IIFL: Implicit Interactive Fleet Learning from Heterogeneous Human Supervisors

Anonymous Author(s)
Affiliation
Address
email

Abstract: Imitation learning has been applied to a range of robotic tasks, but can struggle when (1) robots encounter edge cases that are not represented in the training data (distribution shift) or (2) the human demonstrations are heterogeneous: taking different paths around an obstacle, for instance (multimodality). Interactive fleet learning (IFL) mitigates distribution shift by allowing robots to access remote human teleoperators during task execution and learn from them over time, but is not equipped to handle multimodality. Recent work proposes Implicit Behavior Cloning (IBC), which is able to represent multimodal demonstrations using energy-based models (EBMs). In this work, we propose addressing both multimodality and distribution shift with Implicit Interactive Fleet Learning (IIFL), the first extension of implicit policies to interactive imitation learning (including the single-robot, single-human setting). IIFL quantifies uncertainty using a novel application of Jeffreys divergence to EBMs. While IIFL is more computationally expensive than explicit methods, results suggest that IIFL achieves 4.5× higher return on human effort in simulation experiments and an 80% higher success rate in a physical block pushing task over (Explicit) IFL, IBC, and other baselines when human supervision is heterogeneous.

1 Introduction

Imitation learning (IL), the paradigm of learning from human demonstrations and feedback, has been applied to diverse tasks such as autonomous driving [1, 2, 3], robot-assisted surgery [4, 5], and deformable object manipulation [6, 7, 8]. The most common IL algorithm is behavior cloning (BC) [2], where the robot policy is derived via supervised machine learning on an offline set of human task demonstrations. Since BC can suffer from distribution shift between the states visited by the human and those visited by the robot, interactive IL (IIL) algorithms including DAgger [9] and variants [10, 11, 12] 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) [13] algorithms extend IIL to multiple robots and multiple human supervisors. However, all of these IL algorithms can be unreliable if the dataset is multimodal, where a state can have multiple action labels: $\{(s, a_i), (s, a_j), \dots\}, a_i \neq a_j$.

Human demonstrations and interventions often include heterogeneous behaviors that result in multimodal datasets [14, 15]. One human providing demonstrations of a task may make mistakes, become more proficient at the task over time, or execute a different valid action when subsequently encountering the same state. Moreover, standard robot tasks such as grasping, navigation, motion planning, and manipulation can typically be performed in multiple equally correct ways. These issues are exacerbated when learning from multiple humans (e.g. in the IFL setting above), since the humans have varying preferences and proficiency and may demonstrate the same task in different ways.

Florence et al. [16] propose Implicit Behavior Cloning, an IL algorithm that trains an energy-based model (EBM) [17] to represent state-action mappings *implicitly* rather than explicitly. While this

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

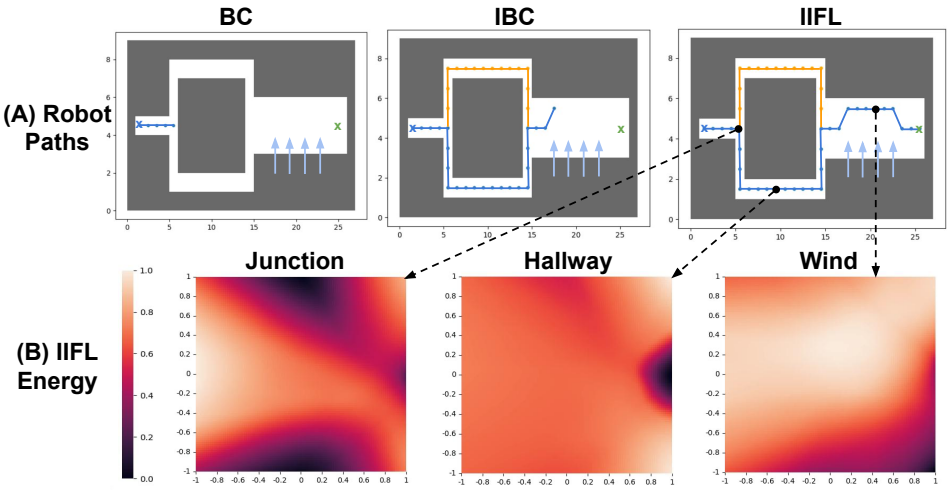

Figure 1: In the 2D navigation experiments from Section 5.1, the robot must navigate from the blue X marker on the left to the green X marker on the right, where the robot can go either above or below the rectangular grey obstacle and continue through a section subject to upward wind forces (blue arrows) 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 1×1 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.

