Safe Explicable Planning

Akkamahadevi Hanni, Andrew Boateng, Yu Zhang

Arizona State University
ahanni@asu.edu, aoboaten@asu.edu, yzhan442@asu.edu

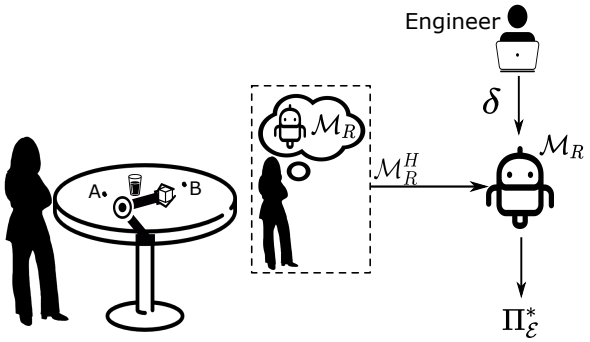
Abstract

Human expectations arise from their understanding of others and the world. In the context of human-AI interaction, this understanding may not align with reality, leading to the AI agent failing to meet expectations and compromising team performance. Explicable planning, introduced as a method to bridge this gap, aims to reconcile human expectations with the agent's optimal behavior, facilitating interpretable decision-making. However, an unresolved critical issue is ensuring safety in explicable planning, as it could result in explicable behaviors that are unsafe. To address this, we propose *Safe Explicable Planning (SEP)*, which extends the prior work to support the specification of a safety bound. The goal of SEP is to find behaviors that align with human expectations while adhering to the specified safety criterion. Our approach generalizes the consideration of multiple objectives stemming from multiple models rather than a single model, yielding a Pareto set of safe explicable policies. We present both an exact method, guaranteeing finding the Pareto set, and a more efficient greedy method that finds one of the policies in the Pareto set. Additionally, we offer approximate solutions based on state aggregation to improve scalability. We provide formal proofs that validate the desired theoretical properties of these methods. Evaluation through simulations and physical robot experiments confirms the effectiveness of our approach for safe explicable planning.

Introduction

Significant strides have been made in advancing the capabilities of AI agents in recent years, from operating in isolated environments to being deployed in environments surrounded by humans. Examples of such agents include Starship's food delivery robots, Amazon's Astro household assistants, Bear Robotics' hospitality robots, and Waymo's autonomous vehicles, among many others. As technologies evolve, these AI agents are poised to become our indispensable partners. It is imperative for AI to learn from human-human interaction where aligning an agent's behavior with others' expectations is a key to such social interaction.

Explicable planning (Zhang et al. 2017; Kulkarni et al. 2016; Hanni and Zhang 2021) is an existing framework addressing human expectations in decision-making. It operates under the assumption that humans form their expectations of an agent's behavior based on their perception of the agent and the environment ($\mathcal{M}_R^H$), which may deviate from the reality captured by the agent's model ($\mathcal{M}_R$) (see Fig. 1(b)). In the original formulation, the objective is to find a plan that closely resembles the human's expected plan, as measured by an explicability metric, while simultaneously minimizing a plan cost metric through a linearly weighted sum of the two metrics. To address explicable behavior generation in stochastic domains, (Gong and Zhang 2022) define a similar objective within a learning framework under Markov Decision Processes (MDPs). However, a key drawback of these methods is the lack of consideration of a bound on the suboptimality of the solution under the ground-truth model (i.e., $\mathcal{M}_R$). This is due to the fact that the trade-off between cost and explicability metrics (at different scales) is governed by a hyper-parameter, referred to as the reconciliation factor by (Zhang et al. 2017). Consequently, generating an explicable behavior may overly compromise the cost in the ground-truth model, leading to potentially unsafe behaviors.[1]



(a) Motivating example (b) Problem setting

Figure 1: The agent uses the ground-truth model - $\mathcal{M}_R$, an estimation of the human's understanding of it - $\mathcal{M}_R^H$, and a bound - δ, to generate safe explicable policies $\Pi_{\mathcal{E}}^*$.

Let us further illustrate the need for safe explicable planning (SEP) via a motivating scenario. Imagine a human

[1]Here, the underlying assumption is that safety is negatively correlated with the cost metric under $\mathcal{M}_R$. Future work will explore other safety criteria, such as considering behavior variability.

working alongside a robot manipulator. The task is for the robot to hand over a box to the human, with two potential locations for placement: '*A*' and '*B*' (depicted in Fig. 1(a)). Location '*A*' is closer to the human but involves a small risk of tipping over a water cup nearby. When the cup is empty, this risk is negligible. In such cases, the preferred action would be for the robot to place the box at '*A*' to align with the human's expectation. However, when the cup is not empty, tipping it over could lead to hazards like electric shocks that incur significant costs in the robot's model. Hence, the preferred action would be for the robot to place the box at '*B*'. When such a subtle difference (i.e., whether the cup is empty) is not apparent from the human's perspective (based on $\mathcal{M}_R^H$), the robot may indistinguishably prioritize conforming to the human's expectations, leading to unsafe behavior despite seeming more explicable. In SEP, the robot's behaviors are constrained by a cost bound in the ground-truth model, ensuring it never chooses an unsafe behavior. SEP prioritizes safety without sacrificing explicability, which can mitigate the risk by preventing hazardous outcomes in human-robot interaction scenarios.

In our Safe Explicable Planning (SEP) approach, we build upon the following assumptions to focus on the planning challenges. First, we assume that the agent has access to its model ($\mathcal{M}_R$) and the human's belief of its model ($\mathcal{M}_R^H$), or simply, the human's model. A similar assumption has been made in prior research on explicable planning (Kulkarni et al. 2016; Hanni and Zhang 2021) and explainable decision-making (Chakraborti et al. 2019). In practice, the human's model may be provided by experts or acquired from human feedback, which has been explored in previous studies (Christiano et al. 2017; Ibarz et al. 2018; Holmes et al. 2004; Juba and Stern 2022). Second, we assume that the human is a rational observer, i.e., a behavior with a higher expected return in the human's model is more expected. Hence, the most expected behavior can be generated by computing the optimal behavior in the human's model. This assumption allows us to equate the problem of maximizing explicability to maximizing the expected return of a policy under the human's model (that is modeled as an MDP). Such an assumption of human rationality is a common simplification in cognitive science and artificial intelligence research, such as in (Baker, Saxe, and Tenenbaum 2011).

