeneous supervisors for the FrankaCubeStack Isaac Gym environment pick different faces of the cube for the same cube pose.

outperforms the baselines across all metrics, indicating the value of implicit policies for heterogeneous supervision. The 74% performance gap between IFL and IIFL corresponds to the severity of multimodality in this environment. Only IFL and IIFL attain nontrivial success rates; while IIFL-R makes progress, it is not able to successfully stack the cube, suggesting that IIFL allocates human attention more judiciously. Table 5.2 shows the results for the FrankaCubeStack environment with 2 heterogeneous policies, which (in conjunction with Table 5.1) suggest that relative performance of IIFL over baselines remains approximately consistent as the number of modes varies and can improve as multimodality increases.

5.3 Physical Experiments: Pushing Block to Target Point amid Obstacle

Experimental Setup: To evaluate IIFL in experiments with real-world human multimodality and high-dimensional state spaces, we run an image-based block-pushing task with a fleet of $N = 4$ ABB YuMi robot arms operating simultaneously and $M = 2$ human supervisors, similar to Hoque et al. [21]. See Figure 5.4 for the physical setup. Each robot has an identical square workspace with a small blue cube and rectangular pusher as an end effector. Unlike Hoque et al. [21], we add a square obstacle to the center of each workspace. The task for each robot is to push the cube to a goal region diametrically opposite the cube's initial position without colliding with the walls or the obstacle. Once this is achieved, the goal region is procedurally reset based on the new cube position. As described in Chapter 3, the role of human superivsion is to (1) teleoperate when requested and (2) provide a physical hard

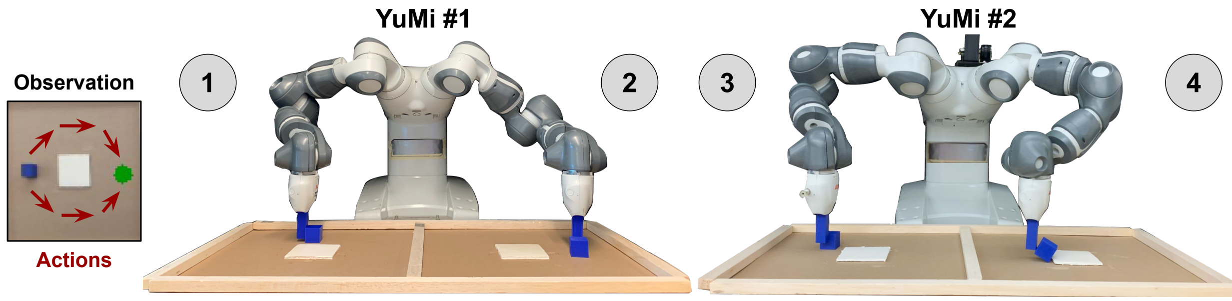


Figure 5.4: Physical experiment setup with 2 ABB YuMi robots for a total of 4 independent arms.

Algorithm	Successes (↑)	Hard Resets (↓)	ROHE (↑)
BC	2.0 ± 0.8	51.0 ± 0.8	N/A
IBC	20.3 ± 4.1	$\mathbf{35.3 \pm 6.8}$	N/A
IFL	7.0 ± 0.8	47.3 ± 0.5	0.13 ± 0.01
IIFL	$\mathbf{36.3 \pm 1.2}$	37.0 ± 2.2	$\mathbf{0.71 \pm 0.01}$

Table 5.3: Physical block pushing experiment results. IIFL outperforms all baselines in number of task successes and ROHE and explicit methods in hard resets. Implicit BC and IIFL incur similar amounts of hard resets.

reset when requested. When both paths to the goal are equidistant, Human 1 pushes the cube *clockwise* around the obstacle while Human 2 pushes the cube *counterclockwise*; if one path is closer, the human takes that path. Hard resets R are defined to be collisions of the cube with the obstacle or the boundaries of the workspace. Furthermore, unlike the discrete action space in Hoque et al. [21], we use a continuous 2D action space of $a = (\Delta x, \Delta y)$ that corresponds to the vector along which to push the block, starting from the block's center. We run 3 trials of each algorithm in Table 5.3 for $T = 150$ timesteps; see Appendix 8.3 for more details.

