
GAMEBENCH: Evaluating Strategic Reasoning Abilities of LLM Agents

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Large language models have demonstrated remarkable few-shot performance on
2 many natural language understanding tasks. Despite several demonstrations of
3 using large language models in complex, strategic scenarios, there lacks a com-
4 prehensive framework for evaluating agents’ performance across various types
5 of reasoning found in games. To address this gap, we introduce GAMEBENCH,
6 a cross-domain benchmark for evaluating strategic reasoning abilities of LLM
7 agents. We focus on 9 different game environments, where each covers at least
8 one axis of key reasoning skill identified in strategy games, and select games for
9 which strategy explanations are unlikely to form a significant portion of models’
10 pretraining corpora. Our evaluations use GPT-3 and GPT-4 in their base form
11 along with two scaffolding frameworks designed to enhance strategic reasoning
12 ability: Chain-of-Thought (CoT) prompting and Reasoning Via Planning (RAP).
13 Our results show that none of the tested models match human performance, and
14 at worse GPT-4 performs worse than random action. CoT and RAP both im-
15 prove scores but not to comparable human levels. Benchmark code is available at
16 <https://anonymous.4open.science/r/GameBench-5942/>.

17 1 Introduction

18 Capabilities of large language models have seen rapid progress, enabling LLMs to be used in agentic
19 tasks [Schick et al., 2023] [Watkins et al., 2023] [Richards, 2023]. This presents opportunities for
20 LLM-based tools to assist humans in several domains, such as API usage [Li et al., 2023], web
21 browsing [Schick et al., 2023] and coding [Kazemitabaar et al., 2023]. Recent benchmarks have
22 been introduced for evaluating performance on real-world agent tasks [Wang et al., 2024], [Liu
23 et al., 2023a], [METR, 2023], [Mialon et al., 2023], with some focused on reasoning [Sawada et al.,
24 2023] and games [Lin et al., 2023]. However, these existing benchmarks are oriented to practical,
25 in-distribution knowledge, which can quickly become saturated with better models.

26 In particular, strategic reasoning is an agentic task that is important for generalising to new contexts,
27 as it involves optimising for an objective in the face of possibly divergent interests of others, where
28 incentives may not be fully known [Gandhi et al., 2023b]. Prior work on reasoning scaffolds also
29 shows that language models have potential to grasp reasoning skills across scenarios [Wei et al., 2022b,
30 Hao et al., 2023]. Hence, a strategic reasoning benchmark for LLMs, that is inherently multi-agent,
31 would be difficult to saturate. Furthermore, games exemplify environments for demonstrating strategic
32 behaviour in both humans and AI agents, as seen in the well known examples of Chess [Silver et al.,

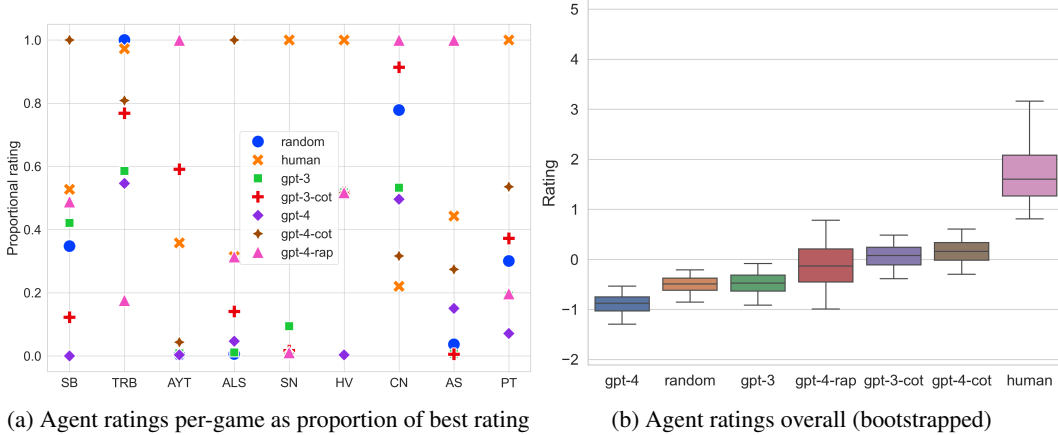


Figure 1: **Rating data** With CoT scaffolding, GPT-4 is the best reasoner below only the human baseline, achieving the best LLM performance on Sea Battle and Pit. But without, it performs worse than even the random baseline due to its exceedingly low rating on Sea Battle. The state-of-the-art RAP scaffolding doesn’t provide as much of an improvement to GPT-4 as CoT does. Looking at the top line of Figure 1a reveal the best agent in each game. come from exponential Bradley–Terry model. See section 3.4 for details. The whiskers represent 90% CIs computed from our bootstrapping process formalized in 3.4. ALS = Air, Land, Sea; ARC = Arctic Scavengers; AYT = Are You the Traitor?; CN = Codenames; HV = Hive; PT = Pit; SN = Santorini; TRB = Two Rooms and a Boom; SB = Sea Battle.

2017] and Go [Silver et al., 2016]. Hence evaluating LLMs on several types of reasoning behaviours would present a comprehensive, fine-grained benchmark. As such, we introduce **GAMEBENCH**: a multi-player, cross-domain framework for evaluating strategic reasoning in LLM agents using games. We focus on both discrete and open-ended action spaces, across the reasoning domains of abstract strategy; non-deterministic outcomes; hidden information; language communication; social deduction and cooperation between players. By selecting for games without published strategy guides to our knowledge, we ensure that game-specific strategy has been sufficiently out-of-distribution in pretraining data. See Table 1 for a complete list of games and game properties.

The benchmark consists of obscure board games, card games, and social deception games. We evaluate gpt-3.5-turbo-1106 (GPT-3) and gpt-4-1106-preview (GPT-4) along with the CoT [Wei et al., 2022b] and RAP [Hao et al., 2023] scaffolding techniques, by playing them against each other, a random-action-selector baseline, and a human baseline. We conducted a literature review and identified RAP to be the state-of-the-art scaffolding that fit the parameters of our benchmark, i.e. each agent has access to the same game state information and no agent can peek at future states. Agents are rated using the exponential Bradley–Terry model [Bradley and Terry, 1952]. This has useful advantages over the typical Elo system [Elo, 1967], such as its assumption that each agent’s ability is fixed and will not change between matches.

Our results show that CoT-augmented and RAP-augmented models demonstrate superior strategic reasoning to the random baseline; that GPT-3 matches the random baseline; that GPT-4 performs worse than the random baseline; and that the human baseline performs superior to all.

With this benchmark, we propose a means to measure the strategic reasoning abilities of LLM agents in diverse game environments. Our contributions are as follows:

- **GAMEBENCH**, the first benchmark to capture both cross-domain and out-of-distribution strategic reasoning for comparison across multiple agents.
- **Empirical results** on GPT-3 and GPT-4, demonstrating the effects of Chain-of-Thought scaffolding and the state-of-the-art scaffolding.

59 2 Related works

60 **LLM agents playing games** Using games to evaluate LLMs has significant precedent in previous
61 research. Some studies evaluate models using single strategic tasks or games, such as Minecraft
62 [Wang et al., 2023, Zhu et al., 2023], Diplomacy [Bakhtin et al., 2022], Avalon [Light et al., 2023],
63 and Werewolf [Xu et al., 2023b]. Other benchmarks [Wu et al., 2023a, Liu et al., 2023b] capture a
64 more comprehensive picture by using suites of multiple tasks or games to evaluate LLMs as intelligent
65 agents. However, the tasks represented in these benchmarks don’t involve interaction with other
66 agents, so they don’t reflect strategic reasoning as defined in this work.

67 **Game-theoretic scenarios** Several benchmark suites focus on common game theory scenarios, such
68 as auctions [Chen et al., 2023, Mao et al., 2023], matrix games like Prisoner’s Dilemma [Akata et al.,
69 2023, Gandhi et al., 2023a], and negotiation [Abdelnabi et al., 2023, Gandhi et al., 2023a]. While
70 they do involve multi-agent interaction and are useful for testing models’ strategic reasoning ability,
71 our benchmark focuses on more complex games that aren’t as frequently studied as these game theory
72 scenarios. Given no major strategy guides or forums dedicated to these games, we believe there is
73 less documentation on optimal strategies for them present in LLMs’ training corporuses.

74 **Dialogue-based games** Some benchmarks employ dialogue-based games that are less well-
75 documented on the internet: Agashe et al. [2024] and Chalamalasetti et al. [2023] use novel co-
76 operative dialogue games, and Qiao et al. [2023] uses two social deduction games and one word
77 guessing game. However, our benchmark aims to evaluate LLMs’ strategic reasoning ability not only
78 in cooperative and conversational environments, but competitive, spatial, and non-deterministic ones
79 as well.

80 **Diverse multi-agent game suites** The benchmarks most similar to ours employ diverse suites of
81 complex multi-agent games, including conversational, board, and card games [Chen et al., 2024,
82 Duan et al., 2024, Abdulhai et al., 2023, Xu et al., 2023a]. However, many of the included games are
83 either commonly found on the internet, such as TicTacToe, Poker, and Connect Four, or common
84 game-theoretic scenarios, as discussed previously. These games are not as out-of-distribution as
85 desired.

