Subgoal-Based Explanations for Unreliable Intelligent Decision Support Systems

Devleena Das [1], **Been Kim,** [2], **Sonia Chernova,** [1]

[1]Georgia Institute of Technology, Atlanta, GA, USA
[2]Google Research, Mountain View, CA, USA
ddas41@gatech.edu, beenkim@google.com, chernova@gatech.edu

Abstract

Intelligent decision support (IDS) systems leverage artificial intelligence techniques to generate recommendations that guide human users through the decision making phases of a task. However, a key challenge is that IDS systems are not perfect, and in complex real-world scenarios may produce incorrect output or fail to work altogether. The field of explainable AI planning (XAIP) has sought to develop techniques that make the decision making of sequential decision making AI systems more explainable to end-users. Critically, prior work in applying XAIP techniques to IDS systems has assumed that the plan being proposed by the planner is always optimal, and therefore the action or plan being recommended as decision support to the user is always correct. In this work, we examine novice user interactions with a non-robust IDS system – one that occasionally recommends the wrong action, and one that may become unavailable after users have become accustomed to its guidance. We introduce a novel explanation type, *subgoal-based explanations*, for planning-based IDS systems, that supplements traditional IDS output with information about the subgoal toward which the recommended action would contribute. We demonstrate that subgoal-based explanations lead to improved user task performance, improve user ability to distinguish optimal and suboptimal IDS recommendations, are preferred by users, and enable more robust user performance in the case of IDS failure.

Introduction

Intelligent decision support (IDS) systems leverage artificial intelligence techniques to generate recommendations that guide human users through the decision making phases of a task (Sutton et al. 2020). While much prior work has focused on decision support for domain experts (e.g., cancer diagnosis for oncologists (Walsh et al. 2019)), increasingly, IDS systems have been proven particularly useful in helping *novice users* make decisions (Arnold et al. 2004; Gutiérrez et al. 2019; Machado, Lam, and Chen 2018). However, a key challenge is that IDS systems are not perfect, and in complex real-world scenarios the actions recommended by IDS systems may be far from optimal (Guerlain, Brown, and Mastrangelo 2000). Such errors particularly strongly affect novice users, who lack the knowledge to assess the correctness of an IDS recommendation (Nourani, King, and Ragan 2020).

The field of explainable AI (XAI) has sought to make the decision making of AI systems more transparent by developing interpretability techniques for black-box AI models (Doshi-Velez and Kim 2017). While prior work on XAI largely focuses on explaining single classification tasks, the subfield of *explainable AI Planning (XAIP)* seeks to explain decisions in sequential decision making tasks. Prior work on XAIP has largely focused on explaining plan solutions that help users answer the question "Why plan P (and not plan Q)?" (Chakraborti, Sreedharan, and Kambhampati 2020). These techniques have been effective in helping domain-experts understand how their proposed solution differs from the planner's solution.

Critically, prior work in applying XAIP techniques to IDS systems has assumed that the plan being proposed by the planner is always optimal, and therefore the action or plan being recommended to the user is always correct (Grover et al. 2020; Valmeekam et al. 2020). However, optimal IDS decision making cannot be guaranteed in complex real world deployments. In fact, in real world systems, other assumed, robust characters of IDS systems may not hold true, including the ability to always receive suggestions at deployment. There may be situations in which a user's query is unanswerable, or the IDS system runs into a failure and is no longer available to the user.

In this work, we examine novice user interactions with a non-robust IDS system – one that occasionally recommends the wrong action, and one that may become unavailable after users have become accustomed to its guidance. A user of such a system, given an IDS action recommendation, must be able to determine whether the recommendation is optimal or not. In the absence of an IDS recommendation, the ideal user will have sufficient understanding of the task such that their task performance is not negatively impacted by the sudden absence of previously available recommendations. Leveraging insights from Psychology, which demonstrate that humans naturally break down large complex tasks into a smaller set of more manageable subgoals (Newell, Simon et al. 1972), we introduce a novel explanation type – *subgoal-based explanations* – that supplements traditional IDS output with information about the subgoal toward which

the recommended action would contribute. We then examine the impact such an explanation has on novice user performance through experiments in a complex planning domain–restaurant planning–with 95 study participants. Our work contributes several key findings:

- In the context of an imperfect IDS system, subgoal-based explanations enable users to successfully detect and avoid detect more erroneous IDS recommendations than users who are given traditional IDS output without explanations.
- Users who receive subgoal-based explanations achieve better performance when performing a task under IDS guidance than users who receive IDS guidance without explanations.
- Users who are exposed to subgoal-based explanations for some period of time, are then able to perform the task more reliably in the absence of further IDS, compared to users who did not receive explanations (only action recommendations). This finding suggests that explanations contribute a significant training benefit beyond traditional IDS output.
- In a direct comparison, users exhibited a strong preference for IDS output that includes subgoal-based explanations versus output that only recommends the next action.

We also show a simple way to generate domain-independent subgoal-based explanation that can be generalized to any hierarchical planning based system and broadly applicable across a wide range of IDS systems and application areas. Together with our findings, our work is a first step towards investigating how and when IDS with XAIP systems are beneficial in complex real-world IDS systems that are not fail-proof.

Related Work

In this section, we situate our work in the context of the two prominent research areas most closely related to our work: Intelligent Decision Support Systems and Explainable AI.

Intelligent Decision Support Systems

Intelligent decision support systems have been developed to assist users in decision making across a wide range of applications, such as providing assistance to domain-experts in clinical settings (Walsh et al. 2019; Zhuang et al. 2009) and aiding novice-users in management settings (Machado, Lam, and Chen 2018; Gutiérrez et al. 2019). Amongst these IDS systems, decision support is provided through a range of mediums, depending on the domain. For instance, in (Machado, Lam, and Chen 2018), the authors develop a mobile app for clinical decision support that allows dental students to answer a series of questions to determine a diagnosis and provide treatment suggestions. In (Gutiérrez et al. 2019), researchers investigate how to best portray visual representations and interaction techniques to aid novice users in business decisions. Alternatively, in (Papamichail and French 2000), authors develop a natural language generator to justify decision support in nuclear emergencies via natural language based reports.

