LLM-WikiRace: A Benchmark for Planning and Reasoning over Real-World Knowledge Graphs

Anonymous Authors[1]

Abstract

We introduce **LLM-Wikirace**, a benchmark for evaluating planning, reasoning, and world knowledge in large language models (LLMs). In LLM-Wikirace, models must efficiently navigate Wikipedia hyperlinks step by step to reach a target page from a given source, requiring look-ahead planning and the ability to reason about how concepts are connected in the real world. We evaluate a broad set of open- and closed-source models, including Gemini-3.1, GPT-5, and Claude Opus 4.6, which achieve the strongest results on the easy split of LLM-WikiRace. Performance drops sharply on hard difficulty: the best-performing model, Gemini-3.1, succeeds in only 29% of hard games, highlighting substantial remaining challenges for frontier models. Our analysis shows that world knowledge is a necessary ingredient for success, but only up to a point, beyond this threshold, planning and long-horizon reasoning capabilities become the dominant factors. Trajectory-level analysis further reveals that even the strongest models struggle to replan after failure, frequently entering loops rather than recovering. LLM-Wikirace is a simple benchmark that reveals clear limitations in current reasoning systems, offering an open arena where planning-capable LLMs still have much to prove. Our code and data are publicly available. [1]

1. Introduction

Large Language Models have shown strong performance on a range of complex planning and reasoning tasks (Cobbe et al., 2021; Hendrycks et al., 2021c). While such successes are typically attributed to post-training approaches (Yu et al., 2025; Guo et al., 2025), these could not succeed without the model's ability to leverage a large breadth of knowledge imparted through pre-training on diverse textual corpora (Gao et al., 2020), including encyclopedic sources such as Wikipedia.

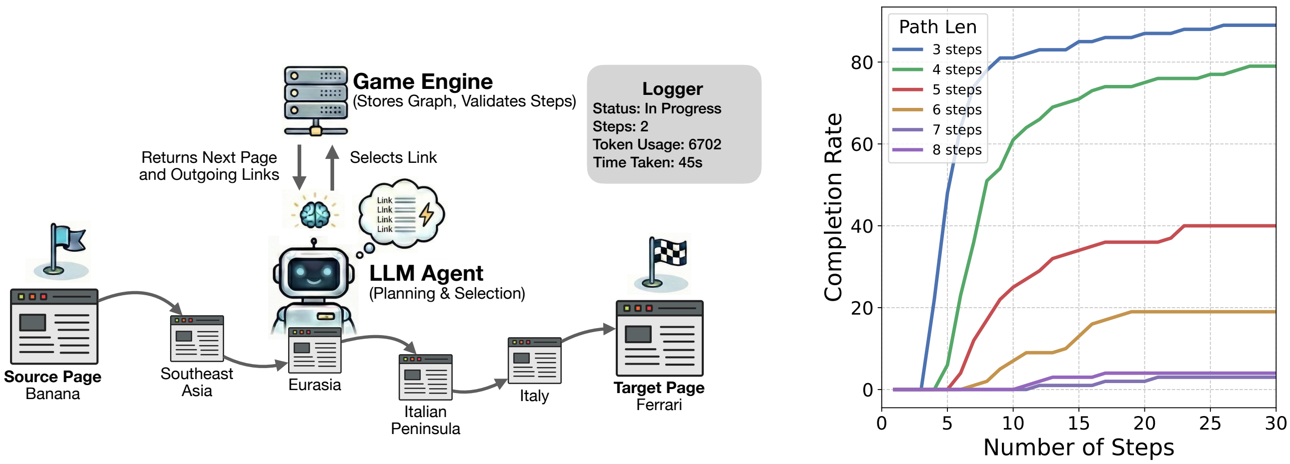
Many existing benchmarks fail to test how well models can leverage this breadth of knowledge focusing instead on narrow domains such as mathematical reasoning (Cobbe et al., 2021; Hendrycks et al., 2021c), or specific real-world tasks such as trip planning (Xie et al., 2024; Zheng et al., 2024), and website navigation (Yoran et al., 2024; Zhou et al., 2023). Benchmarks that do test for breadth of knowledge (Yang et al., 2018; Hendrycks et al., 2021a;b) do not require the model to *actively deploy* world knowledge to make decisions or plan. Reasoning and world knowledge have been studied in isolation, yet many real-world tasks require their tight integration. For tasks such as open-domain problem solving and autonomous research assistants the ability to recall and connect diverse pieces of knowledge is essential. In such settings, failures often arise not from a lack of reasoning ability alone but from an inability to retrieve or appropriately apply relevant world knowledge during planning. This leaves a key question: *to what extent can LLMs leverage pretrained world knowledge to reason and plan in environments which require a large breadth of knowledge?*

To address this we introduce LLM-WikiRace, the first benchmark that requires *interactive, multi-step planning* over the full Wikipedia hyperlink graph, asking the LLM to reason over extensive human knowledge. Our benchmark is based on the popular WikiRace game, where given a source and a target Wikipedia page, the player must traverse the Wikipedia hyperlink graph step by step, selecting links until it reaches the target or exhausts a fixed step budget, see Figure 1a. Success requires a model to retrieve world knowledge encoded in its weights and deploy it during multi-step planning — at each step, judging which outgoing link is most likely to advance toward the target. LLM-WikiRace is also useful for practitioners: by stripping away the domain-specific complications, tool-use quirks, and prompt sensitivities that dominate real-world agentic evaluations, it exposes

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the FoGen Workshop at ICML 2026. Do not distribute.

[1]`https://anonymous.4open.science/r/LLM-wikirace-15B9/`



(a) LLM-WikiRace setup

(b) Completion rate

Figure 1. (a) In LLM-WikiRace, the agent navigates the Wikipedia hyperlink graph from a source page (e.g., *Banana*) to a target page (e.g., *Ferrari*). At each step, it receives the current page, the titles of its outgoing links, the target, and the history of pages visited, then selects one link to follow. (b) The completion rate — the fraction of games solved — plotted against the step budget across different shortest source-to-target path lengths for LLAMA-3-70B-INSTRUCT, this experiment highlights how the number of steps per game and source to target shortest path length affect model performance.

flaws in a model's long-horizon planning ability. LLM-Wikirace is designed with cost and ease of use in mind, the benchmark presents two easy-to-use modes, one for models served by API and another for open-source models run locally. Our implementation runs multiple games in parallel cutting the benchmark run time in half, evaluating LLAMA-3-70B-INSTRUCT in parallel on one split of the benchmark takes 3 hours compared to a synchronous implementation which takes 6 hours. All our code is open-source [2]. Our key contributions are as follows:

- **LLM-WikiRace challenges frontier models on the hardest difficulty.** State of the art frontier models — Gemini 3, GPT-5, Claude Opus 4.5 — succeed on fewer than 30% of hard instances, leaving plenty of room for improvement. We provide a comprehensive study of over 20 models ranging from small open-source models (e.g. GEMMA-3-4B) to large frontier models.
- **Models plan well, but replan badly.** Trajectory-level analysis of model planning behaviours show that, surprisingly, despite adopting sensible high-level strategies, models fail in relatively naive ways - crucially they struggle to replan after failure becoming trapped in loops.
- **A measurable planning gap between reasoning and non-reasoning models** Among open-source instruct-tuned models, success on LLM-WikiRace is well-predicted by a static world-knowledge probe. Reasoning models break this relationship: at similar knowledge levels, success rates differ by up to 20 percentage points. We label this the planning gap, which we hypothesize reflects the contribution of reasoning post-training to a model's planning ability.
- **Post-training has limits.** Finally, we show that post-training open-source models can improve their reasoning capabilities on the easy split of the benchmark by **$\sim$40%**, but as the difficulty increases the gains from post-training decay and have no effect on the benchmark's hardest difficulty.

[2] `https://anonymous.4open.science/r/LLM-wikirace-15B9/`

2. What Type of Task Is WikiRace?

WikiRace is a task defined over the Wikipedia hyperlink graph in which an agent must navigate from a *start page* A to a *target page* B by selecting hyperlinks step by step. At each step, the agent observes the current page and its outgoing links, chooses a single link to follow, and transitions to a new page. The episode terminates when the target is reached or when a fixed step budget is exhausted. The objective is to reach the target in as few steps as possible. We illustrate this in Figure 1a.

The task poses a challenging planning problem over a large, real-world knowledge graph, requiring agents to combine planning and world knowledge in a partially observable, interactive setting. At inference time, the agent has no access to global graph information such as shortest paths, node distances, or future link availability, and must act based only on locally observed information while leveraging world knowledge acquired during pretraining to judge which pages may be relevant. Each step presents many locally plausible actions whose long-term consequences

WikiRace Game Prompt

System Prompt:
You are a helpful assistant helping play the Wikipedia link game.

User Prompt:
You are playing a game where you start at Wikipedia page "{current_page}" and want to reach page "{target_page}" by clicking links.

So far, you have visited the following pages in order:
{history}

You see the following possible links from the current page:

0. {Neighbor 0 Title}
1. {Neighbor 1 Title}
⋮
{N}. {Neighbor N Title}

Which link should you click to get closer to the target? Reply with the number of your choice (0 to {max_choice_num}).

Figure 2. The LLM-WikiRace prompt, at each step of the game an LLM is prompted with the current page, the target page, a history of previously visited states and 50 possible next states.

are uncertain, and early choices can strongly constrain future options. As a result, poor initial decisions may move the agent far from the target, making effective replanning—rather than merely good initial planning—essential for avoiding failure. In particular, LLM-WikiRace probes the following capabilities of a reasoning system:

Long-horizon planning. Reaching the target often requires selecting intermediate pages that are not directly related to the goal, testing the ability to plan coherent action sequences rather than rely on myopic heuristics.

Semantic reasoning over open-domain knowledge. Effective navigation depends on reasoning about relationships between concepts encoded in Wikipedia's hyperlinks.

Operational use of world knowledge. Rather than answering questions, agents must translate world knowledge into concrete action choices that guide navigation.

Replanning and adaptive control. When an initial strategy proves ineffective, successful agents must revise their approach, avoid repeated states, and explore alternative paths under a limited action budget.

2.1. What the Benchmark Is Not Measuring

LLM-WikiRace is not equivalent to shortest-path graph search. Agents are not given access to the global graph structure and each step consumes the limited step budget. As such, the model cannot just algorithmically perform standard graph search procedures (such as BFS).

3. How Does the LLM Play WikiRace?

We build the LLM-WikiRace hyperlink graph from a Wikipedia snapshot dated 23 June 2025 and retain only its largest strongly connected component, guaranteeing that every node is reachable from every other node. The resulting graph contains 549,232 pages. Fixing this snapshot makes results stable and directly comparable across models; preprocessing details are in Appendix A.

