Design for Explicability

UNDER REVIEW · DO NOT DISTRIBUTE

Anagha Kulkarni[1*] · **Sarath Sreedharan**[1*] · **Sarah Keren**[2] · **Tathagata Chakraborti**[3]
David E. Smith · **Subbarao Kambhampati**[1]

[1]Arizona State University · [2]Harvard University · [3]IBM Research AI

anaghak@asu.edu, ssreedh3@asu.edu, skeren@seas.harvard.edu, tchakra2@ibm.com, david.smith@psresearch.xyz, rao@asu.edu

Abstract

Designing agents capable of generating explicable behavior is a pre-requisite for achieving effective human-AI collaboration. However, exhibiting such behavior in arbitrary environments could be quite expensive for the agents involved, and in some cases, the agent may not even be able to exhibit the expected behavior. Given structured environments (like warehouses and restaurants), it may be possible to design the environment so as to boost explicable behavior on the part of the agent or to shape the human's expectations of the agent's behavior. In this paper, we investigate the opportunities and limitations of environment design as a tool to promote explicable behavior generation. We formulate a novel environment design framework that considers design over multiple tasks and over a time horizon. In addition, we explore the longitudinal aspect of explicable behavior and the trade-off that arises between the cost of design and the cost of generating explicable behavior over a time horizon.

1 Introduction

As more and more autonomous agents are deployed into environments cohabited by humans, it becomes important that the agents are capable of acting in a manner that is explicable to the humans in the loop. Inexplicable behavior, on the part of the agent, may not only lead to increased cognitive load on the human but also may lead to loss of trust in the agent's capabilities and in the worst case, may lead to increased risk or danger around the agent [Fan *et al.*, 2008]. In order to be explicable to the human, the agent should make its behavior consistent with the human's expectations of it. However, the human's expectation may deviate from reality as the human may have incorrect mental models about the agent's beliefs and capabilities and about the environment. In addition, the environment in which the agent is operating may not always be conducive to explicable behavior. This may lead to inhibition of certain explicable behaviors or may lead to prohibitively expensive behaviors.

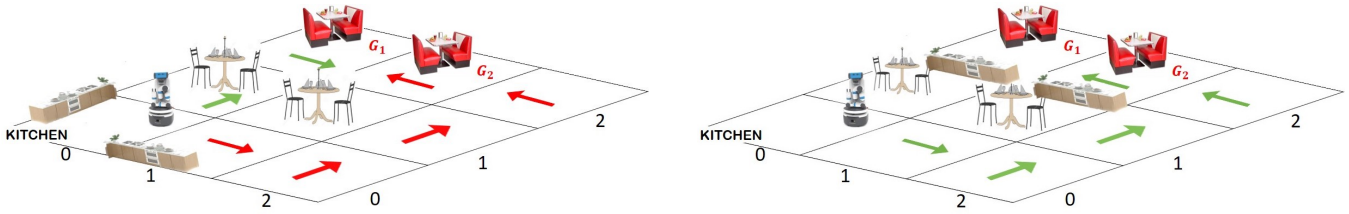
Fortunately, in highly structured settings, where the agent is expected to solve repetitive tasks (like in warehouses, factories, restaurants, etc.), it might be feasible to design the environment in a way that improves explicability with respect to multiple tasks. This brings us to the problem of environment design which involves designing the environment so as to maximize (or minimize) some objective for the agent (for example, optimal-cost to a goal, desired behavioral property) [Zhang *et al.*, 2009]. While the problem of environment design for planning problems has been investigated before, to the best of our knowledge, we are the first to explore the notion of environment design to maximize the explicability of an agent's behavior.

However, environment design alone may not be a panacea for explicability. For one, the design could be quite expensive, not only in terms of making the required environment changes but also in terms of limiting the capabilities of the actor. Moreover, in many cases, there may not be a single set of design modifications that will work for all the problems. For instance, designing a robot with wheels for efficient navigation on the floor will not optimize the robot's motion up a stairwell. This leads us to investigate a novel optimization space, that requires trading off one-time (but potentially expensive) design changes, against repetitive penalties borne by the agent to achieve explicable behavior.

The main contributions of our paper are as follows:

1. We propose a new design framework that:
   (a) balances the cost overhead incurred by the agent for explicable behavior with the design cost for modifying the environment,
   (b) optimizes this objective over a set of tasks over the lifetime of the agent
2. Our work is the first to model the longitudinal aspect of explicable behavior that arises from repetitive execution of tasks over the task lifetime.
3. In solving the objective, we leverage a classical planning compilation [Sreedharan *et al.*, 2019] to generate the most explicable plan for a task in a given environment configuration and explore its theoretical properties.
4. Through empirical evaluation and demonstration of our approach in a simulated domain, we examine the properties of our optimization criterion and the various trade-offs that result from it.

[*]equal contribution



(a) Explicable behavior is costlier without design.

(b) Optimal behavior is explicable with design.

Figure 1: Use of environment design to improve the explicability of robot behaviors in shared environments.

Motivating Example Consider a restaurant with a robot server (Figure 1a). Let G_1 and G_2 represent the robot's goals of serving the two booths: it travels between the kitchen and the two booths. The observers consist of human servers and customers at the restaurant. Given the position of the kitchen, the observers may have expectations on the route taken by the robot. However, unbeknownst to the observers, the robot can not traverse between the two tables and therefore takes the route around the tables. Therefore, the path marked in red is the cheapest path for the robot but the observers expect the robot to take the path marked in green in Figure 1a.

