Cards Against Contamination: TCG-Bench for Difficulty-Scalable Multilingual LLM Reasoning

Sultan Alrashed [1] **Jianghui Wang** [1] **Francesco Orabona** [1]

Abstract

Benchmarks for language models have become essential tools for research. Yet, such benchmarks face a persistent contamination problem, with recent studies finding 25-50% of evaluation datasets appearing in the training corpora. This is true even looking at the two-player zero-sum game setting, where most of the benchmarks are based on popular games, like chess, whose optimal strategies all over the web. Such contamination hinders the possibility to differentiate memorization and reasoning skills. To rectify these problems, we introduce TCG-Bench, a benchmark based on a new two-player trading card game (TCG), which is similar in spirit to games like Magic the Gathering. TCG-Bench offers three key innovations: (1) a contamination-resistant design, by separating the publicly released game engine[1] from the hidden card implementations, (2) a continuous difficulty spectrum via Monte Carlo simulation that prevents benchmark saturation, and (3) a parallel implementation in English and Arabic, being the first multilingual text-based game benchmark to do so. Our analysis across 17 models (42,750+ games) reveals that performance declines exponentially with difficulty, while model size correlates only weakly with strategic ability. We also observe cross-linguistic performance gaps between English and Arabic, with a gap of 47.4% at 32B, highlighting the need for multilingual game benchmarks that target reasoning capabilities in the target language. We host a leaderboard[2] showcasing these results and welcome evaluation requests on our private cards.

[1]King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Kingdom of Saudi Arabia. {firstname.lastname}@kaust.edu.sa. Correspondence to: Sultan Alrashed <sultan.alrashed@kaust.edu.sa>.

ICML 2025 Workshop on Assessing World Models.

[1]https://github.com/AdaMLLab/tcg-bench

[2]http://tcg-bench.com/

1. Introduction

Modern research on language models critically depend on the availability of high quality benchmarks to measure language model capabilities. However, the reliability of language model evaluation has become increasingly uncertain as more benchmark data appears in training corpora. Recent analyses reveal concerning levels of contamination: 29.1% for MMLU (Deng et al., 2024), 28.7% for ARC-Challenge (Deng et al., 2024), 25% for HumanEval (Yang et al., 2023), and over 50% for TriviaQA (Madaan et al., 2024). This contamination significantly distorts performance metrics, with models scoring up to 44.5 percentage points higher on contaminated subsets (Yang et al., 2023; OpenAI, 2023). Perhaps most concerning, models like GPT-4 can identify the correct answer in MMLU even when options are masked (Deng et al., 2024), indicating memorization rather than reasoning.

As language models increasingly incorporate web-scale data, traditional approaches to benchmark protection are showing their limitations. Time-separation strategies quickly become obsolete as training data cutoffs advance. Dynamic benchmarks that continuously generate new questions introduce variability and require substantial maintenance. Private evaluation suites limit broader scientific participation, while synthetic data often fails to match the complexity and nuance of real-world tasks. These challenges invite a different approach to benchmark design, one that ensures evaluation integrity without relying on complete secrecy.

We present TCG-Bench, a trading card game benchmark that addresses contamination through a held-out set, while also addressing many other issues in this space. In detail, our main contributions are

1. **Contamination Resistant Benchmark:** TCG-Bench prevents contamination through structural separation of the game mechanics from their implementation. We host a public leaderboard website (see Appendix A) where researchers can submit their models via HuggingFace URLs for evaluation on our holdout card set and concealed game mechanics, while simultaneously open-sourcing our high-throughput asynchronous benchmarking engine to welcome further

contributions and custom leaderboards. This approach preserves evaluation integrity while enabling broad community participation.

2. **Tunable Difficulty System:** We implement a difficulty scaling system using Monte Carlo simulation with a configurable rollout depth. This creates a continuous spectrum of opponent strength, with win rates declining exponentially from 35.8% at rollout-1 to 2.5% at rollout-1000. This ensures that the benchmark remains challenging even as model capabilities advance, while still allowing smaller models to compete, addressing the saturation problem that has diminished the utility of many established benchmarks.

3. **Bilingual Parallel Implementation:** TCG-Bench provides the first text-based game benchmark with identical parallel implementations in English and Arabic. Unlike traditional benchmarks that translate existing static datasets, our bilingual design creates identical dynamic evaluation environments across languages, enabling direct comparison of strategic reasoning capabilities.

Moreover, our evaluation of Large Language Models (LLMs) on TCG-Bench reveals several insights that challenge common assumptions about language model capabilities. The correlation between parameter count and strategic performance is surprisingly weak ($r = 0.31$), with smaller models sometimes significantly outperforming larger counterparts from the same family. This suggests that architectural design may be more important than scale for certain reasoning tasks, which is further empirically validated through seeing smaller dense models outperform their larger mixture-of-experts counterparts. We also see significant degradation of performance when using Arabic, which cements the need for more multilingual benchmarks that target strategic reasoning and long-horizon planning.

2. Related Work

Mitigating Benchmark Contamination. Early alarms about data leakage in GLUE and SuperGLUE prompted post-hoc filtering and dynamic datasets (Jacovi et al., 2023; Deng et al., 2024). More recent proposals include LiveBench (White et al., 2025) and LatestEval (Li et al., 2024), which refresh items over time, and survey efforts that catalogue mitigation strategies. TCG-Bench differs by enforcing isolation through an engine/content split, eliminating the need for continual item renewal, which would limit the complexity of the game and could introduce variability. ARC-AGI (Chollet et al., 2024) is a benchmark with a fixed pool of puzzles that test reasoning abilities. However, it faces contamination risks if puzzles are leaked, and adding new puzzles is costly and requires rerunning the entire benchmark since each puzzle represents a unique task.

