Evaluations

for reliable planning

Harsha Kokel

IBM Research

Outline

- Core Reasoning Tasks for Reliable Planning
- ACPBench Dataset
- Evaluation with LM-Eval Harness
- Planning Benchmark Desiderata
- Countdown domain

Reasoning for Planning

Planning

Outline

- **Core Reasoning Tasks for Reliable Planning**
- ACPBench Dataset
- Evaluation with LM-Eval Harness
- Planning Benchmark Desiderata
- Countdown domain

Reasoning Tasks

Action Applicability

Progression

Reachability

Action Reachability

Validation

Justification

Landmark

Next Action

The first step of intelligent agent isn't choosing the best action—it's recognizing the valid ones.

1. Action Applicability

Action Applicability

What

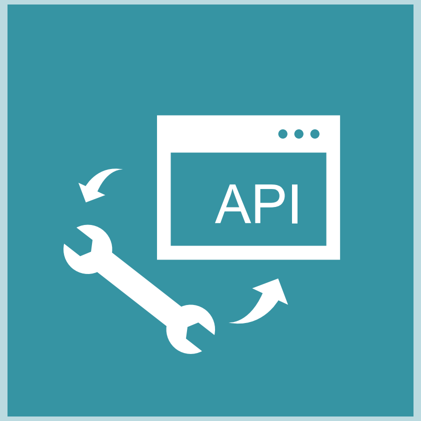
The ability of an agent to identify which actions are valid and executable in a given state or context.

When

Presume validity of precondition or overlooked them

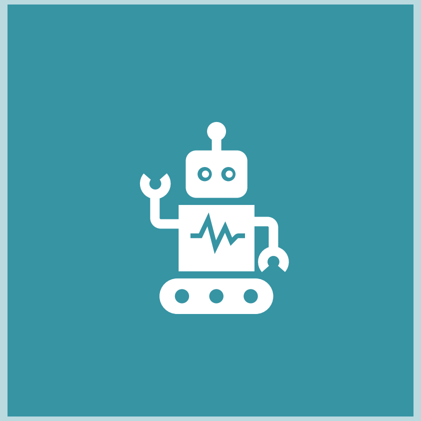
Examples

Seen these before?



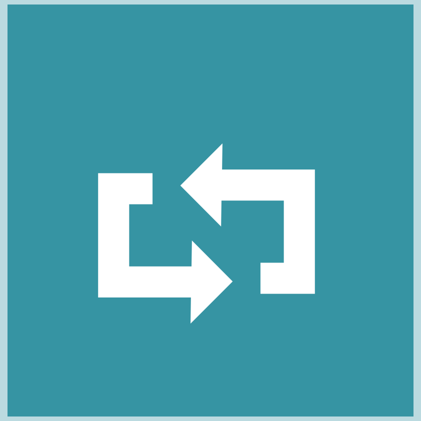
Execution Hallucination

Chose non-existent tools or involve fabricated parameters



Impossible Action

Pick objects that do not exist or move through walls



Dead Loop

Fails to recognize invalid actions and keeps repeating

Incorrect Sequencing

Skips a prerequisite or performs actions out of sequence

Why

Should we care?

Prevents invalid and impossible actions

Enables correct sequencing in multi-step plans

Prevents unnecessary thinking and processing

Planning isn't just about choosing actions — it's about understanding what those actions do.

2. Progression

Progression

What

The ability of an agent to understand how the world state changes after performing an action

When

Missing effect prediction, incorrect outcomes, or wrong state persistence

Examples

Seen these before?

Incorrect State Update

Lose track of newly generated IDs or assumes old inventory persists

Invalid Moves

Move deleted objects or lock a locked door

Ignore side effects

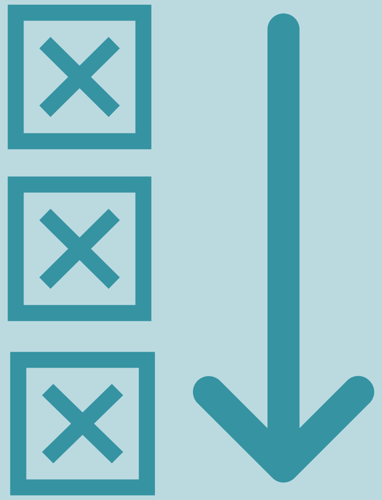
After “canceling a subscription,” assumes premium features are still available

State Loss

Lose track of shuffled objects

Why

Should we care?



Reliable Agents Need a Coherent World Model

Multi-step plans rely on correct state evolution

Prevents cascading errors

Effective agents must distinguish achievable goals from unreachable ones

3. Reachability

Reachability

What

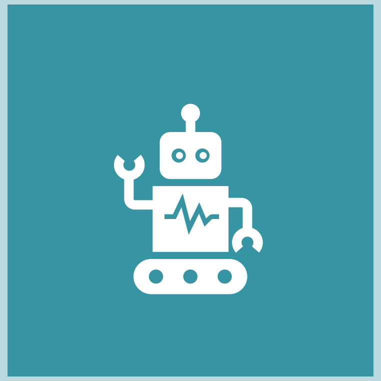
The ability of an agent to determine whether a specific goal or state can be reached from the current state through a sequence of valid actions.