In this environment, there is no way for the robot to be explicable. Applying environment design provides us with alternatives. For example, the designer could choose to build two barriers as shown in Figure 1b. With these barriers in place, the humans would expect the robot to follow the path highlighted in green. Now, the question of whether in this case it is preferable to perform environment modifications or require that the robot avoid inexplicable behavior depends on the cost of changing the environment versus the cost of inexplicability caused by the behavior. In the rest of the paper, we will explore the details of this trade-off.

2 Background

We consider two types of agents: an actor (e.g. a robot) and an observer (human). In this section, we introduce the notion of generating explicable behavior and the problem of environment design, with respect to these two agents.

Explicability Let $\mathcal{P}_A = \langle \mathcal{F}, \mathcal{A}_A, \mathcal{I}_A, \mathcal{G}_A, c_A \rangle$ be the actor's model captured as a classical planning problem [Geffner and Bonet, 2013], where $\mathcal{F}$ is set of fluents, $\mathcal{A}_A$ is the set of actor's actions, $\mathcal{I}_A$ is its initial state, $\mathcal{G}_A$ is its goal which is a set of instantiated fluents, and c_A is the cost of its actions. A plan π is a solution to $\mathcal{P}_A$, if $\Gamma_A(\mathcal{I}_A, \pi) \models \mathcal{G}_A$, where $\Gamma_A(\cdot,\cdot)$ is the actor's transition function. The need for generating explicable behavior arises because the actor's planning model is different from the observer's mental model of it. The difference can be in terms of set of actions, initial state or goal of the actor. Thus an explicable planning problem is defined as $\mathcal{P}_{Exp} = \langle \mathcal{P}_A, \mathcal{P}_O, \delta_{\mathcal{P}_O} \rangle$, where $\mathcal{P}_O = \langle \mathcal{F}, \mathcal{A}_O, \mathcal{I}_O, \mathcal{G}_O, c_O \rangle$ represents the observer's mental model of it, and $\delta_{\mathcal{P}_O}$ is a distance function used by the observer to compute the explicability of a plan. We assume that the observer model is available. This is usually the case when any product is deployed and developers capture a generic user model which can be learned from prior interactions. In this work, we only focus on the reasoning aspects once we have the model, rather than focusing on the acquisition of such a model which can be an input to our approach. Let $\Pi^*_{\mathcal{P}_\mathcal{O}}$ represent the set of expected plans with respect to $\mathcal{P}_\mathcal{O}$. Here, $\Pi^*_{\mathcal{P}_\mathcal{O}}$ captures the notion of the observer's preference on the plans feasible in its mental model. A valid plan that solves $\mathcal{P}_A$ can exist anywhere on the spectrum of inexplicability from high to low.

Definition 1. *The **inexplicability score**, $\mathcal{IE}(\cdot,\cdot,\cdot)$, of the actor's plan π_A that solves $\mathcal{P}_A$ is defined as follows for the observer model $\mathcal{P}_O$ and associated distance function $\delta_{\mathcal{P}_O}(\cdot,\cdot)$:*

$$\mathcal{IE}(\pi_A, \mathcal{P}_O, \delta_{\mathcal{P}_O}) = \min_{\pi_O \in \Pi^*_{\mathcal{P}_O}} \delta_{\mathcal{P}_O}(\pi_A, \pi_O) \tag{1}$$

The actor's objective is to minimize the inexplicability score of its plan in the observer's mental model. We will use the notation $\Pi^*_{\mathcal{IE}(\cdot,\mathcal{P}_O,\delta_{\mathcal{P}_O})}$ to refer to the set of plans in the actor's model with the lowest inexplicability score, and $\mathcal{IE}_{min}(\mathcal{P}_{Exp})$ to represent the lowest inexplicability score associated with the set. Further, let f_{Exp} represent the decision function used by the explicable actor, such that, $f_{Exp}(\mathcal{P}_{Exp})$ represents the cheapest plan that minimizes the inexplicability score, i.e., $f_{Exp}(\mathcal{P}_{Exp}) \in \Pi^*_{\mathcal{IE}(\cdot,\mathcal{P}_O,\delta_{\mathcal{P}_O})}$ and $\nexists \pi' : \pi' \in \Pi^*_{\mathcal{IE}(\cdot,\mathcal{P}_O,\delta_{\mathcal{P}_O})} \wedge c_A(\pi') < c_A(f_{Exp}(\mathcal{P}_{Exp}))$.

Environment Design An environment design problem [Zhang *et al.*, 2009] takes as input the initial environment configuration along with a set of modifications allowed in the environment and computes a subset of modifications that can be applied to the initial environment to derive a new environment in which a desired objective is optimized.

Let $\mathcal{P}^0_A = \langle \mathcal{F}^0, \mathcal{A}^0_A, \mathcal{I}^0_A, \mathcal{G}^0_A, c^0_A \rangle$ denote the initial environment in which the actor is operating, ρ_A be the set of valid configurations of that environment, such that $\mathcal{P}^0_A \in \rho_A$. Let $\mathcal{O}$ be the metric being optimized for, i.e a model with lower value for $\mathcal{O}$ is preferred. A design problem (adapted from [Zhang *et al.*, 2009]) is a tuple $\langle \mathcal{P}^0_A, \Delta, \Lambda_A, C, \mathcal{O} \rangle$ where, Δ is the set of all modifications, $\Lambda_A : \rho_A \times 2^\Delta \rightarrow \rho_A$ is the model transition function that specifies the resulting model after applying a subset of modifications to the existing model, $C : \Delta \rightarrow \mathbb{R}$ is the cost function that maps each design choice to its cost. That is, the modifications are independent of each other and their costs are additive. We will overload the notation and also use C as the cost function for a subset of modifications ($C(\xi) = \sum_{\xi_i \in \xi} C(\xi)$).