We formulate Safe Explicable Planning (SEP) under MDP by defining the objective as maximizing the expected return in the human's model, subject to a constraint in the agent's model. This problem formulation generalizes the consideration of multiple objectives (White 1982) to also consider multiple domain models. The solution to this problem yields a Pareto set of policies for which exact solvers are often intractable. To address this challenge, we propose an action-pruning technique to reduce the policy space significantly. Subsequently, we introduce a novel tree search method that efficiently explores the remaining policies to identify the Pareto set. We formally prove that this search method is sound and complete. Additionally, we introduce a greedy search method for situations where any policy from the Pareto set suffices. Finally, we devise approximate solutions for both search methods using state aggregation, addressing scalability in large domains. We evaluate our methods across several domains via simulation and physical robot experiments, demonstrating their effectiveness for SEP. Furthermore, we conduct ablation studies to analyze the benefits of our pruning techniques, validating their effectiveness in reducing computational costs while generating the desired behaviors.

Related Work

Interest in explainable decision-making has been growing with the aim of creating AI agents whose behaviors are understandable to humans (Chakraborti et al. 2019; Chakraborti, Sreedharan, and Kambhampati 2020; Fox, Long, and Magazzeni 2017). We may broadly classify methods in this area into two categories: those that generate interpretable behaviors (implicit methods) and those that communicate to explain behaviors (explicit methods). Our work belongs to the former category. Researchers have approached implicit methods for explainable decision-making from various but related perspectives, such as generating behaviors that are considered legible (Dragan and Srinivasa 2013), predictable (Dragan, Lee, and Srinivasa 2013), transparent (MacNally et al. 2018), explicable (Zhang et al. 2017), etc. The relationships among these concepts are reviewed comprehensively by (Chakraborti et al. 2019). Our work extends explicable planning by addressing a critical gap in applying such methods to real-world scenarios.

Our problem formulation of SEP shares some key features with the constrained-criterion-based formulation of safe reinforcement learning (RL) (Garcıa and Fernández 2015), which is inherently a Constrained Markov Decision Process (CMDP) (Altman 2021). Similar problem formulations have been proposed for continuous spaces and applied to risk-bounded motion planning (Huang et al. 2019). In these prior works, safety is encoded by constraining the expected cost under some designated cost function while maximizing the agent's reward function under the same model. In SEP, similarly, safety is encoded by constraining the expected return under the agent's reward function. SEP operates under the assumption that safety directly correlates with the expected return in the agent's model, following the intuition that unsafe behaviors would result in low returns. Our formulation can readily accommodate a CMDP (with a single constraint) by aligning the two different models (except for the reward functions) and substituting the robot's reward function in the safety constraint with the cost function.

A distinctive challenge in formulating SEP under CMDP arises from the presence of two different MDP models. Specifically, besides featuring two different reward functions, we must explore a more general setting in SEP that also features two different domain dynamics and discount factors. This additional complexity makes the existing solution methods for CMDP inapplicable to SEP. Take, for instance, the linear programming (LP) based approach for CMDP (Altman 1994). This method defines the LP objective using an occupation measure for different state-action pairs, which is a function of the transition model and the discount factor. However, when dealing with the two different models in SEP, applying the LP solution introduces dis-

crepancies between the occupation measure utilized in the objective and that employed in the constraint. Consequently, resolving these two sets of variables is nontrivial. Similar arguments can be made about the other solution methods.

The objective considered in SEP also bears a similarity to that in Multi-Objective Markov Decision Processes (MOMDP) (White 1982), as SEP must consider the expected return under both the agent's and human's model. MOMDPs, introduced for multiple objectives under the same MDP (refer to the review paper by (Roijers et al. 2013)), typically aim to optimize a vector of expected returns for those objectives to derive a Pareto set of policies or to derive a single policy through linear scalarization of those objectives. Approaches, including but not limited to (Wakuta and Togawa 1998; Russell and Zimdars 2003; Barrett and Narayanan 2008; Van Moffaert, Drugan, and Nowé 2013), are examples of these methods. However, to handle different models, MOMDP methods must produce multiple vectors of expected returns, each derived for a different model due to the difference in domain dynamics. Optimizing these vectors simultaneously poses a significantly greater challenge than optimizing a single vector in traditional MOMDPs.

In lexicographic ordered MOMDPs (Gábor, Kalmár, and Szepesvári 1998; Wray, Zilberstein, and Mouaddib 2015; Pineda, Wray, and Zilberstein 2015), one objective is optimized before the other in a predefined order. (Wray, Zilberstein, and Mouaddib 2015) bears close connections to our work and has influenced the action pruning technique outlined in our paper. However, despite the merits, these methods often focus on computational efficiency and do not guarantee the solution's optimality. In addition, it is unclear how to extend them to handle objectives under different models.

Previous studies have explored solving multiple MDPs (Singh and Cohn 1997; Buchholz and Scheftelowitsch 2019), focusing on identifying a policy that maximizes a combined or weighted sum of objectives, thus reducing it to a single objective optimization problem. While these methods may appear comparable to ours, they can yield policies that breach safety bounds or exhibit poor quality in the human's model. This drawback stems from their inability to explicitly account for safety constraints, a gap that we address in our work.

Problem Formulation

In safe explicable planning, there are two models at play: $\mathcal{M}_R$ and $\mathcal{M}_R^H$. We formulate these models as discrete Markov Decision Processes (MDPs). An MDP is represented by a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$ where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{T}(s'|s,a)$ is a transition function, $\mathcal{R}$ is a reward function, and γ is a discount factor. We assume $\mathcal{M}_R$ and $\mathcal{M}_R^H$ share the same state space $\mathcal{S}$ and action space $\mathcal{A}$, but differ in other parameters. Specifically, $\mathcal{M}_R$ incorporates the true domain dynamics $\mathcal{T}_R$, the engineered reward function $\mathcal{R}_R$, and the engineered discount factor γ_R whereas $\mathcal{M}_R^H$ incorporates the human's belief about the domain dynamics $\mathcal{T}_R^H$, human's belief about the reward function $\mathcal{R}_R^H$, and human's belief about the discount factor γ_R^H. This is reasonable when humans and AI agents coexist in a shared workspace and possess certain shared understanding of the environment. Relaxing such an assumption incurs separate technical challenges (e.g., hierarchical models) that will be deferred to future work.

