The 2025 Planning Performance of Frontier Large Language Models

Augusto B. Corrêa
University of Oxford
United Kingdom

André G. Pereira
Federal University of Rio Grande do Sul
Brazil

Jendrik Seipp
Linköping University
Sweden

Abstract

The capacity of Large Language Models (LLMs) for reasoning remains an active area of research, with the capabilities of frontier models continually advancing. We provide an updated evaluation of the end-to-end planning performance of three frontier LLMs as of 2025, where models are prompted to generate a plan from PDDL domain and task descriptions. We evaluate DeepSeek R1, Gemini 2.5 Pro, GPT-5 and as reference the planner LAMA on a subset of domains from the most recent Learning Track of the International Planning Competition. Our results show that on standard PDDL domains, the performance of GPT-5 in terms of solved tasks is competitive with LAMA. When the PDDL domains and tasks are obfuscated to test for pure reasoning, the performance of all LLMs degrades, though less severely than previously reported for other models. These results show substantial improvements over prior generations of LLMs, reducing the performance gap to planners on a challenging benchmark.

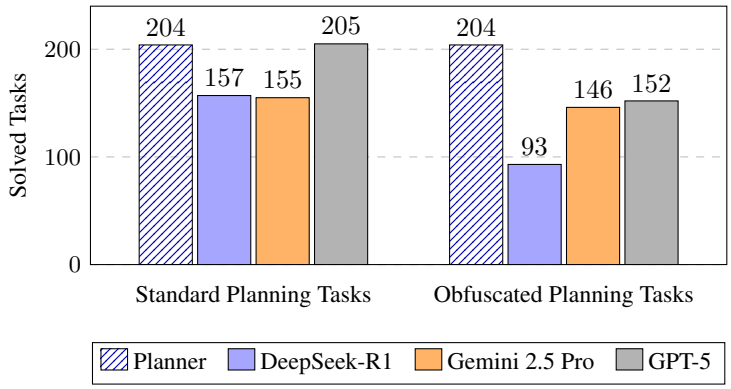


Figure 1: End-to-end planning performance of frontier LLMs and a planner (LAMA) on standard and obfuscated planning tasks from the IPC 2023 Learning Track. For each task, we prompt a model with a single call, providing the problem description in a formal language alongside two examples. We test the models on two versions of the benchmarks: the standard problems and an "obfuscated" version where all symbolic names are replaced with random strings [25, 24]. To reduce the risk of data contamination, we generated 360 fresh new tasks using the parameter distributions from the IPC 2023 test set. The generated plans are then validated using a sound tool. While GPT-5 matches the performance of LAMA in standard domains, its performance decreases when the tasks are obfuscated. Remarkably, Gemini 2.5 Pro is the only LLM that is not severely impacted by obfuscation.

Research note.

1 Introduction

Planning is the problem of finding a sequence of actions that transforms an initial state into a goal state. For Large Language Models (LLMs), solving such problems reliably has been a challenge [25, 10]. LLMs designed for "reasoning" often fail to compete with planners while incurring much higher computational costs [3]. In this research note, we provide an evaluation of the end-to-end planning performance of three frontier LLMs [18, 5, 4] as of 2025, comparing them with a strong planner. Figure 1 summarizes our findings, showing the number of solved tasks. Our results indicate that frontier LLMs have made substantial progress in solving challenging planning tasks [cf. 25].

Planning is a good benchmark for evaluating the reasoning capabilities of LLMs. Among the reasons are that planning domains provide scalable and verifiable benchmarks. Domains are typically described in the Planning Domain Definition Language (PDDL) [6] and many have generators for creating tasks of varying difficulty, enabling scalable evaluation [23]. Furthermore, the formal semantics of PDDL allow for automatic verification of plans using tools like VAL [9].

Another advantage is the diversity of planning domains, each demanding different capabilities. Strong performance on these diverse planning domains may then indicate *general reasoning ability*. Decades of research in automated planning have produced a rich and diverse set of domains, many inspired by real-world applications. Examples include *Airport*, modeling ground control traffic [8]; *Caldera*, a cybersecurity application for automating adversary network attacks [17]; *Elevators*, controlling building elevators [12]; *Pipesworld*, managing oil derivative flow in pipeline networks [16]; *Satellite*, a NASA application for satellite image capture [13]; *Organic Synthesis*, for molecular synthesis [14]; and *Quantum-Layout*, synthesizing quantum circuit layouts [21].

