Magic Tower: A Novel Benchmark for Evaluating Large Language Models in Strategic Game Playing

Xiangxiang Weng
1155211173@link.cuhk.edu.hk

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities in various domains, but their performance in strategic decision-making and long-term planning remains an open question. This paper introduces Magic Tower, a novel benchmark for evaluating LLMs in the context of a single-player, turn-based roguelike RPG game. Magic Tower presents players with complex strategic decisions involving resource management, risk assessment, and long-term planning across multiple game floors with escalating difficulty. We evaluate various LLM models including open-source models (Qwen3-30B), Google's Gemini family (Gemini 2.5 Flash-lite, Gemini 2.5 Flash, Gemini 2.5 Pro), and OpenAI's models (o3-mini, gpt-4o-mini) on this benchmark. Our results reveal significant performance gaps between different model families and highlight the challenges LLMs face in strategic game playing, particularly in balancing immediate rewards with long-term objectives. In addition, we run multiple rounds of gameplay with models and observe that win rates improve across games, with different models exhibiting varying degrees of learning ability. This suggests that the benchmark can also serve as a testbed for evaluating LLMs' capacity to adapt to and learn from repeated interactions with an environment. The Magic Tower benchmark provides a standardized framework for assessing LLM capabilities in strategic decision-making and offers insights into the current limitations and potential improvements for AI systems in complex, multi-step planning scenarios. Codes can be found at: https://github.com/xx-Weng/Magic_Tower

1 Introduction

The rapid advancement of Large Language Models (LLMs) has revolutionized artificial intelligence, enabling systems to understand and generate human-like text across diverse domains. However, evaluating LLMs' capabilities in strategic decision-making and long-term planning remains a significant challenge. Traditional benchmarks such as TextGames [1] often focus on language understanding, reasoning, or specific task performance, but fail to capture the complex interplay of multiple decision points, resource management, and strategic thinking required in real-world scenarios.

Game environments have long been recognized as valuable testing grounds for AI systems, providing controlled yet complex scenarios that require strategic thinking, planning, and decision-making under uncertainty. However, most existing game-based benchmarks for LLMs focus on text-based adventures or simple decision trees, lacking the strategic depth and resource management complexity found in more sophisticated gaming environments [2].

This paper introduces Magic Tower, a novel benchmark designed specifically to evaluate LLMs' capabilities in strategic game playing. Magic Tower is a single-player, turn-based roguelike RPG that presents players with a series of interconnected decisions involving resource management, risk assessment, and long-term planning. The game's structure, with multiple floors of escalating difficulty, provides a natural progression that tests both immediate tactical decisions and long-term strategic planning.

2 Game Description

2.1 Game Overview

Magic Tower is a single-player, turn-based RPG where the player plays an adventurer trapped in a multi-level tower. The primary objective is to reach the final floor by strategically minimizing actions, ultimately defeating a powerful final Boss to achieve escape. A comprehensive game guide is presented at the start of the game and can be accessed at any time by entering 'g'.

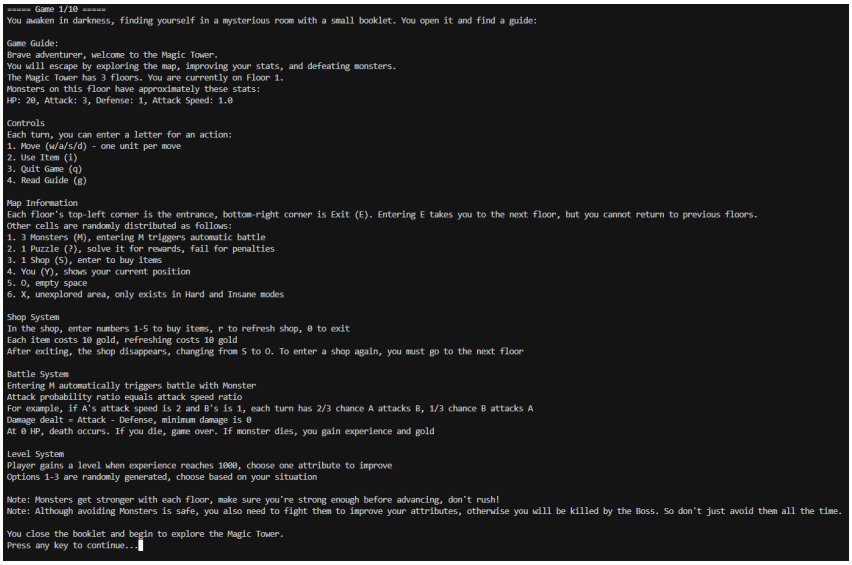

Figure 1: Guide

The core gameplay loop involves strategic decision-making about when to fight monsters (for experience and resources), when to explore, and how to allocate resources to improve character capabilities. Each step of the user interface does not include the Guide, but it does contain some necessary information and a map.