The set of possible modifications may include modifications to the set of states, action preconditions, action effects, action costs, initial state and goal. An optimal solution

for a design problem, identifies a subset of design modifications, ξ, that minimizes the following objective function consisting of the cost of modifications and the metric $\mathcal{O}$: $\min \mathcal{O}(\Lambda_A(\mathcal{P}^0_A, \xi)),\ C(\xi)$.

3 Design for Explicability

In this framework, we not only discuss the problem of environment design with respect to explicability but also in the context of **(1)** a set of tasks that the actor has to perform in the environment, and **(2)** over the lifetime of the tasks i.e. the time horizon over which the actor is expected to repeat the execution of the given set of tasks. These two considerations add an additional dimension to the environment design problem since the design of environments will have lasting effects on the behavior of agents inside it. In the following, we will first introduce the design problem for a single explicable planning problem, then extend it to a set of explicable planning problems and lastly extend it over a design time horizon.

3.1 Single Explicable Problem

In the design problem for explicability, the inexplicability score becomes the metric that we want to optimize for. This problem can be defined as follows:

Definition 2. *The design problem for explicability is a tuple, $\mathcal{DP}_{Exp} = \langle \mathcal{P}^0_{Exp}, \Delta, \Lambda_{Exp}, C, \mathcal{IE}_{min} \rangle$, where:*

- *$\mathcal{P}^0_{Exp} \in \rho_{Exp}$ is the initial configuration of the explicable planning problem, where ρ_{Exp} represent the set of valid configurations for $\mathcal{P}_{Exp}$.*
- *Δ is the set of design modifications that are available in the given set of environment configurations ρ_{Exp}.*
- *$\Lambda_{Exp} : \rho_{Exp} \times 2^{\Delta} \rightarrow \rho_{Exp}$ is the transition function over the explicable planning problem.*
- *C is the cost associated with each design choice in Δ.*
- *$\mathcal{IE}_{min} : 2^{\rho_{Exp}} \rightarrow \mathbb{R}$ is the minimum possible inexplicability score in an explicable problem configuration.*

With respect to the example in Figure 1a, $\mathcal{DP}_{Exp}$ is the problem of designing the environment to improve the robot's explicability given its task of serving every new customer at a booth (say G_1) only once. The optimal solution to $\mathcal{DP}_{Exp}$ involves finding a configuration which minimizes the minimum inexplicability score. We also need to take into account an additional optimization metric which is the effect of design modifications on the actor's plan cost. That is, we need to examine to what extent the decrease in inexplicability is coming at the actor's expense. For instance, if you confine the robot to a cage so that it cannot move, its behavior becomes completely and trivially explicable, but the cost of achieving its goals goes to infinity.

Definition 3. *An **optimal solution** to $\mathcal{DP}_{Exp}$, is a subset of modifications ξ^* that,*

$$\min\ \mathcal{IE}_{min}(\ \mathcal{P}^*_{Exp}),\ C(\xi^*), c_A(f_{Exp}(\mathcal{P}^*_{Exp})) \qquad (2)$$

where $\mathcal{P}^*_{Exp} = \Lambda_{Exp}(\mathcal{P}^0_{Exp}, \xi^*)$ is the final modified explicable problem, $\mathcal{IE}_{min}(\cdot)$ represents the minimum possible inexplicability score for a given configuration, $C(\xi^*)$ denotes the cost of the set of design modifications and $c_A(f_{Exp}(\mathcal{P}^*_{Exp}))$ is the cost of cheapest most explicable plan in a configuration.

3.2 Multiple Explicable Problems

We will now show how $\mathcal{DP}_{Exp}$ evolves when there are multiple explicable problems in the environment that the actor needs to solve. When there are multiple explicable tasks there may not exist a single set of design modifications that may benefit all the tasks. In such cases, a solution might involve performing design modifications that benefit some subset of the tasks while allowing the actor to act explicably with respect to the remaining set of tasks. Let there be k explicable planning problems, given by the set $\mathbf{P}_{Exp} = \{\langle \mathcal{P}_A(0), \mathcal{P}_O(0), \delta_{\mathcal{P}_O(0)} \rangle, \ldots, \langle \mathcal{P}_A(k), \mathcal{P}_A(k), \delta_{\mathcal{P}_O(0)} \rangle\}$, with a categorical probability distribution $\mathcal{D}$ over the problems. We use $\mathcal{P}_{Exp}(i) \in \mathbf{P}_{Exp}$ to denote the i^{th} explicable planning problem. Now the design problem can be defined as: $\mathcal{DP}_{Exp,\mathcal{D}} = \langle \mathbf{P}^0_{Exp}, \mathcal{D}, \Delta, \Lambda_{Exp}, C, \mathcal{IE}_{min,\mathcal{D}} \rangle$, where $\mathbf{P}^0_{Exp}$, is the set of planning tasks in the initial environment configuration, $\mathcal{IE}_{min,\mathcal{D}}$ is a function that computes the minimum possible inexplicability score in a given environment configuration by taking expectation over the minimum inexplicability score for each explicable planning problem, i.e., $\mathcal{IE}_{min,\mathcal{D}}(\mathbf{P}_{Exp}) = \mathbb{E}[\mathcal{IE}_{min}(\mathcal{P}_{Exp}) | \mathcal{P}_{Exp} \sim \mathcal{D}]$. With respect to our running example, $\mathcal{DP}_{Exp,\mathcal{D}}$ is the problem of designing the environment given the robot's task of serving every new customer at either of the two booths (say G_1, G_2) with probability given by distribution $\mathcal{D}$.