While these and many other domains are available, a comprehensive evaluation is prohibitively expensive. Therefore, we focus on a subset of domains from Learning Track of the most recent International Planning Competition (IPC) [22], as they provide a well-established benchmark for evaluating learning-based approaches and LLMs.

Recent work by Katz et al. [11] provides some guidelines for research on planning with LLMs. Our work follows these guidelines. First, all plans generated by LLMs are verified by the sound validation tool VAL [9]. Second, to mitigate data contamination, we generate a novel set of tasks. Finally, we compare the LLMs to a strong baseline from the planning literature [19]. We do not intend to claim that LLMs can replace planners, but rather to offer an updated perspective on their evolving capabilities in end-to-end planning.

2 Background

This paper addresses *classical planning*: problems with a single agent, deterministic actions, and a fully-observable, discrete environment. These tasks are formally described using the Planning Domain Definition Language (PDDL) [15, 6]. We provide an informal overview of a PDDL fragment,[1] using a simple Logistics domain for illustration.

A PDDL task is defined by the following components: a set of *objects* — e.g., vehicles, packages, and locations; a set of *predicates* representing relations between these objects — e.g., the binary relation *in*$(\mathrm{Truck1}, \mathrm{London})$ indicates that $\mathrm{Truck1}$ is in London; a set of *actions* that modify these relations in the state where they are *applied* — e.g. driving a truck changes its location; an *initial state* representing the initial relations of our problem — e.g., the truck starts in Stockholm; a *goal*, indicating which relations must hold in the goal state — e.g., deliverying a package in London. PDDL specifications are typically organized into two files: a *domain* file, which defines the predicates and actions; and a *task* file, which specifies the objects, initial state, and goal.

The objective of a *planner* is to find a *plan*: a sequence of actions transforming the initial state into a state where the goal is satisfied. In this work, we consider *satisficing planning*, where any valid plan is a solution and optimality is not required. The main evaluation metric for a planner is its *coverage* (i.e., number of solved tasks) but we are also interested in *plan quality* (how long is the plan found).

[1] For a complete explanation of the semantics of the PDDL fragment usually used in competitions, see [7, Section 2]; for an explanation of the semantics of the simpler STRIPS fragment — the one most used by the papers cited here — see [2, Section 2]

LLM-based planning is neither *sound* (guaranteed to only return valid plans) nor *complete* (guaranteed to find a plan if one exists). To address the soundness issue, we use the VAL tool to validate all generated plans, making the overall algorithm technically sound.

3 Experiments

We evaluate the end-to-end planning capabilities of frontier LLMs using a few-shot prompting strategy. We use the same prompt format as Corrêa et al. [3], containing general instructions, the PDDL domain and task files, a checklist of common pitfalls, and two illustrative examples from the Gripper and Logistics domains, complete with their plans. All plans are validated using VAL [9].

To test whether the LLMs are reasoning over the PDDL description, we also create an obfuscated version of each domain and task, following the methodology of Valmeekam et al. [25]. In this setup, all symbols (actions, predicates, objects) in the PDDL files are replaced with random strings. This obfuscation is highly adversarial for LLMs, which uses token semantics, but it does not affect symbolic planners. Specifically, we use the obfuscation scheme by Chen et al. [1]. To distinguish between these two sets, we refer to the original problems as the *standard* tasks/domains, while the obfuscated versions are simply called the *obfuscated* tasks/domains.

As a baseline planner, we use the LAMA [19]. More precisely, we use only its first search iteration, sometimes referred to as "LAMA-first". As a sound planner, it only produces correct plans. All experiments with LAMA were run with a 30-minute time limit and a 8 GiB memory limit per task. In our experiments, we use Downward Lab [20] on AMD EPYC 7742 processors running at 2.25 GHz.

We tested the following LLMs: DeepSeek R1 [4], Gemini 2.5 Pro [5], and GPT-5 [18]. All our tests used the official APIs with default parameters and no tools allowed. The experimental costs were approximately $9.13 for DeepSeek R1 and $100.47 for GPT-5; the Gemini 2.5 Pro experiments used the free tier of their API. GPT-5 returned an answer in less than 30 minutes for all tasks, while Gemini 2.5 Pro always answered the prompt in less than 10 minutes. DeepSeek R1, on the other hand, took more than 30 minutes in a handful of obfuscated tasks. In all these cases, the answer provided by DeepSeek R1 was wrong.