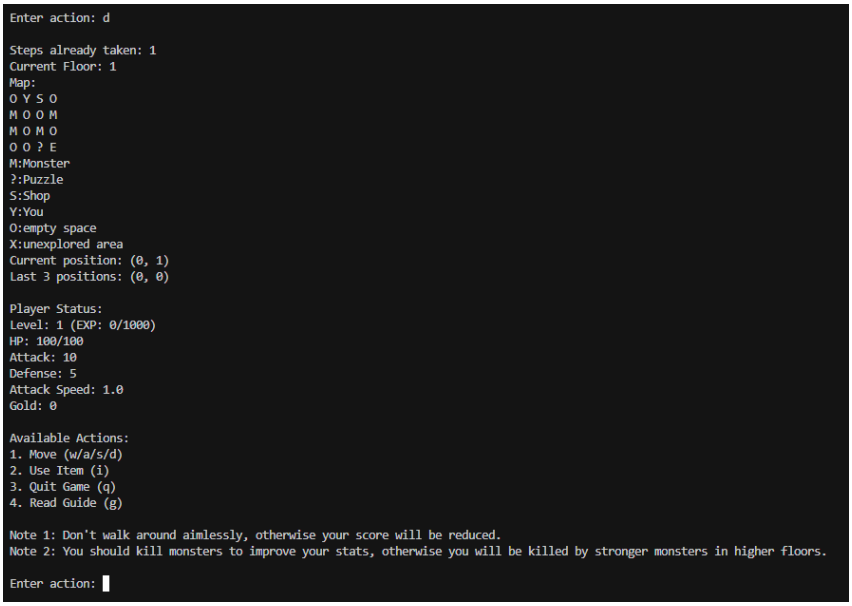

Figure 2: User Interface

2.2 Game Mechanics

The gameplay involves moving across a grid-based map, encountering various elements, including monsters, puzzles, and shops. Combat occurs automatically upon entering a cell occupied by a monster, governed by a probabilistic attack system based on each entity's attack speed, with damage outcomes determined by attack and defense statistics. Successful battles yield experience points and gold, the former contributing to level progression, which in turn allows the player to enhance specific attributes. The shops offer items that can be purchased with gold. Puzzles offer rewards or penalties depending on the player's performance. The player must make strategic decisions regarding combat engagement, resource management, and stat development, as monsters become progressively stronger with each floor. Movement is controlled via directional inputs "w/a/s/d", and the game encourages a balanced approach between exploration, combat, and character growth, cautioning against both excessive evasion and premature advancement.

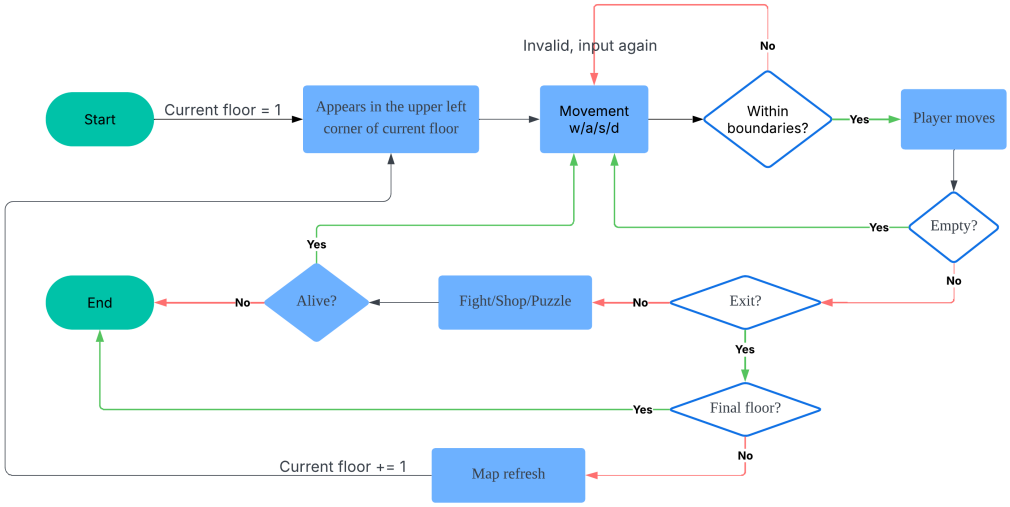

Figure 3: Flowchart

Character System: The character system is defined by four primary attributes: health points (HP), attack, defense, and attack speed. HP reflects both current and maximum health. Attack determines damage output, defense mitigates incoming damage, and attack speed influences the frequency of attacks during combat.