Results: The results are shown in Table 5.3. We observe that implicit policies are crucial for success, as the explicit methods rarely reach the goal and incur many hard resets. Results suggest that IIFL improves the success rate by 80% over IBC and improves ROHE by 4.5× over IFL. However, IIFL incurs a similar number of hard resets to IBC. We hypothesize that the duration of the physical experiment, difficult to extend due to the significant robot and human time required, is insufficient to learn subtle collision avoidance behaviors that noticeably reduce the number of hard resets.

Chapter 6

Limitations

Since IIFL extends IBC, it inherits some of its limitations. First, Florence et al. [15] find that IBC performance falls on some tasks when the action space dimensionality is very high ($|\mathcal{A}| > 16$); we do not observe this in our experiments as $|\mathcal{A}| \leq 12$ but IIFL likely incurs this property with higher-dimensional actions.

Second, model training and inference require 18× and 82× more computation time than explicit methods. In Table 6.1 we report the mean and standard deviation of various computation time metrics. All timing experiments were performed with $N = 100$ robots and averaged across $T = 100$ timesteps in the Ant environment on a single NVIDIA Tesla V100 GPU with 32 GB RAM. Training time is reported for a single gradient step with a batch size of 512. Note that with default hyperparameters, IFL trains an ensemble of 5 (explicit) models and IIFL trains an ensemble of 2 (implicit) models; hence, we also report the training time per individual model. IFL inference consists of a single forward pass through each of the 5 models, while IIFL inference performs 100 iterations of stochastic gradient Langevin dynamics; both of these are vectorized across all 100 robots at once. While IIFL can provide policy performance benefits over IFL, we observe that it comes with a tradeoff of computation time, which may be mitigated with parallelization across additional GPUs. We measure implicit training to take an additional 0.34 seconds per gradient step and implicit inference to take an additional 0.49 seconds in the Ant environment. For the two implicit models that IIFL requires for energy-based switching, training takes an additional 0.66 seconds of computation time per training step.

Third, while it is 7× faster than alternate methods for implicit models and has sub-second latency for a fleet of 100 robots, IIFL uncertainty estimation is nevertheless 340× slower than its highly efficient explicit counterpart. IFL uncertainty estimation consists of a single forward pass through each of the 5 models while IIFL performs both Langevin iterations and 2 forward passes through each of the 2 models. However, while uncertainty estimation is the bottleneck in IIFL, it is performed with sub-second latency for the entire fleet. This is significantly faster than alternatives such as directly estimating the partition function, which is both less accurate and slower; we measure it to take an average of 7.10 seconds per step using annealed importance sampling [38].

Time	IFL	IIFL
Training step (s)	0.0385 ± 0.0205	0.694 ± 0.207
Training step per model (s)	0.0077 ± 0.0041	0.347 ± 0.104
Inference (s)	0.0060 ± 0.0395	0.494 ± 0.045
Uncertainty estimation (s)	0.0029 ± 0.0008	0.988 ± 0.008

Table 6.1: Computation times for training, inference, and uncertainty estimation for IFL and IIFL.

Finally, the real-world evaluation of IIFL is limited to block pushing with fixed block properties; more comprehensive evaluation of IIFL in a wider range of physical domains is required to assess its full applicability.

Chapter 7

Conclusion

This thesis presents IIFL, an interactive imitation learning algorithm that uses implicit energy-based models to learn multimodal policies from heterogeneous human supervisors. We propose a novel method for uncertainty quantification in energy-based models that uses the Jeffreys Divergence between a pair of bootstrapped models, and show how this can be estimated efficiently. We note that this technique does not rely on any IFL assumptions, and may be broadly useful beyond this setting to any applications involving Boltzmann distributions and energy-based models. Our simulated and physical experiments suggest that this method can efficiently allocate human supervisors to robot fleets, at the cost of increased computation time. However, the physical evaluation of IIFL is limited, and a more comprehensive study in additional physical environments to assess the effectiveness of the methods proposed here and of other IFL algorithms would be a valuable contribution. Additionally, there are many other approaches for handling multimodality, such as such as Behavior Transformers [51] and Diffusion Policies [9]. Extending these works to the IFL setting is also an exciting direction for future work, especially as Diffusion Policy removes many of the limitations of energy-based models regarding training stability, but requires some new form of uncertainty quantification. Another possible extension of this work would be to lift the assumption that all human supervisors are nearly optimal at the task and allowing the possibility of suboptimal demonstrations.

