IDIL: Imitation Learning of Intent-Driven Expert Behavior

Sangwon Seo
Rice University
Houston, USA
sangwon.seo@rice.edu

Vaibhav Unhelkar
Rice University
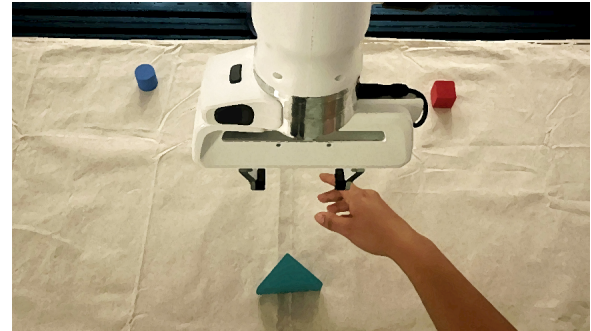Houston, USA
vaibhav.unhelkar@rice.edu

Figure 1: Consider the task of emptying a table. Different individuals may accomplish this task differently; some starting with the red block, while others with the green or blue. Intent-driven imitation learning aims to model this diversity in behaviors (arising from differences in experts' intents) from heterogeneous demonstrations.

ABSTRACT

When faced with accomplishing a task, human experts exhibit intentional behavior. Their unique intents shape their plans and decisions, resulting in experts demonstrating diverse behaviors to accomplish the same task. Due to the uncertainties encountered in the real world and their bounded rationality, experts sometimes adjust their intents, which in turn influences their behaviors during task execution. This paper introduces IDIL, a novel imitation learning algorithm to mimic these diverse intent-driven behaviors of experts. Iteratively, our approach estimates expert intent from heterogeneous demonstrations and then uses it to learn an intent-aware model of their behavior. Unlike contemporary approaches, IDIL is capable of addressing sequential tasks with high-dimensional state representations, while sidestepping the complexities and drawbacks associated with adversarial training (a mainstay of related techniques). Our empirical results suggest that the models generated by IDIL either match or surpass those produced by recent imitation learning benchmarks in metrics of task performance. Moreover, as it creates a generative model, IDIL demonstrates superior performance in intent inference metrics, crucial for human-agent interactions, and aptly captures a broad spectrum of expert behaviors.

CCS CONCEPTS

• **Computing methodologies** → *Learning latent representations*; *Learning from demonstrations*; *Semi-supervised learning settings*; • **Computer systems organization** → Robotics.

KEYWORDS

Hierarchical Imitation Learning, Human Modeling, Intention Prediction

1 INTRODUCTION

With the growing demand for AI systems that can seamlessly interact and emulate human behavior, modeling human behavior and learning thereof have been important problems in the AI community [2, 4, 20]. For instance, when developing an autonomous agent, behavior of human experts serves as a useful starting point,

especially for the tasks where humans excel. Learning from human experts proves especially beneficial in scenarios where rewards are sparse or ill-defined, rendering policy learning through reinforcements impractical [18, 57]. Modeling human behavior is also essential in human-agent interaction contexts, as agents (i.e. autonomous vehicles and robots) have to predict and reason about human behavior for fluent coordination [17, 19, 48, 49]. Without an accurate model of human behavior, robots misinterpret human intentions and can cause sub-optimal human-robot interaction.

Imitation learning, also known as learning from demonstrations, is a widely-used data-driven paradigm to model human behavior in sequential decision-making tasks [1, 5, 18, 57]. Algorithms for imitation learning (IL) most commonly utilize *observation-action* trajectories of human experts' task executions and return an estimate of true expert behavior (denoted as π_E). Mathematically, the behavior is learned in the form of a conditional distribution $\pi_E \approx \pi(a|s)$, where the variable a denotes task-specific actions and s denotes the *observable* task context. In other words, the traditional IL paradigm assumes expert behavior as being influenced solely by observable and known decision-making factors.

However, as motivated in Figure 1, the reality is more nuanced. Different experts exhibit varying behaviors for the same task. Furthermore, even a single expert's behavior can fluctuate markedly based on their *intents* for identical task contexts. In this sense, intent of a human expert encompasses her preferred approach to executing a task, particularly when it can be accomplished in a variety of ways. Moreover, experts often modify their intent during tasks

---

An extended version of this paper, which includes supplementary material mentioned in the text, is available at http://tiny.cc/idil-appendix

to cater to evolving scenarios. Hence, intent emerges as a *latent decision-making factor* upon which broader, high-level task strategies are formed [9]. We posit that *recognizing and factoring intent and its dynamic nature is crucial for accurate modeling of human behavior*. Given that intent is a cognitive construct and inherently unobservable, conventional imitation learning approaches (which assume behavior depends only on the observable context) fall short in addressing this complexity.

Informed by this research gap, we consider the problem of learning **intent-driven expert behavior**. Mathematically, we seek to estimate $\pi_E \approx \pi(a|s,x)$, where x denotes the intent of the human expert, from a set of heterogeneous expert demonstrations. In addition to utilizing diverse demonstrations without segmentation, learning intentional behavior offers two distinct advantages over traditional imitation learning. First, *intent-driven models can be leveraged to discern the intents of human experts either during or post-task*. Such capability is invaluable for assistive robots and autonomous agents, who can use these inferred intents to anticipate experts' actions over extended periods, enabling improved coordination [17, 26, 45]. Secondly, *intent-driven behavior models $\pi(a|s,x)$ can capture variety of expert behaviors and thus serve as a more comprehensive generative model of expert behavior* relative to intent-unaware models $\pi(a|s)$. Representative generative models of human decision-making behavior are particularly useful in a variety of applications, such as development of digital twins.

Recognizing the diversity observed in expert behaviors, several imitation learning methods have been developed in recent years to learn a behavioral policy that depends on both the task context and latent decision-making factors [24, 28, 32, 43, 50]. However, as reviewed in Sec. 2.4 and demonstrated in Sec. 3, existing intent-aware methods either are designed for discrete domains and, thus, cannot scale up to tasks with large or continuous state spaces [32, 50]; or rely on generative adversarial training, whose performance is often unstable in practice [24, 28, 43].

Towards addressing these limitations, we introduce IDIL: a novel algorithm for imitation learning of intent-driven expert behavior. IDIL learns a generative model of heterogeneous expert behaviors, composed of an intent-aware estimate of the expert policy $\pi_E \approx \pi(a|s,x)$ and a model of experts' intent dynamics $\zeta(x'|s,x)$. To realize stable learning in tasks with large or continuous state spaces, IDIL builds upon recent results from classical imitation learning [12]. Next, beginning with problem formulation, we describe IDIL and derive its convergence properties. This analytical investigation is followed by a suite of experiments, which empirically compare IDIL to recent imitation learning baselines. Our experiments confirm that IDIL is successfully able to estimate expert intent, learn predictive models of expert behavior, and generate diverse interpretable set of (both seen and unseen) expert behaviors.

1.1 Motivating Scenario

To further motivate the problem and describe our solution, we consider a 2-dimensional version of the task shown in Figure 1. In this task, which we call MultiGoals-*n*, where a human is tasked with visiting *n* landmarks[1]; however, the order to visit the landmarks is

---

[1]Such tasks occur commonly in domains such as robotic manufacturing, as a component of more complex sorting, binning, and assembly tasks.

not specified. As such, there are multiple ways to accomplish the task optimally and different individuals may accomplish this task differently. Based on their intent, some experts may start with the red block, while others with the green or blue, and so on. While expert behavior depends on both their starting location and intent, only the former is readily observable. We seek algorithms that explicitly model and learn intent-driven policy of expert behavior from a set of heterogeneous expert demonstrations.

1.2 Preliminaries

To formulate the problem, we limit our scope to sequential tasks that can be described as Markov Decision Processes and expert behaviors that can be represented as an Agent Markov Model.