Why

avoid nonsensical tool usage, avoid unnecessary search, prevent wasted resources and infinite loops

Examples

Attempt tasks that no tools provide or even explicitly prohibits



Attempt pathfinding for blocked goals

Planning is pointless if the agent assumes it can perform actions that will never become available.

4. Action Reachability

Action Reachability

What

The ability of an agent to evaluate whether an action can ever become applicable along any valid future trajectory

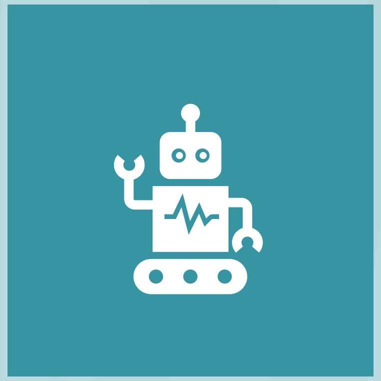
Why

avoid unnecessary search, avoid nonsensical tool usage

Basically, ensure efficient use of resources.

Examples

Accesses resources or invokes tools that do not exists



Attempts to press button higher than arm reach

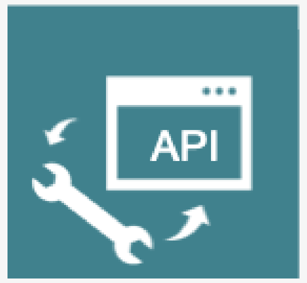
Even one incorrect step breaks the whole plan, so detecting the earliest failure is crucial.

5. Validation

Validation

What

The ability of an agent to verify that an action sequence is executable and actually achieves the goal.



Supervisor agent assumes a sequence is executable even when it is missing a prerequisite step

A critic or a judge agent fails to flag an infeasible step

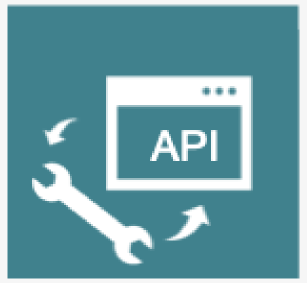
Efficient problem-solving requires identifying and removing redundant actions.

6. Justification

Justification

What

The ability of an agent to detect an unjustified actions in a plan and simply the plan without losing validity or goal achievement



Agent over compresses plans that makes it invalid

A critic or a judge agent fails to flag a redundant step

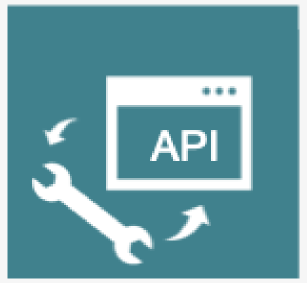
Progress toward a goal depends on hitting certain necessary milestones that structure the planning landscape.

7. Landmark

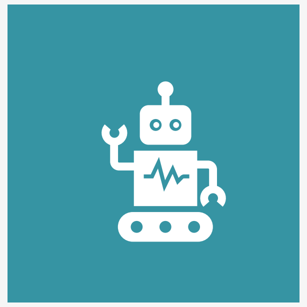
Landmark

What

The ability of an agent to recognizes mandatory subgoals that every valid plan must pass through.



Ignore domain mandated subgoals.
Skipped “git commit” before “git push”.



Collapses subgoal hierarchy --- places books without ensuring correct orientation when asked to keep book verticaly.

Finally, Choosing the right next step is what turns understanding into purposeful action

8. Next Action

Summary

Action Applicability

Progression

Reachability

Action Reachability

Validation

Justification

Landmark

Next Action

More...

Goal Recognition

Plan Generation

Cost Estimation

Hierarchical Decomposition

ACPBench

Action Applicability

Progression

Reachability

Action Reachability

Validation

Justification

Landmark

Next Action

Kokel, Katz, Sohrabi, Srinivas AAAI 2025

Outline

- Core Reasoning Tasks for Reliable Planning
- **ACPBench Dataset**
- Evaluation with LM-Eval Harness
- Planning Benchmark Desiderata
- Countdown domain

Benchmarks

https://plan-fm.github.io/2025/

PlanBench	Auto PlanBench	TRAC	LLM+P	ActionReasoning Bench	ACPBench

LLMs as Planning Formalizers: A Survey for Leveraging Large Language Models to Construct Automated Planning Models, Tantakoun et al. ACL 2025