On a broader note, the past few years have seen attempts at scaling up data to train large models for robotics in a similar fashion to what has been successful for many tasks in computer vision and natural language processing [43, 14, 12, 28, 41]. Though these results are impressive, I conjecture that distribution shift will remain a problem for these approaches that train on an offline dataset and are frozen by the time they interact with the environment. It may be that the collection of ever larger datasets will be enough to mitigate this, but perhaps the long tail problem will preclude this. Therefore, algorithms that make use of interventions and continual learning, such as DAgger [47], may be necessary. As robots are more widely deployed, they must judiciously choose when to ask for help so as to not overwhelm their human supervisors, making it crucial that they know when it is that they do not know what to do. I look forward to seeing how the research community grapples with these challenges.

Bibliography

[1] Pieter Abbeel and Andrew Y Ng. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 1.

[2] Brenna D Argall et al. "A survey of robot learning from demonstration". In: *Robotics and autonomous systems* 57.5 (2009), pp. 469–483.

[3] Saurabh Arora and Prashant Doshi. "A survey of inverse reinforcement learning: Challenges, methods and progress". In: *arXiv preprint arXiv:1806.06877* (2018).

[4] Yahav Avigal et al. "SpeedFolding: Learning Efficient Bimanual Folding of Garments". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2022), pp. 1–8.

[5] Christopher M. Bishop. "Mixture Density Networks". In: *Neural Computing Research Group Report* (1994).

[6] Daniel S Brown, Wonjoon Goo, and Scott Niekum. "Better-than-Demonstrator Imitation Learning via Automaticaly-Ranked Demonstrations". In: *Conference on Robot Learning (CoRL)*. 2019.

[7] Daniel S Brown et al. "Safe Imitation Learning via Fast Bayesian Reward Inference from Preferences". In: *International Conference on Machine Learning (ICML)*. 2020.

[8] Jianyu Chen, Bodi Yuan, and Masayoshi Tomizuka. "Deep Imitation Learning for Autonomous Driving in Generic Urban Scenarios with Enhanced Safety". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2019), pp. 2884–2890.

[9] Cheng Chi et al. "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion". In: *arXiv preprint arXiv:2303.04137* (2023).

[10] Susan E. F. Chipman. *The Oxford Handbook of Cognitive Science*. Oxford University Press, Oct. 2017. ISBN: 9780199842193.

[11] Kurtland Chua et al. "Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models". In: *Neural Information Processing Systems*. 2018.

[12] Open X-Embodiment Collaboration et al. *Open X-Embodiment: Robotic Learning Datasets and RT-X Models*. `https://arxiv.org/abs/2310.08864`. 2023.

[13] Gaurav Datta et al. "Iifl: Implicit interactive fleet learning from heterogeneous human supervisors". In: *Conference on Robot Learning*. PMLR. 2023, pp. 2340–2356.

[14] Danny Driess et al. "Palm-e: An embodied multimodal language model". In: *arXiv preprint arXiv:2303.03378* (2023).

[15] Peter R. Florence et al. "Implicit Behavioral Cloning". In: *Conference on Robot Learning (CoRL)*. 2021.

[16] Kanishk Gandhi et al. "Eliciting Compatible Demonstrations for Multi-Human Imitation Learning". In: *Conference on Robot Learning (CoRL)*. 2022.

[17] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014.

[18] Ian Goodfellow et al. "Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems*. 2014.

[19] Jonathan Ho and Stefano Ermon. "Generative Adversarial Imitation Learning". In: *Neural Information Processing Systems (NeurIPS)*. 2016.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *arXiv preprint arXiv:2006.11239* (2020).

[21] Ryan Hoque et al. "Fleet-DAgger: Interactive Robot Fleet Learning with Scalable Human Supervision". In: *Conference on Robot Learning (CoRL)*. 2022.

[22] Ryan Hoque et al. "LazyDAgger: Reducing context switching in interactive imitation learning". In: *IEEE Conference on Automation Science and Engineering (CASE)*. 2021, pp. 502–509.

[23] Ryan Hoque et al. "Learning to Fold Real Garments with One Arm: A Case Study in Cloud-Based Robotics Research". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022.