The solution to $\mathcal{DP}_{Exp,\mathcal{D}}$ has to take into account the distribution over the set of explicable planning problems. Therefore the optimal solution is given by: $\min\ (\ \mathcal{IE}_{min,\mathcal{D}}(\ \mathbf{P}^*_{Exp}),\ C(\xi^*),$
$\mathbb{E}[c_A(f_{Exp}(\mathcal{P}^*_{Exp})) \mid \mathcal{P}^*_{Exp} \sim \mathcal{D}]\ )$.

3.3 Longitudinal Impact on Explicable Problems

The process of applying design modifications to an environment makes more sense if the tasks are going to be performed repeatedly. This has quite a different temporal characteristic in comparison to that of execution of one-time explicable behavior in an environment. For instance, design changes are associated with a one-time cost (i.e. the cost of applying those changes in the environment). On the other hand, if we are relying on the actor to execute explicable plans at the cost of foregoing optimal plans, then it needs to bear this cost multiple times over the time horizon.

We will use a discrete time formulation where the design problem is associated with a time horizon $\mathcal{T}$. At each time step, one of the k explicable planning problems is chosen. Now the design problem can be defined as: $\mathcal{DP}_{Exp,\mathcal{D},\mathcal{T}} = \langle \mathbf{P}^0_{Exp}, \mathcal{D}, \Delta, \Lambda_{Exp}, C, \mathcal{IE}_{min,\mathcal{D}}, \mathcal{T} \rangle$. In our running example, $\mathcal{DP}_{Exp,\mathcal{D},\mathcal{T}}$ is the problem of designing the environment given the robot's task of serving the same customer at either of the booths with a distribution $\mathcal{D}$ over a horizon $\mathcal{T}$.

In the past literature, the explicability of an actor's behavior has been studied with respect to a single interaction with an observer over a given task [Zhang *et al.*, 2017; Kulkarni *et al.*, 2019a]. However, we consider a time horizon, $\mathcal{T} > 1$, over which the actor's interaction with the observer may be repeated multiple times for the same task. This means the observer's expectations about the task could evolve over time. This may not be a problem if the actor's behavior aligns perfectly with the observer's expectations. Although,

if the actor's plan for a given task is associated with a non-zero inexplicability score, then the observer is likely to be more surprised the very first time they notice the inexplicable behavior than they would be if they noticed the inexplicable behavior the subsequent times. Since, the second time the actor performs the same task, the observer may get used to the inexplicability of the actor's behavior and may expect the actor to perform the same inexplicable behavior. As the observer watches the task being performed over and over, the amount of surprise associated with the inexplicable behavior starts decreasing. In fact, there is a probability that the observer may start ignoring the inexplicability of the actor's behavior after sufficient repetitions of the task. We incorporate this intuition by using an intuitive form of discounting.

We use a Markov reward process to represent the dynamics of this system. Let $(1-\gamma)$ denote the probability that the observer will ignore the inexplicability of the actor's plan, i.e, the reward will have inexplicability score 0. γ times the probability of executing a task represents the probability that the reward will have the minimum inexplicability score associated with that task. Assuming $\gamma < 1$, the minimum possible inexplicability score with respect to the set of explicable planning problems can be written as follows: $f_{\mathcal{T}}(\mathcal{IE}_{min,\mathcal{D}}(\mathbf{P}_{Exp})) = \mathcal{IE}_{min,\mathcal{D}}(\mathbf{P}_{Exp}) * \frac{1-\gamma^{\mathcal{T}}}{1-\gamma}$.

Thus the optimal solution to $\mathcal{DP}_{Exp,\mathcal{D},\mathcal{T}}$ is given by: $\min\ (\ f_{\mathcal{T}}(\mathcal{IE}_{min,\mathcal{D}}(\ \mathbf{P}^*_{Exp})),\ C(\xi^*),$
$\mathbb{E}[c_A(f_{Exp}(\mathcal{P}^*_{Exp}))\mid \mathcal{P}^*_{Exp} \sim \mathcal{D}] * \mathcal{T}\ )$. Note that, since the design cost is not discounted and we always make the design changes before the problem is selected in an episode, there is never a reason to delay the design execution to future steps in the horizon. Instead it can be executed at the first step.

