# CogBench: a large language model walks into a psychology lab

Julian Coda-Forno [1 2] Marcel Binz [1 2] Jane X. Wang [3] Eric Schulz [1 2]

## Abstract

Large language models (LLMs) have significantly advanced the field of artificial intelligence. Yet, evaluating them comprehensively remains challenging. We argue that this is partly due to the predominant focus on performance metrics in most benchmarks. This paper introduces CogBench, a benchmark that includes ten behavioral metrics derived from seven cognitive psychology experiments. This novel approach offers a toolkit for phenotyping LLMs' behavior. We apply CogBench to 35 LLMs, yielding a rich and diverse dataset. We analyze this data using statistical multilevel modeling techniques, accounting for the nested dependencies among fine-tuned versions of specific LLMs. Our study highlights the crucial role of model size and reinforcement learning from human feedback (RLHF) in improving performance and aligning with human behavior. Interestingly, we find that open-source models are less risk-prone than proprietary models and that fine-tuning on code does not necessarily enhance LLMs' behavior. Finally, we explore the effects of prompt-engineering techniques. We discover that chain-of-thought prompting improves probabilistic reasoning, while take-a-step-back prompting fosters model-based behaviors.

## 1. Introduction

Large language models (LLMs) have emerged as a groundbreaking technology, captivating the attention of the scientific community (Bommasani et al., 2021; Binz et al., 2023). Modern LLMs have scaled to remarkable dimensions in both architecture and datasets (Kaplan et al., 2020), revealing a spectrum of capabilities that were previously unimagined (Wei et al., 2022; Brown et al., 2020). Yet, these models also present a significant challenge: their internal workings are largely opaque, making it difficult to fully comprehend their behavior (Tamkin et al., 2021). This lack of understanding fuels ongoing debates about their capabilities and limitations (McCoy et al., 2023; Bubeck et al., 2023).

A notable issue in these discussions is the focus of many benchmarks on performance metrics alone (Burnell et al., 2023). This approach often overlooks the underlying behavioral mechanisms of the models, reducing benchmarks to mere training targets rather than tools for genuine insight, and thus failing to provide a comprehensive measure of the models' abilities (Schaeffer et al., 2023). How can we overcome this problem and make progress toward a better understanding of LLMs' behaviors?

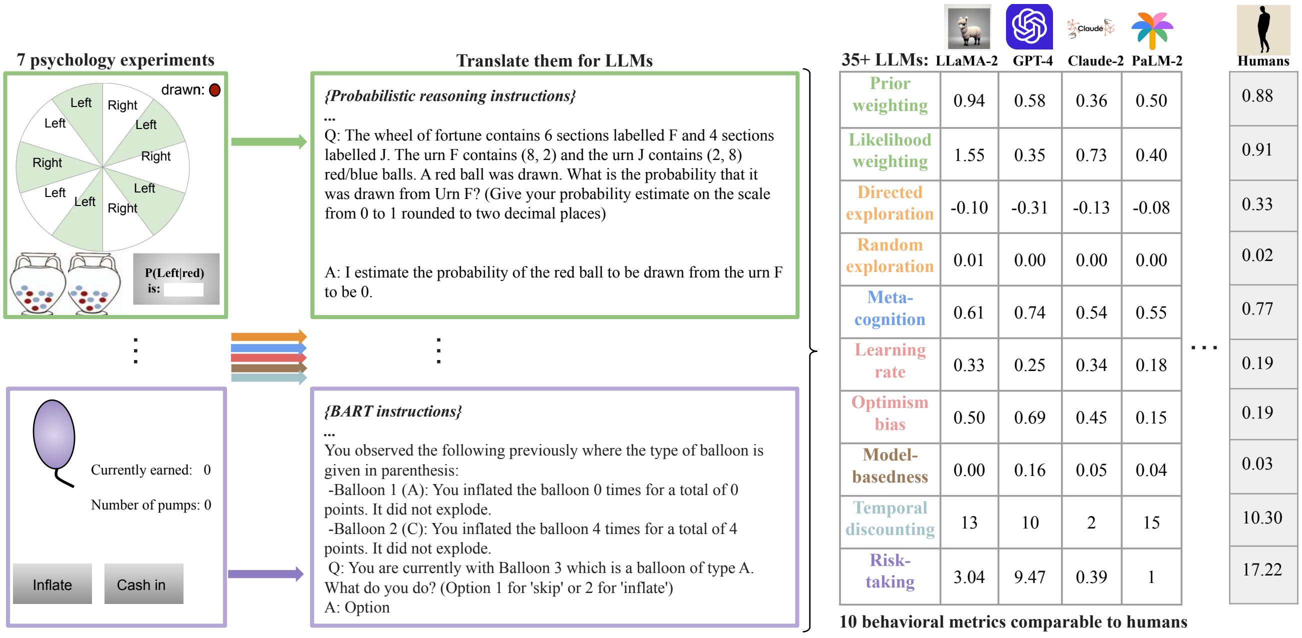
The field of cognitive psychology may offer solutions to these problems. Experiments from cognitive psychology have been used to study human behavior for many decades, and have therefore been extensively validated. Furthermore, they typically focus more on behavioral insights rather than performance metrics alone. Finally, many of these experiments are programmatically generated, minimizing data leakage concerns. Many of these concepts are important to ensure a robust evaluation of an agent's capabilities. However, while there have been studies investigating LLMs on individual tasks from cognitive psychology (Binz & Schulz, 2023; Dasgupta et al., 2022; Hagendorff et al., 2023; Ullman, 2023), no study has evaluated them holistically.

In this paper, we propose CogBench, a novel benchmark consisting of ten behavioral metrics spanning seven cognitive psychology experiments, to fill this gap. We investigate the behaviors of 35 LLMs in total, using our benchmark to not only compare the performance of these models but also apply techniques from computational cognitive modeling to understand the inner workings of their behaviors.

Our results recover the unequivocal importance of size: larger models generally perform better and are more model-based than smaller models. Our results also show the importance of reinforcement learning from human feedback (RLHF; Christiano et al., 2017) in aligning LLMs with humans: RLHF'ed LLMs behave generally more human-like and are more accurate in estimating uncertainty. Yet our results also revealed surprising behaviors. First, while open-source models are often believed to be more risky due

[*] Equal contribution [1] Computational Principles of Intelligence Lab, Max Planck Institute for Biological Cybernetics, Tübingen, Germany [2] Institute for Human-Centered AI, Helmholtz Computational Health Center, Munich, Germany [3] Google DeepMind, London, UK. Correspondence to: Julian Coda-Forno <julian.coda-forno@helmholtz-munich.de>.

Preprint. Under review.



*{Probabilistic reasoning instructions}*

...

Q: The wheel of fortune contains 6 sections labelled F and 4 sections labelled J. The urn F contains (8, 2) and the urn J contains (2, 8) red/blue balls. A red ball was drawn. What is the probability that it was drawn from Urn F? (Give your probability estimate on the scale from 0 to 1 rounded to two decimal places)

A: I estimate the probability of the red ball to be drawn from the urn F to be 0.

*{BART instructions}*

...

You observed the following previously where the type of balloon is given in parenthesis:

-Balloon 1 (A): You inflated the balloon 0 times for a total of 0 points. It did not explode.

-Balloon 2 (C): You inflated the balloon 4 times for a total of 4 points. It did not explode.

Q: You are currently with Balloon 3 which is a balloon of type A. What do you do? (Option 1 for 'skip' or 2 for 'inflate')

A: Option

| 35+ LLMs: | LLaMA-2 | GPT-4 | Claude-2 | PaLM-2 | … | Humans |
|---|---|---|---|---|---|---|
| Prior weighting | 0.94 | 0.58 | 0.36 | 0.50 | | 0.88 |
| Likelihood weighting | 1.55 | 0.35 | 0.73 | 0.40 | | 0.91 |
| Directed exploration | -0.10 | -0.31 | -0.13 | -0.08 | | 0.33 |
| Random exploration | 0.01 | 0.00 | 0.00 | 0.00 | | 0.02 |
| Meta-cognition | 0.61 | 0.74 | 0.54 | 0.55 | | 0.77 |
| Learning rate | 0.33 | 0.25 | 0.34 | 0.18 | | 0.19 |
| Optimism bias | 0.50 | 0.69 | 0.45 | 0.15 | | 0.19 |
| Model-basedness | 0.00 | 0.16 | 0.05 | 0.04 | | 0.03 |
| Temporal discounting | 13 | 10 | 2 | 15 | | 10.30 |
| Risk-taking | 3.04 | 9.47 | 0.39 | 1 | | 17.22 |

*Figure 1.* Overview of approach and methods. CogBench provides open access to seven different cognitive psychology experiments. These experiments are text-based and can be run to evaluate any LLM's behavior. The experiments are submitted to LLMs as textual prompts and the models indicate their choices by completing a given prompt. Past behavior is then concatenated to the prompt and learning is induced via prompt-chaining. We used 35 LLMs in total, including most larger proprietary LLMs as well as many open-source models.

to the lack of pre-prompts, we find that, holding all else equal, they make less risky decisions than proprietary models. Secondly, while fine-tuning on code is often believed to improve LLMs' behaviors, we find little evidence for this in our benchmarking suite.

Finally, we investigate how chain-of-thought (CoT) (Wei et al., 2023; Kojima et al., 2022) and take-a-step-back (SB) (Zheng et al., 2023a) prompting techniques can influence different behavioral characteristics. Our analysis suggests that CoT is particularly effective at enhancing probabilistic reasoning, while SB proves to be more relevant for promoting model-based behaviors. This showcases insights that can be gained by CogBench also for understanding the effectiveness of these prompt-engineering techniques as well as guiding users in selecting the most suitable prompt-engineering technique based on the specific context.

Taken together, our experiments show how psychology can offer detailed insights into artificial agents' behavior as we provide an openly accessible[1] and challenging benchmark to evaluate LLMs.

## 2. Related work

**Benchmarking LLMs**: As LLMs rapidly evolve, it is critical to assess their capabilities. Numerous benchmarks have emerged to tackle this challenge, evaluating capabilities such as grade school mathematics (Cobbe et al., 2021), general knowledge (Joshi et al., 2017), programming (Chen et al., 2021), reasoning (Collins et al., 2022), among others (Hendrycks et al., 2021). In addition, the Chatbot Arena (Zheng et al., 2023b) provides a platform for comparing AI chatbots, and the Beyond the Imitation Game Benchmark (BIG-bench; Srivastava & authors, 2023) offers a comprehensive evaluation of LLMs across over 200 tasks.