[24] Ryan Hoque et al. "ThriftyDAgger: Budget-aware novelty and risk gating for interactive imitation learning". In: *Conference on Robot Learning (CoRL)*. 2021.

[25] Harold Jeffreys. *The Theory of Probability*. Oxford University Press, 1939.

[26] Yunfan Jiang et al. "VIMA: General Robot Manipulation with Multimodal Prompts". In: *NeurIPS 2022 Foundation Models for Decision Making Workshop*. 2022.

[27] Michael Kelly et al. "HG-DAgger: Interactive Imitation Learning with Human Experts". In: *2019 International Conference on Robotics and Automation (ICRA)* (2018), pp. 8077–8083.

[28] Alexander Khazatsky et al. "DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset". In: (2024).

[29] Ji Woong Kim et al. "Towards Autonomous Eye Surgery by Combining Deep Imitation Learning with Optimal Control". In: *Conference on Robot Learning (CoRL)*. 2020.

[30] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *International Conference on Learning Representations (ICLR)*. 2014.

[31] Yann LeCun et al. "A Tutorial on Energy-Based Learning". In: *Predicting Structured Data* 1.0 (2006).

[32] J. Lin. "Divergence measures based on the Shannon entropy". In: *IEEE Transactions on Information Theory* 37.1 (1991), pp. 145–151.

[33] Huihan Liu et al. "Robot Learning on the Job: Human-in-the-Loop Autonomy and Learning During Deployment". In: *arXiv* abs/2211.08416 (2022).

[34] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: `http://jmlr.org/papers/v9/vandermaaten08a.html`.

[35] Viktor Makoviychuk et al. "Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning". In: *arXiv preprint arXiv:2108.10470* (2021).

[36] Ajay Mandlekar et al. "What Matters in Learning from Offline Human Demonstrations for Robot Manipulation". In: *Conference on Robot Learning (CoRL)*. 2021.

[37] Kunal Menda, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. "EnsembleDAgger: A Bayesian Approach to Safe Imitation Learning". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019.

[38] Radford M Neal. "Annealed importance sampling". In: *Statistics and computing* 11 (2001), pp. 125–139.

[39] Frank Nielsen. "Fast Approximations of the Jeffreys Divergence between Univariate Gaussian Mixtures via Mixture Conversions to Exponential-Polynomial Distributions". In: *Entropy* 23 (2021).

[40] Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. "Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks". In: *Neural Information Processing Systems (NeurIPS)*. 2021.

[41] Octo Model Team et al. *Octo: An Open-Source Generalist Robot Policy*. `https://octo-models.github.io`. 2023.

[42] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding". In: *ArXiv preprint arXiv:1807.03748* (2018).

[43] OpenAI. "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774* (2023).

[44] Yunpeng Pan et al. "Agile Autonomous Driving using End-to-End Deep Imitation Learning". In: *Robotics: Science and Systems (RSS)*. 2018.

[45] Samuel Paradis et al. "Intermittent Visual Servoing: Efficiently Learning Policies Robust to Instrument Changes for High-precision Surgical Manipulation". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 7166–7173.

[46] Dean A. Pomerleau. "ALVINN: An Autonomous Land Vehicle in a Neural Network". In: *Neural Information Processing Systems (NeurIPS)*. Ed. by D. Touretzky. Vol. 1. Morgan-Kaufmann, 1988.

[47] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. "A reduction of imitation learning and structured prediction to no-regret online learning". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2011, pp. 627–635.

[48] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[49] John Schulman et al. "Trust Region Policy Optimization". In: *International Conference on Machine Learning*. 2015.

[50] Daniel Seita et al. "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 9651–9658.

[51] Nur Muhammad Shafiullah et al. "Behavior Transformers: Cloning k modes with one stone". In: *Neural Information Processing Systems (NeurIPS)*. 2022.

[52] Evan Shelhamer, Jonathan Long, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2017), pp. 640–651.

[53] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. "Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation". In: *Conference on Robot Learning (CoRL)*. 2022.

[54] Jonathan Spencer et al. "Learning from Interventions: Human-robot Interaction as both Explicit and Implicit Feedback". In: *Robotics: Science and Systems (RSS)*. 2020.

[55] Xiatao Sun, Shuo Yang, and Rahul Mangharam. "MEGA-DAgger: Imitation Learning with Multiple Imperfect Experts". In: *ArXiv* arXiv preprint arXiv:2303.00638 (2023).