4 Solution Methodology

We now discuss a solution strategy for our design problem when a cost-based distance function ($\delta^c_{\mathcal{P}_O}$) is used by the observer to determine the inexplicability of a plan. Given a plan π, such that, $\Gamma_{\mathcal{P}_A}(\mathcal{I}_A, \pi) \models \mathcal{G}_A$, the distance from a plan π' valid in the observer model is given as $\delta^c_{\mathcal{P}_O}(\pi, \pi') =$

$$\begin{cases} \exp(|c_O(\pi) - c_O(\pi')|), & if\ \Gamma_{\mathcal{P}_O}(\mathcal{I}_O, \pi) \models \mathcal{G}_O \\ \infty, & otherwise \end{cases}$$

Here, we will use the set of plans that are optimal in the observer's mental model as the expected plan set. This means that for calculating Equation 1, we don't have to do an additional minimization over the space of expected plans as every plan in the actor's model should be equidistant from every optimal plan in the observer's mental model (and the distance is infinity if the current actor plan is not executable in the observer's mental model). For brevity, we will refer to any plan with infinite inexplicable score as being invalid for a problem in $\mathbf{P}_{Exp}$. Additionally, we will assume that the actions in both the models have unit costs. That is, $c_O(\pi) = c_A(\pi) = |\pi|$.

Proposition 1. $\forall i \in 1, \ldots, k, \pi, \pi' \in \Pi^*_{\mathcal{IE}(\cdot, \mathcal{P}_O(i), \delta_{\mathcal{P}_O(i)})}$, $c_A(\pi) = c_A(\pi')$.

The above proposition states that all plans in $\Pi^*_{\mathcal{IE}(\cdot, \mathcal{P}_O(i), \delta_{\mathcal{P}_O(i)})}$ have equal costs in $\mathcal{P}_A(i)$ due to the assumption of unit costs. Therefore, while calculating the value for the objective function of $\mathcal{DP}_{Exp,\mathcal{D},\mathcal{T}}$, we can choose an arbitrary plan from $\Pi^*_{\mathcal{IE}(\cdot, \mathcal{P}_O(i), \delta_{\mathcal{P}_O(i)})}$ to calculate the term corresponding to the actor's cost.

Search for Optimal Design To find the optimal solution for $\mathcal{DP}_{Exp,\mathcal{D},\mathcal{T}}$, we will perform a breadth first search over the space of environment configurations that are achievable from the the initial configuration through the application of the given set of modifications [Keren *et al.*, 2018]. The performance of the search depends on the number of designs available. By choosing appropriate design strategies, significant scale up can be attained. Thus each search node is a valid environment configuration and the possible actions are the applicable designs. For simplicity, we convert the multi-objective optimization in equation 2 into a single objective that is a linear combination. In the new objective each term is associated with a coefficient, say, α, β and κ, respectively. The value of each node is decided by the aforementioned objective function. For each node, it is straightforward to calculate the design modification cost. However, in order to calculate the minimum inexplicability score and the actor's plan cost, we have to generate a plan that minimizes the inexplicability score for each explicable planning problem in that environment configuration. To achieve this, we compile the problem of generating the explicable plan to a classical planning problem. We will discuss this compilation in the the following subsection. Essentially, our search has two loops: the outer loop which explores all valid environment configurations, and the inner loop which performs search in a valid environment configuration to find a plan that minimizes the inexplicability score. At the end of the search, the node with best value is chosen, and the corresponding set of design modifications, ξ^*, is output.

Compilation for Most Explicable Plan We show that generating the most explicable plan for a $\mathcal{P}_{Exp} = \langle \mathcal{P}_A, \mathcal{P}_O, \delta_{\mathcal{P}_O} \rangle$ is the same as generating an optimal plan, π^*_{mod}, for a transformed planning problem $\mathcal{P}_{mod}$. To this end, we leverage the compilation used by Sreedharan *et al.* [2019] and present a simplified version of it.

Definition 4. *Given an explicable planning problem, $\mathcal{P}_{Exp} = \langle \mathcal{P}_A, \mathcal{P}_O, \delta_{\mathcal{P}_O} \rangle$, the transformed planning problem is $\mathcal{P}_{mod} = \langle \mathcal{F}_{mod}, \mathcal{A}_{mod}, \mathcal{I}_{mod}, \mathcal{G}_{mod}, c_{mod} \rangle$, where,*

- $\mathcal{F}_{mod} = \mathcal{F}_A \cup \mathcal{F}_O$
- *For each* $a_{mod} \in \mathcal{A}_{mod}$, $a_{mod} = \langle pre(a_{mod}), add(a_{mod}), del(a_{mod}) \rangle$, *where,* $pre(a_{mod}) = \{f_A \cup f_O | f_A \in pre(a_A), f_O \in pre(a_O)\}, add(a_{mod}) = \{f_A \cup f_O | f_A \in add(a_A), f_O \in add(a_O)\}, del(a_{mod}) = \{f_A \cup f_O | f_A \in del(a_A), f_O \in del(a_O)\}$
- $\mathcal{I}_{mod} = \{f_A \cup f_O | f_A \in \mathcal{I}_A, f_O \in \mathcal{I}_O\}$, *and* $\mathcal{G}_{mod} = \{f_A \cup f_O | f_A \in \mathcal{G}_A, f_O \in \mathcal{G}_O\}$
- *For each* $a_{mod} \in \mathcal{A}_{mod}$, $c_{mod}(a_{mod}) = c_O(a_O) = 1$

That is, we maintain a separate set of fluents to represent the actor's fluents from $\mathcal{P}_A$, and the observer's belief about it based on $\mathcal{P}_O$. We assume there is a one to one mapping between the actions in the actor's model and those in the observer's mental model, so there are two versions of each action. The action transformation ensures that an action is

executable by the actor if and only if its preconditions are satisfied in both the actor's model and the observer's model, and that it produces effects consistent with both models.