Progression System: Progression is achieved through the accumulation of experience points, primarily gained from defeating monsters and solving puzzles. Upon reaching 1000 experience points, the player levels up and selects one of three randomly generated stat enhancement options, allowing for tailored character development.

Resource Management: Resource management constitutes a critical component of gameplay. Gold serves as the in-game currency, used for purchasing items and refreshing shop inventories. Items are consumables, such as health potions that restore 50% of the character's maximum HP. Experience functions as both a progression mechanism and a strategic resource for long-term character growth.

2.3 Strategic Decision Points

The game presents several key strategic decision points that test LLM capabilities:

Combat Decisions: Every 5 turns during combat, players must decide whether to continue fighting or attempt to escape. This decision requires evaluating current health, enemy strength, and calculated win probabilities.

Resource Allocation: Players must decide how to spend limited gold on items, whether to refresh shops for better options, and when to use consumable items.

Exploration Strategy: Players must balance the need to gain experience through combat with the risk of taking damage, while also managing limited movement actions.

Level-up Choices: When leveling up, players must choose from three random stat boost options, requiring strategic thinking about current needs and future challenges.

2.4 Mathematical Framework for Win Probability

To enable precise evaluation of combat decisions, we developed a mathematical framework for calculating win probabilities in battle scenarios. Consider two units A (player) and B (monster) with attributes $(hp_A, attack_A, defense_A, speed_A)$ and $(hp_B, attack_B, defense_B, speed_B)$ respectively.

The damage from A to B is calculated as:

$$damage_{A \to B} = \max(0, attack_A - defense_B) \tag{1}$$

The probability of A attacking B in each round is:

$$p = \frac{speed_A}{speed_A + speed_B} \tag{2}$$

To win, A needs to attack B n_a times, where:

$$n_a = \left\lceil \frac{hp_B}{damage_{A \to B}} \right\rceil \tag{3}$$

And A must survive being attacked by B less than n_b times, where:

$$n_b = \left\lceil \frac{hp_A}{damage_{B \to A}} \right\rceil \tag{4}$$

The win probability for A is calculated as the sum of negative binomial distribution probabilities:

$$P(A \text{ wins}) = \sum_{k=0}^{n_b - 1} \binom{n_a + k - 1}{k} p^{n_a} (1-p)^k \tag{5}$$

This mathematical framework provides a precise way to evaluate the quality of LLM decisions during combat scenarios.

3 Experimental Setup

3.1 Model Selection

We evaluate a diverse set of LLM models to provide comprehensive coverage of current capabilities:

Open-source Models:

- Qwen3-30B: A large-scale open-source model with strong reasoning capabilities

Google Gemini Family:

- Gemini 2.5 Flash-lite: Lightweight model for fast inference
- Gemini 2.5 Flash: Balanced model for general use
- Gemini 2.5 Pro: High-performance model for complex tasks

OpenAI Models:

- o3-mini: Efficient model for cost-effective evaluation
- gpt-4o-mini-2024-07-18: Optimized mini model

3.2 Evaluation Protocol

Each model is evaluated across multiple difficulty levels (Easy, Normal, Hard, Insane) with the following metrics:

Win Rate: Percentage of games completed successfully (reaching the final Boss and winning).

Quit Rate: The probability of the model choosing to quit during the game.

Step Efficiency: Average number of steps taken to complete the game.

Combat Decision Quality: Analysis of combat decisions using the win probability framework.

3.3 Experimental Conditions

We conduct experiments under two conditions:

No Show Battle Data: Models play the game with normal visibility and information access.

