



Deposited via The University of York.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/238030/>

Version: Accepted Version

Proceedings Paper:

VAZQUEZ FLORES, Grisel, Evangelidis, Alexandros, Shahbeigi Roudposhti, Sepeedeh et al. (Accepted: 2026) Mind the Prompt: Self-adaptive Generation of Task Plan Explanations via LLMs. In: SEAMS '26. (In Press)

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:

<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Mind the Prompt: Self-adaptive Generation of Task Plan Explanations via LLMs

Gricel Vázquez
University of York, UK
gricel.vazquez@york.ac.uk

Alexandros Evangelidis
University of York, UK
alexandros.evangelidis@york.ac.uk

Sepeedeh Shahbeigi
University of York, UK
sepeedeh.shahbeigi@york.ac.uk

Radu Calinescu
University of York, UK
radu.calinescu@york.ac.uk

Simos Gerasimou
Cyprus University of Technology, Cyprus
University of York, UK
simos.gerasimou@cut.ac.cy

Abstract

Integrating Large Language Models (LLMs) into complex software systems enables the generation of human-understandable explanations of opaque AI processes, such as automated task planning. However, the quality and reliability of these explanations heavily depend on effective prompt engineering. The lack of a systematic understanding of how diverse stakeholder groups formulate and refine prompts hinders the development of tools that can automate this process. We introduce COMPASS (COgnitive Modelling for Prompt Automated SynthesiS), a proof-of-concept self-adaptive approach that formalises prompt engineering as a cognitive and probabilistic decision-making process. COMPASS models unobservable users' latent cognitive states, such as attention and comprehension, uncertainty, and observable interaction cues as a POMDP, whose synthesised policy enables adaptive generation of explanations and prompt refinements. We evaluate COMPASS using two diverse cyber-physical system case studies to assess the adaptive explanation generation and their qualities, both quantitatively and qualitatively. Our results demonstrate the feasibility of COMPASS integrating human cognition and user profile's feedback into automated prompt synthesis in complex task planning systems.

Keywords

Task planning, prompt generation, probabilistic verification, LLMs

ACM Reference Format:

Gricel Vázquez, Alexandros Evangelidis, Sepeedeh Shahbeigi, Radu Calinescu, and Simos Gerasimou. 2026. Mind the Prompt: Self-adaptive Generation of Task Plan Explanations via LLMs. In *21st International Conference on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '26)*, April 13–14, 2026, Rio de Janeiro, Brazil. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3788550.3794879>

1 Introduction

Large Language Models (LLMs) are increasingly integrated into software-intensive systems, providing new pathways for automation and effective collaboration between humans and AI [89]. These

pathways have the potential to revolutionise diverse industrial sectors, from reducing administrative burdens by automating medical documentation and subscription in healthcare [16] to streamlining decision-making processes by generating detailed reports and plans from raw data in finance [48], e-commerce [76] and agriculture [71]. In such complex data-intensive computational processes, however, generating human-understandable explanations that help users understand these otherwise opaque processes is equally important [36]. These explanations are vital for ensuring transparency, trustworthiness, and effective human oversight, especially when the underlying AI components (used for task scheduling or probabilistic planning) are inherently opaque or difficult to interpret [46].

The quality, reliability and appropriateness of LLM-driven explanations critically rely on how prompts, i.e., the textual instructions guiding the LLM behaviour, are formulated and adapted to diverse user needs and contexts [20]. Despite growing interest in devising effective prompts, recent approaches rely on human intuition, ad hoc experimentation, or simplistic heuristics [44]. The absence of a principled understanding of how different system stakeholders, e.g., AI experts, roboticists, project managers, and non-expert end users, iteratively construct, revise, and assess prompts hinders further LLM integration into adaptive systems.

Since these systems are typically deployed in operational contexts involving diverse stakeholders, their informational needs, cognitive capacities, and interpretive preferences vary considerably [39]. Inevitably, these stakeholders require distinct explanation forms (varying in granularity, tone, and structure) to effectively comprehend the same information (e.g., a scheduling plan assigning tasks to a human-robot team). Also, user comprehension and attention dynamically vary, depending on cognitive factors such as workload, fatigue and domain familiarity [58]. For instance, explanations deemed appropriate in a previous interaction of a roboticist may become suboptimal or cognitively burdensome in another interaction occurring late during the shift of the same roboticist. Hence, static or generic prompts risk yielding explanations that are overly technical, insufficiently informative, or misaligned with the user's cognitive state. Addressing this challenge entails self-adaptation mechanisms that continuously align explanation generation with user feedback and cognitive variability [46, 54].

While stakeholders could remain involved and explicitly articulate their explanation preferences, reliance on explicit preference elicitation places a non-trivial cognitive burden on users [40]. Also, well-documented cognitive biases, like asymmetric perception of



This work is licensed under a Creative Commons Attribution 4.0 International License. *SEAMS '26, Rio de Janeiro, Brazil*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2445-9/2026/04

<https://doi.org/10.1145/3788550.3794879>

losses and gains, and hyperbolic discounting of future rewards, often result in inconsistent and context-dependent preference elicitation [53]. Accordingly, a self-adaptive explanation system could mitigate these cognitive and status-quo biases [37] by leveraging user profiles, prior knowledge of explanation preferences, and anticipated levels of understanding and attention to generate planner explanations without requiring additional input. Although feedback from different stakeholders can be incorporated to refine and update this model, such feedback should remain intentionally optional [21]. This design choice reflects realistic deployment conditions, where users may be unwilling or unable to provide feedback (e.g., in call-centre settings, feedback requests are common, yet only a subset of users completes the corresponding feedback forms).

We introduce COMPASS (COgnitive Modelling for Prompt Automated Synthesis), a self-adaptive approach that formalises prompt engineering as a cognitive and probabilistic decision-making process. COMPASS integrates probabilistic reasoning and cognitive modelling to support the self-adaptive generation of task plan explanations using LLMs. More specifically, COMPASS leverages a Partially Observable Markov Decision Process (POMDP) to model the user's latent, probabilistic cognitive state, comprising unobservable variables such as attention and understanding, along with observable feedback from user interaction. The POMDP analysis enables synthesising optimal prompt policies (comprising desired tone, level of detail, and structure) that maximise the expected acceptance and effectiveness of generated explanations by the target user. COMPASS dynamically synthesises explanations, adapting their tone, structure, and granularity to the evolving cognitive profile and contextual user needs.

Our COMPASS evaluation is based on two case studies in the construction and agriculture cyber-physical system (CPS) domains. We leverage both proprietary (OpenAI GPT-5, Google Gemini-2.5-Pro) and open-source (DeepSeek-V3.2) LLMs. Our results demonstrate the feasibility of integrating LLMs to automatically generate planning problems, solutions, and explanations from natural language descriptions with minimal human input. Further, through a user study with 32 participants, we extract initial insights into human perception of different automatically generated explanations, and the effectiveness of adapting them to match user preferences and different cognitive states. We consider these results as the basis for performing a more in-depth and longitudinal study on the effect of LLM-driven task plan explanations. We make the following concrete contributions:

- A conceptual formalisation of prompt generation for task planners' explanations as a POMDP that captures both observable user interactions and unobservable cognitive states;
- The COMPASS self-adaptive explanation approach that dynamically adjusts LLM prompts and explanations based on user feedback, cognitive predictions, and contextual changes in task planning;
- A benchmark of 72 explanations tailored to three stakeholder profiles, with variations in tone, level of detail and format automatically generated state-of-the-art LLMs (OpenAI GPT-5, Google Gemini-2.5-Pro, DeepSeek-V3.2); and
- The COMPASS proof-of-concept validation in the CPS case studies of construction and agriculture, showing how adaptive explanations can influence alignment, personalisation, and quality across diverse user profiles.

Table 1: Task cost, success probability, duration, and maximum number of retries per task and agent.

Agent (initial loc.)	Tasks	Cost	Success prob.	Duration	Max. retries
r1 (Room B)	[$t1, t3$]	[2,5]	[0.99,0.95]	[12,40]	[4,2]
r2 (Room C)	[$t1$]	[2]	[0.98]	[13]	[4]
r3 (Room J)	[$t4$]	[1]	[0.99]	[18]	[5]
h1 (Room H)	[$t2, t3$]	[8,10]	[0.97,0.98]	[25,35]	[2,1]

2 Construction CPS Mission

Planning Problem. We motivate COMPASS using a CPS construction scenario provided by our industrial partner. The scenario concerns task automation of a multi-level office building construction project, whose layout and tasks are obtained from a Building Information Model (BIM) [9]. The construction site comprises 10 distinct rooms (A-J) and involves interdependent tasks requiring coordinated execution between autonomous robots and a human worker. The tasks include:

- Foundation preparation - $t1$ (rooms F and G): forming the initial structural base of the building, requiring heavy-duty work and precision, and assigned to robotic agents.
- Electrical installation - $t2$ (room H): laying out and connecting the building's electrical wiring and systems, requiring human expertise.
- Plumbing installation - $t3$ (rooms D and E): including the installation of pipes and fixtures for water distribution and drainage systems, assigned either to a specialised robot or the worker.
- Finishing work - $t4$ (rooms I and J): involving precision activities like painting and installing fixtures, delegated to robots.

The CPS team comprises three robots ($r1, r2, r3$) and a worker ($h1$), each initially positioned at a distinct location on the construction site. Tasks can be retried a finite number in case of failure. Movement between any two connected locations incurs a uniform cost of 10 units. Table 1 shows task costs, success probabilities, durations, and allowed retries associated with each task. Costs are abstract units denoting resources consumed (e.g., energy, time).

Addressing this problem entails producing a plan that maximises the probability of successful mission completion while minimising operational costs. We assume that an AI planner [62, 82] is employed to synthesise effective mission plans such that the minimum overall mission success probability is 95%.

Adaptive Explanation Problem. The CPS mission involves heterogeneous stakeholder groups with distinct expertise and cognitive capabilities, categorised into the following representative profiles (i) AI planning and robotics experts; (ii) domain experts (finance and administrative personnel); and (iii) non-experts and end users. These groups differ significantly in their familiarity with planning models, risk interpretation, and technical understanding. Hence, the investigated problem considers the adaptive generation of profile-tailored explanations for synthesised task plans. Each explanation must consider the estimated likelihood of a user accepting or rejecting a given explanation, and predictions of the user's cognitive state, i.e., their levels of attention and understanding.

When a user requests a new explanation, feedback from prior interactions associated with that profile should be used to refine subsequent generations. Accordingly, solving this problem entails continuously updating the explanation synthesis process as new feedback and state predictions become available, yielding a closed-loop process for adaptive and tailored explanation generation.

3 Background

LLM. Large Language Models (LLMs), an application of generative AI (GenAI), are developed using probabilistic methods to generate human-like text based on extensive and varied datasets [67]. These models operate by estimating the likelihood of the next token in a sequence, given the previous context, an approach known as auto-regressive generation [66, 84]. Widely used LLM-based tools include OpenAI’s ChatGPT [60, 61], Google’s Gemini [31], and DeepSeek [49]. Prompting strategies for LLMs include zero-shot, one-shot and few-shots prompting, where none, a single or multiple examples are provided to guide the generated output [68].