We use eight domains from the International Planning Competition (IPC) 2023 Learning Track [22]. To mitigate data contamination, we generated a novel set of tasks using the parameter distributions from the IPC test set. While we cannot guarantee that these tasks are not available online, this procedure makes it unlikely that the LLMs have encountered them during training. These tasks are considerably more challenging than those typically used when benchmarking planning skills of LLMs [25]. Table 1 shows the parameter distribution of our set, and the longest plans found by the baseline planner and GPT-5. This parameter distribution is similar to the one used in the IPC 2023.

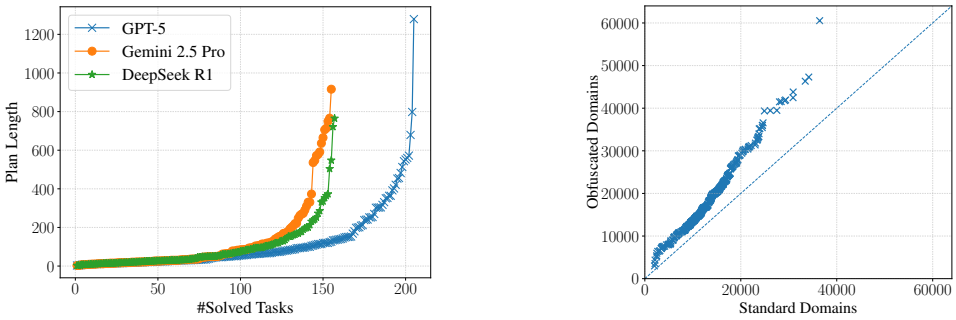
Table 1: Task size for each domain based on their main parameters: n blocks in Blocksworld, c children in Childsnack, t tiles in Floortile, p passengers in Miconic, r rovers in Rovers, b boxes in Sokoban, s spanners in Spanner, and v vehicles in Transport. We also show the longest (and possibly suboptimal) plan found by the baseline planner LAMA.

Domain	**Parameters**	**Max. Length**
Blocksworld	$n \in [5, 477]$	1194
Childsnack	$c \in [4, 284]$	252
Floortile	$t \in [12, 899]$	62
Miconic	$p \in [1, 470]$	1438
Rovers	$r \in [1, 29]$	1194
Sokoban	$b \in [1, 78]$	860
Spanner	$s \in [1, 474]$	21
Transport	$v \in [3, 49]$	212

Table 2 details the coverage for each method across both standard and obfuscated tasks. On the standard tasks, GPT-5 shows strong performance, solving 205 out of 360 tasks and being on par with LAMA (204). DeepSeek R1 and Gemini 2.5 Pro also show notable capabilities, solving 157 and 155 tasks, respectively. LLMs excel in specific domains: for example, in both Childsnack and Spanner, all three LLMs outperform LAMA. In the Spanner domain, GPT-5 solves all 45 tasks. GPT-5 also solves three tasks in the Floortile domain that could not be solved by the baseline, while Gemini 2.5 Pro solves one task in Blocksworld, one in Floortile and one in Transport that were not solved by LAMA. However, in most of the domains LAMA solves more tasks that any LLM.

Table 2: End-to-end planning performance of frontier large language models and a classical planner (LAMA) on standard and obfuscated planning domains from IPC 2023 Learning Track. The best performance for each setting is highlighted in bold.

	Standard Planning Tasks				Obfuscated Planning Tasks			
	Planner	LLMs			Planner	LLMs		
	LAMA	R1	Gemini	GPT-5	LAMA	R1	Gemini	GPT-5
Blocksworld (45)	**28**	19	21	21	**28**	3	**28**	12
Childsnack (45)	18	33	**38**	**38**	18	28	25	**33**
Floortile (45)	**10**	5	6	**10**	**10**	0	5	3
Miconic (45)	**45**	27	23	35	**45**	14	9	28
Rovers (45)	**34**	9	6	14	**34**	5	8	10
Sokoban (45)	**21**	10	9	15	**21**	0	5	7
Spanner (45)	15	38	30	**45**	15	34	**39**	38
Transport (45)	**33**	16	22	27	**33**	9	27	21
Sum (360)	204	157	155	**205**	**204**	93	146	152



(a) Plan lengths for solved tasks.