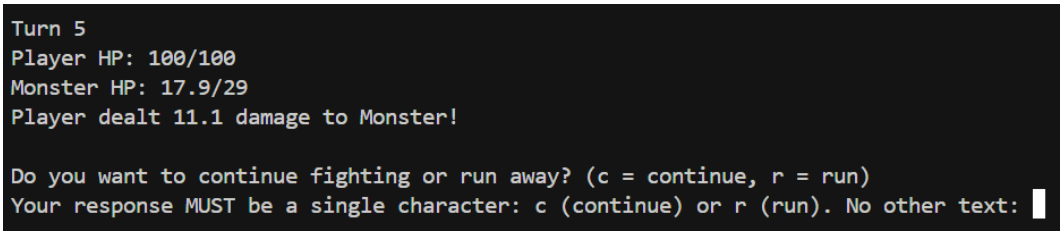

Figure 4: No Show Battle Data

Show Battle Data: Models receive additional battle statistics every 5 turns during combat.

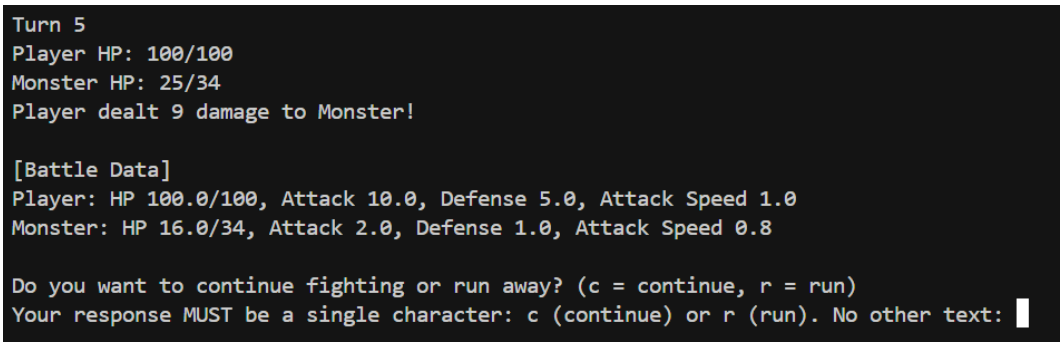

Figure 5: Show Battle Data

This allows us to evaluate both the models' natural decision-making capabilities and their ability to utilize additional quantitative information when available.

4 Experimental Evaluation

4.1 Overall Performance Comparison

A total of 532 game records were collected. Our evaluation reveals significant performance differences across model families and individual models. The results demonstrate that strategic game playing remains a challenging task for current LLM systems.

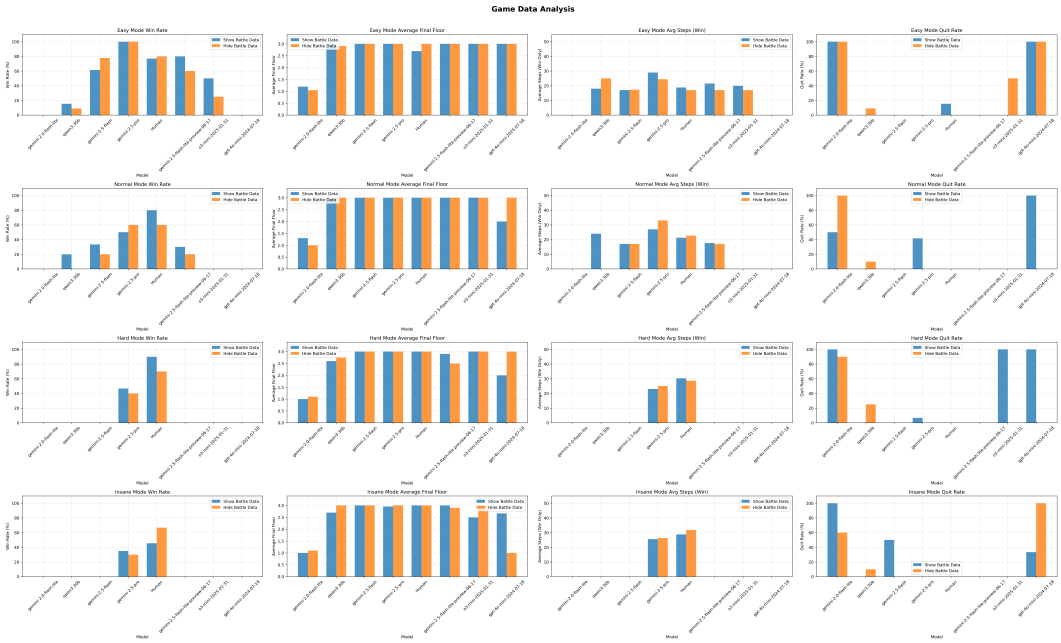

Figure 6: Overall Performance Bar Chart

Win Rate:

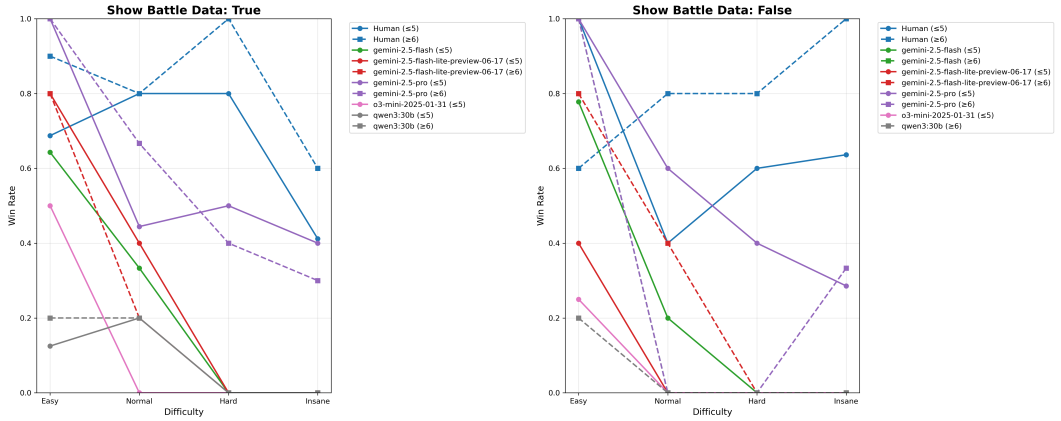

Figure 7: Win Rate Analysis

From these results we see that human players consistently maintain high win rates (around 60–100% across Easy to Hard and still above 40% on Insane) whether or not detailed battle data is shown, whereas all large-language models suffer steep performance drops as difficulty increases. Only the most capable model, Gemini-2.5-pro, manages nonzero success on Hard (40–50%) and Insane (28–40%) levels—especially when battle data is provided—while smaller or "lite" variants fall to zero wins beyond Normal difficulty. This indicates that although richer game-state information can

modestly improve LLM performance, none of the tested models yet achieves the robust gameplay resilience demonstrated by humans.

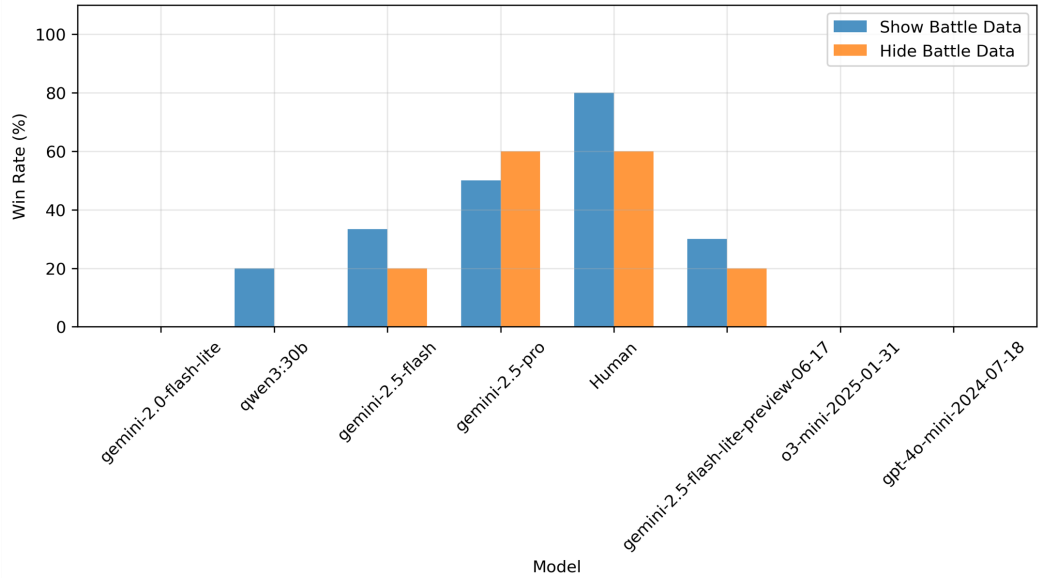

Figure 8: Normal Mode Win Rate

In the Normal-mode benchmark, only the strongest LLMs manage nonzero win rates, and showing the raw battle data generally helps them, but not always. Humans lead with an 80% win rate when data's visible (60% hidden), followed by Gemini-2.5-Pro at 50% (show) versus 60% (hide), the only model that actually does slightly better without the extra detail. Gemini-2.5-Flash and Qwen3:30B achieve more modest success, while all "lite" or earlier-generation variants remain at zero. This underscores that higher model capability, and in most cases richer state information, is required just to approach human-level performance on Normal difficulty.

