Factorial Agent Markov Model: Modeling Other Agents' Behavior in presence of Dynamic Latent Decision Factors

Liubove Orlov-Savko
Rice University
Houston, TX, USA
liubove.orlov.savko@rice.edu

Abhinav Jain
Rice University
Houston, TX, USA
abhinav.jain@rice.edu

Gregory M. Gremillion
CCDC Army Research Lab
Aberdeen Proving Ground, MD, USA
gregory.m.gremillion.civ@army.mil

Catherine E. Neubauer
CCDC Army Research Lab
Aberdeen Proving Ground, MD, USA
catherine.e.neubauer2.civ@army.mil

Jonroy D. Canady
CCDC Army Research Lab
Aberdeen Proving Ground, MD, USA
jonroy.d.canady.civ@army.mil

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

ABSTRACT

Autonomous agents operating in the real world often need to interact with other agents to accomplish their tasks. For such agents, the ability to model behavior of other agents – both human and artificial – without complete knowledge of their decision factors is essential. Towards realizing this ability, we present Factorial Agent Markov Model (FAMM), a model to represent behavior of other agents performing sequential tasks. In contrast with most existing models, FAMM allows for behavior of other agents to depend on *multiple*, *time-varying* latent decision factors and does not assume rationality. To enable learning of FAMM parameters by observing behavior of other agents, we provide a set of variational inference algorithms for the unsupervised, semi-supervised, and supervised settings. These Bayesian learning algorithms for the FAMM enable agents to model other agents using execution traces and domain-specific priors. We demonstrate the utility of FAMM and corresponding learning algorithms using three synthetic domains and benchmark them against existing algorithms for modeling agent behavior. Our numerical experiments demonstrate that, despite the presence of multiple and time-varying latent states, our approach is capable of learning predictive models of other agents with semi-supervision.

CCS CONCEPTS

• **Computing methodologies** → **Intelligent agents**; *Learning from demonstrations*; *Semi-supervised learning settings*; *Markov decision processes*; • **Mathematics of computing** → Bayesian networks; Bayesian computation; **Variational methods**; • **Computer systems organization** → Robotic autonomy.

KEYWORDS

Humans and AI; Semi-Supervised Learning; Bayesian Inference

ACM Reference Format:
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 *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 9 pages.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

1 INTRODUCTION

With increasing adoption of artificially intelligent agents, their ability to work with humans and other agents is becoming increasingly critical. Reliance on these agents by humans is also increasing, as the agents shift from the role of tool to teammate and humans become peers. To accomplish their tasks effectively, these and similar agents need the ability to accurately model behavior of other agents in their environment. For instance, for safe and efficient navigation, autonomous cars require predictive models of other cars [16, 23]. Similarly, robots exhibit poor human-robot teamwork in absence of faithful models of their human collaborators [8, 15, 26], which typically need to be learned from small data sets [27].

Recognizing the need of artificial agents to model humans and other agents, multiple models and learning algorithms have been developed in the last two decades. For ease of exposition, henceforth, we refer to the ego-agent who is interested in modeling another agent as the *Observer*, and to the other-agent *simply* as the *Agent*. Existing techniques for the *Observer* to model an *Agent* include methods based on imitation learning [16, 20], type-based reasoning [1], and graphical models [13, 28], among others. Albrecht and Stone [2] provide a comprehensive survey of techniques to arrive at predictive models of other agents. As identified in the survey, most existing methods assume complete knowledge of factors that influence the other *Agent*'s decisions. Indeed, there has been limited emphasis on modeling other agents in presence of latent, time-varying decision factors (i.e., partially observable states).

In practice, however, an *Observer* seldom has complete observability (or even knowledge) of all the factors that influence the decisions of the *Agent*. This challenge is obvious in adversarial settings, where the other *Agent* has an incentive to hide or obfuscate its decision factors from the *Observer*. Interestingly, even during cooperative tasks within fully observable environments, it can be challenging for the *Observer* to observe the *Agent*'s decision factors. In cases where the *Agent* is a human, her decision may depend on cognitive variables such as workload, belief, and preferences; these cognitive decision factors are critical for modeling human behavior, but are difficult to sense [19]. Moreover, due to a variety of cognitive variables influencing behavior, often more than one latent factors are in play when humans and artificial agents interact in the real world. To add to the challenge, in domains where critical events are observed rarely or collecting observational data is intrusive, the *Observer* has to learn from small data sets of *Agent* behavior.

To address these challenges, in this work, we introduce the Factorial Agent Markov Model (FAMM). FAMM builds upon two generative Bayesian time series models – namely, the Agent Markov Model (AMM) and the Factorial Hidden Markov Model (FHMM) – to provide a unified framework for representing behavior of agents performing *sequential tasks* in *Markovian domains*. In contrast to its parent models [7, 28], FAMM explicitly models *multiple*, *time-varying*, *latent* decision factors as well as *actions* of the *Agent*. The term *factorial* in FAMM refers to the factored latent states in our model, i.e., states that are composed of a set of variables, and hence represented as vectors instead of scalars (cf. FHMM [7]). We detail the FAMM in Sec. 4, which provides a unified framework to mathematically formalize the problem of modeling *Agent* behavior (Sec. 5) and existing solutions (Sec. 7.2). Next, in Sec. 6, we derive Bayesian learning algorithms for the FAMM that enable modeling of agent behavior using data of its observed decision factors, actions, and (optionally) labels of the latent decision factors.

We evaluate our modeling contribution in three sequential tasks of varying complexity (Sec. 7). In each task, the *Agent*'s behavior depends on multiple time-varying decision factors, a subset of which is latent with respect to the *Observer*. We benchmark our approach against three techniques for modeling agent behavior: AMM, BehavioralCloning and InfoGAIL. Our experiments confirm the ability of FAMM to faithfully model the behavior of other agents despite presence of multiple, time-varying, latent decision factors. Further, we observe that the semi-supervised variant of our approach can perform comparably to supervised learning despite significantly fewer annotations. We also observe that FAMM either performs comparably to or outperforms the baselines in these experiments, thereby highlighting the advantage of our contribution. Encouraged by these results, we believe that FAMM can serve as a unified representation for agent behavior and can help inform the development of approaches to model humans and agents alike.

2 MOTIVATIONAL SCENARIO

To understand the associated computational challenges, let us consider a pedestrian navigating the RoadWorld shown in Fig. 1. The pedestrian wants to catch a bus, which stops at either end of the RoadWorld. An autonomous vehicle (not shown in the figure) is interested in modeling the navigation behavior of the pedestrian to compute a safe and efficient trajectory. In this motivating scenario, thus, the pedestrian serves the role of the *Agent*, while the autonomous vehicle corresponds to the *Observer*. Mathematically, the model of pedestrian behavior corresponds to her policy π, which denotes the probability distribution over her actions (i.e., whether to move up, down, left, right) conditioned on her decision factors (or, equivalently, states).

