Evaluating Human Trust in LLM-Based Planners: A Preliminary Study

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large Language Models (LLMs) offer planning capabilities such as natural language explanations and iterative refinement, but human trust in these systems remains underexplored. We present a user study comparing trust in LLM-based and classical planners within a Planning Domain Definition Language (PDDL) domain. Using both subjective (trust ratings) and objective (evaluation accuracy) measures, we find that correctness is the primary driver of trust and performance. Explanations improve evaluation accuracy but have limited impact on trust, while plan refinement showed potential for increasing trust without significantly enhancing evaluation accuracy.

1 Introduction

Planning is the process of determining a sequence of actions to transit from an initial state to a desired goal state. Planners—systems designed to generate such action sequences under given constraints—play a critical role in automating decision-making processes in domains such as robotic navigation, logistics optimization, and medical scheduling.

Traditional planners, while effective in structured and predictable environments, often struggle with rigidity and a lack of explainability. In contrast, Large Language Models (LLMs) have recently demonstrated strong performance in various domains, including text generation Li et al. [2024], question answering Puri et al. [2020], Ram et al. [2021], and code completion Liu et al. [2020]. Unlike traditional planners, LLMs support multi-plan generation (i.e., return multiple plans to enable users to choose), dynamic adjustments based on externally given information, and understandable communication with humans via natural language. These strengths have sparked growing interest in using LLMs as planners across diverse domains, including robotics Ren et al. [2023], Singh et al. [2023], Yang et al. [2024a], Huang et al. [2022], healthcare Cascella et al. [2023], Sallam [2023], and law Wu et al. [2023], Cheong et al. [2024].

However, the increasing use of LLM-based planners raises concerns, particularly regarding trust. Trust, defined as the willingness to rely on automated systems Lee and See [2004], is vital for the adoption of planning systems. Without trust, even systems with superior technical capabilities may struggle to gain acceptance in practical settings Vorm and Combs [2022]. Planning tasks are uniquely challenging due to their reliance on high correctness, sequential reasoning, and adaptability to dynamic environments Allmendinger [2017]. These factors amplify the importance of trust, as both over-trust and under-trust can introduce errors or inefficiencies in planning and can have cascading effects on task success Talvitie [2012], Laurian [2009]. Thus, fostering appropriate trust levels in LLM-based planners is essential for maximizing their potential while minimizing risks.

While prior research has explored factors influencing trust in LLM-based systems, such as anthropomorphic cues Cohn et al. [2024], the framing and presence of explanations Sharma et al. [2024],

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

and user interface design Sun et al. [2024], factors influencing human trust in LLMs in the context of planning tasks remain underexplored. As the Planning Domain Definition Language (PDDL) has become a common benchmark for evaluating the planning capabilities of LLMs Silver et al. [2022, 2024], existing work primarily focuses on technical performance metrics, such as plan correctness and efficiency. To the best of our knowledge, no prior studies have empirically investigated human trust in LLM-based planners compared to classical PDDL solvers in a PDDL domain. *This work bridges this gap by conducting an exploratory user study that evaluates trust in a PDDL domain.*

Specifically, LLMs possess unique capabilities and limitations compared to classical PDDL planners McDermott [2000], Ghallab et al. [2004] that may affect trust levels. For instance, LLMs can generate natural language explanations to clarify why specific decisions were made Wang et al. [2023], Karia et al. [2024] and iteratively refine their outputs based on user feedback Stiennon et al. [2020], Christiano et al. [2017], Ouyang et al. [2022], Yang et al. [2024b]. These capabilities have been shown in other contexts to enhance user trust by making the planning process more transparent and interactive Kunkel et al. [2019], Sebo et al. [2019]. However, LLMs also exhibit significant limitations, such as their inability to reliably generate or validate plans independently, even for relatively simple tasks Kambhampati et al. [2024], Valmeekam et al. [2022], Silver et al. [2022], Valmeekam et al. [2023]. These capabilities and limitations highlight the need for a deeper understanding of the interplay among correctness, explanation, and refinement.

Trust can be evaluated using Likert-scale user questionnaires Martelaro et al. [2016], Xu and Dudek [2015], Choi and Ji [2015] and broader instruments like the Propensity to Trust scale Merritt et al. [2013], which assesses general attitudes toward machines. This study combines subjective 7-point Likert scale trust scores with objective user evaluation accuracy of generated plans.

Key findings: Our results show that correctness is the primary driver of both evaluation accuracy and trust, with the PDDL solver outperforming all LLM-based planners. While explanations improved participants' ability to assess plan correctness, they had little effect on trust. In contrast, plan refinement increased trust despite no gain in evaluation accuracy—indicating that users may perceive refinement as a signal of competence. This suggests that LLMs can earn user trust without actual improvements in performance, since refined plans are generated by the same underlying model. As many LLMs are fine-tuned using subjective human feedback Stiennon et al. [2020], Christiano et al. [2017], this highlights the risk of overtrust—where models appear more trustworthy than they are. Our findings offer practical insights for designing human-centered AI planning systems.

2 Methods

We evaluate factors influencing user trust in planners by comparing a language-model-based planner, denoted as an *LLM Planner* (GPT-4o Achiam et al. [2023]), with a traditional graph-search-based planner, denoted as a *PDDL Solver* (Fast Downwards Helmert [2006]). Unlike the PDDL Solver, which relies on graph search algorithms, the LLM Planner can reason through the planning problem, explain its proposed solution, and iteratively refine the solution based on external feedback.