Markov decision process (MDP). A MDP is a stochastic model represented by the tuple $M = (S, s_I, A, P, L, R)$, comprising a finite set of states S ; an initial state $s_I \in S$; a finite set of actions $A \neq \emptyset$; a partial probabilistic transition function $P : S \times A \rightarrow \text{Dist}(S)$ that maps state-action pairs to discrete probability distributions over S ; a labelling function $L : S \rightarrow 2^{AP}$ that maps each state to a set of atomic propositions AP ; a reward function R , defined as a tuple $R = (r_{\text{state}}, r_{\text{action}})$ comprising a state reward function $r_{\text{state}} : S \rightarrow \mathbb{R}_{\geq 0}$ and an action reward function $r_{\text{action}} : S \times A \rightarrow \mathbb{R}_{\geq 0}$.

POMDP. A POMDP is a tuple $M = (S, s_I, A, P, L, R, O, \text{obs})$ where (S, s_I, A, P, L, R) is an MDP; O is a finite set of observations; $\text{obs} : S \rightarrow O$ is a labelling of states with observations; such that, for any states $s, s' \in S$ with $\text{obs}(s) = \text{obs}(s')$, their available actions must be identical, i.e., $A(s) = A(s')$. States in S cannot be directly determined, only their corresponding observation $\text{obs}(s) \in O$ [59]. An observation-based strategy of a POMDP M is a function $\sigma : \text{FPaths}_M \rightarrow \text{Dist}(A)$ such that σ is a strategy of the MDP (S, s_I, A, P, L, R) ; and for any paths $\pi = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \dots s_n$, and $\pi' = s_0 \xrightarrow{a'_0} s_1 \xrightarrow{a'_1} \dots s_n$ satisfying $\text{obs}(s_i) = \text{obs}(s'_i)$ and $a_i = a'_i$ for all i , we have $\sigma(\pi) = \sigma(\pi')$.

Probabilistic Computation Tree Logic (PCTL). PCTL [7, 32] is used to express quantitative properties over probabilistic models. State formulae are defined as: $\Phi ::= \text{true} \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid P_{\bowtie q}[\varphi] \mid R_{\bowtie q}[\rho]$, where $a \in AP$ is an atomic proposition, $\bowtie \in \{<, \leq, >, \geq\}$, $p \in [0, 1]$, $q \geq 0$, and q denotes a reward or cost structure. Path formulae are defined as: $\varphi ::= X\Phi \mid \Phi_1 U^{\leq k} \Phi_2 \mid \Phi_1 U \Phi_2$. Operators such as eventually ($F\Phi$) can be derived from this ($\text{true} U \Phi$). Query formulae such as “what is the maximum/minimum probability of eventually reaching a state satisfying the predicate goal” ($P_{\max/\min=?}[F \text{goal}]$), can also be defined for rewards $R_{\max/\min=?}$.

4 COMPASS

An overview of COMPASS, which comprises six steps (1-6), is shown in Figure 1. These steps are divided into two stages: **(S1)** for the generation of task plans given the problem description in natural language, and **(S2)** for the adaptive synthesis of human-readable profile-specific explanations of synthesised task plans. Table 2 details the terminology used throughout to present COMPASS.

4.1 Problem Generation (S1)

Stage (S1) begins by receiving a planning problem described in natural language, assembled from various stakeholders. COMPASS utilises GenAI and off-the-shelf AI planners [82, 88] to convert this

Table 2: Terms and explanations used in COMPASS

Terminology	Definition
user	An individual interacting with COMPASS
profile	A specific role or expertise of a user (e.g., researcher, roboticists, software developer, policymaker, educator, student).
profiled explanations	Explanations tailored to different users, e.g., generated using GenAI.
cognitive skills	Mental abilities, e.g., reasoning, understanding and attention.
cognitive skill level	Discrete level of a cognitive skill (e.g., low, medium, high).
cognitive state	The mental condition of the user, defined by the <i>levels</i> of their cognitive skills (e.g., low attention and high understanding levels).
prompt	A piece of text or instruction given to a language model to elicit a specific response or generate content (see [85]).
slots	A placeholder in a prompt that is completed through <i>slot filling</i> [47].
explanation	A GenAI generated summary that describes the reasoning behind the solution to a task planning problem in the context of COMPASS.

textual input into executable plans. First, our GenAI-based **Planning Problem Generator (1)** receives three input files ($h1, i1, i2$), and generates the output of the selected planner ($o1$):

- **Planning problem in natural language ($h1$):** A description of the planning problem specified by stakeholders, including information about the available robots and human agents, tasks, locations, paths, and mission constraints. For instance, the description of the construction planning problem from Section 2 is shown in Figure 2.
- **One-Shot Planner input example ($i1$):** A well-structured planning problem example that can be processed by the AI Planner, serving as a one-shot structural template [68] and defining the correct syntax and accepted format expected by the AI Planner. This input is mandatory since the expected format varies significantly across different planners. For instance, classical planners consume problem definitions written in the Planning Domain Definition Language (PDDL) [25, 92], modern frameworks like the Unified Planning library accept Python-based problem specifications [70], while other planners may utilise custom DSLs and formats such as JSON [88].
- **GenAI instructions ($i2$):** A crafted prompt with instructions for the Problem Generator to output a file aligned with the planner input example format $i1$. Figure 3 shows an example set of instructions to generate a JSON-formatted planner input. These instructions detail the role of the GenAI component (lines 1-3), the supporting documents (lines 5-7), the overall goal (lines 9-11), and a breakdown of the different content parts of the expected output (lines 13-14). Validity checks assess the completeness of $i1$ and compliance with the expected planner input (lines 17-27). For example, lines 18-19 instruct the execution of a typical validity check to ensure that the minimum required success probability is specified in $i1$. Finally, error handling instructions are also included (lines 26–28).
- **Planner input ($o1$):** The output from the Planning Problem Generator is a structured, machine-readable version of the natural language problem. The output can be represented as a function from tokens-to-tokens $o1 = f1(h1, i1, i2)$. Since this output is a direct product of a GenAI model, it is available for expert review to verify its correctness before further processing in COMPASS Stage 2.

Next, the **Adaptive-AI Planner (2)** generates one or multiple feasible task plans ($o2$) using a selected AI planner, represented as a function $o2 = f2(o1)$. Example planners that can be utilised include fast downward [33], NSPH [72], and Tamer [57, 82]. Within COMPASS, the AI planner is treated as a black-box, making our approach generic and agnostic to the selected planner. Furthermore, the AI planner might internally have incremental mechanisms to generate new plans on-the-fly by adapting existing plans as changes or failures of the system are detected [62]. This is particularly useful

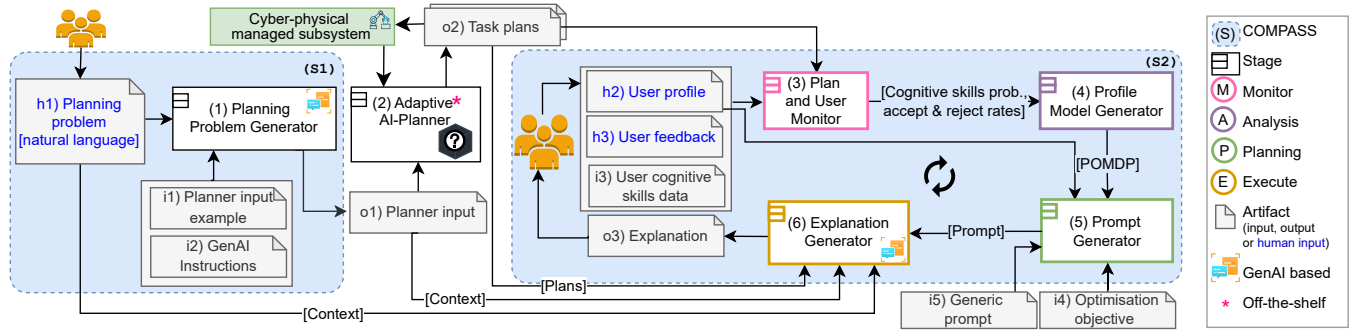


Figure 1: COMPASS stages and artefacts, showing the adaptation of explanations (S2) positioned within the MAPE-K loop.

1 This scenario outlines a complex planning problem for a new multi-level office building construction
 2 project. The site has been cleared, and the initial phase of construction is ready to begin. The project
 3 involves a variety of tasks that require coordination between human workers and autonomous robots to
 4 ensure timely and successful completion. The site is divided into several distinct zones, each with a
 5 specific purpose, and connected by a network of paths for movement of personnel and equipment.
 6
 7 **Locations.** The construction site is mapped into ten key locations. l1: "Room A" at coordinates (5, 5)
 8 ...
 9
 10 **Paths.** For simplicity, all direct paths between connected locations have a uniform distance of 10. The
 11 site is connected via the following bidirectional paths. A path connects Room B... to Room A
 12 ...
 13 **Tasks.** There are four types of construction tasks required for the project. t1: "Foundation
 14 preparation," with a base duration of 15. There are two instances: t1_msa at Room F (l6) and t1_msb
 15 at Room G (l7). t2: "Electrical installation,"
 16 ...
 17 **Agents.** The workforce consists of three autonomous robots and one human electrician. Robot r1:
 18 Starts at Room B (l2). It is capable of performing "Foundation preparation" (t1) and "Plumbing
 19 installation" (t3). For t1, its success probability is 0.99, cost is 2, duration is 12, and it has 4
 20 retries. For t3,
 21 ...
 22 **Constraints.** The project plan requires a minimum overall probability of success of 0.95...

Figure 2: Planning problem description of scenario in Section 2 provided in natural language by stakeholders (h1).

```
1 ## Instructions
2 You are an expert AI assistant tasked with generating structured JSON files for a complex planning
3 and scheduling system.
4
5 You will be provided with two key inputs:
6 1. A single example of a valid JSON file that conforms to the required schema.
7 2. A natural language description of a new planning problem scenario.
8
9 Your task is to generate a single, complete, and valid JSON file that accurately represents the new
10 planning problem described in the natural language text. The generated JSON must strictly adhere
11 to the structure and schema of the provided example.
12
13 ## JSON Schema Breakdown
14 The JSON file has five main top-level keys: locations, paths, tasks, agents, and constraints ...
15
16 ## Additional Rules:
17 For constraints:
18 1. If the description of a new planning problem scenario does not contain a minimum probability of
19 success required, assign "mission_probability_of_success": 0.0.
20 ...
21 For paths:
22 3. If the description of a new planning problem scenario does not contain any paths, define a path
23 between all given locations...
24
25 ## Errors
26 If one of the following errors is identified in the description of a new planning problem scenario, do
27 not create a JSON file. Instead create a new Error.txt file with feedback on the error.
28 1. The description does not contain any agents...
```

Figure 3: Example of GenAI instructions (i2).

when system changes are localised (e.g., in the plan minimum probability of success requirement); AI planners can leverage this aspect to incrementally and efficiently generate new plans by building on or adapting previously computed plans [22, 88].