Even in this simplified setting, the pedestrian behavior can depend on a variety of decision factors, including her location, preferred bus stop, and the rules of the road (i.e., walk along the right or left side of the road). To the *Observer*, only a subset of these decision factors are observable. For instance, using its sensors the autonomous vehicle may be able to detect the pedestrian location but not her preferred bus stop. Moreover, the latent decision factors (such as the preferred bus stop) may change over time. In this work, we are interested in developing computational techniques to model *Agent* behavior in this and similar settings from their observable execution traces and human supervision. Due to the cost of collecting execution traces and human supervision, solutions that are both sample- and label-efficient are desirable.

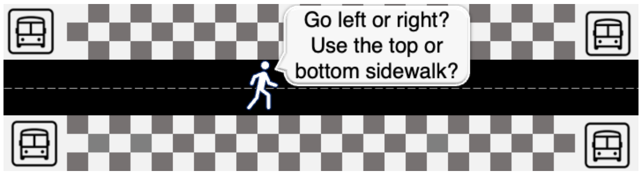

Figure 1: RoadWorld and the latent preferences affecting the pedestrians' policy. The pedestrian's goal is to reach either of the buses. However, the observed behavior depends not only the task goal but also on pedestrian's preferences.

3 RELATED WORK

Before describing our solution, we begin with a concise review of related work on modeling other agents and Bayesian models.

3.1 Latent Decision Factors

Design of future human-autonomy systems presents unique challenges, as human behavior is driven by mental states and models, and they cannot be assumed to act as rational agents [19, 25]. This motivates a need to understand and predict human decisions that govern interactions and behaviors in these systems, particularly as the nature of those interactions evolves to be sensitive to changes in context, time constraints, data confidence, and problem complexity. Towards this need, prior work has explored dependence of human behavior on a variety of mental states, including trust in automation, fatigue, workload, preferences, belief, and prior expertise [5, 14, 21, 24]. During a task, these mental states are difficult to observe and may even change over time. To arrive at predictive models of behavior that depend on these latent states, multi-modal measures of human physiological response from wearable sensor technologies offer one avenue to infer these partially observable states [19, 30]. For example, based on research in psychophysiology, arousal and fatigue states can be detected via electroencephalogram, electrocardiogram [4], pupil diameter [17], and eyelid closure [29].

However, a gap exists for effectively utilizing this available domain expertise for learning predictive models of behavior, especially in the presence of multiple latent decision factors. This gap is especially critical in settings where data collection is challenging and utilizing the domain expertise is essential to learning accurate predictive models. Towards addressing this gap, our work provides a novel representation that explicitly models multiple latent states and a semi-supervised approach to learning behavioral models. While we have emphasized latent decision factors in human behavior thus far, similar challenges exist for modeling artificial agents whose behavior often depends on parameters latent to an *Observer* (such as reward functions, discount factor and learning rates).

3.2 Observational Learning of Agent Behavior

The aforementioned complex relationship between environment, task context, latent decision factors, and behavior motivates the use of data-driven modeling approaches to predict actions of the *Agent*. Several methods to model and imitate other agents by observing their behavior have been developed in the last two decades [2, 3, 20]. Popular paradigms for observational learning of *Agent*'s sequential decision-making behavior include policy learning, inverse reinforcement learning, plan recognition, among others [2]. The approach presented here is closest to the paradigm of direct policy learning, wherein the agent's objective (reward) is not explicitly modeled and, thus, assumptions of rationality are unnecessary.

Although several method exist for policy learning from demonstrations, most do not model latent decision factors [2, 20]. The solutions presented in this work build upon one recent framework, the Agent Markov Model (AMM), which addresses this limitation and enables policy learning in presence of latent states [28]. In AMM, the *Agent*'s decision-making depends on both task context and a scalar mental state. In practice, however, *Agent* behavior may depend on more than one latent feature, e.g., both workload and preference. Thus, we explore an extension of AMM, wherein agent policies can depend on multiple latent states. Both this work and the AMM utilize Bayesian inference for policy learning.

Recently, guided by advances in deep learning and generative modeling, deep approaches to agent modeling have also received increasing interest [9, 11, 12, 16]. For instance, Ho and Ermon [9] provide GAIL, which utilizes generative adversarial training. InfoGAIL extends this approach to account for latent decision factors (or modes) in agent behavior [16]. These methods are powerful tools for learning high-dimensional nonlinear policies and allow for data-driven discovery of latent decision factors, especially when sufficient data is available. However, the latent modes considered in these methods can lack interpretability and are assumed to be time-invariant. This lack of interpretability can limit the use of domain knowledge (e.g., specification of latent decision factors based on physiology) during policy learning. In circumstances where data collection is challenging, such as human-autonomy teaming, the use of domain knowledge to partially constrain the model is an attractive alternative to these purely data-driven methods.

4 FACTORIAL AGENT MARKOV MODEL

To enable modeling of multiple interpretable latent decision factors and learning from small data sets by utilizing domain expertise, we adopt a Bayesian perspective and provide the Factorial Agent Markov Model (FAMM). FAMM is an *Observer*-centric generative model of *Agent* behavior, which extends the AMM to encode more general *Agent* behaviors. The design of FAMM is also informed by long-standing Bayesian time series models, namely, the Hidden Markov Model (HMM) and its extension, the Factorial HMM [7]. In addition to the latent state and observations in HMM, our model additionally includes actions for modeling decision-making. Similar to AMM, FAMM models *Agent* behavior to depend on decision factors (x) that are time-varying and latent to the *Observer*. However, while AMM assumes only one latent decision factor (x), FAMM allows for presence of multiple latent states, $\boldsymbol{x} = (x_a, x_b, ..., x_m)$. Thus, analogous to FHMM, which extends HMM to consider multiple latent states and provides both computational and modeling benefits, FAMM provides a factorial extension of AMM for modeling behavior of other agents.

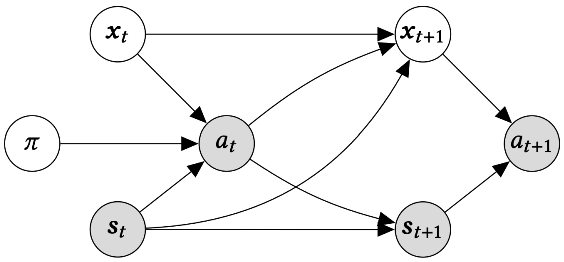

Figure 2: The Factorial Agent Markov Model (FAMM) represented using a two-timeslice Dynamic Bayesian Network. Each node represents a (multi-dimensional) random variable, with observable variables shown in grey. At each time step, the agent selects its action a_t based on its decision factors $(\boldsymbol{x}_t, \boldsymbol{s}_t)$ and policy π. The transition of decision factors in turn depends on the previously selected action.