2.1 Planning Problem

A planning problem in PDDL consists of a *planning domain* (aspects of a problem that remain consistent, i.e., objects, predicates, actions) and a *problem description* (particular instance of a planning task, i.e., initial state, goal state).

We select the *gripper* planning problems from the International Planning Competition Vallati et al. [2015], where a robot moves balls between a set of rooms using two grippers (see Appendix A for an example). The objective is to create a *plan*—a sequence of actions—for the robot to move the balls to the defined target rooms. We present a few examples of the gripper problem in Figure 1.

2.2 PDDL Solver

The PDDL Solver takes the planning domain and the problem description as inputs and then generates a plan (a sequence of actions with specific input parameters) described in PDDL. Next, we convert the generated plan into natural language for user studies following the procedure in Seipp et al. [2022] and display it to users. We present an example in Figure 1. The planner either gener-

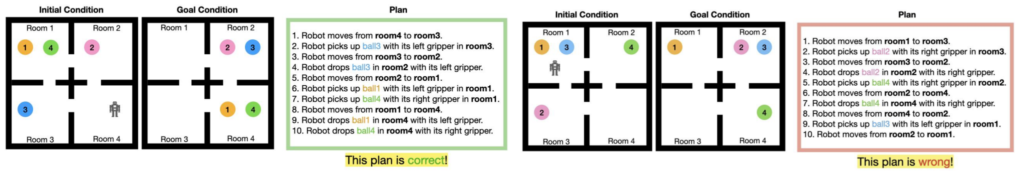


Figure 1: Examples of correct (left) and incorrect (right) plans generated for the gripper problem.

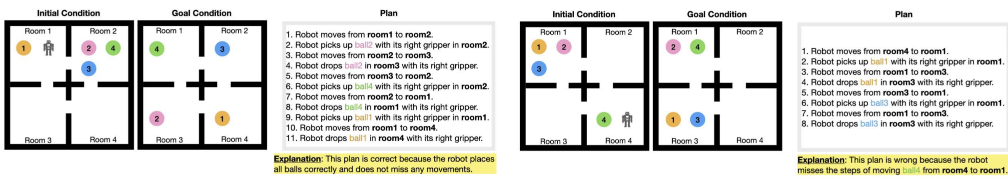


Figure 2: Examples of explanations for correct (left) and incorrect (right) plans.

ates a *correct* plan defined as the shortest path between the initial and goal states or returns a signal indicating that no solution exists for the given problem.

2.3 LLM Planner

The LLM Planner uses a structured prompt format to address planning problems by querying a large language model. The planner then retrieves a natural language plan from the language model. We include a few in-context examples within the prompts to ensure the output adheres to the desired format. We present an example of the prompts and responses in Appendix B (Listing 2).

Unlike the PDDL Solver, the LLM Planner may generate *incorrect* plans that violate the problem specifications (e.g., preconditions of actions) or fail to achieve the goal, as language models may struggle with large state spaces compared to classical planners.

LLM Planner with Explanation (LLM+Expl) To examine the influence of explanation on user trust, we create a natural language explanation of each generated plan. The trust improvement by adding explanations will motivate training an LLM to explain its plan. This explanation includes an assessment of the plan's correctness, identifying any violations of action preconditions, and an analysis of inconsistencies between the final state achieved and the intended goal state. If a plan is correct, the explanation is "the plan successfully satisfies the goal conditions." If a plan is incorrect, we identify the underlying cause as a violation of action preconditions or a failure to achieve the goal state. In cases involving precondition violations, we specify the action responsible for the issue.

For example, consider the action "robot moves from room 1 to room 2," but the robot is initially located in room 3. This scenario violates the precondition for the "move" action. In the latter case, we describe the differences between the final state achieved and the intended goal state, e.g., "fail to move ball 2 to room 2." This function enables the user to better understand why actions are chosen and their effect on the overall plan. We present examples of explanations in Figure 2.

LLM Planner with Refinement (LLM+Refine) Refining an LLM-generated plan is also possible. So, we offer a prompting mechanism for the LLM Planner to refine the generated plan according to the user feedback. We present a sample user interface on the left of Figure 3. The mechanism works as follows: First, request the user to indicate the step number where refinement should begin. Second, send the planning domain, problem description, and the original plan to the language model. Next, query the model to rewrite the subsequent steps starting from the user-specified step number. Finally, replace the original plan with the newly refined plan and display it to the user. This mechanism enables the user to focus on a subset of steps, facilitating a deeper interpretation of those actions. However, the correctness of the refined plan is still not guaranteed.

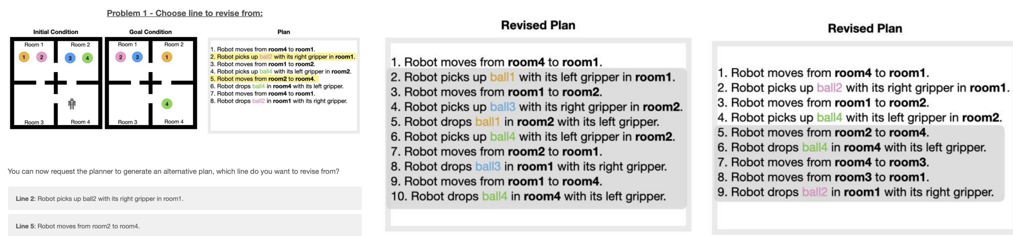


Figure 3: Plan refinement by the LLM Planner. Left: planning problem (initial and goal states). Middle: correct plan refined from step 2. Right: incorrect plan refined from step 5.