The successful generation of feasible task plans concludes stage S1. These plans are ready for deployment, and can be inspected by a human-in-the-loop or directly consumed and executed by the subject CPS [15, 90]. The specific deployment mechanisms are beyond the scope of this work.

4.2 Adaptive Synthesis of Plan Explanations (S2)

During deployment in Stage (S2), the **Plan and User Monitor** (3) COMPASS component monitors and responds to user requests for

new explanations. The objective of this stage is to generate explanations (o3) from the available plans, considering the user's preferences. The stage starts by monitoring updates on the generated task plans, and changes to the following user data:

- **User profile (h2):** This encapsulates the characteristics of the end-user seeking an explanation (Table 2). Through profiling, COMPASS categorises individuals into distinct groups to enable modelling and adapting explanations based on each group's common expectations. For instance, robotics experts might prefer detailed insights into the sequence of tasks assigned to the deployed robots, whereas finance experts may prioritise explanations centred on plan-related costs.
- **User feedback (h3):** User feedback on preferred explanations is monitored and accumulated across profiles based on whether an explanation is accepted or rejected by a user. Prior knowledge based on past experience is used to set the baseline for h3 (informing the initial initial POMDP model reward structure as explained later), while new feedback collected at runtime guides COMPASS in tailoring explanations for different profiles.
- **User cognitive skills data (i3):** In addition to user feedback h3, COMPASS assumes the availability of a cognitive skills predictor, likely through using a Deep Learning model [27], that captures the probability that a user is in a specific cognitive level (e.g., high, medium or low attention level). Following research like DeepDecs [12, 13], these probabilities can be derived from the confusion matrix of DNN predictors. This information is quite important and enable to combine user feedback (explicitly provided by the user themselves) with data predicted from DNNs trained on relevant data (capturing intangible and difficult to express information).

Concerning the cognitive skills predictor, a substantial body of research in the ergonomics and human factors literature addresses the estimation of users' cognitive states, such as attention and mental workload, from observable data [83]. Prior work has explored how such cognitive variables may be operationalised and inferred using combinations of behavioural indicators and physiological signals [4], task performance metrics, interaction logs, and non-invasive sensing modalities such as eye-tracking or heart-rate measures [23]. Recent approaches leverage data-driven techniques to estimate cognitive states, with convolutional neural networks also being explored for this purpose [74]. COMPASS leverages this past research and assumes the availability of such a predictor [27].

4.2.1 Profile POMDP Model Generation. COMPASS frames prompt generation as a sequential decision-making problem under several uncertainty sources: (i) uncertainty in the user's latent cognitive

state (e.g., level of understanding, attention, or cognitive load); (ii) uncertainty due to imperfect predictors that estimate such cognitive states; and (iii) uncertainty in the user’s explanation acceptance, which may vary across users under the same observable profile.

A key modelling consideration is that a user’s internal cognitive condition cannot be directly observed or extracted explicitly [53]. Instead, the system relies on noisy and partial observations derived from users’ believed state, interactions and feedback. A POMDP naturally formalises this setting by: (i) representing unobservable cognitive variables as a hidden state; (ii) maintaining a belief state, i.e., a probability distribution over possible cognitive states, rather than committing to a single estimate; and (ii) modelling how prompt design choices (e.g., tone, level of detail, structure) are influenced by past acceptance data and the user’s underlying cognitive state. Simpler modelling alternatives, such as fully observable MDPs, rule-based systems, heuristic policies or contextual bandits [43] entail direct access to users’ internal cognitive states or strong assumptions that the relationship between observations and actions is deterministic and well understood.

To generate explanations, the **Profile Model Generator (4)** constructs a profile-specific POMDP model representing the user’s cognitive skills and the likelihood of accepting or rejecting a generated explanation. In this model, the true cognitive skill level of the user are hidden variables; only the predicted by the DNN levels (*ie*) are known. The model further integrates user feedback on the acceptance rates based on previously presented explanations. The policy obtained by solving the POMDP denotes the prompt elements that inform the LLM to produce an explanation appropriate to the user’s profile and its anticipated cognitive skills level.

In the POMDP, the user’s cognitive skill levels are not directly observable. Instead, a *belief* state is maintained and the presumed levels are derived from an imperfect DNN. We define a cognitive state as a collection of k , $1 \leq k \leq K$ skills, where $cogSkill_k$ represents the k -th real cognitive skill and $cogSkillPred_k$ its predicted level, both classified into M discrete levels. For example, if $cogSkill_1$ is attention and $M = 3$, we can classify a user’s attention level into low, medium or high. **Listing 1** defines these predictions as observable variables in the PRISM language. A support variable $step$ is used to control the execution flow of the POMDP.

The *Turn* module, shown in **Listing 2**, controls the model evolution by allowing only one action at a time. Let P be the number of prompt elements (slots) to fill in a prompt by solving the POMDP. • $S_{total} = K + P + 1$ is the number of transitions in the POMDP that control the flow of the model. The +1 adds an end state to ensure the inclusion of the final transition reward (self-transitions of terminal sink states are excluded).

• $done$ is a Boolean formula that denotes the termination point of the model flow, occurring when “the prompt has been filled”.

The *Turn* module synchronises with other modules in the POMDP and manages the flow of actions using the counter variable $step$. Synchronisation occurs through transitions sharing the same action label, denoted within square brackets ‘[]’, e.g., “ObserveCogSkill_1”. The first K transitions set the K cognitive skill levels of the user; the following P transitions correspond to informing the policy synthesis to select prompt elements.

The *Cognitive skill level prediction* module (**Listing 3**) selects the user’s unobservable and predicted cognitive skill levels. Given user

```
1 pomdp
2 observables
3   step, cogSkillPred_1, ..., cogSkillPred_K
4 endobservables
```

Listing 1: Observable state variables.

```
1 const int S_total = K + P + 1; // total number of transitions
2 formula done = step = S_total; // end of model flow
3 module Turn
4   step: [0..S_total] init 0;
5   [ObserveCogSkill_1] step=0 → 1:(step'=1);
6   ...
7   [ObserveCogSkill_K] step=k-1 → 1:(step'=k);
8   [SelectPrompt_1_1] step=k → 1:(step'=k+1);
9   ...
10  [SelectPrompt_p_q] step=k+p-1 → 1:(step'=k+p);
11  [End] step=S_total-1 → 1:(step'=S_total);
12 endmodule
```

Listing 2: Turn module controls the actions flow.

```
1 const int profile_1 = 1;
2 ...
3 const int profile_N = N;
4 const int profile; // current user profile
5 const int level_1 = 1; // cognitive level (low)
6 ...
7 const int level_M = M; // cognitive level (high)
8 // For profile 1, Cognitive skill 1:
9 const double P_1_1_L1L1, P_1_1_L1L2, ..., P_1_1_LMLM;
10 ...
11 // For profile N, Cognitive skill K:
12 const double P_n_K_L1L1, P_n_K_L1L2, ..., P_n_K_LMLM;
13 module CognitiveSkillLevelPrediction
14   cogSkill_1: [1..M]; // true, hidden cognitive skill
15   cogSkillPred_1: [1..M]; // observable prediction
16   ...
17   cogSkill_K: [1..M];
18   cogSkillPred_K: [1..M];
19   // Probabilistic skill transitions based on profile
20   // For each profile n from 1 to N, and each cognitive skill
21   // k from 1 to K:
22   [ObserveCogSkill_k] profile=profile_n →
23     P_n_k_L1L1:(cogSkill_k'=level_1)&(cogSkillPred_k'=level_1)
24     +
25     P_n_k_L1L2:(cogSkill_k'=level_1)&(cogSkillPred_k'=level_2)
26     +
27     ...
28     P_n_k_LMLM:(cogSkill_k'=level_M)&(cogSkillPred_k'=level_M);
29 endmodule
```

Listing 3: Cognitive skills prediction module.

profile n (lines 1-3), cognitive skill k with a value of M possible levels (lines 5-7), this module defines the joint probability $P_{n,k,LiLj}$

$$P_{n,k(i,j)} = P(cogSkill_k = level_i, cogSkillPred_k = level_j \mid profile_n)$$

These values (lines 8-12) represent the probability that the user truly has cognitive skill k at level i , while the predicted level for the same skill is j (lines 8-12). The Cognitive Skills Prediction module consists of state variables for each cognitive skill level ($cogSkill_k \leq M$) and its prediction ($cogSkillPred_k \leq M$) (lines 14-18). Transitions in this model are used to probabilistically set the user’s skill levels and predicted level (lines 19-25). Only transitions matching the user profile are available, as specified by the guard $profile = profile_n$ (line 21). Lines 22-25 show the selection of one out of the M possible cognitive skill levels (for both the true and predicted values) of cognitive skill k , for a user with profile n .

By building and analysing the POMDP model, COMPASS finds an optimal policy that, given a user profile and the user’s skill level, maximises the acceptance likelihood of an explanation generated from a prompt. An action in such a policy represents the selected value of a prompt element (i.e., a prompt slot).

The “acceptance” reward structure (**Listing 4**) defines the utility gained when a user accepts a specific prompt option, conditional on

their profile and cognitive skill levels. Let $U = cogSkill_1, \dots, cogSkill_K$ be the vector representing the user's K cognitive skill levels. The model assigns a reward based on the function $Utility(n, p, q, U)$, representing the utility for a user with profile n accepting option q for prompt slot p , given their true cognitive skill levels are represented by the vector U (with values from 1 to M).

The reward "acceptance" structure assigns a utility value when a transition is taken. Multiple transitions might exist with the same label *SelectPrompt_p_q*, shown in lines 5 and 9. The assignment is guarded by the user's profile and their specific cognitive skills combination. Since the latter are hidden variables, these transitions are indistinguishable from the model's analysis perspective. Hence, the model cannot assign a single, definite reward for the observed action. Instead, each potential utility is weighted by the current probability in the model's belief state that the user possesses the corresponding vector of hidden cognitive skills. The complete module contains one such rule for every possible combination of user profiles, prompt selections, and cognitive skill vectors.

