Improving Competence via Iterative State Space Refinement

Connor Basich[1], Justin Svegliato[1], Allyson Beach[1], Kyle H. Wray[2], Stefan Witwicki[2], Shlomo Zilberstein[1]

***Abstract*— Despite considerable efforts by human designers, accounting for every unique situation that an autonomous robotic system deployed in the real world could face is often an infeasible task. As a result, many such deployed systems still rely on human assistance in various capacities to complete certain tasks while staying safe. Competence-aware systems (CAS) is a recently proposed model for reducing such reliance on human assistance while in turn optimizing the system's global autonomous operation by learning its own competence. However, such systems are limited by a fixed model of their environment and may perform poorly if their a priori planning model does not include certain features that emerge as important over the course of the system's deployment. In this paper, we propose a method for improving the competence of a CAS over time by identifying important state features missing from the system's model and incorporating them into its state representation, thereby refining its state space. Our approach exploits information that exists in the standard CAS model and adds no extra work to the human. The result is an agent that better predicts human involvement, improving its competence, reliability, and overall performance.**

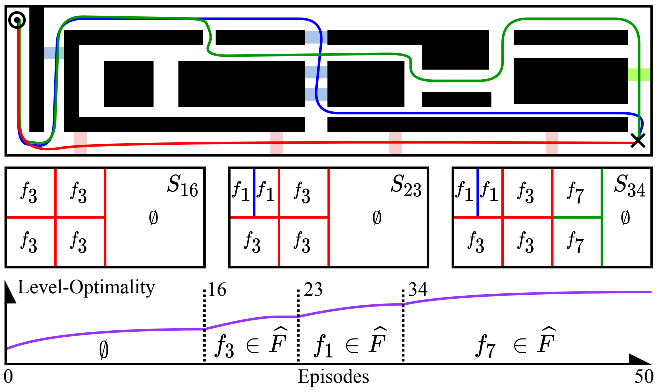


Fig. 1: An illustration of our approach in a navigation task. Colored blocks represent different types of doors. Each colored path corresponds to the optimal path under a different granularity of the state space representation. As features are identified and added to the state representation, the system can better learn to exploit human assistance and take paths that better match its own competence.

I. Introduction

Autonomous robotic systems are increasingly being deployed in highly complex, unstructured domains where they are expected to operate reliably on the order of months or even years. Familiar examples include autonomous navigation [1], extraterrestrial exploration [2], personal assistance [3], and other applications [4]. Despite the recent progress made in artificial intelligence and robotics, it is still often infeasible to account for every environmental feature relevant to all possible scenarios that the system may face in a single planning model. Consequently, many autonomous systems still rely on human assistance in various capacities to successfully accomplish their tasks [5].

Competence-aware systems (CAS) has been recently proposed as a planning framework to reduce unnecessary reliance on human assistance by learning their own competence and accounting for it during planning [6]. A CAS is a semi-autonomous system [7] comprised of an autonomous agent and a human authority that can operate in one of a number of distinct levels of autonomy each corresponding to some set of constraints on its autonomous operation. For example, a service robot may not be allowed to operate fully autonomously when an unrecognized obstacle blocks its path but may request that the human operator supervises its execution, ready to take control. Necessary reliance on a human authority stems from limited competence on the part of the autonomous system, but unnecessary reliance stems from limited information.

While the CAS model enables a semi-autonomous system to optimize its autonomy over time, it still relies on an approximate model of its domain for planning. Generally, the high-level features used by the planning model are designed carefully by experts prior to deployment, and do not represent the full breadth of information available to the agent from sensors or external sources. This could be due to underspecified objectives or dynamics [8], particularly in a deployed setting where the use and environment may vary with each system, or environmental complexity that is too computationally challenging to fully specify in a single planning model [9]. This presents a particular issue in the context of CAS and human-agent interactions when the initial fixed model that the agent is deployed with does not align well with that of the human interacting with the system. The CAS may not have the representational power to properly discriminate between feedback provided by the human during the course of operation, leading to low competence, poor performance, and an over-reliance on the human as the system cannot properly learn its own competence in parts of the state space.

In this paper, we propose a method for providing a CAS the ability to improve its competence over time by increasing the granularity of its state representation through

Supported by the National Science Foundation grants IIS-1724101, IIS-1954782, and DGE-1451512, and the Alliance Innovation Lab Silicon Valley.

[1]College of Information and Computer Sciences, University of Massachusetts, Amherst, MA, USA. Emails: {cbasich, jsvegliato, ajbeach, shlomo}@cs.umass.edu

[2]Alliance Innovation Lab Silicon Valley, Santa Clara, CA, USA. Emails: {kyle.wray,stefan.witwicki}@nissan-usa.com

online model updates. As illustrated in Figure 1, this process leads to a more nuanced drawing of the boundaries between regions of the state space with different levels of competence. Our approach leverages the CAS model in two critical ways. First, it exploits information available in a standard CAS model in the form of human feedback, adding no extra work to the human, to identify where new features should be added. Second, it exploits properties of the human-agent interaction to avoid needing to ever alter the transition function or reward function, modifying only the state space directly, allowing the process to be done in a fully automated way. We prove that when the human and agent share a sufficient state representation, this approach is guaranteed to terminate at which point no additional features will be added. Additionally, we provide both simulated experiments and experiments with a physical mobile robot that validate the efficacy of our approach.

