FixMyPlan: Leveraging Large Language Models to Fix Ill-Defined Models and Incorrect Plans

Anonymous submission

Abstract

Large Language Models (LLMs) have shown promise in generating formal representations such as PDDL in Classical Planning, capable of producing parsable and solvable code; however, despite these recent breakthroughs, they are limited to the natural language ambiguity of the user's descriptions of the model and can result in semantically incorrect or infeasible real-world plans. We propose *FixMyPlan*, a general framework that leverages the common sense capabilities of LLMs to judge the semantics of error-prone PDDL plans and back prompt to fix their corresponding models in a closed-loop fashion, produce coherent plans—all while minimizing human intervention. We conduct experiments on 5 flawed PDDL domains, producing solvable—yet *incorrect* plans: Blocksworld, Logistics, Mystery Blocksworld and Logistics, and our self-produced domain. We aim to analyze common pitfalls such as semantic ambiguity, unintentional constraints, and logical inconsistencies that hinder effective plan generation and alignment with real-world tasks.

1 Introduction

With the limitations of Large Language Models (LLMs) in direct planning tasks (Valmeekam, Stechly, and Kambhampati 2024; Pallagani et al. 2023; Momennejad et al. 2023), Automated Planning (AP), or AI planning, emerges as a promising alternative, offering a robust and logic-driven solution to direct LLM planning challenges, while LLMs complement it by extracting and refining classical planning models from natural language for effective plan generation. A significant body of research has focused on LLMs in model extraction for planning (Xie et al. 2023; Guan et al. 2023; Gestrin, Kuhlmann, and Seipp 2024; Liu et al. 2023), bringing light to LLM-driven planning approaches. While LLMs can produce syntactically valid, solvable plans, it remains uncertain whether these models align with user intentions and if the resulting plans, when generated by external planners, can be effectively achieved in real-world scenarios. Furthermore, achieving semantically correct plans entirely depends on the accuracy and adaptability of the underlying domain models. Often, LLM-extracted domain models contain inconsistencies, ambiguities, or unintended constraints, which can misalign the generated plans with practical goals or operational environments.

2 Problem Statement

External planners, as previously noted, are capable of generating robust plans; however, the quality of these plans is reflected in the accuracy and completeness of the model's setup. For example, in the Blocksworld domain, a user might inadvertently omit a critical constraint, such as failing to include the effect `(arm-empty)` in the *unstack* action. While an external planner might still produce a solvable plan, it could lack logical coherence. Such oversights by the user can lead to significant consequences, particularly in critical real-world applications.

To tackle this issue, our work focuses on using LLMs to detect these logically flawed plans as a foundation for generating semantically accurate ones. Specifically, we propose *FixMyPlan*, a general framework that leverages the commonsense capabilities of LLMs to perform an in-depth analysis and critique of plans generated by classical planners. This framework identifies semantic inconsistencies and assesses whether these plans align with user requirements, thereby facilitating the refinement of the domain model. To our knowledge, this is the first effort that focuses on repairing domain models in the realm of plan *correctness* while maintaining the formally structured encoding in the Planning Domain Definition Language (PDDL). Our contributions can be summarized as follows:

- Leveraging LLMs to capture implicit constraints and underlying assumptions, overcoming the bottleneck of subjective interpretation and semantic ambiguity.
- Analyzing common pitfalls and providing insight into where LLMs fail to accurately interpret natural language requirements, leading to misaligned and incomplete domain models that hinder effective, real-world plan generation.
- **FixMyPlan**, a novel general framework that allows users to interchange LLMs and prompts to optimize planning workflows, and seamlessly integrate different LLM-based approaches for various planning tasks—in hopes to encourage further research in this direction.

3 FixMyPlan

FixMyPlan (Figure 1) leverages LLMs' commonsense reasoning to detect semantic errors and propose edits directly

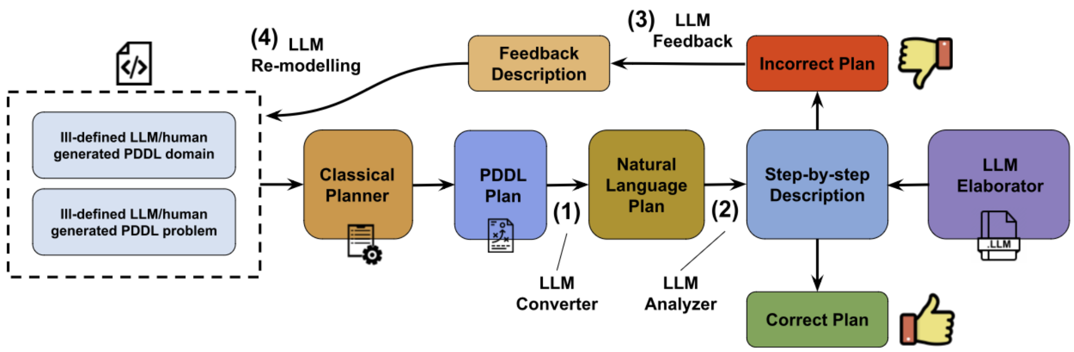

Figure 1: FixMyPlan general framework

on ill-defined PDDL models. The pipeline comprises four stages. Each step is fully automated, relying solely on LLMs without human input. Recognizing users often overlooking constraints when extracting these domain models—often creating vague natural language descriptions, an additional component, *LLM Elaborator*, involves the LLM capturing implicit constraints and underlying assumptions by tasking them to extend the relations between predicates and interactions between actions. This will provide a stronger prompt for the LLM to reflect the given plans with the descriptions of the environment.

Step One - LLM Conversion: Given the domain and problem description, their PDDL counterparts, and the PDDL plan, this converts PDDL plans into natural language via LLMs for interpretability. For each step in the plan, the LLM outputs the action preconditions met, effects made, and a summary of the action taken reflective of the state of the environment.