We work with the set of all stationary deterministic policies Π, where $\forall \pi \in \Pi$, $\pi : \mathcal{S} \mapsto \mathcal{A}$. An agent's optimal policy maximizes the expected return in the agent's model and is given by $\pi^* = \arg\max_\pi \mathbb{E}^\pi_{\mathcal{T}_R}\left[\sum_{t=0}^{\infty} \gamma_R^t r_R(t)\right]$. We define a safe behavior as any behavior with a return within a bound of the agent's optimal return. Similar criteria have been used in safe RL (Garcıa and Fernández 2015; Moldovan and Abbeel 2012). More formally, a policy π is considered safe or feasible if its return satisfies the following condition:

$$\mathbb{E}^\pi_{\mathcal{T}_R}\left[\sum_{t=0}^{\infty} \gamma_R^t r_R(t)\right] \geq \delta \mathbb{E}^{\pi^*}_{\mathcal{T}_R}\left[\sum_{t=0}^{\infty} \gamma_R^t r_R(t)\right], \quad (1)$$

where $\delta \in (0, 1]$ is the designer-specified safety bound. Since execution may start from any state, we require such a condition to hold true under *any state*. It also implies that the condition would hold from any step during execution. These are desirable features of safety critical systems.

In prior work on explicable planning, the objective is to maximize a weighted sum of the return in the agent's model and an explicability metric. Explicability metric has been defined, for example, via plan distances (Kulkarni et al. 2016) in deterministic domains and KL divergence between trajectory distributions (Gong and Zhang 2022) in stochastic domains. In our work, we define the explicability metric simply as the return in the human's model $\mathcal{M}_R^H$. Given that the human user generates expectations from $\mathcal{M}_R^H$, this assumes a rational human observer: the higher the return in the human's model, the more expected the policy is.

Definition 1. Safe Explicable Planning (SEP), given by $\mathcal{P}_\mathcal{E} = \langle \mathcal{M}_R, \mathcal{M}_R^H, \delta \rangle$, is the problem to search for a policy that maximizes the return in $\mathcal{M}_R^H$ subject to a constraint on the return in $\mathcal{M}_R$ under *any state*, or formally:

$$\pi^*_\mathcal{E} = \arg\max_\pi \mathbb{E}^\pi_{\mathcal{T}_R^H}\left[\sum_{t=0}^{\infty} {\gamma_R^H}^t r_R^H(t)\right] \text{ subject to}$$
$$\mathbb{E}^\pi_{\mathcal{T}_R}\left[\sum_{t=0}^{\infty} \gamma_R^t r_R(t)\right] \geq \delta \mathbb{E}^{\pi^*}_{\mathcal{T}_R}\left[\sum_{t=0}^{\infty} \gamma_R^t r_R(t)\right]. \quad (2)$$

The maximization of the expected return above across all states introduces a Pareto set of optimal policies where no policies in this set are strictly dominated by any feasible policy. Briefly, a policy π_1 strictly dominates another policy π_2 if its state values are no smaller in any state, and larger in at least one state. Formally, we denote such a relationship as $\pi_1 \succ \pi_2$, which holds if $\forall s \in \mathcal{S}\ [V^{\pi_1}_{\mathcal{M}_R^H}(s) \geq V^{\pi_2}_{\mathcal{M}_R^H}(s)] \wedge \exists s' \in \mathcal{S}\ [V^{\pi_1}_{\mathcal{M}_R^H}(s') > V^{\pi_2}_{\mathcal{M}_R^H}(s')]$. The Pareto set $\Pi^*_\mathcal{E}$ is then given by:

$$\Pi^*_\mathcal{E} = \{\pi^*_\mathcal{E} \in \Pi_\delta \mid \neg\exists \pi \in \Pi_\delta [\pi \succ \pi^*_\mathcal{E}]\}, \quad (3)$$

where $\Pi_\delta = \{\pi \in \Pi | \forall s \in \mathcal{S}\ [V^\pi_{\mathcal{M}_R}(s) \geq \delta V^{\pi^*}_{\mathcal{M}_R}(s)]\}$ is the set of policies that satisfy the safety bound.

Safe Explicable Planning

In this section, we motivate and discuss our solution methods for SEP. Given the large policy space to search for, we first discuss a technique to reduce the policy space. Since any policy Π_δ may be in the Pareto set, it necessitates the expansion of all policies in Π_δ . We propose an exact method that selectively expands policies in Π_δ to determine the Pareto set $\Pi^*_\mathcal{E}$. Additionally, we discuss a greedy method that expands only a subset of policies in Π_δ , returning a single policy in $\Pi^*_\mathcal{E}$. Finally, we propose approximate solutions via state aggregation, using handcrafted features, to condition similar states to choose the same actions to further scalability in large domains. Complete proofs, evaluation details, and additional discussions are presented in the full version of this paper (Hanni, Boateng, and Zhang 2024).

Policy Space Reduction via Action Pruning

Even though the set Π_δ cannot be obtained directly from the entire policy space Π, we aim to reduce the policy space based on the safety constraint to produce a subset of policies in Π, referred to as $\widetilde{\Pi}$. The challenge here is to ensure that $\widetilde{\Pi} \supseteq \Pi_\delta$ (see Fig. 2(a)).

We achieve this by pruning sub-optimal actions for every state that are guaranteed to violate the constraint. Specifically, let $\mathcal{A}(s)$ be the set of all actions that are available in any state s. The set of actions after pruning is given by:

$$\widetilde{\mathcal{A}}(s)=\{a \in \mathcal{A}(s) | Q^{\pi^*}_{\mathcal{M}_R}(s,a) \geq \delta \max_{a' \in \mathcal{A}(s)} Q^{\pi^*}_{\mathcal{M}_R}(s,a')\}. \tag{4}$$

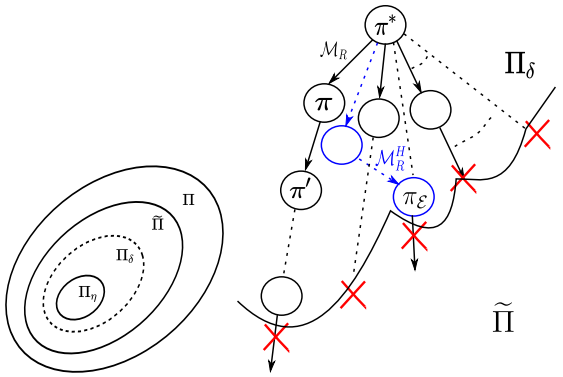
The policy space obtained from the resulting actions in all states is $\widetilde{\Pi}$. Our action pruning technique draws inspiration from (Wray, Zilberstein, and Mouaddib 2015). In their work, to provide a worst-case guarantee under $\mathcal{M}_R$, the authors employ $1-(1-\gamma)(1-\delta)$ instead of δ in Eqn. (4), resulting in a different set of policies, denoted by Π_η. Their pruning condition is more stringent than ours and may result in pruning actions prescribed by certain policies that satisfy the constraint in Eqn (2). Consequently, the guarantee that $\Pi_\eta \supseteq \Pi_\delta$ is lost there (see Fig. 2(a)).

Lemma 1. The set of policies after pruning actions based on Eqn. (4) is a superset of the set of policies that satisfy the constraint in Eqn. (2), i.e., $\widetilde{\Pi} \supseteq \Pi_\delta$.