Game-Based Evaluation of Language Models. Text-adventure frameworks such as Jericho (Cui et al., 2025) and TextWorld (Côté et al., 2018) both test language understanding and planning, but their public game files remain vulnerable to memorization. Indeed, we detected both of them in C4 (Raffel et al., 2020). Other works instrument existing commercial games, yet winning strategies are openly documented. Recent surveys (Yang et al., 2024) show increasing interest in game-based evaluation, with platforms like GameBench (Costarelli et al., 2024) using multiple existing games to test strategic reasoning, for which the optimal strategies have been present on the web and scraped into training data. TextArena (Guertler et al., 2025) benchmarks agentic reasoning through competitive text-based games, showing that game environments can effectively assess complex decision-making capabilities. However, like previous approaches, TextArena does not specifically address contamination concerns or cross-linguistic evaluation. TCG-Bench is the first purpose-built game designed for benchmarking LLMs on two-player zero-sum settings using only text, with an architecture specifically created to prevent contamination.

Multilingual Reasoning Benchmarks. Recent benchmarks for multilingual reasoning include mCSQA, a unified commonsense QA dataset built via a human–model pipeline across eight languages (Sakai et al., 2024), MLissard, which evaluates sequential reasoning and length-sensitive tasks in multiple languages with controlled complexity (Bueno et al., 2024), mCoT, a large-scale chain-of-thought math-reasoning dataset spanning eleven typologically diverse languages (Lai & Nissim, 2024), and M4U, a multimodal benchmark assessing scientific understanding and reasoning in three languages (Wang et al., 2024). While these efforts illuminate cross-lingual inference and reasoning consistency, they fail to capture the zero-sum settings, where one has to synthesize partial information, anticipate opponents' moves, and commit to multi-turn strategies whose payoff may be delayed by many steps. We address this gap by framing evaluation as a bilingual TCG with hidden information and tunable opponent strength, thereby directly assessing understanding and adaptive planning under competitive pressure.

3. Methodology

Game Design. TCG-Bench is a two-player trading card game where players begin with 10 life points and compete to reduce their opponent's life points to zero through strategic card play. The game features three card types with distinct strategic roles: Champions (persistent attackers), Spells (single-use effects), and Tricks (conditional abilities). We include three cards to showcase the code base, see

Appendix C.

As in similar TCG, each turn consists of three phases: Draw (adding one card to hand from a shuffled deck), Main (playing one card), and Combat (attacking with eligible Champions). The game's strategic depth comes from balancing immediate impact against long-term advantage, managing limited resources, and anticipating opponent actions.

With approximately 10^{14} possible game states, a branching factor of 3-4 legal moves per turn, and 10-15 turns per game on average, TCG-Bench offers complexity comparable to intermediate classic games like Connect Four, while adding hidden information and non-deterministic elements that prevent memorized play sequences.

Architecture-Based Contamination Prevention. TCG-Bench prevents contamination through an architectural separation:

1. **Public Game Engine**: The core mechanics and rules are open-sourced, including turn structure, card types, and victory conditions. This transparency allows models to learn general game concepts and for people to potentially train on the environment itself with custom cards.

2. **Private Card Implementations**: 30 specific card effects, abilities, and interactions remain private. We host a public leaderboard website where researchers can submit their models via HuggingFace URLs for evaluation on our holdout card set. This approach preserves evaluation integrity while enabling broad participation, models cannot memorize test scenarios because they never see the underlying implementation.

To ensure that the cards are not leaked, we only preserve local copies at our institute alongside backups. Through this air-gapped approach where the cards are kept securely private, we can ensure that contamination risk is kept well below comparable benchmarks. That said, thanks to the structure of the game, a leak of the cards would not significantly hinder this benchmark. It would be cheap to refresh the cards with a new set post-contamination, making the benchmark always private.

Monte Carlo Evaluation Framework. TCG-Bench employs a Monte Carlo simulation approach to create a scalable difficulty spectrum:

$$score(m) = \frac{1}{k} \sum_{i=1}^{k} outcome(s_i), \tag{1}$$

where m is a move, k is the rollout count, and s_i is a simulation from the resulting state to game completion using random plays. The Monte Carlo opponent evaluates each legal move by running k random simulations and selecting the move with the highest win rate.

By adjusting k from 1 to 1000+, we create a continuous difficulty spectrum from near-random play to increasingly optimal decisions. This approach differs from typical Monte Carlo Tree Search (Chaslot et al., 2008) by using simple random rollouts without tree building, prioritizing efficiency and parallelism over maximum playing strength.

Language models face this configurable opponent through a text interface, receiving the game state description and outputting their chosen move. The experimental setup compares multiple difficulty levels, measuring how performance decreases as opponent strength increases, creating a benchmark that remains challenging as language model capabilities improve.

Bilingual Implementation. TCG-Bench features parallel implementations in English and Arabic, with attention to linguistic and cultural appropriateness. All prompts alongside the cards and rules have been translated to Arabic. Also, the code base was designed with the possibility of adding even more languages.

4. Experiments

We evaluated 17 language models across 6 difficulty levels (rollouts: 1, 2, 5, 10, 100, 1000) and 2 languages (English and Arabic), totaling 42,750 games. Models included commercial services and open weights models. Each model played 600 games per difficulty level to ensure statistical significance, given our compute constraint. We provide an in-depth analysis of the results in Appendix B.