1.2.1 Markov Decision Processes (MDP).
An MDP represents a sequential task via the tuple $\mathcal{M} \doteq (S, A, T, R, \mu_0, \gamma)$, where S denotes the set of task states s, A denotes the the set of actions, $T : S \times A \times S \rightarrow [0, 1]$ denotes the state transition probabilities, $\mu_0 : S \rightarrow [0, 1]$ denotes the initial state distribution, $R : S \times A \rightarrow \mathbb{R}$ denotes the task reward, and $\gamma \in (0, 1)$ is the discount factor. For our running example, the state denotes the agent location and action denotes the direction in which they intend to move next. The solution to an MDP is denoted as policy, π, which prescribes the agent's next action. The degree of optimality of a policy is quantified through its expected cumulative reward $\bar{G}_\pi \doteq E_\pi[\sum_{t=0}^{\infty} \gamma^t R_t]$. An expert solving an MDP seeks to maximize $\bar{G}_\pi$ by solving for and executing the optimal policy $\pi^* \doteq \arg\max_\pi \bar{G}_\pi$. Due to the Markovian property, there exists at least one optimal policy that is Markovian. Mathematically, this Markovian optimal policy can be summarized via the conditional distribution $\pi^*(a|s) : S \times A \rightarrow [0, 1]$, which denotes the probability of selecting an action a in a given state s.

1.2.2 Imitation Learning.
A Markovian policy $\pi(a|s)$ induces a stationary distribution of states and actions, referred to as the normalized occupancy measure

$$\rho(s, a) \doteq (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p(s^t = s, a^t = a|\pi, \mathcal{M}), \qquad (1)$$

where $p(s^t, a^t|\pi, \mathcal{M})$ is the probability of being in state s and taking action a at time t when following the policy π in the task $\mathcal{M}$ [36]. Conventionally, imitation learning seeks to learn a policy π that matches the behavior of an expert from demonstrations D:

$$\min_\pi L(\pi, \pi_E; D) \qquad (2)$$

where π_E is the expert's policy that is unknown to the imitation learner. Due to the one-to-one correspondence between a policy and its occupancy measure, policy matching problem can be reformulated as an occupancy measure matching problem [13, 18]:

$$\arg\min_\pi D_f \left(\rho_\pi(s, a) \| \rho_{\pi_E}(s, a) \right) \qquad (3)$$

where D_f denotes f-Divergence, and ρ_π and ρ_{π_E} are the occupancy measures induced by the imitation learner and expert respectively.

1.2.3 Agent Markov Model.
While optimal behavior in MDP can be represented simply as a function of the state, in practice, expert behavior – especially that of humans – often hinges on additional decision-making factors. As exemplified by our motivating scenario, several optimal policies might exist in the real world. Human
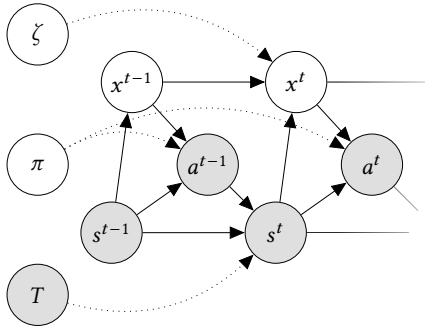
Figure 2: Dynamic Bayesian network representing Intent-Driven Behavior. Shaded nodes denote known or observable variables; other variables are latent.

experts tend to show individual inclinations in choosing among these optimal policies. The Agent Markov Model (AMM) provides a principled approach to represent these variety of behaviors exhibited by agents while solving sequential tasks [50]. An AMM models behavior to depend both on the task state and latent states (e.g., trust, intent, and workload). Mathematically, for a task described as an MDP $\mathcal{M}$, an Agent Markov Model (AMM) defines the behavior of agents as a tuple (X, ζ, b, π), where X is the set of latent states, $\zeta : X \times S \times X \rightarrow [0, 1]$ is the transition probabilities of the latent states, $b : S \times X \rightarrow [0, 1]$ is the initial distribution of the latent states, and $\pi : S \times X \times A \rightarrow [0, 1]$ is the agent policy. Hence, actions of an AMM agent depend on both the task and latent states $a \sim \pi(a|s, x)$. Further, the agent's latent state may change during the task execution based on the corresponding transition probability, $\zeta(x|s, x^-)$.

1.3 Problem Formulation

We now formalize the problem of learning intent-driven expert behavior. Similar to the conventional imitation learning setting, we focus on sequential tasks that can be described as MDPs. However, instead of assuming that expert behavior depends on the task state alone, we model it also depend on their time-varying intent.

1.3.1 Model of Expert Behavior. Mathematically, we model the expert as a special case of AMM in which the latent state (x) corresponds to the expert intent. A dynamic Bayesian network for this representation is depicted in Figure 2. Similar to Seo and Unhelkar [42], we assume X the set of intents is known and finite. Following Jing et al. [24], we introduce the notation # to represent the imaginary value of the intent at time $t = -1$ for the sake of notational brevity. By defining $X^+ \doteq X \cup \{\#\}$ and representing the initial intent distribution as $\zeta(x|s, x^- = \#) \doteq b(x|s)$, we redefine the intent transition probabilities to take the form $\zeta : X^+ \times S \times X \rightarrow [0, 1]$. As ζ now encompasses the initial intent distribution and X is known, we henceforth refer to the AMM model describing the expert as $\mathcal{N}_E$ as the reduced tuple (π_E, ζ_E).

1.3.2 Inputs and Outputs. Although intents influence expert behavior, they cannot be readily sensed or measured by imitation learners. Thus, we assume an expert demonstration does not include the information of intents x. We define a demonstration as

$\tau \doteq (s^{0:h}, a^{0:h})$ and a set of d demonstrations as $D = \{\tau_m\}_{m=1}^d$, where h is a task horizon. In addition, we denote the label of intents for the m-th demonstration as $x_m^{0:h}$. Formally, our problem assumes the following inputs: a task model $\mathcal{M} \setminus R$, a set of intents X, a set of expert demonstrations D, and optionally labels of intents for $l(\leq d)$ demonstrations $\{x_m^{0:h}\}_{m=1}^l$. Given these inputs, our goal is to learn an AMM model $\mathcal{N} = (\pi, \zeta)$ that mimics expert behavior.

1.3.3 Objective. Since the learner only has access to the observable part of the expert behavior $\{s, a\}$, we can consider simply matching the occupancy measure of the observable variables:

$$\arg \min_\pi D_f \left(\rho_\mathcal{N}(s, a) \| \rho_{\mathcal{N}_E}(s, a) \right)$$

However, under this criteria, the learned model could be far from the expert behavior, since multiple AMM models can exhibit the same (s, a)-occupancy measure [24]. To mitigate this ambiguity, hence, we define the normalized intent-aware occupancy measure:

$$\rho_\mathcal{N}(s, a, x, x^-) \doteq (1 - \gamma) \sum_{t=0}^\infty \gamma^t p(s^t=s, a^t=a, x^t=x, x^{t-1}=x^- | \mathcal{N}, \mathcal{M})$$

and pose the problem of learning intent-driven expert behavior as the following occupancy measure matching problem:

$$\min_\mathcal{N} L_{saxx^-}(\mathcal{N}) \doteq \min_\mathcal{N} D_f \left(\rho_\mathcal{N}(s, a, x, x^-) \| \rho_{\mathcal{N}_E}(s, a, x, x^-) \right). \quad (4)$$

2 LEARNING INTENT-DRIVEN BEHAVIOR

To effectively solve the optimization problem of Eq. 4, we introduce IDIL: an imitation learning algorithm that can learn intent-driven expert behavior from heterogeneous demonstrations.