Other

- NL Planning Benchmarks
  - Travel Planner, Kie et al ICML 24 (https://osu-nlp-group.github.io/TravelPlanner/)
  - Natural Plan, Zheng et al 24 (https://github.com/google-deepmind/natural-plan)
- NL to PDDL translations
  - NL2PDDL, Oswald et al ICAPS 24 (https://github.com/IBM/NL2PDDL)
  - LLM+P, Liu et al 23 (https://github.com/Cranial-XIX/llm-pddl/)
  - Planetarium, Zuo et al 24 (https://github.com/BatsResearch/planetarium)
- Agent
  - Agent Board, Ma et al NeurIPS 24 (https://github.com/hkust-nlp/AgentBoard)
  - TextCraft Prasad et al. NAACL 24 (https://github.com/archiki/ADaPT/tree/main/TextCraft)
  - ALFRED, ALFWorld, WebShop, WebArena etc…

Benchmarks Tutorial at PLAN-FM Bridge, AAAI 2025

Valmeekam et al. NeurIPS 23, Stein et al. 23, He et al. ACL 23, Handa et al. 2025

Dataset

ACPBench

Reasoning about Action, Change, and Planning

Harsha Kokel, Michael Katz, Kavitha Srinivas, Shirin Sohrabi

IBM Research

harsha.kokel@ibm.com, michael.katz1@ibm.com, kavitha.srinivas@ibm.com, ssohrab@us.ibm.com