makes model training and inference more computationally expensive (Section 6), implicit models can represent multiple actions for each state. However, IBC suffers from the same distribution shift problem as (Explicit) BC.

In this paper we combine implicit models with interactive fleet learning to address both distribution shift and multimodality. See Figure 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 (Section 4.3), we propose a new algorithm for estimating the epistemic uncertainty in EBMs using Jeffreys divergence [18].

This paper 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 using Jeffreys divergence, (3) simulation experiments with a fleet of 100 robots and 10 heterogeneous algorithmic supervisors, (4) physical experiments with a fleet of 4 robots and 2 heterogeneous human supervisors. Code to reproduce simulation results is available in the supplement and will be open-sourced if the paper is accepted.

2 Preliminaries and Related Work

2.1 Interactive Imitation Learning

Learning from an offline set of human task demonstrations with behavior cloning (i.e., supervised learning) is an intuitive and effective way to train a robot control policy [19, 2, 6, 1]. However, behavior cloning can suffer from distribution shift [9], 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. To mitigate distribution shift, Ross et al. [9] 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 [11, 10, 20]. Human-gated IIL [11, 21, 22] has the human decide when to take and cede control, while robot-gated IIL [23, 10, 12, 20] has the robot autonomously decide. Hoque et al. [13] 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 et al. [24] 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. [25] 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 [26].

2.2 Robot Learning from Multimodal Data

Learning from multimodal demonstrations is an active challenge in machine learning and robotics. A mixture density network [27] 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 [28] can be applied to learning pixelwise multimodality [8, 29]. Shafiullah et al. [30] propose Behavior Transformers, a technique that applies the multi-token prediction of Transformer neural networks [31] to imitation learning. Other Transformer-based policies report similar benefits for multimodal data [32, 33]; however, these approaches require action discretization to cast behavior prediction as next-token prediction. In a very recent paper, Chi et al. [34] introduce diffusion policies, an application of diffusion models [35] to imitation learning from multimodal data.

Florence et al. [16] propose implicit behavior cloning, a technique that trains a conditional energy-based model [17] 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:

$$\hat{a} = \arg\min_{a \in \mathcal{A}} E(s, a) \tag{1}$$

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 imitation learning.

2.3 Jeffreys Divergence

The "Jeffreys divergence" [18] 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 [36] and generative adversarial networks [37]. It has also been used for dimensionality reduction [38], information bottlenecks [39], and policy gradient methods for reinforcement learning [40, 41]. The Jensen-Shannon divergence [42] is another symmetric KL divergence that sums the KL divergences of both distributions against the mixture of the two, but neither it nor the asymmetric KL have the structural properties that make Jeffreys divergence amenable to our setting (Section 4.3). Nielsen [43] derives a proposition similar to Identity 1 (Section 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.

3 Problem Statement

We consider the interactive fleet learning (IFL) setting proposed by Hoque et al. [13]. A fleet of N robots operate in parallel independent Markov Decision Processes (MDPs) that are identical apart from their initial state distributions. 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). As in [13], 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 [13], 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, with no more than one human per robot and one robot per human at a time:

$$\omega : (\mathbf{s}^t, \pi_{\theta_t}, \cdot) \mapsto \boldsymbol{\alpha}^t \in \{0,1\}^{N \times M} \quad \text{s.t.} \quad \sum_{j=1}^{M} \boldsymbol{\alpha}_{ij}^t \leq 1 \text{ and } \sum_{i=1}^{N} \boldsymbol{\alpha}_{ij}^t \leq 1 \quad \forall i, j. \tag{2}$$