To adapt WikiRace for LLMs and explore the interaction between internal world knowledge and planning, we make several changes to the standard WikiRace game. Firstly, we provide the LLM with a structured prompt that specifies the current page, the target page, the traversal history, and the set of outgoing links available at each step; see Figure 2. Unlike the original game, we exclude the body of the current article from the LLM-WikiRace prompt. The agent has already reached this page and must choose from among outgoing links based on their names alone, this matches the information available to a player in the original game. Excluding the article body further requires the agent to ground its position in pre-trained world knowledge rather than in-context text. Successful navigation also demands planning several steps ahead, thus the agent must anticipate the contents of unvisited pages. Including the article body would also more than double prompt length, inflating cost, and would demand either a local Wikipedia snapshot of roughly 100 GB or a flood of calls against the rate-limited Wikipedia API. Appendix D reports an ablation on this design choice.

Secondly, we split WikiRace into three varying difficulty levels. We define an easy, medium, and hard difficulty using the shortest path length between the source and target pages as a measure of difficulty; we investigated this design decision on LLAMA-3-70B-INSTRUCT, as can be seen in Figure 1b, the number of successfully completed games (completion rate) monotonically decreases as the path length increases. The easy split comprises 200 pairs with shortest path lengths of 3 or 4, medium comprises 150 pairs with lengths of 5 or 6, and hard comprises 100 pairs with lengths of 7 or 8, with each split evenly balanced across its two path lengths.

Each game is capped at 30 steps. Our LLAMA-3-70B-INSTRUCT experiment in Figure 1b shows completion rate plateauing well before this limit across all path lengths, giving models more than three times the headroom of the longest shortest path. We also cap the link list shown to the model at 50; 71% of pages in our graph have fewer than 50 outgoing links. For the remaining pages we keep the 50 links with the shortest residual path length to the target and present them in random order; this shortest-path information lives only inside the environment and is never revealed to the model, so it serves to trim the prompt rather than guide the agent's reasoning. We report results with random link selection in Appendix B.

4. Evaluation

Models. We evaluate a diverse set of models on our benchmark, spanning both open- and closed-source systems across a wide range of scales and modalities. Our closed-source evaluations include the GPT-5 family (Singh et al., 2025), Gemini 2.0–3.1 (Comanici et al., 2025), Claude Sonnet 4.5 and Opus 4.5 & 4.6 (Anthropic, 2024), and Grok 4.1-Fast. For open-source models, we evaluate DeepSeek R1 (Gao et al., 2023a), Kimi K2 (Team et al., 2025), LLaMA 3 at scales ranging from 1B to 70B parameters (Grattafiori et al., 2024), Gemma 3 models from 4B to 27B parameters (Team et al., 2024), Apertus 8B and 70B (Hernández-Cano et al., 2025), Mistral 7B (Jiang et al., 2023) and Ministral 8B (Mistral AI, 2024), and Qwen 2.5-7B (Yang et al., 2025a). We additionally include two diffusion-based language models, Dream-v0-Instruct 7B (Ye et al., 2025) and LLaDA-Instruct 8B (Nie et al., 2025). For all open source models we test the Instruct finetuned version.

Metrics. We report several complementary metrics to characterise model performance. First, we measure the fraction of games successfully completed on each split, referred to as the *success rate*. Second, to assess solution quality among successful runs, we compute the number of *suboptimal steps*, defined as the excess number of steps taken beyond the shortest possible path. This metric captures how efficiently a model navigates once it succeeds, separating path quality from binary success. Finally, we report the average cost per game, computed over all evaluated episodes, including the total number of tokens generated and, for closed-source models, the corresponding monetary cost.

5. Results

We present full results in Table 1 and summarise the performance of the strongest models in Figure 3. Performance is clearly differentiated by difficulty: leading models achieve over 90% success on the easy split, approximately 50–70% on medium, and below 30% on hard. This confirms that the length of the shortest-path provides a meaningful stratification of task difficulty.

Among the evaluated models, Gemini 3 and 3.1 perform best at all difficulty levels and are the only model to exceed a success rate of 20% on the hard split. GPT-5 and Claude Opus 4.5/4.6 follow closely, with performance varying by difficulty. DeepSeek R1 is competitive with these models despite being open source, typically trailing by a small margin, while Kimi K2 ranks second among open source models with a more pronounced performance gap. Surprisingly, GPT-5.2 saw a notable decrease in performance on the medium and hard benchmarks in comparison to GPT-5.

Despite strong performance on the easy split, both the medium and hard difficulties remain largely unsolved. Even the best-performing model completes fewer than 30% of hard instances, indicating substantial headroom for improvement. Open source models from the LLaMA and Gemma families achieve near-zero success on the hard split, with the exception of LLAMA-3-70B-INSTRUCT, which reaches a 7% success rate. For these models, improving medium-difficulty performance—currently below 40% across the board—represents a more attainable near-term target.

Overall, results across model families suggest that all three difficulty levels remain relevant at present, with each split providing a distinct and informative benchmark for models at different scales.

5.1. Does Graph Knowledge Affect Model Performance?

To understand how the internal world knowledge of a model affects their performance on LLM-WikiRace we design a classification task to probe the model's understanding of the Wikipedia hyperlink graph. We construct a dataset consisting of $1k$ pairs of source and target nodes. Pairs are labeled as *connected* if there exists a direct hyperlink from the source node to the target node and as *unconnected* otherwise. For each pair in this dataset we ask the model to predict if the nodes are directly connected or not, details of the dataset construction and model prompt for this task can be found in Appendix E. We use the F1-Score of the model on this classification task as a metric for how well a model understands how pages are connected through the Wikipedia hyperlink graph.

In Figure 4 we plot the F1-Score against the average success rate of each model across all LLM-WikiRace splits for a range of instruct and reasoning models. We fit a linear regression model on the instruct models and plot the 95% confidence interval assuming independent gaussian noise. Among instruct-tuned models, the world knowledge probe is a reasonable predictor of success rate; however, reasoning models break this relationship. At similar levels of world knowledge, success rates differ by up to 20% for reasoning models, lying outside the 95% regression confidence interval. We label this phenomenon the Planning Gap, as reasoning models seem to be able to leverage their world knowledge more effectively than instruct tuned models. We hypothesize that this is due to their improved planning capabilities.

In Figure 4 one reasoning model, GPT-5-mini, is firmly

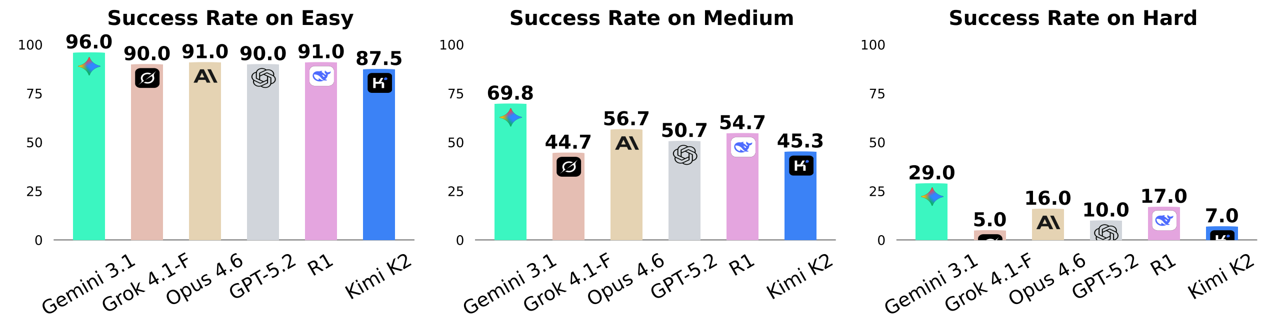


Figure 3. The Success Rates for the best performing models across all three difficulties of the LLM-WikiRace benchmark.

Table 1. Performance of different models across Easy, Medium, and Hard datasets. Success Rate is the percentage of games completed within the allowed 30 steps. Suboptimal Steps is the average number of suboptimal steps made in the successfully completed games. If no games were successfully completed for a given model and difficulty, we set Suboptimal Steps to "-". The best result is in **bold**, and the second best result is <u>underlined</u> for each difficulty level's success rate.

Family	Model	Success Rate (%) ↑			Suboptimal Steps ↓			Usage Per Step	
		Easy	Medium	Hard	Easy	Medium	Hard	Tokens	Cost (cents)
GPT	GPT-5 Nano	71.5	24.7	4.0	3.4	7.6	9.2	2170	0.07
	GPT-5 Mini	85.5	46.0	11.0	2.2	3.9	4.6	874	0.11
	GPT-5	92.5	60.0	15.0	1.8	2.7	6.3	1826	1.51
	GPT-5.2	90.0	50.7	10.0	2.2	3.2	4.0	1008	0.85
Gemini	Gemini 2.0 Flash	88.0	41.3	6.0	4.1	6.0	14.7	501	0.01
	Gemini 2.5 Flash	91.0	53.0	10.2	2.7	4.5	8.1	547	0.05
	Gemini 2.5	91.0	56.7	15.2	2.1	3.5	6.9	527	0.19
	Gemini 3	<u>95.0</u>	<u>66.0</u>	<u>23.0</u>	0.8	1.9	4.7	1848	1.76
	Gemini 3.1	**96.0**	**69.8**	**29.0**	1.5	3.0	5.8	1905	1.84
Claude	Sonnet 4.5	88.5	43.3	10.1	2.3	4.9	10.1	1242	1.31
	Opus 4.5	91.5	56.0	18.0	1.3	2.3	8.1	1906	3.73
	Opus 4.6	91.0	56.7	16.0	1.7	3.4	10.1	1354	2.38
Grok	Grok 4.1-Fast	90.0	44.7	5.0	1.6	4.3	6.4	4458	0.21
DeepSeek	R1	91.0	54.7	17.0	1.4	4.0	6.4	2598	-
Kimi	Kimi K2	87.5	45.3	7.0	2.3	5.0	4.9	8105	-
LLaMA	LLaMA 3 1B	16.5	0.0	0.0	12.9	-	-	511	-
	LLaMA 3 3B	47.0	3.3	0.0	11.4	13.2	-	783	-
	LLaMA 3 8B	64.5	9.3	0.0	7.3	7.8	-	641	-
	LLaMA 3 70B	84.5	39.3	7.0	2.5	4.2	5.1	651	-
Gemma	Gemma 3 4B	48.0	2.7	0.0	5.2	11.5	-	555	-
	Gemma 3 12B	72.5	22.7	1.0	3.4	6.8	4.0	651	-
	Gemma 3 27B	80.0	30.0	0.0	2.8	6.4	-	684	-
Apertus	Apertus 8B	42.0	4.0	0.0	10.2	16.5	-	805	-
	Apertus 70B	65.0	10.7	0.0	8.2	13.9	-	1832	-
Mistral	Mistral 7B	59.0	10.0	1.0	7.9	8.7	12.0	664	-
	Ministral 8B	65.5	8.7	0.0	7.4	9.8	-	624	-
Qwen	Qwen 2.5-7B	22.5	1.3	0.0	13.2	16.5	-	2597	-
Diffusion	Dreamv0-In. 7B	53.0	3.3	1.0	8.7	11.2	12.0	1549	-
	LLaDA-In. 8B	40.5	4.7	0.0	7.6	12.4	-	1669	-