Given that many IDS systems interact with end-users who are not AI-experts, several bodies of work have investigated how to enhance the transparency of IDS systems to improve user trust. These transparency techniques have been studied in the context of explaining recommendations for single-classification tasks, such as clinical decision support to identify failure modes (Jones, Mateer, and Harrison 2019), and prediction of Sepsis and mortality in ICUs (Feng, Shaib, and Rudzicz 2020). By contrast, our work investigates IDS in sequential decision making settings. Specifically, we investigate how to provide explanations that help novice users improve their sequential decision making performance in the presence of potentially incorrect suggestions from IDS.

Explainable AI

The field of explainable AI aims to improve a user's understanding of the inner workings of complex models (Doshi-Velez and Kim 2017). Given that AI and ML models are not guaranteed to be correct, an important objective of XAI techniques includes being able to help users identify vulnerabilities or "bugs" within a model (Adebayo et al. 2020) as well as identify any spurious correlations (Kim et al. 2018). Our work explores a scenario in which the underlying AI model, and therefore the explanation that results from it, may be incorrect. We examine how users can leverage explanations to ultimately improve their performance at a task.

To provide greater context for our work, we first review the types of explanations that have been developed in sequential decision making. The subfield of explainable AI Planning (XAIP) seeks to develop methods for explaining sequential decision making problems, where an agent engages in a longer-term interaction with a user (Chakraborti, Sreedharan, and Kambhampati 2020). Within the community, techniques have primarily focused on explaining an agent's entire plan solutions to end-users. A recent survey by Chakraborti et al. (Chakraborti, Sreedharan, and Kambhampati 2020) highlights the key areas of plan explanations that have been investigated, including generating contrastive explanations (Krarup et al. 2019; Hoffmann and Magazzeni 2019), explaining unsolvable plans (Sreedharan et al. 2019), and generating explicable plan explanations (Chakraborti et al. 2017).

Additionally, XAIP techniques have also been applied to plan-based decision support systems in efforts to improve human-in-the-loop planning. For example, RADAR by Grover et al. provides XAIP features such as plan summarization, plan explanations in the form of minimally complete contrastive explanations, plan validation, and action and plan suggestions to improve decision making in time critical scenarios (Grover et al. 2020). Valmeekam et al extend this RADAR and develop RADAR-X which leverages user queries understand user preferences for providing refined plan suggestions (Valmeekam et al. 2020). Our work similarly aims to support human-in-the-loop planning, with an important difference that we do not make the assumption of a perfect AI planner or perfect recommendations from IDS systems.

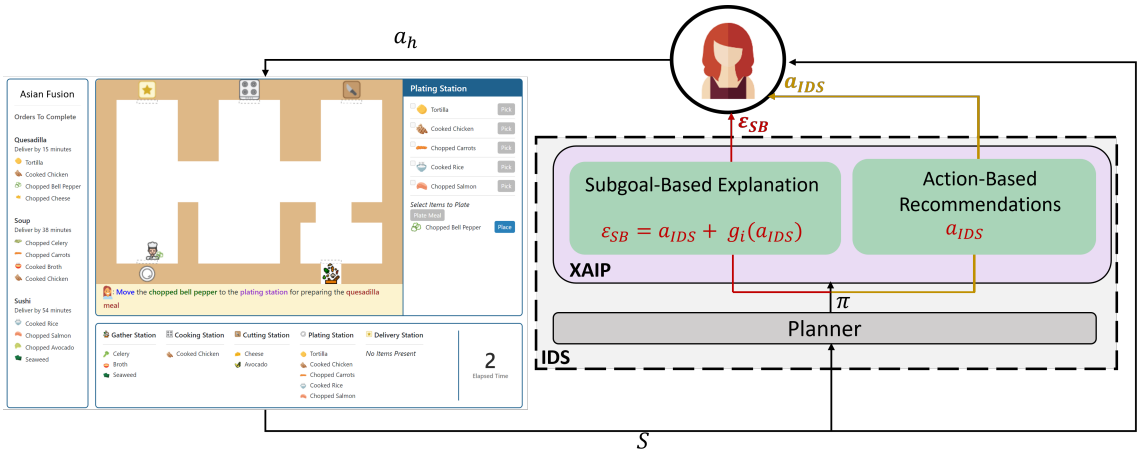


Figure 1: Our IDS system utilized to evaluate the efficacy of $\mathcal{E}_{\mathcal{SB}}$. A user performs any action a_h, which updates the environment S. The underlying planner utilizes S to produce a plan solution π. The IDS system provides either a subgoal-baed explanation, $\mathcal{E}_{\mathcal{SB}}$, or action recommendation, a_{IDS}. If the user ignores the planner's suggestion, the planner will replan for a new $\pi^{'}$, repeating the process.

Problem Formulation

In this section, we first provide definitions for the planning problem that underlies our intelligent decision support system. We then formalize the problem of providing explanations in support of plan-based IDS and present our research hypotheses.

Planning Problem

A planning problem is defined by a model $\mathcal{M} = \langle \mathcal{D}, \mathcal{I}, \mathcal{G} \rangle$ where domain $\mathcal{D}$ is represented by $\langle F, A \rangle$, such that F is a finite set of fluents that define a state $s \subseteq F$, and A represents a finite set of actions. $\mathcal{I}$ and $\mathcal{G}$ represent the initial and goal states, respectively, such that $\mathcal{I}$,$\mathcal{G} \subseteq F$. Note that $\mathcal{G}$ may be modeled as a set of $\langle g_0 ... g_j \rangle$, where g_i represents a subgoal. An action $a \in A$ is defined by a tuple $\langle c_a, pre(a), eff^+(a), eff^-(a) \rangle$, where c_a is the associated cost of a, and $pre(a), eff^+(a), eff^-(a)$ denote the set of preconditions, add and delete effects, respectively. An action $a \in A$ can only be executed in a state s if $s \models pre(a)$. A transition function, $\delta_M(s, a)$ is used to transition an agent from $\mathcal{I}$ to $\mathcal{G}$, performing a sequence of actions $\langle a_1 ... a_n \rangle$, each with an associated cost c_a. In other words, the cost of plan $C(\pi, \mathcal{M})$ is defined by $\sum_{a \in \pi} c_a$, the sum cost of all actions within the plan, or ∞ if the goal is not met. The solution to a planning problem is a plan $\pi = \langle a_1 ... a_n \rangle$ such that $\delta_M(\mathcal{I}, \pi \models \mathcal{G})$. The optimal plan solution, π^*, is defined by $argmin_\pi \{C(\pi, \mathcal{M}) \forall \pi$ such that $\delta_{\mathcal{M}}(\mathcal{I}, \pi) \models \mathcal{G}\}$.