Quit Rate:

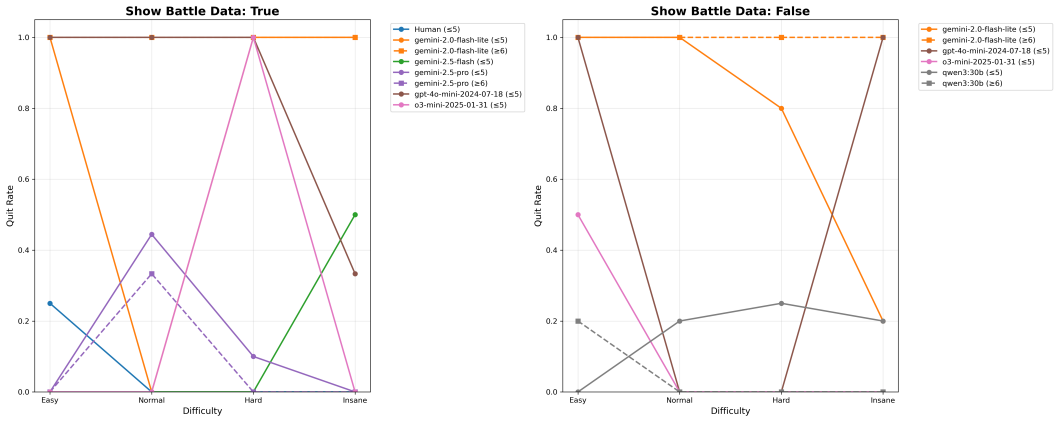

Figure 9: Quit Rate Analysis

The quit-rate curves highlight significant differences among various LLMs. The smallest "flash-lite" versions exhibit very high quit rates. Thinking models, such as Gemini-2.5-Pro, sometimes quit on Normal difficulty (33–45%) and have a 10% quit rate on Hard, but if they manage to persist on Insane difficulty, they almost never quit. In tests with hidden data, Qwen3:30B has the lowest overall quit rate, with 25% on Hard and 20% on Insane. The data suggests that even top-tier models still give up far more readily than expected.

Step Efficiency:

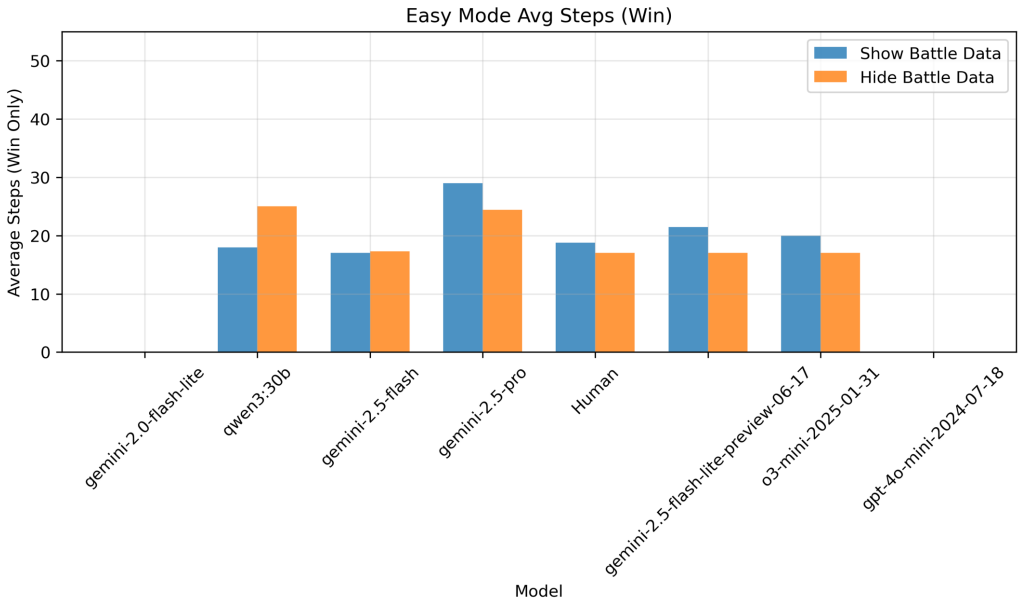

Figure 10: Easy Mode Average Steps (Win)

In Easy mode games that end in victory, most LLMs complete the task in roughly 17 to 21 steps, which is similar to the human average of 19. The exception is Gemini-2.5-Pro, which takes almost 30 steps when given full battle data, showing a much less efficient strategy. Hiding the detailed combat information reduces a few extra moves for most players and models, with humans dropping from 19 to 17 steps, O3-mini from 20 to 17, and Gemini-2.5-Pro from 29 to 24, while Qwen3:30B actually slows down from 18 to 25 steps without the extra state cues. Overall, apart from the overlong play patterns of the highest-capacity model and the sensitivity of some models to visible data, LLMs reach Easy wins with efficiency close to that of humans.