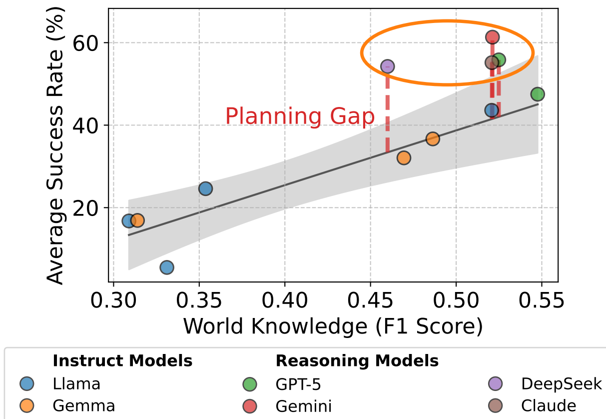


Figure 4. LLM-WikiRace evaluates both internal world knowledge and planning, we identify a clear performance gap between the Reasoning models and Instruct tuned models despite both showing similar levels of world knowledge.

within the confidence interval of the instruction-tuned models and does not exhibit a planning gap. We suspect that this is because the model is throttled to reduce its inference costs; however, since the exact implementation is unknown, we can only speculate. It is worth noting that the world knowledge probed by our classification task may differ from that required by LLM-WikiRace. Whereas classification entails a binary judgment about whether two specified pages are linked, the game demands open-ended retrieval of pages likely to be linked from a given source. This difference may affect the trends we found in our analysis. Despite these limitations, this experiment provides strong evidence that *both* reasoning and world knowledge are necessary for success on LLM-WikiRace.

5.2. Qualitative Analysis and Examples

To better understand the behavioral differences that underlie model performance, we analyze reasoning traces from two of the strongest models that expose intermediate reasoning, Gemini 3 and Claude Opus 4.5 (GPT-5 does not provide explicit reasoning traces). For each model, we randomly sample three successful and three unsuccessful trajectories at each difficulty level, yielding 18 trajectories per model.

Planning strategies and forward reasoning. Across inspected trajectories, both models frequently employ sensible high-level strategies. In particular, they often navigate from narrow topics toward broader, highly connected pages early in the episode, a behaviour reminiscent of common human WikiRace strategies. Gemini 3 explicitly frames this "hub-seeking" strategy in 72% of inspected games, while Claude Opus 4.5 does so in 94%. Representative examples of this behaviour are shown in Figure 5. In addition, both models frequently produce forward-looking reasoning, speculating about which pages may be encountered in future steps and outlining tentative multi-step plans toward the target.

Replanning failures and looping behaviour. Despite exhibiting forward planning, both models frequently revisit the same pages multiple times, entering loops from which they fail to escape. Gemini 3 encounters loops in 66% of the inspected trajectories, and Claude Opus 4.5 in 61%. In nearly all such cases, the models explicitly acknowledge the repeated state in their reasoning traces, indicating awareness of being stuck. However, awareness does not reliably translate into effective strategy revision. In many cases, the models continue alternating between neighbouring pages until the step limit is reached. This failure mode is especially pronounced on the hard split. Gemini 3 encounters loops in 86% of hard tasks, of which only 10% are solved successfully; by contrast, all hard tasks without loops are solved. Claude Opus 4.5 exhibits a similar pattern, encountering loops in 92% of hard tasks, with a success rate of only 1% among those trajectories.

In Figure 6 we plot the relationship between looping frequency and success rate for different model families. We see a strong and consistent linear trend, showing a clear negative correlation between the two quantities (regression coefficient -1.02, with 95% CI $(-1.13, -0.91)$). This provides strong quantitative evidence for the general replanning failure trend we observed during qualitative analysis.

Illustrative examples. Representative trajectories illustrating both successful and unsuccessful reasoning traces are provided in the Appendices I, J and K, and are consistent with the aggregate patterns described above.

Summary. Together, these qualitative findings show that frontier models can plan, reason about semantics, and monitor their own behaviour, but struggle with replanning—adjusting their strategy after observing that an existing approach is ineffective. Importantly, these replanning failures persist among the strongest models, highlighting remaining limitations that are not captured by aggregate success rates alone.

5.3. Does Post-Training Help?

Table 2. Success rate across difficulty levels for the base model and DAPO-finetuned checkpoints. Number after $FT-$ is the number of training steps model was finetuned for.

Model	Easy	Medium	Hard
Base	22.5	1.3	0.0
FT-50	24.0	1.3	0.0
FT-100	41.0	4.0	0.0
FT-300	**67.5**	**4.6**	0.0

We conduct a preliminary study to investigate if RL post-training can improvement the ability of models to reason on

Example 1: Gemini 3 pro, *Herstory* → *Warth*

The target page "Warth" most likely refers to a location (e.g., Warth, Vorarlberg in Austria; Warth, Thurgau in Switzerland) or a disambiguation page leading there. The current page "Herstory" is a sociological and linguistic concept regarding the perspective of history.

UNDERSTANDING TARGET AND CURRENT PAGE

To reach a specific geographical location from a sociological page, the best strategy is to move to the broadest possible topic that encompasses geography or nations.

STRATEGY

Among the options:

- **Link 25 (History)** is the parent concept of "Herstory". The page "History" is a vital, top-level article that links to broad categories like "Geography", "World", "Europe", and specific countries. This is the most efficient hub to transition from a specific feminist term to general locations.
- **Link 37 (Latin)** is a language. While it connects to European history, "History" is a more direct path to general knowledge and geography.
- Other links (Feminism, Neologism, etc.) keep the context too narrow.

CONSIDER OPTIONS

Therefore, "History" is the best link to widen the scope and eventually navigate to Austria, Switzerland, or lists of municipalities where "Warth" can be found. 25

COMMIT TO DECISION

[Solves the game succesfully by following path Herstory → History → Geography → Berlin → Bregenz → Warth]

Figure 5. Raw response of Gemini 3 pro for the game instance *Herstory* → *Warth* (optimal path length $= 4$), first step. The response highlights the model's knowledge of the start and target page as well as the overall strategy of moving towards a broad topic initially.

the benchmark. We construct a dataset of 1,000 source and target page pairs with shortest path length ranging from 2 to 6 steps; and ensure there is overlap with pairs in the easy or medium difficulty splits. Each training example consists of a game prompt (see Figure 2) instantiated at the source page and a set of candidate links. We use the DAPO algorithm (Shao et al., 2024; Guo et al., 2025; Yu et al., 2025) to train a Qwen-2.5-7B base model on this dataset. The model receives a binary reward signal, a reward of one is given if it reasons and selects the link with the shortest path length to the target page and a reward of zero otherwise. We train for up to 300 steps stopping when the validation reward plateaus, and evaluate checkpoints at steps 50, 100, and 300 on the easy, medium, and hard splits. For full details, including training hyperparameters, see Appendix F.

As shown in Table 2, fine-tuning leads to substantial improvements on the easy split, with success rates increasing from 22.5% for the base model to 67.5% after 300 fine-tuning steps. Performance on the medium split improves more modestly, from 1.3% to 4.6%, while performance on the hard split remains at 0 across all evaluated checkpoints. Taken together, these results indicate that, while DAPO-based fine-tuning can meaningfully improve performance on simpler instances, it does not, in this preliminary setting, address the challenges posed by harder LLM-WikiRace tasks.

6. Related Work

Many LLM benchmarks require planning either as a primary objective or as a subtask for success (Li et al., 2025). Prior work has focused on classical planning settings, including benchmarks such as PlanBench (Valmeekam et al., 2023) and structured games like Blockworld (Gupta & Nau, 1992), Sudoku (Seely et al., 2025), and related environments (Wu et al., 2023; Huang et al., 2024; Dagan et al., 2024; Duan et al., 2024). Analyses of these tasks (Valmeekam et al., 2024; 2025; Kambhampati et al., 2024) report mixed planning performance and question whether LLMs perform genuine planning or rely on pattern retrieval. Planning is also central to multimodal, embodied benchmarks (Shridhar et al., 2020a;b; Chen et al., 2023; Su et al., 2024; Chang et al., 2024; Yang et al., 2025b; Yin et al., 2024). In contrast, LLM-WikiRace isolates long-horizon planning and world knowledge within a single textual modality.

Realistic and open-domain planning. Recent benchmarks have moved toward more realistic settings by embedding planning within naturalistic tasks, including travel itinerary construction (Xie et al., 2024), personal scheduling (Zheng et al., 2024), agentic frameworks where planning is required to solve coding problems (Liu et al., 2023; Jimenez et al., 2023; Deng et al., 2025; Xu et al., 2024), and web-based assistant environments (Yoran et al., 2024; Zhou et al., 2023; Koh et al., 2024; Deng et al., 2023). While these benchmarks introduce real-world constraints and complexity, they are typically limited to specific domains. In con-

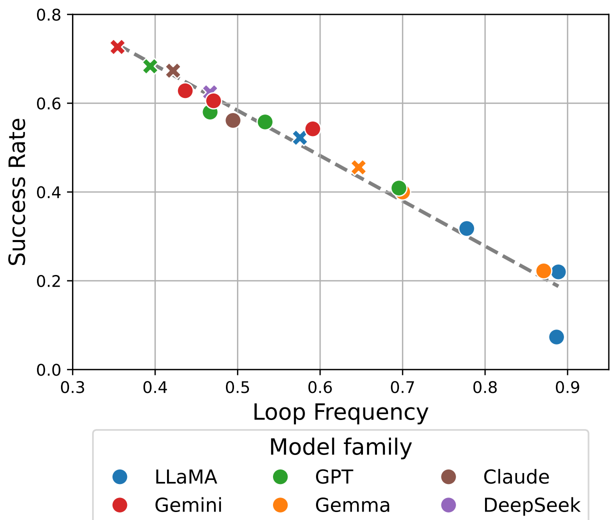


Figure 6. Success rate vs loop frequency for different models. We say that a game contains a loop if model visited any page more than once. Each model family is displayed with different colors on the scatter plot and the best model in each family is shown with an X mark. Dashed gray line shows linear regression fitted to points.

trast, LLM-WikiRace evaluates planning over a large, open-domain knowledge graph, requiring models to integrate long-horizon planning with broad, general world knowledge across a wide range of topics.