Proposition 2. *The $\mathcal{P}_{mod}$ produces a plan that solves $\mathcal{P}_{Exp}$, so that following properties hold:*

- ***Soundness*** *A plan π_{mod} that solves $\mathcal{P}_{mod}$ is a valid solution for $\mathcal{P}_{Exp}$.*
- ***Completeness*** *For every valid plan that solves $\mathcal{P}_{Exp}$, there is a corresponding valid plan that solves $\mathcal{P}_{mod}$.*
- ***Optimality*** *A plan π^*_{mod} that solves $\mathcal{P}_{mod}$ optimally is the most explicable plan for $\mathcal{P}_{Exp}$.*

The transformed planning problem has the union of the constraints imposed by both $\mathcal{P}_A$ and $\mathcal{P}_O$. Hence, a plan π_{mod} that solves $\mathcal{P}_{mod}$ is a valid plan for $\mathcal{P}_{Exp}$.

From the definition of inexplicability score of a plan π_A that solves $\mathcal{P}_{Exp}$, we know that $\Gamma_{\mathcal{P}_O}(\mathcal{I}_O, \pi_A) \models \mathcal{G}_O$. Therefore, for every valid plan that solves $\mathcal{P}_{Exp}$, there exists a corresponding plan that solves $\mathcal{P}_{mod}$.

Given $\mathcal{P}_{Exp}$, let π' be the plan with lowest inexplicable score, such that, it is not an optimal plan for $\mathcal{P}_{mod}$. Further, by completeness property, we know that π' must be a valid plan for $\mathcal{P}_{mod}$. This means for a plan, π^*_{mod}, optimal in $\mathcal{P}_{mod}$, we have $c_O(\pi^*_{mod}) < c_O(\pi')$ (since $\mathcal{P}_{mod}$ uses c_O). Hence, $|c_O(\pi^*_{mod}) - c^*_O| < |c_O(\pi') - c^*_O|$, where c^*_O is the cost of a optimal plan in $\mathcal{P}_O$ (and we know $c^*_O \leq c_O(\pi^*_{mod})$ and $c^*_O \leq c_O(\pi')$), which means $\mathcal{IE}(\pi^*_{mod}, \mathcal{P}_O, \delta^c_{\mathcal{P}_O}) < \mathcal{IE}(\pi', \mathcal{P}_O, \delta^c_{\mathcal{P}_O})$. This contradicts the original assertion, hence proving that there is a one to one correspondence between optimal plans for $\mathcal{P}_{mod}$ and $\Pi^*_{\mathcal{IE}(\cdot, \mathcal{P}_O, \delta^c_{\mathcal{P}_O})}$.

5 Evaluation

We will now demonstrate for our running example how the explicability value and design cost of the optimal solution evolve when optimizing for a single explicable problem, multiple explicable problems and multiple problems with a design horizon. We will also evaluate the performance of our approach on three IPC domains.

Demonstration We use the restaurant domain from our running example Figure 1a to demonstrate how the design problem evolves. We constructed a domain where the robot had 3 actions: *pick-up* and *put-down* to serve the items on the tray and *move* to navigate between the kitchen and the booths. In the grid, some cells are blocked due to the tables and the robot cannot pass through these: cell(0, 1) and cell(1, 1). Therefore, the following passages are blocked: cell(0, 0)-cell(0, 1), cell(0, 1)-cell(0, 2), cell(0, 1)-cell(1, 1), cell(1, 0)-cell(1, 1), cell(1, 1)-cell(1, 2), cell(1, 1)-cell(2, 1). We considered 6 designs, each consisting of putting a barrier at one of the 6 passages to indicate the inaccessibility to the human.

For the following parameters: $\alpha = 1$, $\beta = 30$, $\kappa = 0.25$ and $\gamma = 0.9$, we ran our algorithm for three settings: (a) single explicable problem for $\mathcal{T} = 1$, (b) multiple explicable problems for $\mathcal{T} = 1$, and (c) multiple explicable problems for $\mathcal{T} = 10$. As mentioned before, (a) involved serving a new customer at a booth (say G_1) only once, (b) involved serving a new customer only once at either of the booths with equal probability and (c) involved serving a customer at most 10 times at either of the booths with equal probability. We found that for setting (a) and (b) there was no design chosen. This is because these settings are over a single time step and the cost of installing design modifications in the environment is higher than the amount of inexplicability caused by the actor. On the other hand, for setting (c), the algorithm generated the design in Figure 1b, which makes the robot's roundabout path completely explicable to the customers.

Domain setup We used three IPC domains for evaluation: `Blocksworld`, `IPC-Grid` and `Driverlog`. For each domain, we created two versions: the actor's domain and the observer's domain. We generated 20 design problems for each domain, and each had 3 planning problems with uniform probability distribution. We used Fast Downward with A* search and the *lmcut* heuristic [Helmert, 2006] to solve the compiled planning problems. The variable parameters in our implementation are α, β, κ (coefficients associated with the terms in the objective function), γ (discount factor) and $\mathcal{T}$ (time horizon). For all the domains we used actions and design modifications of unit cost.

For `Blocksworld`, the actor's domain was the original IPC domain, and the observer's domain assumed that the robot can pick up multiple blocks simultaneously. The set of allowed designs ensured that stacking for every block was preceded by picking the block up from the table. This would reduce the inexplicability for the observer as the only block that would be stacked is the one that was picked up from the table before stacking. This may involve notifying the observer about the new rule. For `IPC-Grid`, the actor's domain was the original IPC domain and the observer's domain assumed that diagonal movements were possible in the grid. We allowed design modifications that pruned diagonal actions. In actuality, this may involve notifying the observer that diagonal actions are not possible at certain locations. For `Driverlog`, the actor's domain was the original IPC domain and the observer's domain assumed that packages can be loaded and unloaded from anywhere regardless of the location of the driver. We allowed modifications that required load and unload actions to occur only after a disembark action. This may again involve notifying the observer about the new rules concerning load/unload actions.