II. Background on Competence Awareness

We begin by reviewing the primary model used in this approach. A competence-aware system (CAS) is a semi-autonomous system that operates in and plans for multiple levels of autonomy, each of which is associated with different restrictions on autonomous operation and distinct forms of human involvement [6]. A CAS combines three models—a *domain model*, an *autonomy model*, and a *feedback model*—into a single decision-making framework. For an in-depth background we direct the reader to earlier work [6].

The *domain model* (*DM*) represents the environment in which the agent operates with a stochastic shortest path (SSP) problem. An SSP is represented by the tuple $\langle S, A, T, C, s_0, G\rangle$ where S is a set of states (feature vectors), A is a set of actions, $T : S \times A \to \Delta^{|S|}$ is a transition function, $C : S \times A \to \mathbb{R}^+$ is a positive cost function, $s_0 \in S$ is an initial state, and $G \subset S$ is a set of goal states.

The *autonomy model* (*AM*) represents the extent of autonomous operation the system can perform in any situation. *AM* is represented by the tuple $\langle \mathcal{L}, \kappa, \mu\rangle$. $\mathcal{L} = \{l_0, ..., l_n\}$ is the set of *levels of autonomy*, where each l_i corresponds to a set of constraints on the system's autonomous operation. $\kappa : S \times A \to \mathcal{P}(\mathcal{L})$ is the *autonomy profile* that indicates the allowed levels of autonomy when performing action $a \in A$ in state $s \in S$. κ may model external constraints that could represent legal or ethical considerations [10]. κ constrains the full policy space so that the system can never follow a policy that violates κ. $\mu : S \times \mathcal{L} \times A \times \mathcal{L} \to \mathbb{R}^+$ is the *cost of autonomy* that represents the cost of operating at level $l' \in \mathcal{L}$ when taking action $a \in A$ in the state $s \in S$ given that the agent just operated in level $l \in \mathcal{L}$.

The *human feedback model* (*HM*) models the autonomous agent's current knowledge and belief about its interactions with the human agent. *HM* is represented by the tuple $\langle \Sigma, \lambda, \rho, \tau\rangle$. Σ is the set of feedback signals the agent can receive from the human. $\lambda : S \times \mathcal{L} \times A \times \mathcal{L} \to \Delta^{|\Sigma|}$ is the *feedback profile* that represents the probability of receiving signal σ when performing action $a \in A$ at level $l' \in \mathcal{L}$ given that the agent is in state $s \in S$ and just operated in level $l \in \mathcal{L}$ and is typically represented using a machine learning model. $\rho : S \times \mathcal{L} \times A \times \mathcal{L} \to \mathbb{R}^+$ is the *human cost function* that represents the cost to the human of performing action $a \in A$ at level $l' \in \mathcal{L}$ given that the agent is in state $s \in S$ and just operated in level $l \in \mathcal{L}$. $\tau : S \times \mathcal{L} \times A \times \mathcal{L} \to \Delta^{|S|}$ is the *human state transition function* that represents the probability of the human taking the agent to state $s' \in S$ when the agent attempted to perform action $a \in A$ in state $s \in S$ but the human took over control.

Definition 1. *A* ***competence-aware system*** $\mathcal{S}$ *is represented by the tuple* $\langle \overline{S}, \overline{A}, \overline{T}, \overline{C}, \overline{s}_0, \overline{G}\rangle$*, where:*

- $\overline{S} = S \times \mathcal{L}$ *is a set of factored states, each comprised of a domain state* $s \in S$ *and a level of autonomy* $l \in \mathcal{L}$.
- $\overline{A} = A \times \mathcal{L}$ *is a set of factored actions, each comprised of a domain action* $a \in A$ *and a level of autonomy* $l \in \mathcal{L}$.
- $\overline{T} : \overline{S} \times \overline{A} \to \Delta^{|\overline{S}|}$ *is a transition function with* $T : S \times A \to \Delta^{|S|}$, $\lambda : \overline{S} \times \overline{A} \to \Delta^{|\Sigma|}$, *and* $\tau : S \times A \to \Delta^{|S|}$.
- $\overline{C} : \overline{S} \times \overline{A} \to \mathbb{R}^+$ *is a cost function with* $C : S \times A \to \mathbb{R}^+$, $\mu : \overline{S} \times \overline{A} \to \mathbb{R}$, *and* $\rho : \overline{S} \times \overline{A} \to \mathbb{R}^+$.
- $\overline{s}_0 \in \overline{S}$ *is the initial state* $\overline{s}_0 = \langle s_0, l\rangle$ *for some* $l \in \mathcal{L}$.
- $\overline{G} \subset \overline{S}$ *is the set of goal states.*

The objective is to find an optimal policy $\pi^*_\kappa \in \Pi$ that minimizes the value function V^π subject to the condition that, for every state $\overline{s} = (s, l') \in \overline{S}$, the policy $\pi(\overline{s})$ *never* indicates an action $\overline{a} = (a, l) \in A$ for which the level of autonomy l is *not* allowed for state s and action a by $\kappa(s, a)$.