For each game, the LLM agent received a text description of all cards and game rules. Then, in each round it receives the current game state, including life points, board state, and available cards. Models generated moves using standard prompting with format-constrained outputs (card names between <BEGIN_MOVE> and <END_MOVE> tags). For cross-linguistic evaluation, we used identical game mechanics with language-appropriate prompts and card descriptions.

We collected win rates as our primary performance metric, along with decision times, token usage, and card type distributions. We include decision time to capture each model's computational efficiency, its latency to performance trade-off, which is critical for real-time or "thinking" agents. We provide the results in Appendix B.6

4.1. Results

Hard to Saturate Difficulty Scaling: Table 1 demonstrates TCG-Bench's difficulty of saturating, with win rates declining exponentially as rollout count increases. From rollout-1 to rollout-1000, mean win rates drop from 35.8% to just

Table 1. Performance Scaling Data Table.

MC Rollouts	Mean Win Rate	Standard Deviation	R^2 (Exponential Fit)
1	35.8%	9.2%	0.97
2	24.3%	7.8%	0.96
5	13.1%	5.4%	0.95
10	10.7%	4.1%	0.95
100	5.2%	2.3%	0.94
1000	2.5%	1.2%	0.93

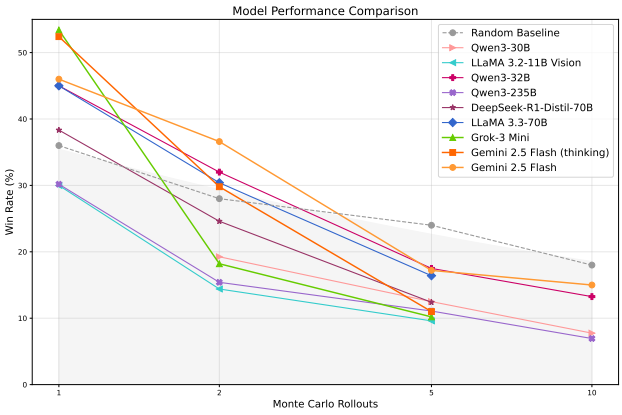

Figure 1. Performance decay across difficulty levels demonstrates TCG-Bench's difficulty, with all models showing exponential decline in win rates as opponent strength increases.

2.5%, with exponential decay curves fitting with high statistical confidence (R^2 ¿ 0.93). Figure 1 visualizes this decay pattern, which persists across all model families, indicating that the benchmark will remain challenging even as models improve. The strongest models achieve only 2-3% win rates at rollout-1000, leaving substantial headroom for future improvement. This exponential decay pattern ensures TCG-Bench will maintain discriminative power despite advances in model capabilities.

We can also notice that some models experience a faster decline in performance compared to the random baseline, which does not rely on long-term planning. This suggests that deeper planning is challenging for many current models. This weakness is magnified as the Monte Carlo rollout depth increases. Current literature on the topic affirms this (Duan et al., 2024), showing that large language models struggle in particular with long-horizon planning when stressed

Size-Performance Relationship: Surprisingly, we find only a weak correlation between parameter count and strategic performance (r = 0.31). Among models from the same family, we observe cases where smaller variants significantly outperform larger counterparts. For example, Qwen3-32B (Yang et al., 2025) achieves a 21.5% win rate at rollout-5, while Qwen3-235B reaches only 10.1%, despite having 7.3x more parameters. This efficiency advantage (win ratio × size ratio) challenges conventional scaling assumptions

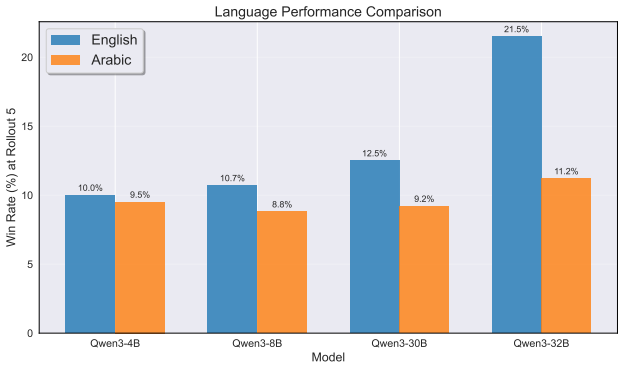

Figure 2. Cross-linguistic performance comparison for Qwen3: language gaps widen with model scale.

and suggests that model architecture (e.g., dense vs mixture-of-experts) and training approach may matter more than scale for certain reasoning tasks.

Cross-Linguistic Performance Gaps: Figure 2 shows cross-linguistic performance gaps between English and Arabic widen with model scale. The relative gap increases from just 5.0% for 4B parameter models to 47.4% for 32B models. This could be due to the smaller models being too weak to exhibit any linguistic bias, with them not passing the random baseline. Once the model reaches a sufficient capacity, the linguistic bias shows itself, where we see a significant degradation in Arabic.

Strategic Patterns: Analysis of over 40,000 game logs reveals distinct strategic signatures across model families. Gemini (Team et al., 2023) prefer Champion cards (44.8%), LLaMA models favor Spells (40.3%), and DeepSeek (Bi et al., 2024) models use more Tricks (38.3%). These preferences persist across model scales within families, suggesting biases in reasoning that transcend parameter count.

Stronger models tend to use more specialized (low-entropy) strategies, while weaker ones show near-uniform behavior, indicating that strategic focus is linked to performance and that model architecture influences decision-making beyond scale.