[56] Naftali Tishby and Noga Zaslavsky. "Deep learning and the information bottleneck principle". In: *2015 IEEE Information Theory Workshop (ITW)* (2015), pp. 1–5.

[57] Faraz Torabi, Garrett Warnell, and Peter Stone. "Generative Adversarial Imitation from Observation". In: *ArXiv* abs/1807.06158 (2018).

[58] Hsiao-Yu Tung et al. "Reward learning from narrated demonstrations". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7004–7013.

[59] Ashish Vaswani et al. "Attention Is All You Need". In: *Neural Information Processing Systems (NeurIPS)*. 2017.

[60] Max Welling and Yee Whye Teh. "Bayesian Learning via Stochastic Gradient Langevin Dynamics". In: *Proceedings of the 28th International Conference on Machine Learning*. ICML'11. Bellevue, Washington, USA: Omnipress, 2011, pp. 681–688. ISBN: 9781450306195.

[61] Andy Zeng et al. "Transporter Networks: Rearranging the Visual World for Robotic Manipulation". In: *Conference on Robot Learning (CoRL)* (2020).

[62] Jiakai Zhang and Kyunghyun Cho. "Query-Efficient Imitation Learning for End-to-End Autonomous Driving". In: *Association for the Advancement of Artificial Intelligence (AAAI)*. 2017.

Chapter 8

Appendix

8.1 Additional Details on Implicit Models

We use the following hyperparameters for implicit model training and inference:

Hyperparameter	Value
learning rate	0.0005
learning rate decay	0.99
learning rate decay steps	100
train counter-examples	8
langevin iterations	100
langevin learning rate init.	0.1
langevin learning rate final	1e-5
langevin polynomial decay power	2
inference counter-examples	512

Table 8.1: Implicit model hyperparameters.

8.2 Uncertainty Estimation with Larger Ensembles

Prior works using ensembles of explicit models to estimate epistemic uncertainty [11, 37, 24] typically employ larger ensembles of $n \geq 5$ models, whereas IIFL uses $n = 2$. We wish to evaluate the impact of this smaller number of models. However, the Jeffreys divergence is only defined for two distributions, and while other divergence measures (e.g. Jensen-Shannon) can be generalized to an arbitrary number of distributions, they typically require knowledge of the intractable partition functions of the distributions. Accordingly, we consider estimating the uncertainty of $n = 5$ implicit models by computing the average of the Jeffreys divergences between every pairwise combination of models. Figure 8.1 provides intuition on this measure.

We evaluate the effect of adding more models by comparing the estimate of the Jeffreys divergence with $n = 2$ models and the averaged estimate with $n = 5$ models to the L2 distance between the robot policy's proposed action and the expert policy's action at the same state. While ground truth epistemic uncertainty is intractable to calculate, the ground truth action discrepancy between the human and robot can provide a correlate of uncertainty: higher discrepancy corresponds to higher uncertainty. The results are shown in Figure 8.2. We observe that both ensemble sizes are positively correlated with action discrepancy, and that the ensemble with $n = 5$ models has a higher correlation ($r = 0.804$) than the ensemble with $n = 2$ models ($r = 0.688$). We also observe that the $n = 5$ ensemble has lower variance than $n = 2$: the standard deviation is 0.176 compared to 0.220. These results suggest that larger ensembles can improve the uncertainty estimation at the cost of increased computation time (2.6×, requiring 2.599 ± 0.002s). While the time complexity should grow as quadratic in n, in practice we observe that for small values of n the growth is closer to linear as the latency is dominated by the $O(n)$ sampling process rather than the $O(n^2)$ forward passes.