Explainability in Plan-Based IDS Systems

The goal of a plan-based IDS system is to provide the user with action recommendations $a_{IDS} \in \pi$. In turn, the user, who is given a_{IDS} as input, must select their own action a_h to take in response. In the ideal case, the IDS guides the user along some optimal plan π^* by always recommending an optimal action, $a_{IDS} = a^* \in \pi^*$, which results in the user always taking the optimal action, $a_h = a^* \in \pi^*$. However, there are two limitations to this idealized formulation.

First, in complex real-world scenarios an IDS systems may not be able to generate an optimal plan, resulting in suboptimal recommendations (Guerlain, Brown, and Mastrangelo 2000). In this case, the user is faced with the challenge to discern whether the IDS system's recommended action is optimal (i.e., $a_{IDS} \stackrel{?}{=} a^*$)). Relating to this, we state the following hypothesis:

> **H1:** We hypothesize that there exists a type of explanation, $\mathcal{E}$, that when presented in conjunction with a_{IDS} can aid users in determining $a_{IDS} \stackrel{?}{=} a^*$.

Specifically, that with the aid of $\mathcal{E}$, users will be able to accept a_{IDS} with greater accuracy when $a_{IDS} = a^*$ and correctly reject a_{IDS} when $a_{IDS} \neq a^*$.

Second, in complex real-world scenarios an IDS system may not always be available due to being offline, a failure, or the query being outside its scope. In this case, the user may suddenly be required to select a_h without the benefit of an IDS system's guidance. This scenario will pose a particularly significant challenge to users who had previously only performed the task under the support of an IDS system. Relating to this, we state the following hypothesis:

> **H2:** We hypothesize that exposure to explanations $\mathcal{E}$ improves user understanding of the task, such that when IDS recommendations are turned off, users with previous exposure to $\mathcal{E}$ will achieve greater task performance than users who had the same amount of domain experience but without exposure to $\mathcal{E}$.

Specifically, we posit that users previously exposed to IDS action recommendations will be able to select actions a_h that lead to more optimal task performance ($C(\pi_h^{\mathcal{E}}, \mathcal{M}) < C(\pi_h^{\not\mathcal{E}}, \mathcal{M})$) than users who were not exposed to explanation $\mathcal{E}$. In this perspective, explanations can be seen as a *training*

mechanism that leverages IDS to improve user understanding of the task.

Finally, prior work across many XAI applications has demonstrated that incorporating explanations into the output of automated systems improves user performance in a given task (Das and Chernova 2020; Tabrez, Agrawal, and Hayes 2019). Relating to this, we state the following two hypotheses in the context of IDS systems:

> **H3:** We hypothesize that user performance on the task will improve when IDS output, a_{IDS}, is supplemented with explanation $\mathcal{E}$.
>
> **H4:** We hypothesize that users will prefer the output of a system that includes $\mathcal{E}$ over a system that only includes a_{IDS}.

Specifically, we posit that overall plan cost for users exposed to explanations will be lower than for users who did not receive explanations (i.e., $C(\pi_h^{\mathcal{E}}, \mathcal{M}) < C(\pi_h^{\not\mathcal{E}}, \mathcal{M})$), and that users will prefer to see explanations as part of an IDS output.

In the following sections, we first present a novel variant of $\mathcal{E}$ for explaining action recommendations in IDS systems, leveraging findings in psychology (Newell, Simon et al. 1972). We then describe our validation domain, and the experiments that were conducted to support the above hypotheses.

Subgoal-Based Explanations

Research in psychology shows that humans faced with a complex sequential decision making task naturally construct a mental model of that task as a decomposition of multiple subgoals (Newell, Simon et al. 1972). Similarly, many AI techniques utilize underlying hierarchical structures in order to leverage the improved computational efficiency of such representations (Iovino et al. 2020; Kaelbling and Lozano-Pérez 2010).

In this work, inspired by the natural hierarchical representations used by both human users and AI systems, we introduce a new type of explanation known as subgoal-based explanations, $\mathcal{E}_{\mathcal{SB}}$. The objective of $\mathcal{E}_{\mathcal{SB}}$ is improve current IDS-supported user task performance both in optimal IDS as well suboptimal IDS settings. Below we further detail the definition of $\mathcal{E}_{\mathcal{SB}}$ by leveraging the definition of a planning problem (see previous section):

> $\mathcal{E}_{SB}$: Given planning goal $\mathcal{G}$, which is decomposed into subgoals $\langle g_0 ... g_n \rangle$, a subgoal-based explanation is described by $\mathcal{E}_{SB} = a_{IDS} + g_i(a_{IDS})$.

In other words, a subgoal-based explanation provides the next recommended action from an IDS system, a_{IDS} along with the associated subgoal g_i that is satisfied by a_{IDS}. An example of a subgoal-based explanation is "Chop the tomato to prepare the lasagna," where "to prepare the lasagna" is the subgoal. Plan subgoals for explanation generation may be predefined as part of the planning representation, or may be autonomously derived using one of a number of established methods (Richter, Helmert, and Westphal 2008; Czechowski et al. 2021). In this work, we encode the subgoals within our plan problem definition.

IDS in Restaurant Planning Domain

We investigate the efficacy of $\mathcal{E}_{\mathcal{SB}}$ explanations in the context of a complex sequential planning scenario: running a restaurant kitchen. In our task a user plays as a chef to deliver a set of M meals, each within unique delivery times, with the help of an anthropomorphized IDS known as Manager Molly. Below we further detail the restaurant planning game and our plan-based IDS system.