Game-based benchmarks. BALROG (Paglieri et al., 2024) evaluates planning and reasoning across a collection of game environments with varying horizons. LLM-WikiRace complements this line of work by emphasizing open-domain knowledge: success depends on understanding and exploiting relationships across a wide range of real-world concepts encoded in the Wikipedia hyperlink graph.

Concurrent work (Margiotta et al., 2025) uses the WikiRace game to contrast LLM and human reasoning capacities and (Ebrahimi et al., 2025) explores planning algorithms on a similar setting. Our work is an LLM focused benchmark and conducts a broader analysis of available models. Furthermore, we specify our benchmark independent of a human notion of difficulty and do not allow models to follow an algorithmic framework forcing the LLM to reason unaided on the graph.

7. Future Work and Conclusion

LLM-WikiRace is built on the Wikipedia hyperlink graph, yielding a diverse set of verifiable tasks that probe planning and reasoning over varying horizon lengths. Our analysis exposes several promising directions for future work. In particular, our initial fine-tuning experiments reduce the game to a one-step reasoning problem during training, which could naturally be extended to multi-step reinforcement learning approaches. More broadly, understanding how planning frequency (Paglieri et al., 2025), retrieval-augmented generation (Gao et al., 2023b), and other agentic frameworks influence long-horizon reasoning remains an open and fertile area of study.

LLM-WikiRace evaluates an LLM's ability to integrate world knowledge into planning and decision-making. While frontier models perform well on the easiest split, none achieve a success rate above 30% on the hardest split. We highlighted how strong reasoning capabilities differentiate the best models despite showing similar levels of knowledge about the underlying graph the game is based on. Even these models exhibit systematic planning failures on difficult tasks, including looping behavior, reluctance to adapt initial strategy, and insufficient exploration. Overall, LLM-WikiRace is easy for models to understand yet difficult to master, exposing persistent limitations in long-horizon reasoning and planning when models must act on their world knowledge.

References

Anthropic. The Claude 3 model family: Opus, sonnet, haiku. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf, 2024. Accessed: 2024-03-04.

Chang, M., Chhablani, G., Clegg, A., Cote, M. D., Desai, R., Hlavac, M., Karashchuk, V., Krantz, J., Mottaghi, R., Parashar, P., et al. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. *arXiv preprint arXiv:2411.00081*, 2024.

Chen, Y., Ge, Y., Ge, Y., Ding, M., Li, B., Wang, R., Xu, R., Shan, Y., and Liu, X. Egoplan-bench: Benchmarking multimodal large language models for human-level planning. *arXiv preprint arXiv:2312.06722*, 2023.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Dagan, G., Keller, F., and Lascarides, A. Plancraft: an evaluation dataset for planning with llm agents. *arXiv preprint arXiv:2412.21033*, 2024.

Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., and Su, Y. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Deng, X., Da, J., Pan, E., He, Y. Y., Ide, C., Garg, K., Lauffer, N., Park, A., Pasari, N., Rane, C., et al. Swe-bench pro: Can ai agents solve long-horizon software engineering tasks? *arXiv preprint arXiv:2509.16941*, 2025.

Duan, J., Zhang, R., Diffenderfer, J., Kailkhura, B., Sun, L., Stengel-Eskin, E., Bansal, M., Chen, T., and Xu, K. Gt-bench: Uncovering the strategic reasoning capabilities of llms via game-theoretic evaluations. *Advances in Neural Information Processing Systems*, 37:28219–28253, 2024.

Ebrahimi, R., Fuhrman, S., Nguyen, K., Gurusankar, H., and Franceschetti, M. Textual understanding boost in the wikirace. *arXiv preprint arXiv:2511.10585*, 2025.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023a.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H., and Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023b.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Gupta, N. and Nau, D. S. On the complexity of blocks-world planning. *Artificial intelligence*, 56(2-3):223–254, 1992.

Hendrycks, D., Burns, C., Basart, S., Critch, A., Li, J., Song, D., and Steinhardt, J. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021a.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021b.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021c.

Hernández-Cano, A., Hägele, A., Huang, A. H., Romanou, A., Solergibert, A.-J., Pasztor, B., Messmer, B., et al. Apertus: Democratizing open and compliant LLMs for global language environments. *arXiv preprint arXiv:2509.14233*, 2025.

Huang, J.-t., Li, E. J., Lam, M. H., Liang, T., Wang, W., Yuan, Y., Jiao, W., Wang, X., Tu, Z., and Lyu, M. R. How far are we on the decision-making of llms? evaluating llms' gaming ability in multi-agent environments. *arXiv preprint arXiv:2403.11807*, 2024.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Le Scao, T., Lavril, T., Wang, T., Lacroix, T., and El Sayed, W. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

Kambhampati, S., Valmeekam, K., Guan, L., Verma, M., Stechly, K., Bhambri, S., Saldyt, L., and Murthy, A. Llms can't plan, but can help planning in llm-modulo frameworks. *arXiv preprint arXiv:2402.01817*, 2024.

Koh, J. Y., Lo, R., Jang, L., Duvvur, V., Lim, M., Huang, P.-Y., Neubig, G., Zhou, S., Salakhutdinov, R., and Fried, D. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, H., Chen, Z., Zhang, J., and Liu, F. Planet: A collection of benchmarks for evaluating llms' planning capabilities. *arXiv preprint arXiv:2504.14773*, 2025.

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023.

Margiotta, D., Croce, D., and Basili, R. Evaluating large language models on wikipedia graph navigation: Insights from the wikigame. In *Proceedings of the Eleventh Italian*

Conference on Computational Linguistics (CLiC-it 2025), pp. 659–669, 2025.

Mistral AI. Un Ministral, des Ministraux: Introducing the world's best edge models. `https://mistral.ai/news/ministraux`, October 2024.

Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., and Li, C. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Paglieri, D., Cupiał, B., Coward, S., Piterbarg, U., Wolczyk, M., Khan, A., Pignatelli, E., Kuciński, Ł., Pinto, L., Fergus, R., et al. Balrog: Benchmarking agentic llm and vlm reasoning on games. *arXiv preprint arXiv:2411.13543*, 2024.

Paglieri, D., Cupiał, B., Cook, J., Piterbarg, U., Tuyls, J., Grefenstette, E., Foerster, J. N., Parker-Holder, J., and Rocktäschel, T. Learning when to plan: Efficiently allocating test-time compute for llm agents. *arXiv preprint arXiv:2509.03581*, 2025.

Seely, J., Imajuku, Y., Zhao, T., Cetin, E., and Jones, L. Sudoku-bench: Evaluating creative reasoning with sudoku variants. *arXiv preprint arXiv:2505.16135*, 2025.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., and Fox, D. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10740–10749, 2020a.

Shridhar, M., Yuan, X., Côté, M.-A., Bisk, Y., Trischler, A., and Hausknecht, M. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020b.

Singh, A., Fry, A., Perelman, A., Tart, A., Ganesh, A., El-Kishky, A., McLaughlin, A., Low, A., Ostrow, A., Ananthram, A., et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Su, Y., Ling, Z., Shi, H., Jiayang, C., Yim, Y., and Song, Y. Actplan-1k: Benchmarking the procedural planning ability of visual language models in household activities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14953–14965, 2024.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Team, K., Bai, Y., Bao, Y., Chen, G., Chen, J., Chen, N., Chen, R., Chen, Y., Chen, Y., Chen, Y., et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Valmeekam, K., Marquez, M., Olmo, A., Sreedharan, S., and Kambhampati, S. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36:38975–38987, 2023.

Valmeekam, K., Stechly, K., and Kambhampati, S. Llms still can't plan; can lrms? a preliminary evaluation of openai's o1 on planbench. *arXiv preprint arXiv:2409.13373*, 2024.

Valmeekam, K., Stechly, K., Gundawar, A., and Kambhampati, S. A systematic evaluation of the planning and scheduling abilities of the reasoning model o1. *Transactions on Machine Learning Research*, 2025.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Wu, Y., Tang, X., Mitchell, T. M., and Li, Y. Smartplay: A benchmark for llms as intelligent agents. *arXiv preprint arXiv:2310.01557*, 2023.

Xie, J., Zhang, K., Chen, J., Zhu, T., Lou, R., Tian, Y., Xiao, Y., and Su, Y. Travelplanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*, 2024.

Xu, F. F., Song, Y., Li, B., Tang, Y., Jain, K., Bao, M., Wang, Z. Z., Zhou, X., Guo, Z., Cao, M., et al. Theagentcompany: benchmarking llm agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025a.

Yang, R., Chen, H., Zhang, J., Zhao, M., Qian, C., Wang, K., Wang, Q., Koripella, T. V., Movahedi, M., Li, M., et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025b.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., and Kong, L. Dream 7B: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Yin, S., Pang, X., Ding, Y., Chen, M., Bi, Y., Xiong, Y., Huang, W., Xiang, Z., Shao, J., and Chen, S. Safeagentbench: A benchmark for safe task planning of embodied llm agents. *arXiv preprint arXiv:2412.13178*, 2024.

Yoran, O., Amouyal, S. J., Malaviya, C., Bogin, B., Press, O., and Berant, J. Assistantbench: Can web agents solve realistic and time-consuming tasks? *arXiv preprint arXiv:2407.15711*, 2024.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Zheng, H. S., Mishra, S., Zhang, H., Chen, X., Chen, M., Nova, A., Hou, L., Cheng, H.-T., Le, Q. V., Chi, E. H., et al. Natural plan: Benchmarking llms on natural language planning. *arXiv preprint arXiv:2406.04520*, 2024.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

Appendix

Table of Contents

A Technical details of the implementation **13**

B How was the difficulty split decided? **14**

C Ablation of Action Space Trimming Method **15**

D Ablation on Page Context **16**

E World Model Knowledge Testing Details **17**

F Post-training Experiment **18**

G Comparison with Human Performance **19**

H Additional Analysis of Loops and Recovery **21**

H.1 Loop awareness and adaptive resilience 21

I Example traces - Gemini 3 **22**

J Example traces - Claude Opus 4.5 **26**

K Example traces - Contrasting Gemini 3 and Claude Opus 4.5 **29**

A. Technical details of the implementation

Algorithm 1 WikiRace Game Procedure

1: **Input:** Source page s, target page g, maximum steps T, maximum links N
2: Initialize current page $p \leftarrow s$
3: **for** $t = 1$ **to** T **do**
4:     Construct a prompt containing the current page p, target page g, traversal history, and outgoing links
5:     Retrieve outgoing pages $\mathcal{N}(p)$ from the WikiGraph
6:     **if** $|\mathcal{N}(p)| > N$ **then**
7:         Select up to N outgoing links with the shortest precomputed distance to g
8:     **end if**
9:     Prompt the LLM to select one link
10:     Update $p \leftarrow$ selected link
11:     **if** $p = g$ **then**
12:         **Success**
13:         **break**
14:     **end if**
15: **end for**
16: **Failure**

WikiGraph Structure. We represent the Wikipedia hyperlink graph as a sparse, directed graph in compressed sparse row (CSR) format. Each node corresponds to a Wikipedia page (mapped from page ID to title), and each directed edge represents an outgoing hyperlink. Formally, the graph is encoded as an adjacency matrix $A \in \{0,1\}^{N\times N}$, stored using two arrays: (i) `row_ptr` of length $N{+}1$, and (ii) `col_indices` of length $\mathrm{nnz} = |\{(i,j) : A_{ij} = 1\}|$.