Utility function. COMPASS adopts Prospect Theory principles [38, 80] to define the utility function for supporting the cognitive model. The utility of an accepted prompt is not a fixed value; it is a subjective gain evaluated relative to a specific reference point. Concretely, this reference point represents a baseline utility b_{min} (Listing 5 line 3). We initialise this value to 5 because we want to capture the fact that a suboptimal explanation for a user is better than receiving no explanation at all. We also set a value of 20 for the maximum utility that can be achieved (b_{max} , (Listing 5 line 4)). We assign these values because we want to have a range of utility values sufficient enough that will allow to distinguish between the various policies effectively. Note that, from a mathematical point of view, changing b_{min} and b_{max} will not have any effect on the computed policy. To construct the reward structure in PRISM, we employ Bayesian learning [94], leveraging the beta-binomial model to estimate user acceptance probabilities from limited data. First, we define $r_{n,p,q} \in [0, 1]$ which represents the probability that a user with profile n accepts an explanation generated using option q for prompt p . We then use the estimated probability $r_{n,p,q}$ in the computation of the utility using the prospect-theoretic formulas defined in lines 18-22 of Listing 5. The general form is: $utility = b_{min} + \kappa \cdot (b_{max} - b_{min}) \cdot r_{n,p,q}^\alpha$, where α denotes a parameter based on the concept of diminishing returns from Prospect Theory [80]. This parameter controls how sensitive the utility of a prompt is with respect to changes in its acceptance probability. We set $\alpha = 0.88$ following the guidelines from [80]. Intuitively, a value of 0.88 ensures that improvements to poorly-performing prompts are valued higher than equivalent improvements to already high-performing ones. For example, if a prompt's acceptance rate increases from 20% to 30%, the utility gain will be higher compared to a prompt that increases from 95% to 98%.

In Listing 5, we define three utility functions that capture different degrees of alignment between the chosen prompt setting and the user's latent cognitive state:

- *utility_n_p_q_match*: when the chosen prompt setting aligns with the latent state of the user
- *utility_n_p_q_okay*: when the chosen prompt setting partially aligns with the latent state of the user

```

1 rewards "acceptance"
2
3 // For each profile n,
4 // for each slot p with option q, 'Slot_p_q' and
5 [SelectPrompt_p_q] (profile=profile_n) &
6 (cogSkill_1 = L1) & ... & (cogSkill_N = L1):
7     utility_n_p_q_u1;
8
9 [SelectPrompt_p_q] (profile=profile_n) &
10 (cogSkill_1 = LM) & ... & (cogSkill_N = LM):
11     utility_n_p_q_uN;
12 endrewards

```

Listing 4: Reward structure for the expected acceptance

```

1 const double alpha0 = 1;
2 const double beta0 = 1;
3 const double b_min = 5;
4 const double b_max = 20;
5 const double kappa_okay = 0.75;
6 const double kappa_mismatch = 0.5;
7 // For each user profile n
8 const double alpha = 0.88;
9
10 // For each prompt slot p, option q
11 const int r_accepted_p_q;
12 const int r_rejected_p_q;
13
14 // For each prompt slot p, for user n
15 formula r_n_p_q = (r_accepted_p_q + alpha0) /
16 (r_accepted_p_q + r_rejected_p_q + alpha0 + beta0);
17
18 formula utility_n_p_q_match = b_min + (b_max - b_min) *
19     pow(r_n_p_q, alpha);
20
21 formula utility_n_p_q_okay = b_min + kappa_okay * (b_max -
22     b_min) * pow(r_n_p_q, alpha);
23
24 formula utility_n_p_q_mismatch = b_min + kappa_mismatch * (b_max
25     - b_min) * pow(r_n_p_q, alpha);

```

Listing 5: Utility formulas for rewards, and their constants

- *utility_n_p_q_mismatch*: when there is a mismatch between the prompt setting and the latent state of the user

The role of κ is to penalise cases where there is a misalignment between the prompt we provide to the user and the user's latent cognitive state. For example, a $\kappa = 1.0$ indicates a perfect match, and as a result, we do not apply a penalty, i.e., the original calculated utility is retained. We assign values of 0.75 and 0.5 to κ to penalise partial and total mismatches, respectively. In the former case, 75% of the potential gain is retained and 50% in the latter.

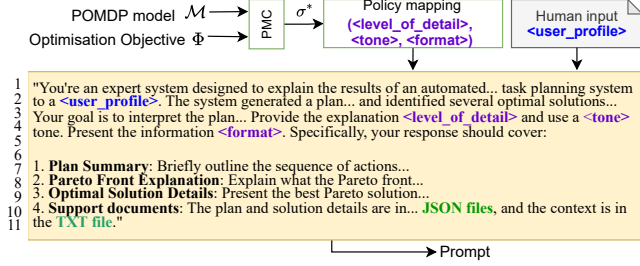
The acceptance and rejection rates are obtained from user feedback. Although using binary accept/reject feedback provides only partial information about user reactions, this choice reflects realistic deployment scenarios in which detailed or continuous feedback is often unavailable or burdensome to collect.

4.2.2 Prompt and Explanations Generation. The generation of a tailored explanation is initiated by the **Prompt Generator** (5), which formally synthesises an optimal policy σ^* from the POMDP model M using probabilistic model checking (Figure 4) [14, 29]. Given the POMDP M and a property maximising the expected acceptance of an explanation ($\phi = R\{\text{"acceptance"}\}_{max=?} [F \text{ done}]$), the optimal policy satisfying ϕ is computed. The resulting policy maps belief states to actions, providing the optimal choice for each slot in our prompt template. Internally, the **Prompt Generator** also contains a *Prompt Selection Module* that constructs the final prompt by populating the slots with the choices dictated by σ^* .

The instantiated prompt serves as a comprehensive set of instructions for the LLM, and includes directives to process and integrate data from various sources: the problem context and planner input

Table 3: Mapping from POMDP actions to prompt options and user profiles for the construction case study. IDs show the POMDP variable and real-valued instantiation.

ID	Text in prompt for <user_profile>
profile1	AI expert (assumes deep understanding of optimisation, probability...)
profile2	Domain expert (in finance or administration) (assumes focus should be on cost, and high-level risk)
profile3	Non-expert (assumes minimal technical knowledge, requiring analogies and simple terms)
ID	Text in prompt for <level_of_detail>
p1_q1	in high detail (long answer)
p1_q2	in a concise summary with only the necessary details to understand the main points (short answer)
ID	Text in prompt for <tone>
p2_q1	precise, technical, formal and precise tone
p2_q2	casual, conversational and simple tone (use examples if needed)
ID	Text in prompt for <format>
p3_q1	in a step-by-step list
p3_q2	as a single coherent paragraph, as a summary, no bullet points nor list
p3_q3	as a series of bullet points highlighting key items (avoid paragraphs)

**Figure 4: Prompt generator (5) flow. See also Table 3.**

($h1$ and $o1$, respectively), the most recent plan generated by the planner ($o2$), and the user's profile. Figure 4 shows an example of such a prompt, demonstrating how the actions selected in the POMDP correspond to parameters such as the level of detail, tone, and format of the prompt (whose values are given in Table 3).

By combining the dynamically selected slot choices with this contextual data, the prompt guides the **Explanation Generator** (6) to produce a relevant and personalised explanation ($o3$) tailored to the user's profile, inferred cognitive state, and the current task-planning scenario. Given this information, the **Explanation Generator** leverages GenAI to produce a tailored explanation aligned with the selected policy, ensuring that the generated content considers the user's cognitive state, their profile preferences, and *contextual* factors. Context information about the planning problem and results is obtained from the initial planning problem ($h1$), the planner's input file ($o1$), and the generated plan(s) ($o2$).

Example 1. Assume that the policy computed by COMPASS selects the following actions for the decision points in the prompt: $\langle p1_q1, p2_q1, p3_p3 \rangle$ (see reward actions in Listing 4). According to the mapping in Table 3, this yields an instantiated prompt with: {level of detail: high level, tone: formal, format: bullet point}.

We use POMDPs due to the several uncertainties shaping a user's likelihood of accepting an explanation. The joint POMDP structure maintains a belief over the latent user states and selects parameter values that maximise the expected acceptance probability.

4.2.3 Adaptive synthesis. COMPASS adapts its explanations in response to new information, ensuring that they remain aligned with

the user's preferences and (predicted and real) cognitive state. The adaptation mechanism is triggered by the Plan and User Monitor (3), and involves updating the POMDP model and re-synthesising the optimal policy. The primary triggers for adaptation are:

- Change in user feedback ($h3$): When a user accepts or rejects an explanation, the system updates the corresponding $R_{acceptance}$ and $R_{rejection}$ counts for that user's profile (Listing 5). This alters the POMDP reward structure (Listing 4), potentially leading to a new optimal policy that better reflects the updated user preferences.
- Change in cognitive state prediction ($i3$): As the system gathers more data on the user's interactions, the underlying cognitive skills predictor may provide updated probabilities for the user's cognitive state ($i3$). These new probabilities update the transition dynamics in the POMDP (Listing 3), refining the model's belief about the user's hidden state and prompting a re-evaluation of the optimal policy.
- Change in context ($h2$): An adaptation is also triggered if a new user with a *different profile* requests an explanation (Listing 3, line 4), if the Adaptive AI Planner generates a **new or modified plan** ($o2$) in response to changes in the operating environment, or when a brand *new planning problem* is presented ($h1$). This ensures the generated explanation is always relevant to the current user and the latest version of the task plan.

Example 2. Assume that a non-expert user requests an explanation, resulting in a policy that maps to: {level of detail: summarised, tone: casual, format: list}. Later, an AI expert uses COMPASS and requests an explanation triggering a change in context (i.e., change in profile $h2$). COMPASS modifies the POMDP model setting the parameter value *profile* (see Listing 3, line 4) to a new value representing the AI expert profile. After verifying the updated POMDP, the new policy results in prompt: {level of detail: high-detailed, tone: technical, format: paragraph style}. After some time, a new change is detected when the "AI expert profile" receives feedback as multiple rejections for the generated explanation which inform COMPASS to update the rejection rates (line 11 in Listing 5). This results in a different explanation: {level of detail: high-detailed, tone: technical, format: step-by-step list}. Finally, at the end of a long working day, the AI expert has a predicted lower level of attention; this changes the probabilities on the prediction of the user's cognitive state (lines 12 in Listing 3). A new explanation is hence generated with prompt: level of detail: low (or concise), tone: technical, format: bullet points.

4.3 Implementation

We instantiate COMPASS using the adaptive task planner from [88]. The planner example ($i1$) and generated problem file ($o1$) are JSON-formatted; instructions ($i1$) are given as a single text file (Figure 3). The planner generates a JSON file of multiple task plans ($o2$) showing Pareto-optimal trade-offs between cost and success probability (see details in [88]). For the Prompt Generator (5), we use the PRISM model checker to extract the POMDP policy. Our COMPASS tool and evaluation results are available at our GitHub website [1].

5 Evaluation

We evaluate COMPASS through two distinct multi-robot multi-human CPS case studies: a construction scenario introduced in Section 2, and an agricultural domain scenario adapted from [88].

5.1 Research Questions

RQ1 (Problem Generation Feasibility): Can the planning problem generator create correct planner inputs? We examine whether the Planning Problem Generator (Stage S1) produces both syntactically compliant (with the required JSON format) and semantically valid outputs, yielding solvable planning problems.