This extension is motivated by both modeling and computational reasons. In practice, behavior of humans and other agents often depends on multiple cognitive states, such as goals, beliefs, trust, and workload. Allowing the generative model to encode multiple x, enables straightforward modeling of these multiple cognitive states. On the computational front, guided by factorial extension of HMM [7], we posit that FAMM will enable more efficient and interpretable model learning. A dynamic Bayesian representation of the FAMM is provided in Fig. 2. Mathematically, the FAMM is specified as the tuple $(S, X, A, T_s, T_x, b_s, b_x, \pi)$, where[1]

- $\boldsymbol{s} \in S$, are the set of decision factors observable to both the *Agent* and *Observer*;
- $\boldsymbol{x} \in X$, are the set of decisions factors that are latent to the *Observer*. $\boldsymbol{x}$ is factored and includes multiple features $\boldsymbol{x} = (x_a, x_b, \cdots, x_m)$. The set of i-th feature is given as X_i;
- $a \in A$, are the set of decisions[2] available to the *Agent*;
- $b_s(\boldsymbol{s}_0)$ and $b_x(\boldsymbol{x}_0)$ are the probability models for the initial values of the observable and latent decision factors;
- $T_s(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a_t)$ and $T_x(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{s}_t, a_t)$, are the transition models of the observable and latent decision factors;[3] and
- $\pi(a_t|\boldsymbol{s}_t, \boldsymbol{x}_t)$ denotes the *Agent* policy, which models the agent's probability of selecting action a_t in state $(\boldsymbol{s}_t, \boldsymbol{x}_t)$.

For the RoadWorld, $\boldsymbol{s}$ corresponds to the agent position, x_a to agent's preference on its goal (left or right), and x_b on the sidewalk preference (top or bottom). T_s models the physics of the environment, while T_x is simply identity (i.e., in this case, the latent decision factors are modeled as time-invariant). In more general settings, T_x models the evolution of latent states. We denote an execution sequence of agent's behavior as $\tau \doteq \{\boldsymbol{s}_{0:K}, \boldsymbol{x}_{0:K}, a_{0:K}\}$, where K denotes the length of the sequence. Given the FAMM describing an agent's behavior, the probability of its execution sequence is

$$p(\tau) = b_x(\boldsymbol{x}_0) \prod_{t=0}^{K} \pi(a_t|\boldsymbol{s}_t, \boldsymbol{x}_t) T_s(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a_t) T_x(\boldsymbol{x}_{t+1}|\boldsymbol{s}_t, \boldsymbol{x}_t, a_t)$$

[1] We use subscripts to denote both time and feature indices based on the context.
[2] We use the terms of action and decisions to describe a interchangeably.
[3] We assume that the latent decision factors represent cognitive states of the *Agent*, which influence transition of state $\boldsymbol{s}$ indirectly via actions.

In practice, due to the presence of latent states $\boldsymbol{x}$, only a subset of these sequence is observable. We denote this observable sequence as $\zeta \doteq \{\boldsymbol{s}_{0:K}, a_{0:K}\}$.

5 PROBLEM STATEMENT

Motivated by the need to model other agents from small sets of data and by utilizing domain expertise, we formulate the problem of learning an FAMM. We consider the setting where a human expert had identified the key features relevant for modeling *Agent* behavior (S, X), a subset of which may be unobservable. Further, we assume that the relevant parameters of the domain, namely (A, T_s, b_s), and latent states (b_x, T_x) are also available based on domain expertise. We highlight that computationally learning each of these problem inputs are complementary problems, with a few existing solutions [6, 18]. These solutions can be potentially used as a preprocessing step to the problem of agent modeling, if a particular problem input is unavailable.

Given these specifications, we seek to learn the *Agent* policy from observable execution sequences ζ and semi-supervision of latent states. Mathematically, we consider the problem of learning the behavioral policy π of an *Agent*, whose true behavior is given by the FAMM using the following inputs

- partial FAMM tuple, $(S, X, A, T_s, T_x, b_s, b_x)$
- N observable execution sequences of agent's behavior, $\zeta^{1:N}$
- partial supervision of latent factors $\boldsymbol{x}^{0:M}$, with $M < N$.

As discussed in Sec. 3, the partial supervision of $\boldsymbol{x}$can be derived using psychophysiology and, in certain domains, through human annotation. We believe that utilization of this domain expertise is both beneficial and important to the problem of *Agent* modeling, as it has the potential for model learning from smaller data sets (relative to unconstrained approaches) and results in interpretable latent decision factors.

6 FAMM POLICY LEARNING

As discussed in Sec. 3, multiple paradigms exist to learn *Agent* policies from observational data. Due to our focus on learning from small datasets, we explore a Bayesian approach. Bayesian methods by encoding structure in the problem and through probabilistic priors have the potential to learn from small datasets [27].

6.1 Algorithm Overview

In the general case, calculating the exact posterior of the policy is computationally expensive, and hence intractable to obtain. We use mean-field variational inference (MFVI) [10] to overcome this problem, which allows us to approximate the posterior $p(\pi, \boldsymbol{x}|\zeta)$ by a parametric probability distribution $q(\pi, \boldsymbol{x}; \lambda)$. We highlight that to learn the policy, we also need to infer the latent decision factors corresponding to the unsupervised execution sequences of agent's behavior. To compute the joint posterior distribution, MFVI assumes the posterior distribution of each unknown quantity to be independent, $q(\pi, \boldsymbol{x}) = q(\pi)q(\boldsymbol{x})$, and seeks to maximize the evidence lower bound:

$$\arg\max_{q} \mathbb{E}_q \left[\log \frac{p(\boldsymbol{x}, \pi)}{q(\boldsymbol{x})q(\pi)} \right] \tag{1}$$