For each node $i \in \{0, \ldots, N{-}1\}$, the slice

```
col_indices[row_ptr[i] : row_ptr[i+1]]
```

returns the indices of all out-neighbours of i, corresponding to the targets of outgoing hyperlinks from page i. This representation enables $O(1)$ access to a node's neighbour list boundaries and requires $O(N + \mathrm{nnz})$ memory, with good cache locality for row-wise traversal.

Shortest-Path Computation. To compute shortest-path distances between pages, we use the SciPy (Virtanen et al., 2020) function `scipy.sparse.csgraph.shortest_path` over the sparse WikiGraph. Shortest-path distances from each target page are precomputed once and cached prior to gameplay. These distances are used exclusively by the environment to determine task difficulty and to filter outgoing links, and are never exposed to the model during inference.

Finally, a detailed algorithm of how the LLM-WikiRace game is played can be found in Algorithm 1.

Compute Resources. For all experiments the open source models with $\leq 70B$ parameters were run locally on 2 H200 GPUs. Larger models were run through the open router API.

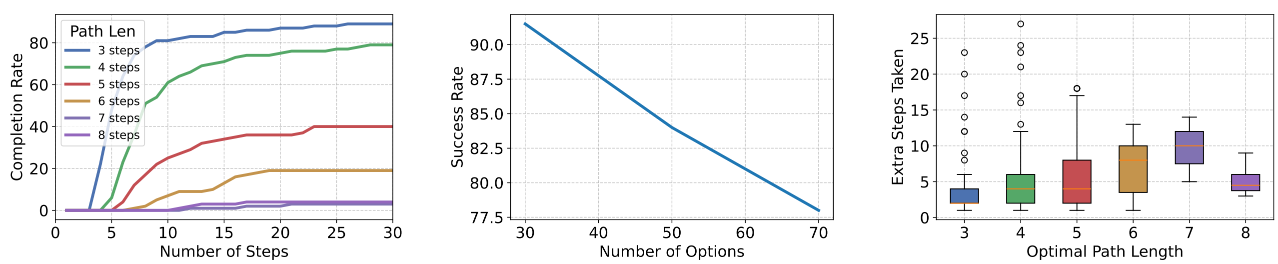
B. How was the difficulty split decided?

To better understand how large language models approach the task, we analyse the performance of LLAMA-3.3-70B-INSTRUCT across a range of experimental settings. In Figure 7 a), we plot the average completion rate, whether the LLMs successfully reaches the target page within T steps, over 50 games as a function of the maximum number of allowed steps T, stratified by the minimum path length, O, between the source and target pages. We observe a clear relationship between task difficulty—as measured by the shortest-path distance, and model performance: completion rates decrease as the minimum path length increases.

Increasing the number of allowed steps initially improves performance, as it enables the model to recover from early suboptimal decisions. However, the benefits of additional planning steps diminish beyond a certain point, and performance eventually plateauing despite increasing compute. In Figure 7 b), we analyse how the number of options presented to the LLM affects performance at step 30 on paths with lengths 3 & 4. We found, as one might expect, that more options makes the task harder. Based off this experiment we set the number of options to be 50 at each timestep, we did not pick 70 to reduce the number of tokens in each prompt and further reduce the cost of running LLM-WikiRace.

In Figure 7 c), we further analyze the number of steps taken in excess of the minimum path length. This metric captures the extent to which the model deviates from optimal play. We find that suboptimality increases with the optimal path length, suggesting that small errors compound over longer trajectories. Based on these exploratory results, we fix the maximum number of allowed steps to $T = 30$ for all subsequent experiments.

Based on this initial exploration, we formulate three LLM-WikiRace difficulty splits: EASY, MEDIUM, and HARD. These splits are created by sampling source pages at random from the graph and selecting a target pages at minimum O pages away. For the easy split we select 200 nodes with the shortest possible path equally split between 3 and 4 pages, the medium baseline consists of 150 nodes with an equal split between 5 and 6 pages, finally the hard baseline consists of 100 pairs equally split between lengths 7 and 8. For the harder baselines models often require more steps, to reduce the cost of running the baseline we reduce the number of points to be evaluated as the difficulty of the splits increases.



(a) Completion rate *(b)* Options vs. success *(c)* Extra steps

Figure 7. **We create easy, medium, and hard difficulty splits using the length of the shortest path between the start and target concept to determine difficulty. a)** The completion rate is plotted against the number of game steps for LLAMA-3.3-70B-INSTRUCT, over a variety different optimal path lengths. Across different path lengths most models consistently pass or fail by step 30. **b)** We analyse the performance of the easy split, when the model is provided with $\{30, 50, 70\}$ next node options, we settle on 50 options at each step to ensure the easy split is not saturated. **c)** The extra steps taken for successful runs increases as the optimal path length increases, we hypothesize this is because more sub-optimal moves accumulate over longer tasks and require more steps to correct.

C. Ablation of Action Space Trimming Method

In this section we explore how our link pruning approach affects the performance of a range of models on LLM-WikiRace. In Table 1 we show results for the setting where the top 50 links ordered by the length of the shortest path from that page to the target page are presented to the model in a random order. As discussed in Section 3 around 70% of the nodes in the preprocessed hyperlink graph have less than 50 outgoing links, leaving just 30% of nodes in the graph effected by this design decision. To provide a thorough analysis of our design choices we also investigate how selecting links randomly affects a range of open source and closed source models. The results shown in Table 3 and Table 4 compare the random link selection success rate with the top-50 link selection results reported in Table 1.

The majority of models show little to no difference in performance, as measured by success rate, when switching from top-50 shortest path link selection to random link selection. The exception to this is Gemini 2.5-Flash which sees a huge drop in performance of 10% and 20% on the easy and medium difficult splits respectively. This large swing in performance is not observed in the open source model results, which shows comparable performance between the random and top-50 link selection settings.

Model	Link Selection	Easy	Medium	Hard
GPT-5 Mini	Top-50	85.5	46.0	11.0
GPT-5 Mini	Top-1 + Random	86.5	46.98	7.00
Gemini 2.5-Flash	Top-50	91.0	53.00	10.2
Gemini 2.5-Flash	Top-1 + Random	82.0	33.33	3.00

Table 3. Closed source success rate (%) under different link selection strategies

Model	Link Selection	Easy	Medium	Hard
Apertus 70B	Top-50	65.0	10.7	0.0
Apertus 70B	Top-1 + Random	65.0	12.0	0.0
Apertus 8B	Top-50	42.0	4.0	0.0
Apertus 8B	Top-1 + Random	41.5	4.7	0.0
Gemma 3 12B	Top-50	72.5	22.7	1.0
Gemma 3 12B	Top-1 + Random	70.5	20.7	1.0
Gemma 3 27B	Top-50	80.0	30.0	0.0
Gemma 3 27B	Top-1 + Random	80.5	30.0	0.0
Gemma 3 4B	Top-50	48.0	2.7	0.0
Gemma 3 4B	Top-1 + Random	45.0	4.0	0.0
LLaMA 3 1B	Top-50	16.5	0.0	0.0
LLaMA 3 1B	Top-1 + Random	16.5	0.0	0.0
LLaMA 3 3B	Top-50	47.0	3.3	0.0
LLaMA 3 3B	Top-1 + Random	51.5	6.7	0.0
LLaMA 3 70B	Top-50	84.5	39.3	7.0
LLaMA 3 70B	Top-1 + Random	84.5	39.3	7.0
LLaMA 3 8B	Top-50	64.5	9.3	0.0
LLaMA 3 8B	Top-1 + Random	67.5	14.0	1.0

Table 4. Open Source success rate (%) under different link selection strategies

D. Ablation on Page Context

As part of the LLM-WikiRace prompt, shown in Figure 2, we provide the model the current page, target page, and available links; we exclude the article body for the current page. We do this for a variety of reasons, practical and principled, covered in Section 3. To thoroughly ablate the design of LLM-WikiRace we briefly conduct an ablation experiment of this design decision over a range of open source models. We generate a new prompt, shown in Figure 8, and use the Wikipedia API to add a summary of the {current_page} to the prompt. The results are shown in Table 5.

Including a summary of the current page in the prompt does not consistently help models. The Apertus-8B and two Gemma-3 models show improvements of ∼5% on the easy split but no significant performance improvements on the medium and hard difficulties. For the Llama-3 models and the Qwen-2.5-7B-Instruct model, we observe that including this additional context reduces the model performance on the easy split, in some cases by up to 8%, and dramatically hurts the model performance on the medium difficulty.

WikiRace Context Game Prompt

System Prompt:
You are a helpful assistant helping play the Wikipedia link game.

User Prompt:
You are playing a game where you start at Wikipedia page "{current_page}" and want to reach page "{target_page}" by clicking links.

Context for the current page ("{current_title}"):
"{current_context}"

So far, you have visited the following pages in order:
{history}

You see the following possible links from the current page:

0. {Neighbor 0 Title}
1. {Neighbor 1 Title}
⋮
{N}. {Neighbor N Title}

Which link should you click to get closer to the target? Reply with the number of your choice (0 to {max_choice_num}).

Figure 8. The LLM-WikiRace prompt with current page context added.