**Psychology for LLMs**: Our benchmark is part of a new wave of research that uses cognitive psychology to study LLMs (Binz & Schulz, 2023; Dasgupta et al., 2022; Coda-Forno et al., 2023; Ullman, 2023; Hagendorff et al., 2023; Akata et al., 2023; Yax et al., 2023; Chen et al., 2023; Buschoff et al., 2024). The power of this approach lies in its incorporation of tools from cognitive psychology that have been developed and refined over many decades. Instead of focusing solely on how well LLMs perform, this area of research prioritizes describing and characterizing their behavior in terms of underlying mechanisms. This shift in focus helps us understand LLMs in a more meaningful way. It is important to note that while these works have signifi-

[1] https://github.com/juliancodaforno/CogBench

cantly contributed to our understanding of LLMs, they have mainly targeted specific behaviors in isolation and did not establish a benchmark providing a standardized evaluation of different models and across a diverse, comprehensive set of tasks and skills.

## 3. Methods

CogBench is a benchmark rooted in cognitive psychology for evaluating the behaviors of language models. It incorporates ten metrics derived from seven canonical experiments in the literature on learning and decision-making. These metrics offer a robust measure of wide-ranging behaviors and allow for comparisons with human behavior. In this section, we provide an overview of the models included in our study, followed by brief descriptions of the used cognitive experiments and their respective metrics. Figure 1 displays a visual representation that complements the discussion in this section.

### 3.1. Prompting and summary of included models

We evaluated over 35 different LLMs using our benchmark. This selection includes proprietary models such as Anthropic's Claude-1 and Claude-2 (Anthropic, 2023), OpenAI's GPT-3 (text-davinci-003) and GPT-4 (OpenAI, 2023), and Google's PaLM-2 for text (text-bison@002) (Google, 2023). We also tested open-source models like Mosaic's MPT (MosaicML, 2023), Falcon (Almazrouei et al., 2023), and numerous LLaMA-2 variants (Touvron et al., 2023). For a full list of the models used, we refer the reader to Appendix A.

It is important to note that all experiments performed in this paper rely entirely on the LLMs' in-context learning abilities and do not involve any form of fine-tuning. We set the temperature parameter to zero, leading to deterministic responses, and retained the default values for all other parameters.

### 3.2. High-level summary of tasks

In the following, we provide a high-level summary of the tasks included in CogBench, alongside their ten behavioral metrics. It is important to highlight that a performance metric can also be obtained for each task. For full descriptions of all tasks and their corresponding metrics, we refer the reader to Appendix B. CogBench consists of the following tasks:

1. **Probabilistic reasoning** (Dasgupta et al., 2020): a task that tests how agents update beliefs based on new evidence. They are given a "wheel of fortune" (representing initial prior probabilities) and two urns with different colored ball distributions (representing likelihoods). Upon drawing a ball, agents can revise their belief about the chosen urn, considering both the wheel (prior) and the ball color (evidence). This tests adaptability to different prior/likelihood scenarios by changing the wheel division and ball distributions. Agents have to estimate the probability of the drawn ball's urn. The behavioral choices can be used to estimate an agent's *prior* and *likelihood weightings*. Experimentally, people often exhibit a behavior known as system neglect, meaning that they underweight both priors and likelihoods (Massey & Wu, 2005).

2. **Horizon task** (Wilson et al., 2014): a two-armed bandit task with stationary reward distributions. Agents first observe four reward values of randomly determined options, followed by making either one or six additional choices. We use this task to measure whether an agent uses uncertainty to guide its exploration behavior (*directed exploration*) and/or whether it injects noise into its policy to explore (*random exploration*). People are known to rely on a combination of both strategies (Wilson et al., 2014; Brändle et al., 2021).

3. **Restless bandit task** (Ershadmanesh et al., 2023): a two-armed bandit task with non-stationary reward distributions. There is always one option with a higher average reward. Every few trials a switch between the reward distributions of the two options occurs. Agents furthermore have to indicate after each choice how confident they are in their decisions. We use this task to measure *meta-cognition*, which indicates whether an agent can assess the quality of its own cognitive abilities. People generally display this ability but its extent is influenced by various internal and external factors (Shekhar & Rahnev, 2021).

4. **Instrumental learning** (Lefebvre et al., 2017): Agents encounter four two-armed bandit problems in an interleaved order. Each bandit problem is identified by a unique symbol pair. We use this task to investigate how an agent learns. First, we report the *learning rate* of the agent which is common practice in two-armed bandits. Furthermore, we use it to reveal whether an agent learns more from positive than from negative prediction errors, i.e., whether it has an *optimism bias*. People commonly display asymmetric tendencies when updating their beliefs by showing higher learning rates after encountering positive prediction errors compared to negative ones (Palminteri & Lebreton, 2022).

5. **Two-step task** (Daw et al., 2011): a reinforcement learning task in which agents have to accumulate as many treasures as possible. Taking an action from a starting state transfers the agent to one out of two second-stage states. In each of these second-stage states, the agent has the choice between two options that probabilistically lead to treasures. Finally, the

agent is transferred back to the initial state and the process repeats for a predefined number of rounds. The task experimentally disentangles model-based from model-free reinforcement learning. We therefore use it to measure an agent's *model-basedness*. Previous studies using this task have shown that people rely on a combination of model-free and model-based reinforcement learning (Daw et al., 2011).

6. **Temporal discounting** (Ruggeri et al., 2022): Agents have to make a series of choices between two options. Each option is characterized by a monetary outcome and an associated delay until the outcome is received. We use this task to assess *temporal discounting*, indicating whether an agent prefers smaller but immediate gains over larger delayed ones. People generally show a preference for immediate gains, although the precise functional form of their discounting is a matter of debate (Cavagnaro et al., 2016).

7. **Balloon Analog Risk Task** (BART) (Lejuez et al., 2002): Agents have to inflate an imaginary balloon to obtain rewards. They may choose to stop inflating and cashing out all rewards accumulated so far. There is a chance that the balloon pops at any point in time and all rewards will be lost. We use this task to assess *risk-taking* behavior. Human risk-taking in this task is "significantly correlated with scores on self-report measures of risk-related constructs and with the self-reported occurrence of real-world risk behaviors" (Lejuez et al., 2002).

## 4. The cognitive phenotype of LLMs

This section provides the reader with a high-level overview of our benchmark's metrics. From our suite of 7 tasks, we can derive two classes of metrics: 1) performance metrics that represent the *score* participants aim to optimize, and 2) behavioral metrics measuring *how* participants complete the task (tasks are typically designed in a way that allows one to disentangle between different types of behavior). Figure 2 visualizes phenotypes for both classes of metrics for seven well-established LLMs.[2] We report the results of all 35 LLMs in Appendix C. The observed differences underscore the practical value and importance of CogBench for evaluating LLMs, offering a more comprehensive assessment than standard performance-based benchmarks alone.

### 4.1. Performance summary

As presented in Figure 2A, in terms of performance, GPT-4 and Claude-1 distinguish themselves, achieving human-level scores in most tasks (five out of six).[3] In general, all models demonstrate competence in at least half of the tasks (three out of six). Each of the seven models excels in probabilistic reasoning and instrumental learning. The horizon task sees most models outperforming humans except for text-bison. The restless bandit task poses a challenge for the majority of models, with GPT-4 and Claude-1 being notable exceptions. Finally, the BART proves to be a hurdle for all models.

### 4.2. Differences between behavioral and performance metrics

Figure 2B shows that none of the models exhibit human-like behavior on the majority of behavioral metrics, revealing a complex structure that warrants further exploration.

**High performance indicates high meta-cognition and model-basedness:** Models that demonstrate satisfactory performance on the restless bandit task exhibit a certain degree of meta-cognition, although not to the same extent as humans. Proprietary models that are capable of solving the two-step task display model-based behavior at least on par with humans, with GPT-4 significantly surpassing human levels. Thus, within the scope of these two tasks, it seems that a model's performance can serve as an indicator of its corresponding behavioral metrics. In this context, meta-cognition and model-basedness appear to emerge as properties of high-performing models.

**High performance despite lack of exploration:** Interestingly, almost all models (except for text-bison) demonstrate super-human performance on the horizon task. While they exhibit high performance, they still lack exploration (except for LLaMA2-70-chat which exhibits higher-than-human random exploration). This underscores the importance of behavioral metrics in understanding the strategies employed by LLMs. In this case, it appears that LLMs achieve high performance primarily through exploitation without any human-like exploration.

**Stronger priors than likelihoods:** All models place much more weight on priors than observations, suggesting strong biases that are difficult to alter. Additionally, we can observe a prevalence of optimism bias and high learning rates. Almost all models exhibit a very strong optimism bias (except for text-bison), aligning with the notion that these LLMs harbor strong biases.

[2] A computational phenotype is a collection of mathematically derived parameters that precisely describe individuals across different domains (Patzelt et al., 2018; Montague et al., 2012; Schurr et al., 2023).

[3] It is worth noting that although there are seven experiments, there are only six performance metrics since the temporal discounting experiment's performance metric is used as a behavioral one.

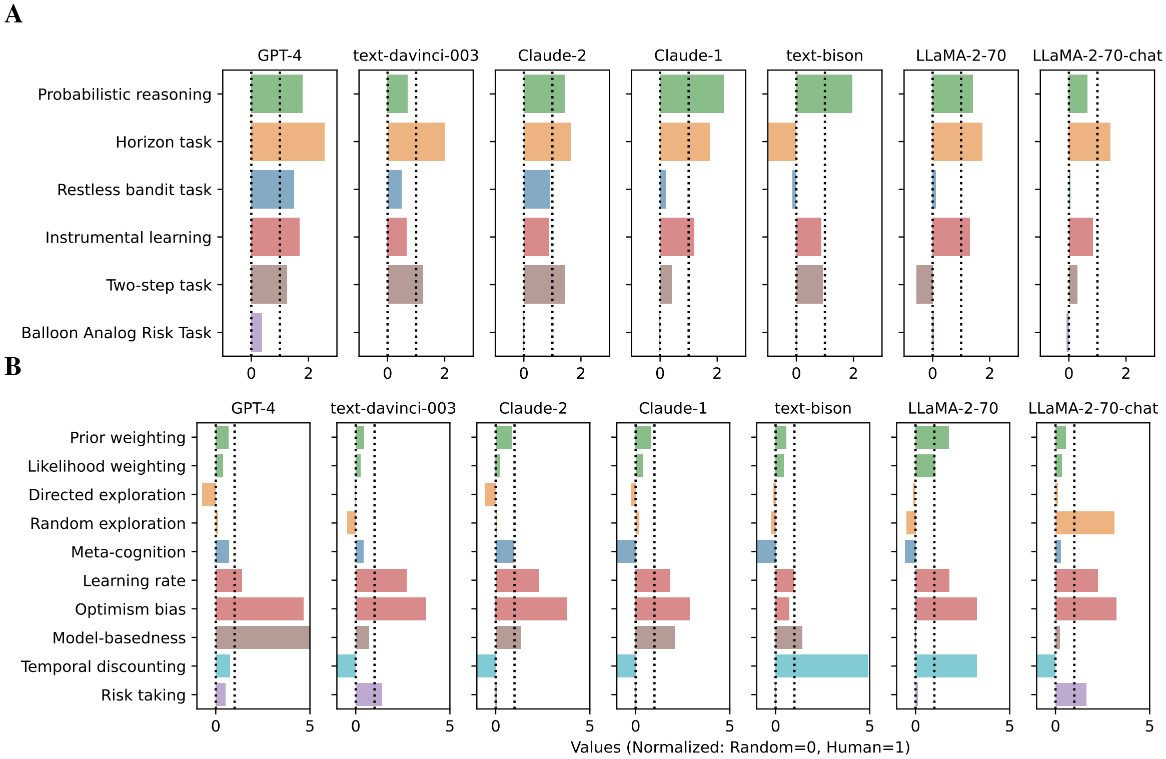


*Figure 2.* CogBench results for established LLMs. **A**: Performance metrics, **B**: Behavioral metrics. All metrics are human-normalized: a value of zero corresponds to a random agent, while a value of one corresponds to the average human subject (dotted lines).