4.2. Analysis

It is natural to think that the noise would heavily affect the evaluation. Looking at the data however, this does not seem to be the case. In fact, the noise seems small relative to the effect sizes we observe, and our statistical treatment ensures that claimed differences are robust. More in detail, we tested hundreds of games, reporting the exact 95% confidence intervals (CI) using the optimal method in [20], see Figure 7. We report the numbers in Table 7 for additional clarity. Even considering the confidence intervals, the exponential decay

of the performance ($R^2 > 0.93$) with the number of rollouts is real.

Empirically the game averages $\approx$ 3.5 legal moves per turn (branching factor b) over $\approx$ 12 turns, yielding $b^{12} = 1.4 \times 10^6$ nodes, tractable for lightweight rollouts but far too large for exhaustive minimax. In Figure 1, we can see the performance of a uniform-random agent against our single-rollout Monte Carlo baseline (MC-1), and the random agent wins only 35% of games (MC-1 wins 65%). Modern LLM agents split matches with MC-1 roughly 50:50, and as we increase rollout depth k, MC-k's error decays roughly as b^{-k}, further widening the LLMs' margin. These trends, together with the strategies we catalogue in Section 4.1, show that success in TCG-Bench does require coherent planning beyond random play, yet the benchmark's state-space complexity remains on par with classics like Connect Four, but enriched with partial information and stochastic effects. This shows TCG-Bench is strategically non-trivial yet still computationally manageable.

5. Conclusion

We presented TCG-Bench, a trading card game benchmark that addresses the challenge of benchmark contamination by separating the game engine from the card implementations. Implementing a continuous difficulty spectrum via Monte Carlo simulation, and providing parallel English and Arabic versions, TCG-Bench offers a new approach to reliable language model evaluation. Our analysis across 17 models and over 42,000 games reveals a limited reasoning ability and it challenges conventional scaling assumptions. In the future, we plan to expand the number of languages as well as implementing stronger baselines.

Limitations

While TCG-Bench introduces several methodological innovations, it has limitations that should be addressed in future work. First, the benchmark currently supports only English and Arabic; expanding to additional languages would provide more comprehensive insights into cross-linguistic capabilities. Second, strategic performance in a trading card game may not generalize to all reasoning domains, necessitating complementary evaluation approaches.

Third, we have not yet established human performance baselines, making it difficult to contextualize model results relative to human capabilities. Fourth, the Monte Carlo opponent uses pure random simulation rather than more sophisticated planning techniques, potentially limiting its maximum strength despite high rollout counts. Fifth, although our cards are air-gapped to prevent leakage, it is not provably immune.

Finally, our leaderboard evaluation system currently focuses on win rates as the primary metric. Future work should explore multi-dimensional evaluation that captures strategic diversity, adaptability to changing game conditions, and reasoning transparency, providing a more nuanced understanding of model capabilities beyond binary outcomes.

References

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., Gao, H., Gao, K., Gao, W., Ge, R., Guan, K., Guo, D., Guo, J., Hao, G., Hao, Z., He, Y., Hu, W., Huang, P., Li, E., Li, G., Li, J., Li, Y., Li, Y. K., Liang, W., Lin, F., Liu, A. X., Liu, B., Liu, W., Liu, X., Liu, X., Liu, Y., Lu, H., Lu, S., Luo, F., Ma, S., Nie, X., Pei, T., Piao, Y., Qiu, J., Qu, H., Ren, T., Ren, Z., Ruan, C., Sha, Z., Shao, Z., Song, J., Su, X., Sun, J., Sun, Y., Tang, M., Wang, B., Wang, P., Wang, S., Wang, Y., Wang, Y., Wu, T., Wu, Y., Xie, X., Xie, Z., Xie, Z., Xiong, Y., Xu, H., Xu, R. X., Xu, Y., Yang, D., You, Y., Yu, S., Yu, X., Zhang, B., Zhang, H., Zhang, L., Zhang, L., Zhang, M., Zhang, M., Zhang, W., Zhang, Y., Zhao, C., Zhao, Y., Zhou, S., Zhou, S., Zhu, Q., and Zou, Y. DeepSeek LLM: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Bueno, M. C., Lotufo, R., and Frassetto Nogueira, R. MLissard: Multilingual long and simple sequential reasoning benchmarks. In Hupkes, D., Dankers, V., Batsuren, K., Kazemnejad, A., Christodoulopoulos, C., Giulianelli, M., and Cotterell, R. (eds.), *Proceedings of the 2nd GenBench Workshop on Generalisation (Benchmarking) in NLP*, pp. 86–95, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.genbench-1.6. URL https://aclanthology.org/2024.genbench-1.6/.

Chaslot, G., Bakkes, S., Szita, I., and Spronck, P. Monte-carlo tree search: A new framework for game AI. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 4, pp. 216–217, 2008.

Chollet, F., Knoop, M., Kamradt, G., and Landers, B. Arc prize 2024: Technical report. *arXiv preprint arXiv:2412.04604*, 2024.

Costarelli, A., Allen, M., Hauksson, R., Sodunke, G., Hariharan, S., Cheng, C., Li, W., Clymer, J., and Yadav, A. GameBench: Evaluating strategic reasoning abilities of LLM agents. *arXiv preprint arXiv:2406.06613*, 2024.

Côté, M., Kádár, Á., Yuan, X., Kybartas, B., Barnes, T., Fine, E., Moore, J., Tao, R. Y., Hausknecht, M., El Asri, L., Adada, M., Tay, W., and Trischler, A. TextWorld: A learning environment for text-based games. In *Workshop on Computer Games at the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Cui, C. Z., Yuan, X., Xiao, Z., Ammanabrolu, P., and Côté, M.-A. Tales: Text adventure learning environment suite. *arXiv preprint arXiv:2504.14128*, 2025.