Intuitively, the *competence* of a CAS for executing action a in state $\overline{s}$ is the most cost-effective level of autonomy given perfect knowledge of the authority's feedback model. If the authority is likely to deny or override the system autonomously carrying out action a, the competence will likely be low. Similarly, if the authority is likely to allow the action to be carried out autonomously, we expect the competence to be high, although this is not formally required. We emphasize that this is a definition on the human-agent system as a whole, and not simply the autonomous agent, as the competence is directly affected not just by the technical capabilities of the agent, but also the human authority's perception of the agent's capabilities.

Definition 2. *Let* $\lambda^{\mathcal{H}}$ *be the stationary distribution of feedback signals that the human authority follows. The* ***competence*** *of CAS* $\mathcal{S}$*, denoted* $\chi_{\mathcal{S}}$*, is a mapping from* $\overline{S} \times A$ *to the optimal (least-cost) level of autonomy given perfect knowledge of* $\lambda^{\mathcal{H}}$*. Formally:*

$$\chi_{\mathcal{S}}(\overline{s}, a) = \operatorname*{argmin}_{l \in L} Q(\overline{s}, (a, l); \lambda^{\mathcal{H}})$$

where $Q(\overline{s}, (a, l); \lambda^{\mathcal{H}})$ *is the expected cumulative reward when taking action* $\overline{a} = (a, l)$ *in state* $\overline{s}$ *conditioned on knowing the human authority's feedback distribution,* $\lambda^{\mathcal{H}}$.

A CAS is *level-optimal* in state $\overline{s}$ if the system operates at its competence in that state under its current optimal policy. The higher the system's level-optimality is, the better it exploits the capabilities of the authority as well as its own.

Definition 3. *A CAS $\mathcal{S}$ is **level-optimal** if $\pi^*(\overline{s}) = (a, \chi_{\mathcal{S}}(\overline{s}, a))$ $\forall \overline{s} \in \overline{S}$. $\mathcal{S}$ is otherwise **γ-level-optimal** for $\gamma \in [0, 1)$ if this holds for a γ fraction of states.*

III. Improving Competence

While recent work shows that a CAS will converge to be level-optimal in the limit under certain assumptions [6], the result makes no claim about the level of competence of the CAS. In particular, if a CAS is missing the features necessary to correctly represent its domain in a way that aligns with the human authority, even upon convergence to level-optimality, its competence may be quite low.

Hence, the objective of this work is ***to provide a CAS with the ability to improve its competence over time*** by leveraging the existing feedback available to the agent. Formally, this means that the CAS will increase both its competence—the optimal level of autonomy to use—and the total level-optimality of the system. Our approach relies on the assumption that an authority is *ϵ-consistent* in their feedback; that is, given the same $(\overline{s}, \overline{a})$, the feedback signal returned will be the same with probability at least ϵ. We address practical considerations regarding this assumption later in the paper. Under this assumption, the system can identify cases where feedback appears inconsistent or random, indicating a potential *missing* feature—a feature used by the authority to make their decisions, but currently not used by the system. By identifying the most likely feature or combination of features missing from the system's domain model, the agent can update its model to better align with the internal model of the authority, enabling it to improve its overall competence by discriminating between situations where it can and cannot act in any given level of autonomy.

Let $\mathcal{S} = \langle AM, FM, DM \rangle$ be a competence-aware system. The *complete feature space* available to $\mathcal{S}$, e.g., from its sensors or other external sources, can be partitioned into an *active feature space* that is used by $\mathcal{S}$ and an *inactive feature space* that is not yet used by $\mathcal{S}$. As $\mathcal{S}$ receives additional feedback over time, $\mathcal{S}$ will learn to exploit inactive features in order to more effectively align with the features used by the human authority.

Definition 4. *Given the **complete feature space (FS)** $F = \{F_1, F_2, ..., F_n\}$ available to $\mathcal{S}$, the **active FS** is denoted as $\hat{F} \subseteq F$, and the **inactive FS** as $\breve{F} = F \setminus \hat{F}$.*

In order to ensure that level-optimality can be improved, we assume that the human authority produces feedback that remains *consistent* during the operation of $\mathcal{S}$. This means that when the agent performs the same action in the same state at the same level of autonomy, the authority provides, with high probability, the same feedback each time. Observing violations of this assumption is central to our approach.

Definition 5. *Let $F^{\mathcal{H}} \subset F$ be the set of features used by the human authority, $\mathcal{H}$, and let $\overline{S}_{\mathcal{H}} = F_1^{\mathcal{H}} \times \cdots \times F_{|\mathcal{H}|}^{\mathcal{H}} \times \mathcal{L}$. The **ground truth feedback function** is a deterministic mapping $f : \overline{S}_{\mathcal{H}} \times \overline{A} \to \Sigma$. $\mathcal{H}$ is **perfectly consistent** if $\lambda^{\mathcal{H}}(f(\overline{s}, \overline{a})|\overline{s}, \overline{a}) = 1$ $\forall \overline{s} \in \overline{S}, \overline{a} \in \overline{A}$. If $\lambda^{\mathcal{H}}(f(\overline{s}, \overline{a})|\overline{s}, \overline{a}) \geq \epsilon$ for $\epsilon \in (0, 1)$ $\forall \overline{s} \in \overline{S}, \overline{a} \in \overline{A}$, then $\mathcal{H}$ is **ϵ-consistent**.*

A state $\overline{s}$ is *indiscriminate* if it is missing information, leading to the feedback profile having a low predictive confidence. Intuitively, the condition states that for at least one action, there is *no* feedback signal that the system predicts with sufficiently high probability.