(b) Reasoning tokens used by Gemini 2.5 Pro.

Figure 2: A comparison of plan lengths and reasoning effort. (a) The distribution of plan lengths for tasks solved by each LLM, sorted independently. (b) The reasoning tokens used by Gemini 2.5 Pro to solve tasks in the standard vs. obfuscated settings.

The situation changes with the obfuscated tasks, which are designed to test pure symbolic reasoning by removing semantic clues from predicate and action names. The performance of all LLMs drops, while performance of LAMA remains unchanged as it is invariant to such renaming. GPT-5 solves 152 tasks, still the best-performing LLM. Gemini 2.5 Pro solves 146 tasks, a comparatively small drop that suggests a stronger robustness to this type of modification. The performance of DeepSeek R1 sees the most significant drop, solving only 93 tasks. This performance degradation across all LLMs indicates that they still depend on the semantic information in the names of predicates and actions. However, the fact that LLMs solve a non-trivial number of these hard, purely symbolic tasks is an interesting result and an improvement over prior model generations.

Figures 2a and 2b provide more details on the performance of the LLMs. Figure 2a shows the distribution of plan lengths for solved tasks. LLMs are capable of generating very long plans, including several plans with more than 500 steps. For a plan to be valid, every action must be correct in sequence. The ability to maintain such a long sequence of correct reasoning steps suggests an improved reliability in these models. Figure 2b shows the reasoning tokens used by Gemini 2.5 Pro for solved tasks in the standard and obfuscated settings. While performance of Gemini in terms of solved tasks is only slightly lower in the obfuscated setting, the number of reasoning tokens increases substantially. This suggests that greater computational effort is required to compensate for the lack of semantic information.

It is important to consider the computational resources required for these experiments. LAMA was run on AMD EPYC 7742 processors using a single core with 8 GiB of memory, requiring minimal

memory and CPU resources. In contrast, the LLMs demand vastly more resources. While the exact hardware for the proprietary GPT-5 and Gemini 2.5 Pro models is not public, it is understood to be massive-scale GPU clusters. For DeepSeek R1, a 671B parameter Mixture-of-Experts model (with 37B active parameters), the hardware requirements are more concrete: it is estimated to require over 1 000 GiB of GPU memory for inference. This vast difference in hardware, energy consumption, and cost highlights a critical trade-off: while LLMs might become more capable at planning, they are orders of magnitude less efficient than specialized planners, a crucial consideration for their practical application.

4 Discussion

In this work, we evaluated the 2025 generation of frontier LLMs on a challenging set of automated planning domains. Our results indicate progress: on standard PDDL tasks, the performance of GPT-5 is now competitive with LAMA, a strong classical planner. Moreover, while performance on obfuscated tasks confirms that LLMs still rely on token semantics, the degradation is less severe than for prior models [25, 26], hinting at improved symbolic reasoning capabilities. These models can also produce remarkably long valid plans. However, this progress must be weighted against the computational costs of LLMs, which remain orders of magnitude less efficient than specialized planners.

References

[1] Dillon Z. Chen, Johannes Zenn, Tristan Cinquin, and Sheila A. McIlraith. Language models for generalised PDDL planning: Synthesising sound and programmatic policies. In *ICAPS 2025 Workshop on Planning in the Era of LLMs (LM4Plan)*, 2025.

[2] Augusto B. Corrêa and Giuseppe De Giacomo. Lifted planning: Recent advances in planning using first-order representations. In Kate Larson, editor, *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*, pages 8010–8019. IJCAI, 2024.

[3] Augusto B. Corrêa, André G. Pereira, and Jendrik Seipp. Classical planning with LLM-generated heuristics: Challenging the state of the art with Python code. In *Proceedings of the Thirty-Ninth Annual Conference on Neural Information Processing Systems (NeurIPS 2025)*, 2025.

[4] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z.F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, and Guanting Chen et al. Deepseek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv:2501.12948 [cs.CL], 2025.

[5] Gemini Team Google. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv:2507.06261 [cs.CL], 2025.

[6] Patrik Haslum, Nir Lipovetzky, Daniele Magazzeni, and Christian Muise. *An Introduction to the Planning Domain Definition Language*, volume 13 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool, 2019.