4.2 Combat Decision Analysis

Using our mathematical framework for win probability calculation, we analyze the quality of combat decisions made by different models:

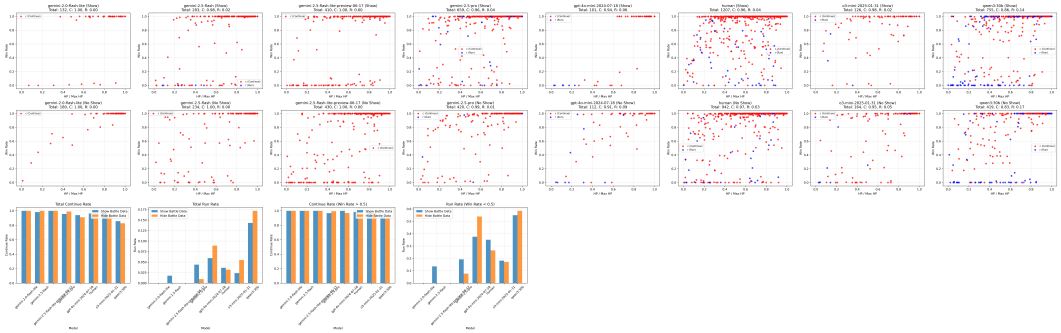

Figure 11: Overall Combat Decision Scatter & Bar Chart

We calculate the model's run rate when the win rate is below 50% (which means these runs are "good runs"). Note that the win rate data is not provided to the model; they can only determine whether to run based on the current state and information. The run-rate plot shows a clear correlation between model capability and the tendency to run. The smallest "flash-lite" variants never choose to run, while

larger models such as Gemini-2.5-Flash and O3-mini retreat only 13 to 18 percent of the time. In contrast, GPT-4o-mini and Qwen3:30B, run away in more than one-third to more than one-half of all battles. Hiding the detailed battle statistics affects models in different ways. Gemini-2.5-Pro's run rate actually decreases from about 19 percent with data to under 8 percent without, suggesting it relies on explicit metrics to recognize hopeless fights. GPT-4o-mini and Qwen3:30B, however, become even more likely to retreat when deprived of that information, rising from around 38 percent to 54 percent and from about 55 percent to 58 percent, respectively. Human players also flee a little less when numeric readouts are hidden, dropping from roughly 35 percent to about 27 percent. Overall, these patterns show that higher-capacity LLMs actively use the run option as a loss-minimization strategy but may rely too heavily on raw state data to trigger that choice.

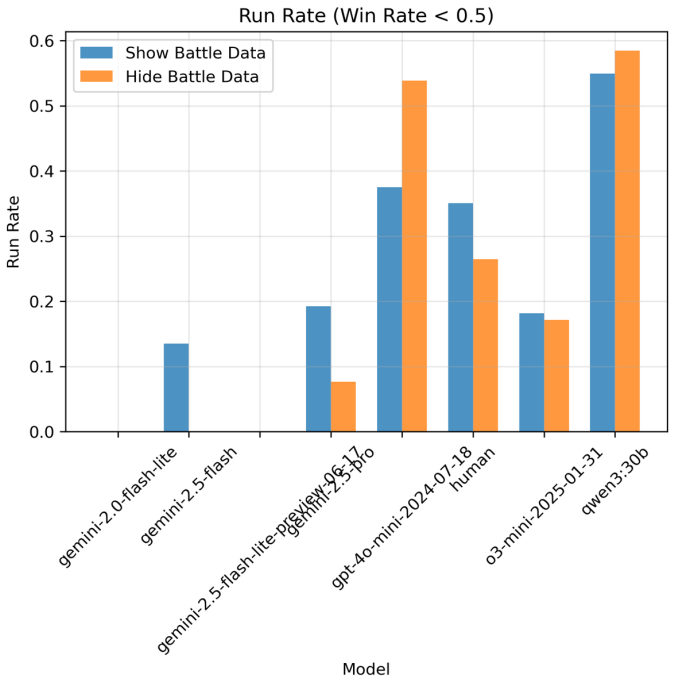

Figure 12: Run Rate (Win Rate < 0.5)