2.1 IDIL: Intent-Driven Imitation Learner

A naïve approach to solve Eq. 4 is to first collect data of expert demonstrations (s, a) and intent (x) and then apply techniques for conventional imitation learning. However, as human cognitive states cannot be readily measured using sensors, collecting intent data is non-trivial and resource-intensive in practice [7, 16, 31]. Moreover, as detailed in Sec. 2.3, even in the case when intent data is available a naïve application of conventional imitation learning is inadvisable. Hence, we derive an iterative factored approach to solve Eq. 4 using the following two occupancy measures:

$$\rho_\mathcal{N}(s, a, x) \doteq (1 - \gamma) \Sigma_{t=0}^\infty \gamma^t p(s^t=s, a^t=a, x^t=x | \mathcal{N}, \mathcal{M}) \quad (5)$$

$$\rho_\mathcal{N}(s, a, x^-) \doteq (1 - \gamma) \Sigma_{t=0}^\infty \gamma^t p(s^t=s, a^t=a, x^{t-1}=x^- | \mathcal{N}, \mathcal{M}) \quad (6)$$

We have the following proposition relating the three occupancy measures, which is proved in the Appendix.

Proposition 2.1. *Consider two arbitrary AMM models: $\mathcal{N}$ and $\mathcal{N}'$. Then, $\rho_\mathcal{N}(s, a, x) = \rho_{\mathcal{N}'}(s, a, x)$ and $\rho_\mathcal{N}(s, x, x^-) = \rho_{\mathcal{N}'}(s, x, x^-)$ if and only if $\rho_\mathcal{N}(s, a, x, x^-) = \rho_{\mathcal{N}'}(s, a, x, x^-)$.*

Proposition 2.1 implies that if the measures $\rho(sax)$ and $\rho(sxx^-)$ induced by the learner match with those induced by the expert, then $\mathcal{N} = \mathcal{N}_E$. In other words, instead of directly optimizing L_{saxx^-}, one can alternatively solve the following factored sub-problems:

$$\min_\mathcal{N} L_{sax}(\mathcal{N}) \doteq \min_\mathcal{N} D_f \left(\rho_\mathcal{N}(s, a, x) \| \rho_{\mathcal{N}_E}(s, a, x) \right) \quad (7)$$

$$\min_\mathcal{N} L_{sxx^-}(\mathcal{N}) \doteq \min_\mathcal{N} D_f \left(\rho_\mathcal{N}(s, x, x^-) \| \rho_{\mathcal{N}_E}(s, x, x^-) \right). \quad (8)$$

Algorithm 1 IDIL: Intent-Driven Imitation Learner

1: **Input**: Data $D = \{\tau_m\}_{m=1}^d$ and $\{x_m^{0:h}\}_{m=1}^l$
2: **Initialize**: (θ, ϕ) where $\mathcal{N}_{\theta,\phi} = (\pi_\theta, \zeta_\phi)$
3: **repeat**
4:    E-step: Infer expert intent $\{x_m^{0:h}\}_{m>l}^d$ using $(\pi_\theta^n, \zeta_\phi^n)$
5:    Define $\tilde{D} \doteq D \cup \{\hat{x}_m^{0:h}\}_{m=1}^d$
6:    Generate trajectories $D_g = \{(s, x, a)^{0:h}\}$ with $(\pi_\theta^n, \zeta_\phi^n)$
7:    M1-step: Compute π_θ^{n+1} via Eq. 7 using $\tilde{D}, D_g$
8:    M2-step: Compute ζ_ϕ^{n+1} via Eq. 8 using $\tilde{D}, D_g$
9: **until** Convergence

Algorithm 1 provides an *iterative* approach to solving these factored sub-problems via online imitation learning. In line 2, IDIL initializes the estimate of expert's model as $\mathcal{N}_E \approx \mathcal{N}_{\theta,\phi} = (\pi_\theta, \zeta_\phi)$, where θ, ϕ represent parameters of a known function (e.g. neural network). We denote the iteration number as n and the n-th estimate as $\mathcal{N}^n = (\pi^n, \zeta^n)$. As the demonstrations generally do not include data of expert intent, line 4, performs intent inference using the current estimate of the expert model. In lines 5-6, the algorithm augments its training data with on-policy experience. The expert demonstrations and on-policy experience are then used to update the estimate of the expert model (π, ζ) (lines 7-8). Due to its iterative nature, IDIL can also be viewed as an expectation-maximization (EM) algorithm in which the E-step corresponds to intent inference and M-step corresponds to model learning. We detail IDIL's subroutines next.

2.1.1 E-step: Intent Inference. Following algorithms for AMM learning [32, 42, 50], we take a Bayesian approach to inferring values of expert intent. Mathematically, to infer intent data $x^{0:h}$ for each demonstration τ, IDIL computes the following maximum a posteriori (MAP) estimate for the posterior distribution:

$$\hat{x}^{0:h} = \arg\max_{x^{0:h}} p(x^{0:h}|\tau, \mathcal{N}^n) \tag{9}$$

using a variant of Viterbi algorithm introduced in [24, 50, 51], which has a time complexity of $O(h|X|^2)$. Observe that, using these estimates, one can augment the expert demonstrations $\tilde{D} \doteq D \cup \{\hat{x}_m^{0:h}\}_{m=1}^d$ to arrive at $(s, a, \hat{x})$-trajectories. Further, these trajectories can then be used to compute the empirical estimates of expert's occupancy measures – $\rho_E^n(s, a, x, x^-)$, $\rho_E^n(s, a, x)$, and $\rho_E^n(s, a, x^-)$ – needed for the subsequent steps of the algorithm.

2.1.2 M1-step: Learning Agent's Policy π. Assuming ρ_E^n and ζ^n as fixed, in line 7, IDIL solves the sub-problem described in Eq. 7 to update the agent's estimate of the expert policy π_θ. To derive the solution, we begin with an intent-augmented state representation, $u \doteq (s, x) \in S \times X$. Using this representation, we define an intent-informed MDP $\tilde{\mathcal{M}} \doteq ((S \times X), A, \tilde{T}, \tilde{\mu}_0, \gamma)$, where $\tilde{T}(u'|u, a) = T(s'|s, a)\zeta^n(x'|x, s)$ and $\tilde{\mu}_0(u) = \zeta(x|s, \#)\mu_0(s)$. Observe that, for this intent-informed MDP, Eq. 7 reduces to the conventional imitation learning problem of learning $\pi(a|u)$ by matching the occupancy measure $\rho(u, a) \equiv \rho(s, x, a)$. Given this reduction, a variety of existing imitation learning algorithms can be used to solve Eq. 7. Our implementation of IDIL uses a recent algorithm IQ-Learn to solve Eq. 7 and update π_θ. This choice is motivated by

IQ-Learn's state-of-the-art performance and stable training; unlike many related techniques, IQ-Learn and consequently IDIL do not rely on (potentially unstable) adversarial training.

2.1.3 M2-step: Learning Agent's Intent Dynamics ζ. Assuming ρ_E^n and π^{n+1} as fixed, in line 8, IDIL updates ζ_ϕ by solving Eq. 8. To derive the solution, we utilize an alternate intent-augmented state representation $v \doteq (s, x^-) \in S \times X^+$. Using this representation and modeling the set of intents X as agent's macro-actions, we define the intent-transition MDP $\bar{\mathcal{M}} = ((S \times X^+), X, \bar{T}, \bar{\mu}_0, \gamma)$, where the transition model of v is defined as $\bar{T}(v'|v, x) = \sum_a \pi^{n+1}(a|s, x)T(s'|s, a)$ and $\bar{\mu}_0(v) = \mu_0(s)$ represents the initial distribution of v. Observe that $\zeta(x|v)$ corresponds to the policy of the intent-transition MDP $\bar{\mathcal{M}}$, which describes the agent's approach to selecting its intent x given the task state and prior intent $v = (s, x^-)$. Hence, for the intent-transition MDP, Eq. 8 too reduces to the conventional imitation learning problem of learning $\zeta(x|v)$ by matching the occupancy measure $\rho(v, x) \equiv \rho(s, x, x^-)$. Consequently, similar to M1-step, a variety of existing imitation learning algorithms can be used to solve Eq. 8. For reasons stated in Sec. 2.1.2, our implementation of IDIL uses IQ-Learn to also solve Eq. 8 and update agent's estimate of the expert's intent dynamics ζ_ϕ.