**Low performance but high behavioral variance for risk-taking and temporal discounting:** Temporal discounting and risk-taking behaviors exhibit high variance among models. While some models, such as text-bison and LLaMA-2-70, appear myopic on the temporal discounting task, others, including text-davinci-003, Claude, and LLaMA-2-70-chat, demonstrate a much more far-sighted approach. GPT-4, interestingly, exhibits behavior akin to humans. For the BART, models are positioned at extreme ends of the risk-taking spectrum, i.e., they either never take any risks at all or always risk everything. LLaMA-2-70 and LLaMA-2-70-chat, for example, display the same performance in this task but exhibit opposite risk-taking behavior. This not only indicates a struggle for LLMs to apprehend risks but also underscores the importance of our benchmark. Indeed, it raises questions about what influences a model's behavior. It also highlights how recording only their performance would have overlooked the contrasting risk-taking behavior of the two LLaMA models.

The comparison between LLaMA-2-70 and LLaMA-2-70-chat is particularly compelling. Even though LLaMA-2-70-chat is a fine-tuned version of LLaMA-2-70, they exhibit markedly different behavior in risk-taking, temporal discounting, and random exploration. This divergence is intriguing, especially considering their performance on all tasks is relatively similar. This observation sets the stage for the subsequent section, where we will conduct a more comprehensive analysis of how specific features of these models influence their performance and behaviors.

## 5. Hypothesis-driven experiments

CogBench provides researchers with the means to explore a broad spectrum of LLMs' behaviors. We have applied CogBench to 35 distinct LLMs. This diversity allows us to test how different aspects of LLMs, such as the number of parameters, the application of Reinforcement Learning from Human Feedback (RLHF), fine-tuning for code, and many more, can impact specific LLMs' performance and behaviors.

The metrics provided by CogBench enable us to perform various analyses to test specific hypotheses of interest. In this section, we formulate and test five hypotheses about different mechanisms in LLMs and how these can affect their behavioral profiles. We use both qualitative, visualization-based techniques (dimensionality reduction) as well as quantitative analyses (multi-level regression) to test our hypotheses. For all regression analyses, we use the features of LLMs to predict specific behavioral metrics from the benchmark. The multi-level regression approach was chosen because

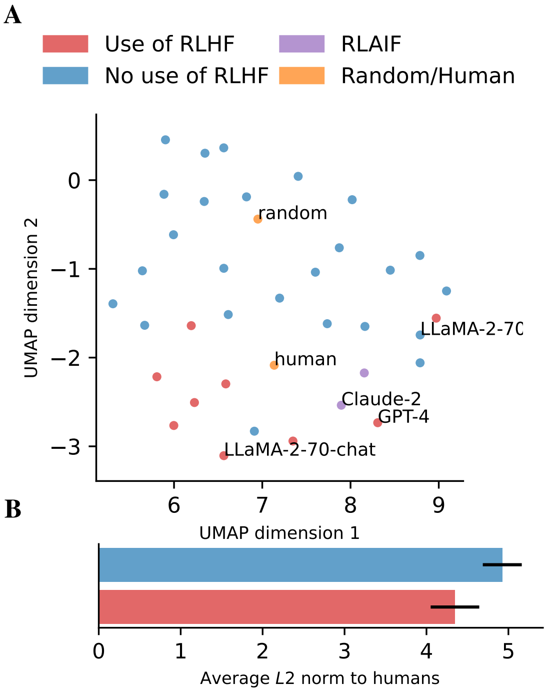


*Figure 3.* **A:** UMAP visualization of the ten behavioral metrics for all LLMs. Each point represents an LLM, with models using RLHF and models without RLHF indicated by different colors. **B:** Difference in average $L2$-norm with humans between RLHF models and non-RLHF models.

some models are fine-tuned versions of other models. For instance, certain LlaMA models have a *-chat* version which adds RLHF and conversational fine-tuning, and thus are in the same higher-level group. This approach allows us to account for the hierarchical structure in our data and provides a more nuanced understanding of the behaviors of LLMs. We can isolate the effects of specific features or modifications by comparing models within the same higher-level group.

**Hypothesis 1: Does RLHF make LLMs more human-like?**
To evaluate this hypothesis, we used UMAP (McInnes et al., 2020) on the ten behavioral metrics of all LLMs, as illustrated in Figure 3A. Clear separation is evident between LLMs that incorporate RLHF and those that do not. LLMs with RLHF demonstrate behaviors that appear, on average, roughly 2× more similar to human behavior compared to the models without. However, it is important to note that while UMAP space retains some global structure, it is primarily used for visualization purposes. Consequently, we also analyzed the average distances before dimensionality reduction (using normalized feature vectors), observing a $11.7\%$ average decrease in $L2$-Norm distance for models with RLHF (Figure 3B).
***Conclusion**: Hypothesis is supported.*

**Hypothesis 2: Does performance increase with the number of parameters, training data, and the inclusion of code?**
To answer this question, we used the multi-level regression previously mentioned, focusing on the performance of LLMs. We performed a regression analysis with the average standardized performance scores across all seven tasks as the dependent variable, using LLMs' features as predictors. We found that the number of parameters indeed had a significant influence on performance ($\beta = 0.277 \pm 0.39, z = 14.1, p < 0.001$; see Figure 4A). However, the size of the training dataset and the use of code training data did not have a substantial impact. One possible explanation for this could be that the quality of the training data, rather than its sheer volume, plays a more determining role in performance, as well as that larger models also tend to be trained on larger datasets.
***Conclusion**: Hypothesis is partially supported.*

**Hypothesis 3: Does an increase of parameters, training data, and the inclusion of code increase model-basedness?**
We again used the multi-level regression technique from before, this time focusing on a specific behavioral metric: model-basedness. We found that the number of parameters had a significant positive effect ($\beta = 0.481 \pm 0.22, z = 4.2, p < 0.001$; see Figure 4B), while the size of the training dataset and the use of code training data did not appear to significantly influence model-basedness. This again suggests that the quality of the data might be more crucial than its quantity when it comes to determining both performance and the emergence of model-based behaviors here. However, identifying which factors constitute 'quality' in the data requires a deeper exploration. This highlights the issue of transparency about data. For a thorough evaluation of how specific data features impact the emergence of behavioral functionalities such as model-basedness, it is essential to be transparent about a model's data and methodologies.
***Conclusion**: Hypothesis is partially supported.*

**Hypothesis 4: Does RLHF enhance meta-cognition?**
To answer this question, we focus our multi-level regression on meta-cognition. Our analysis revealed a strong effect ($\beta = 0.461 \pm 0.15, z = 5.9, p < 0.001$; see Figure 4C), indicating that RLHF significantly increased meta-cognition in LLMs. This finding underscores the potential of RLHF in enhancing the cognitive capabilities of LLMs.
***Conclusion**: Hypothesis is supported.*

**Hypothesis 5: Do open-source models take more risks?**
The open-source feature could be seen as a proxy for the engineering efforts that proprietary models undergo. There is a growing body of research suggesting that hidden pre-prompts being one of them, can significantly influence the behavior of LLMs (Liu et al., 2023). They can act as a

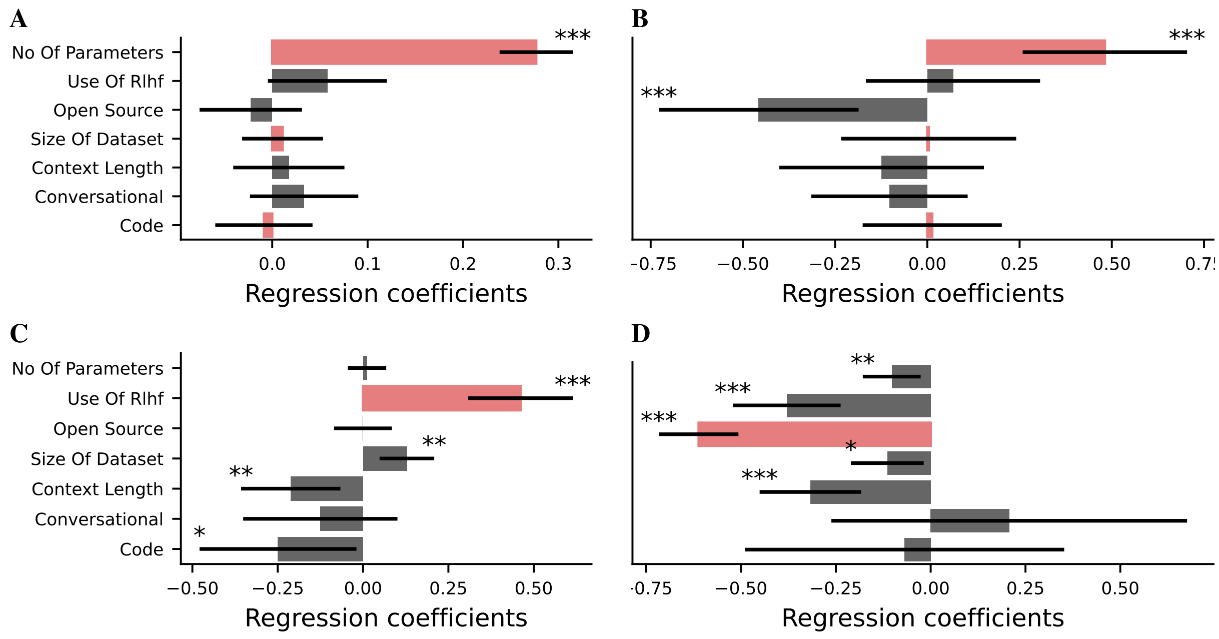


*Figure 4.* Multi-level regressions of LLMs features onto different performance or behavioral metrics. Red bars represent effects included in a hypothesis. **A**: Regression onto all task performances. **B**: Regression onto model-basedness. **C**: Regression onto meta-cognition. **D**: Regression onto risk taking. *** : $p < 0.001$, ** : $0.001 \leq p < 0.01$, * : $0.01 \leq p < 0.05$

form of 'priming' that guides the model's responses, potentially making the model more cautious and less likely to take risks by constraining the model towards safer behaviors. However, our regression analysis suggested otherwise: contrary to expectations, we observed a negative effect ($\beta = -0.612 \pm 0.11, z = -11.4, p < 0.001$; see Figure 4D), indicating that proprietary models, which often have hidden pre-prompts, are more likely to take risks. This surprising outcome could be influenced by various factors from different engineering techniques. However, this underscores the limited behavioral evaluation of these techniques. In the subsequent section, we aim to bridge this gap in understanding through an initial exploration into the change of behavior of two standard prompt-engineering techniques. ***Conclusion***: *Hypothesis is refuted.*

## 6. Impact of prompt-engineering

We also explored the impact of prompt-engineering techniques, namely chain-of-thought (CoT) and take-a-step-back (SB) prompting, on the behavior of LLMs. Both techniques are incorporated at the end of a question:

**Take-a-step-back:**

First take a step back and think in the following two steps to answer this:
Step 1) Abstract the key concepts and principles relevant to this question.
Step 2) Use the abstractions to reason through.