Definition 6. *A state $\overline{s}$ is **indiscriminate** if there exists at least one action $\overline{a} \in \overline{A}$ where $\forall \sigma \in \Sigma$, $\lambda(\sigma|\overline{s}, \overline{a}) \leq 1 - \delta$ where $\mathcal{H}$ is ϵ-consistent and $\delta \in (1 - \epsilon, 1 - \frac{1}{|\Sigma|})$.*

We call δ the *discrimination slack*. The lower δ is set, the higher the predictive confidence needed for $(\overline{s}, \overline{a})$ to not be considered indiscriminate.

A *discriminator* is any subset of features available to the agent but not currently used by the agent in its planning model which could help the agent better predict the authority's feedback. For example, consider a state that represents a closed door. With no additional features, the agent may perceive having received equal approvals and disapprovals from the human authority, while the human was in fact disallowing the robot from opening doors it felt were too heavy for the robot to open without damaging itself. By including features that represent the door's size, the robot can then observe this disambiguation, allowing it to better plan around interactions with a door.

Definition 7. *A **discriminator** is any subset of $\breve{F}$ which, if added to $\hat{F}$, will improve the accuracy of λ by at least α, for some $\alpha \in (0, 1)$.*

The larger that α is set, the stricter the requirement is on including a new feature. The methodology for selecting discriminators is well explored in the feature selection literature and not the focus of this paper; standard approaches include mRMR [11], JMI [12], and correlation-based methods [13].

IV. State Space Refinement

Algorithm 1 presents the pseudocode of our approach for improving the competence of a CAS via iterative partitioning of the state space by adding new features to the state representation over time. The algorithm first identifies the current set of indiscriminate states (Lines 1–6). To avoid labeling sparsely sampled state-action pairs as indiscriminate through chance, we limit the process to only consider a state-action pair if the probability of having observed all labeled instances of that element in the existing dataset $\mathcal{D}$, referred to in Algorithm 1 as $Obs(\mathcal{D}(\overline{s}, \overline{a}))$, conditioned on the assumption that there exists a true correct feedback signal returned with probability at least ϵ by the human for every state-action pair, is at least some threshold p_ϵ (Line 4). Next, the algorithm samples an indiscriminate state from the set (Line 9) and identifies the most likely discriminators for that state using any standard feature selection technique, in our case mRMR [11] (Line 11). For each potential discriminator, a new feedback profile is trained using a portion of the full dataset with the discriminator temporarily added to the active feature set (Lines 12–13). The discriminator that leads to the best performing feedback profile, in our case the highest Matthews coefficient, is selected for validation (Line 14).

Algorithm 1: Single–Step State Space Refinement

```
Input: A CAS $\mathcal{S}$, dataset $\mathcal{D}$, slack $\delta$, and threshold $M$
Result: An updated CAS $\mathcal{S}$
1  $\overline{S}^* \leftarrow \{\}$
2  for $\overline{s} \in \mathcal{S}.getStates()$ do
3      for $\overline{a} \in \mathcal{S}.getActions()$ do
4          if $\max_{\sigma \in \Sigma} \lambda(\sigma|\overline{s}, \overline{a}) \leq 1 - \delta$ and
5          $\max_{\sigma \in \Sigma} \Pr[Obs(\mathcal{D}(\overline{s}, \overline{a}))|\sigma$ is ground truth$] < p_\epsilon$
6              $\overline{S}^* \leftarrow \overline{S}^* \cup \{\overline{s}\}$
7  if $\overline{S}^* = \emptyset$
8      return $\mathcal{S}$
9  $\overline{s}^* \sim \overline{S}^*$
10 $\mathcal{D}_{train}, \mathcal{D}_{val} \leftarrow Split(\mathcal{D})$
11 $D \leftarrow mRMR(\mathcal{D}_{train}, \breve{F}, \overline{s})$
12 for $d \in D$ do
13     $\lambda_d \leftarrow train(\hat{F}_1 \times \cdots \times \hat{F}_{|\hat{F}|} \times d, \mathcal{D}_{train})$
14 $d^* = \text{argmax}_{d \in D}\, evaluate(\lambda_d, \mathcal{D}_{val})$
15 if $validate(d^*, \mathcal{S})$ is True
16     $\hat{F} \leftarrow \hat{F} \cup d^*$
17     $\mathcal{S}' \leftarrow update(\mathcal{S})$
18 return $\mathcal{S}'$
```

If validation is successful, the discriminator is added to the active feature set and the system is updated (Lines 15–17).

In this work, we make a few key assumptions. First, we assume that the initial transition function provided in the domain model is *sufficiently correct* for any scenario where the agent is allowed, under κ, to act autonomously. We aim to improve the robustness of deployed systems where accounting for every scenario a priori is infeasible, but where the scenarios that are considered a priori are well-designed. While we only investigate improving the competence of a CAS by iteratively refining the state space, it may also be possible to increase the competence by updating the transition function directly and replanning as the human authority improves *its* understanding of the agent's capabilities.