2.1.4 Implementation Considerations. As suggested by our analysis presented next, we opt for small learning rates while updating the model parameters. This leads to small incremental updates to the parameters θ and ϕ in the M1-step and M2-step. Further, to enhance learning performance, we conduct multiple iterations of M1-step and M2-step (lines 7-8) in every loop of Algorithm 1 after the subsequent E-step (line 4) in our implementation. For detailed insights and other considerations, please refer to the Appendix.

2.2 Sufficient Conditions for Convergence

Proposition 2.1 establishes that $\mathcal{N} = \mathcal{N}_E$ when both $L_{sax} = 0$ and $L_{sxx^-} = 0$. However, due to the iterative nature of Algorithm 1, this alone does not ensure the convergence to the expert model. Minimizing L_{sax} in M1-step might inadvertently increase L_{sxx^-}, and the opposite could occur in M2-step. To address this, we derive the conditions under which Algorithm 1 monotonically reduces the primary loss function (Eq. 4) and, thus, asymptotically converges to a sound estimate of the expert's behavioral model.

Consider the following approximations of the expert occupancy measure $\rho_E(s, a, x, x^-)$, which are computed using the n-th estimate of the expert model and factored occupancy measures:

$$\check{\rho}_E^n(\cdot) = \check{\rho}_E^n(s, a, x, x^-) \doteq \sum_{s^-, a^-} \pi^n(\cdot)\zeta^n(\cdot)T(\cdot)\rho_E^n(s^-, a^-, x^-) \tag{10}$$

$$\hat{\rho}_E^n(\cdot) = \hat{\rho}_E^n(s, a, x, x^-) \doteq \pi^n(a|s, x)\rho_E^n(s, x, x^-) \tag{11}$$

For these approximations, we have the following two lemmas.[2]

Lemma 2.2. *Define $|\Delta(\theta, \theta_n)| \doteq \epsilon$. If ϵ is sufficiently small, then*

$$D_f\left(\rho_{\pi_\theta}(s, a, x)\big\|\rho_E^n(s, a, x)\right) = D_f\left(\rho_{\pi_\theta, \zeta^n}(s, a, x, x^-)\big\|\check{\rho}_E^n(\cdot)\right)$$

Lemma 2.3. *Define $|\Delta(\theta_{n+1}, \theta_n)| \doteq \epsilon$. If ϵ is sufficiently small, then*

$$D_f\left(\rho_{\zeta_\phi}(s, x, x^-)\big\|\rho_E^n(s, x, x^-)\right) = D_f\left(\rho_{\pi^{n+1}, \zeta_\phi}(s, a, x, x^-)\big\|\hat{\rho}_E^n(\cdot)\right)$$

---

[2]Please refer to the Appendix for detailed proofs.

Lemma 2.2 implies that, if the update to policy parameters in M1-step is sufficiently small,[3] then the value of sub-objective L_{sax} approximately equals the overall objective L_{saxx^-}. Here, the approximation arises from the use of n-th estimates of expert model to approximate the expert occupancy measures $\rho_E(s, a, x, x^-) \approx \breve{\rho}_E^n(\cdot)$. Under similar conditions and the approximation $\rho_E(s, a, x, x^-) \approx \dot{\rho}_E^n(\cdot)$, Lemma 2.3 implies that the value of sub-objective L_{sxx^-} equals the overall objective L_{saxx^-}. Thus, under the given conditions, a decrease in L_{sax} and L_{sxx^-} will also decrease L_{saxx^-}. The proof of these lemmas rely on first-order approximations with respect to the policy parameters and convexity of the function used to compute the f-divergence.

Given these lemmas, we can now proceed to stating the sufficient conditions for convergence of IDIL. Consider a third approximation of the expert occupancy measure $\rho_E(s, a, x, x^-)$ computed using the Viterbi intent estimates derived in the E-step as follows:

$$\rho_E^n(\cdot) = \rho_E^n(s, a, x, x^-) \doteq \rho_E(s, a)p(x, x^-|s, a, \mathcal{N}^n) \quad (12)$$

Further, denote the empirical estimate of the original objective of Eq. 4 at the n-th iteration as

$$L_{saxx^-} \approx L_{saxx^-}^n \doteq D_f\left(\rho_{\mathcal{N}}(s, a, x, x^-) \big\| \rho_E^n(s, a, x, x^-)\right) \quad (13)$$

Then, we have the following theorem for the convergence of IDIL.

Theorem 2.4. *If (1) $\rho_E^n \approx \breve{\rho}_E^n \approx \dot{\rho}_E^n$; and (2) $|\Delta(\theta_{n+1}, \theta_n)|$ is sufficiently small, then $L_{saxx^-}^{n+1} \leq L_{saxx^-}^n$.*

Note that, while important, Theorem 2.4 only guarantees convergence to a local optima computed with respect to the approximate objective $\lim_{n\to\infty} L_{saxx^-}^n$. A stronger result can be achieved for the supervised case of the intent-driven imitation learning, in which labels of expert intent available for all expert demonstrations.[4] As evidenced in our experiments, however, the empirical results are more encouraging. IDIL, even with no or small supervision, converges to the expert behavior in a variety of problems.

2.3 IDIL-J: A joint variant of IDIL

Given the inferred values of expert intent, IDIL factors the original problem (Eq. 4) into two sub-problems (Eq. 7 and Eq. 8) which are each solved separately. As the analysis of its convergence relies on showing that optimizing the two factored objectives is equivalent to optimizing the original joint objective L_{saxx^-}, a natural question arises: *why not directly optimize the joint objective?* To answer this question, we provide an alternate approach that indeed uses the inferred intent values to directly optimize the original joint objective L_{saxx^-}. To derive this approach, consider the joint MDP model, with state defined as $w \doteq (s, x^-) \in S \times X^+$ and actions as $a_w \doteq (a, x) \in A \times X$. The policy for this joint MDP corresponds to a joint estimate of the expert policy and intent dynamics

$$\tilde{\pi}(a_w|w) \doteq \tilde{\pi}(a, x|s, x^-) = \pi(a|s, x)\zeta(x|s, x^-) \quad (14)$$

Further, given the inferred values of (x, x^-) computed via the E-step and the aforementioned MDP, Eq. 4 reduces to the conventional imitation learning problem of learning $\tilde{\pi}(a_w|w)$ by matching the occupancy measure $\rho(w, a_w) \equiv \rho(s, a, x, x^-)$. Given this reduction,

similar to the factored M1-step and M2-step, we can now apply existing techniques to solve the original problem. For instance, we can apply IQ-Learn to first learn $Q(w, a_w) = Q(s, x^-, a, x)$ from the expert demonstrations and online experience. Next, using the learnt Q-values we can then recover the expert model (π and ζ) by optimizing the following soft actor-critic (SAC) objective [14]:

$$J(\theta, \phi) = E_{(s,x^-)\sim\tilde{D}}\left[D_{KL}\left(\pi_\theta \cdot \zeta_\phi \middle\| \frac{\exp(Q(s, x^-, a, x))}{Z}\right)\right] \quad (15)$$

We refer to this unfactored approach, which replaces the two M1-step and M2-step with a joint M-step, as IDIL-Joint or IDIL-J in short. This approach is technically sound. However, as learning the joint function does not leverage the inherent factored structure of the expert behavior, we anticipate its learning performance and interpretability to be subpar than IDIL. In Sec. 3, we evaluate this hypothesis by empirically comparing IDIL and IDIL-J.