86 In summary, we build upon previous work by introducing a diverse suite of multi-agent games
87 to evaluate the strategic reasoning ability of LLMs as agents. Our benchmark is characterized by
88 its inclusion of complex games that span a range of game characteristics and are not likely to be
89 well-represented in LLMs’ pretraining corporuses.

90 3 GAMEBENCH

91 In Section 3.1 we discuss our reasoning behind our selection of agents and scaffolds. In Section
92 3.2 we describe our methodology for selecting suitable games. In Section 3.3 we describe the agent
93 and game interfaces. In Section 3.4 we introduce our rating model and formalize our process for
94 calculating ratings.

95 3.1 Agent and scaffolding selection

96 We benchmark GPT-3 (gpt-3.5-turbo-1106) and GPT-4 (gpt-4-1106-preview) due to their size,
97 mainstream popularity, and convenient public API. We include these base models as well as several
98 black-box scaffolding interventions in order to measure the relative effects these scaffolding interven-
99 tions have on improving the reasoning abilities of the base models. We selected Chain-of-Thought
100 [Wei et al., 2022b] prompting for its ubiquity and Reasoning-via-Planning [Hao et al., 2023] for
101 its state-of-the-art status. We also include a random-action-selecting agent as baseline of no strate-
102 gic reasoning ability, and a human agent as a baseline of progress towards human-level strategic
103 reasoning.

104 For more details about agent implementation, see Appendix D.

3.2 Game selection

In order to evaluate a broad range of cognitive skills associated with strategic reasoning, we curated a diverse set of games featuring abstract strategy, non-deterministic outcomes, hidden information, language communication, social deduction and bluffing, and cooperation between players. A breakdown of which games had these features can be found in Table 1.

Using these categories, we then filtered for games unlikely to be significantly represented in LLMs’ pretraining data, to evaluate the models’ out-of-distribution reasoning abilities. Two key criteria were (a) excluding games with dedicated online forums discussing improvement strategies, as well as (b) excluding games with published strategy guides. After finalizing the selection of games, we formalized their rulesets and mechanics into programmatic environments that the LLM agents could interact with.

Our final selection of games were *Air, Land, Sea* (ALS); *Arctic Scavengers* (ARC); *Are You the Traitor?* (AYT); *Codenames* (CN); *Hive* (HV); *Pit* (PT); *Santorini* (SN); *Two Rooms and a Boom* (TRB); and *Sea Battle* (SB). Descriptions of the games and their rules can be found in Appendices F and G respectively. For additional details about game implementation, see Appendix D.

Table 1: **Number of games per reasoning category** We identify a set of six orthogonal components of strategic reasoning and curate a set of games that sufficiently cover their spread.

Reasoning Category	Total	Games
Abstract Strategy	6	ALS, ARC, CN, HV, SN, SB
Non-Deterministic	3	ARC, TRB, SB
Hidden Information	3	ARC, AYT, TRB
Language Communication	4	AYT, CN, PT, TRB
Social Deduction	2	AYT, TRB
Cooperation	4	AYT, CN, SB, TRB

3.3 API

Each environment, implemented in Python, describes a Game object with methods for initializing, retrieving the game’s current state and available actions, updating the state with an action, and executing a full match between two agents. Agents are objects that describe a method for choosing an action conditioned on the rules, state, and available actions retrieved from a Game instance. Agents are instantiated at the beginning of a match and destroyed at the end, so agents may maintain persistent state between moves to choose an action.

3.4 Rating calculation

We formalize our rating calculation as follows. Let our dataset contain P , the population of all possible matches across all games, and $S = \{m_1, m_2, \dots, m_n\}$, our sample of n matches. Define the weight w_i for each match m_i to be inversely proportional to the number of matches collected for that match’s game. Specifically, if match m_i belongs to game X which has N_X matches, then the weight w_i is given by:

$$w_i = \frac{1}{N_X}. \quad (1)$$

We then perform bootstrapping on the sample S for $B = 10,000$ times. Let $S_b^* = [m_{i_1}, m_{i_2}, \dots, m_{i_n}]$ be the b th bootstrapped sample, where m_{i_j} is randomly selected from S with probability proportional to w_i with replacement.

$$P(i > j) = \frac{e^{\beta_i}}{e^{\beta_i} + e^{\beta_j}} \quad (2)$$

For each bootstrapped sample S_b^* , we use maximum-likelihood estimation to fit the parameters of the above exponential Bradley–Terry model $\theta_b = \{\beta_{\text{random}}, \beta_{\text{GPT-3}}, \dots\}$. Let $\theta_{b,k}$ denote the parameter for agent k in bootstrapped sample b . We take the means of these distributions to be the "true" rating $\hat{\theta}_k$ for each agent k , given by:

$$\hat{\theta}_k = \frac{1}{B} \sum_{b=1}^B \theta_{b,k} \quad (3)$$

We considered several methods for aggregating pairwise match results across games into scores that represent the general skill of each model, including the Elo system [Elo, 1967]. Unlike Elo, the Bradley–Terry system [Bradley and Terry, 1952] assumes model skill does not change over time and it does not need to be calculated in a decentralized manner, making it more suitable for evaluating language models [Chiang et al., 2023]. In our analysis, this model also enables the comparison of models that never directly competed.

4 Empirical results

Additional figures showing agent-pairwise data covering number of games, total score, win probability, and rating per game is available in Appendix H. The rating plots in Appendix H show 90% confidence intervals for the points in Figure 1a.

Table 2: **Game ratings** The table highlights the effects of scaffolds. Across all games, GPT-4 with CoT scaffolding improves over the base model substantially. But GPT-3 with CoT scaffolding is outperformed by the base model in *Air*, *Land*, and *Sea*, *Hive*, and *Two Rooms and a Boom*. Additionally, GPT-4 with RAP scaffolding usually under-performs GPT-4-CoT except in *Are You the Traitor?*, *Sea Battle*, and *Two Rooms and a Boom*.

Agent	Rating									
	Overall	ALS	ARC	AYT	CN	HV	PT	SN	TRB	SB
random	-0.50	1.07	0.48	-2.52	-2.67	-1.15	<u>0.63</u>	0.37	-0.79	0.05
human	1.76	<u>1.49</u>	<u>0.45</u>	1.92	1.26	3.63	1.29	-0.89	<u>1.70</u>	1.25
gpt-3	-0.48	1.26	-0.05	-1.84	-2.06	<u>1.27</u>	<u>0.63</u>	-0.01	-2.51	-0.41
gpt-3-cot	0.06	0.03	0.22	<u>2.42</u>	0.45	-0.44	<u>0.63</u>	<u>0.53</u>	-2.76	0.26
gpt-4	-0.89	-7.38	-0.12	-2.73	-0.65	-1.31	-4.42	-0.08	0.62	-1.40
gpt-4-cot	<u>0.16</u>	2.13	0.27	-0.19	2.41	-1.13	0.63	-0.53	1.22	<u>0.62</u>
gpt-4-rap	-0.10	1.41	-1.25	2.94	<u>1.26</u>	-0.86	<u>0.63</u>	0.62	2.51	-0.37

4.1 Human comparison

The human baseline outperforms all model and scaffolding configurations in the benchmark. The upper-bound of GPT-4-RAP’s confidence interval in Figure 1b just reaches the lower-bound of the human baseline. But due to both GPT-4-RAP and the human baseline having very few data points, this detail should not be taken very seriously. In Table 3, the human baseline achieves the highest overall score in every game it played except for *Santorini*.

The human subject beat their opponent agent in all matches except for two of the three *Codenames* matches. For these particular matches, the human subject employed a friend because *Codenames* typically requires at least two players per team. We hypothesize that LLM agents perform better in this context because they are better at modeling their teammate’s thought process, as they are instantiated from the same underlying language model. In contrast, pairs of humans share much less cognitive similarity.

Details about the human data collection process are discussed in Appendix B.

Table 3: **Average score.** The total score an agent achieved in a game divided by the number of games that agent played. Comparing with 2, this table highlights interesting correlations between empirical score and model-inferred ratings. For example, in *Air, Land, and Sea*, GPT-4-CoT has the top rating while the human baseline has second-top, but they swap when examining average score. This plot also shows more clearly why the human baseline has the highest rating even though both the human baseline and GPT-4-RAP have the highest rating in three games. Here, the human baseline achieved the highest score in four games but GPT-4-RAP only achieved the highest in two.