Restaurant Game Overview

Figure 1 provides a visualization of the restaurant game; inspired by the online game Overcooked, this domain has been studied extensively in the sequential decision making community (Carroll et al. 2019; Wu et al. 2021; Liu et al. 2020). Within the game, the user controls a chef avatar, and utilizes five meal prep stations to prepare M meals consisting of various ingredients. The game objective is to deliver the meals to restaurant customers within the designated meal prep time for each meal, denoted as $t^m_{goal\_delivery}$. The five meal prep stations are: *gather station*, *cutting station*, *cooking station*, *plating station*, and *delivery station*. The game state is represented as S, and displayed to the user in a user-friendly manner in the bottom panel of the game interface (see Fig. 1). Specifically, $S = \{S_l, S_i\}$ where S_l defines the location of the chef and each ingredient (i.e which station), and S_i defines the state of each ingredient,(i.e. chopped tomato, cooked chicken). The user is able to perform an action a^h from the action space $A = \{$*cut, move-chef, move-item, start-cook, end-cook, deliver*, *prepare-meal*$\}$, so long as the preconditions of a_h are met and that the effects of a_h result in a valid state S. The game does not allow players to perform an invalid action (e.g., preparing the steak meal without cooking the steak). When a_h is performed, game state S is updated and utilized by the underlying planner to provide a recommendation to the user, either in the form of a subgoal-explanation $\mathcal{E}_{\mathcal{SB}}$, or as an action recommendation a_{IDS}. Given a recommendation, the user has the choice to conform to a_{IDS} or select an alternative action (i.e. $a_h \neq a_{IDS}$). If the user selects an alternative action, the underlying planner in the game utilizes the updated state information to find a new plan solution π' from which subsequent actions are suggested to the user.

Restaurant Game Planner

To find a plan solution π, we utilize the Temporal Fast Downward (TFD) planner (Eyerich, Mattmüller, and Röger 2009). Although TFD is a temporal planner with abilities to handle durative actions, we formulate our planning problem as a classical planning problem in which the objective is to minimize action costs as opposed to duration. We utilize TFD to more accurately model a restaurant domain where actions can occur simultaneously. To simulate multitasking, we leverage TFD's support for numeric fluents which allows us to track the cost of actions $a \in A$ being performed while an ingredient is cooking to ensure that an appropriate duration has elapsed for an ingredient to cook.

Action ($a \in \pi$)	**Action-Based Recommendation** (a_{IDS})	**Subgoal-Based Explanation** ($\mathcal{E}_{\mathcal{SB}}$)
cut chef gatherStation cutStation tomato1	Chop the tomato.	Chop the tomato for the salad meal.
move-item chef cookStation plateStation salmon1	Move the cooked salmon to the plating station.	Move the cooked salmon to the plating station for preparing the teriyaki salmon meal.
end-cook chef cookStation broth1	Finish cooking the broth.	Finish cooking the broth for preparing the soup.
move-item chef cookStation plateStation salmon2	Move the salmon to the plating station.	Move the salmon to the plating station for preparing the sushi meal.

Table 1: Examples of action-based recommendations, a_{IDS} and subgoal-based explanations, $\mathcal{E}_{\mathcal{SB}}$ for providing select $a \in \pi$.

To solve for a plan solution π, the planner utilizes the same action space A and state space S as that available to the user. The planner's initial state $\mathcal{I}$ is defined with select, pre-performed actions to avoid plan solutions longer than 35 actions and the goal state $\mathcal{G}$ is defined by reaching the *delivered* state for all necessary M meals. Each action $a \in A$ has a static cost of c_a where the cost represents the time needed to perform a. If each meal m is delivered at $t^m_{delivered}$, the objective of the planner is to minimize the overtime delivery cost, $\sum_{m=1}^{|M|} t^m_{delivered} - t^m_{goal\_delivery}$.

Generating Suboptimal Plans

A central objective of our work is to examine explanatory action suggestions in the context of *suboptimal* IDS recommendations. To achieve this objective, we intentionally corrupt the optimal plan, π^*, generated by our planner such that the resulting plan $\tilde{\pi}$ is suboptimal. At run time, we randomly select with probability p whether a recommended action, a_{IDS}, is provided from an optimal or suboptimal plan. Specifically:

$$a_{IDS} = \begin{cases} a_{IDS} \in \tilde{\pi}, & \text{if } rand() \leq p \\ a_{IDS} \in \pi^*, & \text{otherwise} \end{cases}$$

Recall, the goal of the planner in our restaurant planning domain is to minimize the overtime in delivering each meal $m \in M$. Therefore, to generate a suboptimal plan $\tilde{\pi}$, we replace the optimal action of interest required for a particular meal m_i with a random action required for some other random future meal m_j. The resulting suboptimal action is therefore still relevant to the overall cooking task, and is not an obvious and trivially identifiable error (e.g., throw steak on floor). Reordering actions in this way is guaranteed to result in a suboptimal plan because it delays meals and leads meals to be completed out of order, causing the planner's overtime delivery cost to be non-zero.

Generating Subgoal-Based Explanations

Given a set of subgoals $\langle g_1 ... g_n \rangle$, we employ a post-hoc search to map actions $\langle a_0 ... a_n \rangle$ within π with a corresponding subgoal $g_i \in G$. In our work, subgoals are defined as the designated meal for which an action a is being performed. To present $\mathcal{E}_{\mathcal{SB}}$ in a manner understandable by novice users, we leverage natural language. We parse each a output in π for the contextual information the action a is acting upon. In our work, the contextual information corresponds to the *ingredient(s)* the action would be applied on or the *location* the action would be applied to. In this manner, we template our explanation as follows, *"⟨action⟩ the ⟨ingredients /location⟩ for ⟨$g_i(a)$⟩"*. Table 1 provides example explanations of our $\mathcal{E}_{\mathcal{SB}}$ explanations in comparison to action-based suggestions, a_{IDS}, that provide the next recommended action. Note, a_{IDS} is most closely modelled after the action-based suggestion feature in RADAR (Grover et al. 2020).