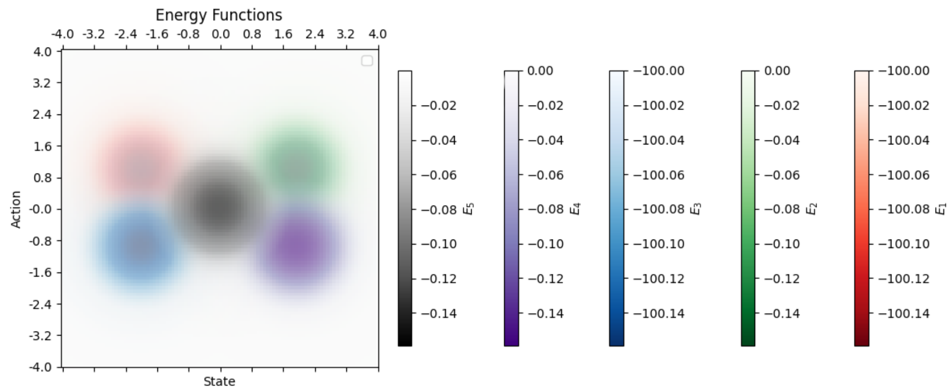
8.3 Additional Experimental Details

IFL Benchmark Hyperparameters

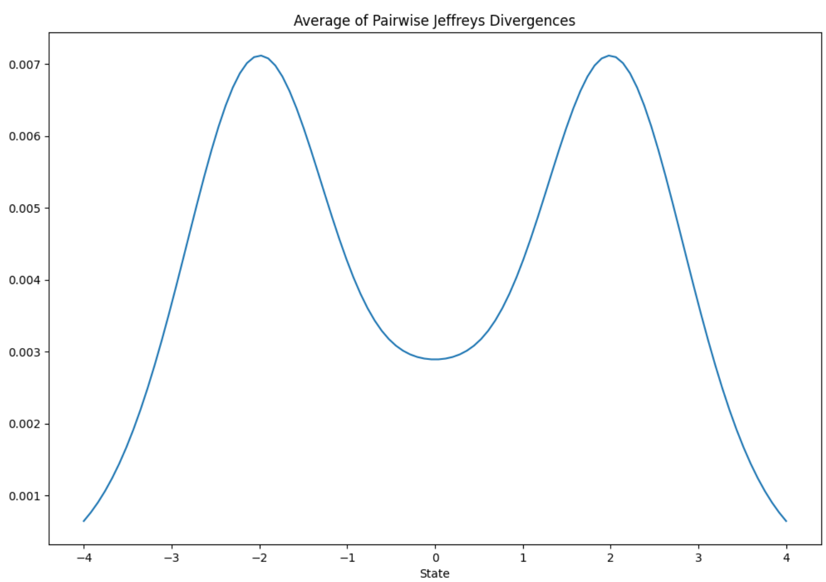
Implementations of Implicit Interactive Fleet Learning and baselines are available in the code supplement and are configured to run with the same hyperparameters we used in the experiments. To compute the uncertainty thresholds $\hat{u}$ for Explicit IFL and IIFL (see Section 8.3.1 in [21] for definition), we run Explicit BC and Implicit BC respectively with $N = 100$ robots for $T = 1000$ timesteps and choose the 99th percentile value among all 100×1000 uncertainty values. The FrankaCubeStack environment sets these thresholds to zero since there are no constraint violations (i.e., this sorts robot priority by uncertainty alone). See Table 8.2 for these values, state and action space dimensionality, and other hyperparameters. The batch size is 512 and all algorithms pretrain the policy for $N/2$ gradient steps, where N is the number of data points in the 10 offline task demonstrations. Finally, as in prior work [21], the Random IIFL baseline is given a human action budget that approximately equals the average amount of human supervision solicited by IIFL. See the code for more details.

Environment	$\lvert S\rvert$	$\lvert A\rvert$	Explicit $\hat{u}$	Implicit $\hat{u}$
BallBalance	24	3	0.1179	0.1206
Ant	60	8	0.0304	0.9062
Anymal	48	12	0.0703	2.2845
FrankaCubeStack	19	7	0.0	0.0

Table 8.2: Simulation environment hyperparameters.



(a) Consider 5 isotropic Gaussian energy functions, each a negative Gaussian probability density function with some offset.



(b) We use numerical integration to calculate at each state the Jeffreys divergences between each of the $\binom{5}{2} = 10$ unique pairs of models, and report the average value. As intuition suggests, the calculated uncertainty is highest at states -2 and 2, where two of the Gaussians have means that are far apart, meaning that they strongly prefer very different actions. At state 0, the uncertainty is lower as one model strongly prefers the action 0, and the others are closer to uniform. Far from state 0, the uncertainty is lowest as all the energy functions are approximately flat.