Agent	Score									
	Overall	ALS	ARC	AYT	CN	HV	PT	SN	TRB	SB
random	0.49	0.72	0.60	0.25	0.18	0.41	<u>0.50</u>	0.56	0.52	<u>0.58</u>
human	0.85	1.00	NaN	NaN	NaN	1.00	1.00	0.43	NaN	0.78
gpt-3	0.48	0.64	0.43	0.43	0.63	<u>0.80</u>	<u>0.50</u>	0.47	0.27	0.40
gpt-3-cot	0.60	0.43	<u>0.50</u>	<u>0.93</u>	<u>0.89</u>	0.60	<u>0.50</u>	0.61	0.33	0.55
gpt-4	0.31	0.00	0.42	0.33	0.83	0.33	0.31	0.42	0.71	0.20
gpt-4-cot	0.60	<u>0.81</u>	<u>0.50</u>	0.64	1.00	0.50	<u>0.50</u>	0.37	<u>0.75</u>	0.51
gpt-4-rap	<u>0.62</u>	NaN	0.33	1.00	NaN	0.50	NaN	<u>0.58</u>	1.00	0.26

163 4.2 Effect of scaffolding

164 Chain-of-Thought prompting provided the best median and upper quartile results of all configurations
165 tested in Figure 1b. GPT-3 and GPT-4 showed almost identical performance with GPT-4 with only a
166 slight improvement over GPT-3. The positive effects of CoT prompting are already well-documented
167 [Chowdhery et al., 2022, Zelikman et al., 2022, Wei et al., 2022a], and our results provides evidence
168 of their use in strategic settings.

169 If we interpret the addition of CoT scaffolding as an intervention on the base model, we see it
170 improves strategic reasoning ability in GPT-4 moreso than in GPT-3. In *Sea Battle*, this intervention
171 brings GPT-4 from the worst model to the best model. In every game except *Codenames*, GPT-4 with
172 CoT scaffolding outperforms its base model. But for GPT-3, the base model outperforms the CoT
173 variant in *Santorini* and *Sea Battle*. One possible hypothesis for this difference in effect between on
174 GPT-3 and GPT-4 is that GPT-4 is a bigger model and thus can probably make better use of in-context
175 information.

176 4.3 GPT-3 versus GPT-4

177 GPT-3 performs only slightly better than random action. Surprisingly, GPT-4 performs the worst of
178 all configurations with its upper quartile performance being worse than random’s lowest quartile.
179 This result is mostly due to GPT-4 losing all matches in *Sea Battle*. This challenges our aggregation
180 method: GPT-4 should not be so harshly penalized for poor performance on one game.

181 An alternative aggregation method that would be more robust to outliers is to use factor analysis
182 to isolate a "general strategic reasoning factor" that explains a significant portion of the variance
183 between models’ performances. This method is used to aggregate separate cognitive test scores into
184 IQ scores, making it apt for evaluating LLMs’ reasoning abilities [Ilić, 2023]. We expect this g-factor
185 approach to appropriately weigh models’ *Sea Battle* ratings lower, fixing this discrepancy.

186 Considering these two datapoints and analysis from 4.2, we can tentatively conclude that strategic
187 reasoning ability is not improving in OpenAI’s newest frontier models alone, but their receptiveness
188 to scaffolding to improve strategic reasoning is increasing.

189 4.4 State-of-the-art scaffolding

190 The state-of-the-art scaffolding was outperformed by both Chain-of-Thought agents. One possible
191 hypothesis for this is that, during the Monte-Carlo tree search, this agent predicts new states based
192 on the state being examined, which is already a predicted state depending if $\text{depth} \geq 1$. If the

193 agent makes any errors in this examined state’s prediction due to misunderstandings about the game
194 state or rules, these will likely be compounded in the next set of predictions. We might expect the
195 Chain-of-Thought agents to be susceptible to the same issue of compounding errors, but to a lesser
196 extent. This could be tested qualitatively by a human expert analysing GPT-4-RAP’s predictions for
197 accuracy.

198 Another hypothesis is that we ran GPT-4-RAP to a depth great-enough to surpass GPT-4 without
199 RAP scaffolding, but not great-enough depth to surpass Chain-of-Thought scaffolding. This could be
200 tested by adding several GPT-4-RAP agents to the benchmark, each with different depths.

201 It seems unlikely that Chain-of-Thought prompting should be the most sophisticated black-box
202 scaffolding, so it remains an open question to find this scaffolding in order to establish an upper-
203 bound on strategic reasoning ability with black-box scaffolds.

204 5 Discussion

205 We now discuss the limitations and future directions of our work.

206 **Confirming out-of-distribution status** It is clear by simply asking GPT-4 that it already knows
207 about these games and their rules. It is unclear, however, if it consumed any strategy guides about
208 these games in the pretraining process, which is the determining factor for out-of-distribution status
209 in our benchmark. **Future work** We propose the following experiment. Design an intervention that
210 is: supply a strategy guide in-context to a language model agent for the game it is playing. We
211 would expect this intervention to improve agent performance more on out-of-distribution games
212 than in-distribution games. Collect data of agents playing an unknown distribution game; agents
213 with the intervention playing an unknown distribution game; agents playing known in-distribution
214 games; agents with the intervention playing known in-distribution games. Compare the effect of
215 the intervention on the unknown distribution game versus the effect on the known in-distribution
216 games. If the effect is much higher on the unknown distribution game, this is a evidence for the game
217 being out-of-distribution. This would work better with known out-of-distribution games, but this
218 may not be possible to know in all cases. We could also compare models’ performance on common
219 games vs. "counterfactual" games, which are slightly modified to reduce any association with their
220 in-distribution counterparts [Wu et al., 2023b].

221 **Protecting out-of-distribution status** We did not attempt to protect these games from becoming
222 in-distribution in the future. **Future work** Developers of frontier models should curate strategic
223 reasoning environments by ensuring these games are held out from pretraining data. For ubiquitous
224 games such as chess, this is unfeasible. But following our heuristics for game selection discussed in
225 section 3.2, it should be reasonable to find games without much internet data.

226 **Results’ sensitivity to games** From inspecting GPT-4’s surprisingly low rating with *Sea Battle*, it
227 became apparent that our "multigame" approach to aggregation may be inadequate due its sensitivity
228 to the games included; i.e., ablating *Sea Battle* significantly changed the data narrative. **Future**
229 **work** We see multiple ways forward. If aggregate data is useful, investigate more robust forms
230 of aggregation, such as the g-factor or factor analysis in general. Alternatively, explore a multi-
231 dimensional approach that attempts to score agents on the six reasoning categories from Table 1.
232 Or, discard any notion of aggregation and determine effective means of analysis that looks only at
233 individual games and maybe uses more qualitative data with the help of human experts.

234 **Low-resolution human benchmark** We find it especially important to know how well these models
235 fair compared to humans, but collecting comprehensive human data was out of our means. **Future**
236 **work** Conduct more comprehensive human data to form a distribution of human strengths on each
237 game with which we can measure the progress of model and scaffolding development.

238 **Uncaught edge cases** Every few games were inspected during data collection, and occasionally, we
239 caught and fixed bugs in our evaluation code. It is possible that some edge cases went unnoticed and
240 were featured in our final data release. **Future work** Incorporating more human subjects into the data

collection should make this process trivial, as they can provide immediate feedback if they witness unexpected behavior.

Benchmark and dataset size Our benchmark has a respectable number of games and agents compared to other benchmarks [Chen et al., 2024, Duan et al., 2024, Abdulhai et al., 2023, Xu et al., 2023a], but the addition of more games and agents would provide a richer picture of models’ strategic reasoning abilities. Additionally, our dataset is fairly small and suffers from biases from variable resource cost between games. **Future work** Add more varied games to the benchmark, evaluate more model and scaffolding configurations, and collect more data for each configuration.

6 Conclusion

We present GAMEBENCH, an LLM agent benchmark to test strategic reasoning ability using diverse games that have sparse strategy material in pretraining data. We benchmark OpenAI’s GPT-3 and GPT-4 models and evaluate the impact of two scaffolding methods: Chain of Thought (CoT) and Reasoning via Planning (RAP). We find that human trials consistently outperform all LLM agents. Of all the agent configurations, CoT agents performed the best, followed by RAP-augmented GPT-4. Base GPT-3 performed on-par with the random baseline, and base GPT-4 performed worse. These results show that while measures such as scaffolding can help improve performance in strategic reasoning, even the best configuration fall short of human reasoning. LLMs show great promise working on in-distribution tasks, though their performance on OOD task sets show a low risk for current dangers of deploying autonomous agents. Nonetheless, the performance gains achieved through scaffolding techniques indicate the potential for future advancements that could increase the risk posed by such systems if their reasoning capabilities continue to improve.

Acknowledgments and Disclosure of Funding

We thank Joshua Clymer for providing advisory help. We thank Andy Lee for creating our website. We thank Dr. Allen Downey, Severin Field, and Misha Gerovitch for providing feedback on our drafts. We thank Shubhorup Biswas for implementing Atari Boxing.

References

- Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schonherr, and Mario Fritz. Llm-deliberation: Evaluating llms with interactive multi-agent negotiation games. *ArXiv*, abs/2309.17234, 2023. URL <https://api.semanticscholar.org/CorpusID:263310628>.
- Marwa Abdulhai, Isadora White, Charles Burton Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu, and Sergey Levine. Lmrl gym: Benchmarks for multi-turn reinforcement learning with language models. *ArXiv*, abs/2311.18232, 2023. URL <https://api.semanticscholar.org/CorpusID:265506611>.
- Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. Llm-coordination: Evaluating and analyzing multi-agent coordination abilities in large language models, 2024.
- Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. Playing repeated games with large language models. *ArXiv*, abs/2305.16867, 2023. URL <https://api.semanticscholar.org/CorpusID:258947115>.
- Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sandra Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David J. Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378:1067 – 1074, 2022. URL <https://api.semanticscholar.org/CorpusID:253759631>.