Deng, C., Zhao, Y., Tang, X., Gerstein, M., and Cohan, A. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 8706–8719, 2024.

Duan, J., Zhang, R., Diffenderfer, J., Kailkhura, B., Sun, L., Stengel-Eskin, E., Bansal, M., Chen, T., and Xu, K. Gtbench: Uncovering the strategic reasoning limitations of llms via game-theoretic evaluations, 2024. URL https://arxiv.org/abs/2402.12348.

Guertler, L., Cheng, B., Yu, S., Liu, B., Choshen, L., and Tan, C. TextArena. *arXiv preprint arXiv:2504.11442*, 2025.

Jacovi, A., Caciularu, A., Goldman, O., and Goldberg, Y. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5075–5084, Singapore, December 2023. Association for Computational Linguistics.

Lai, H. and Nissim, M. mCoT: Multilingual instruction tuning for reasoning consistency in language models. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12012–12026, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.649. URL https://aclanthology.org/2024.acl-long.649/.

Li, Y., Guerin, F., and Lin, C. LatestEval: Addressing data contamination in language model evaluation through dynamic and time-sensitive test construction. *arXiv preprint arXiv:2312.12343*, 2024.

Madaan, L., Singh, A. K., Schaeffer, R., Poulton, A., Koyejo, S., Stenetorp, P., Narang, S., and Hupkes, D. Quantifying variance in evaluation benchmarks. *arXiv preprint arXiv:2406.10229*, 2024.

OpenAI. GPT-4 technical report. Technical report, OpenAI, 2023.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Roumeliotis, K. I. and Tselikas, N. D. ChatGPT and OpenAI models: A preliminary review. *Future Internet*, 15(6): 192, 2023.

Sakai, Y., Kamigaito, H., and Watanabe, T. mCSQA: Multilingual commonsense reasoning dataset with unified creation strategy by language models and humans. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14182–14214, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Voracek, V. Treatment of statistical estimation problems in randomized smoothing for adversarial robustness. *Advances in Neural Information Processing Systems*, 37: 133464–133486, 2024.

Wang, H., Xu, J., Xie, S., Wang, R., Li, J., Xie, Z., Zhang, B., Xiong, C., and Chen, X. M4U: Evaluating multilingual understanding and reasoning for large multimodal models. *arXiv preprint arXiv:2405.15638*, 2024. doi: 10.48550/arXiv.2405.15638. URL https://arxiv.org/abs/2405.15638.

White, C., Dooley, S., Roberts, M., Pal, A., Feuer, B., Jain, S., Shwartz-Ziv, R., Jain, N., Saifullah, K., Dey, S., Shubh-Agrawal, Sandha, S. S., Naidu, S. V., Hegde, C., LeCun, Y., Goldstein, T., Neiswanger, W., and Goldblum, M. LiveBench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yang, D., Kleinman, E., and Harteveld, C. GPT for games: A scoping review (2020–2023). *arXiv preprint arXiv:2404.17794*, 2024.

Yang, Y., Shi, X., Wei, L., and Huang, M. Quantifying contamination in code generation benchmarks. *arXiv preprint arXiv:2303.17470*, 2023.

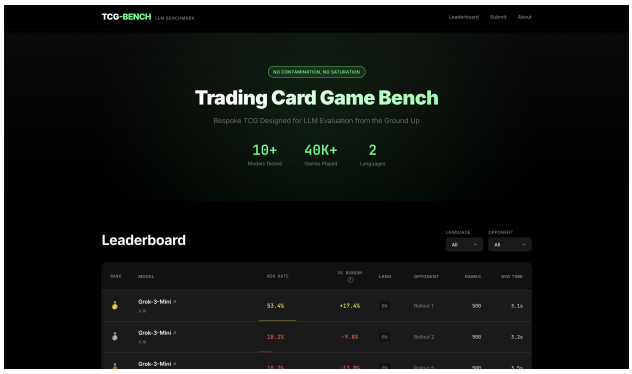

Figure 3. TCG-Bench Website

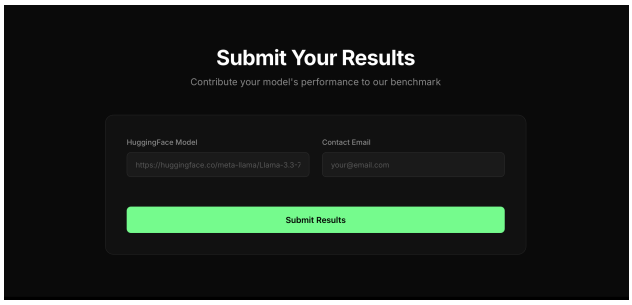

Figure 4. Website's Model Request Form

A. TCG-Bench Website

We have developed a public-facing website (Figure 3,4) that serves as the central hub for our TCG-Bench. Users can upload their own models to participate in the TCG competitions, and their performance is automatically evaluated and displayed on a dynamic leaderboard.

B. Analysis of the Data

To contextualize the scope and reliability of our benchmark evaluation, we provide a statistical overview of the full dataset and experimental setup (Table 2). The analysis spans 42,750 games, encompassing 17 distinct models evaluated across 6 difficulty levels and in 2 languages (English and Arabic), ensuring broad coverage and linguistic generalizability. The average game duration is 11.4 turns, indicating that matches are long enough to capture mid- and late-game reasoning patterns, yet short enough to support scalable evaluation. The median decision latency of 3.2 seconds reflects realistic deployment speeds for most transformer-based models, with response time kept within acceptable interaction thresholds for turn-based strategic tasks. On average, models generate 412 tokens per turn, highlighting the verbosity and complexity of in-context decision-making

Table 2. Statistical Summary.

