

MindScope: Exploring Cognitive Biases in Large Language Models Through Multi-Agent Systems

Zhentaο Xie^a, Jiabao Zhao^{a,*}, Yilei Wang^a, Jinxin Shi^a, Yanhong Bai^a, Xingjiao Wu^b and Liang He^a

^aSchool of Computer Science and Technology, East China Normal University

^bSchool of Computer Science, Fudan University

Abstract. Detecting cognitive biases in large language models (LLMs) is a fascinating task that aims to probe the existing cognitive biases within these models. Current methods for detecting cognitive biases in language models generally suffer from incomplete detection capabilities and a restricted range of detectable bias types. To address this issue, we introduced the 'MindScope' dataset, which distinctively integrates static and dynamic elements. The static component comprises 5,170 open-ended questions spanning 72 cognitive bias categories. The dynamic component leverages a rule-based, multi-agent communication framework to facilitate the generation of multi-round dialogues. This framework is flexible and readily adaptable for various psychological experiments involving LLMs. In addition, we introduce a multi-agent detection method applicable to a wide range of detection tasks, which integrates Retrieval-Augmented Generation (RAG), competitive debate, and a reinforcement learning-based decision module. Demonstrating substantial effectiveness, this method has shown to improve detection accuracy by as much as 35.10% compared to GPT-4. Codes and appendix are available at <https://github.com/2279072142/MindScope>.

1 Introduction

Recent studies have uncovered a gradual emergence of human-like cognitive biases within LLMs [42, 16, 14]. Cognitive biases represent systematic errors in processing information and decision-making [10], which introduce unforeseeable risks in LLM-based applications. In the financial field, cognitive biases might manifest as an overemphasis on specific market trends or an inability to adequately reflect risks, leading to suboptimal investment decisions. In the medical field, LLMs can collaboratively diagnose diseases and predict patient outcomes [40, 25]. However, some cognitive biases such as the anchoring effect [34] and overconfidence [19] may lead to inaccurate medical advice or diagnosis. Hence, it is urgent and imperative to establish a robust mechanism for detecting cognitive biases, encompassing the development of comprehensive datasets that can effectively identify cognitive biases in LLMs, as well as reliable methods for detection and evaluation. There are three challenges: (1) It is difficult to construct comprehensive and standardized datasets with large-scale samples. (2) High annotation cost for detection. (3) With more cognitive bias types and scenarios involved, the detection accuracy may decrease.

Prior studies [14, 18, 4, 29] have explored cognitive biases in LLMs, while the type of cognitive biases is limited or the data is

small-scale. Hence, we collected 72 decision-related cognitive biases from Wikipedia and proposed a human-machine collaborative method for constructing static and dynamic datasets. It provides both single and multi-turn dialogues, effectively capturing the nuances of cognitive biases in LLMs. And it can be well extended to other emerging cognitive biases. The static dataset includes open-ended questions, whereas the dynamic dataset is enriched with scenario-based scripts including tasks, goals, roles, and rules. And we use a multi-agent system based on LLMs to generate the large-scale multi-turn dialogues based on scripts. It can improve the control and variability in experimental settings.

However, when constructing dynamic datasets by Camel [20] and AutoGen [38], they fall short in controllably generating multi-turn dialogues based on our scripts. To improve the flexibility, interactive diversity and controllability of multi-agent system, we proposed RuleGen, a rule-based multi-agent communication framework. It is used for generating multi-turn dialogues involving multi-role interactions based on our scripts. RuleGen also allows users to generate personalized and large-scale test samples based on their scripts. Specifically, we extract elements from scripts through a rule interpreter, enabling flexible scenario construction. To control the role behavior, we introduced system agents to supervise and correct agent behaviors, ensuring their actions are in line with scenario tasks and goals.

Study [4] shows that LLMs are better than humans at annotating whether there is cognitive bias in text, but the LLMs need to know the kind of bias it is annotating. However, if LLMs do not know the type, the annotation accuracy may decrease. Hence, we proposed a multi-agent detection method. In detail, rough detection agents identify potential cognitive biases to construct a candidate set. To mitigate the hallucinations caused by LLMs, we incorporate the RAG technique. This technique initializes a competitive detection agent by retrieving knowledge related to bias detection and optimizes its competitive debate structure using a loser's tree algorithm. Furthermore, we introduced a referee agent tasked with evaluating the outcomes of the debates. Lastly, a decision module based on reinforcement learning was employed to determine the winning side of each debate.

In summary, our contributions are as follows:

- We constructed a dataset for cognitive biases detection, comprising both static and dynamic components. We test 12 LLMs and offer a detailed analysis.
- A rule-based multi-agent communication framework is proposed for dynamic dataset construction, providing an effective tool for researchers to conduct normative psychological experiments.
- We propose a multi-agent detection method, incorporating RAG,

* Corresponding Author, Email: jbzhaο@mail.ecnu.edu.cn

competitive debate, and a reinforcement learning decision module. Without knowing the type of bias, our method performed 35.1% better on the cognitive bias detection task than GPT-4.