Algorithm 1: Semi-Supervised FAMM Policy Learning

```
Data: Unlabeled execution sequences $\zeta^{1:N}$,
      Semi-supervision for latent decision factors $\boldsymbol{x}^{0:M}$ ,
      Known FAMM parameters $(T_s, T_x, b_x)$, and
      hyper-parameters $(\rho, K)$
Result: Learned policy $\hat{\pi}$ and latent decision factors $\hat{\boldsymbol{x}}^{0:N}$
Initialize $\lambda \leftarrow \rho$; and $\pi(\cdot|s, \boldsymbol{x}) \sim \text{Dirichlet}(\lambda)$;
for $K$ epochs do
    Local Update
        $P(\boldsymbol{x}_t = \boldsymbol{j}_{\boldsymbol{x}}) \propto F(t, \boldsymbol{j}_{\boldsymbol{x}})B(t, \boldsymbol{j}_{\boldsymbol{x}})$
    Global Update
        $\beta_a^{\boldsymbol{s},\boldsymbol{x}} \leftarrow \mathbb{E}_{q(\boldsymbol{x})} \sum_t \mathbb{I}(a_t = a, \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{s}_t = \boldsymbol{s})$
        $\lambda_a^{\boldsymbol{s},\boldsymbol{x}} \leftarrow \rho_a^{\boldsymbol{s},\boldsymbol{x}} + \beta_a^{\boldsymbol{s},\boldsymbol{x}} \quad \forall a \in A, \boldsymbol{s} \in S, \boldsymbol{x} \in X$
Return $\hat{\pi}(\cdot|s, \boldsymbol{x}) \sim \text{Dirichlet}(\lambda)$, $\hat{\boldsymbol{x}}^{0:N}$;
```

To solve this optimization problem for the FAMM, we develop Algorithm 1. Algorithm 1 computes the (local) optimum by iteratively updating the parameters of the local $q(\boldsymbol{x})$ and global $q(\pi)$ variational factors. We provide MFVI-based policy learning algorithms for three problem settings: unsupervised, where only observable expert trajectories $\zeta^{1:N}$ are provided; supervised, where the latent states are also made available during training; and semi-supersvised, as described in Sec. 5. Often labeled data is difficult to collect, especially of latent features corresponding to an *Agent*'s cognitive states. This requirement motivates the development of algorithms that can take advantage of both labeled and unlabeled data. Thus, we focus on semi-supervised setting, where only partial access to the latent states is possible. We first describe the semi-supervised setting in detail, and then arrive at algorithms for the other two cases as its special case. As we will see next, the choice of problem setting only affects the computation of local updates.

6.2 Local Updates

The local variational updates correspond to estimating the posterior of the latent decision factors, $\boldsymbol{x}$. This posterior can be efficiently calculated using forward-backward message passing. We define the forward and backward messages for the FAMM as follows

$$F(t, \boldsymbol{j}_{\boldsymbol{x}}) \equiv P(\boldsymbol{x}_t = \boldsymbol{j}_{\boldsymbol{x}}, \boldsymbol{s}_{0:t}, a_{0:t}) \tag{2a}$$

$$B(t, \boldsymbol{j}_{\boldsymbol{x}}) \equiv P(\boldsymbol{s}_{t+1:N}, a_{t+1:N}|\boldsymbol{s}_{0:t}, \boldsymbol{x}_t = \boldsymbol{j}_{\boldsymbol{x}}) \tag{2b}$$

where, $\boldsymbol{j}_{\boldsymbol{x}} = (j_a, \ldots, j_m)$ is a specific value of the multi-dimensional latent state $\boldsymbol{x}$. To compute these messages efficiently, we employ iterative forward-backward message passing by utilizing the following derived formulae,

$$\begin{aligned} F(t, \boldsymbol{j}_{\boldsymbol{x}}) = \sum_{\boldsymbol{k}_{\boldsymbol{x}} \in X} \Big(& F(t-1, \boldsymbol{k}_{\boldsymbol{x}}) T_s(\boldsymbol{s}_t | a_{t-1}, \boldsymbol{s}_{t-1}) \\ & T_x(\boldsymbol{j}_{\boldsymbol{x}} | \boldsymbol{k}_{\boldsymbol{x}}, \boldsymbol{s}_{t-1}, a_{t-1}) \pi(a_t | \boldsymbol{s}_t, \boldsymbol{x}_t = \boldsymbol{j}_{\boldsymbol{x}}) \Big) \\ B(t, \boldsymbol{j}_{\boldsymbol{x}}) = \sum_{\boldsymbol{k}_{\boldsymbol{x}} \in X} \Big(& B(t+1, \boldsymbol{k}_{\boldsymbol{x}}) T_s(\boldsymbol{s}_{t+1} | \boldsymbol{s}_t, a_t) \\ & T_x(\boldsymbol{k}_{\boldsymbol{x}} | \boldsymbol{j}_{\boldsymbol{x}}, \boldsymbol{s}_t, a_t) \pi(a_{t+1} | \boldsymbol{s}_{t+1}, \boldsymbol{x}_{t+1} = \boldsymbol{k}_{\boldsymbol{x}}) \Big) \end{aligned} \tag{3}$$

where, similar to $\boldsymbol{j_x}$, $\boldsymbol{k_x} = (k_a, \dots, k_m)$ is a specific value of the multi-dimensional latent state $\boldsymbol{x}$. To iteratively compute the forward and backward messages, their initial values are needed. These are given as follows,

$$\begin{aligned} B(N, \boldsymbol{j_x}) &= 1 \\ F(0, \boldsymbol{j_x}) &= b_x(x_0 = \boldsymbol{j_x})\pi(a_0|s_0, x_0 = \boldsymbol{j_x}) \end{aligned} \tag{4}$$

Given the forward and backward messages, the posterior of the latent state is given as

$$P(\boldsymbol{x_t} = \boldsymbol{j_x}|s_{0:N}, a_{0:N}) \propto F(t, \boldsymbol{j_x})B(t, \boldsymbol{j_x}). \tag{5}$$

In the semi-supervised setting, a subset of the executions are labeled, i.e., for these instances, the true value of $\boldsymbol{x}$is known. We treat these labeled instances separately. In this case, the latent states are converted to probability vectors correspond to their (known) posterior, where the entry corresponding to the value of the given latent state equals to one. After the change of representation from latent state to probability vector, the probability vectors for both given and estimated latent states are combined to form the posterior $q(\boldsymbol{x})$ and used indifferently in the global update.