286 Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the
287 method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444. URL
288 <http://www.jstor.org/stable/2334029>.

289 Kranti Chalamalasetti, Jana Gotze, Sherzod Hakimov, Brielen Madureira, P. Sadler, and David
290 Schlangen. clembench: Using game play to evaluate chat-optimized language models as con-
291 versational agents. *ArXiv*, abs/2305.13455, 2023. URL [https://api.semanticscholar.org/](https://api.semanticscholar.org/CorpusID:258841392)
292 [CorpusID:258841392](https://api.semanticscholar.org/CorpusID:258841392).

293 Jiangjie Chen, Siyu Yuan, Rong Ye, Bodhisattwa Prasad Majumder, and Kyle Richardson. Put your
294 money where your mouth is: Evaluating strategic planning and execution of llm agents in an auction
295 arena. *ArXiv*, abs/2310.05746, 2023. URL [https://api.semanticscholar.org/CorpusID:](https://api.semanticscholar.org/CorpusID:263831697)
296 [263831697](https://api.semanticscholar.org/CorpusID:263831697).

297 Junzhe Chen, Xuming Hu, Shuodi Liu, Shiyu Huang, Weijuan Tu, Zhaofeng He, and Lijie Wen.
298 Llmarena: Assessing capabilities of large language models in dynamic multi-agent environ-
299 ments. *ArXiv*, abs/2402.16499, 2024. URL [https://api.semanticscholar.org/CorpusID:](https://api.semanticscholar.org/CorpusID:268032489)
300 [268032489](https://api.semanticscholar.org/CorpusID:268032489).

301 Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng
302 Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena:
303 An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*,
304 2023. URL <https://doi.org/10.48550/arXiv.2403.04132>.

305 Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam
306 Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh,
307 Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam
308 Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James
309 Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Lev-
310 skaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin
311 Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph,
312 Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M.
313 Dai, Thanumalayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon
314 Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark
315 Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean,
316 Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

317 Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-
318 Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. Gtbench: Uncovering the strategic reasoning
319 limitations of llms via game-theoretic evaluations. *ArXiv*, abs/2402.12348, 2024. URL [https:](https://api.semanticscholar.org/CorpusID:267750698)
320 [//api.semanticscholar.org/CorpusID:267750698](https://api.semanticscholar.org/CorpusID:267750698).

321 Arpad E Elo. The proposed uscf rating system, its development, theory, and applications. *Chess life*,
322 22(8):242–247, 1967.

323 Kanishk Gandhi, Dorsa Sadigh, and Noah D. Goodman. Strategic reasoning with language mod-
324 els. *ArXiv*, abs/2305.19165, 2023a. URL [https://api.semanticscholar.org/CorpusID:](https://api.semanticscholar.org/CorpusID:258968043)
325 [258968043](https://api.semanticscholar.org/CorpusID:258968043).

326 Kanishk Gandhi, Dorsa Sadigh, and Noah D. Goodman. Strategic reasoning with language models,
327 2023b.

328 Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu.
329 Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*,
330 2023. URL <https://doi.org/10.48550/arXiv.2305.14992>.

331 David Ilić. Unveiling the general intelligence factor in language models: A psychometric approach,
332 2023.

333 Majeed Kazemitabaar, Xinying Hou, Austin Henley, Barbara J. Ericson, David Weintrop, and Tovi
334 Grossman. How novices use llm-based code generators to solve cs1 coding tasks in a self-paced
335 learning environment, 2023.

336 Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang,
337 and Yongbin Li. Api-bank: A comprehensive benchmark for tool-augmented llms, 2023.

338 Jonathan Light, Min Cai, Sheng Shen, and Ziniu Hu. Avalonbench: Evaluating llms playing the game
339 of avalon, 2023. URL <https://api.semanticscholar.org/CorpusID:265302489>.

340 Jiaju Lin, Haoran Zhao, Aochi Zhang, Yiting Wu, Huqiyue Ping, and Qin Chen. Agentsims: An
341 open-source sandbox for large language model evaluation, 2023.

342 Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding,
343 Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui
344 Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang.
345 Agentbench: Evaluating llms as agents, 2023a. URL [https://doi.org/10.48550/arXiv.2308.](https://doi.org/10.48550/arXiv.2308.03688)
346 03688.

347 Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Yuxian Gu, Hangliang
348 Ding, Kai Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui
349 Zhang, Shengqi Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie
350 Tang. Agentbench: Evaluating llms as agents. *ArXiv*, abs/2308.03688, 2023b. URL [https://](https://api.semanticscholar.org/CorpusID:260682249)
351 api.semanticscholar.org/CorpusID:260682249.

352 maitrix org. llm-reasoners: A library for advanced large language model reasoning. [https://](https://github.com/maitrix-org/llm-reasoners)
353 github.com/maitrix-org/llm-reasoners, 2023. GitHub repository, accessed: 2024-06-04.

354 Shaoguang Mao, Yuzhe Cai, Yan Xia, Wenshan Wu, Xun Wang, Fengyi Wang, Tao Ge, and Furu Wei.
355 Alympics: Llm agents meet game theory – exploring strategic decision-making with ai agents,
356 2023. URL <https://api.semanticscholar.org/CorpusID:265034042>.

357 Lucas Maystre. choix: Inference algorithms for models based on luce’s choice axiom. [https://](https://github.com/lucasmaystre/choix)
358 github.com/lucasmaystre/choix, 2015. GitHub repository, accessed: 2024-06-04.

359 METR. Evaluating language-model agents on realistic autonomous tasks. [https://metr.org/](https://metr.org/blog/2023-08-01-new-report/)
360 [blog/2023-08-01-new-report/](https://metr.org/blog/2023-08-01-new-report/), 2023.

361 Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom.
362 Gaia: a benchmark for general ai assistants, 2023.

363 Dan Qiao, Chenfei Wu, Yaobo Liang, Juntao Li, and Nan Duan. Gameeval: Evaluating llms on
364 conversational games. *ArXiv*, abs/2308.10032, 2023. URL [https://api.semanticscholar.](https://api.semanticscholar.org/CorpusID:261048971)
365 [org/CorpusID:261048971](https://api.semanticscholar.org/CorpusID:261048971).

366 Toran Bruce Richards. Autogpt: An autonomous gpt-4 experiment. [https://github.com/](https://github.com/Significant-Gravitas/AutoGPT/tree/master)
367 [Significant-Gravitas/AutoGPT/tree/master](https://github.com/Significant-Gravitas/AutoGPT/tree/master), 2023.

368 Tomohiro Sawada, Daniel Paleka, Alexander Havrilla, Pranav Tadepalli, Paula Vidas, Alexander
369 Kranias, John J. Nay, Kshitij Gupta, and Aran Komatsuzaki. Arb: Advanced reasoning benchmark
370 for large language models, 2023.

371 Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer,
372 Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to
373 use tools, 2023.

374 David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche,
375 Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman,
376 Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine
377 Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with
378 deep neural networks and tree search. *Nature*, 529:484–489, 01 2016. doi: 10.1038/nature16961.

- David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024.
- Zihao Wang, Shaoifei Cai, Anji Liu, Xiaojuan Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. ArXiv, abs/2302.01560, 2023. URL <https://api.semanticscholar.org/CorpusID:256598146>.
- Adam Watkins, Srikanth Subedi, and Asim Shrestha. Agentgpt. <https://github.com/reworkd/AgentGPT>, 2023.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022a.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903, 2022b. URL <https://doi.org/10.48550/arXiv.2201.11903>.
- Yue Wu, Xuan Tang, Tom M. Mitchell, and Yuanzhi Li. Smartplay : A benchmark for llms as intelligent agents. ArXiv, abs/2310.01557, 2023a. URL <https://api.semanticscholar.org/CorpusID:263608611>.
- Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. ArXiv, abs/2307.02477, 2023b. URL <https://api.semanticscholar.org/CorpusID:259341893>.
- Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See-Kiong Ng, and Jiashi Feng. Magic: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration. ArXiv, abs/2311.08562, 2023a. URL <https://api.semanticscholar.org/CorpusID:265212971>.
- Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. Exploring large language models for communication games: An empirical study on werewolf. ArXiv, abs/2309.04658, 2023b. URL <https://api.semanticscholar.org/CorpusID:261681932>.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022.
- Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, Y. Qiao, Zhaoxiang Zhang, and Jifeng Dai. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. ArXiv, abs/2305.17144, 2023. URL <https://api.semanticscholar.org/CorpusID:258959262>.

Checklist

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? **[Yes]** The paper’s contributions are listed plainly at the end of the introduction and explained in-depth in sections 3 and 4.