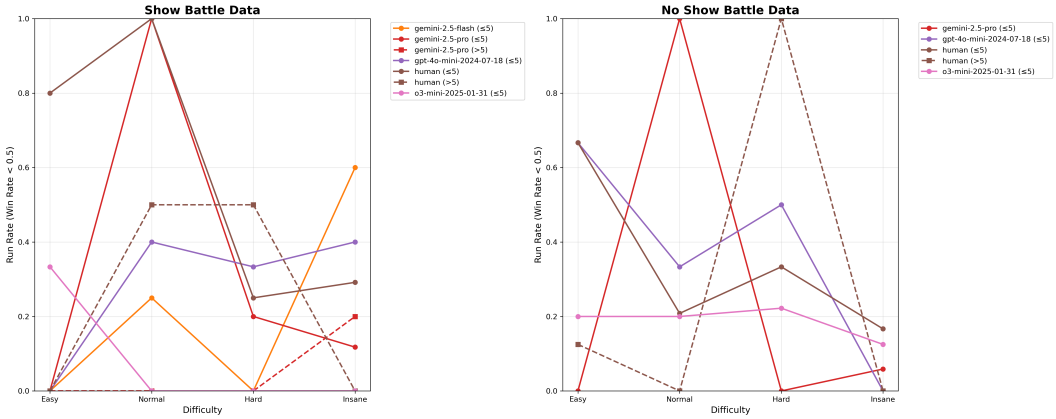

Figure 13: Run Rate (Win Rate < 0.5)

4.3 Discussion

Our experiments show that even the strongest language models struggle with long-term strategic planning. Unlike humans, who maintain high win rates across all difficulties, models quickly lose

effectiveness when multiple moves must be coordinated. Their performance drops sharply on harder modes, and many runs end early because quitting or repeatedly choosing "run" becomes the default response rather than part of a broader strategy.

The results also reveal problems with probabilistic reasoning. When key battle statistics are hidden, some models become overly cautious, while others turn more reckless when the same data is visible. Neither behavior resembles the balanced, probability-aware decisions made by human players, who generally continue playing unless defeat is certain.

Another weakness is the lack of exploration. Instead of trying new tactics when a plan fails, models tend to fall back on the same limited options. This is especially clear in medium-sized models, which avoid quitting but rarely recover from poor positions, leading to average performance without genuine strategic adaptation.

Scaling up model size improves short-term tactics and overall win rates, but it does not solve the deeper issues. Larger models still waste many steps on simple tasks and behave inconsistently when information is incomplete. This suggests that while bigger architectures help, they are not enough to reach human-level strategic competence. Closing this gap will require more progress in multi-step planning, handling uncertainty, and developing mechanisms that encourage flexible, dynamic strategies.

5 Conclusion

The Magic Tower benchmark is a new framework for testing how well large language models can play strategic games. It focuses on multi-step decision-making and highlights the challenges models face in long-term planning. Unlike other benchmarks, Magic Tower uses a multi-floor progression system to test strategy over time, includes resource systems that require models to optimize limited assets, and provides a mathematical way to measure decision quality. It also allows evaluation across multiple dimensions of AI performance.

Experiments show that while language models perform strongly in many areas, they still struggle with strategic game playing. Different models show clear performance gaps, suggesting there is significant room for improvement.

Future research should work on better training for strategic reasoning, improving the balance between short-term and long-term goals, strengthening probabilistic reasoning and risk assessment, and developing architectures that can handle complex planning.

The benchmark can be expanded with features like skill trees, larger boards, and more levels to keep testing new models. In addition, this study did not include statistics on resource management and decision-making during the upgrading of large language models; future work can incorporate this data.

Magic Tower provides a consistent way to measure progress in strategic decision-making and helps identify both the current limits and future possibilities for large language models in this domain.

References

[1] F. Hudi, G. I. Winata, R. Zhang, and A. F. Aji, "Textgames: Learning to self-play text-based puzzle games via language model reasoning," arXiv preprint arXiv:2502.18431, 2025.

[2] O. Topsakal, C. J. Edell, and J. B. Harper, "Evaluating large language models with grid-based game competitions: an extensible LLM benchmark and leaderboard," arXiv preprint arXiv:2407.07796, 2024.