RQ2 (Adaptation): How effective is COMPASS in dealing with monitored changes by adapting the generated explanation? We investigate COMPASS’s end-to-end ability to dynamically re-configure explanations based on monitored changes in user interactions, shifts in cognitive state and acceptance rates.

RQ3 (Alignment): How well do the generated and adapted explanations align with the initial prompt and the user’s provided feedback, respectively? Since plan explanations are tailored to individual users’ needs, we conduct a user study with 32 participants to evaluate if COMPASS can yield explanations that adhere to explicit constraints such as tone, format, and detail.

RQ4 (Personalisation): Does the Explanation Generator of COMPASS produce explanations tailored to individual stakeholder profiles? We conduct a second user study to assess whether the intention of the generated explanation to be informative for a specific user profile truly matches the survey participants’ opinions.

5.2 Experimental Setup

RQ1 (Feasibility). We generated 20 variants per case study (randomly selecting 3-5 tasks, 1-3 robots, 1-3 humans, and 5-9 locations). For each variant, a planner input (JSON) file is created using different LLMs (Google Gemini-2.5-Pro [31], OpenAI GPT-5 [61], and DeepSeek-V3.2 [49]). We assess feasibility by checking whether the JSON is correctly formatted (parsable), whether the AI planner can successfully interpret the data and extract the necessary information (processable), and whether plans are generated (feasible).

RQ2 (Adaptation). We present a proof-of-concept adaptation scenario for the construction scenario, where all changes are present, as done in related research [28]. We generated plans from the original case study, initially for a *non-expert profile*. The initial explanation is then adapted based on changes in the user profile, explanation acceptance rates, and predictions of the user’s cognitive state.

RQ3 (Alignment). We generated explanations for all combinations of profiles (three), two levels of detail, two tone types, and three explanation formats, yielding 36 explanations (per case study) using Google Gemini-2.5-Pro. We used a Google Forms survey to collect responses to predefined questions (*Q*). First, each participant was shown an explanation obtained from a prompt with the following options: (i) non-experts: summary, casual, list formatted; (ii) AI experts: high detail, formal, paragraph formatted, and (iii) domain experts (finance): high detail, formal, bullet points formatted. These were derived from the optimal POMDP policy, initialised with probabilities and rewards informed by domain knowledge.

Since we know the prompt from which the initial explanation was generated (ground truth), we assess how participants *perceived* the explanation with the following questions: (*Q1*) *How would you describe the level of detail?*; (*Q2*) *How would you rate its tone?*; and (*Q3*) *How would you describe its format?*, with nominal-categorical answers outlined in Table 3. We generated confusion matrices based on these results. Participants were also asked to rate, on a scale from 1 to 5 (5 is better), the appropriateness of the *level of detail* (*Q4*),

tone (*Q5*), and *format* (*Q6*) of the explanation. These ratings were later compared with their evaluations on the adapted explanation.

Next, participants were invited to select their *preferred level of detail*, *tone*, and *format*. A second personalised explanation was then generated based on this feedback. As before, a set of questions was shown with ordinal numerical answers from 1 to 5 (5 is better): (*Q7*) *Is the tone better?*; (*Q8*) *Is the level of detail better?*; (*Q9*) *Is the format better?*; and (*Q10*) *Did the new explanation help you better understand the described solution?* Finally, participants were asked to provide demographic information.

RQ4 (Personalisation). A second survey classified all explanations according to the user profile (AI expert, domain expert, or non-expert) that participants believed they were intended for. All explanations from RQ3 were used in the survey (with all combinations of user profiles, level of detail, tone, and format) and shown to participants in a randomised order. Participants were then asked to select the user profile most closely matching the subject of the explanation (or the *unsure* option). The goal of this survey was to evaluate whether the intended personalisation cues embedded in the LLM-driven explanations were perceived as such by independent evaluators. This enabled assessing the effectiveness and clarity of the personalisation strategies used, as well as identifying potential mismatches between the target and perceived audiences per explanation. For each explanation, we calculated the percentage of correct matches to the intended user profiles and then computed an overall average that is reported in the results. We also analysed the correctly classified explanations for each user profile to gain further insights into which profiles were easier to identify.

5.3 Results and Discussion

RQ1 (Feasibility). All evaluated LLMs (Gemini, GPT-5, DeepSeek) successfully generated parsable and processable JSON files for all 20 variants in both construction and agricultural case studies. This outcome demonstrates that COMPASS’s Planning Problem Generator, when shown a one-shot example, can reliably produce syntactically valid and semantically-interpretable input structures compatible with the COMPASS-instantiated Adaptive AI Planner [88].

A detailed inspection of the generated JSON files confirmed that all relevant information from the original problem descriptions was accurately preserved; for instance, the number of robots and humans, their respective capabilities, and the bidirectional paths. The LLMs also correctly instantiated the task entries, i.e., task copies derived from other general tasks but localised at specific coordinates. This result shows full alignment between the generated data structures and the LLMs’ interpretive ‘understanding’ of the presented problems, with a plausible explanation being that LLMs have been extensively trained to produce structured JSON outputs [55].

From these generated JSON inputs, the AI planner successfully created feasible plans for all agricultural variants, while difference emerges in the construction case study. DeepSeek-V3.2 consistently produced feasible plans across all 20 variants, whereas GPT-5 and Gemini-2.5-Pro succeeded in only 4 out of 20 cases each. Further analysis revealed that the issue was not due to LLM limitations but to incomplete problem definitions. For example, in a problem variant with 3 tasks, 1 human, 1 robot, and 6 locations, none of the agents could perform the “construction finishing work”.

These results signify that while LLMs can often generate syntactically correct planner inputs σ_1 (JSON), the AI planner may fail to produce plans if the problem definition is incomplete. COMPASS users must thus ensure that problem descriptions are fully specified, e.g., via a human-on-the-loop check between stages S1 and S2. Using other prompting strategies beyond one-shot could potentially alleviate the issues affecting GPT-5 and Gemini.

RQ2 (Adaptation). We show the adaptation process for a concrete scenario of the construction case study, assuming a continuous adaptation loop for the changes monitored by COMPASS: change in user profile, acceptance rates and predicted cognitive state of user (Figure 5). The first explanation at time t_0 is generated when a non-expert user ($profile = 3$, Table 3) logs into the construction logistics system and requests an explanation on the currently available plans from the AI planner. COMPASS modifies the POMDP by assigning the corresponding profile value (line 4, Listing 3). The new POMDP policy obtained, $\langle p1_q2, p2_q2, p3_q1 \rangle$, yields a prompt with instructions to generate a *summarised, casual tone* and *list formatted* explanation (see Table 3).

At time t_1 , a new user (AI expert) logs in ($profile = 1$), reflected in the corresponding profile value change (line 4, Listing 3). The AI expert requests an explanation, based on which COMPASS generates a new policy $\langle p1_q1, p2_q1, p3_q2 \rangle$, which in turns produces a *high-detailed, technical* and as a *paragraph style* explanation.

At time t_2 , the same AI expert (from t_1) logs in and requests a new explanation. During this period, the system has collected feedback from other AI expert users who also requested explanations. COMPASS updates the acceptance number from 29 to 34, and rejections from 8 to 57, for explanations generated from option $p3_q2$ (lines 11-12, Listing 5). Also, COMPASS yields a new POMDP policy $\langle p1_q1, p2_q1, p3_q1 \rangle$, changing the format from paragraph style to *step-by-step* list. This adaptation shows how COMPASS employs user feedback to iteratively refine explanations; here, optimising the explanation format to better suit expert user preferences.

Next, the same user logs in after a long working day (time t_3). The cognitive state predictor used to estimate the user’s cognitive skills ($i3$ in Figure 1) estimates a higher probability that the user’s attention is lower due to the time of the day. COMPASS updates the POMDP transition probabilities (line 12, Listing 3), resulting in a new policy $\langle p1_q2, p2_q1, p3_q3 \rangle$, changing the level of detail from *high* to *low*, and the format from list to *bullet points*. Such modifications aim to minimise interaction effort, and thus mitigate potential cognitive fatigue, by reducing the volume of information shown and emphasising key points in a visually analysable structure. By simplifying the content and highlighting essential information, COMPASS helps sustain the user’s attention and comprehension even under reduced cognitive capacity, ensuring that explanations remain accessible and actionable in low-attention contexts.

This representative scenario illustrates how COMPASS could support the runtime adaptation of explanation generation to maintain user profile preferences and user’s attention engagement. Although COMPASS does not directly measure cognitive load via self-report instruments, its design explicitly targets the reduction of interaction-related cognitive effort. This design choice is grounded in Prospect Theory, which characterises human decision-making as reference-dependent and sensitive to cognitive effort, losses, and

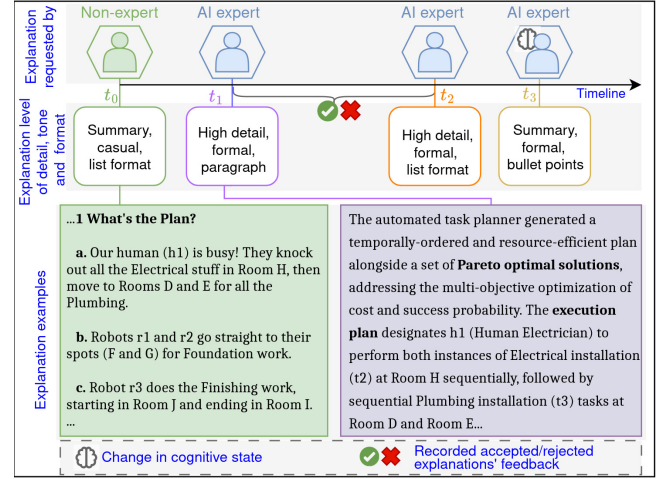


Figure 5: Scenario showing adaptive explanations

diminishing returns. By proactively adapting explanation granularity and structure to inferred attention and understanding levels, COMPASS aims to avoid explanations that users would view as cognitively costly relative to their current reference point.

RQ3 (Alignment). We analyse the survey results using descriptive statistics and quantitative data analysis as per [90].

Population and sampling. Our target population are individuals from academia and industry classified into three profiles. We analysed anonymised data from: 15 AI experts, 4 domain experts (in finance and management), and 13 non-experts in these areas.

Survey results. Table 4 presents the findings from questions Q1-Q3, comparing the real prompt attributes (tone, level of detail, and format) against those perceived by participants for the construction case study. Most participants correctly identified the tone, but discrepancies emerged in the detail levels (“summary” and “detailed”). Evidently, without a direct reference point for what constitutes a “short” or “long” explanation, participants found it difficult to evaluate the level of detail. Regarding format, the paragraph style was consistently well recognised, but list- and bullet-formatted explanations exhibited greater overlap. After reviewing the generated explanations, we found that some list outputs included numbering (e.g., “1, 2, 3” or “a, b, c”) alongside bullet symbols (·), which may have introduced perceptual noise and contributed to this confusion. These findings indicate the need to refine prompt construction to reduce ambiguity in perceived attributes. Specifically, prompt templates should more clearly distinguish levels of detail and enforce consistent formatting rules, ensuring that automatically generated explanations align more closely with users’ expectations across tone, detail, and format, and any other choices.