Category	Value
Total Games Evaluated	42,750
Models Tested	17
Languages	2
Difficulty Levels	6
Average Game Duration	11.4 turns
Median Decision Time	3.2s
Mean Tokens per Turn	412
Confidence Interval	95% (Voracek, 2024)

Table 3. Error Pattern Analysis. Here, "Late losses" refer to games lost after 10 turns, while "No tricks/spells/champs" indicate cases where the model never played cards of that type. "Low diversity" includes games where fewer than 3 unique cards were used. Game Length refers to mean game length in turns.

Type	Count	Percentage	Game Length
Late losses	4,631	59.9%	14.2
No tricks	1,487	19.2%	8.7
No spells	835	10.8%	9.1
No champs	750	9.7%	7.4
Low diversity	29	0.4%	11.3

required by our task. This metric also underscores the computational demands of multi-turn planning and serves as a proxy for reasoning trace density. All reported win rates and comparative statistics are accompanied by 95% confidence intervals calculated the state-of-the-art randomized estimator in (Voracek, 2024) that has *exact* coverage, ensuring rigorous and conservative estimation of model performance.

B.1. Error Pattern Analysis

To better understand the failure modes of LLM agents in the TCG benchmark, we conducted a large-scale analysis of decision trace logs, categorizing common error types and quantifying their impact (Table 3). Each error type is associated with distinct behavioral deficiencies and temporal signatures, offering insight into model limitations and potential avenues for alignment improvement.

The most frequent failure mode is late-game losses (defined as games lasting over 10 turns that end in defeat), accounting for 59.9% of all identified errors. These games exhibit a relatively long mean duration (14.2 turns), suggesting that models struggle with maintaining strategic coherence. This pattern aligns with the hypothesis that long-horizon credit assignment remains a core challenge for current LLM architectures. A second prominent class involves card-type omissions. Notably, 19.2% of failed games involved never playing a single Trick, while 10.8% and 9.7% respectively omitted Spells and Champions altogether. These omissions imply a lack of policy coverage over the full card action

Table 4. Temporal variance across four game segments. "WinRateVar" = variance in win rate; "DecTimeVar" = variance in model decision latency; "TokenUsedVar" = variance in token consumption.

Period	WinRateVar	DecTimeVar	TokenUsedVar
First 25%	2.3%	0.8s	45
Second 25%	1.9%	0.6s	38
Third 25%	2.1%	0.7s	41
Final 25%	2.0%	0.9s	39

space, potentially stemming from insufficient exposure during pretraining or overfitting to certain card-type heuristics. Furthermore, the average game lengths for these categories (7–9 turns) suggest early collapses due to suboptimal or overly narrow strategies. Rare but revealing is the Low card diversity error, observed in only 0.4% of failures, but associated with moderate-length games (11.3 turns). These cases indicate that some models repeatedly deploy a narrow subset of cards, reflecting rigid decision loops or failure to generalize across game contexts.

Taken together, these findings underscore the importance of diversity, adaptability, and long-horizon reasoning in emergent strategic performance.

B.2. Temporal Stability Data

We analyze the temporal stability of model performance and resource utilization across different segments of gameplay, dividing each match into four equal time periods. For each segment, we compute variance in win rate, decision latency, and token usage (Table 4). The goal is to assess whether LLMs exhibit consistent behavior throughout gameplay or if performance degrades over time.

Across all time periods, we observe low variance in win rates (1.9%–2.3%), indicating a stable level of competence from the opening move through to endgame. The first 25% shows slightly higher variance (2.3%), possibly reflecting early-game exploration or stochastic variability in opening strategies. Decision time variance remains consistently low (0.6–0.9 seconds), with a slight uptick in the final 25% of gameplay. This increase may reflect the greater complexity of endgame scenarios, where models must evaluate a broader state space. However, the observed increase is modest. Similarly, token usage variance remains bounded (38–45 tokens per turn), with the first segment again showing the highest variability.

Overall, these results suggest that the evaluated LLMs exhibit strong temporal coherence and operational stability, with minimal behavioral drift or computational inconsistency over time. This stability is promising for downstream applications that rely on predictable, multi-step reasoning trajectories.

Table 5. Family Strategy Distribution

Family	Champions	Spells	Tricks	Sample Size
Qwen	33.2%	33.4%	33.4%	8,000
Gemini	44.8%	30.1%	25.1%	1,500
LLaMA	35.2%	40.3%	24.5%	2,250
DeepSeek	30.1%	31.6%	38.3%	1,000
GPT	37.5%	35.2%	27.3%	500

Table 6. Correlation between model size and performance. "R1 Win%" and "R5 Win%" denote mean win rates under top-1 and top-5 evaluation respectively; "Corr. w/ Size" indicates Pearson correlation with model parameter count.

Size Range	R1 Win%	R5 Win%	Corr. w/ Size
$<$ 10B	26.0%	9.8%	0.12
10-50B	32.5%	11.2%	0.18
50-100B	**37.5%**	13.6%	0.23
$>$100B	31.1%	**15.8%**	**0.31**

B.3. Model Family Strategy Distribution

To better understand strategic tendencies across different LLMs, we analyze the distribution of card types, Champions, Spells, and Tricks, played by each model family in our benchmark (Table 5). Notably, each family exhibits a distinct style of play, suggesting learned priors or architectural biases.