Performance on IPC domains For this objective, we set the parameters α, β and κ to 1.0, 0.25, 0.25 respectively for all domains. That is, we gave more weight to minimizing inexplicability in the objective function. For all the problems, we set $\mathcal{T}$ to 1 and 10 and γ to 0.9. For each problem, we allowed the meta-search to run for at most 30 minutes. If the search completed within 30 minutes we output the optimal design modification, else we output the design modification which gave the best optimization value (or total cost) among the explored nodes. To show the impact of design modifications, we computed the inexplicability score, plan cost, total cost for most explicable plan in the initial model (i.e. without any design modification). To compare the impact of longitudinality, we compute these parameters for single step horizon and multi-step horizon.

In Table 1, we report the results for the 3 domains. By comparing the inexplicability score with and without design, we see that the inexplicability always decreases as expected.

Domain	Horizon	Metrics	Design Size	Inexplicability			Plan Cost			Total Cost			Time Taken (secs)
				w/o Design	w Design	% Difference	w/o Design	w Design	% Difference	w/o Design	w Design	% Difference	
Blocksworld	1	Avg	1.25	14.11	2.18	-84.54	8.69	9.52	9.58	16.28	4.87	-70.07	1800
		SD	0.79	16.86	0.92	-	1.39	1.85	-	17.11	1.38	-	
	10	Avg	1.25	91.90	14.20	-84.54	8.69	9.52	9.57	113.63	38.33	-66.27	
		SD	0.78	109.80	5.98	-	1.39	1.85	-	112.36	9.59	-	
IPC-Grid	1	Avg	0.75	3571.84	1455.39	-59.25	24.84	24.84	0	23326.29	1461.79	-93.73	1800
		SD	0.44	12043.62	4428.98	-	3.01	3.01	-	78444.61	4429.19	-	
	10	Avg	0.75	23264.19	9479.32	-59.25	24.84	24.84	0	23326.29	9541.61	-59.09	
		SD	0.44	78442.72	28846.93	-	3.01	3.01	-	78444.61	28848.86	-	
Driverlog	1	Avg	0.8	2.26	1.6	- 29.14	8.46	9.17	8.46	4.37	4.09	- 6.39	219.42
		SD	0.77	0.54	0.57	-	0.59	0.89	-	0.61	0.54	-	
	10	Avg	1.2	14.70	8.93	-39.28	8.45	9.71	14.76	35.85	33.50	- 6.57	
		SD	0.69	3.54	2.78	-	0.59	0.97	-	4.30	3.94	-	

Table 1: We report the impact of design modifications on inexplicability score, plan cost and total cost. We also report the average and standard deviation values for the three optimization terms in the objective function along with the run time taken by the our approach.

For Blocksworld and IPC-Grid, the percentage decrease is the same for one-step and multi-step horizon, this is because same set of designs were the best found solutions for both settings (under the time-limit) and the values got multiplied with the value of $\mathcal{T}$. On the other hand, for `Driverlog`, there were different designs selected, as is evident from the values. By comparing the plan cost with and without design, we can see that for `Blocksworld` and `Driverlog`, there is substantial increase in the plan cost. This is because for these two domains, the designs ensured an action could be performed only after execution of another action. In this case, the actor bears additional cost for improving the explicability. On the other hand for `IPC-Grid`, the action pruning strategy removed actions from the observer's mental model and therefore there is no increase in the plan cost. Similarly, by comparing the total cost with and without design, we can see that there is a significant decrease in the total cost after applying design modifications. This is because the optimization chooses design modifications that minimize the overall cost associated with the initial model.

6 Related Work

This work explores the connection between two parallel threads of work: one on environment design and the other on explicable behavior. The problem of environment design is connected to that of mechanism design [Narahari, 2014], which has been thoroughly investigated by the game theory community. Environment design [Zhang *et al.*, 2009] involves modifying the environment so as to maximize or minimize some objective for the actor [Keren *et al.*, 2017]. The problem of design has been leveraged to simplify related problems like goal recognition [Keren *et al.*, 2014], plan recognition [Mirsky *et al.*, 2019] etc. These works have studied the possibility of modifying the environment so as to make the agent's behavior easily recognizable. These works have also looked at various types of designs, including, action pruning [Keren *et al.*, 2014], action conditioning [Keren *et al.*, 2018], sensor refinement [Keren *et al.*, 2018], sensor placement [Keren *et al.*, 2016], etc. The problem of environment design has also been studied for stochastic actions [Wayllace *et al.*, 2016; Wayllace *et al.*, 2017].

The notion of explicability was introduced in Zhang *et al.* [2017], which discussed generating explicable behavior by learning the sequence of actions that are explicable to the observers. Kulkarni *et al.* [2019a] explored the notion of explicability given knowledge of the human's mental model, and used plan distances as a stand in for the observer's distance function. Generation of explicable behavior has also been studied in combination with explanations [Chakraborti *et al.*, 2019b]. Further, Sreedharan *et al.* [2019] explores the use of explanatory actions to convert the explanation generation problem to a sequential decision making problem. A recent work [Chakraborti *et al.*, 2019a], has also explored the connections between explicability and other types of interpretable behavior like legibility [Kulkarni *et al.*, 2019b], predictability [Dragan *et al.*, 2013].