Figure 8.1

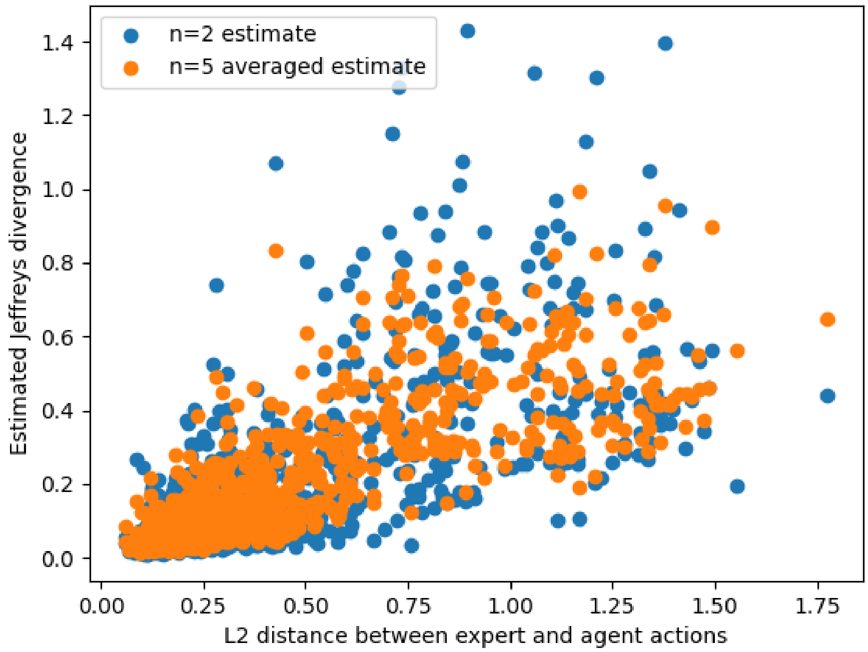


Figure 8.2: We plot the Jeffreys divergence estimates and the ground truth action discrepancies at the first 1000 states visited by a robot with a unimodal policy. Both variants of the Jeffreys divergence calculation are positively correlated with the $L2$ distance between the robot policy's and expert policy's actions. In the $n = 2$ case, the correlation coefficient is $r = 0.688$; in the $n = 5$ case, the correlation coefficient is $r = 0.804$, indicating that additional models can make the ensemble more predictive of when the agent will deviate from the expert (at the cost of increased computation time).

Physical Experiment Protocol

We largely follow the physical experiment protocol in Hoque et al. [21] but introduce some modifications to human supervision. We execute 3 trials of each of 4 algorithms (Explicit BC, Implicit BC, Explicit IFL, Implicit IFL) on the fleet of 4 robot arms. Each trial lasts 150 timesteps (synchronous across the fleet) for a total of $3 \times 4 \times 4 \times 150 = 7200$ individual pushing actions. The authors provide human teleoperation and hard resets, which differ from prior work due to the continuous action space and the square obstacle in the center of the workspace. Teleoperation is done using an OpenCV (https://opencv.org/) GUI by clicking on the desired end point of the end-effector in the overhead camera view. Hard resets are physical adjustments of the cube to a randomly chosen side of the obstacle. IIFL is trained online with updated data at $t = 50$ and $t = 100$ while IFL is updated at every timestep (with an equivalent total amount of gradient steps) to follow prior work [21].

The rest of the experiment protocol matches Hoque et al. [21]. The 2 ABB YuMi robots are located about 1 km apart; a driver program uses the Secure Shell Protocol (SSH) to connect to a machine that is connected to the robot via Ethernet, sending actions and receiving camera observations. Pushing actions are executed concurrently by all 4 arms using multiprocessing. We set minimum intervention time $t_T = 3$ and hard reset time $t_R = 5$. All policies are initialized with an offline dataset of 3360 image-action pairs (336 samples collected by the authors with 10× data augmentation). 10× data augmentation on the initial offline dataset as well as the online data collected during execution applies the following transformations:

- Linear contrast uniformly sampled between 85% and 115%
- Add values uniformly sampled between -10 and 10 to each pixel value per channel
- Gamma contrast uniformly sampled between 90% and 110%
- Gaussian blur with σ uniformly sampled between 0.0 and 0.3
- Saturation uniformly sampled between 95% and 105%
- Additive Gaussian noise with σ uniformly sampled between 0 and $\frac{1}{80} \times 255$