Qwen (Bai et al., 2023) demonstrates a remarkably balanced strategy, allocating nearly equal proportions ($\sim$33.3%) to all three card types. In contrast, Google's Gemini models (Team et al., 2023) heavily favor Champions (44.8%). LLaMA models (Touvron et al., 2023) exhibit a strong preference for Spells (40.3%), likely reflecting a propensity for indirect or support-based interventions. DeepSeek (Bi et al., 2024) stands out for its emphasis on Tricks (38.3%), a card type associated with reactive or deceptive play. Finally, GPT models (Roumeliotis & Tselikas, 2023) adopt a moderately balanced profile, with a slight lean toward Champions and Spells, but a noticeably lower use of Tricks.

B.4. Performance vs Model Size Correlation

We examine the relationship between model scale (in billions of parameters) and performance using average R1 (Top-1) and R5 (Top-5) win rates across four parameter ranges (Table 6). The results show a generally increasing trend in performance with model size, though this trend is not strictly monotonic.

Models with fewer than 10B parameters perform poorly, with an average R1 win rate of 26.0% and R5 win rate of 9.8%. Performance improves in the 10–50B range (R1: 32.5%, R5: 11.2%), and peaks in the 50–100B range (R1:

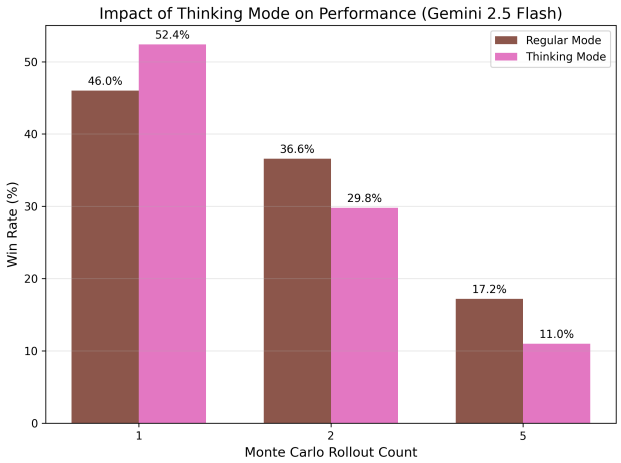

Figure 5. Thinking Mode vs. Regular Mode

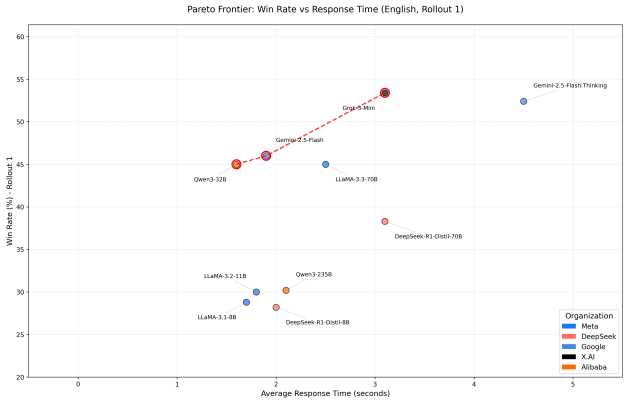

Figure 6. Decision Time vs. Performance

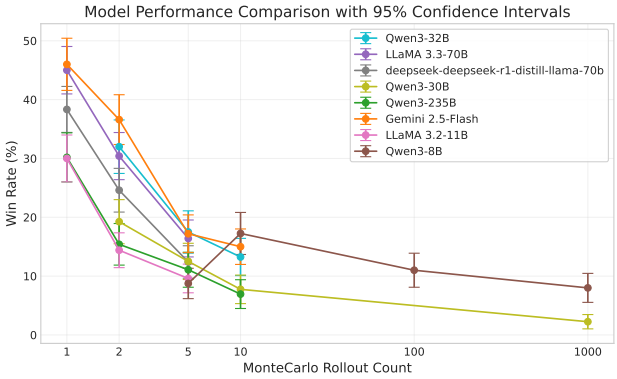

Figure 7. Model comparision

37.5%, R5: 13.6%), indicating that this middle scale is optimal for top-1 accuracy. Interestingly, models above 100B parameters see a drop in R1 performance (31.1%), but their R5 win rate continues to rise (15.8%). This may suggest that ultra-large models tend to generate more diverse high-quality candidates, improving overall top-5 performance at the cost of slightly reduced precision. The final column in the table reflects Pearson correlation values within each scale segment. These increasing values, from 0.12 in the smallest range to 0.31 in the largest, suggest a strengthening relationship between size and performance.

B.5. Effects of Reasoning Modes

To investigate how internal reasoning configurations impact performance, we compare Regular Mode and Thinking Mode within the same Gemini 2.5 Flash model architectur under varying Monte Carlo rollout budgets (Figure 5). Each mode is evaluated at rollout counts of 1, 2, and 5. Surprisingly, the effect of the Thinking Mode is not uniformly beneficial. At a rollout count of 1, Thinking Mode significantly outperforms Regular Mode, achieving a win rate of 52.4% compared to 46.0%. However, as the rollout count increases, the advantage of Thinking Mode diminishes and eventually reverses. At 2 rollouts, Regular Mode achieves 36.6% while Thinking Mode drops to 29.8%. The trend continues at 5 rollouts, where Regular Mode maintains a win rate of 17.2%, notably higher than Thinking Mode's 11.0%.

B.6. Efficiency Trade-Off

B.7. Full Model Result Comparison

Building on the baseline results presented in Figure 1, we extended the Monte Carlo rollout count for selected models to 100 and 1000 to further investigate the impact of planning depth. As shown in Figure 7, our model's win rate, when acting as the opponent, consistently decreases as the Rollout Count increases. In addition, we report the associated variance to illustrate the fluctuation in performance between runs. The results can also be found at Table 7, where the first column represents the win rate, while the second column indicates the confidence interval.

C. Gameplay in TCG-Bench

This section presents a sample game using the toy cards to illustrate TCG-Bench's core mechanics and strategic elements.

C.1. Card Reference

We provide in our code base three example cards to showcase the gameplay mechanics available, specifically choosing one from each type, champion, spell, and trick. The cards are in Table 8. The rules are also described in our

Table 7. TCG-Bench Full Results with 95% Confidence Intervals

Organization	Model	Language	Type	Rollout 1 (%)	CI	Rollout 2 (%)	CI	Rollout 5 (%)	CI	Rollout 10 (%)	CI
Meta	LLaMA	EN	3.1-8B	28.8	[25.1%, 32.5%]	–	–	–	–	–	–
	LLaMA	EN	3.2-11B	30.0	[26.4%, 33.8%]	14.4	[11.6%, 17.4%]	9.6	[7.3%, 12.1%]	–	–
	LLaMA	EN	3.3-70B	45.0	[41.0%, 49.1%]	30.4	[26.7%, 34.2%]	16.4	[13.5%, 19.5%]	–	–
Qwen	Qwen3	EN	4B	–	–	19.3	[16.1%, 22.5%]	10.0	[7.7%, 12.7%]	–	–
	Qwen3	EN	8B	–	–	–	–	8.8	[6.5%, 11.2%]	–	–
	Qwen3	EN	30B	–	–	19.3	[16.1%, 22.5%]	12.5	[10.0%, 15.4%]	7.80	[5.7%, 10.1%]
	Qwen3	EN	32B	45.0	[41.0%, 49.1%]	32.0	[28.3%, 35.9%]	17.5	[14.3%, 20.7%]	13.3	[10.6%, 16.1%]
	Qwen3	EN	235B	30.2	[26.5%, 34.0%]	15.4	[12.5%, 18.5%]	11.1	[8.6%, 13.5%]	6.9	[4.7%, 8.8%]
	Qwen3	AR	4B	–	–	15.9	[13.0%, 19.0%]	9.5	[7.3%, 12.1%]	–	–
	Qwen3	AR	8B	–	–	14.9	[12.1%, 17.9%]	8.8	[6.5%, 11.2%]	–	–
	Qwen3	AR	30B	–	–	–	–	9.3	[7.0%, 11.8%]	5.8	[4.0%, 7.8%]
	Qwen3	AR	32B	–	–	–	–	17.5	[14.3%, 20.7%]	–	–
	Qwen3	AR	235B	–	–	–	–	6.5	[4.7%, 8.8%]	–	–
Google	Gemini	EN	2.5-Flash-Preview	46.0	[42.0%, 50.1%]	36.6	[32.6%, 40.5%]	17.2	[14.2%, 20.4%]	15.0	[12.2%, 18.1%]
	Gemini	EN	2.5-Flash-Preview:thinking	52.4	[48.3%, 56.4%]	29.8	[26.0%, 33.5%]	11.0	[8.6%, 13.8%]	–	–
DeepSeek	DeepSeek	EN	R1-Distil-8B	28.2	[25.0%, 31.4%]	–	–	–	–	–	–
	DeepSeek	EN	R1-Distil-70B	38.3	[34.4%, 42.3%]	24.6	[21.1%, 28.1%]	12.4	[9.7%, 15.1%]	–	–
x-AI	Grok	EN	3-Mini	53.4	[49.3%, 57.4%]	18.2	[15.2%, 21.5%]	10.2	[7.9%, 12.9%]	–	–

GitHub repository.

Table 8. A Toy Card Set

Card Name	Type	Effect
Mighty Warrior	Champion	Power: 3, Guard: 2. When summoned, gain 1 Life Point.
Fireball	Spell	Deal 2 damage to opponent.
Counterattack	Trick	When attacked directly, block and deal 1 damage to attacker.

C.2. Game Flow

Initial State:

- **Player 1 (LLM)**: Life Points: 10, Hand: Mighty Warrior, Fireball
- **Player 2 (MCTS)**: Life Points: 10, Hand: Counterattack, Mighty Warrior

Turn 1 (Player 1):

```
Player 1 draws Counterattack.
Player 1 plays Mighty Warrior.
Effect: Player 1 gains 1 LP.
```

Turn 2 (Player 2):

```
Player 2 draws Fireball.
Player 2 plays Counterattack (hidden).
```

Turn 3 (Player 1):

```
Player 1 draws Fireball.
Player 1 plays Fireball.
Effect: Player 2 loses 2 Life Points.
Player 1's Mighty Warrior attacks.
Player 2 activates Counterattack.
Effect: Attack blocked, Player 1 loses 1 LP.
```

Turn 4 (Player 2):

```
Player 2 draws Mighty Warrior.
Player 2 plays Mighty Warrior.
Effect: Player 2 gains 1 LP.
```

Final State:

- **Player 1 (LLM)**: Life Points: 10, Hand: Fireball, Board: Mighty Warrior
- **Player 2 (MCTS)**: Life Points: 9, Hand: Fireball, Board: Mighty Warrior

C.3. Strategic Elements

This example demonstrates key strategic elements in TCG-Bench:

1. **Resource Management:** Players must decide whether to deploy cards immediately or save them.
2. **Hidden Information:** Trick cards create uncertainty, forcing players to reason about incomplete information.
3. **Multi-Step Planning:** The sequence of Champion followed by Spell illustrates forward planning.
4. **Adaptive Reasoning:** Players must adjust when tricks disrupt their expected outcomes.