7 Discussion and Conclusion

In this paper, we bridge the gap between past works on environment design and those on generation of explicable behavior. We present a novel framework of environment design for explicability. The notion of environment design makes sense when there is repeated execution of tasks or when there are multiple tasks in the environment. This allows us to explore a novel trade-off that arises between one-time cost of design modifications and the repeated cost overhead incurred by the actor for generating explicable behavior. In prior works on explicable plan generation, the underlying assumption has been that there is a one-time interaction between a human and a robot, and that the robot's inexplicable behavior may lead to increased cognitive load on the human or loss of trust in the robot. In this work, we relaxed this assumption and explored the notion of inexplicability given repeated interactions with a single observer over a distribution of tasks.

In this work, we assumed that the actor is capable of performing explicable behavior. However, we can also consider the problem of environment design for explicability when the actor is rational but not cooperative (i.e. it will only generate cost-optimal plans in the given environment and not bear the overhead cost of being explicable). In this case, the emphasis

is on choosing a set of design modifications which reduce the worst case inexplicability score associated with cost-optimal plans for a task. Similarly, we can also consider the problem of environment design for explicability when the actor is a rational actor but can communicate (i.e. it will only generate cost-optimal plans in the given environment but it will provide an explanation to make its behavior explicable). In this case, we again see similar trade-offs between cost of one-time environment design versus the cost of repeated explanations borne by the actor over a time horizon. This would require modeling the influence of longitudinal interactions on explanations which stems from the fact that the observer will update their mental model when they receive an explanation.

References

[Chakraborti *et al.*, 2019a] Tathagata Chakraborti, Anagha Kulkarni, Sarath Sreedharan, David Smith, and Subbarao Kambhampati. Explicability? Legibility? Predictability? Transparency? Privacy? Security?: The Emerging Landscape of Interpretable Agent Behavior. In *ICAPS*, 2019.

[Chakraborti *et al.*, 2019b] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. Balancing Explicability and Explanation in Human-Aware Planning. In *IJCAI*, 2019.

[Dragan *et al.*, 2013] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and Predictability of Robot Motion. In *HRI*, 2013.

[Fan *et al.*, 2008] Xiaocong Fan, Sooyoung Oh, Michael McNeese, John Yen, Haydee Cuevas, Laura Strater, and Mica R Endsley. The influence of agent reliability on trust in human-agent collaboration. In *ECCE*, 2008.

[Geffner and Bonet, 2013] Hector Geffner and Blai Bonet. A Concise Introduction to Models and Methods for Automated Planning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2013.

[Helmert, 2006] Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

[Keren *et al.*, 2014] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal Recognition Design. In *ICAPS*, 2014.

[Keren *et al.*, 2016] Sarah Keren, Avigdor Gal, and Erez Karpas. Privacy Preserving Plans in Partially Observable Environments. In *IJCAI*, 2016.

[Keren *et al.*, 2017] Sarah Keren, Luis Pineda, Avigdor Gal, Erez Karpas, and Shlomo Zilberstein. Equi-reward utility maximizing design in stochastic environments. In *IJCAI*, 2017.

[Keren *et al.*, 2018] Sarah Keren, Avigdor Gal, and Erez Karpas. Strong stubborn sets for efficient goal recognition design. In *Twenty-Eighth International Conference on Automated Planning and Scheduling*, 2018.

[Kulkarni *et al.*, 2019a] Anagha Kulkarni, Tathagata Chakraborti, Yantian Zha, Satya Gautam Vadlamudi, Yu Zhang, and Subbarao Kambhampati. Explicable Robot Planning as Minimizing Distance from Expected Behavior. In *AAMAS*, 2019. Extended Abstract.

[Kulkarni *et al.*, 2019b] Anagha Kulkarni, Siddharth Srivastava, and Subbarao Kambhampati. A Unified Framework for Planning in Adversarial and Cooperative Environments. In *AAAI*, 2019.

[Mirsky *et al.*, 2019] Reuth Mirsky, Kobi Gal, Roni Stern, and Meir Kalech. Goal and plan recognition design for plan libraries. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):14, 2019.

[Narahari, 2014] Yadati Narahari. *Game theory and mechanism design*, volume 4. World Scientific, 2014.

[Sreedharan *et al.*, 2019] Sarath Sreedharan, Tathagata Chakraborti, Christian Muise, and Subbarao Kambhampati. Planning with explanatory actions: A joint approach to plan explicability and explanations in human-aware planning. *CoRR*, abs/1903.07269, 2019.

[Wayllace *et al.*, 2016] Christabel Wayllace, Ping Hou, William Yeoh, and Tran Cao Son. Goal recognition design with stochastic agent action outcomes. In *IJCAI*, 2016.

[Wayllace *et al.*, 2017] Christabel Wayllace, Ping Hou, and William Yeoh. New metrics and algorithms for stochastic goal recognition design problems. In *IJCAI*, pages 4455–4462, 2017.

[Zhang *et al.*, 2009] Haoqi Zhang, Yiling Chen, and David C Parkes. A general approach to environment design with one agent. In *IJCAI*, 2009.

[Zhang *et al.*, 2017] Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. Plan Explicability and Predictability for Robot Task Planning. In *ICRA*, 2017.