ACPBench | ACPBench-Hard | Code | Dataset | Evaluation

https://ibm.github.io/ACPBench/

ACPBench

Action Applicability

Progression

Reachability

Action Reachability

Validation

Justification

Landmark

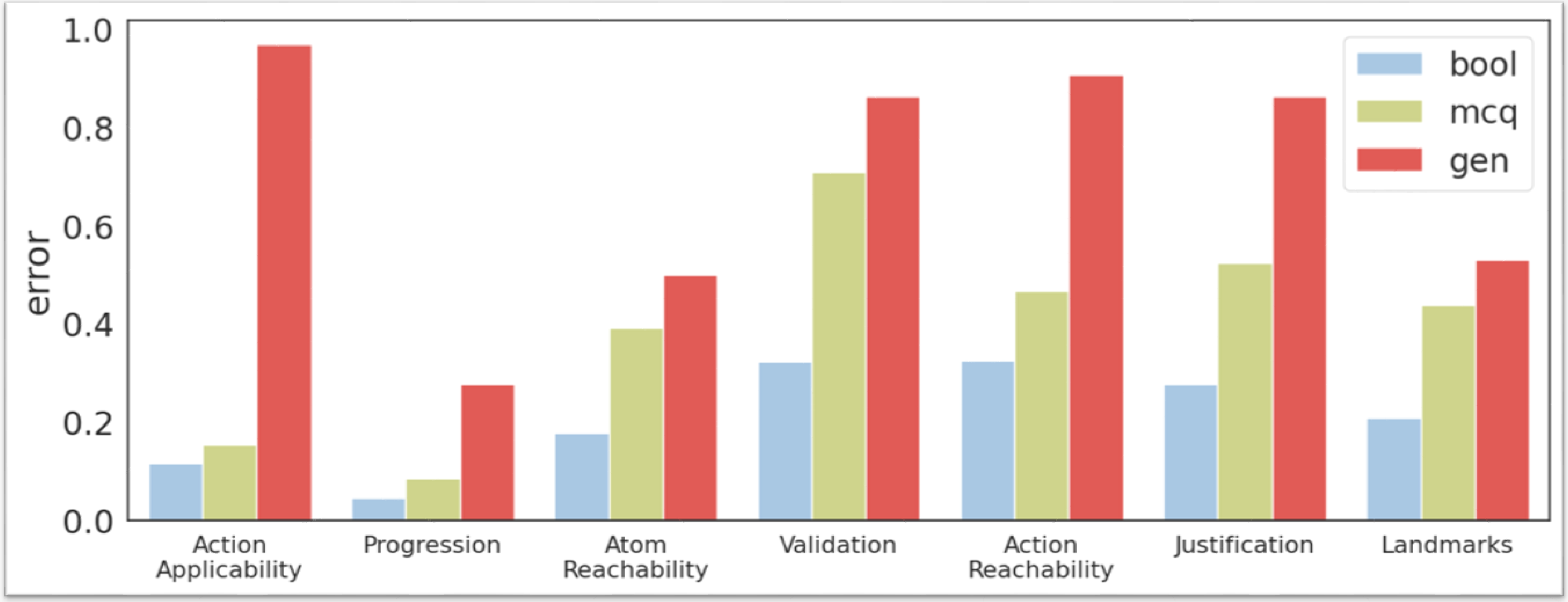
Next Action

11 classical planning domains, ALFWorld, and a novel Swap

3 formats: Boolean, Multi-choice and Generative

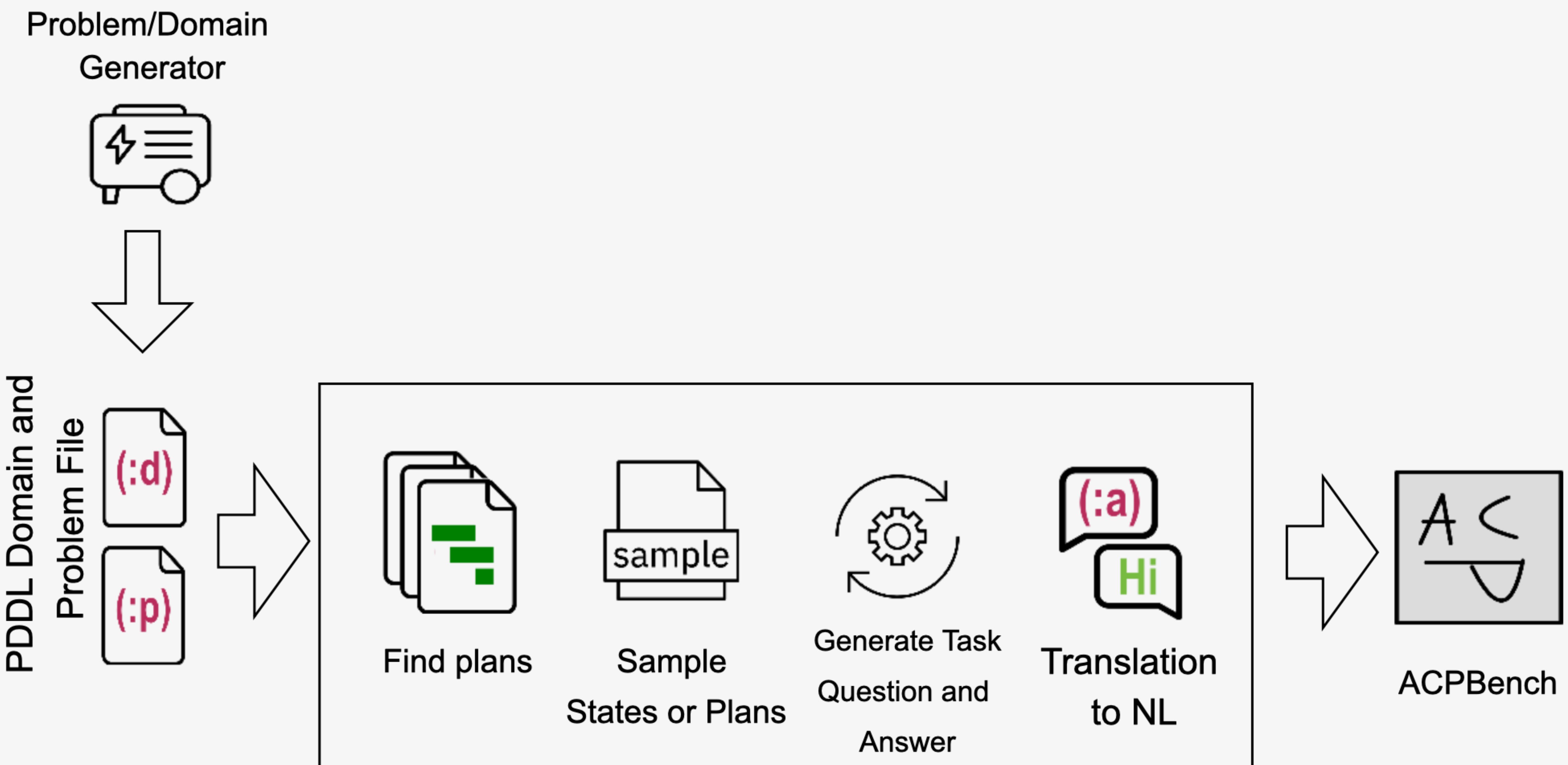
Kokel, Katz, Sohrabi, Srinivas AAAI 2025

Performance



GPT–OSS-120B

Generation Process



https://ibm.github.io/ACPBench/

Outline

- Core Reasoning Tasks for Reliable Planning
- ACPBench Dataset
- **Evaluation with LM-Eval Harness**
- Planning Benchmark Desiderata
- Countdown domain

Evaluation

LM Eval Harness

1. Select your model and provider

```
export WATSONX_PROJECT_ID=
export WATSONX_API_KEY=
export WATSONX_URL=
```

2. Install LM evaluation harness in your python env

```
conda create -n lmeval python=3.12
git clone --depth 1 https://github.com/EleutherAI/lm-evaluation-harness
cd lm-evaluation-harness
pip install .[ibm_watsonx_ai,acpbench]
```

3. Run evaluation

```
lm_eval --model watsonx_llm --model_args  model_id=openai/gpt-oss-120b
--tasks acp_bench --limit 2 --output ./temp --log_samples
```

https://github.com/IBM/ACPBench/blob/main/GettingStarted.md

Outline

- Core Reasoning Tasks for Reliable Planning
- ACPBench Dataset
- Evaluation with LM-Eval Harness
- **Planning Benchmark Desiderata**
- Countdown domain

Planning Benchmark Desiderata

In the era of LMs

- It should have a precise yet concise natural language description, including initial state, goal, and task dynamics.
- The problem should be sequential in nature, the order in which the actions need to be performed should matter.
- It should have a well defined action and state space.
- The problem should be of a non-trivial complexity.
- Must have sound validators for candidate solutions.
- It should have a large instance space and a dynamic generation procedure, thus allowing for the avoidance of memorization concerns.