Study Design

Our primary goal is to evaluate the effect explanations $\mathcal{E}_{\mathcal{SB}}$ have on IDS-supported user task performance. For our analysis, we conducted a five-way, between-subjects study in which participants were asked to play the restaurant planning game. Specifically, we used a 2 x 2 factorial study design with one factor being the type of IDS (with a_{IDS} suggestions or $\mathcal{E}_{\mathcal{SB}}$ explanations) and the second factor being optimality of IDS recommendation (optimal IDS and noisy IDS). The fifth study condition was a baseline condition in which participants did not receive any help from an IDS. Below, we detail each study condition:

- None (Baseline): Participants receive no suggestions from an IDS system.
- $\pi(a_{IDS})$: Participants receive action recommendations from an optimal IDS system. This scenario is closely modeled after the action-based suggestion feature currently available in plan-based IDS systems (Grover et al. 2020).
- $\pi(\mathcal{E}_{\mathcal{SB}})$: Participants receive subgoal-based explanations from an optimal IDS system.
- $\tilde{\pi}(a_{IDS})$: Participants receive action recommendations from a suboptimal IDS system.
- $\tilde{\pi}(\mathcal{E}_{\mathcal{SB}})$: Participants receive subgoal-based explanations from a suboptimal IDS system.

The study consisted of three stages in which participants played a total of five games, each consisting of a unique set of M meals to prepare. Participants proceeded to the next game when they finished delivering all required M meals, or when time cost in a game reached 80, whichever came first. The study consisted of three stages: the familiarization stage, IDS stage and an assessment stage, detailed below.

Familiarization: The participants first played through an interactive tutorial which explained the components of the interface as the participants made a burrito meal. While the

interactive tutorial remained the same across all conditions, the None condition did not receive support from the IDS system, whereas all other conditions received their respective guidance from the IDS system. Participants also played a second game to get further acquainted with the system. In the practice round, participants were tasked with making two meals (BLT sandwich, hotdog), and participants received IDS based on their study conditions. The familiarization stage was designed to familiarize users with all aspects of the interface and to minimize learning effects in future games.

IDS: Participants played two more games, each with the objective of preparing three meals with the help from an IDS system (or no help in the None condition). These games were themed by cuisine: Italian Bistro (salad, pasta, veggie burger) and Asian Fusion (chicken quesadilla, soup, sushi). Prior to playing, participants were told that the anthropomorphized IDS system, Manager Molly, may provide suboptimal suggestions. The goals of the participants in all conditions were to delivery meals on time and to identify suboptimal suggestions (in the four conditions with IDS). Both games were counterbalanced, such that a random 50% of participants played the Italian Bistro game first, while remaining played the Asian Fusion game first. In the two suboptimal recommendation study conditions ($\tilde{\pi}(a_{IDS})$ and $\tilde{\pi}(\mathcal{E}_{\mathcal{SB}})$), $p = 0.85$, such that the accuracy of our IDS was 85%, and approximately 15% of recommendations viewed by the users were corrupted to be suboptimal [1].

Assessment: Participants in all five conditions played a final game, with **no support from the IDS system**. Similar to the IDS stage, the assessment game required delivering 3 meals (teriyaki salmon, steak & potatoes, chili). In this assessment, participants in all study conditions had one objective, which was to deliver meals by their designated delivery time. The goal of the assessment stage was to simulate a scenario where a failure occurs in an IDS system, and guidance is no longer available to a novice user. Our goal is to understand how previous exposure to IDS in an optimal or suboptimal setting may impact participant performance on a task when in the absence of any IDS.

Metrics & Hypotheses

To measure user task performance and overall understanding of the task, we evaluate three metrics:

- **User Plan Cost** (*UPC*): represents participant overtime cost in delivering meals *per game*. This metric is analogous to how the planner optimizes for the optimal plan solution. Below, M represents the total number of meals to complete for a game, $t^m_{delivered}$ represents the accumulated time cost at which meal m is delivered, and $t^m_{goal\_delivered}$ represents the designated time cost at which a meal should have been delivered.

$$UPC = \sum_{m=1}^{|M|} t^m_{delivered} - t^m_{goal\_delivery} \quad (1)$$

- **Optimal Action Conformance** (OAC%): represents the total percentage of optimal actions suggested from the IDS system that participants performed.
- **Suboptimal Action Avoidance** (SAA%): represents the total percentage of suboptimal actions suggested from the IDS system that participants avoided.
- **Perceived Preference** (*Pref%*): represents the total percentages of a_{IDS} or $\mathcal{E}_{\mathcal{SB}}$ IDS types preferred by participants for understanding the chef's next action.

Both *OAC* and *SAA* are measured during the *IDS Stage* of the user study, while *UPC* is measured for games within the *IDS Stage* and *Assessment Stage*.

Participants

We recruited 90 individuals from Amazon's Mechanical Turk. We filtered any participant that showed no effort to play the game in the form of taking repeated actions until the timeout. The filtration process yielded 75 remaining participants in which 40 were males and 35 were females. All participants were 18 years or older (M = 37.6, SD = 10.9). Each study condition had 15 participants. The task took on average 40 minutes and participants were compensated $5.00.

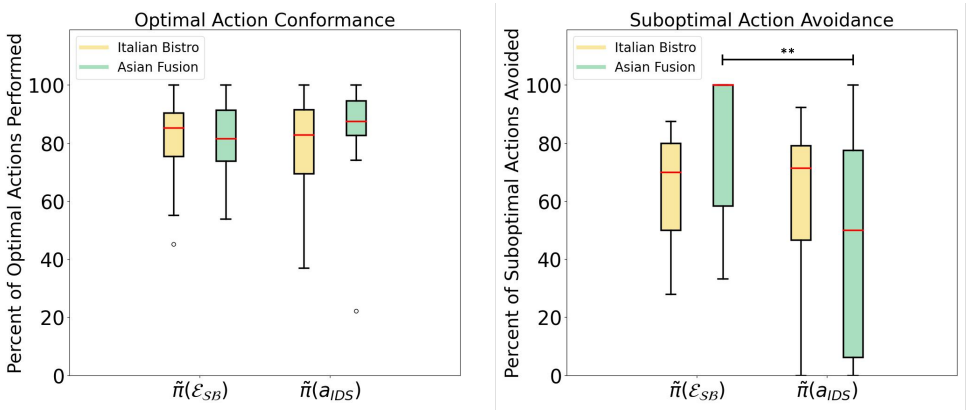
Study Results

The User Plan Cost (*UPC*) metric in Figure 3 and Figure 4 were analyzed with a one-way ANOVA, followed by a post-hoc Tukey Test. The Optimal Action Conformance (*OAC*) and Suboptimal Action Avoidance (*SAA*) metrics were analyzed with a two-tailed T-test with Bonferonni correction.