Second, we assume that the authority has a sufficient understanding of the agent's capabilities to both prevent the execution of an action that the agent cannot perform successfully and also provide consistent feedback. We make this assumption for two reasons. First, there are different ways to improve the authority's understanding of the system's capabilities so that it has the appropriate trust [14], or reliance, on the system. These include pre-deployment training, standardized feedback criteria, and expert knowledge of the system. Second, recognizing potential failure and handling fault recovery are separate areas of active research that are orthogonal to what we examine in this paper.

Under these assumptions, *we do not need to update the domain model's transition or reward functions directly at any point*. It suffices for the agent to be able to discriminate between actions that it has the competence to perform autonomously and actions that require human involvement.

A natural question is whether in the process of adding a discriminator so as to make some indiscriminate states discriminate, we will as an unintended by-product make some discriminate state indiscriminate.

Remark 1. *Adding a discriminator will never cause a discriminate state to become indiscriminate.*

While not obvious a priori, this remark is trivially true. Observe that any given discriminate state will either be affected by the discriminator or it will not. If it is not affected, the feedback profile for the state will not change. If the state is affected, then the initial state in question by definition no longer exists. More importantly, we want to ensure that every state is eventually properly discriminated given a sufficient set of features.

The following theorem states that if every feature that the human uses to determine their feedback is available to the robot, then there must be a point in time at which the robot has fully discriminated all states, and no state will become indiscriminate past that point.

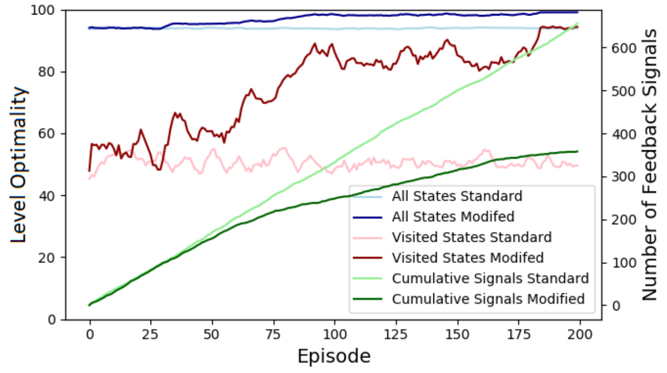
Theorem 1. *Let $F^{\mathcal{H}}$ be the set of features used by the human to determine their feedback, I_t be the number of indiscriminate states at time t, and λ_t be the feedback profile at time t. If $F^{\mathcal{H}} \subseteq F$, $\{\lambda_t\}$ converges in distribution, no $(\overline{s}, \overline{a}) \in \overline{S} \times \overline{A}$ is starved, $\mathcal{H}$ is ϵ-consistent and there is positive discrimination slack, then there exists some $t^* > 0$ for which it holds that $I_{t'} = 0$ for all $t' > t^*$.*

Proof. First, observe that as $F^{\mathcal{H}} \subseteq F$, if there is a point at which $F^{\mathcal{H}} \subseteq \hat{F}$, then because the sequence $\{\lambda_t\}$ converges in distribution $\lim_{t \to \infty} \Pr(|\lambda_t - \lambda^{\mathcal{H}}| > \gamma) = 0$ $\forall \gamma > 0$. Hence, there exists some $t^* > 0$ for which $\Pr(|\lambda_t - \lambda^{\mathcal{H}}| > \delta) = 0$ at which point it is clear that no state will be indiscriminate under δ. Consequently, for the claim to not hold, it must be the case that for every $t > 0$, $F^{\mathcal{H}} \setminus (F^{\mathcal{H}} \cap \hat{F}) \neq \emptyset$. Pick such a t, sufficiently large, for which there is an indiscriminate state $\overline{s} \in \overline{S}$. There is some subset, $G \subseteq F^{\mathcal{H}} \setminus (F^{\mathcal{H}} \cap \hat{F})$, which is a discriminator of s. As this holds for all $t > 0$ and $\overline{s} \in \overline{S}$, we either reach a satisficing t^* where $F^{\mathcal{H}} \setminus (F^{\mathcal{H}} \cap \hat{F}) \neq \emptyset$, and hence are done, or where $F^{\mathcal{H}} \subseteq \hat{F}$ which contradicts our assumption. □

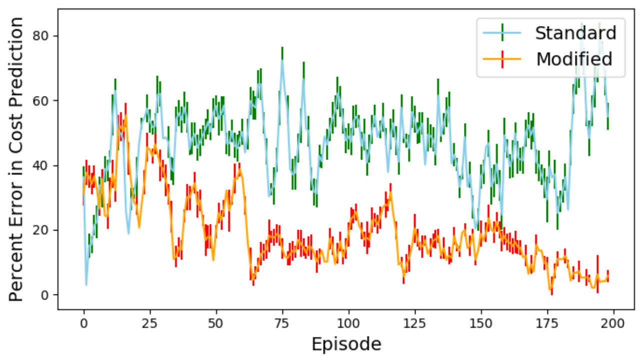
V. Experiments

We implemented our framework in a simulated domain in which a mobile robot is tasked with delivering packages to various rooms on a college campus. An illustration of the map used in our experiments can be seen in Figure 3. To accomplish its task, the agent may need to navigate two types of obstacles: crosswalks and doors.