3 User Study Design

We conducted a user study via Qualtrics to evaluate human trust in plans generated by the planners discussed above. This study was approved by the University of Texas at Austin IRB#6873.

Participants. We recruited 30 fluent English-speaking adults via Prolific Palan and Schitter [2018]. After informed consent and a reCAPTCHA check, participants completed the study with bonus payments tied to evaluation accuracy (i.e., correctly accepting correct plans and rejecting incorrect ones). Participants (80% male, 17% female, 3% preferred not to say) had a mean age of 34.00 (SD=10.11). Prior LLM usage was reported by 80%. When asked about the frequency of using LLMs specifically for planning tasks, 33% indicated that they use them frequently, 43% occasionally, and 23% never.

Procedure. Participants completed four randomized sessions, each using a different planner: PDDL, LLM, LLM+Expl, and LLM+Refine (Figure 4). Each session included two Gripper tasks of similar difficulty, with plan presentation, intervention, and trust evaluation.

In each task, participants first viewed a planner-generated plan and rated their trust (**trust before**). Then, they received an *intervention*:

- **PDDL, LLM:** Only plan consequence (e.g., "This plan is correct/wrong").
- **LLM+Expl:** Consequence + explanation.
- **LLM+Refine:** Participants selected a refinement step; a revised plan was shown.

Participants then re-rated trust (**trust after**) and decided to accept or reject the plan—before the intervention for PDDL and LLM, and after for LLM+Expl and LLM+Refine. This enabled comparison of plan correctness versus intervention effects. A demo and debrief surrounded the main sessions. Evaluation accuracy was computed as the number of correctly judged plans across 8 tasks.

Independent Variables. We use a within-subjects design where each participant completes four sessions, each with one of four planners. The PDDL planner always generates correct plans (100%), while the others (LLM, LLM+Expl, LLM+Refine) produce 50% correct plans. We set this accuracy to ensure non-perfect but meaningful performance across two tasks per session, approximating the observed accuracy in practice Zuo et al. [2024], Hao et al. [2024]. LLM+Expl includes

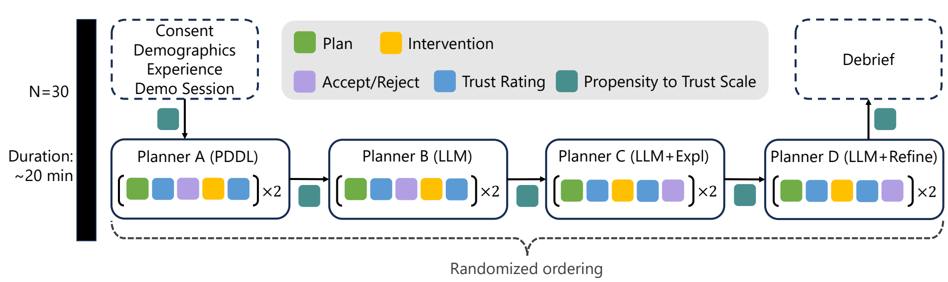
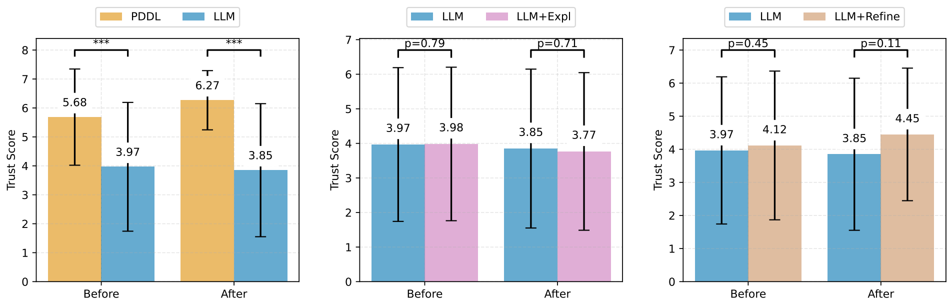


Figure 4: User study procedure. Full details in Appendix C.



(a) PDDL vs. LLM. (b) LLM vs. LLM+Expl. (c) LLM vs. LLM+Refine.

Figure 6: Trust scores on a 7-point Likert scale before and after.

plan explanations; LLM+Refine allows participants to revise the plans. The independent variables are: *correctness* (PDDL vs. LLM), *explanation* (LLM vs. LLM+Expl), and *refinement* (LLM vs. LLM+Refine).

Dependent Measures. We measure **evaluation accuracy** as the number of correctly judged tasks (0–2) and **trust**, rated on a 7-point Likert scale before and after intervention. We also assess **propensity to trust** using a 5-point Likert scale Merritt et al. [2013] (see Appendix D).

Hypotheses. We hypothesize that user performance (measured by plan evaluation accuracy) and trust are influenced by three planner properties: correctness, explanations, and refinement. Specifically, **H1**: more correct planners increase evaluation accuracy; **H2**: providing explanations increases evaluation accuracy; **H3**: allowing plan refinement increases evaluation accuracy; **H4**: more correct planners improve user trust; **H5**: providing explanations improves user trust; and **H6**: allowing plan refinement improves user trust.