Proof Sketch: To prove this result, we show that an action pruned in a state per Eqn. (4) is guaranteed to introduce policies that violate the constraint in Eqn. (2) in at least one state. Then, we show the expected return of choosing a pruned action once (in the state it was pruned) and following the optimal policy thereafter, violates the constraint. Hence, any policy that chooses the pruned action for that state will result in violating the constraint.

Policy Descent Tree Search (PDT)

To determine $\Pi^*_\mathcal{E}$, intuitively, we can evaluate every policy in $\widetilde{\Pi}$. However, this would be impractical and proves to be unnecessary. A more efficient strategy involves further reducing $\widetilde{\Pi}$ by expanding policies in a specific order. There



(a) Policy sets (b) PDT vs. PAG

Figure 2: (a) Relationship between different set of policies. (b) PDT vs. PAG on pruned-action space $\widetilde{\Pi}$. The black nodes are expanded by PDT in descending order of state values under $\mathcal{M}_R$. The blue nodes are expanded by PAG in ascending order under $\mathcal{M}^H_R$. Solid lines represent single-action policy updates and dashed links represent multi-action updates.

are two possible search strategies to explore. First, consider initializing the search to the optimal policy in the human's model and perform policy improvement under the agent's objective until the bound is satisfied. Alternatively, consider initializing the search to the optimal policy in the agent's model and perform policy decrement under the agent's objective while simultaneously identifying better policies under the human's objective, until the bound is violated. While the first search strategy is simpler it can lead to missed policies in $\Pi^*_\mathcal{E}$, hence we choose the latter option in our work.

In tree search, we start from an optimal policy in $\mathcal{M}_R$, denoted by π^*, as the root node. The benefit of doing so is that, first, we already know that π^* satisfies the bound under the agent's model as it is the optimal policy in $\mathcal{M}_R$. Second, we can leverage the known state values $V^{\pi^*}_{\mathcal{M}_R}$ to expand policies that have lower state values than that of the parent node, recursively. Since this is the opposite of policy improvement, we refer to it as policy descent. Formally, all descendants of a policy π under single-action policy updates in PDT can be obtained by replacing $\pi(s)$ under *any* state s with an action a that satisfies:

$$Q^{\pi}_{\mathcal{M}_R}(s,a) \leq Q^{\pi}_{\mathcal{M}_R}(s,\pi(s)). \tag{5}$$

Once a branch reaches a policy whose state values no longer satisfy the bound in $\mathcal{M}_R$ (any state suffices), it is pruned as illustrated in Fig. 2(b). The search continues until all branches are explored or pruned while the set of non-dominated policies in $\mathcal{M}^H_R$ are maintained. The algorithm is presented in Alg. 1, which we refer to as PDT+ (includes action pruning). Next, we formally show that such a process returns the Pareto set $\Pi_\mathcal{E} = \Pi^*_\mathcal{E}$.

Lemma 2. Let π and π' be two deterministic policies that differ only by a single action in some state i.e., $\exists s_i \in \mathcal{S}$ $[\pi'(s_i) \neq \pi(s_i)] \wedge \forall s_j \in \mathcal{S} \setminus \{s_i\}$ $[\pi'(s_j) = \pi(s_j)]$ and satisfy $Q^{\pi}_{\mathcal{M}_R}(s_i, \pi'(s_i)) \leq V^{\pi}_{\mathcal{M}_R}(s_i)$. Then, policy π' is a

Algorithm 1: PDT+

Input: $\mathcal{M}_R, \mathcal{M}_R^H, \delta$
$V^*_{\mathcal{M}_R} \leftarrow$ `ValueIteration`$(\mathcal{M}_R)$; retrieve π^*
Compute $\widetilde{\mathcal{A}}(s), \forall s \in S$;
Initialize $\Pi_\mathcal{E} \leftarrow \emptyset$; `fringe.push`$(\pi^*)$;
while `fringe` $\neq \emptyset$ **do**
  $\pi \leftarrow$ `fringe.pop`$()$;
  for a *in* $\widetilde{\mathcal{A}}(s), s \in S$ **do**
    if *Eqn.* (5) *is satisfied* **then**
      $\pi' \leftarrow$ `Modify`$(\pi, \pi(s) = a)$;
      if $\forall s \in S\ [V^{\pi'}_{\mathcal{M}_R}(s) \geq \delta V^{\pi^*}_{\mathcal{M}_R}(s)]$ **then**
        `fringe.push`(π');
        if `nonDominated`$(\pi', \Pi_\mathcal{E}, \mathcal{M}_R^H)$ **then**
          $\Pi_\mathcal{E}$`.update`(π');
return $\Pi_\mathcal{E}$

Algorithm 2: PAG+

Input: $\mathcal{M}_R, \mathcal{M}_R^H, \delta$
$V^*_{\mathcal{M}_R} \leftarrow$ `ValueIteration`$(\mathcal{M}_R)$; retrieve π^*
Compute $\widetilde{\mathcal{A}}(s), \forall s \in S$;
Initialize $\pi_\mathcal{E} \leftarrow \pi^*$; `changed` $\leftarrow true$;
while `changed` **do**
  $V^{\pi_\mathcal{E}}_{\mathcal{M}_R^H} \leftarrow$ `PolicyEvaluation`$(\pi_\mathcal{E}, \mathcal{M}_R^H)$
  `changed` $\leftarrow false$
  for a *in* $\widetilde{\mathcal{A}}(s), s \in S$ **do**
    if *Eqn.* (6) *is satisfied* **then**
      $\pi' \leftarrow$ `Modify`$(\pi_\mathcal{E}, \pi_\mathcal{E}(s) = a)$;
      if $\forall s \in S[V^{\pi'}_{\mathcal{M}_R}(s) \geq \delta V^{\pi^*}_{\mathcal{M}_R}(s)]$ **then**
        Update $\pi_\mathcal{E} \leftarrow \pi'$
        `changed` $\leftarrow true$
return $\pi_\mathcal{E}$

descendant of π in PDT, i.e., policy π' is no better than π, or formally, $\forall s \in \mathcal{S}\ [V^{\pi'}_{\mathcal{M}_R}(s) \leq V^{\pi}_{\mathcal{M}_R}(s)]$.

Proof Sketch: This is an extension of the policy improvement theorem (Sutton and Barto 2018) but in the opposite direction (hence referred to as a policy descent step). First, we introduce a temporary non-stationary policy π'_1 that chooses an action as per π' under the initial state and follows π thereafter. We can show that the return of π'_1 is no better than that of π. We can repeat such a pattern to update π'_1 for the next state and so on, resulting in π' at the end.