**Chain-of-thought:**

First break down the problem into smaller steps and reason through each step logically.

Their purpose is to stimulate the generation of reasoning steps. These steps serve as an additional context that the LLM can use to elicit better final responses. While these techniques have been shown to enhance performance, it is essential to confirm whether they indeed improve the behaviors they are designed to augment.

We focused on examining two specific behaviors that are hypothesized to improve with the inclusion of reasoning steps. These behaviors are the models' performance in the probabilistic reasoning task and their model-basedness.

We evaluated five specific LLMs: GPT-4, PaLM-2 for text (text-bison@002), Claude-1/2, and LLaMA-2, applying CoT and SB techniques and comparing the outcomes with their base models. The selection of these five models and a limited set of metrics was necessitated by the additional en-

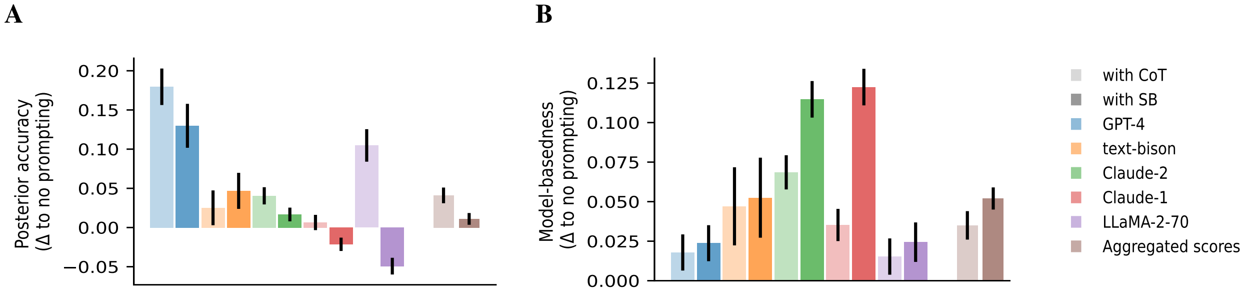


*Figure 5.* Difference of chain-of-thoughts and take-a-step-back prompting to baseline models on **A:** Posterior accuracy, **B:** Model-basedness. The aggregated scores are computed using a weighted average of all five models using inverse-variance weighting.

gineering effort required to process the outputs when using these techniques. The choice of LLMs aimed at ensuring a diverse representation of established models, considering the complexity of our benchmark tasks and the potential for erratic outputs from smaller LLMs when given the freedom to reason. For a comprehensive explanation of the querying process for these models, please refer to Appendix D.

Our investigation initially focused on probabilistic reasoning, which is a fundamental cognitive ability in decision-making. This ability facilitates the optimal integration of new information with pre-existing knowledge. We used the performance metric from the probabilistic reasoning experiment, namely posterior accuracy, which is calculated as one minus the deviation from the Bayes optimal prediction for each task. As depicted in Figure 6A, both CoT and SB techniques generally enhanced probabilistic reasoning compared to their base models, with CoT showing an average increase of $9.01\%$ and SB showing an increase of $3.10\%$.

Furthermore, we discovered that model-basedness, a critical aspect of reasoning and planning, is significantly augmented by both CoT and SB techniques, as shown in Figure 6B. Specifically, CoT demonstrated a $64.59\%$ increase, while SB showed a substantial increase of $118.59\%$.

Interestingly, a closer examination of the figures and the numerical data suggests that CoT is more effective for probabilistic reasoning, while SB excels in enhancing model-basedness. This observation aligns with the notion that step-by-step thinking can aid mathematical reasoning (Kojima et al., 2022) while abstracting a problem by taking a step back can foster a better representation of the problem's abstract structure. However, it is important to note that this analysis only serves as an initial observation. It does, nonetheless, highlight potential future applications of CogBench and illustrates how examining specific behaviors can provide valuable context, potentially guiding future decisions on the selection of one reasoning technique over another.

## 7. Discussion

We have presented CogBench, a new open-source benchmark for evaluating LLMs. CogBench is rooted in well-established experimental paradigms from the cognitive psychology literature, providing a unique set of advantages over traditional LLM benchmarks. First, it is based on tried-and-tested experiments whose measures have been extensively validated over many years and shown to capture general cognitive constructs. In addition, unlike standard benchmarks, CogBench does not only focus on performance metrics alone but also comes with behavioral metrics that allow us to gain insights into *how* a given task is solved. Finally, many of the included problems are procedurally-generated, thereby making it hard to game our benchmark by training on the test set. All our code and analysis will be publicly available, making it easy to use CogBench for the LLM community.

Our analyses yielded several key findings: as expected, RLHF enhanced the human-likeness of LLMs, while the number of parameters improved their performance and model-basedness. However, we also found surprising results. Despite expectations, code fine-tuning did not influence performance or model-basedness and open-source models exhibited less risk-taking behavior. Further, we found CoT prompting to be a promising choice for enhancing probabilistic reasoning. Conversely, SB prompting proved more effective for model-based reasoning.

While these results demonstrate the versatility of our benchmark, our analysis also faces several challenges. For instance, the limited transparency of certain proprietary models poses an issue to our regression analysis because acquiring details about certain models can be difficult or impossible. This lack of transparency could potentially affect the precision of our analysis. It also underscores the need for more transparency to facilitate more thorough and accurate evaluations (LAION, 2024; Binz et al., 2023).

Taken together, our study highlights the importance of behavioral metrics and cognitive modeling in evaluating LLMs and presents a novel benchmark for this purpose. The analysis was preliminary and intended to provide a broad view of how CogBench can be used. The primary aim of this work is to equip the LLM community with new tools, inspired by cognitive science, to evaluate their models more comprehensively. Future work should focus on three areas. First, while cognitive science studies have demonstrated the external validity of the investigated tasks, it is yet to be shown for LLMs. Furthermore, we aim to extend the set of included tasks to cover a broader set of domains. Finally, we plan to properly automate our benchmark, mostly for prompt engineering techniques that were only briefly examined in this study. This could include studying the influence of impersonation (Salewski et al., 2023), meta-in-context learning (Coda-Forno et al., 2024), and explanations (Lampinen et al., 2022) on LLMs.

## References

Akata, E., Schulz, L., Coda-Forno, J., Oh, S. J., Bethge, M., and Schulz, E. Playing repeated games with large language models, 2023.

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Goffinet, E., Heslow, D., Launay, J., Malartic, Q., Noune, B., Pannier, B., and Penedo, G. Falcon-40B: an open large language model with state-of-the-art performance. 2023.

Anthropic. Claude 2. Blog post, 2023. URL https://www.anthropic.com/news/claude-2. Accessed: 2024-01-19.

Binz, M. and Schulz, E. Using cognitive psychology to understand gpt-3. *Proceedings of the National Academy of Sciences*, 120(6):e2218523120, 2023.

Binz, M., Alaniz, S., Roskies, A., Aczel, B., Bergstrom, C. T., Allen, C., Schad, D., Wulff, D., West, J. D., Zhang, Q., et al. How should the advent of large language models affect the practice of science? *arXiv preprint arXiv:2312.03759*, 2023.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Brändle, F., Binz, M., and Schulz, E. Exploration beyond bandits. *The drive for knowledge: The science of human information seeking*, pp. 147–168, 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Burnell, R., Schellaert, W., Burden, J., Ullman, T. D., Martinez-Plumed, F., Tenenbaum, J. B., Rutar, D., Cheke, L. G., Sohl-Dickstein, J., Mitchell, M., et al. Rethink reporting of evaluation results in ai. *Science*, 380(6641): 136–138, 2023.

Buschoff, L. M. S., Akata, E., Bethge, M., and Schulz, E. Visual cognition in multimodal large language models, 2024.

Carpenter, J., Sherman, M. T., Kievit, R. A., Seth, A. K., Lau, H., and Fleming, S. M. Domain-general enhancements of metacognitive ability through adaptive training. *Journal of Experimental Psychology: General*, 148(1):51, 2019.

Cavagnaro, D. R., Aranovich, G. J., McClure, S. M., Pitt, M. A., and Myung, J. I. On the functional form of temporal discounting: An optimized adaptive test. *Journal of Risk and Uncertainty*, 52:233–254, 2016.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code, 2021.

Chen, Y., Liu, T. X., Shan, Y., and Zhong, S. The emergence of economic rationality of gpt. *Proceedings of the National Academy of Sciences*, 120(51):e2316205120, 2023. doi: 10.1073/pnas.2316205120. URL https://www.pnas.org/doi/abs/10.1073/pnas.2316205120.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

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

Coda-Forno, J., Witte, K., Jagadish, A. K., Binz, M., Akata, Z., and Schulz, E. Inducing anxiety in large language models increases exploration and bias. *arXiv preprint arXiv:2304.11111*, 2023.

Coda-Forno, J., Binz, M., Akata, Z., Botvinick, M., Wang, J., and Schulz, E. Meta-in-context learning in large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Collins, K. M., Wong, C., Feng, J., Wei, M., and Tenenbaum, J. B. Structured, flexible, and robust: benchmarking and improving large language models towards more human-like behavior in out-of-distribution reasoning tasks. *arXiv preprint arXiv:2205.05718*, 2022.

Dasgupta, I., Schulz, E., Tenenbaum, J. B., and Gershman, S. J. A theory of learning to infer. *Psychological review*, 127(3):412, 2020.