Step Two - LLM Judge: Given the natural language plan, the LLM analyzes each step for semantic accuracy using its commonsense reasoning. Specifically, we prompt the LLM to assess whether the preconditions of each step can be satisfied based on prior steps in the plan. Finally, we ask the LLM if the plan as a whole is semantically sound: if it is, it responds with '[PASS],' confirming the plan as correct. If not, it responds with '[FAIL]' and a summary of why it failed, which then initiates the next stage.

Step Three - LLM Feedback: The LLM is tasked with conducting a comprehensive analysis of the domain specification with the provided domain and problem descriptions, their PDDL counterparts, and the issue identified in the previous steps. We supply a feedback checklist guiding the LLM. The final output includes a suggestion section aimed at refining the domain model.

Step Four - LLM Re-modelling: We use the feedback to refine the domain model. This is formatted back into PDDL format for reassessment.

At the end of the pipeline, we assess whether the models generate solvable plans. Solvable plans undergo iterative refinement to address further semantic inaccuracies. Unsolvable plans revert to the original for rerunning, avoiding the compounding issues that previous research has shown can arise when LLMs attempt self-refinement (Stechly, Marquez, and Kambhampati 2023; Valmeekam, Marquez, and Kambhampati 2023; Huang et al. 2024). This process repeats for a fixed number of iterations, with the LLM operating at higher temperatures to explore diverse solutions for addressing incorrect plans.

4 Experimentation Setup

FixMyPlan will be evaluated across five domains: Blocksworld, Logistics, Mystery Blocksworld, Mystery Logistics, and a custom-designed domain. Drawing inspiration from (Kambhampati et al. 2024), we obfuscated predicates and actions to create the Mystery domains, rendering them semantically nonsensical yet logically valid. Our custom domain, which scales in difficulty, is specifically designed to ensure it has not appeared in any LLM training corpus. To assess its performance, we will modify domain PDDL instances by altering action schemas—specifically adding or removing predicates from preconditions and effects—while ensuring the modified instances still yield solvable (yet incorrect) plans using the *FastDownward* planner. We will be conducting experiments with GPT4o, GPT4o-mini (OpenAI 2023), and Llama-3.1-8B-Instruct-evals (Touvron et al. 2023) as the underlying LLMs; although this framework is general enough to support different LLMs. Our baseline will utilize a zero-shot Chain-of-Thought method to evaluate whether this framework outperforms direct prompting from an LLM.

5 Future Work

In future evaluations, we aim to explore several ways to enhance LLM performance. First, future work could investigate the impact of incorporating domain-specific human feedback or external APIs (into step three). By integrating expert feedback, we can better assess how such input improves the accuracy and relevance of the LLM's output, especially in more complex or highly specialized domains. Second, to address the challenge of the LLM context token window limit, we could provide a current state predicate list from the plan for the LLM to compare its updated states against, helping it detect misalignments. This could be further mitigated by implementing a dynamic windowing approach that prioritizes key predicates.

References

Gestrin, E.; Kuhlmann, M.; and Seipp, J. 2024. NL2Plan: Robust LLM-Driven Planning from Minimal Text Descriptions. *CoRR*, abs/2405.04215.

Guan, L.; Valmeekam, K.; Sreedharan, S.; and Kambhampati, S. 2023. Leveraging Pre-trained Large Language Models to Construct and Utilize World Models for Model-based Task Planning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Huang, J.; Chen, X.; Mishra, S.; Zheng, H. S.; Yu, A. W.; Song, X.; and Zhou, D. 2024. Large Language Models Cannot Self-Correct Reasoning Yet. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Kambhampati, S.; Valmeekam, K.; Guan, L.; Stechly, K.; Verma, M.; Bhambri, S.; Saldyt, L.; and Murthy, A. 2024. LLMs Can't Plan, But Can Help Planning in LLM-Modulo Frameworks. *CoRR*, abs/2402.01817.

Liu, B.; Jiang, Y.; Zhang, X.; Liu, Q.; Zhang, S.; Biswas, J.; and Stone, P. 2023. LLM+P: Empowering Large Language Models with Optimal Planning Proficiency. *CoRR*, abs/2304.11477.

Momennejad, I.; Hasanbeig, H.; Frujeri, F. V.; Sharma, H.; Jojic, N.; Palangi, H.; Ness, R. O.; and Larson, J. 2023. Evaluating Cognitive Maps and Planning in Large Language Models with CogEval. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Pallagani, V.; Muppasani, B.; Murugesan, K.; Rossi, F.; Srivastava, B.; Horesh, L.; Fabiano, F.; and Loreggia, A. 2023. Understanding the Capabilities of Large Language Models for Automated Planning. *CoRR*, abs/2305.16151.

Stechly, K.; Marquez, M.; and Kambhampati, S. 2023. GPT-4 Doesn't Know It's Wrong: An Analysis of Iterative Prompting for Reasoning Problems. *CoRR*, abs/2310.12397.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR*, abs/2302.13971.

Valmeekam, K.; Marquez, M.; and Kambhampati, S. 2023. Can Large Language Models Really Improve by Self-critiquing Their Own Plans? *CoRR*, abs/2310.08118.

Valmeekam, K.; Stechly, K.; and Kambhampati, S. 2024. LLMs Still Can't Plan; Can LRMs? A Preliminary Evaluation of OpenAI's o1 on PlanBench. *arXiv preprint arXiv:2409.13373*.

Xie, Y.; Yu, C.; Zhu, T.; Bai, J.; Gong, Z.; and Soh, H. 2023. Translating Natural Language to Planning Goals with Large-Language Models. *CoRR*, abs/2302.05128.