Similarly, we can show that a special case of the policy improvement theorem holds when a single action is updated (referred to as a policy ascent step).

Theorem 1. PDT+ returns all Pareto optimal policies in $\Pi^*_\mathcal{E}$.

Proof Sketch: To prove this, we show that there exists a policy descent path from any optimal policy (denoted by π^*) in $\mathcal{M}_R$ (i.e., the root node in PDT) to any Pareto optimal policy (denoted by $\pi^*_\mathcal{E}$) in $\Pi^*_\mathcal{E}$ by induction. When $\pi^*_\mathcal{E}$ differs from π^* in only 1 action, $\pi^*_\mathcal{E}$ must be one of the direct descendants of π^* in PDT as π^* is the optimal policy in $\mathcal{M}_R$. Hence, $\pi^*_\mathcal{E}$ will be expanded by PDT. Assume any policy that differs from π^* in k actions is expanded. When $\pi^*_\mathcal{E}$ differs from π^* in $k+1$ actions, we show that there must exist a policy π that differs from π^* in k out of the $k+1$ actions (hence differing from $\pi^*_\mathcal{E}$ in 1 action) and (by Lem. 2) is no worse than $\pi^*_\mathcal{E}$ under $\mathcal{M}_R$ via proof by contradiction. Consequently, $\pi^*_\mathcal{E}$ must be a descendant of π in PDT. Since π is expanded under our inductive assumption, $\pi^*_\mathcal{E}$ will be expanded. Then by Lem. 1, the conclusion holds.

Policy Ascent Greedy Search (PAG)

In certain situations, it may be unnecessary to compute $\Pi^*_\mathcal{E}$: any policy in the set would suffice. To this end, we introduce a greedy method that only searches through a subset of Π_δ, making it computationally more efficient than PDT.

Similar to PDT, we start with π^* at the root node. However, unlike in PDT where we expand policies that have lower state values in $\mathcal{M}_R$ via single-action policy updates, we expand only a single policy that has higher values in $\mathcal{M}_R^H$ than its parent node via multi-action policy updates (see Fig. 2(b)). Formally, only one descendant of policy π is expanded in PAG, which is obtained by replacing $\pi(s)$ under *each* state s with an action a that satisfies the following condition (similar to a policy improvement step):

$$Q^{\pi}_{\mathcal{M}_R^H}(s, a) \geq Q^{\pi}_{\mathcal{M}_R^H}(s, \pi(s)), \tag{6}$$

where each such state-action update is checked against the constraint in Eqn. (2) (in $\mathcal{M}_R$) incrementally and incorporated only if the constraint is not violated, resulting in a multi-action policy update for $V^{\pi}_{\mathcal{M}_R^H}$.

In PAG, we maintain a single candidate policy $\pi_\mathcal{E}$ as opposed to a set in PDT. The current policy $\pi_\mathcal{E}$ is updated to its descendant π' if at least one of the state-action updates is incorporated. This process is repeated until $\pi_\mathcal{E}$ remains unchanged. The algorithm, referred to as PAG+ (includes action pruning), is presented in Alg. 2.

Theorem 2. PAG+ returns a policy in the Pareto set $\Pi^*_\mathcal{E}$.

Proof Sketch: The PAG search process stops when it can no longer improve or find a policy that is equivalent in values to $\pi_\mathcal{E}$ under $\mathcal{M}_R^H$ while satisfying the safety constraint. This implies that there does not exist a state-action update that implements a policy ascent step under the constraint. However, if $\pi_\mathcal{E} \notin \Pi^*_\mathcal{E}$, there must exist another policy $\pi \in \Pi^*_\mathcal{E}$ that dominates $\pi_\mathcal{E}$, which contradicts with the fact that no policy ascent step exists. Then by Lem. 1, $\pi_\mathcal{E} \in \Pi^*_\mathcal{E}$.

Approximate Solution via State Aggregation

In the worst-case scenario, both PDT+ and PAG+ must explore a number of policies on the order of $|\widetilde{\Pi}|$, which remains exponential. Consequently, directly applying these methods to complex domains proves challenging. Approximate solutions become essential. However, conventional methods relying on function approximation for state value functions to search for optimal policies (Sỳkora 2008; Abel, Hershkowitz, and Littman 2016; Abel et al. 2018; Ferrer-Mestres et al. 2020) are not applicable here, as the search is conducted over the policy space.

We aim to devise an approximate solution that minimizes the number of unique policies to be explored. Inspired by

function approximation, one approach is to condense the state space by grouping together states that exhibit similar action selection tendencies. The similarity of states can be measured using domain-specific features. By conditioning states within the same clusters to select the same actions under any policy with either model, we effectively reduce the state space size and consequently the number of policies. Formally, this process involves introducing a mapping $\Phi : \mathcal{S}_K \mapsto \mathcal{S}$, establishing a one-to-many correspondence from clusters to states, where K denotes the number of clusters. Both PDT and PAG can operate using the aggregated state space (i.e., clusters), treating $\mathcal{S}_K$ as the new state space.

Under the assumption that the states within any cluster are "correlated" in action selection under any given policy, the theoretical guarantees of optimality, completeness, and constraint satisfaction remain intact. Such a situation may occur, for example, when two states are topologically equivalent, such that a reasonable policy should always choose the same action under these states. Investigating the introduction of such states and their impact on guarantees when this assumption does not hold or holds only approximately would be interesting. From this perspective, our approximation method resembles function approximation in Q-learning.

Evaluation

We evaluate our methods across various domains through simulation and physical robot experiments, aiming to achieve three main objectives. First, we compare safe explicable behaviors with optimal behaviors to validate the efficacy of our approach. Second, given that solving SEP involves searching for the optimal policy in the feasible policy space to obtain the Pareto set, we evaluate the efficiency of our proposed methods and compare them with baselines (BF & BF+) that employ brute-force policy search. Notably, our comparisons are against brute-force methods because prior studies discussed in the related work section lack consideration for multiple models or safety bounds (refer to related work). Additionally, we conduct ablation studies for each proposed method to analyze the benefits of pruning actions and our approximate solutions in more complex domains. Third, we conduct physical robot experiments to demonstrate the applicability of our approach to real-world scenarios. Following our naming convention, we append '+' to an algorithm's name to denote the incorporation of our action pruning technique, resulting in a reduced policy space $\widetilde{\Pi}$; a method without '+' must search the original policy space, or Π (refer to Fig. 2(a)). All evaluations were run on a MacBook Pro (16 GB, 3.1 GHz Dual-Core Intel Core i5).