Domain Model Domain states are specified by location, heading, and obstacle information when relevant. The full set of features available to the system for crosswalk obstacles include the traffic condition, the visibility condition, and whether the street is one way or two ways. The total available features available to the system for door obstacles include whether they are open or closed, their size, color, and whether they have a handle or not (and hence whether they can be pushed open). Finally, the current hour in the day is also available to the agent as a global state feature.



(a) Level optimality of the modified versus standard CAS, along with cumulative number of feedback signals. *All states* refers to the entire state space. *Visited states* are states entered at least once during any episode.



(b) Percent error in expected cost of reaching the goal as computed when solving for the agent's policy against the actual incurred cost. The values plotted are the mean and standard deviation over 10 trials.

Fig. 2: Comparison between a modified CAS and a standard CAS. Tasks (start and goal state) were randomly selected each episode and provided to both CAS models. Each episode was simulated 10 times.

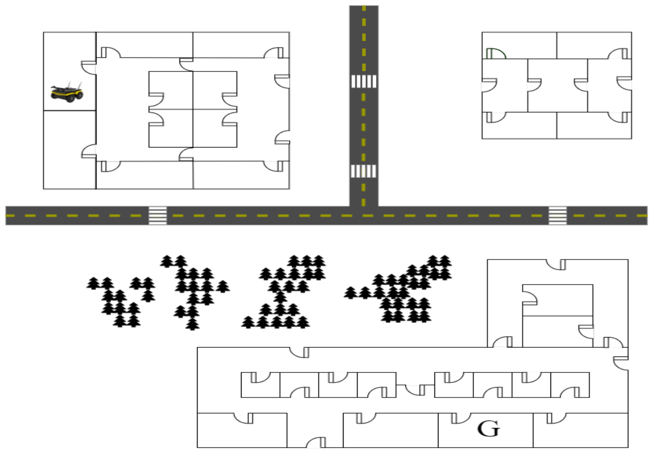


Fig. 3: The domain used for simulations.

However, as the state space of an SSP grows exponentially in the size of its feature space, and it is not known a priori which features will be utilized by the human in providing their feedback, only a small subset of these features are initially used by the planning model. Specifically, the CAS is initialized with traffic conditions at crosswalk obstacles and at door obstacles whether the obstacle is open or closed. To increase its competence, the CAS must learn which additional features it needs to add to its model to correctly predict human feedback with high confidence. The additional causal features that the human authority uses to determine their feedback are the visibility condition of the crosswalk, the pedestrian traffic (which is unavailable to the agent directly but can be indirectly modeled by the time of day), and both the size and opening mechanism of doors. To obfuscate things further for the system, non-causal features are intentionally correlated with certain causal features. For example, in one building, the door color is perfectly consistent with the door size, and with one exception, the street type is consistent with the visibility condition. This tests whether our approach uses causal features over correlative features.

Autonomy and Feedback Models We consider an agent with four levels of autonomy: *no autonomy* (l_0), *verified autonomy* (l_1), *supervised autonomy* (l_2), and *unsupervised autonomy* (l_3). l_0 requires the human authority to complete the action for the agent. l_1 requires the agent to query the authority for explicit approval prior to executing its intended action. l_2 requires the human authority to be available in a supervisory capability while the agent executes the action, with the ability to override and take over control if necessary. l_3 is fully autonomous. The agent can receive the following four feedback signals: *approval* ($\oplus$), *disapproval* ($\ominus$), *override* ($\oslash$), and *none* ($\emptyset$).

Initially, λ is uniformly distributed and $\kappa(s, a) = \{l_0, l_1\}$ for all actions in states with obstacles. For states with no obstacles, the system is allowed to operate in unsupervised autonomy. In all experiments, the human authority is 0.95-consistent, and uniformly random over the possible feedback signals in the other 5%. To avoid generalization between actions which had no relationship, we maintained a separate feedback dataset for each action $a \in A$. The data features used in each dataset were the level of autonomy and the state features currently used by the system.

We implemented our feedback profile λ as a GA^{2}M [15], which works well on low dimensional feature spaces where pairwise feature interactions are important. We trained our λ on all features and pairwise feature tensors using the native grid search function for hyperparameter tuning [16]. In Algorithm 1, we set $\delta = 0.95$, and the train/validation split is 75%/25%. Our validation step (Line 15) entails the feedback profile produced by D^* having a Matthews correlation coefficient above 0.5 that is also at least 0.05 higher than the existing feedback profile's coefficient; the intuition here being that the new feedback profile should be statistically better than random, and at least marginally better than the existing feedback profile. More stringent requirements may inadvertently fail to identify features that are missing but which only affect a very small portion of the state space (in the most extreme case, a single state).

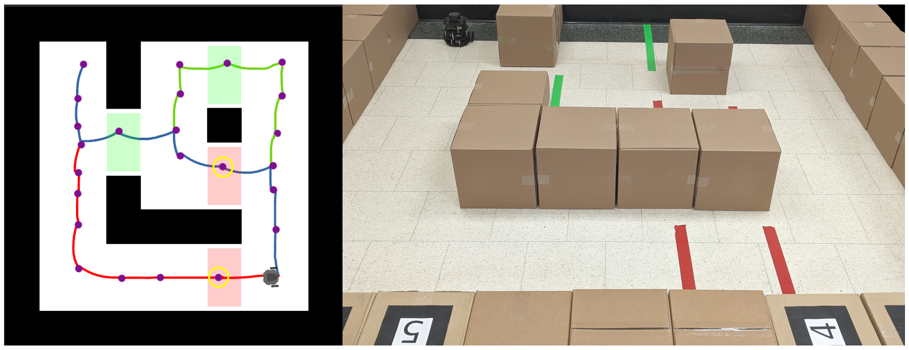


Fig. 4: *Left*: The paths taken by the robot. Yellow circles represent human intervention or assistance. *Right*: The robot in the domain. *Green* denotes doors the human will approve the robot to open and *red* denotes doors the human disallows the robot from opening.

