Semi-Supervised Imitation Learning of Team Policies from Suboptimal Demonstrations

Sangwon Seo and **Vaibhav V. Unhelkar**
Rice University
{sangwon.seo, vaibhav.unhelkar}@rice.edu

Abstract

We present Bayesian Team Imitation Learner (BTIL), an imitation learning algorithm to model the behavior of teams performing sequential tasks in Markovian domains. In contrast to existing multi-agent imitation learning techniques, BTIL explicitly models and infers the time-varying mental states of team members, thereby enabling learning of decentralized team policies from demonstrations of suboptimal teamwork. Further, to allow for sample- and label-efficient policy learning from small datasets, BTIL employs a Bayesian perspective and is capable of learning from semi-supervised demonstrations. We demonstrate and benchmark the performance of BTIL on synthetic multi-agent tasks as well as a novel dataset of human-agent teamwork. Our experiments show that BTIL can successfully learn team policies from demonstrations despite the influence of team members' (time-varying and potentially misaligned) mental states on their behavior.

1 Introduction

Teamwork is essential for the success of human enterprise. As artificial agents increasingly become parts of human life, thus, they too are expected to reason about and contribute to human teams. At the same time, teamwork is highly challenging to perfect. Successful human teams employ a variety of training techniques to improve coordination and teamwork [Tannenbaum and Salas, 2020]. Analogously, spurred by the need to enable and enhance human-agent collaboration, there has been growing work on developing computational techniques for training artificial agents to support human teams [Thomaz *et al.*, 2016]. These techniques build upon a variety of AI paradigms, such as planning under uncertainty, reinforcement learning, and imitation learning.

In this work, we consider the paradigm of imitation learning [Argall *et al.*, 2009; Osa *et al.*, 2018], wherein an agent learns from demonstrations and (in contrast to reinforcement learning) can thereby learn policies for teamwork without the need of unsafe exploration. By providing novel multi-agent imitation learning techniques that are inspired by real-world teaming considerations, this work aims to enable agents to model, assess, and improve both human-human and human-AI teamwork in sequential tasks. Mathematically, imitation learning techniques seek to learn a single-agent behavioral policy (π), a stochastic function that encodes probability of selecting an action (a) in a task-specific context (s), given a dataset of (s, a)-tuples provided by a demonstrator. Typically, the context features (s) and actions (a) are assumed to be fully observable and measured using sensors.

Imitation learning has been extended to model multi-agent systems by seeking to learn a set of behavioral policies $\{\pi_i | i{=}1{:}n\}$, one corresponding to each member $(i{=}1{:}n)$ of the multi-agent system from demonstrations of teamwork [Le *et al.*, 2017; Bhattacharyya *et al.*, 2019; Song *et al.*, 2018; Lin *et al.*, 2019]. We provide a brief survey of related multi-agent imitation learning (MAIL) techniques in Appendix A. MAIL is an emerging area of research, wherein the existing works either focus on learning policies corresponding to game-theoretic equilibria of multi-agent systems, assume homogeneity in agents' capabilities, or assume data of optimal teaming behavior as the training input. However, in contrast to most settings considered in prior art, teamwork observed in practice [Salas *et al.*, 2018; Seo *et al.*, 2021] often differs in three key ways: (1) it may not correspond to a game-theoretic equilibrium, (2) it can be suboptimal due to its dependence on latent performance-shaping factors (such as team members' mental models), and (3) it can involve team members with different capabilities.

Informed by these three considerations of teamwork observed in the real world, in Sec. 3.3, we provide an alternate problem formulation of MAIL for collaborative tasks. In particular, we consider that the team members' behavior (π_i) depends not only on the context features (s) but also on their (time-varying) mental models (x_i) pertaining to teamwork. As the mental models cannot be readily sensed and manually annotating them is resource-intensive, they are modeled as partially observable to the imitation learner. Further, in teaming scenarios where the mental models are misaligned (i.e., the team members do not maintain a shared understanding), the demonstrations available for learning can be suboptimal.

An extended version of this paper, which includes supplementary material (appendices and video supplements) mentioned in the text, is available at https://arxiv.org/abs/2205.02959

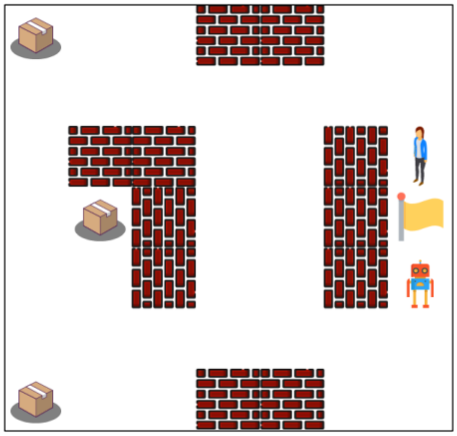

Figure 1: *Movers* domain. The human-robot team needs to move the boxes to the flag. Each box can be moved if and only if both agents pick it up and move it along the same direction.

Thus, adding to the challenge, in our problem formulation the imitation learner also needs to infer which segments of the teaming demonstrations are (sub)-optimal and perform learning under partial observability of the dynamic state (s, x).

Summary of Contributions. Towards this problem setting, we present Bayesian Team Imitation Learner (BTIL, pronounced as "bee-tul"), an imitation learning algorithm that can learn decentralized team policies from both optimal and sub-optimal demonstrations. To effectively learn the team policies, BTIL explicitly models the time-varying mental states of each team member (x_i) and jointly learns their transition model (T_x). To enable sample- and label-efficient policy learning, BTIL utilizes a Bayesian perspective and is capable of learning from partial supervision of the mental states. We benchmark our solutions against two existing techniques [Pomerleau, 1991; Song *et al.*, 2018]. In our evaluations, we emphasize the challenge of collecting teaming demonstrations by collecting a novel dataset of human-agent teamwork for settings where (a) the labels of mental models are only partially available, and (b) the number of demonstrations is small relative to the size of the task's state space. Our experiments show that BTIL can learn team policies from small semi-supervised datasets of suboptimal teamwork and outperform the baselines across relevant metrics.

Running Example. To help describe our approach, we utilize the collaboration scenario shown in Fig. 1 as a running example. In this scenario, a two-member team composed of Alice and Rob is tasked with moving all boxes together to the flag in the least amount of time. At each time step, each agent can choose to move in any one of four cardinal directions, attempt to pick up or drop a box, or perform no operation. As each box is heavy, it cannot be lifted by one agent alone. To effectively move all boxes to the goal location, the team needs to coordinate which box to pick next as well as the path along which to move the box. Thus, in this scenario, the pertinent latent state (x) corresponds to the team's next target location (i.e., one of the boxes or the flag), while the observable state (s) corresponds to the locations of the agents and boxes.