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 to help the robots [13]:

$$\max_{\omega \in \Omega} \mathbb{E}_{\tau \sim p_{\omega,\theta_0}(\tau)} \left[\frac{M}{N} \cdot \frac{\sum_{t=0}^{T} \bar{r}(\mathbf{s}^t, \mathbf{a}^t)}{1 + \sum_{t=0}^{T} \|\omega(\mathbf{s}^t, \pi_{\theta_t}, \boldsymbol{\alpha}^{t-1}, \mathbf{x}^t)\|_F^2} \right] \tag{3}$$

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}.

4 Approach

4.1 Preliminaries: Implicit Models

We build on Implicit Behavior Cloning [16]. IBC seeks to learn a conditional energy-based model $E : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, 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:

$$\pi(a|s) = \frac{e^{-E(s,a)}}{Z(s)} \tag{4}$$

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 and train E_θ on samples $\{s_i, a_i\}$ collected from the expert policies π_H. Training E_θ is described in Appendix 7.2.

4.2 Implicit Interactive Dataset Aggregation

Behavior cloning is prone to distribution shift due to compounding approximation errors [9], and any data-driven robot policy may encounter edge cases during execution that are not represented in the training data [13]. We extend IBC to interactive imitation learning using dataset aggregation of online human data, and iteratively updating the shared robot policy with the aggregate dataset at a fixed interval $1 \leq \hat{t} \leq T$ via supervised learning, as in DAgger [9] and variants [11, 13]:

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

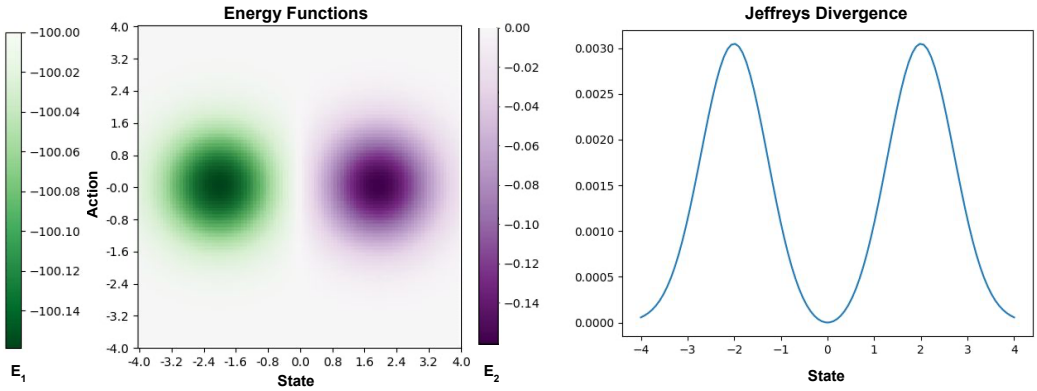

Figure 2: Consider a pair of isotropic Gaussian energy functions $E_1(s, a)$ and $E_2(s, a)$ in green and purple respectively, where each function is a negated Gaussian probability density function and E_1 adds a uniform offset of $Z = -100$ to all values (Left). Using numerical integration to directly 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.

where $\pi_H^j(s_i^t)$ is the teleoperation action from human j for robot i at time t, and $\boldsymbol{\alpha}_{ij}^t$ is the assignment of human j to robot i at time t, as in Equation 2. Ross et al. [9] 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 an ensemble of neural networks to estimate epistemic uncertainty [44, 12, 10], this approach is not applicable to implicit policies because multimodality results in a false positive: different ensemble members may select equally good 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 (Section 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 applicable by considering their *distributions* at a given state rather than the single predicted actions. Accordingly, we propose bootstrapping 2 implicit policies and calculating the Jeffreys divergence D_J [18] between them. 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 possible to compute for EBMs without estimating the intractable partition function $Z(s)$ (Equation 4). To show (2), we derive the following novel identity (proof in Appendix 7.1):

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. Then,*

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

Crucially, the intractable partition functions are absent from the expression due to the symmetry of Jeffreys divergence. We estimate the expectations in Identity 1 using Langevin sampling. See Figure 2 for intuition on the proposed metric. Note that this method may have broad applications beyond IIL for any algorithms or systems that use energy-based models.