Results We compared the performance of a modified CAS that can add to its active feature set over time and a standard CAS that must rely only on its initial set of state features. We ran 200 episodes where, in each episode, the start and goal states were randomly sampled uniformly from the set of rooms across the campus. The modified CAS was able to identify and add all missing causal features (four of them); in particular, the size of the doors, their opening mechanism (if they could be pushed open or not), the visibility condition at crosswalks, and finally the time of day which provides information on the level of pedestrian traffic which is not able to be measured directly by the system.

We first observe in Figure 2a that the modified CAS reached roughly 100% level-optimality across all states, and over 90% level-optimality across all visited states. However, we see that the level-optimality across all states for the standard CAS did not change significantly over the course of the simulation in either case. The level-optimality starts off high (roughly 93%) because in most states the agent is able to act autonomously a priori; it is only in the small set of crucial, or risk-sensitive, states with obstacles that the agent must learn its competence. Furthermore, the modified CAS was able to diminish its reliance on the human operator much more effectively. By episode 200, the number of feedback signals received has begun to flatten, whereas in the standard CAS case it continues to grow linearly at a constant rate. In other words, the modified CAS is able to perform more optimally with fewer feedback signals from the human.

In Figure 2b, we plot the percent error in cost prediction: the error in the expected cost of reaching the goal compared to the true cost incurred in the process of doing so. The modified CAS (orange) approaches close to 0 percent error, demonstrating that it has learned to predict the human's feedback with high accuracy across the domain, whereas the standard CAS continues to have large error throughout.

Robot Experiment We also implemented our approach on a TurtleBot3 mobile robot. The results can be seen in Figure 4 where we depict the three distinct paths taken by the robot over time. The robot starts in the top left corner, as depicted, and its goal is to reach the bottom left corner. The *red* line represents the first path taken by the robot—the robot, knowing nothing about the differences between doors, takes the shortest path to the goal requiring it to request human aid to open a pull door. The *blue* line represents the path taken by the robot after introducing the new feature `doortype`—the robot now travels through the first push door, knowing that the bottom pull door is not openable by it. However, the agent still attempts to open the nearest door after, a pull door, as it has not determined with high confidence if `doortype` is the determining factor, or if it simply is disallowed from opening the bottom door specifically. After receiving a disapproval, it requests human help. The *green* line represents the final path—the robot has now identified that `doortype` is the determining factor, and takes the path that only interacts with push doors, despite being longer than the other two.

VI. Related Work

There has been prior interest in exploring model incompleteness in sequential decision making under uncertainty. A longstanding line of work has investigated Markov decision processes (MDPs) that are imprecisely known a priori. Instead of a singular function used to model domain transition or reward dynamics, these dynamics are modeled as distributions over sets of functions [17], [18], [19]. This notion was more recently generalized under the terminology *uncertain MDPs* (UMDPs) [20], [21], [22], [23]. As in our work, these models are intended to deal with domains where producing a fully precise and accurate model of the domain is hard or infeasible a priori, or where the dynamics are non-stationary. This work, however, is concerned with MDPs in which the representation of the state space itself is incomplete.

Learning from human interaction has been an active area in recent years, including *learning from human input* [24], [25], [26] and *learning from demonstration* [27], [28], [29]. Our work is related to the former, in which an agent must learn parameters of its model through interactions with a human (e.g. advice or questions). However, our work is unique as it exploits unexpected inconsistencies in a model learned from human input to address the problem of an incomplete state representation.

Feature selection is an area that has been extensively studied in particular for its benefits in machine learning and data analysis [30]. Feature selection has most commonly been applied to data analysis where the data is high dimensional, warranting a need to select the features that optimize the objective function while preserving the underlying structure of the data [31]. It has also been used in reinforcement learning as a technique to more efficiently determine the most relevant sensory features for capturing the transition dynamics of the domain [32] and for identifying a sparse set of features for learning robotic manipulation skills that adapt to different objects [33].

VII. Conclusion

We present a method for enabling competence-aware systems to improve their competence and level-optimality over the course of their deployment by refining the state representation so as to better predict human feedback. Our method works by identifying *indiscriminate states*, states that are likely missing information needed to accurately predict human feedback, and adding the most likely *discriminators* for such states, features that improve the feedback profile, to the state representation. This enables the system to better discriminate the feedback received from the human authority. We evaluate our method on a simulated delivery robot, demonstrating that our approach enables the CAS to correctly identify missing state features and significantly improve its level-optimality and plan accuracy when compared to an unmodified CAS. Furthermore, we demonstrate the viability of the approach on a physical mobile robot.