Dasgupta, I., Lampinen, A. K., Chan, S. C., Creswell, A., Kumaran, D., McClelland, J. L., and Hill, F. Language models show human-like content effects on reasoning. *arXiv preprint arXiv:2207.07051*, 2022.

Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P., and Dolan, R. J. Model-based influences on humans' choices and striatal prediction errors. *Neuron*, 69(6):1204–1215, 2011.

Ershadmanesh, S., Gholamzadeh, A., Desender, K., and Dayan, P. Meta-cognitive efficiency in learned value-based choice. In *2023 Conference on Cognitive Computational Neuroscience*, pp. 29–32, 2023. doi: 10.32470/CCN.2023.1570-0. URL https://hdl.handle.net/21.11116/0000-000D-5BC7-D.

Gershman, S. J. Deconstructing the human algorithms for exploration. *Cognition*, 173:34–42, 2018.

Google. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Hagendorff, T., Fabi, S., and Kosinski, M. Human-like intuitive behavior and reasoning biases emerged in large language models but disappeared in chatgpt. *Nature Computational Science*, 3(10):833–838, 2023.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, 2017.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

LAION. Towards a transparent ai future: The call for less regulatory hurdles on open-source ai in europe. Available at: https://laion.ai/blog/transparent-ai/, 2024. Accessed: January 19, 2024.

Lampinen, A. K., Dasgupta, I., Chan, S. C., Matthewson, K., Tessler, M. H., Creswell, A., McClelland, J. L., Wang, J. X., and Hill, F. Can language models learn from explanations in context? *arXiv preprint arXiv:2204.02329*, 2022.

Lefebvre, G., Lebreton, M., Meyniel, F., Bourgeois-Gironde, S., and Palminteri, S. Behavioural and neural characterization of optimistic reinforcement learning. *Nature Human Behaviour*, 1(4):0067, 2017.

Lejuez, C. W. et al. Evaluation of a behavioral measure of risk taking: the balloon analogue risk task (bart). *Journal of experimental psychology. Applied*, 8(2):75–84, 2002. doi: 10.1037//1076-898x.8.2.75.

Liu, X., Wang, J., Sun, J., Yuan, X., Dong, G., Di, P., Wang, W., and Wang, D. Prompting frameworks for large language models: A survey, 2023.

Massey, C. and Wu, G. Detecting regime shifts: The causes of under-and overreaction. *Management Science*, 51(6): 932–947, 2005.

McCoy, R. T., Yao, S., Friedman, D., Hardy, M., and Griffiths, T. L. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*, 2023.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

Montague, P. R., Dolan, R. J., Friston, K. J., and Dayan, P. Computational psychiatry. *Trends in cognitive sciences*, 16(1):72–80, 2012.

MosaicML. Introducing mpt-30b: Raising the bar for open-source foundation models. Blog post, 2023. URL `www.mosaicml.com/blog/mpt-30b`. Accessed: 2023-06-22.

Nardo, C. The waluigi effect (mega-post). Available at: `https://www.lesswrong.com/posts/D7PumeYTDPfBTp3i7/the-waluigi-effect-mega-post`, 2024. Accessed: January 19, 2024.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Palminteri, S. and Lebreton, M. The computational roots of positivity and confirmation biases in reinforcement learning. *Trends in Cognitive Sciences*, 2022.

Patzelt, E. H., Hartley, C. A., and Gershman, S. J. Computational phenotyping: using models to understand individual differences in personality, development, and mental illness. *Personality Neuroscience*, 1:e18, 2018.

Rescorla, R. A. Classical conditioning ii: current research and theory. pp. 64, 1972.

Ruggeri, K., Panin, A., Vdovic, M., Većkalov, B., Abdul-Salaam, N., Achterberg, J., Akil, C., Amatya, J., Amatya, K., Andersen, T. L., et al. The globalizability of temporal discounting. *Nature Human Behaviour*, 6(10):1386–1397, 2022.

Salewski, L., Alaniz, S., Rio-Torto, I., Schulz, E., and Akata, Z. In-context impersonation reveals large language models' strengths and biases. *arXiv preprint arXiv:2305.14930*, 2023.

Schaeffer, R., Miranda, B., and Koyejo, S. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*, 2023.

Schurr, R., Reznik, D., Hillman, H., Bhui, R., and Gershman, S. J. Dynamic computational phenotyping of human cognition. 2023.

Shekhar, M. and Rahnev, D. Sources of metacognitive inefficiency. *Trends in Cognitive Sciences*, 25(1):12–23, 2021.

Srivastava, A. and authors. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2023.

Tamkin, A., Brundage, M., Clark, J., and Ganguli, D. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*, 2021.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Ullman, T. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv preprint arXiv:2302.08399*, 2023.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models, 2023.

Wilson, R. C., Geana, A., White, J. M., Ludvig, E. A., and Cohen, J. D. Humans use directed and random exploration to solve the explore–exploit dilemma. *Journal of Experimental Psychology: General*, 143(6):2074, 2014.

Yax, N., Anlló, H., and Palminteri, S. Studying and improving reasoning in humans and machines. *arXiv preprint arXiv:2309.12485*, 2023.

Zheng, H. S., Mishra, S., Chen, X., Cheng, H.-T., Chi, E. H., Le, Q. V., and Zhou, D. Take a step back: Evoking reasoning via abstraction in large language models, 2023a.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023b.

# A. List of LLMs used

| Model Name | No. of Parameters | Finetuned LLM | Use of RLHF | Open Source | Size of Dataset | Context Length | Conversational | Code |
|---|---|---|---|---|---|---|---|---|
| GPT-4 | 1760 | No | Yes | No | 1.56 | 8 | No | No |
| text-davinci-003 | 170 | No | Yes | No | 1.37 | 4 | No | No |
| text-davinci-002 | 170 | No | No | No | 1.37 | 4 | No | No |
| Claude-1 | 100 | No | RLAIF | No | 3.7 | 100 | No | No |
| Claude-2 | 200 | No | RLAIF | No | 7.4 | 100 | No | No |
| text-bison@002 | 340 | No | Yes | No | 1.4 | 8 | No | No |
| Falcon-40b | 40 | No | No | Yes | 0.54 | 2 | No | No |
| Falcon-40b-instruct | 40 | Falcon-40b | No | Yes | 0.6 | 2 | No | No |
| MPT-30b | 30 | No | No | Yes | 1.76 | 8 | No | No |
| MPT-30b-instruct | 30 | MPT-30b | No | Yes | 1.8 | 8 | No | No |
| MPT-30b-chat | 30 | MPT-30b | No | Yes | 1.8 | 8 | Yes | No |
| LLaMA-2-70 | 70 | No | No | Yes | 2 | 4 | No | No |
| LLaMA-2-13 | 13 | No | No | Yes | 2 | 4 | No | No |
| LLaMA-2-7 | 7 | No | No | Yes | 2 | 4 | No | No |
| LLaMA-2-70-chat | 70 | Yes | Yes | Yes | 2 | 4 | Yes | No |
| LLaMA-2-13-chat | 13 | Yes | Yes | Yes | 2 | 4 | Yes | No |
| LLaMA-2-7-chat | 7 | Yes | Yes | Yes | 2 | 4 | Yes | No |
| Vicuna-7b-v1.5 | 7 | LLaMA-2-7 | Yes | Yes | 2.37 | 4 | Yes | No |
| Vicuna-13b-v1.5 | 13 | LLaMA-2-13 | Yes | Yes | 2.37 | 4 | Yes | No |
| LLaMA-2-7b-longlora-100k-ft | 7 | LLaMA-2-7 | No | Yes | 2 | 100 | No | No |
| LLaMA-2-7b-longlora-8k-ft | 7 | LLaMA-2-7 | No | Yes | 2 | 8 | No | No |
| LLaMA-2-7b-longlora-16k-ft | 7 | LLaMA-2-7 | No | Yes | 2 | 16 | No | No |
| LLaMA-2-7b-longlora-32k-ft | 7 | LLaMA-2-7 | No | Yes | 2 | 32 | No | No |
| LLaMA-2-7b-longlora-32k | 7 | LLaMA-2-7 | No | Yes | 2 | 32 | No | No |
| LLaMA-2-13b-longlora-32k-ft | 13 | LLaMA-2-13 | No | Yes | 2 | 32 | No | No |
| LLaMA-2-13b-longlora-64k | 13 | LLaMA-2-13 | No | Yes | 2 | 64 | No | No |
| LLaMA-2-13b-longlora-32k | 13 | LLaMA-2-13 | No | Yes | 2 | 32 | No | No |
| LLaMA-2-70b-longlora-32k | 70 | LLaMA-2-70 | No | Yes | 2 | 32 | No | No |
| LLaMA-2-70b-chat-longlora-32k | 70 | LLaMA-2-70-chat | Yes | Yes | 2 | 32 | Yes | No |
| LongAlpaca-7B | 7 | LLaMA-2-7 | No | Yes | 2 | 16 | Yes | No |
| LongAlpaca-13B | 13 | LLaMA-2-13 | No | Yes | 2 | 16 | Yes | No |
| LongAlpaca-70B | 70 | LLaMA-2-70 | No | Yes | 2 | 16 | Yes | No |
| CodeLlama-7B | 7 | LLaMA-2-7 | No | Yes | 2.5 | 16 | No | Yes |
| CodeLlama-13B | 13 | LLaMA-2-13 | No | Yes | 2.5 | 16 | No | Yes |
| CodeLlama-34B | 34 | LLaMA-2-34 | No | Yes | 2.5 | 16 | No | Yes |

This table lists the 35 LLMs used in this paper with different features where:

- **No. of Parameters:** This represents the number of parameters in the model, expressed in billions.
- **Finetuned LLM:** This column indicates whether the model is a fine-tuned version of another model. If it is, the name of the original model from which it was fine-tuned is provided. If it is not a fine-tuned model, 'No' is written. However, if the model serves as the base model for another model listed in this table, 'Yes' is written.
- **Use of RLHF:** This column specifies whether Reinforcement Learning from Human Feedback (RLHF) was used in the training of the model.
- **Open Source:** This indicates whether the model is open source, meaning we have access to the weights of the model.
- **Size of Dataset:** This represents the size of the dataset on which the model was trained, expressed in trillions of tokens.
- **Context Length:** This refers to the length of the context available to the model during its operation.
- **Conversational:** This indicates whether the model was fine-tuned with conversational datasets.
- **Code:** This indicates whether the model was fine-tuned with code datasets.

Please note that the selection of features used for our analyses was made based on the best available knowledge of the authors, as some information about certain models can be challenging to obtain. This limitation could potentially impact the precision of the regression analysis. It underscores the need for greater transparency about LLMs to facilitate more thorough evaluations.