4.4 Energy-Based Allocation

To extend IBC to the IFL setting, we synthesize the Jeffreys uncertainty estimate with Fleet-DAgger [13]. Specifically, we set the Fleet-DAgger priority function $\hat{p} : (s, \pi_{\theta_t}) \rightarrow [0, \infty)$ to prioritize

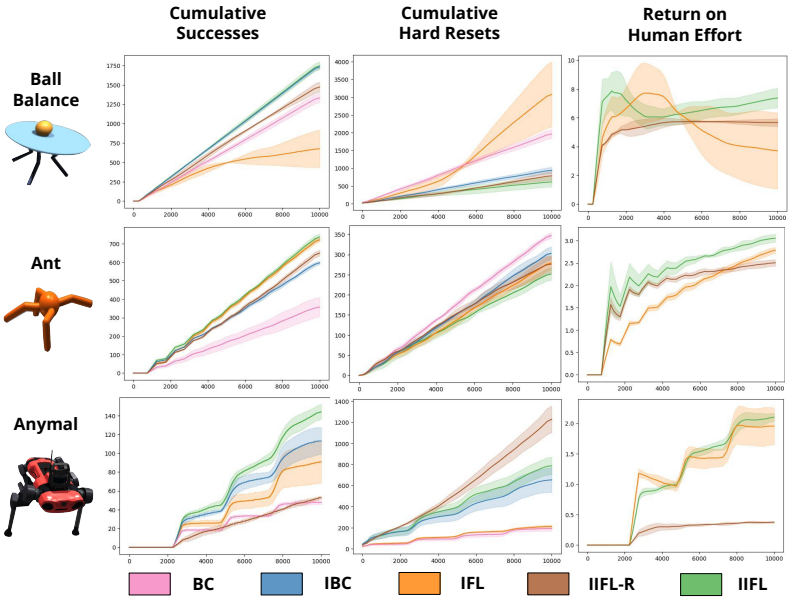

Figure 3: IFL Benchmark simulation experiment results. Despite unimodal supervision, IIFL is competitive with or outperforms IFL and other baselines across 3 environments, suggesting benefits of implicit policies beyond robustness to multimodality. Shading represents ± 1 standard deviation.

robots with high uncertainty, followed by robots that require a hard reset R. This produces a supervisor allocation policy ω with Fleet-EnsembleDAgger, the U.C. (Uncertainty-Constraint) allocation scheme in [13]. We refer to the composite approach as IIFL.

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 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 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 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

Algorithm	Avg. Reward	Task Successes	ROHE
BC	11.07 ± 1.17	0.0 ± 0.0	N/A
IBC	30.32 ± 2.78	0.0 ± 0.0	N/A
IFL	131.42 ± 18.75	4.0 ± 0.8	1.31 ± 0.19
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 1: Execution results from the FrankaCubeStack Isaac Gym environment. IIFL significantly outperforms the baselines in average reward, task successes, and ROHE.

these requirements, following prior work [13] we evaluate with Isaac Gym [45] and the IFL Benchmark [13]. 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 7.3.

Metrics: Following prior work [13], 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) 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 [13]), and IIFL-Random (IIFL-R), which is an ablation of IIFL that allocates humans to robots randomly instead of using the Jeffreys metric. 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 [41]. 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, and are averaged over 3 random seeds. The initial robot policy π_{θ_0} for all algorithms is initialized with behavior cloning on 10 full task demonstrations. While IFL trains at every timestep following prior work [13], 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 (see Appendix 7.3.2 for images and details), has several differences from the other 3 environments. First, since it allows Cartesian space control, we can script 2 *heterogeneous* supervisor policies that grasp the cube at different pairs of antipodal faces; the $M = 10$ supervisors are split into 2 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.