4 Results & Analysis

This section presents findings from our user study on evaluation accuracy, user trust, and the propensity to trust scale.

4.1 On Evaluation Accuracy

Figure 5 shows the average number of correctly evaluated tasks per planner (error bars indicating standard deviations). We test H1–H3 using the Wilcoxon signed-rank test.

For **H1**, participants achieved an average accuracy higher with the PDDL solver (1.76 ± 0.50) than with the LLM planner (1.52 ± 0.56), supporting our hypothesis that correctness is a key determinant of evaluation accuracy. However, the difference was not statistically significant ($W = 18, Z = -4.31, p = 0.071, r = -0.801$). We suspect that increasing the sample size could reduce this uncertainty and strengthen the observed trend.

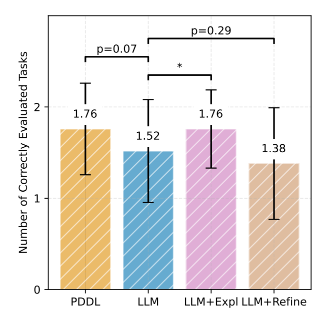


Figure 5: Evaluation accuracy measured by the number of correctly evaluated tasks.

For **H2**, evaluation accuracy improved when explanations were provided (LLM+Expl: 1.76 ± 0.43), a statistically significant gain ($W = 5, Z = -4.59, p = 0.020, r = -0.853$), supporting H2.

For **H3**, accuracy with LLM+Refine (1.38 ± 0.61) was lower than with LLM (1.52 ± 0.56), contrary to our hypothesis. The difference was not significant ($W = 22, Z = -4.23, p = 0.285, r = -0.785$), so H3 remains inconclusive. A possible explanation is overtrust: Participants may assume that the opportunity to revise the plan ensures the planner would correct itself, leading them to evaluate the revised plan less critically and, consequently, with lower accuracy.

Thus, the data suggests support for H1, confirms H2, and suggests rejection of H3.

4.2 On Trust

Figure 6 shows participants' average self-reported trust levels before and after each intervention, measured on a 7-point Likert scale, with error bars representing standard deviations. We used the Wilcoxon signed-rank test to evaluate our hypotheses H4-H6.

For **H4**, Figure 6a shows that PDDL had significantly higher trust than LLM both **before** ($W = 134.5, Z = -5.75, p < 0.001, r = -0.742$) and **after** intervention ($W = 19, Z = -6.60, p < 0.001, r = -0.852$). In terms of trust dynamics, trust in PDDL rose significantly from 5.68 ± 1.66 to 6.27 ± 1.02 ($p = 0.001$), while trust in LLM slightly declined (3.97 ± 2.22 to 3.85 ± 2.30, though this change was not statistically significant ($W = 215.50, Z = -5.15, p = 0.722, r = -0.665$). These findings support the hypothesis that correctness is a key factor influencing human trust.

For **H5**, Figure 6b shows no statistically significant difference in trust levels between LLM and LLM+Expl, both before and after the intervention. This result challenges our hypothesis that providing explanations would increase trust when correctness is controlled. One possible interpretation is that participants primarily value the objective correctness of the plans, with explanations offering little benefit unless correctness improves. Alternatively, explanations may help participants calibrate their trust by revealing the planner's limitations, allowing them to adjust their trust to appropriate levels. This insight suggests that improving trust in LLMs for planning tasks may require prioritizing the objective correctness of the plans over supplementary explanations.

For **H6**, Figure 6c shows a slight increase in trust levels with LLM+Refine. On average, trust rose from 3.97 ± 2.22 to 4.12 ± 2.25 before the intervention and from 3.85 ± 2.30 to 4.45 ± 2.00 after. While this trend is not statistically significant, it suggests a potential positive effect of refinement on human trust with the LLM planner.

Thus, the data supports H4, suggests rejection of H5, and suggests support of H6.

4.3 Propensity to Trust Scale

We include a six-item propensity to trust scale Merritt et al. [2013] to explore participants' general attitudes toward trusting AI planners. While exploratory, we find that trust increases after interacting with the PDDL planner and decreases after the LLM planner. Adding explanations shows limited recovery of trust on select items. These results suggest that plan correctness remains the dominant factor shaping trust, while explanations offer only marginal benefits when correctness is low. Full results are provided in Appendix D.

5 Discussion

Summary Our findings provide significant insights into the influence of correctness, explanations, and refinement on evaluation accuracy and user trust in AI-based planners. In particular, the findings are three-fold: (1) The **correctness** of the generated plans is the most significant factor that impacts the evaluation accuracy and user trust in the planners. As the PDDL solver is more capable of generating correct plans, it achieves the highest evaluation accuracy and trust. (2) The **explanation** component of the LLM planner improves evaluation accuracy, as LLM+Expl achieves higher accuracy than LLM alone. Despite this improvement, LLM+Expl minimally impacts user trust. However, alternative explanation methods may influence user trust differently from the manually generated explanations used in our approach. (3) The **refinement** procedure in the LLM planner does not lead to a significant improvement in evaluation accuracy; however, it exhibits a positive influence on user trust that may indicate an overtrust in some situations. Finally, the propensity-to-trust analysis identifies correctness as the primary determinant of user trust, whereas explanations provided limited improvement in scenarios where the planner's accuracy is diminished.