Model	Page Context	Easy	Medium	Hard
Apertus 8B	False	41.5	4.7	0.0
Apertus 8B	True	46.5	6.7	0.0
Gemma 3 12B	False	70.5	20.7	1.0
Gemma 3 12B	True	75.5	20.0	0.0
Gemma 3 27B	False	80.5	30.0	0.0
Gemma 3 27B	True	82.0	29.3	0.0
LLaMA 3 3B	False	51.5	6.7	0.0
LLaMA 3 3B	True	31.0	0.7	0.0
LLaMA 3 8B	False	67.5	14.0	1.0
LLaMA 3 8B	True	63.5	5.3	1.0
Qwen/Qwen2.5-7B-Instruct	False	49.0	4.0	0.0
Qwen/Qwen2.5-7B-Instruct	True	41.0	3.3	0.0

Table 5. Results for the context prompt vs. no-context prompt ablation experiment of LLM WikiRace

E. World Model Knowledge Testing Details

WikiRace World Model Experiment Prompt

User Prompt:
We are playing a game where you navigate Wikipedia from a starting page to a target page solely by clicking links on each page. Can you tell us if {source_page} contains a link to {target_page}? Reply with either 'yes' or 'no' in the following format:`\boxed{}`.

Figure 9. A schematic of the prompt provided to the LLM, when asking it to select the link to follow

We now provide further details on the world knowledge experiment. To disentangle planning and world-knowledge we construct a classification task to probe the model's understanding of the underlying Wiki-graph. We ask the model to differentiate between pairs of connected and unconnected nodes. To better characterize different types of non-connectivity, we include several classes of unconnected pairs. Specifically, we sample target nodes that are i steps away from the source node for $i \in \{2, 3, 4\}$, with 200 samples included for each distance, and 200 samples of directly connected nodes. In addition, we include a *reversed connection* category, designed to be particularly challenging, in which the target node links to the source node but not visa-versa. For each source–target pair, we prompt the model using the prompt in Figure 9 to predict whether the two nodes are connected, and parse the resulting binary yes/no response in a `\boxed{}` format. In the smaller models if a response is not parsed we disregard the sample when calculating the world knowledge F1-score.

F. Post-training Experiment

Hyperparameter	Value
Model & Algorithm	
Base model	Qwen2.5-7B
RL algorithm	DAPO
KL penalty	low-variance KL
Optimization	
Learning rate	1×10^{-6}
LR schedule	constant
LR warmup ratio	0.0
Gradient clipping	1.0
Total training steps	300
Train batch size	64
PPO mini-batch size	16
PPO epochs	1
DAPO Objective	
Clip ratio (low)	0.2
Clip ratio (high)	0.28
KL loss coefficient	1×10^{-3}
Entropy coefficient	0.0
Loss aggregation	token-mean
Discount γ	1.0
GAE λ	1.0
Rollout / Sampling	
Rollouts per prompt (n)	16
Validation rollouts	8
Sampling temperature	1.0
Top-p / Top-k	1.0 / -1
Max prompt length	1024
Max response length	1024
Filtering	
Rejection sampling	enabled
Rejection multiplier	3.0
Max generation batches	5

Table 6. Training hyperparameters for DAPO fine-tuning experiment on LLM-WikiRace.

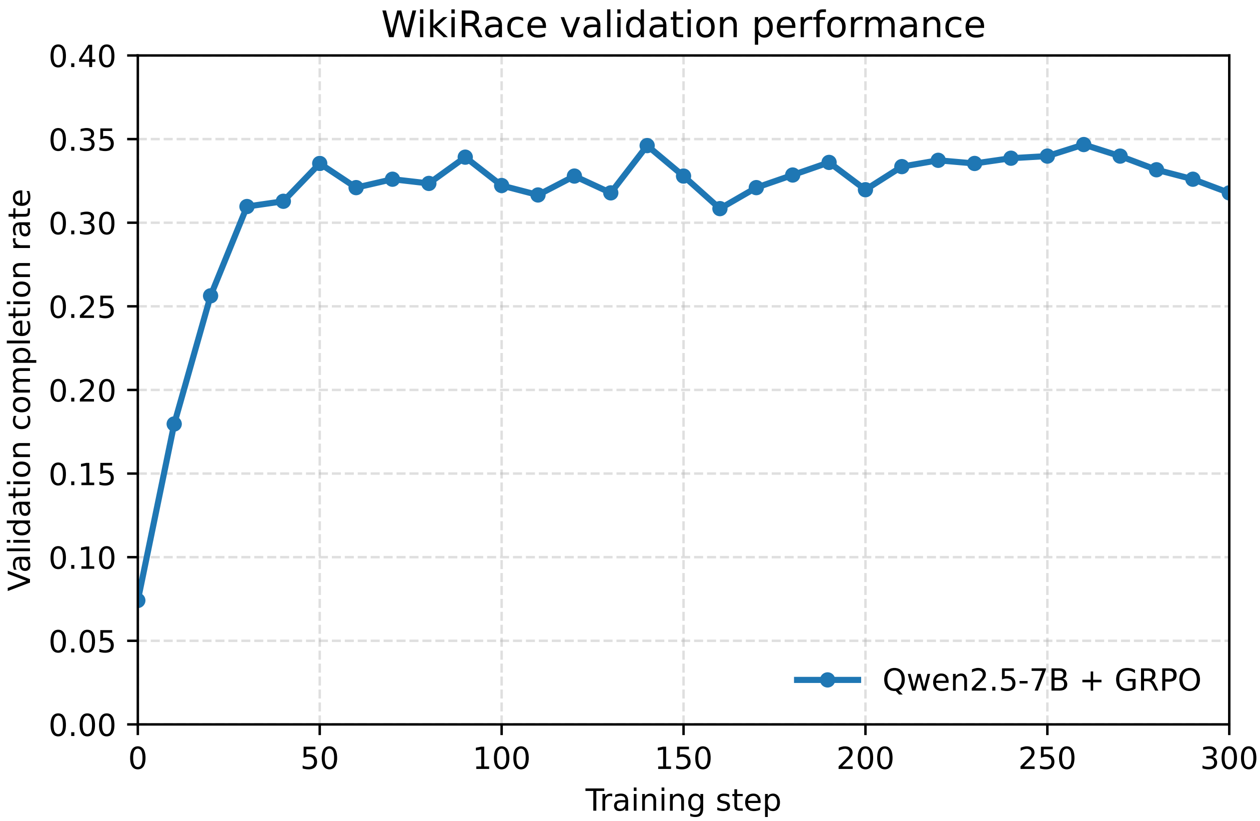


Figure 10. Validation completion rate on LLM-WikiRace throughout DAPO fine-tuning of Qwen2.5-7B. The model improves rapidly during the first ∼30 steps, climbing from a 7.4% zero-shot baseline to over 30%, before plateauing around 33–35% for the remainder of training.

G. Comparison with Human Performance

To provide a coarse reference for human gameplay, we use publicly released statistics from The WikiGame Daily[3], collected from daily WikiRace games announced on the platform's X account[4]. We identify the corresponding start–target page pairs in our Wikipedia graph and refer to this set as the *Human Gameplay Corpus*. We then evaluate the three strongest models—Gemini 3, GPT-5, and Claude Opus 4.5—on this corpus.

All three models achieve a success rate of 100% on the Human Gameplay Corpus, while the reported human success rate is 98.5%. This indicates that the corpus is easier than the easiest split of our benchmark. As success rates are saturated, we instead compare the number of suboptimal steps taken by models and human players. Results are shown in Figure 11. On average, human players take at least one more suboptimal step than any of the evaluated models.

We emphasise that this corpus should be interpreted as a coarse reference rather than a precise estimate of human performance, due to heterogeneous player skill and potential graph reconstruction effects, human players are presented with all the links on a specific page whilst we limit the number of links in LLM-WikiRace. Nevertheless, the evaluated models consistently exceed this reference in terms of path optimality.

[3]https://www.thewikigamedaily.com
[4]https://x.com/dailywikigame

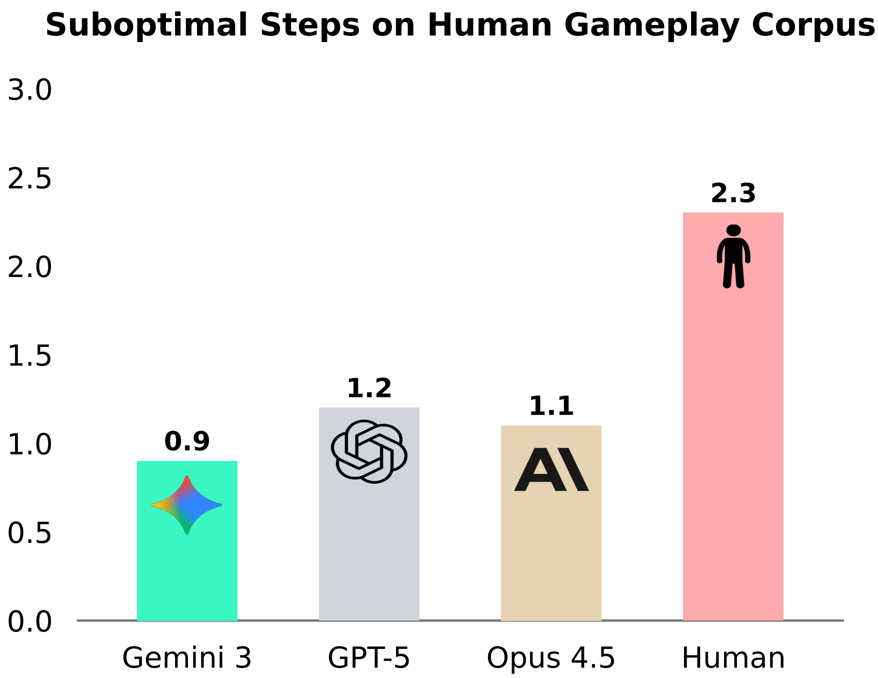


Figure 11. Comparison of suboptimal steps made by the most powerful models and by Human players, across the Human Gameplay Corpus games. The success rate on this corpus for was 98.5% for humans and 100% for all tested models.

H. Additional Analysis of Loops and Recovery

To disentangle navigational precision from agentic adaptability / replanning abilities, we analyze the agent's behaviour regarding state revisitation. We define a *loop* as any instance where an agent visits a specific page more than once within a single trajectory. We utilize three metrics to capture different aspects of this behaviour:

- **Loop Frequency (Precision):** The proportion of tasks where a loop occurred. This measures the agent's ability to navigate the graph linearly without errors.
- **Recovery Rate (Adaptability):** Conditioned on a loop occurring, this measures the proportion of tasks where the agent eventually solves the task. This serves as a proxy for the agent's capacity to recognize it is in a cycle and adapt its strategy successfully to escape and solve the task.
- **Average Maximum Visitation (Stagnation):** The average maximum number of times a single page is visited. High values here indicate "blind" looping, where the agent fails to realize it is repeating actions.