Bound Selection: In our approach, we assume the bound is specified by the designer, based on experience. However, it can often be estimated based on the domain. For instance, consider one of the cliff worlds depicted in Fig. 5 (see below). The optimal return in the agent's model is 94 (i.e., moving along the edge of the cliff to the goal), while the return of the trajectory with the longest detour (i.e., staying as far away from the edge as possible) without falling off the cliff is 90, discount notwithstanding. As unsafe behaviors yield significantly lower returns (in $\mathcal{M}_R$) than the detour, the safety bound can be set to $90/94{=}0.957$, subject to adjustment. Further analyses will be deferred to future work.

Policy Selection: User preferences can serve as a guiding factor to select from the Pareto set $\Pi^*_{\mathcal{E}}$. Alternatively, domain-specific scores may be introduced to aid in the selection process. For instance, policies that are deemed "simpler" may receive higher scores. Example scores are discussed below, wherever applicable.

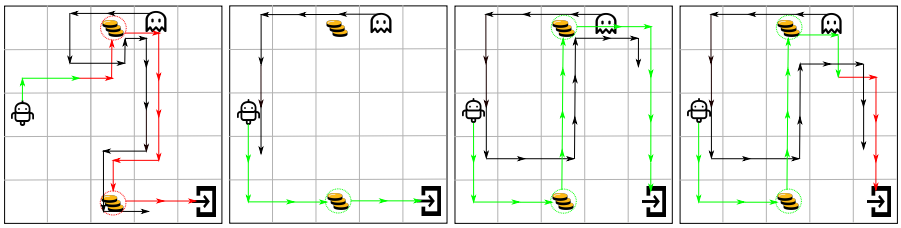
Simulations

Domain Descriptions: 1) *Cliff Worlds (CS & CL)*: The task entails navigating alongside the edge of a cliff to reach the goal, as depicted in Figs. 4 and 5. The ground-truth ($\mathcal{M}_R$) is that the agent can travel alongside the edge without slipping off the cliff. Conversely, the human's belief ($\mathcal{M}^H_R$) is that there is a probability that the agent may slip off from the edge, especially in terrain closer to the cliff, which is more uneven and challenging to traverse. Both models have similarly defined reward functions, $\mathcal{R}_R$ and $\mathcal{R}^H_R$, shown in Figs. 4(a) and 4(b), respectively, for the larger domain. Except that in the smaller domain, the reward at the cliff is -100 and at the goal is 100. We designed a small 4×5 domain (CS) for the exact methods and a large 4×100 domain (CL) for approximate solutions. To apply approximate solutions to CL, we aggregated all non-terminal states based on features such as distance to the cliff and the agent's position in the grid (e.g., along the edge or at the ends) into 10 clusters while retaining the terminal states as they are.

2) *Wumpus World (W)*: The agent's objective is to exit a 5×5 cave while collecting gold coins and avoiding encounters with the wumpus (i.e., entering the same location as the wumpus) (see Fig. 3). The wumpus always moves towards the agent. Each collected gold coin yields a reward of $+30$, and the game ends if the agent encounters the wumpus (-100) or exits the cave ($+100$). The ground-truth ($\mathcal{M}_R$) is that the agent's actions are deterministic, whereas the wumpus's actions are stochastic. Conversely, the human's belief ($\mathcal{M}^H_R$) is that both agents' actions are stochastic. Under this belief, the human perceives it as risky for the agent to approach the wumpus. For approximate solutions, non-terminal states were aggregated into 15 clusters based on features such as the relative direction of the wumpus from the agent and collected gold coin(s).

Results: 1) *Performance Comparison*: Table 1 presents the runtime (except for BF and BF+ due to the large number of policies) and the number of policies expanded (or evaluated) by each method across the three simulation domains. The small cliff world (CS) comprises 16 non-terminal states and 4 actions in each state, resulting in $|\Pi|{=}4^{16}$ policies. The large cliff world (CL) comprises 301 non-terminal states and 4 actions in each state, resulting in $|\Pi|{=}4^{301}$ policies, but upon aggregation, it reduces to 4^{10} policies. The wumpus world (W) comprises 2116 non-terminal states and 4 actions in every state, resulting in $|\Pi|{=}4^{2116}$ policies, but upon aggregation, it reduces to 4^{15} policies. We employ approximate solutions (i.e., PDTs and PAGs on aggregated state space) on CL and W.

We observe that action pruning effectively reduces the



(a) $\tau(\pi^*_{\mathcal{M}_R})$ (b) $\tau(\pi^*_{\mathcal{M}_R^H})$ (c) $\tau(\pi^*_{\mathcal{E}})$ | PDT+ (d) $\tau(\pi^*_{\mathcal{E}})$ | PAG+

Figure 3: Behavior comparison in the wumpus world. Black lines show the trajectories of the wumpus. Red line segments show the parts of the agent's trajectories when the wumpus is in an adjacent cell, and green line segments show when the wumpus is at least two steps away. Presented are the most likely trajectories by (a) the optimal agent's policy, (b) the human's expectation, and the safe explicable policies obtained when $\delta = 0.90$ by (c) PDT+ and (d) PAG+, respectively.

	δ	BF	BF+	PDT		PDT+		$\lvert\Pi^*_{\mathcal{E}}\rvert$	PAG		PAG+	
		#	#	#	RT	#	RT		#	RT	#	RT
CS	1.00	4^{16}	4^{4}	8448	4.8	256	0.3	1	17	0.01	9	0.01
	0.95	4^{16}	$\dot{4}^{9}$	8448	4.8	2816	1.7	1	17	0.01	10	0.01
	0.93	4^{16}	$\dot{4}^{15}$	8448	4.8	7424	4.3	1	17	0.01	17	0.01
	0.90	4^{16}	$\dot{4}^{15}$	169k	102.2	149k	90.3	3	21	0.02	19	0.01
	0.85	4^{16}	$\dot{4}^{15}$	313k	184.4	274k	164.2	3	19	0.01	19	0.01
CL	1.00	4^{10}	4^{2}	368	5.0	16	0.5	1	36	0.5	5	0.1
	0.97	4^{10}	$\dot{4}^{9}$	684	7.9	620	7.1	1	36	0.5	32	0.4
	0.95	4^{10}	$\dot{4}^{9}$	1846	21.0	1677	18.2	3	33	0.5	30	0.4
	0.93	4^{10}	$\dot{4}^{9}$	2254	25.1	2048	22.8	2	30	0.4	27	0.4
	0.90	4^{10}	$\dot{4}^{9}$	2268	25.4	2060	25.0	2	30	0.5	27	0.4
W	1.00	4^{15}	4^{0}	61	0.9	1	0.1	1	25	0.4	1	0.1
	0.97	4^{15}	$\dot{4}^{1}$	61	0.9	3	0.1	1	25	0.4	1	0.1
	0.95	4^{15}	$\dot{4}^{5}$	61	0.9	13	0.3	1	25	0.4	5	0.1
	0.93	4^{15}	$\dot{4}^{5}$	1489	21.7	179	4.1	25	46	0.7	5	0.2
	0.90	4^{15}	$\dot{4}^{5}$	24k	359.1	729	42.1	197	46	0.7	5	0.2

Table 1: Comparison of different methods via the number of policies evaluated (#) and runtime (RT) in minutes. Numbers with a dot are approximate.