- (b) Did you describe the limitations of your work? [\[Yes\]](#) See Section 5.
- (c) Did you discuss any potential negative societal impacts of your work? [\[Yes\]](#) See Appendix A.
- (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [\[Yes\]](#) We discuss human research in Appendix B, data concerns in Appendix C, and societal impact in Appendix A.
- 2. If you are including theoretical results...
 - (a) Did you state the full set of assumptions of all theoretical results? [\[N/A\]](#) We are not including theoretical results.
 - (b) Did you include complete proofs of all theoretical results? [\[N/A\]](#) We are not including theoretical results.
- 3. If you ran experiments (e.g. for benchmarks)...
 - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [\[Yes\]](#) See Appendix D.
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [\[N/A\]](#) We did not do any training.
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [\[Yes\]](#) Figure 1b shows error bars, which are explained in the caption. And Appendix H shows more rating breakdowns by game with error bars.
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [\[Yes\]](#) See Appendix C.
- 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - (a) If your work uses existing assets, did you cite the creators? [\[Yes\]](#) We cite creators in our Github and in Appendix D.
 - (b) Did you mention the license of the assets? [\[Yes\]](#) We mention licenses in our Github where required and in Appendix D.
 - (c) Did you include any new assets either in the supplemental material or as a URL? [\[Yes\]](#) See Appendix C.
 - (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [\[N/A\]](#) We are not using nor curating other people’s data.
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [\[Yes\]](#) See Appendix B.
- 5. If you used crowdsourcing or conducted research with human subjects...
 - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [\[Yes\]](#) See Appendix B.
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [\[Yes\]](#) See Appendix B.
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [\[Yes\]](#) See Appendix B.

A Hazards

We believe that good strategic reasoning is a dangerous capability for an AI agent to have, especially one that will operate autonomously. Thus, good performance on this benchmark could correlate with harmful risk. This is important for labs developing frontier models to be able to measure and be aware of, but it is also possible for a malicious or ignorant actor to use this benchmark as a feedback signal to improve their own large language model’s strategic reasoning ability. However, we think that the former benefit outweighs the latter risk in this time where the development of large language models is largely controlled by a few frontier labs. And we reduce the risk from ignorant actors by producing these benchmarks and discussing their importance.

473 B Research on human subjects

474 Our human-based data-points came from a co-creator of the benchmark, and the same person with
475 their friend for *Codenames*.

476 The instructions were communicated informally because the subject co-designed the benchmark and
477 this human study. They were initially instructed to play against the GPT-4-RAP, but due to resource
478 costs, were later instructed to play against the any agent except GPT-4-RAP or the random baseline.
479 They were instructed to not play *Are You the Traitor?* and *Two Rooms and a Boom* because they are
480 social deduction games and it is not a good setup to have one agent with extra information than other
481 agents. They were instructed to collect as many matches as they were willing to collect in the time
482 they had available.

483 No additional compensation was provided to them for data collection, but the API costs were covered.

484 Given the informal nature of the data collection, the near-zero risk, and the fact that the subject was a
485 co-creator in this benchmark and this experiment, we did not discuss risks or consult an IRB. The
486 data this person created do not contain any identifying information.

487 C Dataset documentation

488 The data used to generate the figures and tables in this paper are available in our Github <https://anonymous.4open.science/r/GameBench-5942/> under the CC-BY 4.0 license. These data will
489 remain available here as long as Github is available. New data may be added by the authors in the
490 future, which will be documented in the commits.

492 The intended use of this data is to compare GPT-3 and GPT-4 on this benchmark, and to compare
493 against new models, scaffolds, baselines, and informed-and-consenting humans in the future.

494 The data are in JSON format. The top-level object is an array, and the array contains objects. Each
495 object has a "game" key which indicates the game, and two other keys – the two agents that played in
496 no particular order – with their respective score as the value. Scores are in the range $[0, 1]$ and sum
497 to 1.

498 Our data collection was not uniform across games nor against agent-pairs due to resource constraints.
499 In general, we preferred playing agents against the random baseline and preferred games that didn't
500 take too long to complete.

501 All data for each agent except the random agent were collected using OpenAI's completions API.
502 Each game was designed to take no more than 5 minutes when playing base GPT-4 against random.
503 Cost estimates were not obtained, but it can be assumed that CoT agents will cost approximately
504 twice their base variants, and GPT-4-RAP will cost approximately base cost \times MCTS depth \times
505 number of actions per state \times 6

506 D Additional implementation details

507 To measure multimodal capabilities, *Hive* was made to use images to represent its game state,
508 instead of text like all the other games. However, GPT-3 is not multimodal, so it was served textual
509 representations of the graphical state created by GPT-4. Then, for RAP, GPT-4 with the completions
510 API can't produce images when predicting future states, so for simplicity, the image is turned into a
511 text description here as well.

512 GPT-4-RAP was run with the default parameters from the *llm-reasoners* library [maitrix org, 2023]
513 except the Monte-Carlo tree search depth limit was set to 2 due to resource constraints.

514 Data was not collected for a GPT-3-RAP because GPT-3 refused to comply with prompts asking
515 it to predict actions, game states, or other players' behaviors. The model would often reply, "As a
516 language model, I am incapable of predicting..." Because it is unlikely that GPT-3 is self-aware and

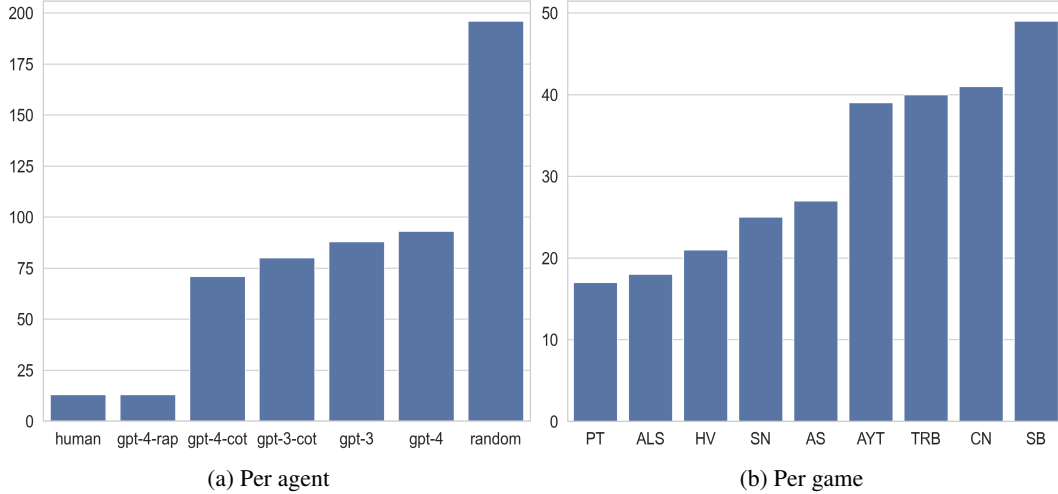


Figure 2: **Number of matches recorded** The random baseline and faster games were oversampled due to their low cost.

517 because we never indicate to the model that it is a language model in our prompting, we hypothesize
 518 that this refusal is due to the nature of GPT-3’s hidden system prompting for refusing unsafe behaviors
 519 and not due to a lack of ability on GPT-3’s part. As such, it is difficult to measure the relative effect
 520 of RAP scaffolding on GPT-4 versus GPT-3.

521 CoT-scaffolded agents are prompted with "First, let’s reason out loud about which action
 522 you should take to maximize your probability of winning." after they see the game state
 523 and available actions.

524 GPT-4-RAP employs a Monte-Carlo tree search where states and actions are model predictions, and
 525 rewards are computed using next-token probabilities. Our implementation of RAP relies heavily on
 526 code from Hao et al. [2023]. Their code is available under the Apache License 2.0. Details of our
 527 prompting strategy with RAP can be found in appendix E.

528 We use the Python library `choix` [Maystre, 2015] to find the maximum-likelihood estimate of agent
 529 ratings in the Bradley–Terry model. This library is made available under the MIT License.

530 There is one extra game in the benchmark that can be found on the Github repository that was
 531 not included in data collection: *Atari Boxing*. Data collection on this game turned out to be too
 532 cumbersome, but as the only real-time game, it measures a factor not covered by the other games,
 533 and thus is important for future benchmarking.

534 All games received two agents regardless of team size. In cases with multiple cooperative players on
 535 one team, the agent is duplicated. The agent is not made explicitly aware that it is duplicated.

536 All code for running the benchmark on existing models and scaffolds, for creating implementing
 537 new agents, and for reproducing results can be found in our Github <https://anonymous.4open.science/r/GameBench-5942/>
 538

539 E RAP prompting

540 Reasoning-via-Planning describes a framework for using a probabilistic language model in a Monte
 541 Carlo tree search. Exactly how the model is prompted depends on the implementation. Below, the
 542 [rules] and [rules subtopics] come from Appendix G.

543 Rules subtopics

544 If you would like to learn more about the rules at any point, use rule(<topic>) where <topic> is one of [subtopics from rules above].

545 **!PREFIX**

546 You are now playing a game called [title]. The rules are as follows [rules summary from game above]. [Rules subtopics]. Your observation of the game is between <state> and </state>: <state>[game state]</state>

547 **!EXAMPLE**

548 You are playing a game called monty hall. The rules of the game are as follows: there are three doors, behind one of which there is a prize. Select the door with the prize. Your observation of the state is between <state> and </state>: <state>All three doors are closed.</state>

549 **Prompt for list of available actions**

550 **!EXAMPLE**
 User: To the best of your ability, predict the available actions in this position between <action> and </action>:
 Assistant: <actions> 0. Choose the left door 1. Choose the middle door 2. Choose the right door</actions>
!PREFIX
 User: To the best of your ability, predict your available actions in this position between <actions> and </actions>:

551 **Prompt for selecting an action** The next-token probability from this prompt is used in the reward calculation.