Future Research We plan to expand the study with more participants and diverse planning problems for more comprehensive evaluation. Other directions include exploring automated explanation methods, comparing LLMs with varying accuracy levels to assess their effect on trust, and enabling real-time user–planner interaction for collaborative plan refinement.

References

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39, 2024.

Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599*, 2020.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. Few-shot question answering by pretraining span selection. *arXiv preprint arXiv:2101.00438*, 2021.

Fang Liu, Ge Li, Yunfei Zhao, and Zhi Jin. Multi-task learning based pre-trained language model for code completion. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, pages 473–485, 2020.

Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023.

Yunhao Yang, Cyrus Neary, and Ufuk Topcu. Multimodal pretrained models for verifiable sequential decision-making: Planning, grounding, and perception. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 2011–2019, Auckland, New Zealand, 2024a. International Foundation for Autonomous Agents and Multiagent Systems / ACM.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9118–9147, Baltimore, Maryland, USA, 2022. PMLR.

Marco Cascella, Jonathan Montomoli, Valentina Bellini, and Elena Bignami. Evaluating the feasibility of chatgpt in healthcare: An analysis of multiple clinical and research scenarios. *Journal of medical systems*, 47(1):33, 2023.

Malik Sallam. Chatgpt utility in healthcare education, research, and practice: Systematic review on the promising perspectives and valid concerns. In *Healthcare*, page 887. MDPI, 2023.

Yiquan Wu, Siying Zhou, Yifei Liu, Weiming Lu, Xiaozhong Liu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. Precedent-enhanced legal judgment prediction with llm and domain-model collaboration. *arXiv preprint arXiv:2310.09241*, 2023.

Inyoung Cheong, King Xia, KJ Kevin Feng, Quan Ze Chen, and Amy X Zhang. Engaging legal experts towards responsible llm policies for legal advice. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 2454–2469, 2024.

John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1):50–80, 2004.

Eric S Vorm and David JY Combs. Integrating transparency, trust, and acceptance: The intelligent systems technology acceptance model (istam). *International Journal of Human–Computer Interaction*, 38(18-20):1828–1845, 2022.

Philip Allmendinger. *Planning theory*. Bloomsbury Publishing, 2017.

Antti Talvitie. The problem of trust in planning. *Planning Theory*, 11(3):257–278, 2012.

Lucie Laurian. Trust in planning: Theoretical and practical considerations for participatory and deliberative planning. *Planning theory & practice*, 10(3):369–391, 2009.

Michelle Cohn, Mahima Pushkarna, Gbolahan O Olanubi, Joseph M Moran, Daniel Padgett, Zion Mengesha, and Courtney Heldreth. Believing anthropomorphism: Examining the role of anthropomorphic cues on trust in large language models. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2024.

Manasi Sharma, Ho Chit Siu, Rohan Paleja, and Jaime D Peña. Why would you suggest that? human trust in language model responses. *arXiv preprint arXiv:2406.02018*, 2024.

Xin Sun, Yunjie Liu, Jan De Wit, Jos A Bosch, and Zhuying Li. Trust by interface: How different user interfaces shape human trust in health information from large language models. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2024.

Tom Silver, Varun Hariprasad, Reece S Shuttleworth, Nishanth Kumar, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Pddl planning with pretrained large language models. In *NeurIPS 2022 foundation models for decision making workshop*, 2022.

Tom Silver, Soham Dan, Kavitha Srinivas, Joshua B Tenenbaum, Leslie Kaelbling, and Michael Katz. Generalized planning in pddl domains with pretrained large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 20256–20264, 2024.

Drew M McDermott. The 1998 ai planning systems competition. *AI magazine*, 21(2):35–35, 2000.

Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning: Theory and practice*. Elsevier, 2004.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with llms enables open-world multi-task agents. In *Advances in Neural Information Processing Systems*, New Orleans,LA, USA, 2023.

Rushang Karia, Jayesh Nagpal, Daksh Dobhal, Pulkit Verma, Rashmeet Kaur Nayyar, Naman Shah, and Siddharth Srivastava. Using explainable ai and hierarchical planning for outreach with robots. *arXiv preprint arXiv:2404.00808*, 2024.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize from human feedback. *arXiv preprint arXiv:2009.01325*, 2020.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307, Long Beach, CA, USA, 2017.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, New Orleans, LA, USA, 2022.

Yunhao Yang, Neel P. Bhatt, Tyler Ingebrand, William Ward, Steven Carr, Atlas Wang, and Ufuk Topcu. Fine-tuning language models using formal methods feedback: A use case in autonomous systems. In Phillip B. Gibbons, Gennady Pekhimenko, and Christopher De Sa, editors, *Proceedings of the Seventh Annual Conference on Machine Learning and Systems*, Santa Clara, CA, USA, 2024b. mlsys.org.

Johannes Kunkel, Tim Donkers, Lisa Michael, Catalin-Mihai Barbu, and Jürgen Ziegler. Let me explain: Impact of personal and impersonal explanations on trust in recommender systems. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–12, 2019.