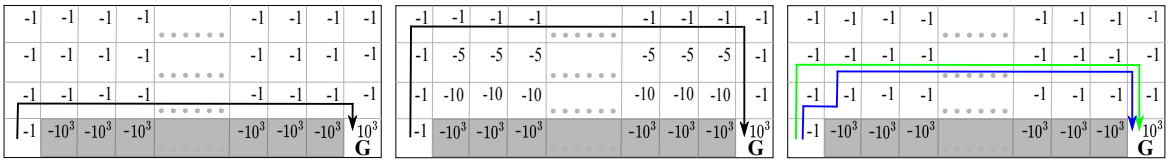
policy space and consequently the number of policies expanded by BF+, PDT+, and PAG+ compared to BF, PDT, and PAG, respectively. The expansion order of policies in PDTs leads to significant additional reduction compared to BF+. With or without action pruning, PAGs expand fewer policies than PDTs as they only need to return a single policy. Lastly, although the number of policies expanded in PDTs increases (for lower δ), it is worth noting that PAGs sometimes expand fewer policies due to their greedy nature.

2) *Behavior Comparison in Cliff Worlds*: The results of the cliff worlds are shown in Figs. 5 (CS) and 4 (CL). Both the small and large domains introduce similar behaviors (shown only in the large domain): the optimal behavior in the agent's model takes the shortest path (Fig. 4(a)), whereas that in the human's model stays as far away from the cliff as possible (Fig. 4(b)). For SEP, Fig. 5 shows all the three policies in the Pareto set obtained given $\delta = 0.90$ in the small domain. Fig. 4(c) shows the most likely trajectories resulting from the policies in the Pareto set obtained given $\delta = 0.95$ in the large domain using the approximate solution. In general, we observe that the safe explicable policies result in trajectories that steer the agent away but not too far from the cliff to satisfy the bound while aligning with the human's expectation. In cliff worlds, to choose from $\Pi^*_{\mathcal{E}}$, we assign higher scores to policies producing simpler behaviors (e.g., fewer turns), it led to choosing the policy producing the green trajectory in Fig. 4(c) and the policy in Fig. 5(a). PAGs, on the other hand, computed different policies in $\Pi^*_{\mathcal{E}}$ (see figures).

3) *Behavior Comparison in Wumpus World*: The results are shown in Fig. 3. Following the optimal policy in the agent's model ($\mathcal{M}_R$), the agent collected both coins while staying near the wumpus before exiting, as shown in Fig. 3(a). Following the optimal policy in the human's model ($\mathcal{M}_R^H$), the agent avoided getting close to the wumpus and collected a single coin before exiting, as shown in Fig. 3(b). When applying SEP under the bound $\delta = 0.90$, PDT+ returns a large Pareto set (see Tab. 1). To select from $\Pi^*_{\mathcal{E}}$, we score policies based on the average distance between the agent and the wumpus throughout the most likely trajectory. The trajectory from the policy with the highest score is shown in Fig. 3(c): we can observe that the agent managed to collect both coins while maintaining a cautious distance from the wumpus, albeit taking a longer path, which is more explicable than the agent's optimal behavior in Fig. 3(a) and simultaneously more efficient than the human's expectation in Fig. 3(b). Fig. 3(d) showcases the behavior obtained by PAG+, which also maintains a cautious distance from the wumpus, for the most part.

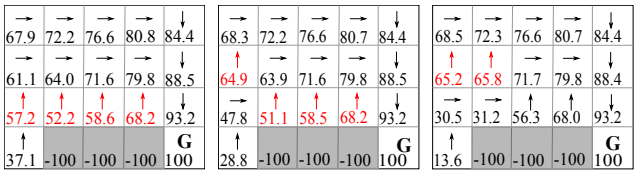
Physical Robot Experiment

Robot Assistant Domain: We implemented a scenario similar to the motivating example, where a MOVO robot assists a human user in setting up the dining table (Fig. 6). The task involves fetching a napkin for the user from another table. However, the user lacks a full understanding of the kinematic constraints of the robot arms, expecting the robot to reach any location within its arm's length. Consequently, the user anticipates that the robot will place the napkin beside the plate, close to her. In contrast, the robot's model accounts for restricted arm movement due to a vase on the table. Placing the napkin close to the user may risk



(a) $\tau(\pi^*_{\mathcal{M}_R})$ (b) $\tau(\pi^*_{\mathcal{M}_R^H})$ (c) $\tau(\pi^*_{\mathcal{E}})|\delta = 0.95$

Figure 4: Behavior comparison in the large cliff world. Grey areas is the cliff and G is the goal. Reward for each state is shown at the top right corner. Displayed are the most likely trajectories from policies: (a) the optimal policy under $\mathcal{M}_R$, (b) the optimal policy under $\mathcal{M}_R^H$ (i.e., human expectation), (c) the safe explicable policies returned by PDT+ (green) and PAG+ (blue).



(a) (b) (c)

Figure 5: Pareto set obtained by PDT+ when $\delta = 0.90$ and their corresponding V values in $\mathcal{M}_R^H$, in the small cliff world. Values highlighted in red are those that result in non-dominated policies. (b) shows the policy obtained by PAG+.

tipping over the vase containing water, posing a safety hazard. Therefore, the robot's optimal behavior dictates placing the napkin next to the vase, albeit farther away from the user.

In this experiment, we operated within a discretized environment where the state space was defined by the following variables: the robot's location, the napkin's location, and the vase's location. Transitions between discrete states were facilitated by pre-generated robot trajectories using *Move It*. Specifically, in $\mathcal{M}_R^H$ the robot can access any location on the dining table regardless of its own position or the vase's placement, whereas $\mathcal{M}_R$ accurately reflects the influences from these factors. Our objective is to showcase that a robot operating under SEP would opt for a costlier policy in $\mathcal{M}_R$ to align with human expectations while ensuring safety.