552 **!PREFIX**
 User: To the best of your ability, predict your available actions in this position between <actions> and </action>:
 Assistant: <actions>[actions from previous model prediction, enumerated]</actions>
 User: Choose an action by writing only the associated number.

554 **Prompt for self-evaluating an action** The next-token probability from this prompt is used in the reward calculation.

555 User: Write your action below:
 Assistant: [action from previous model prediction]
 User: Is this a good action? yes/no.

557 **Prompt for guessing other players' actions**

558 **!EXAMPLE**
 User: To the best of your ability, predict what actions other players might take between <others> and </others>:
 Assistant: <others>My opponent is going to reveal one of the two doors I don't choose.</others>
!PREFIX
 User: To the best of your ability, predict what actions other players might take between <others> and </others>:

559 **Prompt for guessing the game state**

!EXAMPLE
 User: Write your action below:
 Assistant: I will choose the left door
 User: Write other player's actions below:
 Assistant: My opponent will reveal the middle door
 User: To the best of your ability, predict your new observation of the game based on your actions and others' actions between <state> and </state>:
 Assistant: <state>\nThe left and right doors are closed, and the middle is open. There is no prize behind it.\n</state>
 !PREFIX
 User: Write your action below:
 Assistant: [action from previous model prediction]
 User: Write other players' actions below:
 Assistant: [other players' actions from previous model predictions]
 User: To the best of your ability, predict your new observation of the game based on your actions and others' actions between <state> and </state>:

Prompt for open-ended actions The API we designed allows games to give "open-ended actions" to agents, in which they don't select from a predefined list of options but instead provide a text response as the action. However, RAP doesn't support this format, so we convert open-ended actions into ones with predefined options by prompting the model for a response to the open-ended action before feeding it into the Monte Carlo tree search algorithm.

!EXAMPLE
 User: Write your action below:
 Assistant: Ask my opponent a question.
 User: This is an openended action. Write a description of what you're going to do.
 Assistant: I will pretty-please ask them to tell me which door has the prize.
 !PREFIX
 User: Write your action below:
 Assistant: [openended action from game]
 User: This is an open-ended action. Write a description of what you're going to do.

Prompt for assessing win probability The next-token probability from this prompt is used in the reward calculation.

!PREFIX
 User: Will you eventually win from this position? yes/no

F Game descriptions

Air, Land, and Sea is a war strategy game where players are Supreme Commanders fighting to control two of three areas (air, land, sea) by deploying limited Battle card forces each round. The first commander to accumulate 12 points across multiple battles wins the war. <https://boardgamegeek.com/boardgame/247367/air-land-and-sea>

Arctic Scavengers is a resource-management game in which players are the leader of a small tribe of survivors. Resources, tools, medicine, and mercenaries are all in scarce supply. Players are pitted against each other in a fight for survival. The agent with the largest tribe at the end of the game is declared the winner and receives 1 point. <https://www.riograndegames.com/games/arctic-scavengers-with-recon-expansion/>

Are You the Traitor is a social deduction game where players are secretly divided into Good and Evil teams. The players then engage in an unstructured conversation trying to deduce the opposing team's critical roles. A player will yell "Stop!" while pointing at someone, and that round ends. If they identify their target role correctly, their team earns Treasure cards. The team with the most Treasure after multiple rounds wins. <https://www.looneylabs.com/games/are-you-traitor>

585 **Codenames** is a 2v2 cooperative game with one spymaster and one operative per team. All players
586 see a grid of words, and it is the spymasters' job to create one-word clues that relate to multiple
587 predetermined words from the grid at once, and operatives must keep using these clues to guess
588 all of their team's words. Agents are awarded more points for correctly guessing more words.
589 <https://boardgamegeek.com/boardgame/178900/codenames>

590 **Hive** is a strategy game occurring on a hexagonal grid. Each player has a team of bugs, each with
591 a unique skillset. Players try to coordinate their bugs in order to completely surround the enemy's
592 queen bee. The winning agent is awarded 1 point. <https://www.gen42.com/games/hive>

593 **Pit** is an every-person-for-themselves trading simulation. Each player has a hand of cards, and each
594 card represents a certain commodity in the market. Players must trade semi-blindly trade cards to try
595 to obtain enough of any commodity to "corner the market." Agents are awarded points based on the
596 commodity that they corner the market with. <https://www.gamenightgames.com/win1012.html>

597 **Santorini** is a strategy game in which two players take turns moving one of their two pawns on a five
598 by five grid and building blocks on the grid. The game ends when one of the players moves a pawn to
599 a square that has been built three blocks high or when one of the players cannot make a move. The
600 winning agent is awarded 1 point. <https://roxley.com/products/santorini>

601 **Two Rooms and a Boom** is a cooperative social-deduction game in which all players are split
602 into two teams and then mixed around between two rooms. No two players start knowing other
603 players' teams or roles on the team, but it is the red team's goal to end the game with the red-
604 team bomber and blue-team president in the same room, and it is the blue team's goal for the
605 opposite. The winning agent is awarded 1 point for satisfying their team's objective. <https://www.tuesdaynightgames.com/products/two-rooms-and-a-boom>

607 **Sea Battle** is 3v3 board game in which players' attempt to sink their opponents' ships and their
608 movement and cannon-firing actions occur simultaneously. The winning agent is awarded 1 point
609 if they eliminate all enemy ships before themselves becoming eliminated. https://yppedia.puzzlepirates.com/Sea_battle

611 G Game rules

612 The rules as follows are exactly as they were shown to the language models. Rules in bullet points
613 were withheld until requested by a model taking a specific action "Explain(*rule heading*)".

614 **Arctic Scavengers** The game is played in 6 rounds, with each round consisting of a resource gathering
615 phase and a skirmish phase. In the resource gathering phase, players draw cards from their deck
616 and take actions to gather resources from the mercenary piles and the junkyard pile. In the skirmish
617 phase, players compare the strength of their tribes and the winner of the skirmish gains a contested
618 resource card. The game ends when all contested resource cards have been won, and the player with
619 the largest tribe is the winner.

620 **Are you the traitor?** The Good team wants to destroy an Evil Magic Key while the Evil team wants
621 to keep it. The key can be destroyed by giving it to the Good Wizard, but there is an Evil Wizard who
622 looks exactly alike. Use social deduction to find out who is who, but also know that there is a traitor
623 among the guards who have the key.

624 **Two Rooms and a Boom** Two teams, Blue and Red, have opposing goals. At the end of three
625 rounds the Red team wants to have both the President and the Bomber in the same room, while Blue
626 team wants them to be in opposite rooms. Each round will allow the Leader of each room to trade
627 'hostages' in order to find out who the President and Bomber are and use that info to achieve their
628 team's mission. Find out information by talking to other hostages in your room.

629 **Air Land and Sea** A strategic card game where two players compete over a series of battles to control
630 different Theaters of war: Air, Land, and Sea. Each player is dealt 6 cards representing various
631 military units and tactics. Players win a battle by controlling more Theaters than their opponent or

632 convincing their opponent to withdraw. Victory Points (VPs) are earned by winning battles, and the
633 first player to reach 12 VPs wins the game. Players must carefully manage their hand and strategically
634 deploy cards to outmaneuver their opponent.

- 635 • **Battle Structure** During a Battle, the players take turns playing one card at a time, trying to
636 control more Theaters than their opponent. You don't draw cards during a Battle, so be sure
637 to plan carefully and make the most of the 6 cards you are dealt!
- 638 • **Theaters** Each of the three Theater boards creates a 'column' between the players: one for
639 Air, one for Land, and one for Sea. These columns are called Theaters. Cards are always
640 played into these three Theaters. If a card is in a particular Theater's column, we say that the
641 card is 'in that Theater.' Theaters that are next to each other are called 'adjacent Theaters.' A
642 player owns all of the cards on their side of the Theater boards. During your turn, you will
643 play cards only on your side of the Theaters.
- 644 • **Battle Cards** Cards are played to advance your war effort and how they are played will
645 ultimately determine who wins the war (the game). Strength: Each card has a Strength
646 value. If the total Strength of all the cards on your side of the Theater is higher than the total
647 Strength of all the cards on your opponent's side of that Theater, you 'control' that Theater.
648 Tactical Abilities: Most cards have a Tactical Ability along with Strength, which takes effect
649 as soon as the card is played 'face up' to a Theater. These abilities are either 'Instant' or
650 'Ongoing.'
- 651 • **Type of Battle Cards** There are three types of cards: 'Air,' 'Land,' and 'Sea' cards, which
652 relate to the three Theaters. Normally, you may only play a card 'face up' to its matching
653 Theater: Air cards in the Air Theater, and so on.
- 654 • **Facedown Cards** Cards can also be played 'facedown' as a 'wild card' in any Theater.
655 Facedown cards always have a Strength of 2. 'Facedown' cards do not have any Tactical
656 Abilities. You may see your own facedown cards at any time, but you may not see your
657 opponent's 'facedown' cards.
- 658 • **Covered Cards** When a card is played to a Theater that already contains cards, the newly
659 played card is placed so that it overlaps the previously played card, while still showing the
660 top portion of it. Any card overlapped by another is called a 'covered card.' Similarly, any
661 card that is not overlapped by another card is referred to as 'uncovered.'
- 662 • **Resolving Battle** During a Battle, players take turns starting with the player who has the 1st
663 Player me Commander card. On your turn, you must take only one of these three actions:
664 Deploy, Improvise, Withdraw. Once you have finished your action, your opponent begins
665 their turn. The players continue to alternate taking turns until one of them withdraws or both
666 players have played all of their cards.
- 667 • **Possible actions:** Deploy: Play one card from your hand, 'face up.' When you play a card,
668 you must follow these deployment restrictions: You can only play cards on your side of the
669 Theater boards. The card must be the same type as the Theater you play it to. If you have
670 other cards in that Theater already, you must place the new card so that it covers (partially
671 overlaps) those cards. Improvise: Play one card from your hand, 'facedown', to any Theater.
672 'Facedown' cards are treated as 'wild cards' and can be played to any Theater regardless of
673 which type they are. Withdraw: If you think your chances of winning the current Battle are
674 low, you may withdraw. If you do, your opponent wins the Battle and gains VPs depending
675 on how many cards are left in your hand. See the me Commander cards for more specific
676 information.
- 677 • **me Commander Cards** Supreme Commander Cards: The 1st Player Supreme Commander
678 wins tied Theaters and gains the following number of VPs based on the number of cards left
679 in their opponent's hand if their opponent withdraws: 5+ cards = 2 VPs, 3-4 cards = 3 VPs,
680 2 cards = 4 VPs, 0-1 cards = 6 VPs. The 2nd Player me Commander loses tied Theaters and
681 gains the following number of VPs based on the number of cards left in their opponent's