Outline

- Core Reasoning Tasks for Reliable Planning
- ACPBench Dataset
- Evaluation with LM-Eval Harness
- Planning Benchmark Desiderata
- **Countdown domain**

Countdown

Input: {1, 1, 4, 8, 8}
(multiset of n numbers)

Target: 17

Answer:
<START_SEQUENCE>
8 / 4
2 * 1
2 * 8
16 + 1
<END_SEQUENCE>

Definition 1 *A* **Countdown** *problem is defined by a tuple of the form* $\mathcal{C} = \langle I_1, O, \tau \rangle$*, where* input I_1 *is a multi-set of* n *non-negative integers, i.e,* $\forall x \in I_1,\ x \in \mathbb{N}$, operators O *is the set of arithmetic operators and* target τ *is a non-negative integer* $\tau \in \mathbb{N}$. *The solution to a countdown problem consists of a sequence of triplets of the form* $\Theta = \langle \langle x_1, o_1, y_1 \rangle, \ldots, \langle x_{n-1}, o_{n-1}, y_{n-1} \rangle \rangle$*, such that*

(i) *for* $1 \leq i < n,\ o_i \in O$,
(ii) *for* $1 \leq i < n,\ \{x_i, y_i\} \subseteq I_i$ *and* $I_{i+1} = I_i \setminus \{x_i, y_i\} \cup \{o_i(x_i, y_i)\}$, *and*
(iii) $I_n = \{\tau\}$.

NP Complete

Katz, Sreedharan, Kokel 2025

Countdown

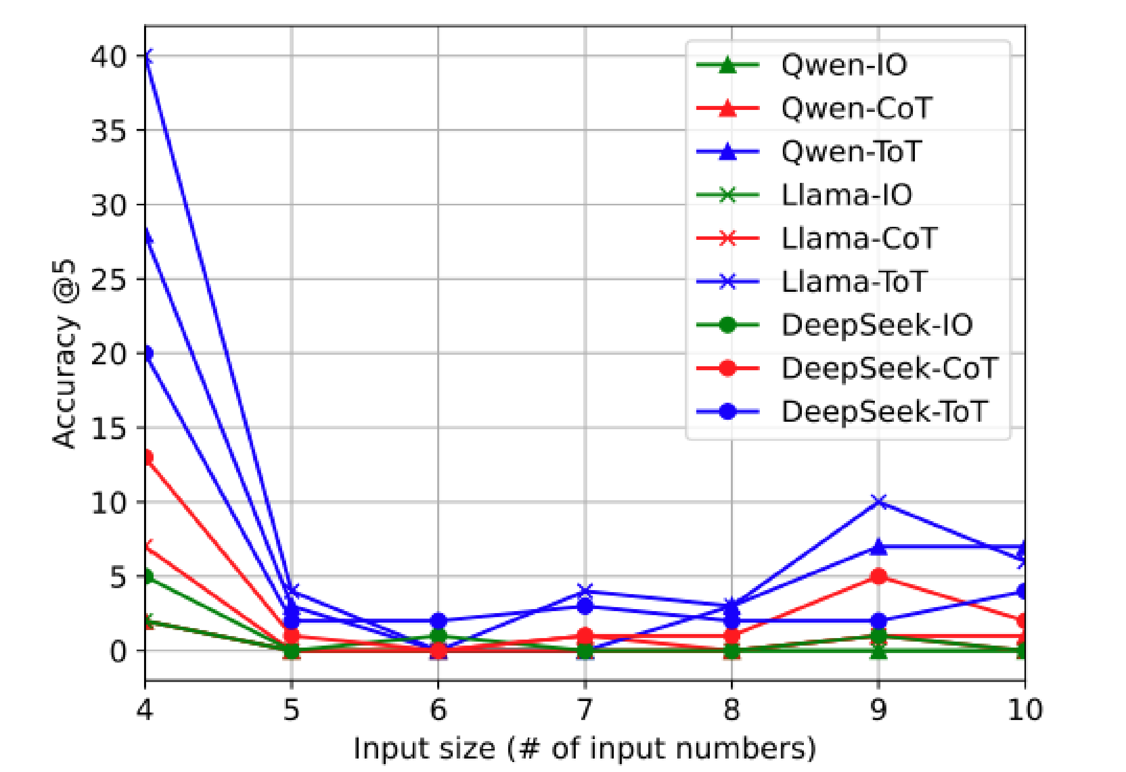


Figure 5: Accuracy @5 of LLM planning methods on CD.