Results: Fig. 6 depicts the safe explicable behaviors obtained from the robot experiment. The optimal behavior in $\mathcal{M}_R$ had two steps: the robot picked the napkin and placed it on the table next to the vase, away from the user. Further, we ran SEP with two different bounds, yielding two distinct safe explicable behaviors. When $\delta = 0.85$ (Fig. 6(a)), the robot picked the napkin, circumvented the obstruction by the vase by moving its entire base closer to the user and then placed the napkin next to the plate. When $\delta = 0.80$ (Fig. 6(b)), the robot initially moved the vase away to clear the obstruction, then picked the napkin and placed it next to the plate.

(a) $\pi^*_{\mathcal{E}} \mid \delta = 0.85$ (b) $\pi^*_{\mathcal{E}} \mid \delta = 0.80$

Figure 6: Safe explicable behaviors generated by PAG+ in the robot assistant domain under different bounds.

Conclusions

In this paper, we introduced the Safe Explicable Planning (SEP) problem, an extension of the explicable planning problem to support a safety bound. Our formulation generalizes the consideration of multiple objectives that are addressed in conventional MOMDPs or CMDPs to multiple models. The solution to SEP is a safe explicable policy that satisfies the safety bound while maximizing explicability. We proposed an action pruning technique to reduce the search space, an exact method to find the Pareto set of policies, and a greedy method to find a single policy in the Pareto set. We discussed approximate solutions through state aggregation based on state features and action choices to address scalability. However, our methods are still susceptible to policy explosion in complex domains our approach shows initial steps towards finding approximate safe explicable policies, with further research needed for more generalized and efficient approximation solutions. We conducted evaluations via simulations and physical robot experiments to validate the efficacy of our approach.

Acknowledgments

This research is supported in part by the NSF grant 2047186. The authors would also like to thank the anonymous reviewers for their helpful comments and suggestions.

References

Abel, D.; Arumugam, D.; Lehnert, L.; and Littman, M. 2018. State abstractions for lifelong reinforcement learning. In *ICML*.

Abel, D.; Hershkowitz, D.; and Littman, M. 2016. Near optimal behavior via approximate state abstraction. In *ICML*.

Altman, E. 1994. Denumerable constrained Markov decision processes and finite approximations. *Mathematics of operations research*.

Altman, E. 2021. *Constrained Markov decision processes*. Routledge.

Baker, C.; Saxe, R.; and Tenenbaum, J. 2011. Bayesian theory of mind: Modeling joint belief-desire attribution. In *CogSci*.

Barrett, L.; and Narayanan, S. 2008. Learning all optimal policies with multiple criteria. In *ICML*.

Buchholz, P.; and Scheftelowitsch, D. 2019. Computation of weighted sums of rewards for concurrent MDPs. *MMOR*.

Chakraborti, T.; Kulkarni, A.; Sreedharan, S.; Smith, D. E.; and Kambhampati, S. 2019. Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior. In *ICAPS*.

Chakraborti, T.; Sreedharan, S.; and Kambhampati, S. 2020. The emerging landscape of explainable ai planning and decision making. *arXiv*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. In *NeurIPS*.

Dragan, A. D.; Lee, K. C.; and Srinivasa, S. S. 2013. Legibility and predictability of robot motion. In *HRI*.

Dragan, A. D.; and Srinivasa, S. S. 2013. Generating Legible Motion. In *RSS*.

Ferrer-Mestres, J.; Dietterich, T. G.; Buffet, O.; and Chades, I. 2020. Solving k-mdps. In *ICAPS*.

Fox, M.; Long, D.; and Magazzeni, D. 2017. Explainable planning. *arXiv*.

Garcıa, J.; and Fernández, F. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*.

Gong, Z.; and Zhang, Y. 2022. Explicable Policy Search. In *NeurIPS*.

Gábor, Z.; Kalmár, Z.; and Szepesvári, C. 1998. Multi-criteria Reinforcement Learning. In *ICML*.

Hanni, A.; Boateng, A.; and Zhang, Y. 2024. Safe Explicable Planning. arXiv:2304.03773.

Hanni, A.; and Zhang, Y. 2021. Generating Active Explicable Plans in Human-Robot Teaming. In *IROS*.

Holmes, M.; et al. 2004. Schema learning: Experience-based construction of predictive action models.

Huang, X.; Hong, S.; Hofmann, A.; and Williams, B. C. 2019. Online risk-bounded motion planning for autonomous vehicles in dynamic environments. In *Proceedings of the International Conference on Automated Planning and Scheduling*.

Ibarz, B.; Leike, J.; Pohlen, T.; Irving, G.; Legg, S.; and Amodei, D. 2018. Reward learning from human preferences and demonstrations in atari. In *NeurIPS*.

Juba, B.; and Stern, R. 2022. Learning probably approximately complete and safe action models for stochastic worlds. In *AAAI*.

Kulkarni, A.; Chakraborti, T.; Zha, Y.; Vadlamudi, S. G.; Zhang, Y.; and Kambhampati, S. 2016. Explicable Robot Planning as Minimizing Distance from Expected Behavior. *arXiv*.

MacNally, A. M.; Lipovetzky, N.; Ramirez, M.; and Pearce, A. R. 2018. Action selection for transparent planning. In *AAMAS*.

Moldovan, T. M.; and Abbeel, P. 2012. Safe exploration in markov decision processes. *arXiv*.

Pineda, L. E.; Wray, K. H.; and Zilberstein, S. 2015. Revisiting Multi-Objective MDPs with Relaxed Lexicographic Preferences. In *AAAI Fall Symposium Series*.

Roijers, D. M.; Vamplew, P.; Whiteson, S.; and Dazeley, R. 2013. A survey of multi-objective sequential decision-making. *JAIR*.

Russell, S. J.; and Zimdars, A. 2003. Q-decomposition for reinforcement learning agents. In *ICML*.

Singh, S.; and Cohn, D. 1997. How to dynamically merge Markov decision processes. In *NeurIPS*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Sỳkora, O. 2008. State-space dimensionality reduction in Markov decision processes. *WDS*.

Van Moffaert, K.; Drugan, M. M.; and Nowé, A. 2013. Scalarized multi-objective reinforcement learning: Novel design techniques. In *ADPRL*.

Wakuta, K.; and Togawa, K. 1998. Solution procedures for multi-objective Markov decision processes. *Journal of Mathematical Programming and Operations Research*.

White, D. 1982. Multi-objective infinite-horizon discounted Markov decision processes. *Journal of mathematical analysis and applications*.

Wray, K.; Zilberstein, S.; and Mouaddib, A.-I. 2015. Multi-Objective MDPs with Conditional Lexicographic Reward Preferences. In *AAAI*.

Zhang, Y.; Sreedharan, S.; Kulkarni, A.; Chakraborti, T.; Zhuo, H.; and Kambhampati, S. 2017. Plan explicability and predictability for robot task planning. In *ICRA*.