During task execution, the team members' mental models (x) can be misaligned. For example, Alice may target the box at the top left while Rob targets the box at the bottom left, thereby demonstrating suboptimal teamwork. Further, to improve the collaboration, team members may or may not choose to change their target at any point during the task based on behavior of the other teammate. Thus, as motivated earlier, the demonstrations of teamwork may not correspond to a game-theoretic equilibrium and can be suboptimal due to dependence on partially observable, dynamic mental models. The goal of the imitation learner is to learn the mental model-dependent policies of the team given these (potentially suboptimal) partially observable demonstrations of teamwork.

2 Related Work

Our work relates to the following three sub-areas of imitation learning: multi-agent imitation learning, learning from suboptimal demonstrations, and learning from partially observable demonstrations. Here, we summarize research from these sub-areas and relate it to our approach. Please see Appendix A for a more detailed discussion of MAIL techniques.

Multi-agent Imitation Learning (MAIL). Although multiple MAIL algorithms exist, the problem setting considered in prior art differs from the one considered in this paper. Reiterating from Sec. 1, prior MAIL techniques either learn behavior corresponding to game-theoretic equilibria of multi-agent systems [Song *et al.*, 2018; Lin *et al.*, 2019], assume homogeneity in agents' capabilities [Bhattacharyya *et al.*, 2019], or do not consider latent performance-shaping factors (such as mental models or cognitive states). Approaches that do model latent states assume that the latent state is either shared between members [Wang *et al.*, 2021; Ivanovic *et al.*, 2018] or time-invariant [Le *et al.*, 2017]. In contrast, inspired by real world teaming considerations, we seek to develop MAIL algorithms that both recognize the individual and dynamically changing latent states of each member and are capable of learning multi-agent policies from different levels of supervision over the latent states.

Learning from Partially Observable Demonstrations. Imitation learning with occlusions or missing features has also received increasing attention in the last decade. These techniques, while not directly valid for the multi-agent setting considered in this work, inform our work. [Torabi *et al.*, 2018; Sun and Ma, 2019] consider learning an agent policy from demonstrations that may not include data of the demonstrator's actions. Similarly, [Choi and Kim, 2011; Gangwani *et al.*, 2020] allow incomplete specification of states by utilizing a belief state. [Unhelkar and Shah, 2019] explicitly model an agent's latent decision factors (e.g., mental states), which can change dynamically within an episode. While related to our work, these techniques only consider single-agent tasks and do not model the interaction between multiple agents. [Bogert and Doshi, 2018] provide an approach for multi-robot inverse reinforcement learning from partially observable demonstrations. In contrast to our approach, their work does not model agents' mental states or seek to learn from demonstrations of suboptimal teamwork.

Learning from Suboptimal Demonstrations. While classical imitation learning assumes demonstrations are generated from experts who behave optimally, a few approaches admit that demonstrations can be suboptimal in practice. For example, with an assumption that the majority of demonstrations are optimal, [Choi *et al.*, 2019; Zheng *et al.*, 2014] focus on imitation learning that is robust to suboptimal outliers. Meanwhile, [Brown *et al.*, 2019; Chen *et al.*, 2021; Zhang *et al.*, 2021] aim to incorporate demonstrations that come from demonstrators whose level of expertise is unknown in order to overcome the challenge of scarce expert demonstrations. [Yang *et al.*, 2021] utilize a latent action representation while learning the optimal policy from potentially suboptimal demonstrations. While related to our approach, these solutions for imitation learning from suboptimal demonstrations neither consider demonstrators' mental states nor multi-agent tasks. In contrast, our goal is to develop an approach that can learn stochastic multi-agent policies from demonstrations that are *both* partially observable and suboptimal.

3 Problem Formulation

To formalize the problem of learning team policies from suboptimal and partially observable demonstrations, we first provide models for the team task and team members' behavior.

3.1 Task Model

Due to our focus on learning task-oriented team policies, we require a model to represent team tasks. Borrowing from prior research in multi-agent systems [Oliehoek and Amato, 2016], we build upon the framework of multi-agent Markov decision processes (MMDP) to describe the tasks of interest. An MMDP models sequential collaborative tasks and is specified by the tuple $M_{\text{task}} \doteq (n, S, A, T, R, \gamma)$, where

- n, is the number of agents i indexed $1:n$;
- $s \in S$, denotes the set of task states;
- $a_i \in A_i$, is the set of actions a_i available to the i-th agent;
- $A = \times_i A_i$ is the set of joint actions, where $a = [a_1, \cdots, a_n]$ denotes the joint action;
- $T_s(s'|s,a) : S \times A \times S \rightarrow [0,1]$ denotes the state transition probabilities, i.e., the probability of the next task state being s' after the team agents executed action a in state s
- $R(s,a) : S \times A \rightarrow \mathbb{R}$ is the joint reward that the team receives after execution action a in state s.
- γ is the discount factor.

The MMDP model assumes that all agents have a shared objective and each agent has full observability of the task state and reward. The shared objective of the set of n agents, whom we jointly refer to as the *team*, is to maximize their expected cumulative discounted reward, $\mathbb{E}[\Sigma_t \gamma^t R(s_t, a_t)]$. In this work, we focus on team tasks that can be modeled as MMDP where the set of states S and the set of actions A are finite. Several real-world tasks can be modeled using MMDP. For instance, the scenario described in the running example can be described as an MMDP with $n=2$, S modeling the task-relevant features (namely, the agent and box locations), and A modeling the actions available to the agent.

The solution to the MMDP is a set of n decentralized agent policies $\pi_{1:n}$, where π_i is the policy of the i-th agent. In the running example, this corresponds to policies of Alice and Rob. Mathematically, $\pi_i(a_i|s)$ is a probability distribution of the i-th agent's actions a_i conditioned on the MMDP state. Since each agent has full state observability, in theory, an MMDP can be solved optimally in a centralized manner by the team using MDP solvers [Puterman, 1990] before the task begins. If each team member follows this optimal policy faithfully, coordination between team members in MMDP tasks is guaranteed during task execution.