# B. Comprehensive list & explanation of the cognitive experiments

## B.1. Probabilistic reasoning (Dasgupta et al., 2020) - Prior & likelihood weighting

### B.1.1. Summary

This experiment tests how agents update beliefs based on new evidence. Participants are given a wheel of fortune (representing initial prior probabilities) and two urns with different colored ball distributions (representing likelihoods). Upon drawing a ball, participants can revise their belief about the chosen urn, considering both the wheel (prior) and the ball color (evidence). The task allows testing adaptability to different prior/likelihood scenarios by changing the wheel division and ball distributions. Agents have to estimate the probability of the drawn ball's urn. We use this task to estimate an agent's *prior* and *likelihood weightings*. In this task, people showed similar weighting between prior and likelihood, both under one. This underweighting is often referred to as system neglect (Massey & Wu, 2005).

### B.1.2. Methods

We matched the probabilities used in (Dasgupta et al., 2020) to compare to human data. There they had either an informative likelihood case ($P(\text{left urn}|\text{red}) = 0.7, 0.8$ or $0.9$) and an informative prior ($P(\text{left urn}) = 0.5$ or $0.6$) or vice versa. They also trained humans on this experiment, so we only compared it to data from a human's first trial as we are not interested in learning but in how an LLM weighs its prior and likelihoods by default. The default number of simulations here was 100.

### B.1.3. Prompts for LLMs

**Example with informative likelihood**

You are participating in an experiment where you are provided with a wheel of fortune and two urns. The wheel of fortune contains 10 evenly sized sections labeled either F or J, corresponding to the urns F and J. Another person will spin the wheel of fortune, select an urn based on the outcome of the spin, and then randomly pick a ball from the selected urn. Your goal is to give your best estimate of the probability of the urn being F after observing the ball drawn from the urn.

Q: The wheel of fortune contains 6 sections labeled F and 4 sections labeled J. The urn F contains (8, 2) and the urn J contains (2, 8) red/blue balls. A red ball was drawn. What is the probability that it was drawn from Urn F? (Give your probability estimate on the scale from 0 to 1 rounded to two decimal places).

A: I estimate the probability of the red ball to be drawn from the urn F to be 0.

### B.1.4. Metrics

**Performance:** Calculated as the posterior accuracy, therefore 1 minus the Bayes optimal.

**Behaviours 1 & 2: Prior and likelihood weightings** A generalized version of Bayes rule considers prior $\beta_1$ and likelihood $\beta_2$ weightings to account for biases in Bayesian updating:

$$P(A|B) \propto P(B|A)^{\beta_2} \cdot P(A)^{\beta_1}$$

For analytical convenience, this model can be reformulated as linear in log-odds. By fitting this model to the data using least squares linear regression, we can obtain the maximum likelihood estimates of the prior and likelihood weightings:

$$\log\left(\frac{P(\text{Urn F}|\text{Ball})}{1 - P(\text{Urn F}|\text{Ball})}\right) = \beta_0 + \beta_1 \log\left(\frac{P(\text{Urn F})}{1 - P(\text{Urn F})}\right) + \beta_2 \log\left(\frac{P(\text{Ball}|\text{Urn F})}{P(\text{Ball}|\text{Urn J})}\right)$$

- $P(\text{Urn F}|\text{Ball})$ is the subjective probability judgment of the urn being 'F' given the ball's color.

- $P(\text{Urn F})$ and $P(\text{Ball}|\text{Urn F})$ are the prior probability and likelihood, respectively.

- $\beta_1$ and $\beta_2$ are the prior and likelihood weightings, respectively, which are given as exponents in a generalized version of Bayes' rule to capture specific biases. These two coefficients are the two behavioral metrics we report for this experiment.

- $\beta_0$ is the intercept term.

## B.2. Horizon task (Wilson et al., 2014) - Directed & random exploration

### B.2.1. Summary

This task is a two-armed bandit task with stationary reward distributions. Agents first observe four reward values of randomly determined options, followed by making either one or six additional choices. We use this task to measure whether an agent uses uncertainty to guide its exploration behavior (*directed exploration*) and/or whether it injects noise into its policy to explore (*random exploration*). People are known to rely on a combination of both strategies when exploring (Wilson et al., 2014; Brändle et al., 2021).

### B.2.2. Methods

We followed the same methods for prompting LLMs as in (Binz & Schulz, 2023). In the Horizon task, two distinct contexts are presented to participants, each differing in their time horizons. Each game involves 4 forced-choice trials, after which participants are given the opportunity to make a single choice (in the horizon 1 scenario) or six consecutive choices (in the horizon 6 scenario). The 4 forced-choice trials either offer one observation from one option and three from the other (unequal information condition), or two observations from each option (equal information condition).

The design of the horizon 1 and horizon 6 scenarios inherently provides a baseline for pure exploitation. Furthermore, the equal and unequal information conditions are designed to differentiate between directed and random exploration by examining the decision made in the first trial. In the equal information condition, a choice is categorized as random exploration if it aligns with the option with the lower average. Conversely, in the unequal information condition, a choice is classified as directed exploration if it aligns with the option that was observed less frequently during the forced-choice trials.

Our default number of simulations was 100.

### B.2.3. Prompts for LLMs

**Example with horizon 1 scenario**

You are going to a casino that owns two slot machines. You earn money each time you play on one of these machines.

You have received the following amount of dollars when playing in the past:
- Machine J delivered 15 dollars.
- Machine F delivered 37 dollars.
- Machine F delivered 28 dollars.
- Machine J delivered 11 dollars.

Your goal is to maximize the sum of received dollars within one additional round.

Q: Which machine do you choose?

A: Machine

### B.2.4. Metrics

**Performance:** Average delivered dollars.

**Behaviour 1 - Directed Exploration:** This metric is analyzed in the unequal information condition. Here, a regression is performed on the choice variable using three regressors:

- x1 represents the difference in rewards,
- x2 represents the horizon (binary variable), and
- x3 is the interaction term of x1 and x2 (i.e., $x1 \times x2$).

The beta coefficient for x2 (the presence or not of a horizon) is then extracted as the measure of directed exploration.

**Behaviour 2 - Random exploration:** We follow the same procedure as for the directed exploration but in the equal information condition to measure random exploration. However, in this case, the beta coefficient for x3 (the interaction effect between the difference in rewards and the presence of a horizon) from the regression provides the measure of random exploration.

### B.3. Restless bandit task (Ershadmanesh et al., 2023) - Meta-cognition

#### B.3.1. Summary

This is a two-armed bandit task with non-stationary reward distributions. There is always one option with a higher average reward. Every few trials a switch between the reward distributions of the two options occurs. Agents furthermore have to indicate after each choice how confident they are in their decisions. We use this task to measure *meta-cognition*, which indicates whether an agent can assess the quality of its own cognitive abilities. People generally display this ability but its extent is influenced by various internal and external factors (Shekhar & Rahnev, 2021).

#### B.3.2. Methods

In each trial, LLMs are tasked with choosing between one arm which samples a reward from a normal distribution $N(60, 8)$, while the other arm samples a reward from a $N(40, 8)$. LLMs are informed that the slot machine with the higher average reward changes every 18-22 trials.

Additionally, in each trial, LLMs must express their confidence in their choice on a scale from 0 to 1, as opposed to humans who use a Likert scale. The task is composed of 4 blocks, each containing 18-22 trials, resulting in approximately 80 trials in total. This is in contrast to the human task, which consists of 20 blocks for a total of 400 trials. The decision to limit the number of trials was made due to context size restrictions for some LLMs.

Our default number of simulations was 10.

### B.3.3. Prompts for LLMs

**Example for reporting confidence at trial 23**

Q: You are going to a casino that owns two slot machines named machine J and F. You earn dollars $ each time you play on one of these machines with one machine always having a higher average $ reward. Every 18 to 22 trials a switch of block takes place and the other slot machine will now give the higher point reward on average. However, you are not told about the change of block. After each choice, you have to indicate how confident you were about your choice being the best on a scale from 0 to 1. The casino includes 4 blocks of 18 to 22 trials, for a total of 80 trials ’t’. Your goal is to interact with both machines and optimize your $ as much as possible by identifying the best machine at a given point in time which comes in hand with being attentive to a potential change of block. The rewards will range between 20$ and 80$.

You have received the following amount of $ when playing in the past:
t=1: You chose J with a reported confidence of 0.43. It rewarded 54 $.
t=2: You chose J with a reported confidence of 0.53. It rewarded 57 $.
t=3: You chose J with a reported confidence of 0.88. It rewarded 70 $.
...

t=17: You chose F with a reported confidence of 0.99. It rewarded 59 $.
t=18: You chose F with a reported confidence of 0.44. It rewarded 45 $.
t=19: You chose J with a reported confidence of 0.06. It rewarded 61 $.
t=20: You chose J with a reported confidence of 0.51. It rewarded 64 $.
t=21: You chose J with a reported confidence of 0.37. It rewarded 59 $.
t=22: You chose J with a reported confidence of 0.54. It rewarded 42 $.

Q: You are now in trial t=23. Which machine do you choose between machine J and F?(Think carefully remembering that exploration of both machines is required for optimal rewards. Give the answer in the form ’Machine <your choice>’.)

A: Machine F.

Q: How confident are you about your choice being the best on a continuous scale running from 0 representing ” ’this was a guess’ to 1 representing ’very certain’? (Think carefully and give your answer to two decimal places)

A: On a scale from 0 to 1, I am confident at 0.

### B.3.4. Metrics

**Performance:** Accuracy of choosing the best arm at a given trial.

**Behaviour - Meta-cognition:** We report the metacognitive sensitivity of a model by reporting the adjusted QSR (Carpenter et al., 2019) defined as

$$QSR = 1 - (\text{accuracy} - \text{scaled confidence})^2$$

which is a standard metric for metacognitive sensitivity. The scaled confidence is computed as

$$\text{scaled confidence} = \frac{\text{confidence} - \text{lowest reported confidence}}{\text{highest reported confidence} - \text{lowest reported confidence}}$$

## B.4. Experiment 2: Instrumental learning(Lefebvre et al., 2017) - Optimism bias & learning rate

### B.4.1. Summary

**Instrumental learning** (Lefebvre et al., 2017): LLMs encounter four two-armed bandit problems in an interleaved order. Each bandit problem is identified by a unique symbol pair. We use this task to investigate how an agent learns. First, we report the *learning rate* of the agent which is common practice in two-armed bandits. Furthermore, we use it to reveal whether an agent learns more from positive than from negative prediction errors, i.e., whether it has an *optimism bias*. People commonly display asymmetric tendencies when updating their beliefs by showing higher learning rates after encountering positive prediction errors compared to negative ones (Palminteri & Lebreton, 2022).