Results: The results are shown in Figure 3 and Table 1. In the homogeneous control experiments, we observe that Implicit Interactive Fleet Learning rivals or outperforms all baselines across all metrics, with the exception of hard resets in the Anymal environment. 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 [16]. In the FrankaCubeStack environment, IIFL significantly outperforms the baselines across all metrics, indicating the value of implicit policies for heterogeneous supervision. Only IFL and IIFL attain a nonzero success rate; while IIFL-R makes progress, it is not able to successfully stack the cube, suggesting that IIFL allocates human attention more judiciously.

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

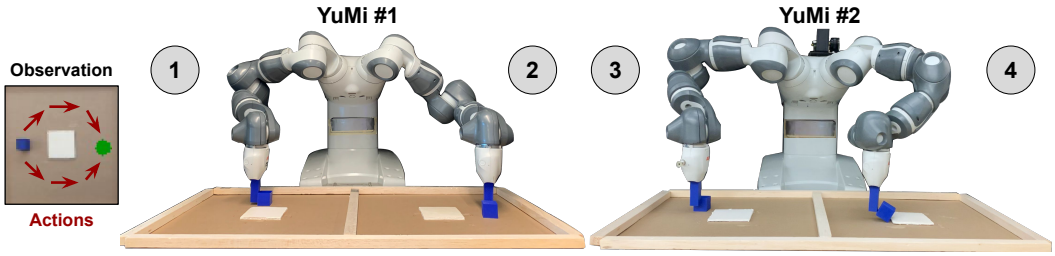

Figure 4: Physical experiment setup with 2 ABB YuMi robots for a total of 4 independent arms.

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 2: 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.

et al. [13]. See Figure 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. [13], 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 Section 3, the role of human superivsion is to (1) teleoperate when requested and (2) provide a physical hard 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. [13], 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 2 for $T = 150$ timesteps; see Appendix 7.3.3 for more details.

Results: The results are shown in Table 2. 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.

6 Limitations and Future Work

Since IIFL extends IBC, it inherits some of its limitations. First, model training and inference require $18\times$ and $82\times$ more computation time than explicit methods: on one NVIDIA V100 GPU, we measure implicit training to take an additional 0.34 seconds per gradient step and implicit inference to take an additional 0.49 seconds. Second, Florence et al. [16] find that IBC performance falls 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. Lastly, while it is $7\times$ faster than alternate implicit methods and has sub-second latency for a fleet of 100 robots, IIFL uncertainty estimation is nevertheless $340\times$ slower than its explicit counterpart (Appendix 7.3.4).

In future work, we will extend recently proposed alternative approaches for handling multimodality such as Behavior Transformers [30] and Diffusion Policies [34] to the IFL setting and compare them to IIFL. We will also develop algorithms that effectively learn from human demonstrations that are not only multimodal but also suboptimal. We note that the Jeffreys uncertainty quantification method does not rely on any IFL assumptions, so it may be broadly useful beyond this setting to any applications involving Boltzmann distributions and EBMs.

References

[1] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, and B. Boots. Agile autonomous driving using end-to-end deep imitation learning. In *Robotics: Science and Systems (RSS)*, 2018.

[2] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky, editor, *Neural Information Processing Systems (NeurIPS)*, volume 1. Morgan-Kaufmann, 1988.

[3] J. Chen, B. Yuan, and M. Tomizuka. Deep imitation learning for autonomous driving in generic urban scenarios with enhanced safety. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2884–2890, 2019.

[4] S. Paradis, M. Hwang, B. Thananjeyan, J. Ichnowski, D. Seita, D. Fer, T. Low, J. E. Gonzalez, and K. Goldberg. 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)*, pages 7166–7173, 2021.