6.3 Global updates

Due to our focus on discrete action spaces, in absence of any additional domain knowledge, we assume the policy to be a Categorical distribution. For a computationally efficient global update rule, it is prudent to select a conjugate prior for the global variational factors. Hence, we use the Dirichlet distribution as the prior for policy learning

$$\pi(\cdot|\boldsymbol{s}, \boldsymbol{x}) \sim \text{Dirichlet}(\rho^{\boldsymbol{s},\boldsymbol{x}}_{1:|A|}) \tag{6}$$

where, $\rho^{\boldsymbol{s},\boldsymbol{x}}_{1:|A|}$ are hyper-parameters. In absence of any prior knowledge, the hyperparameters can be simply selected as $1/|A|$, or tuned using a subset of the training dataset. On the other hand, if domain knowledge is available, the prior and its hyperparameter can be modified to incorporate the same. Due to our choice of a conjugate prior, the variational factor for computing the policy posterior is also a Dirichlet distribution,

$$q^*(\pi^{\boldsymbol{s},\boldsymbol{x}=\boldsymbol{j_x}}) = \text{Dirichlet}(\lambda^{\boldsymbol{s},\boldsymbol{x}=\boldsymbol{j_x}}))$$

where, $\lambda^{\boldsymbol{s},\boldsymbol{x}} = (\lambda_0, \dots, \lambda_{|A|})\forall(\boldsymbol{s}, \boldsymbol{x}) \in (S \times X)$ are variational parameters that need to be learned to arrive at an estimate of the policy. The update rule for policy parameters λ is given as

$$\beta^{\boldsymbol{s},\boldsymbol{x}}_a \leftarrow \mathbb{E}_{q(\boldsymbol{x})} \sum_t \mathbb{I}(a_t = a, \boldsymbol{x}_t = \boldsymbol{x}, \boldsymbol{s}_t = \boldsymbol{s}) \tag{7}$$

$$\lambda^{\boldsymbol{s},\boldsymbol{x}}_a \leftarrow \rho^{\boldsymbol{s},\boldsymbol{x}}_a + \beta^{\boldsymbol{s},\boldsymbol{x}}_a \quad \forall a \in A, \boldsymbol{s} \in S, \boldsymbol{x} \in X \tag{8}$$

where, β_k are computed efficiently using the previously computed forward and backward messages. The algorithm terminates after a prespecified number of epochs. We highlight that our overall approach is one of generative modeling and, hence, can be used to generate a variety of estimates of the learned policy from its posterior. In our experiments, we sample a policy using the learned posterior distribution from the final epoch, i.e., $\pi \sim \text{Dirichlet}(\lambda)$, to arrive at this estimate. A few alternate approaches are to use the mean or mode of the posterior distribution to arrive at the policy estimate; however, we leave exploration of the relative performance each alternative to future work.

6.4 Supervised and Unsupervised Learning

The supervised and unsupervised settings are special cases of the semi-supervised setting. In the case of complete supervision, we convert all latent sequences to probability vector sequences, as described in 6.2, and use those during the global update. Thus, in this setting, the learning algorithm converges after one iteration, since the local updates remain constant across each iteration. On the other hand, when no information about latent states is available, the local updates are computed solely through the observable expert demonstrations ζ using the forward-backward message passing algorithm, described also in 6.2.

7 EXPERIMENTS

We evaluate our approach through a suite of experiments in synthetic domains. For each domain, we handcraft ground truth models for the task and agent, which are used to generate data for experiments. We first benchmark Bayesian FAMM learning against existing techniques and, through ablation studies, characterize the sample- and label-efficiency on our approach.

7.1 Domains

We create and utilize three domains of varying complexity, where the *Agent* behavior depends on multiple latent decision factors. We note that existing data sets of human or agent behavior used in related works typically do not include ground truth labels of latent states states and agent policy, most likely due to the difficulty of annotating cognitive states and estimating the latent policy (the problem that our work seeks to address). By utilizing synthetic domains, we can create these ground truth values and evaluate our algorithm. Collection and annotation of novel datasets of human and agent behavior to evaluate and apply our approach is of high interest and an immediate avenue for future work.

Each domain, introduced next, corresponds to a pair of task and agent models. The task model, defined as an MDP/R, provides a specification of the agent's environment. In all experiments, both the *Agent* and *Observer* have full observability of the task state. The *Agent* behavior, however, depends not only on the task state but also additional decision factors (such as fatigue, trust, and preferences). These additional decision factors are observable to the *Agent* but not to the *Observer*. We define the ground truth model of agent behavior as an FAMM, which explicitly models both the observable and latent decision factors of the *Agent*.

7.1.1 RoadWorld. As our first domain, we utilize a discretized verison of RoadWorld described in Sec. 2. Here, the observable state $\boldsymbol{s}$ corresponds to the agent location and the action a models navigation (move: left, right, up, down, stay). The transition of the observable state $\boldsymbol{s}$ is modeled as deterministic. Agent behavior in RoadWorld additionally depends on two latent decision factors $\boldsymbol{x} = (x_1, x_2)$: x_1 denotes the agent's preferred bus stop and x_2 denotes the preference to walk on left or right side of the road (i.e., top or bottom sidewalk). The size of agent's state space $S \times X$ and action space A is 120 and 5, respectively. The agent policy models goal-oriented behavior and depends on both $\boldsymbol{s}$ and $\boldsymbol{x}$.

7.1.2 BoxWorld. Our second domain is inspired by applications from disaster response, where the *Agent* models a first responder

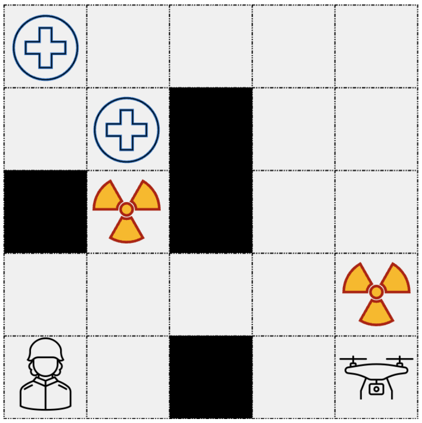

Figure 3: TeamBoxWorld.

(Fig. 3). The *Agent* (depicted as human) is responsible for rescuing items of interest (such as, first-aid kits, depicted as blue circles with cross), which it does by visiting said objects. Due to the disaster, some areas are radioactive and adversely affect the *Agent.* We assume that the *Agent* has complete observability of the environment, i.e., the location of itself, objects, and threats. The actions available to the *Agent* are identical to the RoadWorld. Its decisions depend on both observable and latent states. The observable state $\boldsymbol{s}$ includes the following features: agent location and a binary feature for each object (indicating whether the object has been retrieved or not).

The latent state $\boldsymbol{x}$ consists of two decision factors: x_1 models urgency (a Boolean variable) and x_2 models fatigue (a ternary variable, with values low, medium and high). Mathematically, we arrive at the ground truth agent policy $\pi(a|\boldsymbol{s},\boldsymbol{x})$ by first defining a reward function $R(s,x_1)$ that depends on urgency; computing the corresponding $Q(s,x_1)$ value using MDP solvers; and then selecting actions based on fatigue-dependent soft-maximum: $\pi \propto \exp[Q(s,x_1)/x_2]$. Intuitively, if the agent considers task to be urgent, then it disregards the threats; otherwise it seeks to retrieve the objects while avoiding the threats by assigning threats a high negative reward $R(s,x_1)$. Fatigue influences the agent's ability to take rational decisions via the softmax function. The size of agent's state and action space is 504 and 4, respectively.