This correlation persists in the Medium split but shifts along the diagonal as task complexity increases. In the Hard split, all models are clustered in the bottom-right corner. Here, the difficulty of the tasks forces nearly all models into frequent looping behaviours ($> 80\%$ loop frequency) with correspondingly low success rates ($< 25\%$). This suggests that in complex environments, once an agent loses its linear path, it rarely recovers.

H.1. Loop awareness and adaptive resilience

While Loop Frequency measures error quantity, the *Recovery Rate* and *Maximum Visitation* (Figure 12) reveal the model's response to those errors. This allows us to distinguish between agents that get "stuck" versus those that adapt.

Adaptive Behavior in Easy Tasks. On the Easy split, we observe strong evidence of loop awareness in capable models. For example, Gemini 3 Pro loops in 13% of tasks but achieves a remarkable recovery rate of 62%. This indicates that , a loop is often a temporary detour rather than a failure. As highlighted by the qualitative analysis, the agent frequently recognizes the revisited state and successfully pivots to a new path.

The "Blind Loop" in Complex Tasks. As difficulty increases, this adaptive capacity degrades. For instance on the Medium split, both Kimi K2 and Llama 3.3 70B loop at the exact same rate (0.73). However, Kimi K2 recovers nearly twice as often (0.29 vs. 0.17), suggesting it retains some ability to break cycles, whereas the Llama model is more prone to repetitive failure.

By the Hard split, "blind looping" becomes dominant. Recovery rates decrease to near 0.0 for most models, with the highest recovery rates of 12% and 10% being achieved by Claude Opus 4.5 and Gemini 3 Pro, respectively. The *Average Maximum Visitation* increases significantly for most models (e.g., GPT-5 Nano: 7.1, Kimi K2: 6.2). These high visitation counts confirm that in the hardest scenarios, agents lose their ability to adapt: they are looping and continue to revisit the same states 6+ times without managing to solve the task.

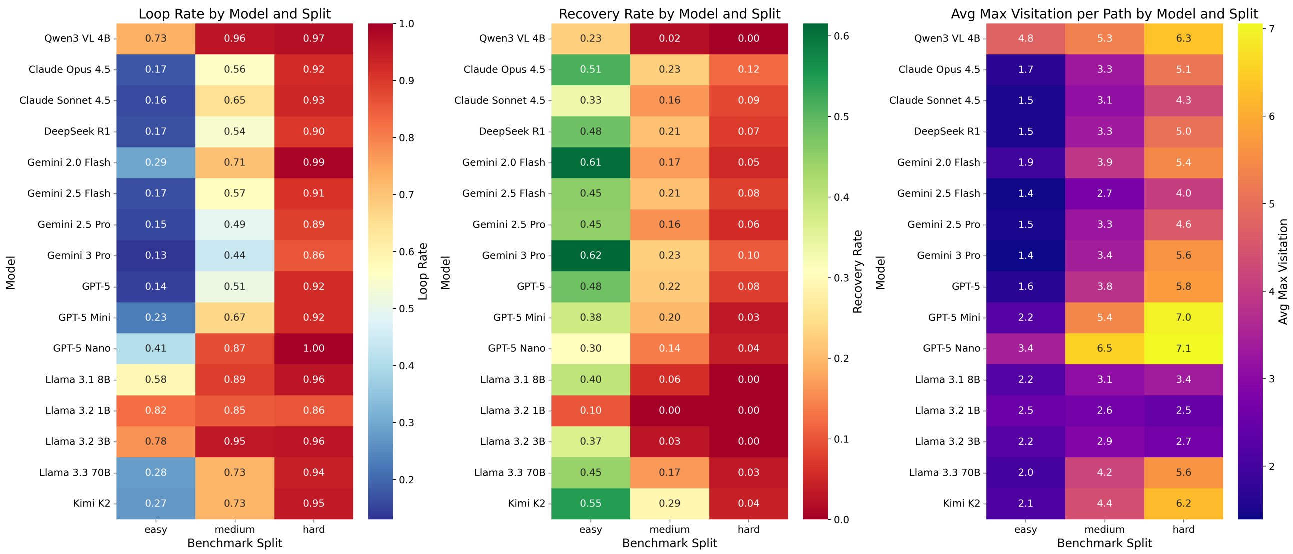


Figure 12. Recovery and max visitation

I. Example traces - Gemini 3

We now present several example traces from games played by the Gemini 3 model across different optimal path lengths. These traces are raw answers provided by the model and have only been cropped where indicated to reduce length and improve readability. Example 2 shown in Figure 13 shows how Gemini 3 plans ahead by reasoning through expected future connections. Similarly, in Example 3 in Figure 14 we observe that the model explicitly states possible paths it may encounter for a given choice. Example 4 (Figure 15) presents a recovery, where the model notices a loop in its path and successfully corrects its decision to complete the task. Conversely, Example 5 (Figure 16) highlights a clear limitation in terms of the replanning ability. Despite repeating the same decision at multiple steps the model fails to escape the loop.

Example 2: Gemini 3, Luminogram → Ionides, optimal path length = 3, first step

Based on the current page "Luminogram", [...], the best strategy is to move to the broadest category available to find links related to art history, museums, or biography.

"Photography" is the main parent topic. From "Photography", one can navigate to "History of photography", "Art", or specific museums like the "Victoria and Albert Museum" (which holds the Ionides bequest and an extensive photography collection), or to famous photographers (like Julia Margaret Cameron, who was close friends with the Ionides family).

PLANNING AHEAD

None of the other links (specific techniques, equipment, or chemical processes) are likely to provide a direct path to a specific family surname or art history topic as efficiently as the main subject article.

37

Figure 13. Raw answer of Gemini 3 highlighting the model's ability to plan ahead.

Example 3: Gemini 3, Pierrefitte-sur-Aire → Fibrin, first step

{thinking tokens}
Analyzing the Path from Pierrefitte-sur-Aire to Fibrin

Okay, here's what I'm thinking. I've got this page on Pierrefitte-sur-Aire, a small commune in France, and the target concept is "Fibrin," the blood clotting protein. I need to figure out the best way to get from one to the other.

[..]

The obvious "hub" to get out of this local geographical area is the link to "France" itself. It's a major page with sections on demographics, economy, science, and education, all of which could potentially branch out towards relevant biological concepts.

I can see paths like "Health in France" leading to "Medicine" and then "Blood," eventually pointing towards "Fibrin." Or I could go through "Science and technology in France" and then Biology. I'm also considering the possibility of something like "Louis Pasteur," a famous French biologist, as another potential route, although that's more of a gamble.
I'm pretty confident that "France" offers the broadest and most promising set of potential links. I'm envisioning paths like: "France" -> "Developed country" -> "Health care/Life expectancy" -> "Medicine" -> "Blood" -> "Fibrin." or "France" -> "French Academy of Sciences" -> Biology.

MAPPING OUT POTENTIAL PATHS

[. . .] So, yeah, I'm sticking with "France" as the best option, the safest bet to get out of this geographic loop and find that biological connection. This hub strategy is the best way to escape a cluster of obscure articles. I'm pretty sure about this.

Figure 14. Raw answer reveals the model is explicitly stating possibly future paths when reasoning about the decision.

Example 4: Gemini 3, Luminogram → Ionides, optimal path length = 3

Step 9: Current page "Parthenon", Gemini chooses "Athens":

[...] Looking at the links:
Most links (Enna, Poseidon, Paestum, Centaur, etc.) keep you deep in Ancient Mythology or Geography.

Athens (48) is the modern capital and the location of the Parthenon. The article on Athens will almost certainly link to London, the British Museum (in the context of the marbles), or United Kingdom, which are the necessary bridges to reach the Anglo-Greek Ionides family and their collections.

SUPPORTING ARGUMENT FOR ATHENS

[...]

Step 15: Current page "Parthenon", Gemini chooses "Athens":
The goal is to reach the page "Ionides" (referring to the prominent Ionides family of art collectors, [. . .] to British museums. To reach "Ionides", the most logical path is to navigate toward London, the British Museum, or the Victoria and Albert Museum.

From the "Parthenon" page, the best link to bridge the gap between Ancient Greece and Modern London/Museums is Athens. The "Athens" article is a major hub that discusses the Elgin Marbles (Parthenon Sculptures), which provides a direct link to the British Museum and mentions London. From the British Museum or London pages, it is much easier to find the Victoria and Albert Museum or the Ionides family.

SUPPORTING ARGUMENT FOR ATHENS (FAILS TO RECOGNIZE REPETITION)

[...]

Step 26: Current page "Parthenon", Gemini chooses "Ephesus":
[...]

The user has been stuck in a loop around Athens and the Acropolis. To break this loop and move towards the British art world or the British Museum (which holds the Parthenon/ Elgin marbles and links to other UK collections):

1. Athens (20) is a hub, but the user has visited it multiple times, leading to loops.

RECOGNIZES LOOP

2. Ephesus (44) is a major ancient site (Temple of Artemis). [...] Clicking Ephesus offers a strong path to the British Museum, which is a key node to reach specific British collectors or the Victoria and Albert Museum (where the Ionides collection resides).[...]

Ephesus is the best option to break the geographical loop of Athens while maintaining a thematic connection to antiquities that reside in London museums.

CHOICE TO BREAK LOOP

[Solves the game within 30 steps]

Figure 15. The model is initially stuck in a loop repeating actions in steps 9 and 15 without realising the repetition. However, in step 26 the model shows awareness and chooses an alternative options to break out of the loop. This leads to a successful completion of the task.

Example 5: Gemini 3, Megalocaria → Rojhuwas, optimal path length = 8

Initial Progress (Steps 1–7):
Megalocaria → Coccinellidae → Shellac → Mahabharata → India → Haryana → Dharuhera → Bawal → Rewari

Loop Pattern (Steps 8–30):
The model then oscillated between **Bawal** and **Rewari** for the remainder of the task, with the exception of visiting Haryana, Dharuhera, and Nikhri before returning to the loop.

Some of the steps illustrating the loop:

Step 8: Rewari → Bawal

> "Rojhuwas is a village located in the **Bawal** tehsil of the **Rewari** district in Haryana... Wikipedia pages for Tehsils often contain lists or templates of the villages contained within them. **Bawal** is the specific Tehsil where Rojhuwas is located...clicking Bawal is the most direct route to find the village Rojhuwas."

Step 9: Bawal → Rewari

> "Since 'Rojhuwas' is not listed in the available links (0–26), you need to navigate to a page that is likely to list villages in the Rewari district or Bawal tehsil. **Rewari** is the district headquarters. Wikipedia pages for districts or district headquarters often contain lists of tehsils, blocks, and sometimes significant villages..."

Step 14: Rewari → Bawal

> "Rojhuwas is a village in the Rewari district... Since I didn't find 'Rojhuwas' or a 'Villages in Bawal' list, I will reassess the potential to navigate to the 'Rewari district' page... My strategy involves prioritizing direct administrative links..."