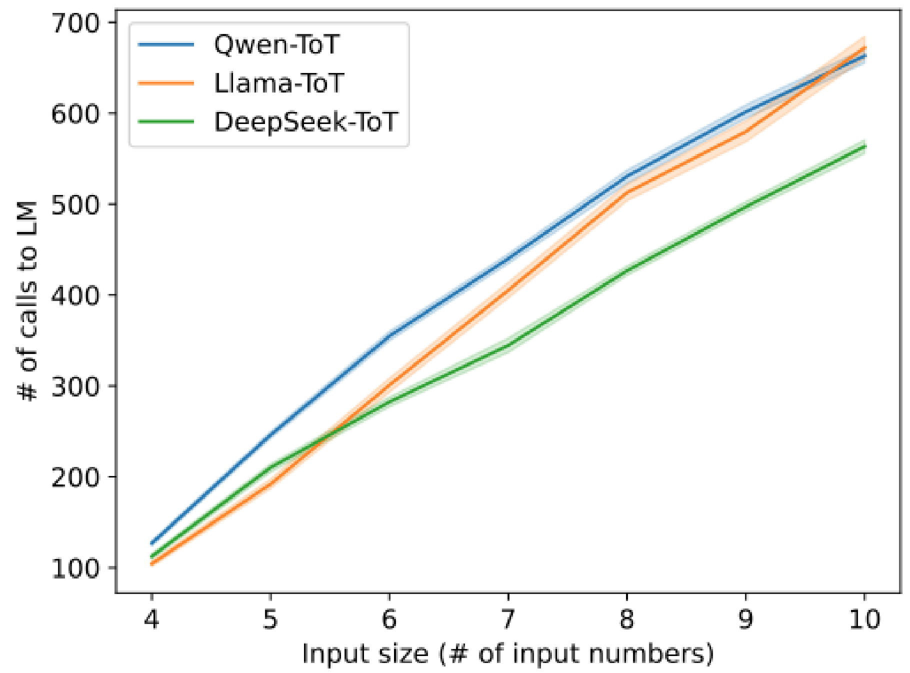


Figure 6: The average number of calls made to language models by the ToT approach with various language models.

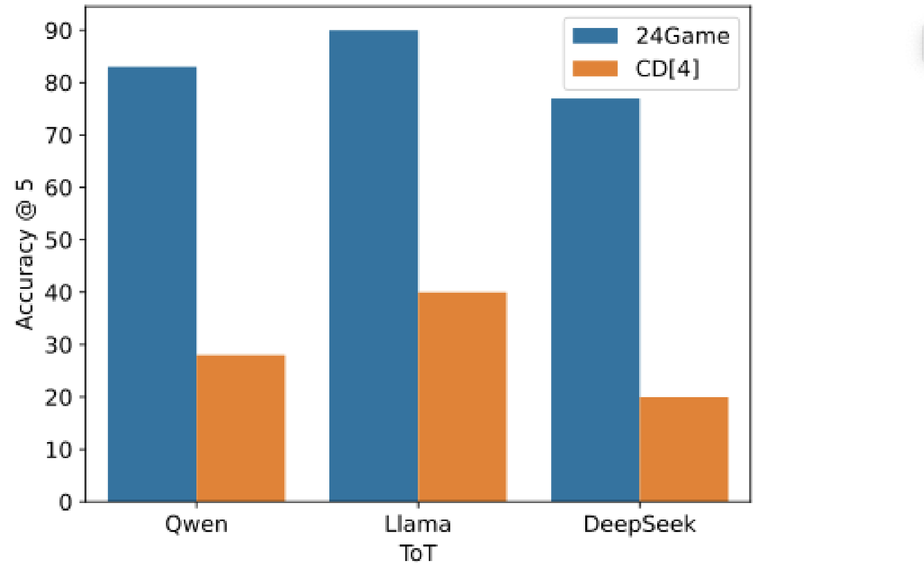


Figure 7: Accuracy @5 of various language models using the Tree of Thought (ToT) approach, comparing the 24Game dataset to instances of the same size (4) from our dataset.