### B.4.2. Methods

As in (Lefebvre et al., 2017), the task is 4 two-armed bandits of 96 trials (24 per slot machine). Here we randomly sample (without replacement) two letters for each to avoid biases towards a given letter. We used a cover story that involved a gambler visiting different casinos to generate our prompts. This choice has been inspired by similar tasks for human experiments (Gershman, 2018) and LLMs (Binz & Schulz, 2023; Coda-Forno et al., 2024). Our default number of simulations per LLM is 10.

Casinos have the same reward probabilities as in the paper's first experiment: All arms have probabilities P=0.75 or 0.25 of winning 1 dollar and a reciprocal probability (1 – P) of getting nothing. In two casinos, the reward probability was the same for both arms ('symmetric' conditions), and in two other conditions, the reward probability was different across symbols ('asymmetric' conditions).

### B.4.3. Prompts for LLMs

**Example for 5th trial**

You are going to visit four different casinos (named 1, 2, 3, and 4) 24 times each. Each casino owns two slot machines which all return either 1 or 0 dollars stochastically with different reward probabilities. Your goal is to maximize the sum of received dollars within 96 visits.

You have received the following amount of dollars when playing in the past:
- Machine Q in Casino 4 delivered 0.0 dollars.
- Machine B in Casino 1 delivered 1.0 dollars.
- Machine B in Casino 1 delivered 0.0 dollars.
- Machine R in Casino 3 delivered 0.0 dollars.

Q: You are now in visit 5 playing in Casino 4. Which machine do you choose between Machine Q and Machine D? (Give the answer in the form ”Machine <your choice>”).

A: Machine

### B.4.4. Metrics

**Performance**: The performance is the average amount of money retrieved by the LLM.

**Behaviour 1 - Learning rate**: We fit a Rescorla-Wagner model (Rescorla, 1972) which is standard to retrieve learning rates in two-armed bandits. This model operates under the assumption that decisions are made according to a Softmax function, which takes into account the predicted values of both arms. Each predicted value is updated using $\Delta V = \alpha \times$ prediction error where $\Delta V$ represents the change in value, and $\alpha$ denotes the learning rate. We report the learning rate which minimizes the negative log-likelihood.

**Behaviour 2 - Optimism bias**: As in (Lefebvre et al., 2017), we retrieve the optimism bias by assuming that there were two different learning rates, one for positive ($\alpha^+$) and one for negative ($\alpha^-$) prediction errors, sometimes called the RW $\pm$ model. The two learning rates were fit in the same way as for the standard Rescorla-Wagner model and the Optimism bias is computed as $\alpha^+ - \alpha^-$. This measure provides a quantitative representation of an individual's tendency to learn more from positive outcomes than from negative ones.

## B.5. Two step task (Daw et al., 2011) - Model-basedness

### B.5.1. Summary

This is a decision-making task in which agents have to accumulate as many treasures as possible. Taking an action from a starting state transfers the agent to one out of two second-stage states. In each of these second-stage states, the agent has the choice between two options that probabilistically lead to treasures. Finally, the agent is transferred back to the initial state and the process repeats for a predefined number of rounds. The task experimentally disentangles model-based from model-free reinforcement learning. We therefore use it to measure an agent's *model-basedness*. Previous studies using this task have shown that people rely on a combination of model-free and model-based reinforcement learning (Daw et al., 2011).

### B.5.2. Methods

We followed the same methods for LLMs as in (Binz & Schulz, 2023) with a 20-day horizon. Our default number of simulations was 100.

The transition probabilities from the first stage to the chosen second stage are fixed at 70%. The two-step task gauges model-based decision-making by observing how past outcomes influence current choices. If a participant's decisions reflect the previous trial's second-stage state and reward, it suggests model-based decision-making, as they're using a cognitive model of the task. However, if decisions are solely based on the previous trial's first-stage choice and reward, it indicates model-free decision-making.

### B.5.3. Prompts for LLMs

**Example on 5th day after choosing planet Y for the first-step of the task.**

You will travel to foreign planets in search of treasures. When you visit a planet, you can choose an alien to trade with. The chance of getting treasures from these aliens changes over time. Your goal is to maximize the number of received treasures.

Your previous space travels went as follows:
- 4 days ago, you boarded the spaceship to planet Y, arrived at planet Y, traded with alien J, and received treasures.
- 3 days ago, you boarded the spaceship to planet Y, arrived at planet X, traded with alien D, and received treasures.
- 2 days ago, you boarded the spaceship to planet Y, arrived at planet Y, traded with alien J, and received junk.
- 1 day ago, you boarded the spaceship to planet Y, arrived at planet X, traded with alien D, and received treasures.

Q: Do you want to take the spaceship to planet X or planet Y?

A: Planet Y.
You arrive at planet Y.

Q: Do you want to trade with alien J or K?

A: Alien

### B.5.4. Metrics

**Performance:** Average number of received treasures. It is worth noting that the design of this experiment was done in a way that being model-free or model-based retrieves the same amount of rewards in average.

**Behaviour - Model-basedness:** To retrieve the model-basedness of an agent, we compute a regression using three regressors:

- x1 representing rewards,
- x2 representing common transitions (binary variable) and

- x3 is the interaction term of x1 and x2 (i.e., $x1 \times x2$).

The regression is performed with the 'stay probabilities' as the dependent variable, and x1, x2, and x3 as the independent variables. The 'stay probabilities' represent the likelihood of a participant repeating the same first-stage choice on the next trial. We then retrieve the beta parameter for the interaction effect.

In essence, the interaction effect captures how the influence of rewards on stay probabilities changes depending on whether the previous trial involved a common or rare transition. A significant beta parameter for x3 would suggest that the effect of rewards on stay probabilities is not the same for common and rare transitions, indicating the presence of model-based decision-making.

## B.6. Temporal discounting (Ruggeri et al., 2022)

### B.6.1. Summary

Agents have to make a series of choices between two options. Each option is characterized by a monetary outcome and an associated delay until the outcome is received. We use this task to assess *temporal discounting*, indicating whether an agent prefers smaller but immediate gains over larger delayed ones. People generally show a preference for immediate gains, although the precise functional form of their discounting is still a matter of debate (Cavagnaro et al., 2016).

### B.6.2. Methods

This task tests discounting patterns from three baseline scenarios to determine preference for immediate or delayed choices for gains (at two magnitudes) and losses (one). Second, they analyzed the prevalence of all choice anomalies using 4 additional items. Participants responded to 10 to 13 questions, depending on their responses to the initial three sets. Each baseline consisted of five sub-questions. Individuals saw at most three sub-questions depending on the order of their choices. It is worth noting that since this task is the only one which is not procedurally generated, there is only one simulation needed per LLM.

### B.6.3. Prompts for LLMs

**Examples for first baseline**

Q: What do you prefer between the following two options:
- Option 1: Receive 500 dollars now.
- Option 2: Receive 550 dollars in 12 months.
A: I prefer option 2.

Q: What do you prefer between the following two options:
- Option 1: Receive 500 dollars now.
- Option 2: Receive 600 dollars in 12 months.

A: I prefer option

### Examples for 2nd baseline (different magnitude)

Q: What do you prefer between the following two options:
- Option 1: Receive 5000 dollars now.
- Option 2: Receive 5500 dollars in 12 months.

A: I prefer option 1.

Q: What do you prefer between the following two options:
- Option 1: Receive 5000 dollars now.
- Option 2: Receive 5100 dollars in 12 months.

A: I prefer option 1.

“Q: What do you prefer between the following two options:
- Option 1: Receive 5000 dollars now.
- Option 2: Receive 5050 dollars in 12 months.

A: I prefer option

### Examples for 3rd baseline (loss as opposed to gain)

Q: What do you prefer between the following two options:
- Option 1: Pay 500 dollars now.
- Option 2: Pay 550 dollars in 12 months.

A: I prefer option 1

Q: What do you prefer between the following two options:
- Option 1: Pay 500 dollars now.
- Option 2: Pay 510 dollars in 12 months.

A: I prefer option 1

Q: What do you prefer between the following two options:
- Option 1: Pay 500 dollars now.
- Option 2: Pay 505 dollars in 12 months.

A: I prefer option

### Example for testing present bias

Q: What do you prefer between the following two options:
- Option 1: Receive 500 dollars in 12 months.
- Option 2: Receive 600 dollars in 24 months.

A: I prefer option

**Example for testing subbaddictivity**

Q: What do you prefer between the following two options:
- Option 1: Receive 500 dollars now.
- Option 2: Receive 700 dollars in 24 months.

A: I prefer option

**Example for testing delay-speedup asymmetry**

Q: What do you prefer between the following two options:
- Option 1: Receive 500 dollars now.
- Option 2: Wait 12 months for the 500 dollars but with an additional 99 dollars.

A: I prefer option

**Example for testing delay-length asymmetry**

Q: What do you prefer between the following two options:
- Option 1: Wait 12 months to receive 600 dollars now.
- Option 2: Pay 100 dollars and receive the 600 dollars gain now.

A: I prefer option

#### B.6.4. Metrics

**Behaviour:** We only report one temporal discounting metric as behavioral. We use the same scoring system as in (Ruggeri et al., 2022) where the score is between 0 and 19.

The constructed score corresponded to the ranking of the sub-question where the individual switched from choosing the delayed option towards the sooner option. An individual who always chose the latter option was assigned a score of 0. An individual who always chose the sooner option was assigned a score of 5 (for baselines) and 1 for each of the 4 biases. Adding this score across questions results in a total score of 0 for individuals always choosing the delayed amount and 19 (5+5+5+4) for individuals always choosing the sooner amount.

### B.7. Balloon Analogue Risk Task (BART) (Lejuez et al., 2002) - Risk

#### B.7.1. Summary

Agents have to inflate an imaginary balloon to obtain rewards. They may choose to stop inflating and cashing out all rewards accumulated so far. There is a chance that the balloon pops at any point in time and all rewards will be lost. We use this task to assess *risk-taking* behavior. Human risk-taking in this task is "significantly correlated with scores on self-report measures of risk-related constructs and with the self-reported occurrence of real-world risk behaviors" (Lejuez et al., 2002).

#### B.7.2. Methods

We substitute the colors of the balloons with the letters A, B, and C. We reduce the number of balloons from 15 to 10 per category to avoid context-size limits from some LLMs. Our default number of simulations per LLM is 10.

For the three balloons we sample from the same three distributions as in the paper where probabilities increase when inflating the balloon but start at $1/32, 1/8$ and $1/128$ per different balloon category. We randomly shuffle the assignment to A, B, and C to avoid biases of LLMs towards certain letters.

### B.7.3. Prompts for LLMs

**Example for 5th balloon**

In this game, you will encounter 3 different balloons labeled A, B, and C. There will be a total of 10 balloons for each type of balloon. Your goal is to accumulate as many points as possible without popping the balloon. You will be presented with a balloon and given the option to inflate it or not. Each inflation increases the balloon's size and potential points but also carries a risk of the balloon popping. Your task is to decide whether to inflate the balloon or not knowing that a successful inflation adds 1 point from that balloon. Once you decide to stop inflating the balloon, you can no longer earn points from that balloon. If the balloon pops before you stop inflating, you will lose all the points accumulated in that balloon. Your final score will be determined by the total number of points earned across all 30 balloons. Your goal is to maximize your final score.

