Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract: Interventions are an intuitive way for a human to convey information about desired behaviors to a robotic agent and improve the performance of the deployed robot policy. However, such interventions cost significant human time, giving rise to the central challenge in interactive imitation learning: *how can we control the timing and length of interventions to facilitate learning and limit burden on the human supervisor?* To address this challenge, we present ThriftyDAgger, a novel algorithm for actively querying a human supervisor for interventions given a desired human intervention budget. ThriftyDAgger uses a learned switching policy to solicit interventions only at states that are sufficiently (1) *novel*, where the robot policy has no reference behavior to imitate, or (2) *risky*, where the robot has low confidence in task completion. Experiments in simulation and on a physical cable routing experiment suggest that ThriftyDAgger balances task performance and supervisor burden more effectively than prior algorithms. After training, ThriftyDAgger applied at execution time achieves a 100% success rate on both the simulation and physical tasks. A user study ($N = 10$) in which users control a three-robot fleet while also performing a concentration task suggests that ThriftyDAgger increases human and robot performance by 58% and 80% respectively compared to the next best algorithm while reducing supervisor burden. See `https://tinyurl.com/thrifty-dagger` for supplementary material.

Keywords: Imitation Learning, Fleet Learning, Human Robot Interaction

1 Introduction

Imitation learning (IL) [1, 2, 3] has seen success in a variety of robotic tasks ranging from autonomous driving [4, 5, 6] to robotic manipulation [7, 8, 9, 10, 11]. In its simplest form, the human provides an offline set of task demonstrations to the robot, which the robot uses to match human behavior. However, this offline approach can lead to low task performance due to a mismatch between the state distribution encountered in the demonstrations and that visited by the robot [12, 13], resulting in brittle policies that cannot be effectively deployed in real-world applications [14]. *Interactive imitation learning*, in which the robot periodically cedes control to a human supervisor for corrective interventions, has emerged as a promising technique to address these challenges [15, 16, 17, 18]. However, while interventions make it possible to learn robust policies, these interventions require significant human time. Thus, the central challenge in interactive IL algorithms is to control the timing and length of interventions to balance task performance and the burden imposed on the human supervisor [19, 18]. Achieving this balance is even more critical if the human supervisor must oversee multiple robots at once [20, 21, 22], for instance supervising a robot fleet in a warehouse. Since even relatively reliable robot policies inevitably encounter new situations that must fall back on human expertise, this problem is immediately relevant to modern day industrial applications.

One way to determine when to solicit interventions is to allow the human supervisor to decide when to provide the corrective interventions. However, these approaches—termed "human-gated" interactive IL algorithms [15, 16, 23]—require the human supervisor to continuously monitor the robot to determine when to intervene. This imposes significant burden on the supervisor and cannot effectively scale to settings in which a small number humans supervise a large number of robots. To address this challenge, there has been recent interest in approaches that enable the robot to actively query humans

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.

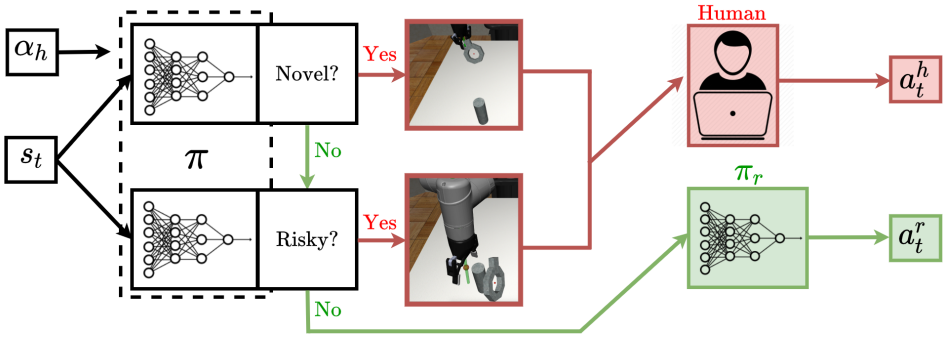

Figure 1: **ThriftyDAgger:** Given a desired context switching rate α_h, ThriftyDAgger transfers control to a human supervisor if the current state s_t is (1) sufficiently novel or (2) sufficiently risky, indicating that the probability of task success is low under robot policy π_r. Intuitively, one should not only distrust π_r in states significantly out of the distribution of previously-encountered states, but should also cede control to a human supervisor in more familiar states where the robot predicts that it is unlikely to successfully complete the task.

for interventions, called "robot-gated" algorithms [19, 24, 25, 18]. Robot-gated methods allow the robot to reduce burden on the human supervisor by only requesting interventions when necessary, switching between robot control and human control based on some intervention criterion. Hoque et al. [18] formalize the idea of supervisor burden as the expected total cost incurred by the human in providing interventions, which consists of the expected cost due to *context switching* between autonomous and human control and the time spent actually providing interventions. However, it is difficult to design intervention criteria that limit this burden while ensuring that the robot gains sufficient information to imitate the supervisor's policy.

This paper makes several contributions. First, we develop intervention criteria based on a synthesis of two estimated properties of a given state: *novelty*, which measures whether the state is significantly out of the distribution of previously encountered states, indicating that the robot policy should not be trusted; and *risk*, which indicates whether the robot is unlikely to make task progress. While state novelty has been considered in prior work [25], the key insight in our intervention criteria lies in combining novelty with a new risk metric to estimate the probability of task success. Second, we present a new robot-gated interactive IL algorithm, ThriftyDAgger (Figure 1), which employs these measures jointly to solicit human interventions only when necessary. Third, while prior robot-gated algorithms [19, 18] require careful parameter tuning to modulate the timing and frequency of human intervention requests, ThriftyDAgger only requires the supervisor to specify a desired context switching rate and sets thresholds accordingly. Fourth, experimental results demonstrate ThriftyDAgger's effectiveness for reducing supervisor burden while learning challenging tasks both in simulation and in an image-based cable routing task on a physical robot. Finally, the results of a human user study applying ThriftyDAgger to control a fleet of three simulated robots suggest that ThriftyDAgger significantly improves performance on both the robots' task and an independent human task while imposing fewer context switches, fewer human intervention actions, and lower mental load and frustration than prior algorithms.

2 Related Work

Imitation Learning from Human Feedback: There has been significant prior work in offline imitation learning, in which the agent leverages an offline dataset of expert demonstrations either to directly match the distribution of trajectories in the offline dataset [4, 26, 1, 3, 27, 28, 29], for instance via Behavior Cloning [30, 31], or to learn a reward function that can then be optimized via reinforcement learning [32, 26, 33]. However, while these approaches have shown significant success in a number of domains [7, 10, 9, 31], learning from purely offline data leads to a trajectory distribution mismatch which yields suboptimal performance both in theory and practice [12, 13]. To address this problem, there have been a number of approaches that utilize online human feedback while the agent acts in the environment, such as providing suggested actions [12, 34, 35, 17] or preferences [36, 37, 38, 39, 40, 41]. However, many of these forms of human feedback may be unreliable if the robot visits states that significantly differ from those the human supervisor would themselves visit; in such situations, it is challenging for the supervisor to determine what correct behavior should look like without directly interacting with the environment [16, 42].

Interactive Imitation Learning: A natural way to collect reliable online feedback for imitation learning is to periodically cede control to a human supervisor, who then provides a corrective intervention to illustrate desired behavior. Human-gated interactive IL algorithms [15, 16, 23] such as HG-DAgger require the human to determine when to engage in interventions. However, these algorithms require a human to continuously monitor the robot to determine when to intervene, which imposes significant burden on the supervisor and is particularly impractical if a small number of humans must supervise a large number of robots. Furthermore, it requires the human to determine when the robot needs help and when to cede control, which can be unintuitive and unreliable.

By contrast, robot-gated interactive IL algorithms, such as EnsembleDAgger [25], SafeDAgger [19], and LazyDAgger [18], allow the robot to actively query for human interventions. In practice, these algorithms estimate various quantities correlated with task performance [19, 18, 43, 24] and uncertainty [25] and use them to determine when to solicit interventions. Prior work has proposed intervention criteria which use the novelty of states visited by the robot [25] or the predicted discrepancy between the actions proposed by the robot policy and by the supervisor [19, 18]. However, while state novelty provides a valuable signal for soliciting interventions, we argue that this alone is insufficient, as a state's novelty does not convey information about the level of precision with which actions must be executed in that state. In practice, many robotic tasks involve moving through critical "bottlenecks" [23], which, though not necessarily novel, still present challenges. Examples include moving an eating utensil close to a person's mouth or placing an object on a shelf without disturbing nearby objects. Similarly, even if predicted accurately, action discrepancy is often a flawed risk measure, as high action discrepancy between the robot and the supervisor may be permissible when fine-grained control is not necessary (e.g. a robot gripper moving in free space) but impermissible when precision is critical (e.g. a robot gripper actively trying to grasp an object). In contrast, ThriftyDAgger presents an intervention criteria incorporating both state novelty and a novel risk metric and automatically tunes key parameters, allowing efficient use of human supervision.

3 Problem Statement

Given a robot, a task for the robot to accomplish, and a human supervisor with a specified context switching budget, the goal is to train the robot to imitate supervisor performance within the budget. We model the robot environment as a discrete-time Markov Decision Process (MDP) $\mathcal{M}$ with continuous states $s \in \mathcal{S}$, continuous actions $a \in \mathcal{A}$, and time horizon T [44]. We consider the interactive imitation learning (IL) setting [15], where the robot does not have access to a shaped reward function or to the MDP's transition dynamics but can temporarily cede control to a supervisor who uses policy $\pi_h : \mathcal{S} \to \mathcal{A}$. We specifically focus on tasks where there is a goal set $\mathcal{G}$ which determines success, but that can be challenging and long-horizon, making direct application of RL highly sample inefficient.

We assume that the human and robot utilize the same action space (e.g. through a teleoperation interface) and that task success can be specified by convergence to some goal set $\mathcal{G} \subseteq \mathcal{S}$ within the time horizon (i.e., the task is successful if $\mathcal{G}$ is reached within T timesteps). We further assume access to an indicator function $\mathbb{1}_{\mathcal{G}} : \mathcal{S} \to \{0, 1\}$, which indicates whether a state belongs to the goal set $\mathcal{G}$.

The IL objective is to minimize a surrogate loss function $J(\pi_r)$ to encourage the robot policy $\pi_r : \mathcal{S} \to \mathcal{A}$ to match π_h:

$$J(\pi_r) = \sum_{t=1}^{T} \mathbb{E}_{s_t \sim d_t^{\pi_r}} \left[\mathcal{L}(\pi_r(s_t), \pi_h(s_t)) \right], \tag{1}$$

where $\mathcal{L}(\pi_r(s), \pi_h(s))$ is an action discrepancy measure between $\pi_r(s)$ and $\pi_h(s)$ (e.g. MSE loss), and $d_t^{\pi_r}$ is the marginal state distribution at timestep t induced by the robot policy π_r in $\mathcal{M}$.

In the interactive IL setting, meanwhile, in addition to optimizing Equation (1), a key design goal is to minimize the imposed burden on the human supervisor. To formalize this, we define a switching policy π, which determines whether the system is under robot control π_r (which we call *autonomous mode*) or human supervisor control π_h (which we call *supervisor mode*). Following prior work [18], we define $C(\pi)$, the expected number of *context switches* in an episode under policy π, as follows: $C(\pi) = \sum_{t=1}^{T} \mathbb{E}_{s_t \sim d_t^{\pi}} \left[m_I(s_t; \pi) \right]$, where $m_I(s_t; \pi)$ is an indicator for whether or not a context switch occurs from autonomous to supervisor control. Similarly, we define $I(\pi)$ as the expected number of *supervisor actions* in an intervention solicited by π. We then define the total burden $B(\pi)$ imposed on the human supervisor as follows:

$$B(\pi) = C(\pi) \cdot \big(L + I(\pi)\big), \tag{2}$$

where L is the *latency* of a context switch between control modes (summed over both switching directions) in units of timesteps (one action per timestep). The interactive IL objective is to minimize the discrepancy from the supervisor policy while limiting supervisor burden within some Γ_{b}:

$$\pi = \arg\min_{\pi' \in \Pi} \{ J(\pi_r) \mid B(\pi') \leq \Gamma_{\mathrm{b}} \}. \tag{3}$$

Because optimizing policies explicitly to satisfy the supervisor burden constraint in Equation (3) is challenging, we present novel intervention criteria that enable reduction of supervisor burden by limiting the total number of interventions to a user-specified budget. Given sufficiently high latency L, limiting the interventions $C(\pi)$ directly corresponds to limiting supervisor burden $B(\pi)$.

4 ThriftyDAgger

ThriftyDAgger determines when to switch between autonomous and human supervisor control modes by leveraging estimates of both the *novelty* and *risk* of states. Below, Sections 4.1 and 4.2 discuss the estimation of state novelty and risk of task failure, respectively, while Section 4.3 discusses ThriftyDAgger's integration of these measures to determine when to switch control modes. Section 4.4 then describes an online procedure to set thresholds for switching between control modes. Finally, Section 4.5 describes the full control flow of ThriftyDAgger.

4.1 Novelty Estimation

When the robot policy visits states that lie significantly outside the distribution of those encountered in the supervisor trajectories, it does not have any reference behavior to imitate. This motivates initiating interventions to illustrate desired recovery behaviors in these states. However, estimating the support of the state distribution visited by the human supervisor is challenging in the high-dimensional state spaces common in robotics. Following prior work [25], we train an ensemble of policies with bootstrapped samples of transitions from supervisor trajectories. We then measure the novelty of a given state s by calculating the variance of the policy outputs at state s across ensemble members. In practice, the action $a \in \mathcal{A}$ outputted by each policy is a vector; thus, we measure state novelty by computing the variance of each component of the action vector a across the ensemble members and then averaging over the components. We denote this quantity by $\mathrm{Novelty}(s)$. Once in supervisor mode, as noted in Hoque et al. [18], we can obtain a more precise correlate of novelty by computing the ground truth action discrepancy between actions suggested by the supervisor and the robot policy.

4.2 Risk Estimation

Interventions may be required not only in novel states outside the distribution of supervisor trajectories, but also in familiar states that are prone to result in task failure. For example, a task might have a "bottleneck" region with low tolerance for error, which has low novelty but nevertheless requires more supervision to learn a reliable robot policy. To address this challenge, we propose a novel measure of a state's "riskiness," capturing the likelihood that the robot cannot successfully converge to the goal set $\mathcal{G}$. We first define a Q-function to quantify the discounted probability of successful convergence to $\mathcal{G}$ from a given state and action under the robot policy:

$$Q_{\mathcal{G}}^{\pi_r}(s_t, a_t) = \mathbb{E}_{\pi_r}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} \mathbb{1}_{\mathcal{G}}(s'_t) | s_t, a_t\right], \tag{4}$$

where $\mathbb{1}_{\mathcal{G}}(s_t)$ is equal to 1 if s_t belongs to $\mathcal{G}$. We estimate $Q_{\mathcal{G}}^{\pi_r}(s_t, a_t)$ via a function approximator $\hat{Q}_{\phi,\mathcal{G}}^{\pi_r}$ parameterized by ϕ, and define a state's riskiness in terms of this learned Q-function:

$$\mathrm{Risk}^{\pi_r}(s, a) = 1 - \hat{Q}_{\phi,\mathcal{G}}^{\pi_r}(s, a). \tag{5}$$

In practice, we train $\hat{Q}_{\phi,\mathcal{G}}^{\pi_r}$ on transitions (s_t, a_t, s_{t+1}) collected by the supervisor from both offline data and online interventions by minimizing the following MSE loss inspired by [45]:

$$J_{\mathcal{G}}^{Q}(s_t, a_t, s_{t+1}; \phi) = \frac{1}{2}\left(\hat{Q}_{\phi,\mathcal{G}}^{\pi_r}(s_t, a_t) - (\mathbb{1}_{\mathcal{G}}(s_t) + (1 - \mathbb{1}_{\mathcal{G}}(s_t))\gamma \hat{Q}_{\phi,\mathcal{G}}^{\pi_r}(s_{t+1}, \pi_r(s_{t+1})))\right)^2. \tag{6}$$

Note that since $\hat{Q}_{\phi,\mathcal{G}}^{\pi_r}$ is only used to solicit interventions, it must only be accurate enough to distinguish risky states from others, rather than be able to make the fine-grained distinctions between different states required for accurate policy learning in reinforcement learning.

4.3 Regulating Switches in Control Modes

We now describe how ThriftyDAgger leverages the novelty estimator from Section 4.1 and the risk estimator from Section 4.2 to regulate switches between autonomous and supervisor control. While in autonomous mode, the switching policy π initiates a switch to supervisor mode at timestep t if either (1) state s_t is sufficiently unfamiliar or (2) the robot policy has a low probability of task success from s_t. Stated precisely, π initiates a switch to supervisor mode from autonomous mode at timestep t if the predicate $\text{Intervene}(s_t, \delta_h, \beta_h)$ evaluates to TRUE, where $\text{Intervene}(s_t, \delta_h, \beta_h)$ is TRUE if (1) $\text{Novelty}(s_t) > \delta_h$ or (2) $\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) > \beta_h$ and FALSE otherwise. Note that the proposed switching policy only depends on Risk^{π_r} for states which are *not* novel (as novel states already initiate switches to supervisor control regardless of risk), since the learned risk measure should only be trusted on states in the neighborhood of those on which it has been trained.

In supervisor mode, π switches to autonomous mode if the action discrepancy between the human and robot policy and the robot's task failure risk are both below threshold values (Section 4.4), indicating that the robot is in a familiar and safe region. Stated precisely, π switches to autonomous mode from supervisor mode if the predicate $\text{Cede}(s_t, \delta_r, \beta_r)$ evaluates to TRUE, where $\text{Cede}(s_t, \delta_r, \beta_r)$ is TRUE if (1) $||\pi_r(s_t) - \pi_h(s_t)||_2^2 < \delta_r$ and (2) $\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) < \beta_r$, and FALSE otherwise. Here, the risk metric ensures that the robot has a high probability of autonomously completing the task, while the coarser 1-step action discrepancy metric verifies that we are in a familiar region of the state space where the $\hat{Q}^{\pi_r}_{\phi,\mathcal{G}}$ values can be trusted. Motivated by prior work [18] and hysteresis control [46], we use stricter switching criteria in supervisor mode ($\beta_r < \beta_h$) to encourage lengthier interventions and reduce context switches experienced by the human supervisor.

4.4 Computing Risk and Novelty Thresholds from Data

One challenge of the control strategy presented in Section 4.3 lies in tuning the key parameters $(\delta_h, \delta_r, \beta_h, \beta_r)$ governing when context switching occurs. As noted in prior work [25], performance and supervisor burden can be sensitive to these thresholds. To address this difficulty, we assume that the user specifies their availability in the form of a desired intervention budget $\alpha_h \in [0, 1]$, indicating the desired proportion of timesteps in which interventions will be requested. This desired context switching rate can be interpreted in the context of supervisor burden as defined in Equation (2): if the latency of a context switch dominates the time cost of the intervention itself, limiting the expected number of context switches to within some intervention budget directly limits supervisor burden.

Given α_h, we set β_h to be the $(1 - \alpha_h)$-quantile of $\text{Risk}^{\pi_r}(s, \pi_r(s))$ for all states previously visited by π_r and set δ_h to be the $(1 - \alpha_h)$-quantile of $\text{Novelty}(s)$ for all states previously visited by π_r. We set δ_r to be the mean action discrepancy on the states visited by the supervisor after π_r is trained and set β_r to be the median of $\text{Risk}^{\pi_r}(s, \pi_r(s))$ for all states previously visited by π_r. (Note that β_r can easily be set to different quantiles to adjust mean intervention length if desired.) We find that these settings strike a balance between informative interventions and imposed supervisor burden.

4.5 ThriftyDAgger Overview

We now summarize the ThriftyDAgger procedure, with full pseudocode available in the supplement. ThriftyDAgger first initializes π_r via Behavior Cloning on offline transitions ($\mathcal{D}_h$ from the human supervisor, π_h). Then, π_r collects an initial offline dataset $\mathcal{D}_r$ from the resulting π_r, initializes $\hat{Q}^{\pi_r}_{\phi,\mathcal{G}}$ by optimizing Equation (5) on $\mathcal{D}_r \cup \mathcal{D}_h$, and initializes parameters $\beta_h, \beta_r, \delta_h$, and δ_r as in Section 4.4. We then collect data for N episodes, each with up to T timesteps. In each timestep of each episode, we determine whether robot policy π_r or human supervisor π_h should be in control using the procedure in Section 4.3. Transitions in autonomous mode are aggregated into $\mathcal{D}_r$ while transitions in supervisor mode are aggregated into $\mathcal{D}_h$. After each episode, π_r is updated via supervised learning on $\mathcal{D}_h$, while $\hat{Q}^{\pi_r}_{\phi,\mathcal{G}}$ is updated on $\mathcal{D}_r \cup \mathcal{D}_h$ to reflect the probability of task success of the resulting π_r.

5 Experiments

In the following experiments, we study whether ThriftyDAgger can balance task performance and supervisor burden more effectively than prior IL algorithms in three contexts: (1) training a simulated robot to perform a peg insertion task (Section 5.3); (2) supervising a fleet of three simulated robots to perform the peg insertion task in a human user study (Section 5.4); and (3) training a physical surgical robot to perform a cable routing task (Section 5.5). In the supplementary material, we also include results from an additional simulation experiment on a challenging block stacking task.

5.1 Evaluation Metrics

We consider ThriftyDAgger's performance during training and execution. For the latter, we evaluate both the (1) *autonomous success rate*, or success rate when deployed after training without access to a human supervisor, and (2) *intervention-aided success rate*, or success rate when deployed after training with a human supervisor in the loop. These metrics are reported in the Peg Insertion study (Section 5.3) and the Physical Cable Routing study (Section 5.5). For all experiments, during both training and intervention-aided execution, we evaluate the number of interventions, human actions, and robot actions per episode. These metrics are computed over successful episodes only to prevent biasing the metrics by the maximum episode horizon length T; such bias occurs, for instance, when less successful policies appear to take more actions due to hitting the time boundary more often. Additional metrics including cumulative statistics across all episodes are reported in the supplement. In our user study (Section 5.4), we also report the following quantities: throughput (total number of task successes across the three robots), performance on an independent human task, the idle time of the robots in the fleet, and users' qualitative ratings of mental load and frustration. By comparing the amount of human supervision and success rates across different algorithms, we are interested in evaluating how effectively each algorithm balances supervision with policy performance.

5.2 Comparisons

We compare ThriftyDAgger to the following algorithms: Behavior Cloning, which does not use interventions; HG-DAgger [15], which is human-gated and always requires supervision; SafeDAgger [19], which is robot-gated and performs interventions based on estimated action discrepancy between the human supervisor and robot policy; and LazyDAgger [18], which builds on SafeDAgger by introducing an asymmetric switching criterion to encourage lengthier interventions. We also implement two ablations: one that does not use a novelty measure to regulate context switches (ThriftyDAgger (-Novelty)) and one that does not use risk to regulate context switches (ThriftyDAgger (-Risk)).

5.3 Peg Insertion in Simulation

We first evaluate ThriftyDAgger on a long-horizon peg insertion task (Figure 2) from the Robosuite simulation environment [47]. The goal is to grasp a ring in a random initial pose and thread it over a cylinder at a fixed target location. This task has two bottlenecks which motivate learning from interventions: (1) correctly grasping the ring and (2) correctly placing it over the cylinder (Figure 2). A human teleoperates the robot through a keyboard interface to provide interventions. The states consist of the robot's joint angles and ring's pose, while the actions specify 3D translation, 3D rotation, and opening or closing the gripper. For ThriftyDAgger and its ablations, we use target intervention frequency $\alpha_h = 0.01$ and set other parameters via the automated tuning method (Section 4.4). We collect 30 offline task demos (2,687 state-action pairs) from a human supervisor to initialize the robot policy for all compared algorithms. Behavior Cloning is given 15 additional offline demos for a fair comparison. For ThriftyDAgger and each interactive IL baseline, we perform 10,000 environment steps, during which each episode takes at most 175 timesteps and system control switches between the human and robot. Hyperparameter settings for all algorithms are detailed in the supplement.

Results (Table 1) suggest that ThriftyDAgger achieves a significantly higher autonomous success rate than prior robot-gated algorithms, although it does request more human actions due to its conservative exit criterion for interventions (Cede(s_t, δ_r, β_r)). However, the number of interventions is similar to prior robot-gated algorithms, indicating that while ThriftyDAgger requires more human actions, it imposes a similar supervisor burden to SafeDAgger and LazyDAgger in settings in which context switches are expensive or time-consuming (e.g. high latency L in Equation 2). We find that all interactive IL algorithms substantially outperform Behavior Cloning, which does not have access to supervisor interventions. Notably, ThriftyDAgger achieves a higher autonomous success rate than even HG-DAgger, in which the supervisor is able to decide the timing and length of interventions. This indicates that ThriftyDAgger's intervention criteria enable it to autonomously solicit interventions as informative as those chosen by a human supervisor with expert knowledge of the task. Furthermore, ThriftyDAgger achieves a 100% intervention-aided success rate at execution time, suggesting that ThriftyDAgger successfully identifies the required states at which to solicit interventions. We find that both ablations of ThriftyDAgger (Ours (-Novelty) and Ours (-Risk)) achieve significantly lower autonomous success rates, indicating that both the novelty and risk measures are critical to ThriftyDAgger's performance. We calculate ThriftyDAgger's context switching rate to be 1.15% novelty switches and 0.79% risk switches, both approximately within the budget of $\alpha_h = 0.01$.

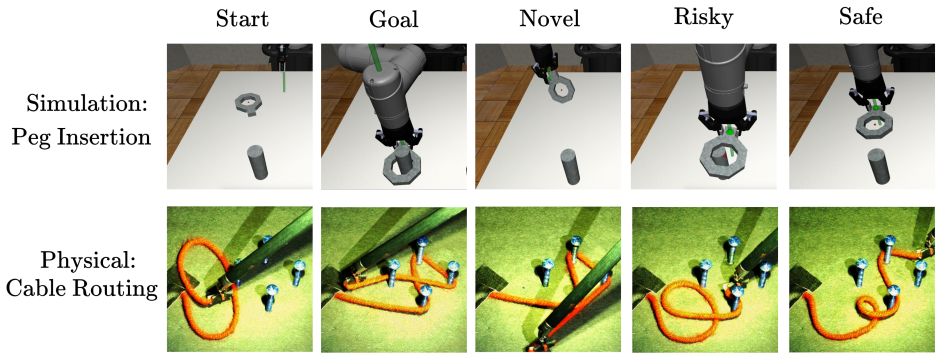

Figure 2: **Experimental Domains:** We visualize the peg insertion simulation domain (top row) and the physical cable routing domain with the physical robot (bottom row). We visualize sample start and goal states, in addition to states which ThriftyDAgger categorizes as novel, risky, and neither. ThriftyDAgger marks states as novel if they are far from behavior that the supervisor would produce, and risky if it is stuck in a bottleneck, e.g. if the ring is wedged against the side of the cylinder (top) or the cable is near all four obstacles (bottom).