Tables 5a-f summarise this data from participants’ ratings on the initial and adapted explanations collected from Q4-Q9. For example, two non-experts ranked the level of detail as very appropriate (selecting 5, from 1-5), for their initial explanation, increasing to 4 non-experts for the final adapted explanation. A shift toward greater acceptance after adaptation is visualised for non-experts and AI experts, as the number of yellow and white cells shifts to the left post-adaptation; however, the opposite happens for domain experts. This is better visualised in Tables 5g-i, showing the number of

Table 4: Confusion matrices for ground truth (column) vs perceived (row—P) explanations for the construction study

Tone			Level of Details			Format			
	Cas.	Form.		Summ.	Det.		Par.	List	Bul.
Cas. (P)	11	4	Summ. (P) Det. (P)	12	12	Par. (P)	8	2	1
Form. (P)	2	15		1	7	List (P)	4	4	1
						Bull. (P)	3	7	2

Cas: Casual; Form: Formal; Summ: Summary; Det: Detailed; Par: Paragraph; Bul: Bullets

Table 5: Construction case study acceptance score

	Initial explanation acceptance score 1-5 (5 is higher)					Adapted explanation acceptance score 1-5 (5 is higher)					Change in score per participant			Total
	1	2	3	4	5	1	2	3	4	5	Decrease	None	Increase	
Non-experts	Detail	0	2	2	7	2	0	0	6	3	4	6	1	13
	Tone	0	3	2	3	5	0	0	4	4	5	4	5	13
	Format	0	1	5	4	3	0	1	4	3	5	3	4	13
AI experts	Detail	0	2	2	9	2	0	2	2	4	7	4	5	15
	Tone	0	3	1	7	4	0	0	3	6	6	5	4	15
	Format	1	5	1	4	4	0	1	0	4	10	1	5	15
Domain experts	Detail	0	0	0	3	1	0	1	0	1	2	2	0	4
	Tone	0	0	0	0	4	1	1	0	1	1	3	1	4
	Format	0	0	0	3	1	0	1	0	2	1	2	1	4

individual participants that increased, decreased or did not change their score post-adaptation. For non-experts and AI experts, the total number of increased scores (16 and 21, respectively) is higher compared to no-change (10 and 14) or decreased scores (13 and 10). This shows that the adaptive explanation was perceived as better, although no consensus was reached among participants. However, this trend is reversed for non-experts, where a decrease in score was registered 7 times, compared to 3 increases and 2 equal scores. Further studies with a larger participant pool with this profile are needed to confirm these trends. Similar trends are observed for the agricultural case study (Tables 6 and 7).

Overall, these exploratory results yield indicative evidence that COMPASS can influence user satisfaction and perceived appropriateness among non-experts and AI experts. The observed trends across user groups and case studies suggest that tailoring explanations to users' expertise could improve acceptance in AI planners, though these effects should be interpreted cautiously.

These findings should also be interpreted probabilistically, consistent with the probabilistic nature of our underlying POMDP model. For all profiles, COMPASS can incorporate these acceptance and rejection scores to modify the POMDP rewards accordingly. Hence, after this adaptation occurs, both non-experts and AI experts are *more likely* to accept explanations similar to the adapted one whereas domain experts are not.

For Q10, we obtained scores on how the new explanation helped participants better understand the planner's solution. Figure 6 shows a consistent trend towards increasing understanding level for the adapted explanation compared to the initial. Both construction and agriculture tasks showed similar distributions; most participants reported moderate to high increase (scores 3–5). For domain experts, even when previous results show a decrease in the acceptance score (especially in the explanation format), almost all participants in this group provided a 4/5 score. Overall, these results suggest that adaptive explanations could support users' understanding from different expertise levels and domains.

Table 6: Confusion matrices for ground truth (column) vs perceived (row—P) explanations for the agriculture study

Tone			Level of Details			Format			
	Cas.	Form.		Summ.	Det.		Par.	List	Bul.
Cas. (P)	8	5	Summ. (P)	9	6	Par. (P)	8	3	0
Form. (P)	5	14	Det. (P)	4	13	List (P)	5	3	2
						Bull. (P)	2	7	2

Cas: Casual; Form: Formal; Summ: Summary; Det: Detailed; Par: Paragraph; Bul: Bullets

Table 7: Agricultural case study acceptance scores

	Initial explanation acceptance score 1-5 (5 is higher)					Adapted explanation acceptance score 1-5 (5 is higher)					Change in score per participant			Total
	1	2	3	4	5	1	2	3	4	5	Decrease	None	Increase	
Non-experts	Detail	1	2	0	7	3	0	0	5	3	5	3	4	13
	Tone	1	2	1	5	4	0	0	4	5	4	3	6	13
	Format	0	4	1	4	4	0	0	3	5	5	1	5	13
AI experts	Detail	0	3	5	5	2	0	5	2	4	4	6	3	15
	Tone	0	7	2	4	2	1	2	5	4	3	3	5	15
	Format	0	11	0	2	2	0	2	3	3	7	2	2	15
Domain experts	Detail	0	1	1	2	0	0	0	1	2	1	1	1	4
	Tone	0	0	1	2	1	0	0	1	2	1	1	1	4
	Format	0	0	0	0	4	0	0	1	2	1	3	1	4

RQ4 (Personalisation). Similar to RQ3, we used descriptive statistics and quantitative data analysis to analyse these survey results. **Population and sampling.** We analysed anonymised data from five participants with a computer science background working in academia (excluding participants from RQ3). Each participant classified all 72 explanations gathered from both case studies.

Survey results. For the construction case study, 49.8% (35.6% std) responses correctly matched the user profile for which the explanation was generated. The value increased for the agricultural case study (62.4% with 40.1% std). As there are three possible profile options from which participants could select (AI expert, domain expert, non-expert), a random selection would result in 33.33% correct matches. Thus, on average, participants correctly identified the intended user profile at a rate substantially above chance. These results suggest that some profile personalisation cues embedded in the explanations were generally perceivable by participants, though variability was high, as indicated by the standard deviations.

To provide further insights into the misclassification per profile, we computed the proportion of explanations correctly classified for each user profile, and the proportion of explanations for which participants selected the *unsure* option. In the construction study, domain expert explanations were the easiest to identify (55.0% correct), followed by non-expert explanations (36.7%) and AI experts (44.0%). The unsure option (11.1%) signified a light ambiguity level.

In contrast, the agricultural case study showed higher correct classification rates for AI experts (90% accuracy), domain experts (56.7%) and the unsure option (11.7%); with the lowest percentage also reported for non-experts (23.3%). Overall, these results demonstrate that participants found it easier to recognise AI expert explanations in both settings, while non-expert explanations were consistently more difficult to classify across both domains.

A plausible explanation for this insight is that the generated plans from the AI planner are defined in technical terms such as "Pareto optimality", "probability of mission success", "task-completion expected cost", and "task identifiers", which might be perceived as

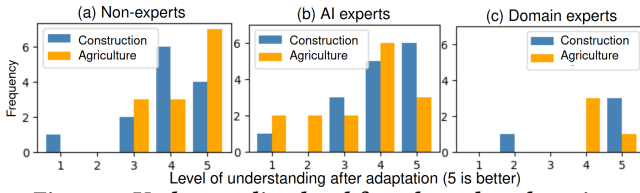


Figure 6: Understanding level for adapted explanation

tailored for AI experts. Another explanation is that we did not have participants with finance domain expertise to assess explanations intended for this profile. Thus, the accuracy of this profile detection may be underestimated. The survey results show that personalisation of AI explanations is not always detectable by independent evaluators, particularly when the content remains highly technical.

6 Threats to Validity

Internal validity threats concern causal inferences within our study. The effectiveness of COMPASS in refining prompts may vary with different LLMs (e.g., with stronger reasoning or instruction-following abilities) and different prompting strategies. An advanced LLM might produce strong explanations even from weak prompts, while a light LLM could fail even with a perfect prompt, thus confounding COMPASS’s measured impact. We mitigated this threat by generating AI planner input files using three state-of-the-art LLMs to compare performance across multiple planning problems in two case studies. The prompts in Figures 3 and 4 were iteratively refined. We also acknowledge that LLMs are prone to hallucinations and output variability, even under identical prompts. Further, the binary accept/reject feedback in the POMDP provides only a coarse signal of user response, reflecting realistic operational settings where fine-grained or continuous feedback is rarely available or costly to obtain. Finally, we evaluate the feasibility of the PDDL file generation step as an initial validation of correctness. Verifying the extraction process, specifically, if information from the human input $h1$ is preserved, would strengthen our results and is thus part of our future work as in [79, 87].

To mitigate **external validity** threats affecting the generalisation of COMPASS, our evaluation employed two CPS domains. Further, we validated both case studies through collaboration with industrial partners, ensuring practical relevance and applicability.

Conclusion validity threats might arise given the relatively small sample size, reducing statistical inference. While trends in acceptance and perceived level of detail, tone and format were observed, results should be interpreted as indicative rather than conclusive. Replication with larger participant pools, especially more domain experts, would strengthen our findings’ robustness.

7 Related Work

LLMs for explainability and planning. LLMs have been used to generate explanations for black-box models [6, 42] and to provide interactive explanations through conversational systems [73, 75]. In medical contexts, LLMs support diagnostic reasoning [3] and generate interpretable explanations to help the overall clinical decision-making [81]. In finance, LLMs generate explanations that improve the transparency of credit scoring models [24] and risk assessment [52]. Approaches like Tree-of-Thoughts [91], “inner monologue” [35] and LLM+P [50] use LLMs as planning engines. Our

work is complementary: rather than using an LLM to compute plans, COMPASS models human behaviour to adapt explanations of plans produced by formal planning methods.

LLMs for explainable AI (XAI). Within XAI, LLMs are powerful tools that can translate complex and often cryptic AI model behaviours into easy-to-understand explanations [8, 93]. LLMs have also been used to convert numeric outputs from traditional XAI methods, such as SHAP [51] and LIME [69], into natural language explanations [10, 95]. For further details, see the survey in [8].

Explainable AI Planning (XAIP). XAIP is concerned with generating explanations for automated planning [19]. Foundational work on model reconciliation formalises explanation as resolving mismatches between an agent’s planning model and a human’s mental model [18], enabling planners to answer contrastive queries [17, 26, 41]. Expectation-aware planning further integrates human expectations into plan synthesis, often trading optimality for interpretability [77]. In contrast, COMPASS adaptively tailors LLM-generated plan explanations by modelling users’ cognitive states and feedback, focusing on presentation-level adaptation across user profiles rather than modifying the planning process itself.