7.1.3 TeamBoxWorld. Our third domain is a variant of BoxWorld, where the first responder (*Agent*) has access to an aerial drone. As the drone is not affected by the radioactivity, the *Agent* can team up with it to complete the task both safely and efficiently. This domain focuses on human's use of autonomy, which has been shown to depend on her trust in autonomy [22]. In this domain, the *Agent* in addition to its navigation actions can deploy the drone to retrieve an object (send robot to: object 1, object 2, ..., stop robot). Given an object specified by the *Agent*, the drone moves to the object autonomously and then retrieves it with a success rate of 80%. Thus, in addition to the states of BoxWorld, the observable decision factors of the *Agent* also include the drone state. For this domain, we assume that the *Agent* always seeks to avoid the threats (i.e., x_1 =low). However, along with fatigue x_2, the *Agent* behavior now depends on another latent quantity: trust in autonomy (x_3). We model x_3 as a binary variable (with values low and high). The *Agent* chooses not to use the drone if x_3 is low; otherwise it deploys the drone to complete the task efficiently. The size of agent's state space is 1512 and action space is 6 (4 navigational actions and 2 instructional actions for robot to retrieve 2 objects).

7.1.4 Generating training and testing datasets. Thus far, we have described the states $(\boldsymbol{s},\boldsymbol{x})$, actions a, transition model T_s, and policy π for each domain. The transition model of latent states T_x differ across experiment settings, wherein we explore both the static and time-varying settings. Given these specifications, we generate data of agent's behavior by first specifying the initial value of the decision factors $(\boldsymbol{s},\boldsymbol{x})$ and then iteratively sampling the agent's next action $a \sim \pi(\cdot|\boldsymbol{s},\boldsymbol{x})$, observable state $\boldsymbol{s}' \sim T_s(\cdot|\boldsymbol{s},a)$, and latent state $\boldsymbol{x}' \sim T_x(\cdot|\boldsymbol{s},a,\boldsymbol{x})$ until the task terminates. The resulting data corresponds to a set of agent's execution traces, where each trace is a sequence of $(\boldsymbol{s},\boldsymbol{x},a)$-tuples.

7.2 Baselines

We benchmark our approach using three approaches: BehavioralCloning, InfoGAIL, and Bayesian AMM learning. All algorithms, including FAMM learning, are implemented using Python. Our implementation of BehavioralCloning (BC) and InfoGAIL employs network architecture similar to that used in [16] for modeling 2D synthetic environments. In the unsupervised setting, BC does not model latent decision factors; while in the supervised setting, a Densely connected layer is used to encode the provided labels of latent modes. Additional implementation details of the algorithms are provided in the supplementary material.

InfoGAIL and AMM are recent techniques that both consider latent states to model *Agent* behavior. However, in contrast to FAMM, they both assume the latent state to be one-dimensional and, in case of InfoGAIL, model it as time-invariant. Thus, the baselines allow us to evaluate the effect of modeling latent states (cf. BehavioralCloning), their dynamics (cf. InfoGAIL), and their factored nature (cf. InfoGAIL and AMM). We enable InfoGAIL and AMM to learn in presence of multiple latent decision factors by flattening the multi-dimensional latent state. For instance, in RoadWorld, two latent features with two values each are represented as one latent feature with four values during the learning process.

7.3 Metrics

We evaluate the learning performance using metrics for model learning as well as the ability of the learned model to decode latent decision factors on test datasets. We report the error between the learnt and ground truth policies using two metrics: KL divergence and 0-1 action prediction loss. Note that the policy corresponds to $|S||X|$ number of probability distributions, one corresponding to each agent state, i.e., $(\boldsymbol{s},\boldsymbol{x})$-tuple. Hence, the composite KL divergence is obtained by first computing the KL divergence between each corresponding pair of learnt and ground truth distributions and then taking an (unweighted) average. To compute 0-1 loss, we compare the modes of corresponding probability distribution from the learnt and ground truth policies.

For the unsupervised algorithms, the identity of learned $\boldsymbol{x}$ may not correspond to the true $\boldsymbol{x}$; hence, we first compute the best correspondence between true and learned latent state labels to

Table 1: Policy Error: KL-Divergence and 0-1 Action Prediction Loss.

	RoadWorld		BoxWorld: *static*		BoxWorld: *dynamic*		TeamBoxWorld: *dynamic*	
	KL div.	0 − 1 loss	KL div.	0 − 1 loss	KL div.	0 − 1 loss	KL div.	0 − 1 loss
BC (U)	1.73 ± 0.35	0.60 ± 0.00	2.21 ± 0.17	0.18 ± 0.01	2.11 ± 0.09	0.18 ± 0.01	1.39 ± 0.20	0.25 ± 0.02
InfoGAIL	8.56 ± 2.26	0.62 ± 0.14	6.77 ± 0.55	0.17 ± 0.01	7.04 ± 0.94	0.21 ± 0.07	9.70 ± 0.37	0.38 ± 0.05
AMM (U)	**1.37 ± 0.12**	0.50 ± 0.05	0.77 ± 0.01	**0.01 ± 0.00**	0.84 ± 0.03	**0.01 ± 0.00**	1.18 ± 0.02	**0.11 ± 0.01**
FAMM (U)	1.39 ± 0.24	**0.48 ± 0.08**	**0.65 ± 0.01**	**0.01 ± 0.00**	**0.65 ± 0.01**	**0.01 ± 0.00**	**0.66 ± 0.03**	0.13 ± 0.01
p-value	*< 0.01*	*< 0.05*	*< 0.01*	*< 0.01*	*< 0.01*	*< 0.01*	*< 0.01*	*< 0.01*
AMM (Semi.)	0.85 ± 0.00	0.28 ± 0.00	0.69 ± 0.01	0.01 ± 0.00	0.71 ± 0.00	**0.00 ± 0.00**	1.03 ± 0.01	**0.09 ± 0.01**
FAMM (Semi.)	**0.81 ± 0.11**	**0.19 ± 0.04**	**0.60 ± 0.00**	**0.00 ± 0.00**	**0.60 ± 0.00**	**0.00 ± 0.00**	**0.62 ± 0.00**	0.11 ± 0.01
p-value	*0.11*	*< 0.01*	*< 0.01*	*< 0.01*	*< 0.01*	*< 0.01*	*< 0.01*	*< 0.05*
BC (S)	**0.31 ± 0.60**	0.14 ± 0.27	2.66 ± 0.53	0.16 ± 0.01	2.28 ± 0.25	0.15 ± 0.02	1.16 ± 0.13	0.28 ± 0.01
AMM (S)	0.75 ± 0.00	**0.00 ± 0.00**	**0.59 ± 0.00**	**0.00 ± 0.00**	0.66 ± 0.00	0.01 ± 0.00	0.81 ± 0.00	**0.04 ± 0.00**
FAMM (S)	0.75 ± 0.00	**0.00 ± 0.00**	**0.59 ± 0.00**	**0.00 ± 0.00**	**0.59 ± 0.00**	**0.00 ± 0.00**	**0.59 ± 0.00**	**0.04 ± 0.00**
p-value	*0.09*	*0.77*	*< 0.01*	*< 0.01*	*< 0.01*	*< 0.01*	*< 0.01*	*< 0.01*

Table 2: Latent Factor Decoding Error on test data comprising of 100 expert demonstrations.