Table 1: **Peg Insertion in Simulation Results:** We first report training performance (number of interventions (Ints), number of human actions (Acts (H)), and number of robot actions (Acts (R))) and report the success rate of the fully-trained policy at execution time when no interventions are allowed (Auto Succ.). We then evaluate the fully-trained policies with interventions allowed and report the same intervention statistics and the success rate (Int-Aided Succ.). We find that ThriftyDAgger achieves the highest autonomous and intervention-aided success rates among all algorithms compared. Notably, ThriftyDAgger even achieves a higher autonomous success rate than HG-DAgger, in which the human decides when to intervene during training.

Algorithm	Training Interventions			Auto Succ.	Execution Interventions			Int-Aided Succ.
	Ints	Acts (H)	Acts (R)		Ints	Acts (H)	Acts (R)	
Behavior Cloning	N/A	N/A	108.0 ± 15.9	24/100	N/A	N/A	N/A	N/A
SafeDAgger	3.89 ± 1.44	19.8 ± 9.9	88.8 ± 19.4	24/100	4.00 ± 1.37	19.5 ± 5.3	77.5 ± 11.7	17/20
LazyDAgger	1.46 ± 1.15	13.2 ± 12.4	102.1 ± 18.2	48/100	1.73 ± 1.29	12.6 ± 14.4	91.7 ± 24.0	11/20
HG-DAgger	1.49 ± 0.88	20.3 ± 15.6	97.1 ± 17.5	57/100	1.15 ± 0.73	17.1 ± 11.6	103.6 ± 14.0	**20/20**
Ours (-Novelty)	$\mathbf{0.79 \pm 0.81}$	35.1 ± 23.1	70.0 ± 35.8	49/100	$\mathbf{0.33 \pm 0.62}$	2.5 ± 5.0	114.0 ± 26.0	12/20
Ours (-Risk)	0.99 ± 0.96	7.8 ± 12.0	104.2 ± 19.2	49/100	1.39 ± 0.95	9.8 ± 12.0	109.1 ± 22.9	18/20
Ours: ThriftyDAgger	0.88 ± 1.01	43.6 ± 24.5	60.0 ± 32.8	**73/100**	1.35 ± 0.66	21.3 ± 15.0	84.8 ± 21.8	**20/20**

5.4 User Study: Controlling A Fleet of Three Robots in Simulation

We conduct a user study with 10 participants (7 male and 3 female, aged 18-37). Participants supervise a fleet of three simulated robots, each performing the peg insertion task from Section 5.3. We evaluate how different interactive IL algorithms affect the participants' (1) ability to provide effective robot interventions, (2) performance on a distractor task performed between robot interventions, and (3) levels of mental demand and frustration. For the distractor task, we use the game Concentration (also known as Memory or Matching Pairs), in which participants identify as many pairs of matching cards as possible among a set of face-down cards. This is intended to emulate tasks which require continual focus, such as cooking a meal or writing a research paper, in which frequent context switches between performing the task and helping the robots is frustrating and degrades performance.

The participants teleoperate the robots using three robot-gated interactive IL algorithms: SafeDAgger, LazyDAgger, and ThriftyDAgger. The participant is instructed to make progress on the distractor task only when no robot requests an intervention. When an intervention is requested, the participant is instructed to pause the distractor task, provide an intervention from the requested state until the robot (or multiple robots queued after each other) no longer requires assistance, and then return to the distractor task. The participants also teleoperate with HG-DAgger, where they no longer perform the distractor task and are instructed to continually monitor all three robots simultaneously and decide on the length and timing of interventions themselves. Each algorithm runs for 350 timesteps, where in each timestep, all robots in autonomous mode execute one action and the human executes one action on the currently-supervised robot (if applicable). The supplement illustrates the user study interface and fully details the experiment protocol. All algorithms are initialized as in Section 5.3.

Results (Table 2) suggest that ThriftyDAgger achieves significantly higher throughput than all prior algorithms while requiring fewer interventions and fewer human actions, indicating that ThriftyDAgger requests interventions more judiciously than prior algorithms. Furthermore, ThriftyDAgger also enables a lower mean idle time for robots and higher performance on the distractor task. Notably, ThriftyDAgger solicits fewer interventions and total actions while achieving a higher throughput than HG-DAgger, in which the participant chooses when to intervene. We also report metrics of users'

Table 2: **Three-Robot Fleet Control User Study Results:** Results for experiments with 10 human subjects and 3 simulated robots on the peg insertion task. We report the total numbers of interventions, human actions, and robot actions, as well as the throughput, or total task successes achieved across robots, for all algorithms. Additionally, for robot-gated algorithms, we report the Concentration score (number of pairs found) and the mean idle time of robots in the fleet in timesteps. Results suggest that ThriftyDAgger outperforms all prior algorithms across all metrics, requesting fewer interventions and total human actions while achieving higher throughput, lowering the robots' mean idle time, and enabling higher performance on the Concentration task.