H1: In Figure 2, we present user optimal action conformance (*OAC%*) as well as suboptimal action avoidance (*SAA%*) in order to analyze the benefit of providing subgoal-based explanations, $\mathcal{E}_{\mathcal{SB}}$, for understanding suboptimality in IDS systems. We observe in Figure 2(a) that both the $\tilde{\pi}(a_{IDS})$ and $\tilde{\pi}(\mathcal{E}_{\mathcal{SB}})$ conditions had similarly high *OAC%*s. In other words, both a_{IDS} and $\mathcal{E}_{\mathcal{SB}}$ helped participants discern optimal recommendations. However, in Figure 2(b), we observe that participants in the $\tilde{\pi}(a_{IDS})$ condition had much lower *SAA%s* than participants in the $\tilde{\pi}(\mathcal{E}_{\mathcal{SB}})$ condition. In fact, in the Asian Fusion game, we observe a significant difference in *SAA%* between the $\tilde{\pi}(a_{IDS})$ and $\tilde{\pi}(\mathcal{E}_{\mathcal{SB}})$ conditions (t(16.6)=-3.97, p=0.001). **These results support H1**, indicating that in the context of an imperfect IDS system, subgoal-based explanations, $\mathcal{E}_{\mathcal{SB}}$, help participants detect and avoid more suboptimal suggestions from imperfect IDS systems compared to those who only receive a_{IDS}.

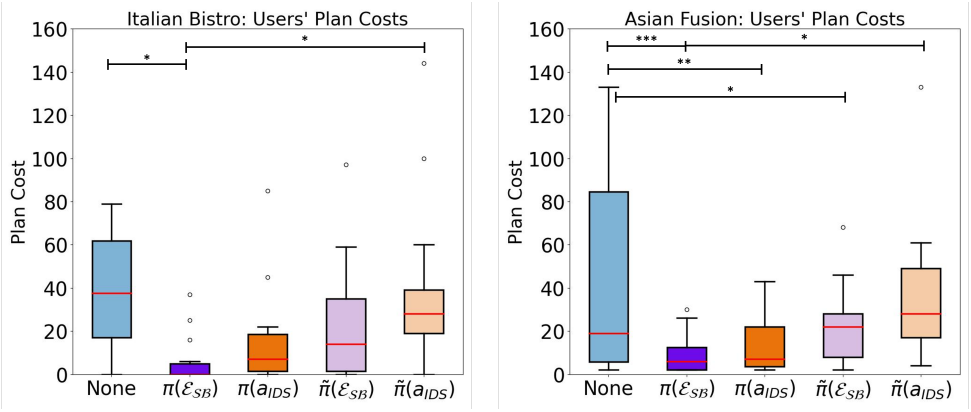
H3: In Figure 3, we present user plan costs (*UPC*) across each study condition in the two themed games to analyze the impact of including subgoal information on IDS-supported user performance. Overall, we observe that in both games

[1] Note that the level of acceptable error in a deployable IDS system varies significantly by application area (e.g., medical diagnosis systems may be expected to perform with greater accuracy than those in lower risk domains). Based on prior literature in the field, we find accuracy rates vary in the 75-95% range across various applications (Rodríguez, Gonzalez-Cava, and Pérez 2020; Rathore, Loia, and Park 2018; Rikalovic et al. 2017). We selected an accuracy of 85% as it approximates many state of the art systems.



(a) (b)

Figure 2: User optimal action conformance and action avoidance percentages for participants that received $\mathcal{E}_{\mathcal{SB}}$ and a_{IDS} from suboptimal IDS systems.



(a) (b)

Figure 3: User Plan Cost across all conditions for the two themed games within the *IDS Stage* of the user study. Statistical significance is reported as: *p <0.05, **p <0.01, ***p <0.001 for all figures.

participants in the $\pi(\mathcal{E}_{\mathcal{SB}})$ conditions, subgoal-based explanations from an optimal IDS system, have the best overall task performance in comparison to the other conditions. Specifically, participants in $\pi(\mathcal{E}_{\mathcal{SB}})$ had significantly lower *UPC* compared to participants in the None condition for both the Italian Bistro game (t(67)=-3.25, p=0.01) and the Asian Fusion game (t(67)=-4.28, p=0.0006). We additionally observe that participants in the $\pi(\mathcal{E}_{\mathcal{SB}})$ condition had significantly lower *UPC* than participants in the $\tilde{\pi}(a_{IDS})$ condition, action-based recommendations from a suboptimal IDS system, for the Italian Bistro game (t(67)=3.28, p=0.01) and Asian Fusion game (t(67)=2.84,p=0.04). Additionally, we see that in the Asian Fusion game, participants from the $\tilde{\pi}(a_{IDS})$ condition have significantly lower *UPC* in comparison to None (t(67)=-4.28, p=0.02). **These results support H3** by indicating that supplementing IDS action recommendation outputs, a_{IDS}, with explanations grounded in subgoal information help participants understand the underlying motivation for a suggestion and therefore perform the task significantly better those who only receive a_{IDS}.

H2: In Figure 4, we present user plan cost (*UPC*) from the Assessment stage across each study condition. None of the participants had access to IDS recommendations in this game, and the results allow us to assess how prior exposure to $\mathcal{E}_{\mathcal{SB}}$ impacts user performance once IDS recommendations are unavailable. Overall, we observe that both $\pi(\mathcal{E}_{\mathcal{SB}})$ and $\tilde{\pi}(\mathcal{E}_{\mathcal{SB}})$ conditions have the lowest *UPC* compared to the other study conditions. In fact, participants in the $\pi(\mathcal{E}_{\mathcal{SB}})$ condition, those previously received $\mathcal{E}_{\mathcal{SB}}$ explanations under an optimal IDS system, have significantly lower *UPC* than participants in the $\tilde{\pi}(a_{IDS})$ condition (t(67)=2.83, p=0.04). Similarly, we observe participants in the $\tilde{\pi}(\mathcal{E}_{\mathcal{SB}})$ condition, those who previously received $\mathcal{E}_{\mathcal{SB}}$ explanations under a suboptimal IDS system, also have significantly lower *UPC* than those who received $\tilde{\pi}(a_{IDS})$ (t(67)=2.87, p=0.04). **These results support H2**, demonstrating the important role of subgoal-based explanations, $\mathcal{E}_{\mathcal{SB}}$, in training users to understand the underlying task, compared to action-based rec-