Sarah Strohkorb Sebo, Priyanka Krishnamurthi, and Brian Scassellati. “i don’t believe you”: Investigating the effects of robot trust violation and repair. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 57–65. IEEE, 2019.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. Llms can't plan, but can help planning in llm-modulo frameworks. *arXiv preprint arXiv:2402.01817*, 2024.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005, 2023.

Nikolas Martelaro, Victoria C Nneji, Wendy Ju, and Pamela Hinds. Tell me more designing hri to encourage more trust, disclosure, and companionship. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 181–188. IEEE, 2016.

Anqi Xu and Gregory Dudek. Optimo: Online probabilistic trust inference model for asymmetric human-robot collaborations. In *Proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction*, pages 221–228, 2015.

Jong Kyu Choi and Yong Gu Ji. Investigating the importance of trust on adopting an autonomous vehicle. *International Journal of Human-Computer Interaction*, 31(10):692–702, 2015.

Stephanie M Merritt, Heather Heimbaugh, Jennifer LaChapell, and Deborah Lee. I trust it, but i don't know why: Effects of implicit attitudes toward automation on trust in an automated system. *Human factors*, 55(3):520–534, 2013.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

Mauro Vallati, Lukas Chrpa, Marek Grześ, Thomas Leo McCluskey, Mark Roberts, Scott Sanner, et al. The 2014 international planning competition: Progress and trends. *Ai Magazine*, 36(3): 90–98, 2015.

Jendrik Seipp, Álvaro Torralba, and Jörg Hoffmann. PDDL generators. `https://doi.org/10.5281/zenodo.6382173`, 2022.

Stefan Palan and Christian Schitter. Prolific. ac—A subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, 17:22–27, 2018.

Max Zuo, Francisco Piedrahita Velez, Xiaochen Li, Michael L Littman, and Stephen H Bach. Planetarium: A rigorous benchmark for translating text to structured planning languages. *arXiv preprint arXiv:2407.03321*, 2024.

Yilun Hao, Yang Zhang, and Chuchu Fan. Planning anything with rigor: General-purpose zero-shot planning with llm-based formalized programming. *arXiv preprint arXiv:2410.12112*, 2024.

A Gripper Planning Problem

The types and predicates collaboratively define the states of a planning environment. Then, the actions define the transition of the environment. Each action consists of a set of input parameters, a precondition, and an effect. We consider the precondition and effect as the initial and final states of the action.

The types of objects of the gripper problem are defined as

```
(:types room ball robot gripper)
; there are several balls distributed in several rooms and a robot with two grippers.
```

and the predicates are defined as

```
(:predicates (at-robby ?r - robot ?x - room) ; a predicate indicating the robot's location
    (at ?o - ball ?x - room) ; a predicate indicating the ball's location
    (free ?r - robot ?g - gripper) ; a predicate indicating whether the robot's gripper is free
    (carry ?r - robot ?o - ball ?g - gripper)) ; indicating the ball carried by a gripper
```

An action moving the robot from one room to another is defined as

```
(:action move
    :parameters  (?r - robot ?from ?to - room) ; we specify the initial and target rooms
    :precondition (and  (at-robby ?r ?from)) ; the robot has to be in the initial room
    :effect (and  (at-robby ?r ?to) (not (at-robby ?r ?from))))
```

Furthermore, we have actions “pick (a ball with a gripper)” and “drop (a ball).”

```
(:action pick
       :parameters (?r - robot ?obj - object ?room - room ?g - gripper )
       :precondition  (and  (at ?obj ?room) (at-robby ?r ?room) (free ?r ?g))
       :effect (and (carry ?r ?obj ?g)
            (not (at ?obj ?room))
            (not (free ?r ?g))))

(:action drop
    :parameters (?r - robot ?obj - object ?room - room ?g - gripper)
    :precondition  (and  (carry ?r ?obj ?g) (at-robby ?r ?room))
    :effect (and (at ?obj ?room)
        (free ?r ?g)
        (not (carry ?r ?obj ?g)))))
```

An example of the initial and goal states is

```
(:init (at-robby robot1 room1) (free robot1 rgripper1) (free robot1 lgripper1)
    (at ball1 room1) (at ball2 room3) (at ball3 room1) (at ball4 room2 ) )
(:goal (and (at ball1 room1) (at ball2 room3) (at ball3 room1) (at ball4 room2) ) )
```

B Additional Details on LLM Planner

Figure 1, 2, and 3 show the complete prompt for querying the language model to solve a planning problem. The blue text represents the prompts to the language model, while the red text corresponds to the responses generated by the language model.