	RoadWorld	BoxWorld: *static*	BoxWorld: *dynamic*	TeamBoxWorld: *dynamic*
InfoGAIL	0.41 ± 0.13	0.48 ± 0.01	0.53 ± 0.03	0.57 ± 0.05
AMM (U)	0.41 ± 0.08	0.70 ± 0.03	0.76 ± 0.02	0.79 ± 0.02
FAMM (U)	**0.31 ± 0.05**	**0.31 ± 0.04**	**0.32 ± 0.08**	**0.46 ± 0.03**
p-value	*0.25*	*< 0.01*	*< 0.01*	*< 0.01*
AMM (Semi.)	0.07 ± 0.00	0.73 ± 0.02	0.75 ± 0.01	0.69 ± 0.02
FAMM (Semi.)	**0.05 ± 0.09**	**0.24 ± 0.00**	**0.23 ± 0.01**	**0.34 ± 0.01**
p-value	*0.08*	*< 0.01*	*< 0.01*	*< 0.01*
AMM (S)	0.07 ± 0.00	0.55 ± 0.00	0.74 ± 0.00	0.61 ± 0.00
FAMM (S)	**0.01 ± 0.00**	**0.22 ± 0.02**	**0.24 ± 0.00**	**0.30 ± 0.00**
p-value	*< 0.01*	*< 0.01*	*< 0.01*	*< 0.01*

compute the policy learning metrics for the unsupervised setting. For InfoGAIL, the correspondence is computed by reporting the lowest policy error across all possible correspondences. For AMM and FAMM, we use the correspondence that minimizes the decoding error on the test set. The ability to accurately decode the latent state is essential for using the learned model for prediction of *Agent*'s behavior. Hence, for algorithms that model latent decision factors (InfoGAIL, AMM, and FAMM), we also report on their ability to decode latent decision factors on an unsupervised test set of 100 execution sequences. The decoding performance is quantified using normalized Hamming distance between the true and decoded latent state sequences.

7.4 Results

We evaluate performance of each learning algorithm across three problem settings: unsupervised (U), semi-supervised (Semi.), and supervised (S). For each domain, we provide an identical training set of 100 unlabeled sequences of agent's task execution. Additionally, for the semi-supervised and supervised settings, we provide labels of latent decision factors for 25% and 100% of the training data. Results for this experimental condition, averaged across 5 trials, are summarized across Tables 1–2. In the supplementary material, we provide additional results for a weighted version of the policy learning metrics. To assess whether the effect of learning algorithm is statistically significant, we also report *p-values* computed using the non-parametric Kruskal-Wallis test for each of the three problem settings. As reported in Tables 1–2, the effect of learning algorithm is statistically significant across all domains except RoadWorld.

7.4.1 Results: Static Latent Decision Factors. Our first set of experiments consider the simpler setting of time-invariant latent decision factors, a strong modeling assumption made by the baseline algorithm InfoGAIL. For these experiments, we utilize the RoadWorld and BoxWorld domains. In this setting, the learning algorithms still need to learn in presence of (multiple) latent decision factors; however, within each trajectory, the latent values are constant. The training data is generated by the following specification of the ground truth transition function, $T_x = \mathbb{1}(\boldsymbol{x} = \boldsymbol{x}')$; rest of the ground truth parameters are described in Sec. 7.1.

Effect of modeling latent states. We first discuss the results for the unsupervised setting. Among our baselines, the unsupervised variant of BC does not model latent decision factors and hence cannot capture dependence of agent behavior based on $\boldsymbol{x}$. Due to this limitation, we observe that BC(U) does not perform as well as the unsupervised learning approaches based on AMM and FAMM, thereby highlighting the importance of explicitly modeling latent states. Somewhat surprisingly, BC(U) either performs comparably to or outperforms InfoGAIL.

Effect of modeling multiple latent states. Next, we discuss the comparison of AMM and FAMM across different level of supervision. Note that AMM models latent decision factors using a scalar state, while FAMM represents them as a factored state (vector). In the fully supervised case, AMM and FAMM policy learning techniques perform comparably on both policy learning and latent state decoding metrics. However, in the unsupervised and semi-supervised problem settings, FAMM policy learning either performs comparably to or outperforms AMM policy learning. These trends suggest that the factored representation is beneficial while computing the local updates of the MFVI-based policy learning algorithm.

Ability to learn from semi-supervision. For the semi-supervised setting, we only compare against Bayesian approaches; to our knowledge, existing deep imitation learning techniques do not consider the setting of semi-supervised policy learning. We observe that despite requiring only a quarter amount of annotation effort, semi-supervised Bayesian learning performs comparably to supervised learning for both AMM and FAMM across metrics and domains. This is especially encouraging, as the setting of semi-supervised learning is of high interest in practical applications. The annotation of cognitive states are challenging to obtain; hence, approaches that can learn from little supervision are desirable. At the same time, semi-supervision allows a human expert to guide the learning and improve performance by judicious of any available human resources.

7.4.2 Results: Dynamic Latent Decision Factors. Next, we evaluate the learning algorithms for the more general setting of time-varying latent decision factors in BoxWorld and TeamBoxWorld. In this challenging setting, the learning algorithms additionally need to segment each trajectory based on the latent decision factors. To generate data for these experiments, we design a ground truth T_x that models time-varying latent decision factors. Similar to the static case, we utilize 100 execution sequences for training, out of which 25% are annotated for the semi-supervised case.

Ability to learn from semi-supervision. The trends observed for the case of static latent states are also reflected in the case of dynamic latent states. In particular, explicit modeling of factored latent states remains useful. Further, the proposed semi-supervised learning techniques perform comparably to their fully supervised counterparts. As evidenced by performance of BC (supervised) relative to FAMM (semi-supervised), however, supervision alone is insufficient to improve learning performance. To effectively utilize available semi-supervision, joint inference of the latent policy and the time-varying factored latent states is necessary.