2.4 Relation to Existing Works

We conclude this section by discussing the relationship of IDIL to relevant human modeling and imitation learning techniques that also consider latent decision-making factors [2, 5, 17, 34].

2.4.1 Human Behavior Modeling.
A prevailing motif in human modeling research is the significance of mental (or latent) states, such as intent, expertise, and trust in the robot [33, 37, 38, 45, 54]. Techniques that incorporate these latent states generally outperform those that exclude them [6, 11, 15, 20, 29, 40, 52, 53]. For instance, explicit modeling of human intent has been used to learn more effective policies for human-robot collaboration [53]. Similarly, extensions of Generative Adversarial Imitation Learning (GAIL) that incorporate demonstrator's strategy code as an latent feature tend to outperform GAIL [15, 18, 29, 52]. While these methods learn human behavior with diverse intents, unlike IDIL, they assume that the intent remains static during an episode.

In contrast, Agent Markov Model (AMM) and its variants consider both the expert's latent states and its temporal dynamics [32, 42, 50]. Thus, the model can capture diverse and dynamically-changing intents representative of human behavior. However, existing learning methods for AMM utilize variational inference and tabular representations, which imposes strong assumptions on the target distributions and cannot scale up to complex tasks. IDIL does not make such assumptions and thus is capable of learning expert behavior in tasks with large or continuous state spaces.

2.4.2 Hierarchical Imitation Learning.
As intents typically shape expert behavior at a macro level, the problem of learning intent-driven behavior shares similarities with hierarchical imitation learning [8, 21, 25, 27, 56].[5] Kipf et al. [25], Le et al. [27], Zhang and Paschalidis [56] propose hierarchical policy learning algorithms based on behavior cloning. These methods, however, do not utilize task dynamics and are prone to compounding errors [39, 47]. To alleviate the problem of compounding errors, several GAIL-based approaches have been proposed to learn hierarchical policies [10, 24, 28, 43]. Among these, Option-GAIL that builds upon the options framework and utilizes an EM approach is most relevant to

---

[3]In practical implementations, sufficiently small updates to the policy parameters can be realized through appropriate choice of the learning rate hyperparameter.
[4]For the supervised case, the E-step of Algorithm 1 is not required, and offline pre-training can greatly help.

[5]For instance, in our problem setting, intent can be viewed as a one-step option [44]. However, in contrast to the options framework, intent is neither associated with an option termination condition nor assumes temporal abstraction.
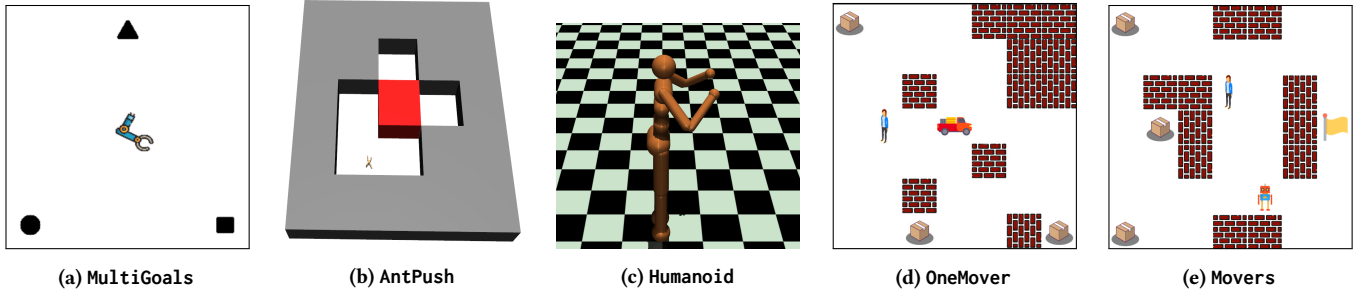
(a) `MultiGoals` **(b)** `AntPush` **(c)** `Humanoid` **(d)** `OneMover` **(e)** `Movers`

Figure 3: Visual illustrations of Experimental Domains.

Table 1: Key Characteristics of Experimental Domains

	State Space		**Expert**	
Domain	Type	$\lvert S \rvert$ or n_s	Intent	Human
`MG-n suite`	Continuous	2	Yes	
`AntPush`	Continuous	107	Maybe	
`Mujoco suite`	Continuous	376		
`OneMover`	Discrete	$\approx 1k$	Yes	
`Movers`	Discrete	$\approx 40k$	Yes	Yes

our work [24, 44, 55]. However, unlike our approach, Option-GAIL relies on adversarial training and thus is prone to unstable gradients [3, 23]. We posit, through the use of a non-adversarial approach, IDIL will result in stable training and superior performance [12]. We evaluate this hypothesis in the following section.

3 EXPERIMENTS

We evaluate IDIL on multiple domains and benchmark its ability to successfully complete tasks, infer expert intent, and generate diverse interpretable behaviors. Through this empirical investigation, we evaluate, in terms of performance,

- (Q1) How does IDIL compare to conventional IL techniques when experts exhibit intent-driven behavior?
- (Q2) How does IDIL compare to conventional IL techniques when experts *do not* exhibit intent-driven behavior?
- (Q3) How does IDIL compare to hierarchical IL techniques?
- (Q4) How does IDIL compare to its variant IDIL-J?

3.1 Domains and Expert Demonstrations

As summarized in Table 1, we consider both discrete and continuous MDPs, low- and high-dimensional state spaces, intent-driven and intent-agnostic behaviors, and human and artificial experts. In Table 1, we list the size of the state space $\lvert S \rvert$ for discrete domains and number of state features n_s for continuous domains. For the `MultiGoals`-n and `Mujoco` suites, we report these values for the most complex task of the suite.

3.1.1 `MultiGoals`*-n (*`MG`*-n).* As introduced in Sec. 1.1, this domain models an agent tasked with visiting n landmarks (goals) in an empty environment of size 5×5. Figure 3a depicts this domain with $n = 3$. No constraint is placed on the visitation order of landmarks,

which is decided based on expert's intent. Both state and action are 2-dimensional vectors, denoting the position and velocity of the agent. The agent receives a reward of 10 upon reaching a previously unvisited landmark and a penalty of 0.1 every time step. We synthetically generate 100 demonstrations (50/50 train/test split) for this task by hand-crafting an expert model. The hand-crafted expert first selects an arbitrary unvisited landmark and then performs goal-oriented motion to visit it. Observe that the state alone is insufficient to predict expert behavior, as it depends on both its location and intent (i.e., which landmark the expert plans to visit next). We consider four versions of this task, by varying n from two to five.

3.1.2 `OneMover`*.* This domain, depicted in Figure 3d, models a single-agent version of the benchmark domain `Movers` introduced by Seo et al. [41]. It simulates a 7×7 discrete grid world, in which the agent is task with moving boxes to the truck. The state denotes task information (agent and box locations) but not the agent intent (i.e., which box it plans to pick or drop next). We assume an expert may have one of four different intents: going to one of the three boxes or a truck). Similar to `MultiGoals`-n, we synthetically generate 100 demonstrations (50/50 train/test split) for this task by hand-crafting an expert model.

3.1.3 `Mujoco` *suite.* To evaluate IDIL in domains that may not involve diverse or time-varying intents, we consider the following benchmark Mujoco environments: `Hopper-v2`, `HalfCheetah-v2`, `Walker2d-v2`, `Ant-v2`, `Humanoid-v2`, and `AntPush-v0` [46]. We use the version of `AntPush-v0` introduced by Jing et al. [24] and their expert dataset to facilitate a direct comparison between the performance of IDIL and Option-GAIL. For the remaining tasks, we employ the dataset made available by Garg et al. [12]. We use demonstrations of 1k, 5k, and 50k time steps for `Hopper-v2`, `Walker2d-v2`, and `AntPush-v0`, respectively, the same as Jing et al. [24]. For other tasks, we use five trajectories (5k time steps). For all tasks, we set the number of intents to four following Jing et al. [24].