Algorithm	Interventions	Human Actions	Robot Actions	Concentration Pairs	Throughput	Mean Idle Time
HG-DAgger	10.6 ± 2.5	198.0 ± 32.1	834.4 ± 38.1	N/A	5.1 ± 1.9	N/A
SafeDAgger	22.1 ± 4.8	234.1 ± 31.8	700.7 ± 70.4	17.7 ± 8.2	3.0 ± 2.4	38.4 ± 14.1
LazyDAgger	10.0 ± 2.1	219.5 ± 43.3	719.2 ± 89.7	20.9 ± 7.9	5.1 ± 1.7	37.1 ± 20.5
Ours: ThriftyDAgger	$\mathbf{7.9 \pm 2.1}$	$\mathbf{179.4 \pm 34.9}$	793.2 ± 86.6	$\mathbf{33.0 \pm 8.5}$	$\mathbf{9.2 \pm 2.0}$	$\mathbf{25.8 \pm 19.3}$

Table 3: **Physical Cable Routing Results:** We first report intervention statistics during training (number of interventions (Ints), number of human actions (Acts (H)), and number of robot actions (Acts (R))) and report the success rate of the fully-trained policy at execution time when no interventions are allowed (Auto Succ.). We then evaluate the fully-trained policies with interventions allowed and report the same intervention statistics and the success rate (Int-Aided Succ.). We find that ThriftyDAgger achieves the highest autonomous and intervention-aided success rates among all algorithms compared. Notably, ThriftyDAgger even achieves a higher autonomous success rate than HG-DAgger, in which the human decides when to intervene during training.

Algorithm	Training Interventions			Auto Succ.	Execution Interventions			Int-Aided Succ.
	Ints	Acts (H)	Acts (R)		Ints	Acts (H)	Acts (R)	
Behavior Cloning	N/A	N/A	N/A	$0/15$	N/A	N/A	N/A	N/A
HG-DAgger	1.55 ± 1.16	13.9 ± 10.9	55.5 ± 10.9	$10/15$	$\mathbf{0.40 \pm 0.49}$	2.7 ± 3.5	73.9 ± 7.9	$\mathbf{15/15}$
Ours: ThriftyDAgger	$\mathbf{1.42 \pm 1.14}$	15.2 ± 12.4	45.5 ± 18.3	$\mathbf{12/15}$	0.40 ± 0.71	1.5 ± 3.1	61.3 ± 6.5	$\mathbf{15/15}$

mental workload and frustration using the NASA-TLX scale [48] in the supplement. Results suggest that users experience lower degrees of frustration and mental load when interacting with ThriftyDAgger and LazyDAgger compared to HG-DAgger and SafeDAgger. We hypothesize that participants struggle with HG-DAgger due to the difficultly of monitoring multiple robots simultaneously, while SafeDAgger's frequent context switches lead to user frustration during experiments.

5.5 Physical Experiment: Visuomotor Cable Routing

Finally, we evaluate ThriftyDAgger on a long-horizon cable routing task with a da Vinci surgical robot [49]. Here, the objective is to route a red cable into a Figure-8 pattern around 4 pegs via teleoperation with the robot's master controllers (see supplement). The algorithm only observes high-dimensional $64 \times 64 \times 3$ RGB images of the workspace and generates continuous actions representing delta-positions in (x, y). As in Section 5.3, ThriftyDAgger uses a target intervention frequency of $\alpha_h = 0.01$. We collect 25 offline task demonstrations (1,381 state-action pairs) from a human supervisor to initialize the robot policy for ThriftyDAgger and all comparisons. We perform 1,500 environment steps, where each episode has at most 100 timesteps and system control can switch between the human and robot. The supplement details the hyperparameter settings for all algorithms.

Results (Table 3) suggest that both ThriftyDAgger and HG-DAgger achieve a significantly higher autonomous success rate than Behavior Cloning, which is never able to complete the task. Furthermore, ThriftyDAgger achieves a comparable autonomous success rate to HG-DAgger while requesting fewer interventions and a similar number of total human actions. This again suggests that ThriftyDAgger's intervention criteria enable it to solicit interventions equally as informative or more informative than those chosen by a human supervisor. Finally, at execution time ThriftyDAgger achieves a 100% intervention-aided success rate with minimal supervision, again indicating that ThriftyDAgger successfully identifies the timing and length of interventions to increase policy reliability.

6 Discussion and Future Work

We present ThriftyDAgger, a scalable robot-gated interactive imitation learning algorithm that leverages learned estimates of state novelty and risk of task failure to reduce burden on a human supervisor during training and execution. Experiments suggest that ThriftyDAgger effectively enables long-horizon robotic manipulation tasks in simulation, on a physical robot, and for a three-robot fleet while limiting burden on a human supervisor. In future work, we hope to apply ideas from ThriftyDAgger to interactive reinforcement learning and larger scale fleets of physical robots. We also hope to study how ThriftyDAgger's performance varies with the target supervisor burden (specified via α_h). In practice, α_h could even be time-varying: for instance, α_h may be significantly lower at night, when human operators may have limited availability. Similarly, α_h may be set to a higher value during training than at deployment, when the robot policy is typically higher quality.

References

[1] 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.

[2] J. A. Bagnell. An invitation to imitation. Technical Report CMU-RI-TR-15-08, Carnegie Mellon University, Pittsburgh, PA, 3 2015.

[3] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters. An algorithmic perspective on imitation learning. *arXiv preprint arXiv:1811.06711*, 2018.

[4] D. A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1), 1991.

[5] 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 *Proc. Robotics: Science and Systems (RSS)*, 2018.

[6] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4693–4700. IEEE, 2018.

[7] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine. One-shot visual imitation learning via meta-learning. *Conf. on Robot Learning (CoRL)*, 2017.