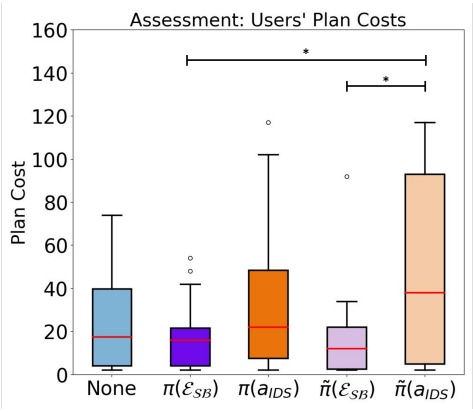


Figure 4: User Plan Costs for the assessment game in which all participants did not receive any guidance from an IDS system.

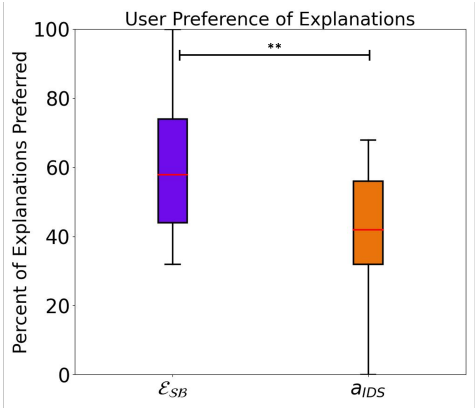


Figure 5: User perceived preferences towards both a_{IDS} and $\mathcal{E}_{\mathcal{SB}}$ in understanding the chef's next action.

ommendations a_{IDS}, even if IDS was suboptimal during training.

H4: To evaluate H4 we conducted an additional experiment in which we presented users with two explanation options, a_{IDS} and $\mathcal{E}_{\mathcal{SB}}$, side by side, and asked them to select their preferred explanation[2]. Specifically, each participant was presented with 25 randomly shuffled, pre-recorded videos of optimal actions the chef would perform while preparing meals in the restaurant game. Each video was 10-15 seconds in duration and included two actions that the chef performed towards the goal. Participants were tasked with watching each video, to gain contextual understanding of which portion of the task the chef was working on, and evaluate which form of the provided IDS, $\mathcal{E}_{\mathcal{AB}}$ or a_{IDS}, they preferred in understanding the chef's next action. Figure 5 presents the Perceived Preference (*Pref%*) metric results for the above experiment, which were analyzed with a paired-samples T-test. We observe that participants significantly preferred $\mathcal{E}_{\mathcal{SB}}$ explanations compared to $\mathcal{E}_{\mathcal{AB}}$ (t(18)=-3.61, p=0.002). **These results support H4** demonstrating that $\mathcal{E}_{\mathcal{SB}}$ explanations are more frequently preferred by users, a factor that may aid in adoption of XAI-based IDS systems.

Discussion

Our user study findings support H1-H4, demonstrating that subgoal-based explanations improve user task performance, improve user ability to distinguish optimal and suboptimal IDS recommendations, are preferred by users, and enable more robust user performance in case of IDS failure. We find the results from the Assessment stage of the study (H2) particularly surprising. All users received detailed task instructions and performed the same tutorials; the Assessment game was the 5th in the series of games, meaning that participants were reasonably familiar with the task at this stage. Yet subgoal-based explanations significantly impacted performance such that in the optimal IDS condition $\pi(\mathcal{E}_{\mathcal{SB}})$, and (more importantly) in the suboptimal condition $\tilde{\pi}(\mathcal{E}_{\mathcal{SB}})$, users learned the task objectives better than users in other explanation conditions. These results point to important benefits explanation-based IDS systems can have in real-world deployments, highlighting that even imperfect IDS systems can serve as a useful training tool for users when subgoal-based explanations are added to IDS output. To our knowledge, this is the first use of subgoal-based explanation in potentially unreliable IDS systems.

Limitations and Future Work

Our work has several limitations that present opportunities for future work. First, we conducted our study only with novice users[3]. Further studies should explore whether the observed benefits of $\mathcal{E}_{SB}$ hold for expert users. Second, we conducted our study over a limited period of time, and thus factors such as long-term learning effects, fatigue, and automation bias were not fully explored. Further work is needed to fully explore the effect that explanations have on long-term IDS deployment. Third, our subgoals were predefined. Coupling our approach with autonomously identified subgoals may yield new insights. Finally, further investigation is needed to see the benefits of subgoal-based explanations when subgoals are more complex and hierarchical. For example, in complex tasks there may exist multiple hierarchical goals, or actions may satisfy multiple goals simultaneously. A more extensive comparison of various types of explanations is required in this space.

References

Adebayo, J.; Muelly, M.; Liccardi, I.; and Kim, B. 2020. Debugging tests for model explanations. *arXiv preprint arXiv:2011.05429*.

Arnold, V.; Collier, P. A.; Leech, S. A.; and Sutton, S. G. 2004. Impact of intelligent decision aids on expert and

[2]Since our previous study was between-subjects and participants were only exposed to one type of IDS, $\mathcal{E}_{\mathcal{SB}}$ or a_{IDS}, we conducted an additional within-subjects study with 20 participants from AMT (Male=15, Female=5, mean=35.5, SD=6.8) to accurately measure user preference between the two types of IDS.

[3]Our domain differs from both real world cooking and the Overcooked online game, and thus has significant novelty to all users.

novice decision-makers' judgments. *Accounting & Finance*, 44(1): 1–26.

Carroll, M.; Shah, R.; Ho, M. K.; Griffiths, T.; Seshia, S.; Abbeel, P.; and Dragan, A. 2019. On the utility of learning about humans for human-ai coordination. *NeurIPS*.

Chakraborti, T.; Sreedharan, S.; and Kambhampati, S. 2020. The Emerging Landscape of Explainable Automated Planning & Decision Making. In *IJCAI*, 4803–4811.