3.2 Agent Model

In practice, however, one seldom observes perfect coordination among team members, including in tasks where the agents have complete or near-complete observability of the task state and complete knowledge of the team's objective (e.g., healthcare team in an operating room, or a team of basketball or soccer players). The potential causes of this imperfect coordination are varied. For instance, imperfect coordination can occur due to inability to compute a joint policy, lack of prior coordination, imperfect execution, and different individual preference. To design an imitation learning algorithm that can effectively recover team policies, it is essential to explicitly consider these imperfections and latent causes of suboptimal teamwork.

Hence, to model teamwork observed in practice, we provide a latent variable model for each team member's (potentially suboptimal) behavior. Our model extends the Agent Markov Model (AMM), which explicitly models latent states of a single agent's behavior [Unhelkar and Shah, 2019], to model teamwork. In particular, we model each team member's behavior as the tuple $(X_i, b_{x_i}, T_{x_i}, \pi_i)$, where

- $x_i \in X_i$ denotes the latent states that influence the i-th agent's behavior during the task. These may include mental models, methods to tie-break if multiple optimal policies exist, or preferences over different task components.
- $b_{x_i}(x_i) \in X_i \rightarrow [0,1]$ denotes the probability distribution of the latent state at the start of the task.
- $T_{x_i}(x_i'|s, x_i, a, s') \in S \times X_i \times A \times S \times X_i \rightarrow [0,1]$ denotes the transition model of the latent state.
- $\pi_i(a_i|s, x_i) \in S \times X_i \times A_i \rightarrow [0,1]$ denotes the team member's policy, a probability distribution of each member's decision a_i conditioned on their decision factors (s, x_i).

Referring to the running example, behavior of Alice and Rob depends not only the task context (MMDP state) but also on their latent preferences over the next target location. For each team member, the agent model helps in modeling this latent preference (as $x_i \in X_i$), their latent state-dependent policy (as π_i), and the evolution of their latent preference (via b_{x_i} and T_{x_i}). While the above model is expressive and can represent a variety of team behaviors (e.g., suboptimal policies, evolution of latent preferences based on past behavior), we assume the transition dynamics T_x to be Markovian for computational tractability. Notationally, we jointly refer to the behavioral models for the whole team as (X, b_x, T_x, π), where $X = \times_i X_i$, $T_x = \{T_{x_1}, \cdots, T_{x_n}\}$, and $\pi = [\pi_1, \cdots, \pi_n]$. The latent state of whole team is denoted as $x = [x_1, \cdots, x_n]$.

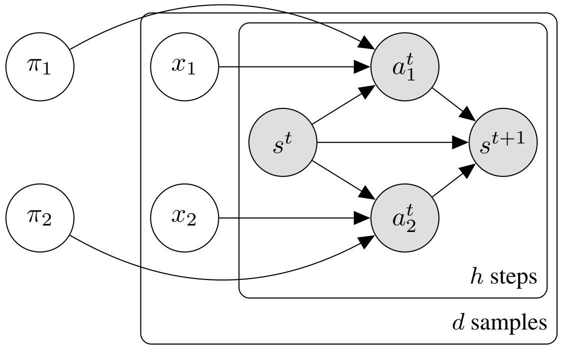

Figure 2: Dynamic Bayesian network for the behavior of a 2-agent team with time-invariant latent states depicted using plate notation.

3.3 Problem Statement

In the classical MAIL setting, the goal is to learn the team policy from a set of teamwork demonstrations[1] $\tau \doteq (s^{0:h}, a^{0:h})$, where h denotes the demonstration length. In our setting, however, team behavior is additionally influenced by the trajectories of team members' latent states $\chi \doteq (x^{0:h})$, which are partially observable and resource-intensive to annotate. Hence, we focus on semi-supervised learning of team policies, where x-labels are available for only a subset of the demonstrations.

Formally, our problem corresponds to learning the team policy π, given the MMDP task model (n, S, A, T, R, γ), a set of d observable demonstrations, $\tau_{1:d} \doteq \{\tau_m\}_{m=1}^{d}$, and labels of x for a subset $l(\leq d)$ of the demonstrations, $\chi_{1:l} \doteq \{\chi_m\}_{m=1}^{l}$. For our running example, the problem corresponds to recovering the behavioral policy of the two-agent team (Alice and Rob), given d observable trajectories of agent and box locations $\tau_{1:d}$ and labels of each agents' preferred target locations for a subset of the trajectories $\chi_{1:l}$.

4 Solution: Static Latent States

For ease of exposition, we first derive the policy learning algorithm for the case wherein team members do not change their latent mental model (x) during task execution; mathematically, $T_x \doteq \mathbb{1}(x{=}x')$. In the next section, we build upon the solution derived for this special case to solve the overall problem of Sec. 3.3. We note that, despite the simplification of static latent states, the learner needs to reason under partial state observability to learn the team policy.

Generative Model. To enable policy learning from a small number of demonstrations, we utilize a Bayesian approach and provide a generative model of team behavior. The generative model, shown in Fig. 2 for a two-agent team, models the process of generating team demonstrations. Each agent (indicated by the subscript) selects its action, $a_i^t \sim \pi_i(\cdot|s^t, x_i)$, at time step t, based on the current task state s^t, her mental state x_i, and policy π_i. In this special case, the latent state x_i does not change during task execution and, thus, is represented without a superscript. The dynamics of task state s^{t+1} depend on the MMDP model $T_s(\cdot|s^t, a^t)$.

To complete the generative process, the model additionally includes priors (not shown in Fig. 2) for the latent state and policy. In absence of any additional domain knowledge, we assume that the policy is given as a Categorical distribution. Hence, we define the prior of a policy as its conjugate prior, Dirichlet distribution: $\pi_{i,sx} \sim \text{Dir}(u^\pi)$, where $u^\pi = (u_1^\pi, \cdots, u_{|A|}^\pi)$ are hyperparameters. Similarly, we assume that the latent state is drawn from the uniform distribution unless additional information is given: $x_i \sim \text{Uni}(X)$. Given the generative model and data of semi-supervised demonstrations $(\tau_{1:d}, \chi_{1:l})$, the policy can be learned by maximizing the posterior: $p(\pi|\tau_{1:d}, \chi_{1:l})$.

Policy Learning with Supervision $(l{=}d)$**.** For the case when labels of x are available for the entire dataset, we can directly compute the posterior distribution of the policy as:

$$p(\pi_{i,sx}|\tau_{1:d}, \chi_{1:l}) = \text{Dir}(w_{i,sx}) \quad (1)$$

where, $w_{i,sxa} = u_a^\pi + \sum_{m=1}^{l} \sum_{(s',a') \in \tau_m} \mathbb{1}_{sxa}(s', x_{i,m}, a_i')$.

Policy Learning with Semi-supervision $(l < d)$**.** When labels of the latent state are only partially available, the likelihood $p(\tau_{1:d}, \chi_{1:l}|\pi)$ cannot be readily computed as it depends on unknown variables, namely, the subset of the data for which latent states x labels are unavailable: $\{x_m\}_{m>l}$. Hence, to calculate the posterior distribution in a computational tractable manner, we explore paradigms for approximate Bayesian computation. We note the computing the posterior through exact inference (i.e., brute-force) is intractable due to the high-dimensional nature of our problem. Inspired by prior work on single-agent modeling [Johnson and Willsky, 2014; Unhelkar and Shah, 2019], we utilize mean-field variational inference (MFVI) [Beal, 2003] and derive an MFVI algorithm for modeling team policies. In MFVI, the posterior of the team policy is approximated as the variational distribution $q(\pi)$ that maximizes the evidence lower bound (ELBO). For our problem, the ELBO corresponds to:

$$\mathcal{L}(q) \doteq \mathbb{E}_q \left[\log \frac{p(\pi, \{x_m\}_{m>l}, \text{data})}{q(\pi)\, q(\{x_m\}_{m>l})} \right] \quad (2)$$

The solution for the optimization problem, $\arg\max_q \mathcal{L}(q)$, corresponds to the iterative computation of the local $q(x)$ and global $q(\pi)$ variational distributions until convergence:

$$q(\pi_{i,sx}) = \text{Dir}\left(w_{i,sx}^\pi\right) \quad (3)$$

$$q(x_{i,m} = x) = (1/Z) \exp\left[\textstyle\sum_{\tau_k} \ln \tilde{\pi}_{i,sxa}\right] \quad (4)$$

where, $Z = \sum_{x' \in X} \exp\left[\sum_\tau \ln \tilde{\pi}_{i,sx'a}\right]$ is the partition function, the operator $\tilde{A}$ denotes $\exp\left(\mathbb{E}_{q'(A)}[\ln A]\right)$, and

$$w_{i,sxa}^\pi = u_a^\pi + \sum_{k=1}^{m} \mathbb{E}_{q'(x_{i,m})} \sum_{\tau_m} \mathbb{1}_{sxa}(s^t, x_{i,m}, a_i^t). \quad (5)$$

Given the posterior $q(\pi)$, the policy is simply estimated as the maximum a posteriori (MAP) estimate: $\hat{\pi} = \arg\max_\pi q(\pi)$.

[1]**Notation**: We use superscript to denote the time step. The subscript is overloaded and, based on the context, is used to denote the i-th agent, m-th demonstration, task state s, agent's latent state x, or action a. Finally, for notational convenience, we use $\mathbb{1}_{abc}(a', b', c')$ to denote $\mathbb{1}(a'{=}a, b'{=}b, c'{=}c)$.

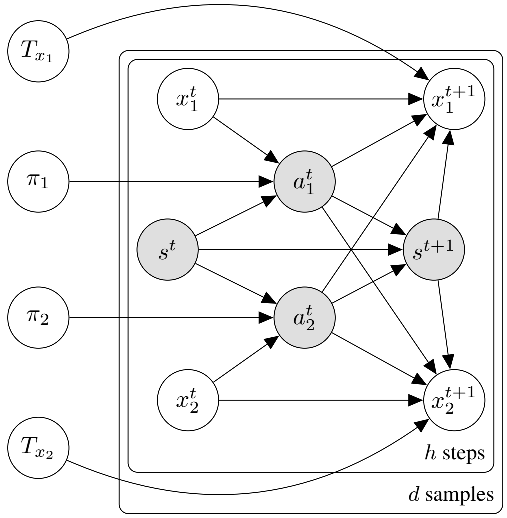

Figure 3: Dynamic Bayesian network for the behavior of a 2-agent team with time-varying latent states depicted using plate notation.

5 Solution: Dynamic Latent States

In contrast to the special case considered above, in practice, each member's task-specific mental model may evolve during the task. Hence, next, we extend the solution presented for static latent states to solve the general problem of Sec. 3.3. Analogous to the previous section, we first provide a generative model of team behavior and provide a MFVI-based approach for computing the MAP estimate of policy.

Generative Model. For the general setting, the generative process of team behavior additionally needs to model the temporal evolution of the team members' mental models. Thus, as shown in Fig. 3, we augment the generative model of Fig. 2 to include each agent's time-varying latent states x_i^t and their dynamics: $T_{x_i} \doteq P(x_i^{t+1}|s^t, x_i^t, a^t, s^{t+1})$. In general, how the mental model evolves may not be known a priori and, thus, requires specification of a prior distribution. Similarly to the policy prior, we utilize Dirichlet distribution as the prior for the latent state transition model, i.e, $T_{x_i,sxas'} \sim \mathrm{Dir}(u_1^T, \cdots, u_{|X|}^T)$, where u_x^T are hyperparameters. Both T_x and π jointly influence the likelihood of the semi-supervised demonstrations of team behavior, $p(\tau_{1:d}, \chi_{1:l}|T_x, \pi)$, where T_x specifies the distribution of the next latent state x^{t+1} and π that of the action a^t. Due to this dependence, to recover the team policy using MFVI, we need an approach to compute their joint posterior $p(T_x, \pi|\tau_{1:d}, \chi_{1:l})$.

Bayesian Team Imitation Learner (BTIL). Similar to Sec. 4, our solution to the overall problem – abbreviated as BTIL– includes iterative computation of variational distributions. However, in contrast to the previous section, the variational distributions additionally include the posterior of latent state dynamics T_x. The BTIL algorithm builds upon MFVI and is derived by maximizing the following ELBO:

$$\mathcal{L}(q) := \mathbb{E}_q\left[\log \frac{p\left(\pi, T_x, \{x_m^{0:h}\}_{m>l}, \text{data}\right)}{q\left(\pi\right) q\left(T_x\right) q\left(\{x_m^{0:h}\}_{m>l}\right)}\right] \tag{6}$$

Algorithm 1 Bayesian Team Imitation Learner (BTIL)

Input: $\tau_{1:d}, \chi_{1:l}$
Parameters: u^π, u^{T_x}, N, T_s

1: Initialize $w_i^\pi, w_i^{T_x}$ for $i=1:n$
2: Initialize posterior of all unlabeled states $q(\{x_m^{0:h}\}_{m>l})$
3: **while** $\mathcal{L}(\mathrm{q})$ converges **do**
4: Update the variational parameters $w_{1:n}^\pi, w_{1:n}^{T_x}$
5: **for all** τ_m **do**
6: Compute forward F and backward B messages
7: Update posterior of all unlabeled states $q(\{x_m^{0:h}\}_{m>l})$
8: **end for**
9: **end while**
10: Compute the policy posterior $q(\pi) \sim \mathrm{Dir}(w_i^\pi)$
11: **return** $\arg\max_\pi q(\pi)$

Alg. 1 provides the pseudocode of BTIL, which approximates the posterior distribution $p(T_x, \pi|\tau_{1:d}, \chi_{1:l})$ as independent variational distributions $q(\pi)q(T_x)$, where $q(\pi_i){=}\mathrm{Dir}(w_i^\pi)$ and $q(T_{x_i}){=}\mathrm{Dir}(w_i^{T_x})$. The estimates of posterior distributions are improved by iteratively updating the variational parameters w^π, w^{T_x} (line 4). Similar to Eq. 5, the variational parameters are updated as:

$$w_{i,jkass'}^{T_x} {=} u_k^{T_x} + \sum_{m=1}^{d} \mathbb{E}_{q(x)}\left[\textstyle\sum_t \mathbb{1}_{jkass'}(x_i^{t:t+1}, a^t, s_i^{t:t+1})\right]$$
$$w_{i,sxa}^\pi = u_a^\pi + \sum_{m=1}^{d} \mathbb{E}_{q(x)}\left[\textstyle\sum_t \mathbb{1}_{sxa}(s^t, x_{i,m}^t, a_i^t)\right] \tag{7}$$

An estimate of the posterior distribution of unlabeled states $q(x)$ is required to compute the expectations in Eq. 7. This local variational distribution is given as follows:

$$\begin{aligned} q(x_{i,m}^{0:h}) &\propto \exp\left(\mathbb{E}\left[\ln p(x_i, \text{data}|T_{x_i}, b_{x_i}, \pi_i, T_s\right]\right) \\ &= p(x_{i,m}^{0:h}, \tau_{i,m}|\tilde{T}_{x_i}, \tilde{\pi}_i, T_s, b_{x_i})/Z_{i,m} \end{aligned} \tag{8}$$

To compute Eq. 8 in a tractable manner, we define the following forward-backward messages. These messages are be computed in a recursive manner (line 6) as follows:

$$\begin{aligned} &F(t, j_1, \cdots, j_n) \\ &\quad \doteq P(x_1^t{=}j_1, \cdots, x_n^t{=}j_n, s^{0:t}, a^{0:t}) \\ &\quad = \sum_{k_1,\cdots,k_n} F(t-1, k_1, \cdots, k_n) T_s \prod_{i=1}^{n}\left(\tilde{T}_{x_i}\tilde{\pi}_i\right) \\ &B(t, j_1, \cdots, j_n) \\ &\quad \doteq P(s^{t+1:h}, a^{t+1:h}|x_1^t{=}j_1, \cdots, x_n^t{=}j_n, s^{0:t}, a^{0:t}) \\ &\quad = \sum_{l_1,\cdots,l_n} B(t+1, l_1, \cdots, l_n) T_s \prod_{i=1}^{n}\left(\tilde{T}_{x_i}\tilde{\pi}_i\right) \end{aligned}$$

$$F(0, j_1, \cdots, j_n) = \prod_{i=1}^{n} \tilde{b}_{x_i}\tilde{\pi}_i, \quad B(h, j_1, \cdots, j_n) = 1$$

For notational convenience, the subscript m is omitted. Given the forward F and backward messages B, BTIL computes the required local probabilities (line 7) as follows:

$$q(x^t) \propto F \cdot B$$
$$q(x_i^t) = \sum_{x_{-i}} q(x_1^t, \cdots, x_n^t)$$
$$q(x^t, x^{t+1}) \propto F \cdot \prod_i \left(\tilde{T}_{x_i} \tilde{\pi}_i\right) \cdot T_s \cdot B$$
$$q(x_i^t, x_i^{t+1}) = \sum_{x_{-i}} q(x^t, x^{t+1})$$

The time complexity of forward and backward messaging passing subroutine is $O(h|X|^{2n})$. Given the converged posterior $q(\pi)$ in line 10, the team policy is estimated as the MAP estimate (line 11).

6 Experiments

We evaluate BTIL using two domains, *Movers* and *Cleanup*, which include aforementioned features and challenges of real world teamwork. These domains build upon the *cooperative box pushing* task of [Oliehoek and Amato, 2016], include unambiguous latent preferences, and allow for opportunities of (mis)-alignment of team members' mental models. Video demonstrations of collaborative task execution in these domains are included in the supplementary material.

Due to the latent nature of mental models, labels of human team members' latent state cannot be ascertained without significant manual effort and annotation. Further, existing multi-agent dataset, to our knowledge, have not recorded mental models at each time step of collaborative tasks. Hence, to conduct the proposed experiments and compute pertinent metrics, we create two novel datasets for each domain: one synthetically generated and the other collected via human subject experimentation.

6.1 Domains

Movers. This domain realizes the running example of Sec. 1 in a 7×7 grid world. The two member team of Alice and Rob is tasked with carrying boxes to the goal position (flag). Boxes cannot be picked up by one agent alone; hence, to efficiently complete the task, the agents should coordinate on their latent preference over which box to pick or drop next. Each box can be either on its original location, held by both agents, or on the goal location. As described in the running example, agents can take one of the six actions at each step. In this domain, there are 38988 observable states and five possible mental states (corresponding to the three box pickup locations and two drop off locations), resulting in around 200000 states affecting each team member's decisions.

Cleanup. This domain has a similar configuration as *Movers*, but the environment includes lighter trash bags are placed instead of heavy boxes. Trash bags can only be picked up by one agent; hence, to effectively complete this task, each agent should carry different trash bags as much as possible. Each trash bag can be either on its original location, held by one of the agents, or be dropped at the goal location. In this domain, there are in total over 450000 states affecting agent decisions. The set of primitive actions and latent preferences available to each agent are same as the *Movers* domain.

6.2 Baselines and Metrics

Due to the novel features of our problem setting, to the best of our knowledge, existing algorithms do not readily apply to the general version of our problem. Existing MAIL solutions either do not model mental states, assume them to be aligned across all team members, or model them as time invariant. Hence, we benchmark our approach against baselines on special cases of our problem. In our design of experiments, we divide our problem into four settings based on two criteria: (a) whether the transition model of latent states T_x is known a priori or not, and (b) whether the latent states are completely labeled or not. We apply the behavioral cloning (BC) and MAGAIL as baselines for the setting of complete labels and the known transition model T_x [Pomerleau, 1991; Song *et al.*, 2018]. The implementation of BC and MAGAIL is similar to those used in [Ho and Ermon, 2016; Song *et al.*, 2018] respectively but adapted to handle discrete states. For the settings where labels are only partially available, we cannot apply existing algorithms; instead, we compare the performance of the supervised (BTIL-Sup) and the semi-supervised (BTIL-Semi) version of our approach. Implementation details of BTIL and the baselines are provided in Appendix D.

We evaluate our approach's ability to effectively learn team policies using the weighted Jensen-Shannon divergence (*JS Div.*) between the true and learned policies. Like [Unhelkar and Shah, 2019], the policy divergence metric is weighted by the relative counts of states (s, x) observed in the training set. The policy learning performance can only be computed in experiments with synthetic data, where the ground truth policy is known. Hence, in addition, we compare the ability to decode the unlabeled latent states using learned policies. In particular, we utilize the normalized Hamming distance (*Hamming*) between the decoded $\hat{x}_{i,m}^{0:h}$ and true $x_{i,m}^{0:h}$ sequences of latent states as the decoding metric. To decode the team member's latent states, as detailed in Appendix B, we extend the algorithm presented by [Seo *et al.*, 2021]. Lastly, to have a better sense of the worst case values of these highly nonlinear metrics, we utilize the Random baseline, which models the team policy as a Uniform distribution.

6.3 Results on Synthetic Data

We first present results on the synthetic dataset, which are summarized in Table 1. These experiments evaluate BTIL in two settings: with and without prior knowledge of T_x. We present additional results in Appendix E.

Data of Multi-agent Teamwork. The first dataset is synthetically generated by simulating teamwork between two artificial agents. For each domain, we implement the Markovian task model T_s, specify ground truth policies π_i, and transitions of the team members T_{x_i}. To arrive at the agent policies, we specify rewards associated with each latent state, utilize value iteration to compute Q-values, and derive stochastic policies π_i using the softmax operation over Q-values. Execution sequences are created by first assigning initial latent states x_i to each team member and, then iteratively, (a) sampling team members' action $a_i \sim \pi(\cdot|s, x_i)$, (b) sampling the next state $s' \sim T(\cdot|s, a)$, and (c) sampling the next latent

		Movers				*Cleanup*			
		Alice		Rob		Alice		Rob	
Setting	Algorithm	*Hamming*	*JS Div.*	*Hamming*	*JS Div.*	*Hamming*	*JS Div.*	*Hamming*	*JS Div.*
with T_x	*Random*	0.29 ± 0.00	0.14 ± 0.00	0.29 ± 0.00	0.14 ± 0.00	0.18 ± 0.00	0.32 ± 0.00	0.18 ± 0.00	0.36 ± 0.00
	BC	0.32 ± 0.02	0.15 ± 0.01	0.31 ± 0.03	0.15 ± 0.01	$\mathbf{0.06\pm0.01}$	0.22 ± 0.03	$\mathbf{0.03\pm0.02}$	0.17 ± 0.03
	MAGAIL	0.30 ± 0.03	0.21 ± 0.02	0.31 ± 0.03	0.24 ± 0.01	0.16 ± 0.03	0.29 ± 0.02	0.17 ± 0.05	0.38 ± 0.04
	BTIL-*Sup*	0.30 ± 0.01	0.07 ± 0.00	0.27 ± 0.01	0.07 ± 0.00	0.09 ± 0.02	0.20 ± 0.01	0.04 ± 0.01	0.19 ± 0.00
	BTIL-*Semi*	$\mathbf{0.15\pm0.02}$	$\mathbf{0.05\pm0.00}$	$\mathbf{0.16\pm0.02}$	$\mathbf{0.05\pm0.00}$	0.09 ± 0.02	$\mathbf{0.07\pm0.01}$	0.04 ± 0.01	$\mathbf{0.04\pm0.00}$
w/o T_x	*Random*	0.72 ± 0.00	0.14 ± 0.00	0.77 ± 0.00	0.14 ± 0.00	0.78 ± 0.00	0.32 ± 0.00	0.82 ± 0.00	0.36 ± 0.00
	BTIL-*Sup*	0.31 ± 0.01	0.07 ± 0.00	0.35 ± 0.01	0.07 ± 0.00	0.54 ± 0.01	0.20 ± 0.01	0.48 ± 0.03	0.19 ± 0.00
	BTIL-*Semi*	$\mathbf{0.30\pm0.01}$	$\mathbf{0.04\pm0.00}$	$\mathbf{0.33\pm0.01}$	$\mathbf{0.04\pm0.00}$	$\mathbf{0.42\pm0.01}$	$\mathbf{0.05\pm0.00}$	$\mathbf{0.36\pm0.01}$	$\mathbf{0.04\pm0.00}$

Table 1: Results on the synthetic data of multi-agent teamwork averaged over five learning trials.

state $x'_i \sim T_{x_i}(\cdot|x_i, s, a, s')$ until the task termination criteria or 200 time steps are reached. For each domain, we generate 200 demonstrations for training and 100 for evaluation. The proportion of suboptimal training demonstrations (as defined in Appendix C) is 49% and 7% for *Movers* and *Cleanup*, respectively.

BTIL Outperforms Baselines in Fully Supervised Settings. The first four rows of Table 1 provide results with 20 labeled demonstrations ($d{=}l{=}20$) and T_x as inputs. For this setting in the *Movers* domain, we observe that the supervised version of our algorithm (BTIL-Sup) learns more accurate team policies than the baselines. Other baselines (BC, MAGAIL) performed no better than the Random baseline. In *Cleanup*, our algorithm also outperformed other baselines in the policy learning metric (*JS Div.*). Despite a small training set, BTIL can learn the team policy with low JS-Divergence by effectively leveraging the x-labels and knowledge of T_x.

In terms of the decoding metric (*Hamming*), BC showed better results than BTIL in *Cleanup* domain. We posit that this trend occurs due to a combination of two reasons. First, the *Movers* domain requires tighter coordination relative to the *Cleanup* domain. In *Movers*, teammates must agree on which object to pick next to achieve coordination while in *Cleanup*, they only need to ensure that they are not picking the same object next. Second, the decoding metric (*Hamming*) assesses learning performance only on a subset of the state space (i.e., the states encountered in the test set); while *JS Div.* assess learning performance for the entire state space and, thus, is a better indicator of generalizability. This explains why, even in *Cleanup*, BC outperforms BTIL only on the decoding metric. In the *Movers* domain, which requires tighter coordination, BC performs poorly in both metrics.

BTIL Is Capable of Learning Team Policies Without Prior Knowledge of T_x. As shown in the bottom section of Table 1, BTIL can maintain its policy learning performance (*JS Div.*) even when T_x is unknown. These results suggest that BTIL is capable of learning T_x along with π and that joint learning of (π, T_x) is essential to imitation learning of team policies. Somewhat unsurprisingly, when the decoding algorithm utilizes the learnt T_x, the latent state decoding performance degrades relative to the known T_x case.

BTIL Effectively Utilizes Unsupervised Demonstrations to Improve Team Policy Learning. Lastly, we compare the performance of BTIL under semi-supervision, i.e., the general setting of Sec. 3.3. For these trials (denoted as BTIL-Sup), we provide the algorithm additional 180 demonstrations without latent state labels ($d{=}200, l{=}20$). Comparing performance of BTIL-Sup and BTIL-Semi, we observe improvement in policy learning performance, highlighting the ability of BTIL to effectively leverage available unsupervised data. This ability is particular critical in practice, where collecting data of (s, a)-tuples can be significantly less resource intensive than arriving at the labels of mental states. In additional experiments reported in Appendix E, which further investigate the effect of training set size and amount of semi-supervision on the learning performance, we observe that the decoding performance is enhanced with more labeled data and semi-supervision provides the most benefit when the labeled training set is small.

6.4 Results on Data of Human-AI Teamwork

Synthetically generated data, while being useful in validating policy learning performance, cannot capture the variability in behavior demonstrated by humans and human-agent teams. Hence, to benchmark our approach in more realistic settings, we evaluate our algorithm on a novel dataset of human-agent teamwork through an ethics board-approved human subject experiment with 33 participants (16 female, 17 male, mean age: 26.7 ± 5.3 years), who were recruited at Rice University.

Data Collection Procedure. To collect this novel dataset of human-AI teamwork, we designed a web-based interface shown in Fig. 4. Through this web-based interface participants completed tasks in the two domains (*Movers* and *Cleanup*) with an AI teammate. The human participant served the role of Alice, while the AI teammate served the role of Rob depicted as a robot avatar.

When the experiment starts, participants are asked to complete a short demographic survey. Further, before the participants conduct any tasks, they were provided with an interactive tutorial to make them familiar with the task and the interface. Upon completing the interactive tutorial, the experiment consisted of 9 sessions: 4 sessions on the *Movers* domains and 5 tasks for *Cleanup*. For each domain, the first two

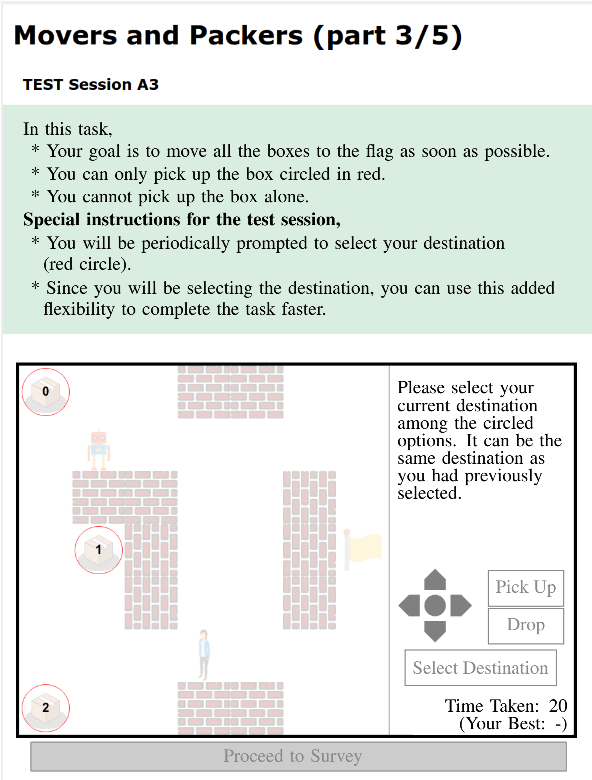

Figure 4: A still of the user interface designed and used for collecting data of human-agent teamwork. Please also see the supplementary material for video demonstrations of the user interface.

sessions were used as practice sessions. These practice sessions were designed for participants to gain further familiarity with the user interface. Through the tutorials and practice sessions, each participant was requested to complete tasks collaboratively with a robot avatar which behaves according to an AI policy. The AI policy is generated similar to the synthetic experiment, i.e, by specifying rewards associated with each mental model, running value iteration, and taking the softmax function over Q-values.

For each task, participants' actions (clicking action buttons) are logged along with the task state (location of Alice, Rob, and the boxes) and their chosen destination (mental state corresponding to which box to pick up or drop next). In general, collecting ground truth values of participants' mental states is challenging; in these experiments, we achieve this through a destination-selection interface detailed in Appendix C. By encouraging them to complete the task as soon as possible through the provided instructions and by showing "Your Best" score, we gamify the collaborative task and expect participants to take goal-oriented actions.

Through this experiment, we collect 66 trajectories for *Movers*, and 99 trajectories for *Cleanup*, of which every time step is labeled. The average lengths of trajectories (in terms of time steps) are 68.1 in *Movers* and 40.1 in *Cleanup*. The proportions of suboptimal demonstrations are 17% and 7% in *Movers* and *Cleanup*, respectively. We use two thirds of the collected trajectories for training and the rest for evaluation.

Algorith	Supervision	*Movers*	*Cleanup*
Random	N/A	0.72±0.00	0.77±0.03
BTIL-Sup	100%	0.14±0.02	0.30±0.02
BTIL-Semi	50%	0.16±0.02	0.35±0.03
BTIL-Semi	20%	0.20±0.02	0.42±0.02

Table 2: State decoding performance (Hamming distance) on the data of human-agent teamwork averaged over five learning trials.

Performance of BTIL Observed with Data of Multi-agent Teamwork Translates to Learning Policies of Human-agent Teamwork. As we cannot ascertain the true policy of a human, we utilize only the state decoding metric in these experiments. Further, as would be the case in practice, the mental model dynamics are also unavailable; hence, T_x needs to be learned by our algorithm and the MAGAIL baseline cannot be applied. Table 2 summarizes the decoding performance computed using π and T_x learned by variants of BTIL. All variants of BTIL are supplied with same amount of (s, a)-demonstrations but varying amount of x-labels. BTIL, even with a small amount of supervision, significantly outperform the Random baseline. In conjunction with the results computed with synthetic data, these experiments provide proof-of-concept in the ability of BTIL to learn team policies from small semi-supervised datasets of optimal and suboptimal teamwork.

7 Concluding Remarks

We provide BTIL, a Bayesian approach to learn team policies from demonstrations of suboptimal teamwork. In most collaboration scenarios, it is challenging to collect large labeled datasets of teamwork due to changes in team membership, adaptations in team policies, and the need for labeling latent states. Inspired by these and other aspects of teamwork observed in practice, BTIL includes multiple desirable features, including (a) the ability to learn from small sets of semi-supervised data, (b) explicit modeling of team members' mental models and model alignment and (c) the ability to jointly infer team policy, latent state dynamics, and latent states. We confirm the ability of our algorithm to learn team policies on two novel datasets of teamwork, including one of human-AI teamwork. Our work also offers several avenues of future work, including the ability to consider collaborative tasks where the task state s itself may be partially observable and consideration of communicative actions.

Acknowledgments

We thank the anonymous reviewers for their detailed and constructive feedback. Sangwon Seo was partially supported by the Army Research Office through Cooperative Agreement Number W911NF-20-2-0214. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

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

[Beal, 2003] Matthew James Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University College London, 2003.

[Bhattacharyya *et al.*, 2019] Raunak P Bhattacharyya, Derek J Phillips, Changliu Liu, Jayesh K Gupta, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 789–795. IEEE, 2019.

[Bogert and Doshi, 2018] Kenneth Bogert and Prashant Doshi. Multi-robot inverse reinforcement learning under occlusion with estimation of state transitions. *Artificial Intelligence*, 263:46–73, 2018.

[Brown *et al.*, 2019] Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*, pages 783–792. PMLR, 2019.

[Chen *et al.*, 2021] Letian Chen, Rohan Paleja, and Matthew Gombolay. Learning from suboptimal demonstration via self-supervised reward regression. In *Conference on Robot Learning*, pages 1262–1277. PMLR, 2021.

[Choi and Kim, 2011] Jaedeug Choi and Kee-Eung Kim. Inverse reinforcement learning in partially observable environments. *Journal of Machine Learning Research*, 12:691–730, 2011.

[Choi *et al.*, 2019] Sungjoon Choi, Kyungjae Lee, and Songhwai Oh. Robust learning from demonstrations with mixed qualities using leveraged Gaussian processes. *IEEE Transactions on Robotics*, 35(3):564–576, 2019.

[Gangwani *et al.*, 2020] Tanmay Gangwani, Joel Lehman, Qiang Liu, and Jian Peng. Learning belief representations for imitation learning in POMDPs. In *Uncertainty in Artificial Intelligence*, pages 1061–1071. PMLR, 2020.

[Ho and Ermon, 2016] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in Neural Information Processing Systems*, 29:4565–4573, 2016.

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

[Johnson and Willsky, 2014] Matthew Johnson and Alan Willsky. Stochastic variational inference for Bayesian time series models. In *International Conference on Machine Learning*, pages 1854–1862. PMLR, 2014.

[Le *et al.*, 2017] Hoang M Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *International Conference on Machine Learning*, pages 1995–2003. PMLR, 2017.

[Lin *et al.*, 2019] Xiaomin Lin, Stephen C Adams, and Peter A Beling. Multi-agent inverse reinforcement learning for certain general-sum stochastic games. *Journal of Artificial Intelligence Research*, 66:473–502, 2019.

[Oliehoek and Amato, 2016] Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

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

[Pomerleau, 1991] Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991.

[Puterman, 1990] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[Salas *et al.*, 2018] Eduardo Salas, Denise L Reyes, and Susan H McDaniel. The science of teamwork: Progress, reflections, and the road ahead. *American Psychologist*, 73(4):593, 2018.

[Seo *et al.*, 2021] Sangwon Seo, Lauren R. Kennedy-Metz, Marco A. Zenati, Julie A. Shah, Roger D. Dias, and Vaibhav V. Unhelkar. Towards an AI coach to infer team mental model alignment in healthcare. In *2021 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, pages 39–44, 2021.

[Song *et al.*, 2018] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 7472–7483, 2018.

[Sun and Ma, 2019] Mingfei Sun and Xiaojuan Ma. Adversarial imitation learning from incomplete demonstrations. In *IJCAI International Joint Conference on Artificial Intelligence*, page 3513, 2019.

[Tannenbaum and Salas, 2020] Scott Tannenbaum and Eduardo Salas. *Teams that work: the seven drivers of team effectiveness*. Oxford University Press, 2020.

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

[Torabi *et al.*, 2018] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4950–4957, 2018.

[Unhelkar and Shah, 2019] Vaibhav V Unhelkar and Julie A Shah. Learning models of sequential decision-making with partial specification of agent behavior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2522–2530, 2019.

[Wang *et al.*, 2021] Chen Wang, Claudia Pérez-D'Arpino, Danfei Xu, Li Fei-Fei, Karen Liu, and Silvio Savarese. Co-GAIL: Learning diverse strategies for human-robot collaboration. In *5th Annual Conference on Robot Learning*, 2021.

[Yang *et al.*, 2021] Mengjiao Yang, Sergey Levine, and Ofir Nachum. TRAIL: Near-optimal imitation learning with suboptimal data. *arXiv preprint arXiv:2110.14770*, 2021.

[Zhang *et al.*, 2021] Songyuan Zhang, Zhangjie Cao, Dorsa Sadigh, and Yanan Sui. Confidence-aware imitation learning from demonstrations with varying optimality. *Advances in Neural Information Processing Systems*, 34, 2021.

[Zheng *et al.*, 2014] Jiangchuan Zheng, Siyuan Liu, and Lionel M Ni. Robust Bayesian inverse reinforcement learning with sparse behavior noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.