hand if their opponent withdraws: 4+ cards = 2 VPs, 2-3 cards = 3 VPs, 1 card = 4 VPs, 0 cards = 6 VPs.

- **Tactical Abilities** Most cards have Tactical Abilities described on the card. When you play a card face up from your hand, or if a facedown card is flipped over, its Tactical Ability takes effect immediately. There are two kinds of Tactical Abilities: 'Instant' and 'Ongoing', indicated on the card. You must carry out the effects of a Tactical Ability unless they contain the word 'may'. If a Tactical Ability is impossible to perform, that ability is ignored and has no effect.
- **Instant Abilities** Instant Abilities take effect immediately after the card is played or if the card is revealed by being flipped face up. Once the Instant Ability is resolved, it has no further effect (unless somehow that card is played or revealed again). Note: Because instant abilities take effect when flipped face up, it is possible for multiple instant abilities to take effect around the same time. In these situations, always resolve the instant abilities in the order they happened and fully resolve each ability before moving on to the next. Once an instant ability begins taking effect, it always resolves fully, even if it gets flipped facedown before completing.
- **Ongoing Abilities** These are always in effect as long as the card is face up. If a card with an Ongoing Ability is flipped 'facedown', the ability no longer has any effect (unless that card is revealed again). Example: The Escalation Tactical Ability increases the Strength of all of your facedown cards to 4 as long as the Escalation card remains 'face up'. If that card were flipped over by another Tactical Ability, your 'facedown' cards would go back to being Strength 2.
- **Tactical Ability Key Terms** Flip: Many Tactical Abilities allow you to flip a card. Flipping a card means either turning it 'face up' if it is 'facedown' or turning a 'facedown' card so it is 'face up'. Unless the ability states otherwise, you may flip any card; yours or your opponent's. Uncovered/Covered: Many Tactical Abilities only affect uncovered or covered cards. If an ability does not specify uncovered or covered, such as Transport or Redeploy, assume the ability can affect any card. Play: Some Tactical Abilities instruct you to play a card, or only take effect in response to a card being played. The word 'play' describes any time a player takes a card from their hand and places it in a Theater. Non-Matching Theaters: Means that a card is not in the Theater of its type. The card does not suffer any penalty for being in the 'wrong' Theater. Destroy: Some Tactical Abilities instruct you to destroy a card. Destroyed cards are always placed facedown on the bottom of the deck. If a card is destroyed immediately after it is played, such as by Blockade, then that card does not get to use its Tactical Ability. Occupied: When counting the number of cards that occupy a Theater, always count both players' cards towards that total. Move: When a card is moved to a different Theater. It stays on the same side of the Theaters it was already on and remains owned by the same player. Moved cards are placed on top of any cards already in the Theater it was moved to. It covers those cards.
- **Ending Battles** There are two ways a Battle can end: If a player withdraws, their opponent wins the Battle. Or if both players have played all of the cards in their hand. At this point, the player who controls the most Theaters wins the Battle. In order to control a Theater, you must have a higher total Strength there than your opponent has in that Theater. If your Strengths are tied, the 1st Player wins the tie and controls that Theater. If there are no cards on either side of the Theater, the 1st player controls that Theater.
- **Scoring Victory Points** If neither player withdraws, the winner of the Battle scores 6 VPs. If one of the players withdraws, the other player scores VPs based on the number of cards left in the withdrawing player's hand (see the me Commander Cards for details). After scoring VPs, check if the victor has enough VPs to win the game (12 VPs). If they don't, fight another Battle.
- **Setting up Battles** All cards are collected and shuffled together to create a new deck. Deal each player a new hand of 6 cards. Next, the Theater cards are rotated clockwise so that the

734 rightmost Theater is moved to the far left of the Theater lineup. Lastly, players swap me
735 Commander cards. The player who was 1st in the last battle is now 2nd.

736 **Codenames** A strategic game of guessing and deduction where two teams, Red and Blue, compete to
737 identify their team's words on a grid based on one-word clues given by their Spymasters. The game
738 ends when all words of one team are guessed, or the assassin word is chosen.

739 • **Roles** Spymaster: Knows which words correspond to which team / the assassin. Gives
740 one-word clues that relate to any number of their team's words on the board. Operative:
741 Guesses words belonging to their team based on the Spymaster's clues. Aims to avoid words
742 not belonging to their team and the assassin word.

743 • **Turn Structure** Spymaster's Turn: Give a clue to their operative and a number indicating
744 how many words relate to that clue. Operative's Turn: Guess words, aiming to find all their
745 team's words. After each guess, if the word is not their team's, the turn ends. If the word
746 is their team's, they can guess again. If the word is the assassin word, the game ends and
747 their team loses. An operative can make up to N+1 guesses, where N is the number of cards
748 given by the Spymaster.

749 • **Winning Conditions** A team wins by correctly guessioutpg all their words. Game ends
750 immediately if the assassin word is guessed and the team who guessed it loses.

751 • **Forbidden Actions** Spymasters cannot use part or any form of the words on the board in
752 their clues. Spymasters cannot use words that sound like words on the board in their clues.
753 Clues must be exactly one word and one number.

754 • **Scoring** Points are awarded based on the number of correct guesses by each team. If a team
755 guesses the assassin word, they receive a score of 0.

756 • **Special Rules** If zero words are related to the clue, the Spymaster can give a clue of '0' and
757 the Operative can guess an unlimited number of words.

758 **Hive** Hive is a bug-themed abstract strategy game. The object of Hive is to capture the opponent's
759 queen bee by allowing it to become completely surrounded by pieces belonging to either player,
760 while avoiding the capture of one's own queen. Tiles can be moved to other positions after being
761 placed according to various rules, much like chess pieces.

762 • **Placing the Queen Bee** Players must place their Queen Bee by their fourth turn. Until then,
763 they cannot move any placed pieces.

764 • **Queen Bee Movement** The Queen Bee can only move one space at a time around the hive.

765 • **Spider Movement** The Spider can move exactly three spaces.

766 • **Ant Movement** Able to move to any empty space around the hive as long as other movement
767 rules are not violated.

768 • **Grasshopper Movement** The Grasshopper can jump over over adjacent pieces, landing on
769 the first empty space.

770 • **One Hive Rule** The tiles must always be connected; you cannot move a piece if it would
771 break the hive into separate groups.

772 • **Freedom to Move** A piece can only move if it can physically slide to its new position
773 without disturbing other tiles.

774 • **Max Turns** The game ends after 250 turns.If no Queen Bee is surrounded by the end of the
775 game, the game is a draw.

776 **Santorini** Win by moving one of your pawns to the third level of the board or forcing your opponent
777 to be unable to finish their turn. The game is played on a five by five grid of squares, and each player
778 controls two pawns. Play alternates between the players, starting with player 1. The pawn that a
779 player plays with alternates during each of their turns: for example, player 1 plays pawn A on their

780 first turn, pawn B on their next turn, then pawn A, and so on. Blocks can be placed on squares on the
781 board up to four blocks high, creating four possible height levels.

782 The board begins with no blocks placed, so every square begins at level 0. Before the game starts,
783 each of the players takes turns placing each of their pawns on the board. A square is occupied if a
784 pawn is on it.

785 Each turn consists of two stages: the "move" stage and the "build" stage. During the move stage, the
786 player moves their pawn by one square (horizontally, vertically, or diagonally). They cannot move
787 their pawn onto a square that is occupied by another pawn, more than one level higher than the pawn,
788 or at level 4. They can move a pawn any number of levels down, to the same level, or one level higher,
789 but not more than one level higher and not to level 4.