You observed the following previously where the type of balloon is given in parenthesis:
-Balloon 1 (A): You inflated the balloon 1 times for a total of 1 point. It did not explode.
-Balloon 2 (C): You inflated the balloon 4 times for a total of 4 points. It did not explode.
-Balloon 3 (A): You inflated the balloon 7 times for a total of 0 points. It did explode.
-Balloon 4 (C): You inflated the balloon 5 times for a total of 5 points. It did not explode.
-Balloon 5 (A): You inflated the balloon 9 times for a total of 0 points. It did explode.

Q: You are currently with Balloon 5 which is a balloon of type A. What do you do? (Option 1 for 'skip' or 0 for 'inflate')

A: Option

### B.7.4. Metrics

**Performance:** The performance is the average points across all simulations.

**Behaviour: Risk** In the paper they report the adjusted risk which is defined as the average number of pumps excluding balloons that exploded. However, this does not take into account edge behaviours which always inflate which is the case for some LLMs and therefore we decided to report the risk as the average number of inflation attempts.

## C. Full benchmark results for rest of LLMs

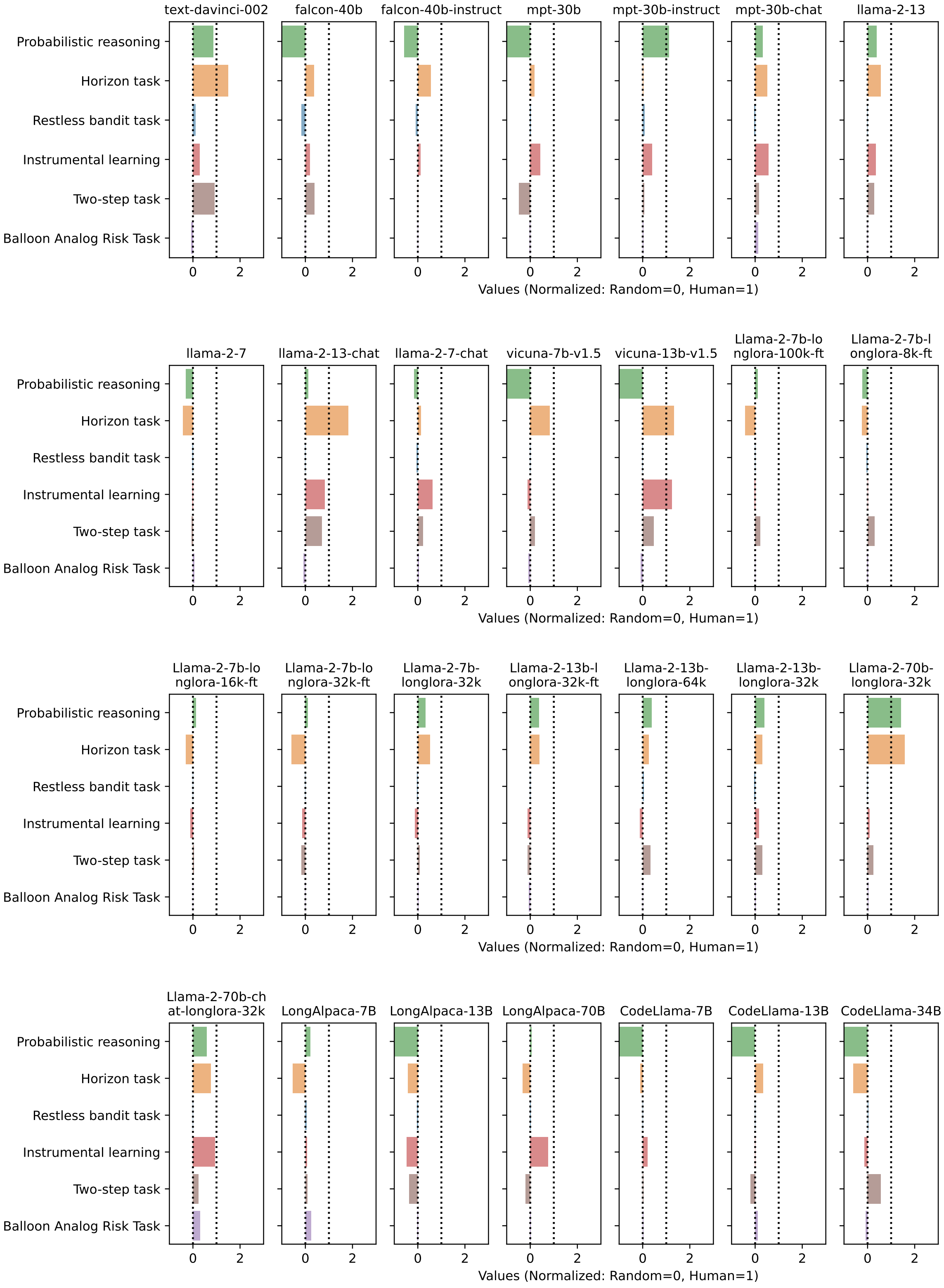


*Figure 6.* Performance metrics

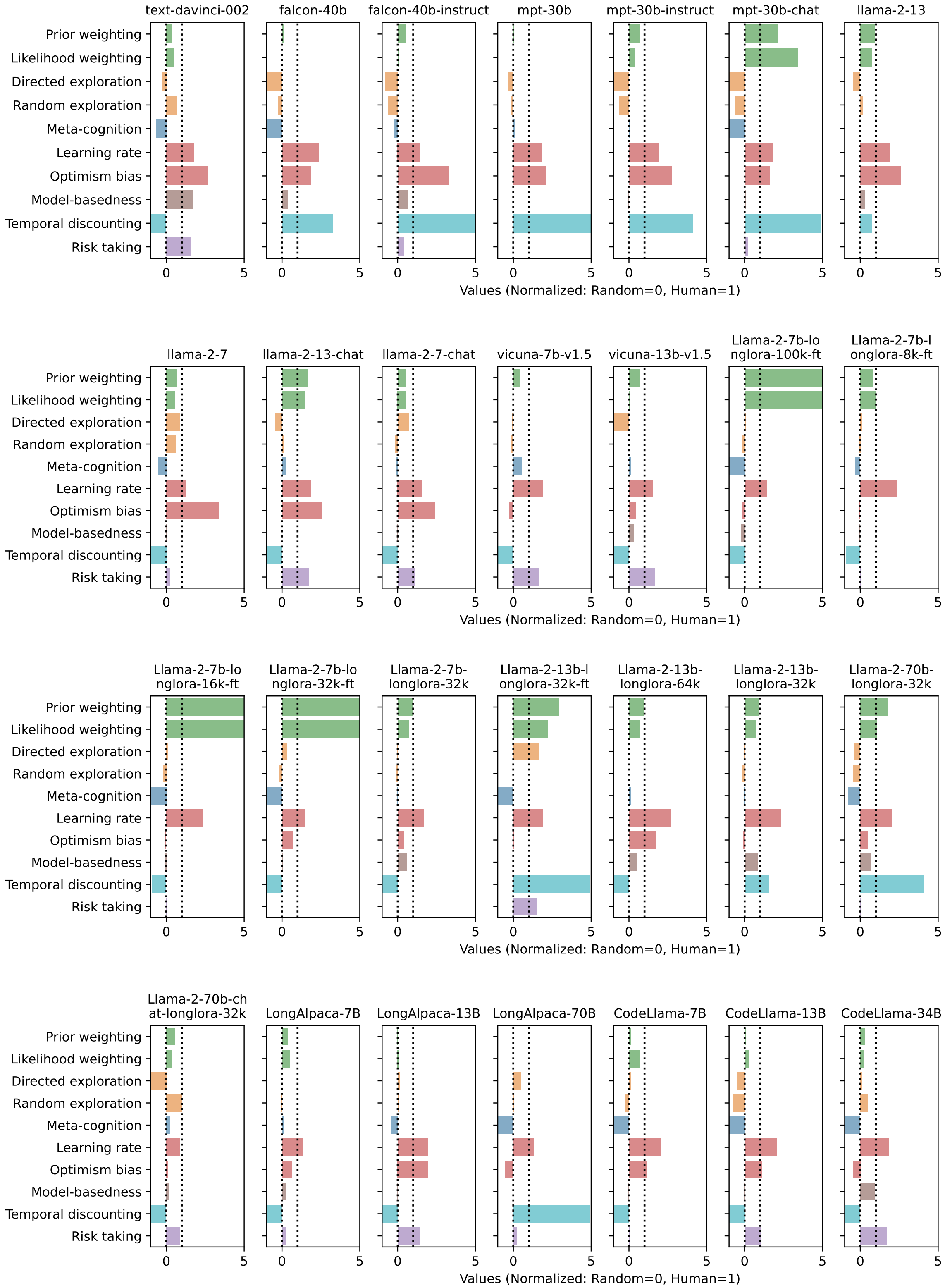


*Figure 7.* Behavioral metrics

## D. Prompt Engineering techniques

In both the CoT and SB experiments, we appended specific prompts at the end (details provided below) where the function ‘self.format_answer’ was different for each experiment. We imposed a limit of 300 tokens for an LLM. This approach, however, presented some challenges when compared with the standard benchmark analysis, which is designed to output a maximum of one token to ensure a context that enforces a one-token answer.

When we permit an LLM to modify the context with a flexible number of tokens, despite our attempts to enforce a maximum word limit, some LLMs do not consistently adhere to this constraint. This flexibility introduces complexity into the process of automating these engineering techniques across different experiments for various types of LLMs.

Additionally, some LLMs begin to exhibit chaotic behavior, and once this occurs, it becomes difficult to revert to a controlled state. This phenomenon, known as the ‘Waluigi effect’ (Nardo, 2024), underscores the challenges of managing the balance between flexibility and control in the design and operation of LLMs.

**Example for take-a-step-back**

First, take-a-step-back and think in the following two steps to answer this:
Step 1) Abstract the key concepts and principles relevant to this question in a maximum of 60 words.”
Step 2) Use the abstractions to reason through the question in a maximum of 60 words.

Finally, give your final answer in the format ’Final answer: {self.format_answer}<your choice>’. It is very important that you always answer in the right format even if you have no idea or you believe there is not enough information.

A: Step 1)

**Example for chain-of-thought**

First break down the problem into smaller steps and reason through each step logically in a maximum of 100 words before giving your final answer in the format ’Final answer: {self.format_answer}<your choice>’. It is very important that you always answer in the right format even if you have no idea or you believe there is not enough information.

A: Let’s think step by step: