**Title:** LLM-Planner: Few-Shot Grounded Planning for Embodied Agents with Large Language Models

**Reviewer:**
Raman Kumar Jha
RJ2712

**Summary:**
Traditional methods for training language-driven agents require extensive labeled instruction-trajectory pairs, making them data-inefficient and impractical for real-world scalability. The authors propose LLM-Planner, which uses large language models (LLMs) like GPT-3 for few-shot hierarchical planning. Key innovations include: LLMs generate high-level plans (HLPs) as subgoal sequences (e.g., "Navigate to potato → Pickup potato"), while low-level planners execute these subgoals. When agents encounter obstacles, the LLM regenerates HLPs using real-time environmental observations (e.g., detected objects) for grounding. Combines with perception modules and low-level controllers (e.g., HLSM) without retraining. Unlike prior work (e.g., SayCan), LLM-Planner does not require enumerating admissible actions beforehand, reduces API calls to LLMs, and dynamically adapts plans to environments. On the ALFRED benchmark, LLM-Planner achieves a 13.41% success rate (unseen environments) using <0.5% of training data, rivaling models trained on full datasets. Existing methods fail under the few-shot setting.

**Strengths:**

1. Achieves competitive performance with 100 examples versus baselines requiring 21,000+ examples. This reduces reliance on costly human annotations.
2. DRe-planning based on object detection (e.g., rerouting to a fridge if a potato isn't found) addresses LLMs' lack of environmental awareness.
3. Integrates seamlessly with existing VLN frameworks (e.g., HLSM, FILM) without architectural changes.

4. Requires ~7 GPT-3 calls per task vs. SayCan's 22 calls, lowering computational expense.

Yes, the paper proposes a new formulation, focusing on planning, but utilizing LLMs networks. It uses LLM for the planning task, which is a new angle for looking at the problem. The model and the experiments are ingenuine. The performance stands out as the utilization of LLMs has been performed very efficiently.

**Weaknesses:**

1. No exploration of smaller/open-source LLMs (e.g., LLaMA), limiting reproducibility.
2. The low-level planner relies on synthetic data; errors here could cascade despite accurate HLPs.
3. Evaluated only on ALFRED's simulated home environments. Real-world noise (e.g., ambiguous instructions) remains untested.
4. Frequent GPT-3 calls in dynamic re-planning may be prohibitive for large-scale deployment.

Yes, the premise of the paper, i.e. using LLM for planning makes sense to me. Yes, the methodology used is comprehensive for me. The real-world experiments are missing in this paper, which could have helped to prove the point. All the potential limitations have been discussed above.

**Possible Future Extensions:**

1. Incorporate visual-language models (e.g., Flamingo) to improve object grounding without separate detectors. Increased complexity in prompt engineering, can be a risk here.
2. Combine LLM-Planner with reinforcement learning to refine low-level policies during execution. The risk includes training instability in hybrid systems.
3. Test generalization to robotics platforms (e.g., ROS) or non-home environments. Domain shifts in object affordances or action spaces can be an issue in this case.

*Please list 2-3 possible extensions based on this work. Explain why you think these extensions are viable and what are the potential risk factors.*

**Conclusion:**

LLM-Planner advances few-shot planning for embodied agents by leveraging LLMs' commonsense reasoning and dynamic re-planning. While the approach is resource-efficient and modular, its reliance on proprietary LLMs and simulated benchmarks limits immediate real-world applicability. Despite these caveats, the paper makes a significant contribution to sample-efficient embodied AI. In a review setting, I would give this work a positive score for its novel integration of LLMs with hierarchical planning and grounded execution.