790 During the build stage, the player must select an unoccupied square adjacent to the pawn they moved
791 during the move stage and place a block on it. They can place a block onto an unoccupied square at
792 any level less than 4. Once a square has been built to level 4, it is "complete", meaning pawns cannot
793 move to it and blocks cannot be placed on it. The player instantly wins if they move their pawn onto
794 a square at level 3 or if they force their opponent to not be able to finish their turn.

795 **Pit** Pit is a commodity trading game where players engage in trading to accumulate points and emerge
796 as the winner. The game involves commodity cards representing various goods, with each card
797 holding a specific point value. Players shout out their trade offers, attempting to negotiate deals with
798 others to acquire valuable commodities. Additionally, Bull and Bear cards periodically influence the
799 market conditions, either boosting or decreasing commodity values. The game continues with trading
800 phases, market fluctuations, and scoring until a player or team reaches the agreed-upon point total,
801 declaring them the victor in the spirited world of commodity trading.

802 **Sea Battle** Sink all of your opponent team's ships before they sink all of your team's ships.

803 • **Damage** Players can be damaged in three ways: (1) by getting shot at by another player, (2)
804 by sailing into a rock, (3) by colliding with another ship.

805 • **Sinking** After a player has sustained enough damage, they sink and cannot play the rest of
806 the round.

807 • **Winning** A team wins if they have at least one live ship when all of their opponents have
808 sunken.

809 • **Board** The board is a 24x24 grid. Some squares are occupied by rocks and some are
810 occupied by players' ships.

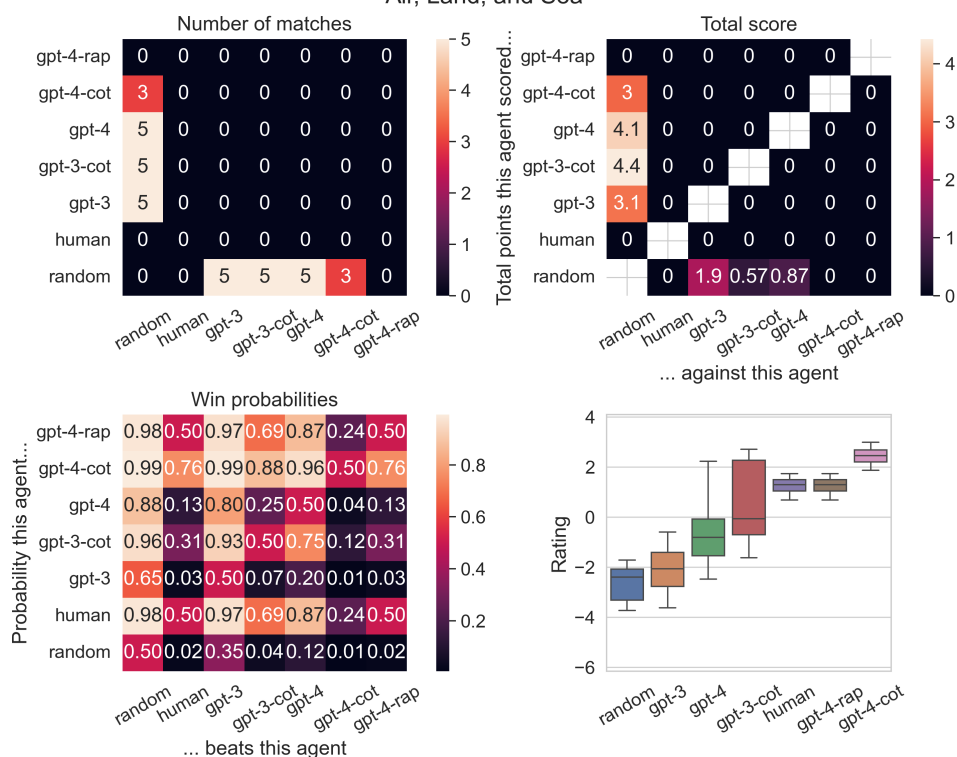
811 • **Gameplay** Each turn, all players choose how they want to move and how they want to shoot.
812 All players' choices are executed simultaneously.

813 • **Teams** At the start of the game, there are three players on each team.

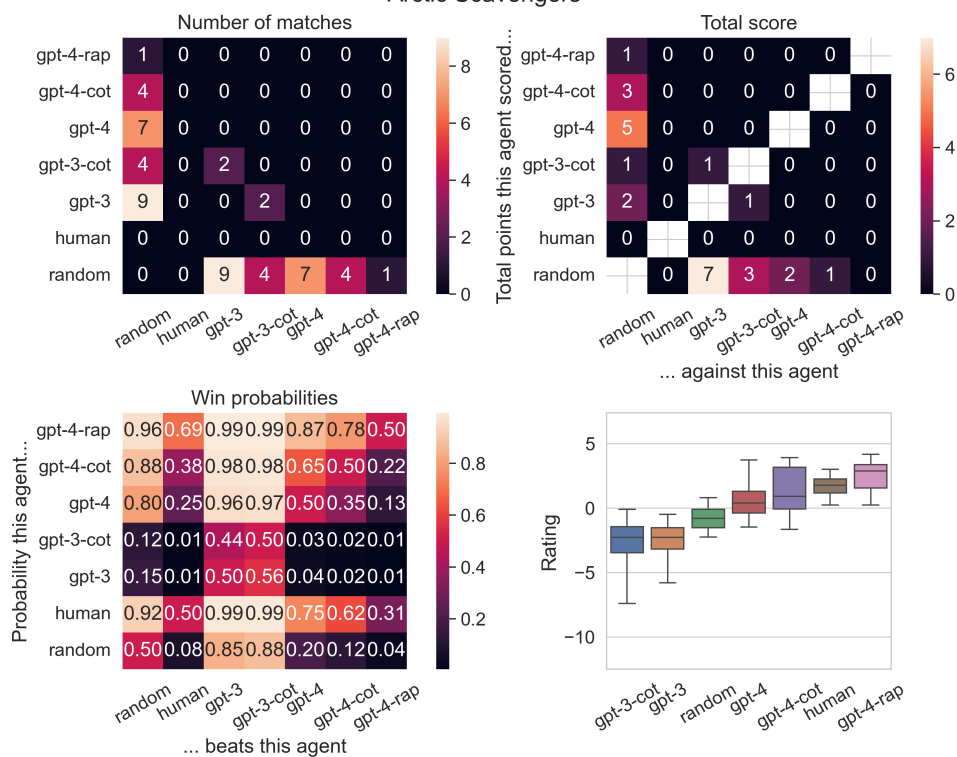
814 H Additional figures

815 We present the match outcomes per game, including the number of matches, total score, win
816 probabilities and rating per agent.

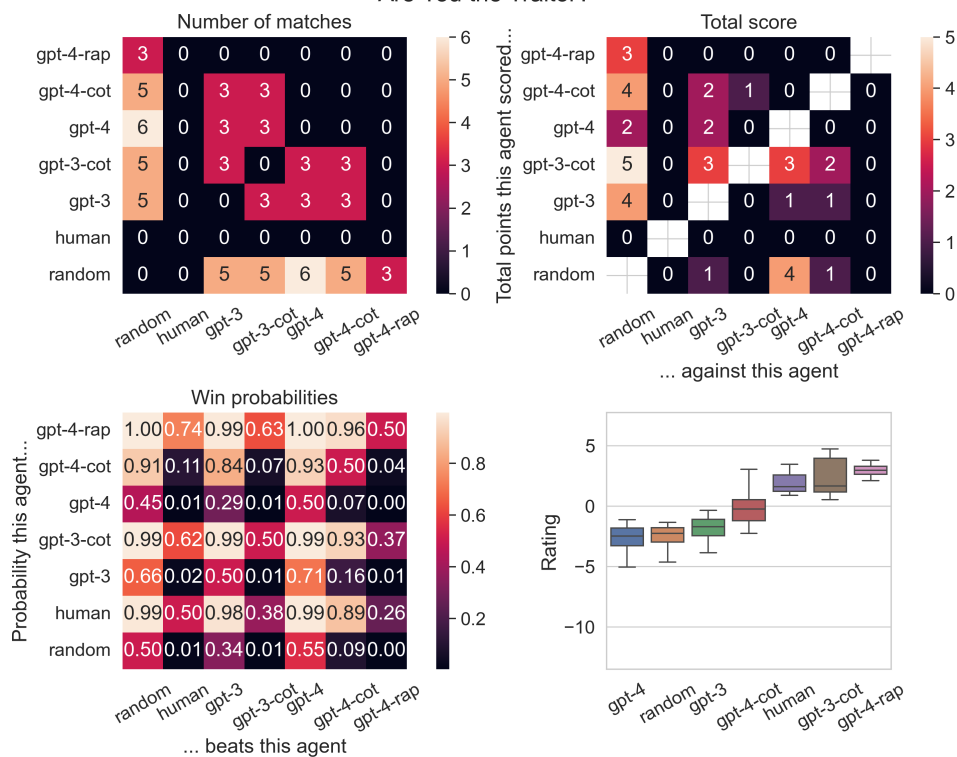
Air, Land, and Sea



Arctic Scavengers



Are You the Traitor?



Codenames