Step 21: Bawal → Rewari

> "Rojhuwas is not listed in the direct links from Bawal (0–26)... Since 'Rojhuwas' is administratively part of the Bawal tehsil of the Rewari district, and it is not explicitly listed... you must navigate to a page that functions as a parent or hub. **Rewari** is the district headquarters... The other links like Alaska, Birds, or distant cities are unlikely to lead to a specific village in Haryana."

[Fails to solve task within the step limit]

Figure 16. Gemini 3 fails to escape the loop and ultimately fails to complete the task, which highlights a critical limitation in terms of replanning ability.

J. Example traces - Claude Opus 4.5

As shown in Examples 6-9 (Figures 17–20), Claude 4.5 Opus exhibits a high-level strategy and response pattern consistent with Gemini 3. It performs explicit planning and deliberate navigation through hub pages to cover multiple topics. The traces also reveal limitations: despite recognising a loop in its path (Example 9 in Figure 20), the model remains stuck for several steps and only adapts after repeated failures.

Example 6: Claude Opus 4.5, *Herstory* → *Warth*

I need to find a path from "Herstory" to "Warth". Let me think about what "Warth" might be - it could be a place name, perhaps a village or town somewhere.

UNDERSTANDING TARGET AND CURRENT PAGE

Looking at the available links, I need to find one that might lead me toward "Warth". Since Warth seems like it could be a geographical location (possibly in Austria, Switzerland, or Germany based on the name), I should look for links that might lead to geography-related topics.

STARTEGY

From the current options, most are related to feminism and gender studies. Let me consider which might be most likely to lead to a geographical location:

1. "History" (#25) - This is a broad topic that could link to many things, including historical places
2. "Latin" (#2) - Language topic, might link to European geography

[...]
The path from feminism-related topics to a small Austrian town is not obvious. Let me think about the best strategy:

1. "History" is very broad and could link to many geographic and historical topics
2. "Latin" might connect to European history/geography

CONSIDER OPTIONS

I think "History" (#25) is probably the best choice as it's the most general topic that could eventually lead to geographical articles.
Let me reconsider - from History, I might find links to European history, then to specific countries, then to specific places.
Actually, let me think more carefully. The most efficient path would be through broader topics. "History" seems like a reasonable choice as it's a very general topic that connects to many other articles.
My choice is #25 (History).

COMMIT TO DECISION

Figure 17. Claude Opus 4.5 follows the same response pattern and overall strategy as Gemini 3 Pro.

Example 7: Claude Opus 4.5, Luminogram → Ionides, optimal path length = 3, first step

Looking at the options:
- Photography (1) is a broad topic that could lead to many other articles
- Angiography (21) is medical imaging - could potentially lead to scientific/medical terms
- The other options are mostly specific photography techniques

[...]

To get to the Ionides family, I might need to go through:
- Art-related pages
- British history
- Greek diaspora
- Patronage of arts

PLANNING AHEAD

From the current options, "Photography" (1) seems like the broadest option that could lead to more general topics, potentially including art history or notable collectors.
Alternatively, "Pictorialism" (16) is an art movement in photography that might connect to art history.
[...]
I think Photography (1) is probably the best choice as it's a very broad article that could link to many other topics including art, history, technology, and potentially to people and families involved in the art world.
37

Figure 18. This example trace showcases Claude Opus 4.5 planning ahead.

Example 8: Claude Opus 4.5 Pierrefitte-sur-Aire → Fibrin, first step

I need to find a path from "Pierrefitte-sur-Aire" to "Fibrin". Fibrin is a protein involved in blood clotting.

Looking at the current page, I'm on a page about a French commune. The links are mostly to other French communes and locations. To get to "Fibrin", I need to find a path that leads me to more general/scientific topics.

The best strategy would be to click on a link that leads to a larger, more general article that might have links to broader topics like biology, medicine, or science.

NAVIGATING TO A HUB

Looking at the options:
- Most are small French communes
- "France" (27) is a major country article that would have many links to various topics
- "Verdun" (20) is a notable city known for WWI battle
- "Bar-le-Duc" (45) is a larger city

I think "France" would be the best choice as it's a major country article with links to many diverse topics including science, medicine, and biology which could eventually lead to Fibrin.

Let me go with France (27) as it's the most likely to have diverse links that could eventually lead to medical/biological topics.

Figure 19. Claude Opus 4.5 also navigates to hub pages to traverse different topics.

Example 9: Claude Opus 4.5, Megalocaria → Rojhuwas, optimal path length = 8

Step 15 (Jhajjar → Beri):
Beri is my best bet since it's a local town in Jhajjar district that could connect to smaller surrounding villages like Rojhuwas.

Step 16 (Beri → Beri):
The options given are:
0: Berry - This would be about the fruit, which would take me away from Indian geography
1: Beri - This would be a link to the same or similar page about Beri (possibly a disambiguation or the actual town page)

"Beri" would keep me in the right geographical area, while "Berry" would take me to an unrelated topic about fruit.
I should click on "Beri" (option 1) to continue exploring links related to Beri, Haryana, which might have connections to Rojhuwas.

Step 17 (Beri → Beri):
However, clicking "Beri" when I'm already on "Beri" seems like it would just keep me on the same page or go to a disambiguation. "Berry" would take me completely off track.
Given the limited options, I should try "Beri" (option 1) to see if it leads to a disambiguation or related page that might have more useful links to navigate toward Rojhuwas.
Actually, looking at this more carefully - both options seem problematic. "Berry" is clearly wrong. "Beri" might just loop back. But I have to choose one, and "Beri" is the only reasonable option to stay in the geographic context I need.

Step 18 (Beri → Beri):
Since Berry (the fruit) is unlikely to lead me to an Indian village called Rojhuwas, I should try Beri (option 1) again, hoping it might be a disambiguation page with more options.

Wait, I've already been to Beri 3 times according to the history. This seems like a loop. But between Berry (a fruit) and Beri (possibly a place in Haryana), Beri is more likely to have connections to Rojhuwas, which appears to be in the same region.

Step 19 (Beri → Berry):
The only other option is "Berry" which would take me to the fruit/botanical topic, which is unlikely to help reach Rojhuwas. However, since clicking "Beri" has kept me in a loop, I might as well try "Berry" to see if it leads somewhere different, even though it's unlikely to be helpful.
[Fails to solve task within the step limit]

Figure 20. The model is stuck on the page Beri for several steps, despite recognising the loop. After looping three times, it adapts its decision but still fails to complete the task within in the step limit.

K. Example traces - Contrasting Gemini 3 and Claude Opus 4.5

To gain further insights into the performance gap between Gemini 3 and the second best Claude Opus 4.5 we now contrast the two models on a task solved by Gemini and failed by Claude. Comparing Gemini 3 in Example 10 (Figure 21) with Claude 4.5 Opus in Example 11 (Figure 22) shows that a lack of accurate world knowledge (in this case the location of a town in Croatia), which Claude mistakes to be in Serbia, causes it to not solve the task.

Example 10: Gemini 3, Vielsalm → Sječevac

Initial Progress (Step 1): Vielsalm → Belgium

Step 2 (Belgium → Croatia):
answer: 20

thinking: The Logical Path to Sječevac

Okay, here's the deal. I'm starting at "Belgium" and need to get to "Sječevac." First things first, "Sječevac" sounds Slavic, probably Balkan. A quick search confirms it's a village in Croatia, specifically in Karlovac County.

So, the optimal path is Belgium to something related to Croatia, then to Croatia, and ultimately to Sječevac. Let's see what's on the table.

Looking at the links available, "Croatia" (option 20) jumps out. That's the obvious choice. Serbia, Montenegro, and Slovenia are nearby, but they're not Croatia. "Europe" is far too broad. Since Sječevac is a village in Croatia, the "Croatia" page is definitely my best bet. It will likely contain information on administrative divisions, like counties and municipalities, which will get me closer to my target.

Therefore, my choice is 20 - Croatia.

Remaining steps to completion:
Croatia → Zagreb → Samobor → Sječevac

Figure 21. Gemini 3 successfully navigates this tasks.

Example 11: Claude Opus 4.5, Vielsalm → Sječevac

Initial Progress (Step 1): Vielsalm → Belgium

Step 2 (Belgium → Serbia): Looking at the target "Sječevac," this is a village in Serbia. The most direct path would be to click on Serbia, which will give me access to Serbian geography and locations.

31: Serbia

reasoning: I need to find a path from "Belgium" to "Sječevac". Let me think about what Sječevac is.

Sječevac is a village in Serbia. It's located in the Brus municipality, in the Rasina District of Serbia.

Looking at the available links, I should click on a link that would eventually lead me to Sječevac. Since Sječevac is in Serbia, the most direct path would be to click on "Serbia" (link 31).

From Serbia, I should be able to find links to Serbian villages, municipalities, or geographic regions that would eventually lead to Sječevac.

Let me choose Serbia (31) as the next step.

Steps 3-14:
Serbia→ Kragujevac → Kraljevo → Sjenica → Prokuplje → Kruševac → Lajkovac → Trstenik → ... → Croatia

Step 15 (Croatia → Serbia):
Looking at the target "Sječevac," this is a village in Serbia. I need to get back to Serbia to navigate toward this destination.

From the available options, the most direct path would be:

1

This will take me to Serbia, from where I can navigate to the specific region where Sječevac is located.
reasoning: I need to get from "Croatia" to "Sječevac". Let me think about what Sječevac is and how to get there.

Sječevac is a village in Serbia. Looking at my previous path, I was visiting various Serbian cities like Kragujevac, Kraljevo, Sjenica, Prokuplje, Kruševac, Lajkovac, Trstenik before somehow ending up at Korčula and Croatia.

To get to Sječevac, I should probably go back to Serbia. Looking at the options:
- 1: Serbia - This would take me directly to Serbia, which makes sense since Sječevac is in Serbia.

Let me check if there are any more direct options... Sječevac is in Serbia, specifically I believe it's associated with the Trstenik municipality or nearby areas in central Serbia.

Given the available options, going to Serbia (option 1) seems like the most logical choice to get closer to Sječevac.

Actually, let me reconsider. From Serbia, I would need to navigate to a more specific location. Sječevac might be in the Kruševac or Trstenik area. But I don't see those specific options here.

The best choice from the current options appears to be:
1: Serbia - as Sječevac is a village in Serbia

This would get me back to Serbia where I can then try to navigate to the specific region where Sječevac is located.

[Fails to solve task within the step limit]

Figure 22. Claude 4.5 Opus fails due to incorrect world knowledge and the inability to adapt.