Effect of modeling latent state dynamics. While training InfoGAIL, we observe that the roll out trajectories obtained using learned policy showed erratic behaviours of redundantly re-visiting some states. This is reflected in its high policy error (KL-divergence from the expert policy) relative to that of approaches based on AMM and FAMM. We posit that this occurs because of interplay of three reasons: first and most importantly, InfoGAIL is designed for considering only one time-invariant latent decision factor; second, learning for any algorithm including InfoGAIL is not augmented using reward augmentation as reward is unavailable; third, due to the difficulty in hyper-parameter tuning which can affect the convergence of Generator-Discriminator pair to a great extent. We reiterate that while there are hyper-parameters in FAMM, they do not require extensive tuning.

8 CONCLUDING REMARKS

This work presents FAMM, a generative model to represent the behavior of other agents in presence of multiple, dynamic, latent decision factors. For FAMM, we provide Bayesian policy learning algorithms from partially observable demonstrations of agent behavior. To evaluate this challenging setting of model learning, we contribute three synthetic domains and conduct a suite of numerical experiments. While performance of the unsupervised approach is poor, we observe that our semi-supervised approach performs as well as the fully supervised approach with only a quarter of annotations. This result is especially encouraging for learning generative models of humans and other agents in practice, where learning typically needs to be done with limited annotation effort. We also note that relative to deep imitation learning methods (such as BC and InfoGAIL), our framework can learn without significant hyperparameter tuning. Our experiments also highlight the need for several directions for further investigation, ranging from analyzing the benefits and limitations of the metrics used to assess the problem of modeling other agents to novel extensions of our approach.

Limitations and Future Directions. The key limitation of our work is the lack of evaluation with behavioral data derived from human users. While using synthetic data is useful for validating the proposed algorithm (as the ground truth models and latent states are synthetically generated and, hence, can be used for benchmarking), we acknowledge that the behavioral data derived from human users remains the gold standard for evaluating the proposed techniques. Encouraged by the obtained results of this paper and to confirm the performance of our approach in real world tasks, we are developing a human subject experiment to collect a novel dataset of task-oriented human behavior along with annotations of multiple latent states. Another avenue of immediate interest is to develop a framework that integrates the desirable features of our Bayesian approach with that of deep generative modeling, thereby enabling label-efficient policy learning in domains with both high-dimensional and latent states.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive feedback. This research was sponsored 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. The authors also thank COMEXUS and the Fulbright Program for supporting Liubove Orlov-Savko through the Fulbright-García Robles fellowship for Graduate Studies.

REFERENCES

[1] Stefano V Albrecht, Jacob W Crandall, and Subramanian Ramamoorthy. 2016. Belief and truth in hypothesised behaviours. *Artificial Intelligence* 235 (2016), 63–94.

[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] 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.

[4] Gianluca Borghini, Laura Astolfi, Giovanni Vecchiato, Donatella Mattia, and Fabio Babiloni. 2014. Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness. *Neuroscience & Biobehavioral Reviews* 44 (2014), 58–75.

[5] Evan A Byrne and Raja Parasuraman. 1996. Psychophysiology and adaptive automation. *Biological psychology* 42, 3 (1996), 249–268.

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

[7] Zoubin Ghahramani and Michael I Jordan. 1997. Factorial hidden Markov models. *Machine learning* 29, 2 (1997), 245–273.

[8] 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.

[9] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016), 4565–4573.

[10] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *Journal of Machine Learning Research* 14, 5 (2013).

[11] Boris Ivanovic, Karen Leung, Edward Schmerling, and Marco Pavone. 2020. Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach. *IEEE Robotics and Automation Letters* 6, 2 (2020), 295–302.

[12] Wonseok Jeon, Seokin Seo, and Kee-Eung Kim. 2018. A Bayesian Approach to Generative Adversarial Imitation Learning.. In *NeurIPS*. 7440–7450.

[13] Ece Kamar, Ya'akov Gal, and Barbara J Grosz. 2009. Incorporating helpful behavior into collaborative planning. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Springer Verlag.

[14] Arthur F Kramer, Leonard J Trejo, and Darryl G Humphrey. 2018. Psychophysiological measures of workload: Potential applications to adaptively automated systems. In *Automation and human performance: Theory and applications*. CRC Press, 137–162.

[15] Przemyslaw A Lasota, Terrence Fong, Julie A Shah, et al. 2017. *A survey of methods for safe human-robot interaction*. Now Publishers.

[16] Yunzhu Li, Jiaming Song, and Stefano Ermon. 2017. Infogail: Interpretable imitation learning from visual demonstrations. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 3815–3825.

[17] Holger Lüdtke, Barbara Wilhelm, Martin Adler, Frank Schaeffel, and Helmut Wilhelm. 1998. Mathematical procedures in data recording and processing of pupillary fatigue waves. *Vision research* 38, 19 (1998), 2889–2896.

[18] Anahita Mohseni-Kabir, Charles Rich, Sonia Chernova, Candace L Sidner, and Daniel Miller. 2015. Interactive hierarchical task learning from a single demonstration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. 205–212.

[19] 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.

[20] 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.

[21] Raja Parasuraman, Toufik Bahri, John E Deaton, Jeffrey G Morrison, and Michael Barnes. 1992. *Theory and design of adaptive automation in aviation systems*. Technical Report. Catholic Univ of America Washington DC cognitive science lab.

[22] Raja Parasuraman and Victor Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human factors* 39, 2 (1997), 230–253.

[23] Zhiqian Qiao, Jing Zhao, Jin Zhu, Zachariah Tyree, Priyantha Mudalige, Jeff Schneider, and John M Dolan. 2020. Human driver behavior prediction based on urbanflow. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 10570–10576.

[24] Mark W Scerbo. 2018. Theoretical perspectives on adaptive automation. In *Automation and human performance: Theory and applications*. CRC Press, 37–63.

[25] Herbert A Simon. 1990. Bounded rationality. In *Utility and probability*. Springer, 15–18.

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

[27] 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.

[28] 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.

[29] Walter W Wierwille, SS Wreggit, CL Kirn, LA Ellsworth, and RJ Fairbanks. 1994. *Research on vehicle-based driver status/performance monitoring; development, validation, and refinement of algorithms for detection of driver drowsiness. final report.* Technical Report.

[30] Ryan W Wohleber, Gerald Matthews, Gregory J Funke, and Jinchao Lin. 2016. Considerations in physiological metric selection for online detection of operator state: A case study. In *International conference on augmented cognition*. Springer, 428–439.