Listing 1: An in-context example provided to the language model to constrain its output format.

```
User: Given the following planning domain:
    (define (domain gripper-strips)
       (:requirements :strips :typing)
       (:types room object robot gripper)
       (:predicates (at-robby ?r - robot ?x - room)
                  (at ?o - object ?x - room)
                  (free ?r - robot ?g - gripper)
                  (carry ?r - robot ?o - object ?g - gripper))
       (:action move
           :parameters  (?r - robot ?from ?to - room)
           :precondition (and  (at-robby ?r ?from))
           :effect (and  (at-robby ?r ?to) (not (at-robby ?r ?from))))
       (:action pick
           :parameters (?r - robot ?obj - object ?room - room ?g - gripper)
           :precondition  (and  (at ?obj ?room) (at-robby ?r ?room) (free ?r ?g))
           :effect (and (carry ?r ?obj ?g) (not (at ?obj ?room)) (not (free ?r ?g))))
       (:action drop
           :parameters (?r - robot ?obj - object ?room - room ?g - gripper)
           :precondition  (and  (carry ?r ?obj ?g) (at-robby ?r ?room))
           :effect (and (at ?obj ?room) (free ?r ?g) (not (carry ?r ?obj ?g)))))

Please solve the following PDDL problem and return the solution only.
    (define (problem gripper-1-2-1)
        (:domain gripper-strips)
        (:objects robot1 - robot
                  rgripper1 lgripper1 - gripper
                  room1 room2 - room
                  ball1 - object)
        (:init (at-robby robot2 room1)
               (free robot1 rgripper1)
               (free robot1 lgripper1)
               (at ball1 room1) )
        (:goal (at ball1 room2) ) )

Assistant:
1. Robot1 moves from room2 to room1.
2. Robot1 picks up ball1 with its left gripper in room1.
3. Robot1 moves from room1 to room2.
4. Robot1 drops ball1 in room2 with its left gripper.
```

Listing 2: Complete prompt for querying a plan from the language model.

```
<In-context example from the previous textbox>

User: Given the same planning domain:
Please solve the following PDDL problem and return the solution only.
(define (problem gripper-1-4-4)
    (:domain gripper-strips)
    (:objects robot1 - robot
            rgripper1 lgripper1 - gripper
            room1 room2 room3 room4 - room
            ball1 ball2 ball3 ball4 - object)
    (:init (at-robby robot1 room1)
        (free robot1 rgripper1)
        (free robot1 lgripper1)
        (at ball1 room1)
        (at ball2 room3)
        (at ball3 room1)
```

```
        (at ball4 room2) )
    (:goal (and (at ball1 room1)
                (at ball2 room2)
                (at ball3 room2)
                (at ball4 room4) ) ) )
Assistant:
1. Robot1 picks up ball3 with its left gripper in room1.
2. Robot1 moves from room1 to room2.
3. Robot1 picks up ball4 with its right gripper in room2.
4. Robot1 drops ball3 in room2 with its left gripper.
5. Robot1 moves from room2 to room3.
6. Robot1 picks up ball2 with its left gripper in room3.
7. Robot1 moves from room3 to room2.
8. Robot1 drops ball2 in room2 with its left gripper.
9. Robot1 moves from room2 to room4.
10. Robot1 drops ball4 in room4 with its right gripper
```

Listing 3: Prompt for querying the LLM Planner to refine an existing plan.

```
Given the planning domain and problem:
(define (domain gripper-strips)
    ......)
(define (problem gripper-1-2-1)
    (:domain gripper-strips)......)

Please refine the plan from step 2:
1. Robot1 moves from room2 to room1.
2. Robot1 picks up ball1 with its left gripper in room1.
3. Robot1 picks up ball2 with its right gripper in room2.
4. Robot1 drops ball3 in room2 with its left gripper.
1. Robot1 moves from room2 to room1.
2. Robot1 picks up ball1 with its left gripper in room1.
3. Robot1 moves from room1 to room2.
4. Robot1 drops ball1 in room2 with its left gripper.
```

C User Study Procedure

Participants began with a *demo session* to familiarize themselves with the Gripper problem task and the study interface.

The main part of the study comprised *four sessions*, each corresponding to a different AI planner: *Planner A (PDDL)*, *Planner B (LLM)*, *Planner C (LLM+Expl)*, and *Planner D (LLM+Refine)*. The four sessions were presented in a randomized order to counterbalance any ordering effects.

Each session contained *two tasks*, each with a new Gripper problem (unique initial and goal conditions) of similar difficulty (comparable number of plan steps, rooms, and balls). Task order was randomized within each session. In each task, participants first viewed a *plan* generated by the planner and rated their trust in the planner (**trust before**). They were then shown an *intervention*, which varied depending on the session planner:

- For **PDDL** and **LLM**, the intervention provided only the *consequence of the plan*, e.g., *“This plan is correct/wrong!”*.
- For **LLM+Expl**, the intervention included both the consequence of the plan and an *explanation* of the outcome, e.g., *“This plan is wrong because the robot misses the steps of moving ball4 from room4 to room1.”*
- For **LLM+Refine**, participants were first asked to *choose between two lines* of the plan as a starting point for refinement. A *revised plan* was then generated beginning from the selected line.

After the intervention, participants rated their trust in the planner (**trust after**). They also chose **accept or reject** the plan before the intervention for PDDL and LLM planners, but after for the other two. This allows evaluation of plan correctness prior to consequences for PDDL and LLM, while focusing on responses to interventions for the others.

At the end of the study, participants were informed of their *evaluation accuracy* as the total number of correctly evaluated tasks out of 8 total tasks (2 tasks per session, 4 sessions in total). The procedure is detailed in Figure 4.

D Propensity to Trust

At the end of each session, participants rated their agreement with six statements adapted from the Propensity to Trust Machines scale Merritt et al. [2013], replacing "machine" with the relevant planner label (e.g., "Planner A," "Planner B"). Each item was rated on a 5-point Likert scale:

- 1 — Strongly Disagree
- 2 — Somewhat Disagree
- 3 — Neither Agree Nor Disagree
- 4 — Somewhat Agree
- 5 — Strongly Agree

The statements were:

1. I usually trust Planner A until there is a reason not to.
2. For the most part, I distrust Planner A.
3. In general, I would rely on Planner A to assist me.
4. My tendency to trust Planner A is high.
5. It is easy for me to trust Planner A to do their job.
6. I am likely to trust Planner A even when I have little knowledge about it.