3.1.4 `Movers`*.* Modeling human behavior and recognizing their intent are particularly important for human-AI collaboration. Hence, we also utilize `Movers`: a human-AI collaboration setting introduced by Seo and Unhelkar [42]. This domain models a human-agent team that needs to work together to move boxes to a truck. Originally a multi-agent problem, we adapt it to a single-agent context by

Table 2: Cumulative task reward (averaged over three experiment trials) for each method.

Domain	Expert	BC	IQL	OGAIL	IDIL-J	IDIL	OGAIL-s	IDIL-J-s	IDIL-s
MG-2	15.19 ± 1.0	-8.69 ± 2.3	-2.61 ± 8.4	-10.51 ± 2.4	1.08 ± 6.7	**16.16 ± 0.3**	-9.10 ± 5.2	1.55 ± 1.2	**16.20 ± 0.2**
MG-3	22.32 ± 0.9	-7.08 ± 1.5	6.06 ± 14.8	-11.25 ± 2.2	-4.58 ± 1.4	**20.64 ± 2.2**	-4.89 ± 5.9	-3.33 ± 0.7	**22.25 ± 0.8**
MG-4	29.14 ± 1.4	-10.4 ± 0.7	-8.67 ± 2.3	-13.33 ± 3.6	-4.17 ± 1.4	**13.63 ± 2.1**	-13.75 ± 3.3	-4.17 ± 2.6	**24.61 ± 1.7**
MG-5	40.09 ± 1.4	-6.25 ± 1.3	-10.92 ± 1.8	-12.92 ± 2.6	4.17 ± 4.4	**27.88 ± 1.8**	-12.08 ± 5.1	6.27 ± 2.5	**32.84 ± 0.7**
OneMover	-46.8 ± 4	-200.0 ± 0	-45.7 ± 3	-200.0 ± 0	-91.3 ± 17	**-45.5 ± 1**	-200.0 ± 0	-70.6 ± 32	**-46.0 ± 2**
Movers	-67.6 ± 6	-200.0 ± 0	**-65.6 ± 0**	-200.0 ± 0	-152.3 ± 16	-68.3 ± 3	-200.0 ± 0	-155.2 ± 28	**-68.2 ± 1**
Hopper	3533 ± 39	861 ± 558	3523 ± 22	3279 ± 426	**3558 ± 40**	3523 ± 21			
½Cheetah	5098 ± 62	4320 ± 314	5092 ± 31	2801 ± 923	**5136 ± 37**	5098 ± 66			
Walker2d	5274 ± 53	3065 ± 800	5185 ± 43	4178 ± 1610	5179 ± 31	**5249 ± 19**		N/A	
Ant	4700 ± 80	4179 ± 175	4606 ± 38	4529 ± 59	**4636 ± 27**	4580 ± 30			
Humanoid	5313 ± 210	496 ± 47	5350 ± 58	458 ± 43	5298 ± 22	**5370 ± 133**			
AntPush	116.6 ± 14	90.2 ± 13	116.6 ± 1	98.9 ± 16	116.6 ± 2	**117.1 ± 0**			

treating the robot teammate as an environmental component. We utilize 66 human demonstrations, 44 for training and 22 for testing.

3.2 Baselines and Metrics

3.2.1 Baselines. We benchmark IDIL and its variant IDIL-J against Behavioral Cloning (BC), IQ-Learn (IQL), and Option-GAIL (OGAIL) [12, 24, 35]. BC is a simple yet effective IL technique [30]. IQ-Learn and Option-GAIL represent the state of the art in conventional and hierarchical IL, respectively. For the intent-aware techniques, we also benchmark the performance of their semi-supervised variants (denoted as IDIL-s, IDIL-J-s, OGAIL-s), which additionally receive ground truth labels of expert intent for 20% of the trajectories.

3.2.2 Implementation Details. We utilize a multi-layer perceptron (MLP) with two hidden layers, each containing 128 nodes, to implement all neural networks of IDIL; this includes Q-networks for ζ and π, and the actor network for π in the case of continuous domains. Other hyperparameters are selected via grid search. The policy network for BC employs an MLP with two hidden layers of 256 nodes each. For implementing IQ-Learn and Option-GAIL, we adopt the neural network structures and hyperparameters proposed in their original papers [12, 24]. IDIL-J shares the same architecture as IQ-Learn. All online imitation learning methods were trained until convergence, capped at 6M exploration steps. Behavior Cloning was trained for 10k updates, using a batch size of 256. Please refer to the Appendix for additional implementation details.

3.2.3 Metrics. We consider three metrics: the cumulative task reward, accuracy of intent inference, and the diversity of behaviors generated by the learned model. To characterize the learning curve, cumulative reward is computed every 20k exploration steps as an average over 8 evaluation episodes. Table 2 reports the maximum of this learning curve. We determine the accuracy of intent inference by computing the fraction of time steps wherein the algorithm accurately identifies the intent, relative to the total steps in the test set. The range and interpretability of behaviors exhibited by the learned model is qualitatively assessed through a visual inspection of its generated trajectories.

3.3 Results

3.3.1 Task Performance: Intent-Driven Experts. First, we consider the performance of IDIL in domains where experts exhibit diverse intent-driven behaviors: namely, MultiGoals, OneMover, and Movers. As shown in Table 2, *IDIL either outperforms or performs comparably to the baselines* in these domains. Further, in domains where the performance is comparable to baselines, both IDIL and the best performing baseline perform as well as the expert.

Towards Q1, we observe that IQ-Learn (a conventional IL technique) performs as well as the expert in some intent-driven tasks but not in others. Its task performance is especially poor in MultiGoals domain, where expert behaviors are not only diverse but also contradictory. For instance, based on their intent, different experts may select opposite actions at the same state. As IQ-Learn does not model intent it is unable to resolve this ambiguity and thus leads to subpar performance. In contrast, IDIL is still able to mimic expert behavior and achieves a high task reward, associating contradictory behaviors to different intents. Moreover, even in domains where IQ-Learn exhibits expert-level performance, the learnt model cannot be used to infer the actor's intent or generate diverse behaviors.

Among intent-aware algorithms considered in Q3 and Q4, IDIL outperforms Option-GAIL and IDIL-J. Option-GAIL particularly performs poorly in OneMover and Movers. We expect that this occurs due to the large input dimension for its discriminator, caused by the conversion of a discrete state into a corresponding one-hot encoding. Such high-dimensional inputs hinder the performance of the discriminator, which is crucial for GAIL-like algorithms to work, by making it challenging for a neural network to differentiate between them. IDIL-J also shows poor performance in discrete domains, presumably due to Gumbel-Softmax reparameterization [22] of Categorical distribution required for backpropagation of gradients when optimizing the SAC objective. As mentioned in Sec. 2.1.4, since IDIL learns a separate $Q(s, a, x)$-function, the policy for discrete action can be directly derived from the learnt Q, removing the additional optimization for the SAC objective. We also observe that semi-supervision of intents leads to faster convergence and improves the performance of IDIL (cf. Table 2: IDIL and IDIL-s).

Table 3: Accuracy of Intent Inference

Domain	Random	OGAIL-s	IDIL-J-s	IDIL-s
MG-3	≈ 0.33	0.44 ± 0.1	0.64 ± 0.1	$\mathbf{0.93 \pm 0.0}$
MG-5	≈ 0.20	0.20 ± 0.0	0.53 ± 0.2	$\mathbf{0.83 \pm 0.1}$
OneMover	≈ 0.25	0.38 ± 0.1	0.32 ± 0.2	$\mathbf{0.62 \pm 0.0}$
Movers	≈ 0.20	0.38 ± 0.3	0.48 ± 0.1	$\mathbf{0.60 \pm 0.1}$

While Option-GAIL and IDIL-J also benefit from this additional information, the improvement is the most noticeable for IDIL.

3.3.2 Task Performance: Intent-Agnostic Experts. For the domains where demonstrations do not include diverse or intent-driven expert behaviors, IQ-Learn, IDIL-J, and IDIL all show comparable performance and achieve the near-expert task reward. Towards Q2, this demonstrates that IDIL does not overfit to the intent-driven model and does not collapse when demonstrations are homogeneous. Also, IDIL's performance in the `Mujoco` suite ensures that it can handle expert's demonstrations in complex and continuous domains, which is not possible with recent AMM-learning algorithms [32, 42, 50]. In addition, compared to Option-GAIL and IDIL-J, IDIL achieves a higher task reward and show lower variance. Taking into account that IDIL-J and Option-GAIL utilize identical objectives Eq. 4, this results demonstrate the advantage of non-generative adversarial approach for achieving consistent training.

3.3.3 Intent Inference. A high reward demonstrates that the algorithm is capable of finding one (near-)optimal behavior. However, it does not imply that the algorithm is adept in capturing intents or diversity observed in expert behavior. Hence, we benchmark the intent inference performance computed using the learnt models in Table 3. To associate the expert and learnt intent, we train models with 20% supervision of intent. Notice that, since IQ-Learn and Behavior Cloning learn a single behavior, models learnt using them cannot be used for intent inference. We observe that IDIL outperforms Option-GAIL, IDIL-J, and random guess in all domains.

3.3.4 Diversity and Interpretability of Generated Behaviors. Finally, in Figure 4, we visualize and qualitatively inspect the diverse intent-driven behaviors learnt by various algorithms for the `MultiGoals`-3 domain. Recall that in this domain each intent-driven behavior corresponds to reaching one of the three landmarks in the environment. We observe that behaviors learnt using IDIL are not only capable of addressing complex tasks (cf. Table 2) but also interpretable and match diverse expert intents. This qualitative analysis also sheds light on superior intent inference performance of IDIL (cf. Table 3). In contrast, IDIL-J does not effectively learn interpretable intent-driven behavior. As we posited in Sec. 2.3, our experiments suggest learning the joint Q-function instead of learning two factored Q-functions decreases both interpretability and overall performance. Figure 4 also suggests that Option-GAIL is not able to learn intent-driven behavior that match expert intent.

4 CONCLUSION

We introduce IDIL, a novel algorithm for learning intent-driven expert behavior from heterogeneous demonstrations. By employing
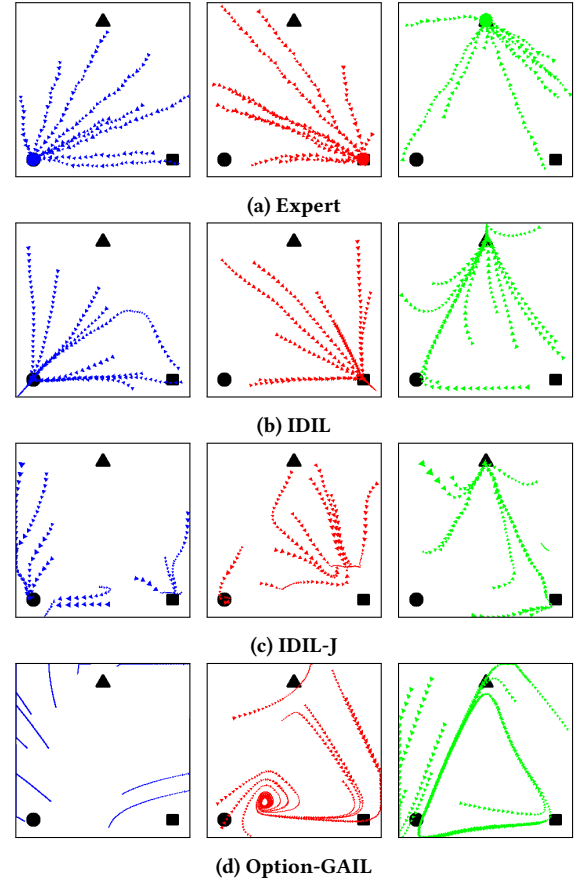

(a) Expert

(b) IDIL

(c) IDIL-J

(d) Option-GAIL

Figure 4: `MultiGoals`-3 trajectories generated by the expert and learnt models according to different intents.

an EM-style approach and implementing factored optimizations, IDIL successfully captures time-varying intents and intent-driven behavior even in complex and continuous tasks. Our contribution includes a theoretical analysis on the convergence properties of IDIL and empirical results showcasing that IDIL outperforms the state-of-the-art baselines.

Our work also motivates several directions of future work. The key limitation of our work is the set of intent has to be finite and known. Moreover, when dealing with a large number of intents, the E-step can become slow. A worthwhile future direction can be exploring methods to relax these assumptions regarding intents. Additionally, while intents tend to change slower compared to actions, IDIL does not explicitly leverage this characteristic. In future work, we hope to develop methods to incorporate this prior knowledge of intent dynamics into the learning process. Building on IDIL, we also plan to extend this work to multi-agent settings where intent of each agent can be influenced by other agents.

ACKNOWLEDGMENTS

REFERENCES

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

[2] Stefano V Albrecht and Peter Stone. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence* 258 (2018), 66–95.

[3] Oleg Arenz and Gerhard Neumann. 2020. Non-adversarial imitation learning and its connections to adversarial methods. *arXiv preprint arXiv:2008.03525* (2020).

[4] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57, 5 (2009), 469–483.

[5] Saurabh Arora and Prashant Doshi. 2021. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence* 297 (2021), 103500.

[6] Mark Beliaev, Andy Shih, Stefano Ermon, Dorsa Sadigh, and Ramtin Pedarsani. 2022. Imitation learning by estimating expertise of demonstrators. In *International Conference on Machine Learning.* PMLR, 1732–1748.

[7] Cindy L Bethel, Kristen Salomon, Robin R Murphy, and Jennifer L Burke. 2007. Survey of psychophysiology measurements applied to human-robot interaction. In *RO-MAN 2007-The 16th IEEE International Symposium on Robot and Human Interactive Communication.* IEEE, 732–737.

[8] Kenneth Bogert and Prashant Doshi. 2022. A hierarchical bayesian process for inverse rl in partially-controlled environments. In *AAMAS Conference proceedings.*

[9] Richard W Byrne and Anne E Russon. 1998. Learning by imitation: A hierarchical approach. *Behavioral and brain sciences* 21, 5 (1998), 667–684.

[10] Jiayu Chen, Tian Lan, and Vaneet Aggarwal. 2023. Option-aware adversarial inverse reinforcement learning for robotic control. In *2023 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 5902–5908.

[11] Jaedeug Choi and Kee-Eung Kim. 2012. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. *Advances in neural information processing systems* 25 (2012).

[12] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. 2021. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems* 34 (2021), 4028–4039.

[13] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. 2020. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning.* PMLR, 1259–1277.

[14] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning.* PMLR, 1861–1870.

[15] Karol Hausman, Yevgen Chebotar, Stefan Schaal, Gaurav Sukhatme, and Joseph J Lim. 2017. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. *Advances in neural information processing systems* 30 (2017).

[16] Jamison Heard, Caroline E Harriott, and Julie A Adams. 2018. A survey of workload assessment algorithms. *IEEE Transactions on Human-Machine Systems* 48, 5 (2018), 434–451.