Chakraborti, T.; Sreedharan, S.; Zhang, Y.; and Kambhampati, S. 2017. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. *arXiv preprint arXiv:1701.08317*.

Czechowski, K.; Odrzygóźdź, T.; Zbysiński, M.; Zawalski, M.; Olejnik, K.; Wu, Y.; Kucinski, L.; and Miłoś, P. 2021. Subgoal Search For Complex Reasoning Tasks. *NeurIPS*.

Das, D.; and Chernova, S. 2020. Leveraging rationales to improve human task performance. In *IUI*, 510–518.

Doshi-Velez, F.; and Kim, B. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Eyerich, P.; Mattmüller, R.; and Röger, G. 2009. Using the context-enhanced additive heuristic for temporal and numeric planning. In *ICAPS*.

Feng, J.; Shaib, C.; and Rudzicz, F. 2020. Explainable clinical decision support from text. In *EMNLP*, 1478–1489.

Grover, S.; Sengupta, S.; Chakraborti, T.; Mishra, A. P.; and Kambhampati, S. 2020. RADAR: automated task planning for proactive decision support. *Human–Computer Interaction*, 35(5-6): 387–412.

Guerlain, S.; Brown, D. E.; and Mastrangelo, C. 2000. Intelligent decision support systems. In *SMC*, volume 3. IEEE.

Gutiérrez, F.; Ochoa, X.; Seipp, K.; Broos, T.; and Verbert, K. 2019. Benefits and Trade-Offs of Different Model Representations in Decision Support Systems for Non-expert Users. In Lamas, D.; Loizides, F.; Nacke, L.; Petrie, H.; Winckler, M.; and Zaphiris, P., eds., *INTERACT*, 576–597. Springer International Publishing.

Hoffmann, J.; and Magazzeni, D. 2019. Explainable AI planning (XAIP): overview and the case of contrastive explanation. *Reasoning Web: Explainable Artificial Intelligence*, 277–282.

Iovino, M.; Scukins, E.; Styrud, J.; Ögren, P.; and Smith, C. 2020. A survey of behavior trees in robotics and ai. *arXiv preprint arXiv:2005.05842*.

Jones, R. W.; Mateer, J. E.; and Harrison, M. J. 2019. Malfunction transparency in clinical decision support systems: a classification approach. In *ICIEA*, 1354–1359. IEEE.

Kaelbling, L. P.; and Lozano-Pérez, T. 2010. Hierarchical planning in the now. In *Workshops at AAAI*.

Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; Viegas, F.; et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *ICML*, 2668–2677. PMLR.

Krarup, B.; Cashmore, M.; Magazzeni, D.; and Miller, T. 2019. Model-based contrastive explanations for explainable planning.

Liu, Y.; Chen, Q.; Jin, K.; and Zhang, Z. 2020. Planning for Overcooked Game with PDDL. *International Core Journal of Engineering*, 6(12): 315–325.

Machado, J. P.; Lam, X. T.; and Chen, J.-W. 2018. Use of a clinical decision support tool for the management of traumatic dental injuries in the primary dentition by novice and expert clinicians. *Dental Traumatology*, 34(2): 120–128.

Newell, A.; Simon, H. A.; et al. 1972. *Human problem solving*, volume 104. Prentice-hall Englewood Cliffs, NJ.

Nourani, M.; King, J.; and Ragan, E. 2020. The role of domain expertise in user trust and the impact of first impressions with intelligent systems. In *AAAI Conference on Human Computation and Crowdsourcing*, volume 8, 112–121.

Papamichail, K.; and French, S. 2000. Decision support in nuclear emergencies. *J. of hazardous materials*, 71(1-3): 321–342.

Rathore, S.; Loia, V.; and Park, J. H. 2018. SpamSpotter: An efficient spammer detection framework based on intelligent decision support system on Facebook. *Applied Soft Computing*, 67: 920–932.

Richter, S.; Helmert, M.; and Westphal, M. 2008. Landmarks Revisited. In *AAAI*, volume 8, 975–982.

Rikalovic, A.; Cosic, I.; Labati, R. D.; and Piuri, V. 2017. Intelligent decision support system for industrial site classification: A GIS-based hierarchical neuro-fuzzy approach. *IEEE Systems Journal*, 12(3): 2970–2981.

Rodríguez, G. G.; Gonzalez-Cava, J. M.; and Pérez, J. A. M. 2020. An intelligent decision support system for production planning based on machine learning. *Journal of Intelligent Manufacturing*, 31(5): 1257–1273.

Sreedharan, S.; Srivastava, S.; Smith, D.; and Kambhampati, S. 2019. Why Can't You Do That HAL? Explaining Unsolvability of Planning Tasks. In *IJCAI*.

Sutton, R. T.; Pincock, D.; Baumgart, D. C.; Sadowski, D. C.; Fedorak, R. N.; and Kroeker, K. I. 2020. An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ digital medicine*, 3(1): 1–10.

Tabrez, A.; Agrawal, S.; and Hayes, B. 2019. Explanation-based reward coaching to improve human performance via reinforcement learning. In *HRI*, 249–257. IEEE.

Valmeekam, K.; Sreedharan, S.; Sengupta, S.; and Kambhampati, S. 2020. RADAR-X: An Interactive Interface Pairing Contrastive Explanations with Revised Plan Suggestions. *arXiv preprint arXiv:2011.09644*.

Walsh, S.; de Jong, E. E.; van Timmeren, J. E.; Ibrahim, A.; Compter, I.; Peerlings, J.; Sanduleanu, S.; Refaee, T.; Keek, S.; Larue, R. T.; et al. 2019. Decision support systems in oncology. *JCO clinical cancer informatics*, 3: 1–9.

Wu, S. A.; Wang, R. E.; Evans, J. A.; Tenenbaum, J. B.; Parkes, D. C.; and Kleiman-Weiner, M. 2021. Too Many Cooks: Bayesian Inference for Coordinating Multi-Agent Collaboration. *Topics in Cognitive Science*, 13(2): 414–432.

Zhuang, Z. Y.; Churilov, L.; Burstein, F.; and Sikaris, K. 2009. Combining data mining and case-based reasoning for intelligent decision support for pathology ordering by general practitioners. *European Journal of Operational Research*, 195(3): 662–675.