Explainability for self-adaptive systems (SAS). Explainability in SAS poses unique challenges due to their architecture, combining the managed and managing systems linked via feedback loops [78]. Existing work [5, 63] uses runtime models to provide explanations for the adaptation decisions. Closer to COMPASS, [2, 46] employ probabilistic modelling to personalise how explanations are delivered to the user. Unlike COMPASS, which synthesises explanations dynamically considering hidden cognitive states, they assume that personality traits are observable and explanations are static.

Modelling human behaviour via POMDPs. POMDPs have been used to model decision-making under uncertainty by constructing models of human behaviour [11]. Illustrative examples include: PsychSim [65] for modelling complex social interactions and cognitive states, an assistance robot in nursing homes to infer the users’ hidden goals [64], and a POMDP for hand-washing aid for individuals with dementia [34]. Contextual bandits [43] offer an alternative by modelling uncertainty in action outcomes conditioned on observable context [45]. However, they assume that context fully captures the user state and model single-step decisions [43]. In contrast, COMPASS uses POMDPs to model latent cognitive states and encode how actions influence future hidden states and rewards.

8 Conclusions and Future Work

COMPASS is a proof-of-concept self-adaptive approach for generating task-plan explanations via LLMs. By treating prompt engineering as a cognitive and iterative process, COMPASS integrates user feedback, cognitive-state prediction, and context to tailor explanations to diverse users. We evaluated COMPASS in two industrial CPS, showing that LLMs can translate natural language problems into valid task-planning inputs and that adaptive explanations improve user understanding. Future work involves refining prompt instructions using zero- and few-shot prompting, supporting additional properties [56, 86], and incorporating additional planners [30, 82].

Acknowledgements. This research was supported by the EU Horizon projects AI4Work/SOPRANO (101135990/101120990), the ARIA-funded project ULTIMATE, and the Centre for Assuring Autonomy.

References

- [1] Project's GitHub. <https://github.com/Gricel-lee/COMPASS-LLM>.
- [2] Mohammed N. Alharbi, Shihong Huang, and David Garlan. 2021. A Probabilistic Model for Effective Explainability Based on Personality Traits. In *Software Architecture: 15th European Conference on Software Architecture*. 21 pages.
- [3] Kingsley Attai, Moses Ekpenyong, Constance Amannah, Daniel Asuquo, Peterben Ajuga, Okure Obot, Ekemini Johnson, Anietie John, Omosivie Maduka, Christie Akwaowo, and Faith-Michael Uzoka. 2024. Enhancing the Interpretability of Malaria and Typhoid Diagnosis with Explainable AI and Large Language Models. *Tropical Medicine and Infectious Disease* 9, 9 (2024).
- [4] Paul Ayres, Joy Yeonjoo Lee, Fred Paas, and Jeroen JG Van Merriënboer. 2021. The validity of physiological measures to identify differences in intrinsic cognitive load. *Frontiers in psychology* 12 (2021), 702538.
- [5] Nelly Bencomo, Kris Welsh, Pete Sawyer, and Jon Whittle. 2012. Self-Explanation in Adaptive Systems. In *2012 IEEE 17th International Conference on Engineering of Complex Computer Systems*. 157–166.
- [6] Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. Towards LLM-guided Causal Explainability for Black-box Text Classifiers. *arXiv:2309.13340 [cs.CL]* <https://arxiv.org/abs/2309.13340>
- [7] Andrea Bianco and Luca De Alfaro. 1995. Model checking of probabilistic and nondeterministic systems. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*. Springer, 499–513.
- [8] Ahsan Bilal, David Ebert, and Beiyu Lin. 2025. LLMs for explainable AI: A comprehensive survey. *arXiv preprint arXiv:2504.00125* (2025).
- [9] André Borrmann, Markus König, Christian Koch, and Jakob Beetz. 2018. Building information modeling: why? What? How? In *Building information modeling: Technology foundations and industry practice*. Springer, 1–24.
- [10] James Burton, Noura Al Moubayed, and Amir Enshaei. 2023. Natural Language Explanations for Machine Learning Classification Decisions. In *2023 International Joint Conference on Neural Networks (IJCNN)*. 1–9.
- [11] Radu Calinescu, Simos Gerasimou, Sinem Getir Yaman, Gricel Vazquez Flores, and Micah Bassett. 2026. Verification of Multi-Model Stochastic Systems. In *48th IEEE/ACM International Conference on Software Engineering (ICSE)*.
- [12] Radu Calinescu, Calum Imrie, Ravi Mangal, Genaina Nunes Rodrigues, Corina Păsăreanu, Misael Alpizar Santana, and Gricel Vázquez. 2022. Discrete-event controller synthesis for autonomous systems with deep-learning perception components. *arXiv preprint arXiv:2202.03360* (2022).
- [13] Radu Calinescu, Calum Imrie, Ravi Mangal, Genaina Nunes Rodrigues, Corina Păsăreanu, Misael Alpizar Santana, and Gricel Vázquez. 2024. Controller synthesis for autonomous systems with deep-learning perception components. *IEEE Transactions on Software Engineering* 50, 6 (2024), 1374–1395.
- [14] Radu Constantin Calinescu, Milan Ceska, Simos Gerasimou, Marta Kwiatkowska, and Nicola Paoletti. 2018. Efficient synthesis of robust models for stochastic systems. *Journal of Systems and Software* 143 (2018), 140–158.
- [15] Javier Cámara, Gabriel A. Moreno, and David Garlan. 2014. Stochastic game analysis and latency awareness for proactive self-adaptation. In *Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. 155–164.
- [16] Marco Cascella, Jonathan Montomoli, Valentina Bellini, and Elena Bignami. 2023. Evaluating the feasibility of ChatGPT in healthcare: an analysis of multiple clinical and research scenarios. *Journal of medical systems* 47, 1 (2023), 33.
- [17] Michael Cashmore, Anna Collins, Benjamin Krarup, Senka Krivic, Daniele Magazzeni, and David Smith. 2019. Towards explainable AI planning as a service. In *2nd ICAPS Workshop on Explainable Planning*. *2nd ICAPS Workshop on Explainable Planning*.
- [18] Tathagata Chakraborti, Sarath Sreedharan, Sachin Grover, and Subbarao Kambhampati. 2019. Plan explanations as model reconciliation—an empirical study. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 258–266.
- [19] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. 2021. The emerging landscape of explainable automated planning & decision making. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (Yokohama, Yokohama, Japan) (IJCAI)*. Article 669, 9 pages.
- [20] Yanda Chen, Ruiqi Zhong, Narutatsu Ri, Chen Zhao, He He, Jacob Steinhardt, Zhou Yu, and Kathleen McKeown. 2024. Do models explain themselves? counterfactual simulatability of natural language explanations. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. JMLR.org.
- [21] Mirko D'Angelo, Simos Gerasimou, Sona Ghahremani, Johannes Grohmann, Ingrid Nunes, Evangelos Pournaras, and Sven Tomforde. 2019. On learning in collective self-adaptive systems: State of practice and a 3D framework. In *2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE, 13–24.
- [22] Neil T Dantam, Zachary K Kingston, Swarat Chaudhuri, and Lydia E Kavrak. 2016. Incremental task and motion planning: A constraint-based approach.. In *Robotics: Science and systems*, Vol. 12. Ann Arbor, MI, USA, 00052.
- [23] Frédéric Dehais, Alex Lafont, Raphaëlle Roy, and Stephen Fairclough. 2020. A neuroergonomics approach to mental workload, engagement and human performance. *Frontiers in neuroscience* 14 (2020), 268.
- [24] Duanyu Feng, Yongfu Dai, Jimin Huang, Yifang Zhang, Qianqian Xie, Weiguang Han, Zhengyu Chen, Alejandro Lopez-Lira, and Hao Wang. 2024. Empowering Many, Biasing a Few: Generalist Credit Scoring through Large Language Models. *arXiv:2310.00566* <https://arxiv.org/abs/2310.00566>
- [25] Maria Fox and Derek Long. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research* 20 (2003).
- [26] Maria Fox, Derek Long, and Daniele Magazzeni. 2017. Explainable planning. *arXiv preprint arXiv:1709.10256* (2017).
- [27] Lina Gao, Zhongying Zhao, Chao Li, Jianli Zhao, and Qingtian Zeng. 2022. Deep cognitive diagnosis model for predicting students' performance. *Future Generation Computer Systems* 126 (2022), 252–262.
- [28] Simos Gerasimou, Radu Calinescu, and Alec Banks. 2014. Efficient runtime quantitative verification using caching, lookahead, and nearly-optimal reconfiguration. In *Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. 115–124.
- [29] Simos Gerasimou, Radu Calinescu, and Giordano Tamburrelli. 2018. Synthesis of probabilistic models for quality-of-service software engineering. *Automated Software Engineering* 25, 4 (2018), 785–831.
- [30] Simos Gerasimou, Javier Cámara, Radu Calinescu, Naif Alasmari, Faisal Al-hwikem, and Xinwei Fang. 2021. Evolutionary-guided synthesis of verified pareto-optimal MDP policies. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 842–853.
- [31] Google. 2025. Gemini. <https://gemini.google.com> Accessed: 2025-10-09.
- [32] Hans Hansson and Bengt Jonsson. 1994. A logic for reasoning about time and reliability. *Form. Asp. Comput.* 6, 5 (Sept. 1994), 512–535.
- [33] Malte Helmert. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research* 26 (2006), 191–246.
- [34] Jesse Hoey, Craig Boutilier, Pascal Poupart, Patrick Olivier, Andrew Monk, and Alex Mihailidis. 2013. People, sensors, decisions: Customizable and adaptive technologies for assistance in healthcare. *ACM Trans. Interact. Intell. Syst.* 2, 4, Article 20 (Jan. 2013), 36 pages.
- [35] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and brian ichter. 2022. Inner Monologue: Embodied Reasoning through Planning with Language Models. In *6th Annual Conference on Robot Learning*.
- [36] Alon Jacovi and Yoav Goldberg. 2020. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness?. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 4198–4205.
- [37] Daniel Kahneman, Jack L. Knetsch, and Richard H. Thaler. 1991. Anomalies: The endowment effect, loss aversion, and status quo bias. *Journal of Economic perspectives* 5, 1 (1991), 193–206.
- [38] Daniel Kahneman and Amos Tversky. 2013. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*. World Scientific, 99–127.
- [39] Frank C. Keil. 2006. Explanation and understanding. *Annu. Rev. Psychol.* 57, 1 (2006), 227–254.
- [40] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2024. The challenge of understanding what users want: Inconsistent preferences and engagement optimization. *Management science* 70, 9 (2024), 6336–6355.
- [41] Benjamin Krarup, Michael Cashmore, Daniele Magazzeni, and Tim Miller. 2019. Model-based contrastive explanations for explainable planning. In *ICAPS 2019 Workshop on Explainable AI Planning (XAIPI)*. AAAI Press, USA.
- [42] Nicholas Kroeger, Dan Ley, Satyapriya Krishna, Chirag Agarwal, and Himabindu Lakkaraju. 2023. Are Large Language Models Post Hoc Explainers?. In *XAI in Action: Past, Present, and Future Applications*.
- [43] John Langford and Tong Zhang. 2007. The Epoch-Greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 21st International Conference on Neural Information Processing Systems (Vancouver, British Columbia, Canada) (NIPS)*. Curran Associates Inc., Red Hook, NY, USA, 817–824.
- [44] Jialong Li, Mingyue Zhang, Nianyu Li, Danny Weyns, Zhi Jin, and Kenji Tei. 2024. Exploring the potential of large language models in self-adaptive systems. In *Proceedings of the 19th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. 77–83.
- [45] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.
- [46] Nianyu Li, Sridhar Adepu, Eunsuk Kang, and David Garlan. 2020. Explanations for human-on-the-loop: A probabilistic model checking approach. In *Proceedings of the IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. 181–187.
- [47] Xuefeng Li, Liwen Wang, Guanting Dong, Keqing He, Jinzheng Zhao, Hao Lei, Jiachi Liu, and Weiran Xu. 2023. Generative Zero-Shot Prompt Learning for Cross-Domain Slot Filling with Inverse Prompting. In *Findings of the Association for*