[7] Malte Helmert. Concise finite-domain representations for PDDL planning tasks. *Artificial Intelligence*, 173:503–535, 2009.

[8] Jörg Hoffmann, Stefan Edelkamp, Sylvie Thiébaux, Roman Englert, Frederico dos Santos Liporace, and Sebastian Trüg. Engineering benchmarks for planning: the domains used in the deterministic part of IPC-4. *Journal of Artificial Intelligence Research*, 26:453–541, 2006.

[9] Richard Howey and Derek Long. VAL's progress: The automatic validation tool for PDDL2.1 used in the International Planning Competition. In Stefan Edelkamp and Jörg Hoffmann, editors, *Proceedings of the ICAPS 2003 Workshop on the Competition: Impact, Organisation, Evaluation, Benchmarks*, 2003.

[10] Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. Position: LLMs can't plan, but can help planning in LLM-modulo frameworks. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[11] Michael Katz, Harsha Kokel, Christian Muise, Shirin Sohrabi, and Sarath Sreedharan. Make planning research rigorous again! arXiv:2505.21674 [cs.CL], 2025.

[12] Jana Koehler and Kilian Schuster. Elevator control as a planning problem. In Steve Chien, Subbarao Kambhampati, and Craig A. Knoblock, editors, *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling (AIPS 2000)*, pages 331–338. AAAI Press, 2000.

[13] Derek Long and Maria Fox. The 3rd International Planning Competition: Results and analysis. *Journal of Artificial Intelligence Research*, 20:1–59, 2003.

[14] Rami Matloob and Mikhail Soutchanski. Exploring organic synthesis with state-of-the-art planning techniques. In *ICAPS 2016 Scheduling and Planning Applications woRKshop*, pages 52–61, 2016.

[15] Drew McDermott. The 1998 AI Planning Systems competition. *AI Magazine*, 21(2):35–55, 2000.

[16] Ruy L. Milidiu and Frederico dos Santos Liporace. Pipesworld: Applying planning systems to pipeline transportation. In *International Pipeline Conference*, 2004.

[17] Doug Miller, Ron Alford, Andy Applebaum, Henry Foster, Caleb Little, and Blake Strom. Automated adversary emulation: A case for planning and acting with unknowns. Technical report, MITRE, 2018.

[18] OpenAI. GPT-5 system card, 2025. URL `https://openai.com/index/gpt-5-system-card/`.

[19] Silvia Richter and Matthias Westphal. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39:127–177, 2010.

[20] Jendrik Seipp, Florian Pommerening, Silvan Sievers, and Malte Helmert. Downward Lab. `https://doi.org/10.5281/zenodo.790461`, 2017.

[21] Irfansha Shaik and Jaco van de Pol. Optimal layout synthesis for quantum circuits as classical planning. In *Proceedings of International Conference on Computer Aided Design (ICCAD)*, pages 1–9. IEEE/ACM, 2023.

[22] Ayal Taitler, Ron Alford, Joan Espasa, Gregor Behnke, Daniel Fišer, Michael Gimelfarb, Florian Pommerening, Scott Sanner, Enrico Scala, Dominik Schreiber, Javier Segovia-Aguas, and Jendrik Seipp. The 2023 International Planning Competition. *AI Magazine*, 45(2):280–296, 2024. doi: 10.1002/aaai.12169.

[23] Álvaro Torralba, Jendrik Seipp, and Silvan Sievers. Automatic instance generation for classical planning. In Robert P. Goldman, Susanne Biundo, and Michael Katz, editors, *Proceedings of the Thirty-First International Conference on Automated Planning and Scheduling (ICAPS 2021)*, pages 376–384. AAAI Press, 2021.

[24] Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Proceedings of the Thirty-Seventh Annual Conference on Neural Information Processing Systems (NeurIPS 2023)*, pages 38975–38987, 2023.

[25] Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models - A critical investigation. In *Proceedings of the Thirty-Seventh Annual Conference on Neural Information Processing Systems (NeurIPS 2023)*, pages 75993–76005, 2023.

[26] Karthik Valmeekam, Kaya Stechly, Atharva Gundawar, and Subbarao Kambhampati. Planning in strawberry fields: Evaluating and improving the planning and scheduling capabilities of LRM o1. arXiv:2410.02162 [cs.CL], 2024.