[5] J. W. Kim, P. Zhang, P. L. Gehlbach, I. I. Iordachita, and M. Kobilarov. Towards autonomous eye surgery by combining deep imitation learning with optimal control. In *Conference on Robot Learning (CoRL)*, 2020.

[6] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali, et al. Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9651–9658, 2020.

[7] Y. Avigal, L. Berscheid, T. Asfour, T. Kroger, and K. Goldberg. Speedfolding: Learning efficient bimanual folding of garments. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, 2022.

[8] R. Hoque, K. Shivakumar, S. Aeron, G. Deza, A. Ganapathi, A. Wong, J. Lee, A. Zeng, V. Vanhoucke, and K. Goldberg. 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.

[9] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 627–635, 2011.

[10] R. Hoque, A. Balakrishna, E. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg. ThriftyDAgger: Budget-aware novelty and risk gating for interactive imitation learning. In *Conference on Robot Learning (CoRL)*, 2021.

[11] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083, 2018.

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

[13] R. Hoque, L. Y. Chen, S. Sharma, K. Dharmarajan, B. Thananjeyan, P. Abbeel, and K. Goldberg. Fleet-dagger: Interactive robot fleet learning with scalable human supervision. In *Conference on Robot Learning (CoRL)*, 2022.

[14] C. G. Northcutt, A. Athalye, and J. Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[15] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Mart'in-Mart'in. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.

[16] P. R. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. S. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. In *Conference on Robot Learning (CoRL)*, 2021.

[17] Y. LeCun, S. Chopra, R. Hadsell, A. Ranzato, and F. J. Huang. A tutorial on energy-based learning. *Predicting Structured Data*, 1(0), 2006.

[18] H. Jeffreys. *The Theory of Probability*. Oxford University Press, 1939.

[19] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[20] J. Zhang and K. Cho. Query-efficient imitation learning for end-to-end autonomous driving. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[21] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa. Learning from interventions: Human-robot interaction as both explicit and implicit feedback. In *Robotics: Science and Systems (RSS)*, 2020.

[22] H. Liu, S. Nasiriany, L. Zhang, Z. Bao, and Y. Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. *arXiv*, abs/2211.08416, 2022.

[23] R. Hoque, A. Balakrishna, C. Putterman, M. Luo, D. S. Brown, D. Seita, B. Thananjeyan, E. Novoseller, and K. Goldberg. LazyDAgger: Reducing context switching in interactive imitation learning. In *IEEE Conference on Automation Science and Engineering (CASE)*, pages 502–509, 2021.

[24] X. Sun, S. Yang, and R. Mangharam. Mega-dagger: Imitation learning with multiple imperfect experts. *ArXiv*, arXiv preprint arXiv:2303.00638, 2023.

[25] K. Gandhi, S. Karamcheti, M. Liao, and D. Sadigh. Eliciting compatible demonstrations for multi-human imitation learning. In *Conference on Robot Learning (CoRL)*, 2022.

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

[27] C. M. Bishop. Mixture density networks. *Neural Computing Research Group Report*, 1994.

[28] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:640–651, 2017.

[29] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee. Transporter networks: Rearranging the visual world for robotic manipulation. *Conference on Robot Learning (CoRL)*, 2020.

[30] N. M. Shafiullah, Z. J. Cui, A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. In *Neural Information Processing Systems (NeurIPS)*, 2022.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Neural Information Processing Systems (NeurIPS)*, 2017.

[32] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

[33] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan. VIMA: General robot manipulation with multimodal prompts. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[34] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[35] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

[36] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems*, 2014.

[38] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[39] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

[40] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

[41] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[42] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[43] F. Nielsen. Fast approximations of the jeffreys divergence between univariate gaussian mixtures via mixture conversions to exponential-polynomial distributions. *Entropy*, 23, 2021.

[44] K. Chua, R. Calandra, R. T. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Neural Information Processing Systems*, 2018.

[45] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[46] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *ArXiv preprint arXiv:1807.03748*, 2018.

[47] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[48] R. M. Neal. Annealed importance sampling. *Statistics and computing*, 11:125–139, 2001.