- Computational Linguistics: ACL 2023*. Association for Computational Linguistics, Toronto, Canada, 825–834.
- [48] Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023. Large language models in finance: A survey. In *Proceedings of the Fourth ACM International Conference on AI in Finance*. 374–382.
 - [49] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Cheng-gang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).
 - [50] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023. LLM+P: Empowering Large Language Models with Optimal Planning Proficiency. *arXiv:2304.11477 [cs.AI]* <https://arxiv.org/abs/2304.11477>
 - [51] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*. 4768–4777.
 - [52] Ayuns Luz. EasyChair. 2024. Enhancing the Interpretability and Explainability of AI-Driven Risk Models Using LLM Capabilities. EasyChair Preprint 13368.
 - [53] Ulrik Lyngs, Reuben Binns, Max Van Kleek, and Nigel Shadbolt. 2018. So, tell me what users want, what they really, really want!. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–10.
 - [54] Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2024. Towards faithful model explanation in NLP: A survey. *Computational Linguistics* 50, 2 (2024).
 - [55] Yuetian Mao, Junjie He, and Chunyang Chen. 2025. From prompts to templates: A systematic prompt template analysis for real-world LLMs. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*.
 - [56] Claudio Menghi, Christos Tsigkanos, Mehrnoosh Askarpour, Patrizio Pelliccione, Grisel Vázquez, Radu Calinescu, and Sergio Garcia. 2022. Mission specification patterns for mobile robots: Providing support for quantitative properties. *IEEE Transactions on Software Engineering* 49, 4 (2022), 2741–2760.
 - [57] Andrea Micheli and Alessandro Valentini. 2021. Synthesis of search heuristics for temporal planning via reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 11895–11902.
 - [58] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267 (2019), 1–38.
 - [59] Gethin Norman, David Parker, and Xueyi Zou. 2017. Verification and control of partially observable probabilistic systems. *Real-Time Systems* 53, 3 (2017).
 - [60] OpenAI. 2024. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2024). <https://arxiv.org/abs/2303.08774>
 - [61] OpenAI. 2025. ChatGPT-5: Large Language Model by OpenAI. <https://chatgpt.com/>. Accessed: 2025-10-24.
 - [62] Ashutosh Pandey, Gabriel A Moreno, Javier Cámara, and David Garlan. 2016. Hybrid planning for decision making in self-adaptive systems. In *2016 IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*.
 - [63] Juan Marcelo Parra-Ullauri, Antonio García-Domínguez, and Nelly Bencomo. 2021. From a Series of (Un)fortunate Events to Global Explainability of Runtime Model-Based Self-Adaptive Systems. In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion*. 807–816.
 - [64] Joelle Pineau, Michael Montemerlo, Martha Pollack, Nicholas Roy, and Sebastian Thrun. 2003. Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems* 42, 3 (2003), 271–281. doi:10.1016/S0921-8890(02)00381-0
 - [65] David V. Pynadath and Stacy C. Marsella. 2005. PsychSim: modeling theory of mind with decision-theoretic agents. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (Edinburgh, Scotland) (IJCAI)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1181–1186.
 - [66] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).
 - [67] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Amodei, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
 - [68] Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended abstracts of the 2021 CHI conference on human factors in computing systems*. 1–7.
 - [69] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. Association for Computing Machinery, 1135–1144.
 - [70] Rovetta, Alberto and Trapasso, Alessandro and Valentini, Alessandro and et al. 2024. Unified Planning Documentation. <https://unified-planning.readthedocs.io/en/latest/index.html> Accessed: 2024-12-02.
 - [71] Ranjan Sapkota, Rizwan Qureshi, Muhammad Usman Hadi, Syed Zohaib Hassan, Ferhat Sadak, Maged Shoman, Muhammad Sajjad, Fayaz Ali Dharejo, Achyut Paudel, Jijia Li, et al. 2025. Multi-modal LLMs in agriculture: A comprehensive review. *IEEE Transactions on Automation Science and Engineering* (2025).
 - [72] Enrico Scala, Patrik Haslum, Sylvie Thiébaux, and Miquel Ramirez. 2016. Interval-based relaxation for general numeric planning. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*. 655–663.
 - [73] Hua Shen, Chieh-Yang Huang, Tongshuang Wu, and Ting-Hao Kenneth Huang. 2023. ConvXAI: Delivering Heterogeneous AI Explanations via Conversations to Support Human-AI Scientific Writing. In *Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing (CSCW '23 Companion)*. Association for Computing Machinery, 384–387.
 - [74] Chanda Simfukwe, Young Chul Youn, et al. 2023. CNN for a regression machine learning algorithm for predicting cognitive impairment using qEEG. *Neuropsychiatric Disease and Treatment* (2023), 851–863.
 - [75] Dylan Z. Slack, Satyapriya Krishna, Himabindu Lakkaraju, and Sameer Singh. 2022. TalkToModel: Explaining Machine Learning Models with Interactive Natural Language Conversations. *NeurIPS Workshop on Trustworthy and Socially Responsible Machine Learning (TSRML)* (2022).
 - [76] Beatriz Soviero, Daniel Kuhn, Alexandre Salle, and Viviane Pereira Moreira. 2024. Chatgpt goes shopping: LLMs can predict relevance in e-commerce search. In *European Conference on Information Retrieval*. Springer, 3–11.
 - [77] Sarath Sreedharan, Tathagata Chakraborti, Christian Muise, and Subbarao Kambhampati. 2020. Expectation-aware planning: A unifying framework for synthesizing and executing self-explaining plans for human-aware planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 2518–2526.
 - [78] Raphael Straub, Florian Sihler, Ali Torbati, Cong Wang, Raffaella Groner, Verena Klös, and Matthias Tichy. 2025. Explainability in Self-Adaptive Systems: A Systematic Literature Review. In *Software Engineering and Advanced Applications: 51st Euromicro Conference (SEAA)*. Springer-Verlag, 280–297.
 - [79] Marcus Tantakoun, Christian Muise, and Xiaodan Zhu. 2025. LLMs as planning modelers: A survey for leveraging large language models to construct automated planning models. In *AAAI 2025 Workshop LLM4Plan*.
 - [80] Amos Tversky and Daniel Kahneman. 1992. Advances in Prospect Theory: Cumulative Representation of Uncertainty. *Journal of Risk and Uncertainty* 5, 4 (1992), 297–323.
 - [81] D. Umerenkov, G. Zubkova, and A. Nesterov. 2023. Deciphering Diagnoses: How Large Language Models Explanations Influence Clinical Decision Making. *arXiv:2310.01708 [cs.CL]* <https://arxiv.org/abs/2310.01708>
 - [82] Alessandro Valentini, Andrea Micheli, and Alessandro Cimatti. 2020. Temporal planning with intermediate conditions and effects. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9975–9982.
 - [83] Bram B Van Acker, Davy D Parmentier, Peter Vlerick, and Jelle Saldien. 2018. Understanding mental workload: from a clarifying concept analysis toward an implementable framework. *Cognition, technology & work* 20, 3 (2018), 351–365.
 - [84] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
 - [85] Shubham Vatsal and Harsh Dubey. 2024. A survey of prompt engineering methods in large language models for different NLP tasks. *arXiv preprint arXiv:2407.12994* (2024).
 - [86] Grisel Vázquez, Anastasia Mavridou, Marie Farrell, Tom Pressburger, and Radu Calinescu. 2024. Robotics: A new mission for fret requirements. In *NASA Formal Methods Symposium*. Springer, 359–376.
 - [87] Kaustubh Vyas, Damien Graux, Sébastien Montella, Pavlos Vougiouklis, Ruofei Lai, Keshuang Li, Yang Ren, and Jeff Z Pan. 2025. An Extensive Evaluation of PDDL Capabilities in off-the-shelf LLMs. *arXiv preprint arXiv:2502.20175* (2025).
 - [88] Grisel Vázquez, Alexandros Evangelidis, Sepehdeh Shahbeigi, and Simos Gerasimou. 2025. Adaptive Human-Robot Collaborative Missions using Hybrid Task Planning. In *2025 IEEE/ACM 20th Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. 73–84.
 - [89] Danny Weyns, Radu Calinescu, Raffaella Mirandola, Kenji Tei, Maribel Acosta, Nelly Bencomo, et al. 2023. Towards a research agenda for understanding and managing uncertainty in self-adaptive systems. *ACM SIGSOFT Software Engineering Notes* 48, 4 (2023), 20–36.
 - [90] Danny Weyns, Ilias Gerostathopoulos, Nadeem Abbas, Jesper Andersson, Stefan Biffl, Premek Brada, Tomas Bures, Amleto Di Salle, Matthias Galster, Patricia Lago, et al. 2023. Self-adaptation in industry: A survey. *ACM Transactions on Autonomous and Adaptive Systems* 18, 2 (2023), 1–44.
 - [91] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: deliberate problem solving with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS)*. Article 517, 14 pages.
 - [92] Håkan LS Younes and Michael L Littman. 2004. PPDDL1. 0: An extension to PDDL for expressing planning domains with probabilistic effects. *Techn. Rep. CMU-CS-04-162* 2 (2004), 99.
 - [93] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for Large Language Models: A Survey. *ACM Trans. Syst. Technol.* 15, 2, Article 20 (Feb. 2024), 38 pages. doi:10.1145/3639372
 - [94] Xingyu Zhao, Simos Gerasimou, Radu Calinescu, Calum Imrie, Valentin Robu, and David Flynn. 2024. Bayesian learning for the robust verification of autonomous robots. *Communications Engineering* 3, 1 (2024), 18.
 - [95] Alexandra Zytek, Sara Pido, Sarah Alnegheimish, Laure Berti-Équille, and Kalyan Veeramachaneni. 2024. Explingo: Explaining AI Predictions using Large Language Models. In *2024 IEEE International Conference on Big Data (BigData)*. 1197–1208.