[8] B. Fang, S. Jia, D. Guo, M. Xu, S. Wen, and F. Sun. Survey of imitation learning for robotic manipulation. *International Journal of Intelligent Robotics and Applications*, 3(4):362–369, 2019.

[9] A. Ganapathi, P. Sundaresan, B. Thananjeyan, A. Balakrishna, D. Seita, J. Grannen, M. Hwang, R. Hoque, J. E. Gonzalez, N. Jamali, K. Yamane, S. Iba, and K. Goldberg. Learning Dense Visual Correspondences in Simulation to Smooth and Fold Real Fabrics. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2021.

[10] P. Sundaresan, P. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg. Learning interpretable and transferable rope manipulation policies using depth sensing and dense object descriptors. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[11] O. Kroemer, S. Niekum, and G. Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *J. Mach. Learn. Res.*, 22:30–1, 2021.

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

[13] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg. DART: Noise injection for robust imitation learning. In *Conf. on Robot Learning (CoRL)*, 2017.

[14] A. Irpan. Deep reinforcement learning doesn't work yet. https://www.alexirpan.com/2018/02/14/rl-hard.html, 2018.

[15] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. HG-DAgger: Interactive imitation learning with human experts. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2019.

[16] 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 immplicit feedback. In *Proc. Robotics: Science and Systems (RSS)*, 2020.

[17] S. Jauhri, C. Celemin, and J. Kober. Interactive imitation learning in state-space. In *Conference on Robot Learning (CoRL)*. PMLR, 2020.

[18] 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 *International Conference on Automation Sciences and Engineering (CASE)*, 2021.

[19] 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.

[20] J. W. Crandall, M. A. Goodrich, D. R. Olsen, and C. W. Nielsen. Validating human-robot interaction schemes in multitasking environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 35(4):438–449, 2005.

[21] J. Y. Chen, M. J. Barnes, and M. Harper-Sciarini. Supervisory control of multiple robots: Human-performance issues and user-interface design. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(4):435–454, 2010.

[22] G. Swamy, S. Reddy, S. Levine, and A. D. Dragan. Scaled autonomy: Enabling human operators to control robot fleets. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5942–5948. IEEE, 2020.

[23] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese. Human-in-the-loop imitation learning using remote teleoperation, 2020.

[24] M. Laskey, S. Staszak, W. Hsieh, J. Mahler, F. Pokorny, A. Dragan, and K. Goldberg. SHIV: Reducing supervisor burden using support vectors for efficient learning from demonstrations in high dimensional state spaces. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2016.

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

[26] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, 2016.

[27] S. Arora and P. Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *arXiv preprint arXiv:1806.06877*, 2018.

[28] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2), 2013.

[29] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann. Probabilistic movement primitives. In *Advances in neural information processing systems*, pages 2616–2624, 2013.

[30] F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 7 2018.

[31] F. Codevilla, E. Santana, L. A. M., and A. Gaidon. Exploring the limitations of behavior cloning for autonomous driving. *International Conference on Computer Vision*, 2019.

[32] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[33] D. S. Brown, W. Goo, and S. Niekum. Better-than-demonstrator imitation learning via automaticaly-ranked demonstrations. In *Conference on Robot Learning (CoRL)*, 2019.

[34] A. Balakrishna*, B. Thananjeyan*, J. Lee, F. Li, A. Zahed, J. E. Gonzalez, and K. Goldberg. On-policy robot imitation learning from a converging supervisor. In *Conference on Robot Learning (CoRL)*. PMLR, 2019.

[35] K. Judah, A. Fern, and T. Dietterich. Active imitation learning via state queries. In *Proceedings of the ICML workshop on combining learning strategies to reduce label cost*. Citeseer, 2011.

[36] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia. Active preference-based learning of reward functions. In *Proceedings of Robotics: Science and Systems (RSS)*, 2017.

[37] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[38] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei. Reward learning from human preferences and demonstrations in Atari. In *Advances in Neural Information Processing Systems*, 2018.

[39] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh. Learning reward functions by integrating human demonstrations and preferences. In *Proceedings of Robotics: Science and Systems (RSS)*, 2019.

[40] E. Bıyık, M. Palan, N. C. Landolfi, D. P. Losey, and D. Sadigh. Asking easy questions: A user-friendly approach to active reward learning. *arXiv preprint arXiv:1910.04365*, 2019.

[41] D. Brown, R. Coleman, R. Srinivasan, and S. Niekum. Safe imitation learning via fast Bayesian reward inference from preferences. In *International Conference on Machine Learning*, pages 1165–1177. PMLR, 2020.

[42] S. Reddy, A. D. Dragan, and S. Levine. Shared autonomy via deep reinforcement learning. *Proc. Robotics: Science and Systems (RSS)*, 2018.

[43] S. Ross and J. A. Bagnell. Reinforcement and imitation learning via interactive no-regret learning, 2014.

[44] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[45] B. Thananjeyan*, A. Balakrishna*, S. Nair, M. Luo, K. Srinivasan, M. Hwang, and J. E. Gonzalez. Recovery rl: Safe reinforcement learning with learned recovery zones. 2020.

[46] B. K. Bose. An adaptive hysteresis-band current control technique of a voltage-fed PWM inverter for machine drive system. *IEEE Transactions on Industrial Electronics*, 37(5), 1990. doi:10.1109/41.103436.

[47] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín. Robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

[48] S. G. Hart. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.

[49] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio. An open-source research kit for the da Vinci® surgical system. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 6434–6439. IEEE, 2014.

[50] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A Physics Engine for Model-Based Control. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.