There are several areas of future work. First, we have only considered the presence of a *human* authority agent; however, in general we believe that our approach extends to situations where the authority is another automated system or a high quality sensor that can be queried. Second, our method currently relies on the human having a sufficient understanding of the system's capabilities, which may not always be valid. To address this, we are investigating how new feedback signals that specifically indicate a lack of knowledge or confidence on the part of the human may be used to prompt the system to gather additional information.

References

[1] S. Liu, L. Li, J. Tang, S. Wu, and J.-L. Gaudiot, "Creating autonomous vehicle systems," *Synthesis Lectures on Computer Science*, 2017.

[2] Y. Gao and S. Chien, "Review on space robotics: Toward top-level science through space exploration," *Science Robotics*, 2017.

[3] N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrova, J. Young, *et al.*, "The STRANDS project: Long-term autonomy in everyday environments," *IEEE Robotics and Automation Magazine*, 2017.

[4] L. Kunze, N. Hawes, T. Duckett, M. Hanheide, and T. Krajník, "Artificial intelligence for long-term robot autonomy: A survey," *IEEE Robotics and Automation Letters (RA-L)*, 2018.

[5] S. Zilberstein, "Building strong semi-autonomous systems," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[6] C. Basich, J. Svegliato, K. H. Wray, S. Witwicki, J. Biswas, and S. Zilberstein, "Learning to optimize autonomy in competence-aware systems," in *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2020.

[7] K. H. Wray, L. E. Pineda, and S. Zilberstein, "Hierarchical approach to transfer of control in semi-autonomous systems," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[8] S. Saisubramanian, S. Zilberstein, and P. Shenoy, "Optimizing electric vehicle charging through determinization," in *International Conference on Automated Planning and Scheduling SPARK Workshop*, 2017.

[9] J. Svegliato, K. H. Wray, S. J. Witwicki, J. Biswas, and S. Zilberstein, "Belief space metareasoning for exception recovery," in *IEEE/RSJ International Conference on Intelligent Robotic Systems (IROS)*, 2019.

[10] J. Svegliato, S. B. Nashed, and S. Zilberstein, "Ethically compliant sequential decision making," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[11] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2005.

[12] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: a unifying framework for information theoretic feature selection," *Journal of Machine Learning Research*, 2012.

[13] B. Senliol, G. Gulgezen, L. Yu, and Z. Cataltepe, "Fast correlation based filter with a different search strategy," in *IEEE International Symposium on Circuits and Systems (ISCIS)*, 2008.

[14] K. A. Hoff and M. Bashir, "Trust in automation: Integrating empirical evidence on factors that influence trust," *Human Factors*, 2015.

[15] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker, "Accurate intelligible models with pairwise interactions," in *ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*, 2013.

[16] D. Servén and C. Brummitt, "pyGAM: Generalized Additive Models in Python," Mar. 2018. [Online]. Available: https://zenodo.org/record/1208724#.YP8uyi1h1MM

[17] J. K. Satia and R. E. Lave Jr., "Markovian decision processes with uncertain transition probabilities," *Operations Research*, 1973.

[18] C. C. White III and H. K. Eldeib, "Markov decision processes with imprecise transition probabilities," *Operations Research*, 1994.

[19] C. C. White III and H. K. El-Deib, "Parameter imprecision in finite state, finite action dynamic programs," *Operations Research*, 1986.

[20] K. Chen and M. Bowling, "Tractable objectives for robust policy optimization," in *International Conference on Neural Information Processing Systems (NeurIPS)*, 2012.

[21] A. Ahmed, P. Varakantham, Y. Adulyasak, and P. Jaillet, "Regret based robust solutions for uncertain Markov decision processes," in *International Conference on Neural Information Processing Systems (NeurIPS)*, 2013.

[22] Y. Adulyasak, P. Varakantham, A. Ahmed, and P. Jaillet, "Solving uncertain MDPs with objectives that are separable over instantiations of model uncertainty," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[23] A. Ahmed, P. Varakantham, M. Lowalekar, Y. Adulyasak, and P. Jaillet, "Sampling based approaches for minimizing regret in uncertain MDPs," *Journal of Artificial Intelligence*, 2017.

[24] M. T. Rosenstein and A. G. Barto, "Supervised actor-critic reinforcement learning," in *Handbook of Learning and Approximate Dynamic Programming*, J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, Eds. IEEE Press, 2004.

[25] H. B. Suay and S. Chernova, "Effect of human guidance and state space size on interactive reinforcement learning," in *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2011.

[26] L. Torrey and M. Taylor, "Teaching on a budget: Agents advising agents in reinforcement learning," in *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2013.

[27] J. A. Clouse, "On integrating apprentice learning and reinforcement learning," Ph.D. dissertation, University of Massachusetts, 1996.

[28] S. Chernova and M. Veloso, "Interactive policy learning through confidence-based autonomy," *Journal of Artificial Intelligence Research*, 2009.

[29] M. Rigter, B. Lacerda, and N. Hawes, "A framework for learning from demonstration with minimal human effort," *IEEE Robotics and Automation Letters (RA-L)*, 2020.

[30] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, 1997.

[31] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[32] C. Diuk, L. Li, and B. R. Leffler, "The adaptive k-meteorologists problem and its application to structure learning and feature selection in reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2009.

[33] O. Kroemer and G. S. Sukhatme, "Feature selection for learning versatile manipulation skills based on observed and desired trajectories," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4713–4720.