Game of 24 instances :
https://www.4nums.com/game/difficulties/

Countdown

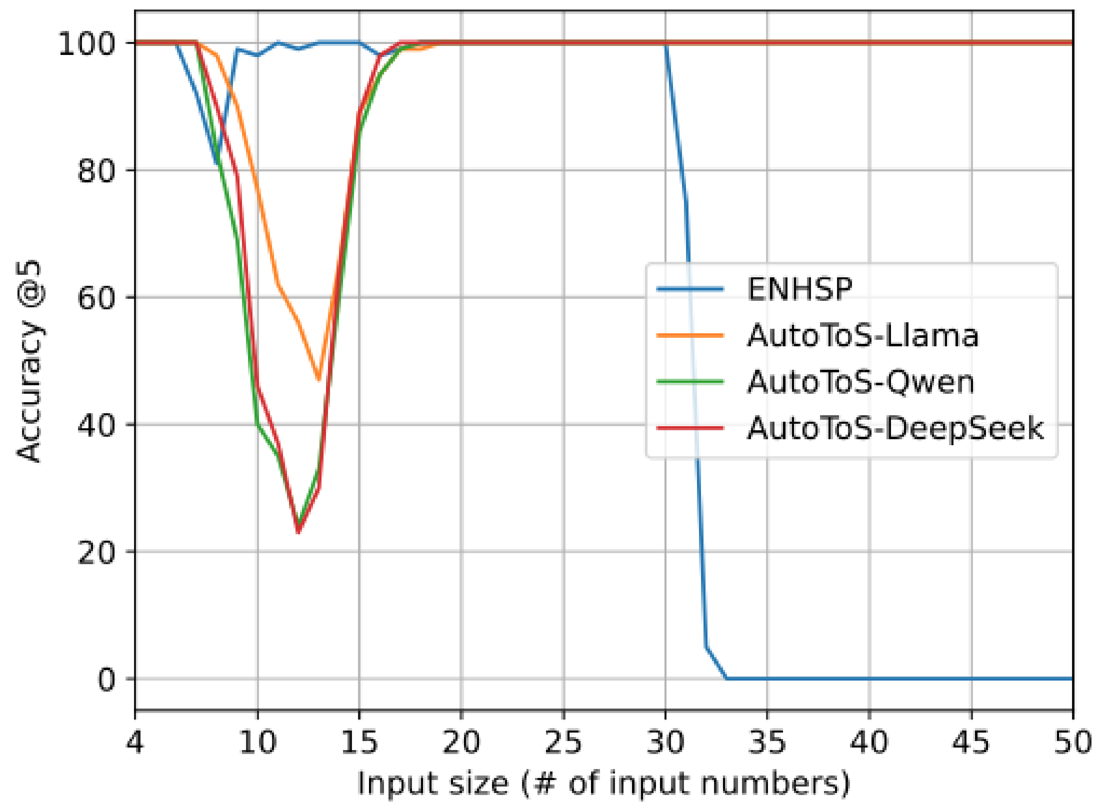


Figure 3: The accuracy of ENHSP and accuracy@5 of AutoToS with different language models for the Countdown problem.

References

ACPBench: Reasoning about Action, Change, and Planning, Harsha Kokel, Michael Katz, Kavitha Srinivas, Shirin Sohrabi, In AAAI 2025.

ACPBench Hard: Unrestrained Reasoning about Action, Change, and Planning, Harsha Kokel, Michael Katz, Kavitha Srinivas, Shirin Sohrabi, In LM4Plan @ AAAI 2025.

Seemingly Simple Planning Problems are Computationally Challenging: The Countdown Game, Michael Katz, Harsha Kokel, Sarath Sreedharan, In LM4Plan @ ICAPS 2025

Thought of Search: Planning with Language Models Through The Lens of Efficiency, Michael Katz, Harsha Kokel, Kavitha Srinivas, Shirin Sohrabi, In NeurIPS 2024.

Automating Thought of Search: A Journey Towards Soundness and Completeness, Daniel Cao, Michael Katz, Harsha Kokel, Kavitha Srinivas, Shirin Sohrabi, In OWA @ NeurIPS 2024.

Make Planning Research Rigorous Again!, Michael Katz, Harsha Kokel, Christian Muise, Shirin Sohrabi, Sarath Sreedharan, In ArXiv 2025.

https://planning-llm-era.github.

Questions?

https://planning-llm-era.github.io

Links and references

Tutorial Webpage

ICAPS 2026 Summer School
June 22-25, 2026

ICAPS conference

LM4Plan WS series

PLAN-FM
Bridge @ AAAI 2026

AI Planning Community Git

Back up

Dataset	PlanBench	AutoPlanBench	TRAC	ARB	ACPBench	ACPBench Hard
# Tasks	8	1	4	6	7	8
# Domains	3 (+variants)	13	1	8	13	13
NL templates	✓	×	✓	✓	✓	✓
Evaluation	🛠	🛠	↔	↔, `LLM`	↔	🛠
Question Format						
Generative	✓	✓	×	✓	×	✓
Boolean	×	×	✓	✓	✓	×
MCQ	×	×	×	×	✓	×
Tasks						
Applicability	×	×	✓	✓	✓	✓
Progression	✓	×	✓	✓	✓	✓
Reachability	×	×	×	×	✓	✓
Action Reachability	×	×	×	×	✓	✓
Validation	✓	×	✓	~	✓	✓
Justification	×	×	×	×	✓	✓
Landmark	×	×	×	×	✓	✓
Next Action	×	×	×	×	×	✓

Table 4: Comparison of ACPBench-hard with existing Planning Benchmarks. Evaluations are either using string matching (↔), symbolic tools (🛠), or using another LLM (`LLM`).

