PlanGenLLMs: A Modern Survey of LLM Planning Capabilities

Hui Wei[1], Zihao Zhang[2], Shenghua He[3], Tian Xia[3], Shijia Pan[1], Fei Liu[2]
[1] University of California, Merced, [2] Emory University, [3] PAII Inc.

Paper GitHub

We present a comprehensive survey of current LLM planners, overviewing key performance criteria, examining evaluation metrics, methods, and datasets, and outlining future work directions.

LLM Planning Foundations

Task Decomposition
- What: Abstract goals into specific subgoals.
- Why: (1) Help mitigate errors; (2) Make LLM reasoning more tractable.
- How: (1) Sequentially; (2) In parallel; (3) Asynchronously; (4) Recursively.

LLM + Classic Planner
- What: LLM + Classic Planner (e.g., Fast Downward).
- Why: Combine the world knowledge of LLMs with the precision and reliability of classical methods.
- How: (1) Natural Language to formal representation; (2) Generate initial plans.

Search Algorithm
- What: BFS, DFS, MCTS, Greedy Best First Search.
- Why: Provides systematic exploration, optimality guarantees, and formal verification.
- How: (1) Search Policy; (2) Expansion; (3) World Models; (4) Evaluation.

Fine-tuning
- What: Update pretrained LLMs parameters.
- Why: Enhance planning correctness fundamentally.
- How: (1) Planning specific tasks; (2) Broader agentic capabilities.

Criterion I: Completeness

Correct Plan Generation
- What: If a valid plan exists, the model should generate it correctly.
- How: (1) LLM + sound and complete solvers; (2) LLM accurately translates the domain and problem.

Unsolvable Problem Recognition
- What: if no feasible plan, the model should identify it and not generate an incorrect or arbitrary plan.
- Currently: LLMs and LRMs struggle on this due to hallucination.

Criterion II: Executability

Executability
- What: If a plan can be carried out (1) in a given environment (2) while meeting all constraints.
- Note: (1) *an executable plan isn't necessarily correct*; (2) *a correct plan isn't always executable*.
- How: (1) Object Grounding, (2) Action Grounding, (3) Sample-then-Filter, (4) Closed-Loop Systems.

Object Grounding: ensure the LLM planner uses objects available in the current environment.

Action Grounding: ensure all actions in a plan can actually be executed in the current environment.

Sample-then-Filter: generate multiple plans and then verifies them, selecting only those that pass all checks.

Closed-Loop System: the planner adapts its plan based on feedback from executors, simulators, validators, other LLMs, or even humans, when the initial plan are inexecutable.

Criterion III: Optimality

Optimality
- What: Achieving the goal state through the *best* possible plan.
- How: (1) LLM + Optimizer; (2) A* Search-Based Methods.

LLM + Optimizer: combines the LLM (turns user requests into symbolic optimization problems) with an optimizer (solves them and finds the best solution).

A* Search-Based Methods: Integrate LLM with A* search, which finds the lowest-cost optimal solution, when the actual cost to the goal is not overestimated.

Criterion IV: Representation

Representation
- What: how inputs and outputs are formatted.
- Inputs: domains (predicates and actions), problems (initial and goal states), and environmental observations.
- Outputs: generated plans.

LLM-as-a-Translator
- LLMs convert between natural language and formal representation (e.g., PDDL, STL, LTL).

LLM-as-a-Planner
- Environment and domain: natural language, tables, condensed symbols, Pythonic code, neural embeddings, graphs.
- Generated Plan: natural language, Pythonic code.

Criterion V: Generalization

Generalization
- What: LLM planners' ability to apply learned strategies to new, more complex out-of-domain scenarios beyond its training environment.
- How: (1) Fine-tuning; (2) Generalized planning; (3) Skill Storage.

Generalized Planning
- Generalized planning extracts common patterns from a limited set of training solutions (i.e., plans) to solve unseen tasks within the same domain, which may be larger and more complex than the training tasks.

Skill Storage
- Skill storage focuses on learning and reusing previously acquired skills to tackle new problems.

Criterion VI: Efficiency

Efficiency
- What: Efficiency in LLM planning means reducing *computational* and *monetary* costs.
- Why: This is crucial especially developing planners based on commercial LLMs.
- How: decreasing (1) LLM calls and world model interactions, (2) input and output lengths, and (3) model sizes.

Evaluation

Dataset
- *Planning-focused datasets*: (1) Embodied environments, (2) Task scheduling , (3) Games, and (4) Task decomposition.
- *Downstream-task datasets*: (1) Agentic tasks, including reasoning-oriented tasks, tool-use-oriented tasks, programming tasks, and web tasks, (2) Generation tasks, including video, image and text generation.

Metric
- *Completeness*: Success Rate; Goal Condition Recall; Step Success Rate; Exact Match Score; True Negative Rate and False Negative Rate; Unreachable Accuracy.
- *Executability*: Executability Rate; Constraint Pass Rate.
- *Optimality*: Optimality Rate.
- *Efficiency*: Inference Time; Number of Output and Input Tokens; Number of plan Steps; Number of LLM and World Model Calls; Model Size.
- *Representation*: Number of Parsable Problems.
- *Generalization*: All above metrics can also be applied to unseen scenarios to assess generalization.

Method
- *Verified by verifier or Compared with ground-truth*: (1) When the plan is tested in a simulated environment, internal or external verifiers are used to verify it; (2) When the ground-truth label is available, generated plans can be compared against reference plans.
- *Human Evaluation*: Applied when (1) No available verifier; (2) Open-ended problems.
- *LLM-as-a-Judge*: (1) Pros: faster and more cost-effective than human evaluation; (2) Cons: Internal limitations (e.g., position bias, length bias).

Future Directions

- **Datasets and Baselines**: *establishing a public, standardized leaderboard* with diverse datasets, consistent metrics, and a variety of baseline and advanced methods.
- **Representation**: *building benchmarks* and *carefully choosing representation formats* in experiments.
- **Hallucination**: improving LLMs' ability to *accurately identify unachievable plans* and *evaluating the impact* of hallucinations.
- **Human Preference Alignment**: better aligning LLM planners with human preferences.
- **Cost-Effectiveness**: *summarizing problem descriptions* and *enhancing heuristic evaluations*, **Multi-Agent Planning**: more attention should be received by multi-agent planning.
- **Reasoning, Tool Use, and Memory:** looking into *enhancing these agentic capabilities* in LLM-based planning.