[17] Laura M Hiatt, Cody Narber, Esube Bekele, Sangeet S Khemlani, and J Gregory Trafton. 2017. Human modeling for human–robot collaboration. *The International Journal of Robotics Research* 36, 5-7 (2017), 580–596.

[18] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016).

[19] Guy Hoffman. 2019. Evaluating fluency in human–robot collaboration. *IEEE Transactions on Human-Machine Systems* 49, 3 (2019), 209–218.

[20] Boris Ivanovic, Edward Schmerling, Karen Leung, and Marco Pavone. 2018. Generative modeling of multimodal multi-human behavior. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 3088–3095.

[21] Abhinav Jain and Vaibhav Unhelkar. 2024. GO-DICE: Goal-conditioned Option-aware Offline Imitation Learning via Stationary Distribution Correction Estimation. In *Proceedings of the AAAI conference on artificial intelligence.*

[22] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations.*

[23] Haoming Jiang, Zhehui Chen, Minshuo Chen, Feng Liu, Dingding Wang, and Tuo Zhao. 2018. On computation and generalization of generative adversarial networks under spectrum control. In *International Conference on Learning Representations.*

[24] Mingxuan Jing, Wenbing Huang, Fuchun Sun, Xiaojian Ma, Tao Kong, Chuang Gan, and Lei Li. 2021. Adversarial option-aware hierarchical imitation learning. In *International Conference on Machine Learning.* PMLR, 5097–5106.

[25] Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. 2019. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning.* PMLR, 3418–3428.

[26] Przemyslaw A Lasota, Terrence Fong, Julie A Shah, et al. 2017. A survey of methods for safe human-robot interaction. *Foundations and Trends® in Robotics*

[27] Hoang Le, Nan Jiang, Alekh Agarwal, Miroslav Dudík, Yisong Yue, and Hal Daumé III. 2018. Hierarchical imitation and reinforcement learning. In *International conference on machine learning.* PMLR, 2917–2926.

[28] Sang-Hyun Lee and Seung-Woo Seo. 2020. Learning compound tasks without task-specific knowledge via imitation and self-supervised learning. In *International Conference on Machine Learning.* PMLR, 5747–5756.

[29] Yunzhu Li, Jiaming Song, and Stefano Ermon. 2017. Infogail: Interpretable imitation learning from visual demonstrations. *Advances in neural information processing systems* 30 (2017).

[30] Ziniu Li, Tian Xu, Yang Yu, and Zhi-Quan Luo. 2022. Rethinking ValueDice - Does It Really Improve Performance?. In *ICLR Blog Track.* https://iclr-blog-track.github.io/2022/03/25/rethinking-valuedice/

[31] Catherine Neubauer, Kristin E Schaefer, Ashley H Oiknine, Steven Thurman, Benjamin Files, Stephen Gordon, J Cortney Bradford, Derek Spangler, and Gregory Gremillion. 2020. *Multimodal Physiological and Behavioral Measures to Estimate Human States and Decisions for Improved Human Autonomy Teaming.* Technical Report. CCDC Army Research Laboratory Aberdeen Proving Ground United States.

[32] Liubove Orlov-Savko, Abhinav Jain, Gregory M Gremillion, Catherine E Neubauer, Jonroy D Canady, and Vaibhav Unhelkar. 2022. Factorial Agent Markov Model: Modeling Other Agents' Behavior in presence of Dynamic Latent Decision Factors. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS).* IFAAMAS.

[33] Liubove Orlov-Savko, Zhiqin Qian, Gregory M Gremillion, Catherine E Neubauer, Jonroy Canady, and Vaibhav Unhelkar. 2024. RW4T Dataset: Data of Human-Robot Behavior and Cognitive States in Simulated Disaster Response Tasks. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI).*

[34] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. 2018. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics* 7, 1-2 (2018), 1–179.

[35] Dean A Pomerleau. 1988. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems* 1 (1988).

[36] Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons.

[37] Peizhu Qian and Vaibhav V Unhelkar. 2022. Evaluating the Role of Interactivity on Improving Transparency in Autonomous Agents. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS).* IFAAMAS.

[38] Zhiqin Qian, Liubove Orlov-Savko, Gregory M Gremillion, Catherine E Neubauer, and Vaibhav Unhelkar. 2024. Measuring Variations in Workload during Human-Robot Collaboration through Automated After-Action Reviews. In *Companion of the ACM/IEEE International Conference on Human-Robot Interaction (HRI).*

[39] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics.* JMLR Workshop and Conference Proceedings, 627–635.

[40] Edward Schmerling, Karen Leung, Wolf Vollprecht, and Marco Pavone. 2018. Multimodal probabilistic model-based planning for human-robot interaction. In *2018 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 3399–3406.

[41] Sangwon Seo, Bing Han, and Vaibhav Unhelkar. 2023. Automated Task-Time Interventions to Improve Teamwork using Imitation Learning. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS).* IFAAMAS.

[42] Sangwon Seo and Vaibhav Unhelkar. 2022. Semi-Supervised Imitation Learning of Team Policies from Suboptimal Demonstrations. In *International Joint Conference on Artificial Intelligence (IJCAI).*

[43] Mohit Sharma, Arjun Sharma, Nicholas Rhinehart, and Kris M Kitani. 2018. Directed-Info GAIL: Learning Hierarchical Policies from Unsegmented Demonstrations using Directed Information. In *International Conference on Learning Representations.*

[44] Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112, 1-2 (1999), 181–211.

[45] Andrea Thomaz, Guy Hoffman, Maya Cakmak, et al. 2016. Computational human-robot interaction. *Foundations and Trends® in Robotics* 4, 2-3 (2016), 105–223.

[46] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems.* IEEE, 5026–5033.

[47] Stephen Tu, Alexander Robey, Tingnan Zhang, and Nikolai Matni. 2022. On the sample complexity of stability constrained imitation learning. In *Learning for Dynamics and Control Conference.* PMLR, 180–191.

[48] Vaibhav V Unhelkar*, Shen Li*, and Julie A Shah. 2020. Decision-Making for Bidirectional Communication in Sequential Human-Robot Collaborative Tasks. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI).*

[49] Vaibhav V Unhelkar, Shen Li, and Julie A Shah. 2020. Semi-supervised learning of decision-making models for human-robot collaboration. In *Conference on Robot Learning.* PMLR, 192–203.

5, 4 (2017), 261–349.

[50] Vaibhav V Unhelkar and Julie A Shah. 2019. Learning models of sequential decision-making with partial specification of agent behavior. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 2522–2530.

[51] Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory* 13, 2 (1967), 260–269.

[52] Chen Wang, Claudia Pérez-D'Arpino, Danfei Xu, Li Fei-Fei, Karen Liu, and Silvio Savarese. 2022. Co-gail: Learning diverse strategies for human-robot collaboration. In *Conference on Robot Learning*. PMLR, 1279–1290.

[53] Annie Xie, Dylan Losey, Ryan Tolsma, Chelsea Finn, and Dorsa Sadigh. 2021. Learning latent representations to influence multi-agent interaction. In *Conference on robot learning*. PMLR, 575–588.

[54] X. Jessie Yang, Vaibhav V. Unhelkar, Kevin Li, and Julie A. Shah. 2017. Evaluating Effects of User Experience and System Transparency on Trust in Automation. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (Vienna, Austria). 408–416.

[55] Shangtong Zhang and Shimon Whiteson. 2019. DAC: The double actor-critic architecture for learning options. *Advances in Neural Information Processing Systems* 32 (2019).

[56] Zhiyu Zhang and Ioannis Paschalidis. 2021. Provable hierarchical imitation learning via em. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 883–891.

[57] Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd national conference on Artificial intelligence-Volume 3*. 1433–1438.