Valmeekam et al. NeurIPS 23, Stein et al. 23, He et al. ACL 23, Handa et al. 2025

Model	Applicability		Progression		Reachability		Validation		Action Reach.		Justification		Landmark		Mean	
	Bool	MCQ	Bool	MCQ	Bool	MCQ	Bool	MCQ	Bool	MCQ	Bool	MCQ	Bool	MCQ	Bool	MCQ
Phi-3 128K	66.15	33.08	68.46	53.85	52.31	26.15	50.77	19.23	53.33	32.50	49.23	33.85	49.23	46.92	55.53	34.75
Gemma 7B	63.23	28.62	64.92	31.08	53.08	23.08	46.92	20.0	55.67	34.50	50.77	36.46	27.54	30.31	51.80	28.93
Mistral 7B	61.54	32.31	73.08	38.46	53.08	28.46	47.85	17.69	**65.00**	19.17	48.46	30.00	35.38	33.08	55.00	28.67
Mistral I. 7B	63.08	31.54	61.54	46.92	61.54	33.08	52.15	36.15	45.83	34.17	43.08	29.23	57.69	50.77	55.45	37.30
Granite C. 8B	59.23	32.31	70.00	34.31	52.31	24.31	44.15	17.08	57.50	25.83	46.92	34.62	37.23	35.38	53.09	29.21
Granite 3.0 8B	72.31	26.92	73.08	53.85	53.08	24.62	53.08	20.00	45.83	30.83	49.23	34.62	42.31	34.62	55.56	32.21
Granite 3.0 I. 8B	76.92	30.00	73.85	57.69	53.08	36.92	55.38	34.62	58.33	44.17	70.77	31.54	51.54	43.08	62.84	39.72
LLAMA-3 8B	72.92	49.23	73.08	56.00	55.23	41.08	51.54	49.23	63.50	36.67	57.54	32.31	56.92	43.85	61.53	44.05
LLAMA-3.1 8B	65.38	56.92	63.85	47.69	53.08	33.85	60.00	37.69	42.50	28.33	46.92	45.38	33.85	40.00	51.46	41.52
Mixtral 8x7B	75.85	57.69	74.00	61.38	76.00	40.00	65.69	34.77	52.83	55.00	55.38	51.38	59.54	60.00	65.53	51.44
Codestral 22B	84.62	39.23	83.85	51.54	54.62	28.46	66.15	24.62	53.33	38.33	67.69	62.31	59.23	42.31	67.40	40.97
Mixtral 8x22B	80.77	37.69	72.31	54.62	50.00	42.62	37.69	16.92	58.50	27.83	43.08	44.62	44.77	45.23	55.63	39.25
Deepseek I. 33B	70.77	37.23	68.46	46.31	53.08	31.69	51.54	37.69	50.00	27.50	46.92	26.15	62.31	39.23	57.58	35.11
LLAMA C. 34B	80.77	42.31	73.08	43.85	53.08	25.69	50.15	28.46	53.17	33.33	55.38	35.38	46.92	40.62	59.02	35.71
LLAMA-2 70B	78.46	24.62	71.54	36.77	53.08	26.92	51.38	16.15	60.83	22.00	49.23	55.54	24.46	26.00	55.72	29.71
LLAMA C. 70B	74.77	36.15	54.77	52.92	48.62	23.69	40.0	17.69	49.67	28.83	46.92	31.54	37.08	42.31	50.90	32.87
LLAMA-3 70B	90.77	82.31	93.08	86.15	**87.69**	**82.31**	**78.62**	56.62	60.50	63.00	62.31	85.38	78.15	64.77	78.71	74.30
LLAMA-3.1 70B	93.08	84.31	89.85	86.77	61.38	54.92	66.15	46.62	63.00	58.00	56.92	68.46	34.62	69.23	66.67	66.94
LLAMA-3.1 405B	95.38	86.92	93.08	**93.85**	59.23	80.77	77.23	**62.92**	**65.00**	**65.00**	**90.00**	**86.92**	83.08	65.38	80.49	**77.42**
GPT-4o Mini	90.77	73.85	**95.38**	79.23	80.77	39.23	67.69	46.15	54.17	21.67	77.69	70.00	76.92	67.69	77.74	56.50
GPT-4o	**96.92**	**89.23**	94.62	90.00	79.23	76.92	61.54	53.85	57.50	52.50	88.46	80.77	**95.38**	**79.23**	**81.84**	74.97

Table 2: Accuracy of 21 LLMs, (I)nstruct and (C)ode models, on 7 ACPBench tasks (boolean and multi-choice). The best results are **boldfaced**, second best are *underlined*, and the best among the small, open-sourced models are *double underlined*. All models were evaluated with two in-context examples and COT prompt. The right-most column is mean across tasks.