Figure 7 presents the full results across planners. While this scale was exploratory, four items showed statistically significant differences between planners using the Wilcoxon signed-rank test

For **Q1** and **Q6**, we observe a clear shift toward agreement after the PDDL condition compared to the initial baseline. This suggests that participants were more inclined to trust AI planners following the PDDL session, likely due to the 100% correctness of PDDL plans, which appears to boost trust. In contrast, for **Q1**, **Q4**, and **Q5**, we see a notable reduction in agreement after interacting with the LLM planner compared to the PDDL solver. This decrease aligns with the reduced correctness of the LLM plans (50%), highlighting the importance of correctness in maintaining trust in AI planners. Interestingly, **Q4** reveals that providing explanations (LLM+Expl) helps recover participants' agreement levels compared to the basic LLM condition. However, this positive effect of explanations on trust propensity is limited, as it is only observed in one of the six questions.

These results underscore that correctness remains the dominant factor influencing participants' general trust attitude towards AI planners, with explanations offering only minimal benefit when correctness is suboptimal.

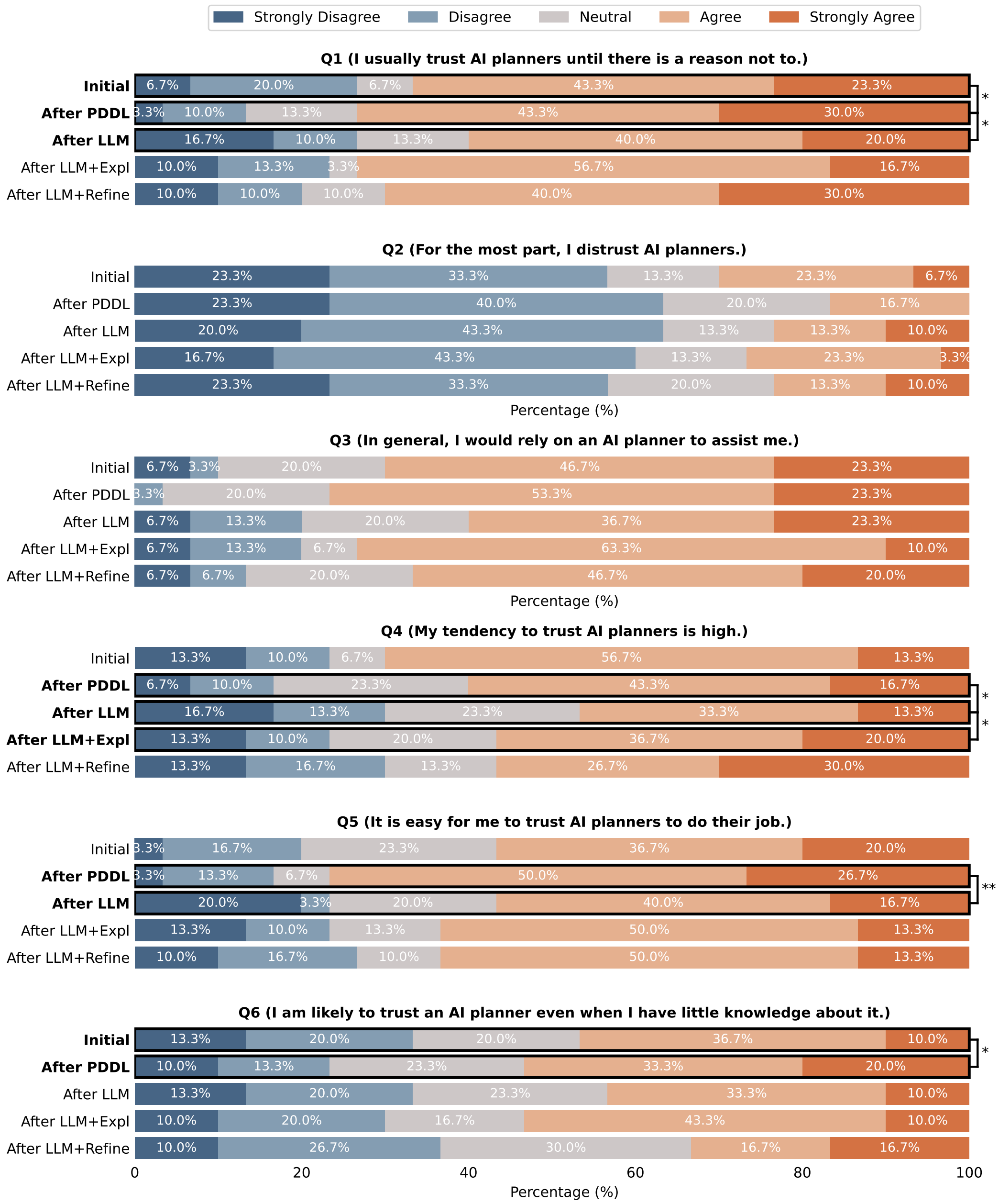


Figure 7: Complete Propensity to trust scale result