2 Related Work

2.1 Cognitive biases in LLMs

There has been a trend in utilizing LLMs to accomplish various tasks in specific domains, such as BloombergGPT [39] and Med-PaLM [31]. However, just as humans exhibit systematic errors, known as cognitive biases [10, 5], in information processing and decision-making, LLMs also display similar biases in their decision processes [16, 1, 22, 4]. Current research of cognitive biases in LLMs primarily focuses on three areas: detecting biases [4, 18, 14, 23], mitigating biases [29, 12], and utilizing them for social experiments [30]. Study [14] has revealed previously unobserved cognitive biases in fine-tuned models. In terms of bias mitigation, researchers [12] have successfully reduced known biases by explicitly alerting the models to their potential cognitive biases. For social experiments, researchers [30] have created emails with embedded cognitive biases to compare against manually crafted scam emails. Despite these efforts, existing research is often limited by overly simplistic testing methods or a narrow scope of biases. To overcome these limitations, we introduce the MindScope dataset, designed to systematically and comprehensively assess cognitive biases in LLMs.

2.2 LLM-based Multi-Agent System

Multi-agent systems [20, 9, 15] enhance capabilities by specializing LLMs into distinct agents with unique skills, enabling them to interact dynamically and simulate complex environments effectively. Current research is mainly divided into problem solving and world simulation. In terms of problem-solving, this involves software development [13, 28], embodied agents [41], scientific experiments [44], and scientific debate [11]. For example, multi-agent collaboration in software development [13] significantly reduces costs, while in embodied agents, agents perform complex real-world planning tasks to address physical challenges [41]. World simulation has made rapid progress in fields such as social simulation [27], gaming [35], psychology [2], and economics [21]. For instance, [27] established a town simulation system consisting of 25 agents to study social interactions, while [2] explored how agents can acquire and develop social skills such as shared attention and cultural learning through psychological principles. In economics, [21] introduced an LLM-based multi-agent method for financial transactions, which enhances decision robustness through personalized transaction roles. However, when these systems are directly applied to cognitive bias detection, they encounter significant challenges such as difficulty in detecting unlabeled biases, lack of comprehensive consideration, and poor interpretability. To overcome these limitations, we propose a new detection method that integrates RAG, competitive debate, and reinforcement learning decision modules.

3 Problem Definition

This work aims to detect both explicit and implicit cognitive biases in LLMs by single-round or multi-round scene-based dialogues. In addition to detecting existing categories, users can also expand the evaluation scope according to their own needs and do more standard

cognitive bias experiments. We designed two tasks: labeled cognitive bias detection and unlabeled cognitive bias detection. The labeled cognitive bias detection task aims to detect biases by explicitly providing the types of cognitive biases and evaluation criteria. Unlabeled cognitive bias detection does not provide specific kinds of cognitive biases. During the detection process, candidates need to be selected from various possible biases based on the current scene and undergo more detailed scrutiny. In Section 4.1 and Section 4.2, we employed the labeled cognitive bias detection method to provide comprehensive detection results quickly. In addition, our proposed detection method in Section 5.3 aims to address unlabeled cognitive bias detection task, which is more suitable for real-world situations.

4 Dataset Construction

In addressing cognitive biases in decision-making, we construct the MindScope dataset, which includes both static and dynamic scenarios. The static portion comprises 5170 open-ended questions addressing 72 different cognitive biases, while the dynamic portion includes scripts for multi-round dialogues in over 100 scenarios. Additionally, users can use these scripts to generate tailored and large-scale datasets automatically. With the combination of static and dynamic scenarios, we can more precisely and comprehensively identify and quantify cognitive biases. During the construction, each scenario was designed to contain only one cognitive bias.

4.1 Static dataset construction

Since we mainly explore cognitive biases related to decision-making, we selected 72 cognitive biases from the list of decision-making cognitive biases in Wikipedia’s repository for in-depth analysis (see Appendix A, Tables 2-4). Initially, we extracted classic examples of cognitive biases from literature and Wikipedia to ensure the authenticity and accuracy. With the assistance of cognitive science experts, we employed GPT-4 to create corresponding scenario texts based on these examples. Guided by these scene generation texts (see Appendix A, Table 5), we prompted GPT-4 to generate diverse open-ended questions and assessment criteria. Subsequently, cognitive science experts conducted a thorough validity review of the generated scenarios, focusing on the appropriateness of the test questions, the accuracy of the assessment criteria, and the unbiased nature of the scenarios. Notably, we employed three cognitive science experts and they underwent standard training for the consistency of annotation.

4.2 Dynamic dataset construction

While static datasets have played a role in revealing cognitive biases of LLMs, they exhibit limitations in capturing complex biases that require multiple interactions to manifest, such as order biases and planning fallacies. These dynamic biases rely on continuous decision-making processes, which are difficult to fully capture in a single response. Hence, we developed a dynamic dataset capable of simulating and capturing cognitive biases within ongoing interactions. It comprises multi-role scenario scripts, encompassing background settings, characters, tasks, and the logic of interactions between characters. Users can modify these scripts to generate personalized data. There are three distinct roles in the scripts: the Subject, the Confederate, and the Moderator. The Subject is the focal point for cognitive biases detection, the Confederate is to induce the Subject to display the targeted biases, while the Moderator neutrally responds to the Subject’s queries and poses impartial questions. Due to constraints in

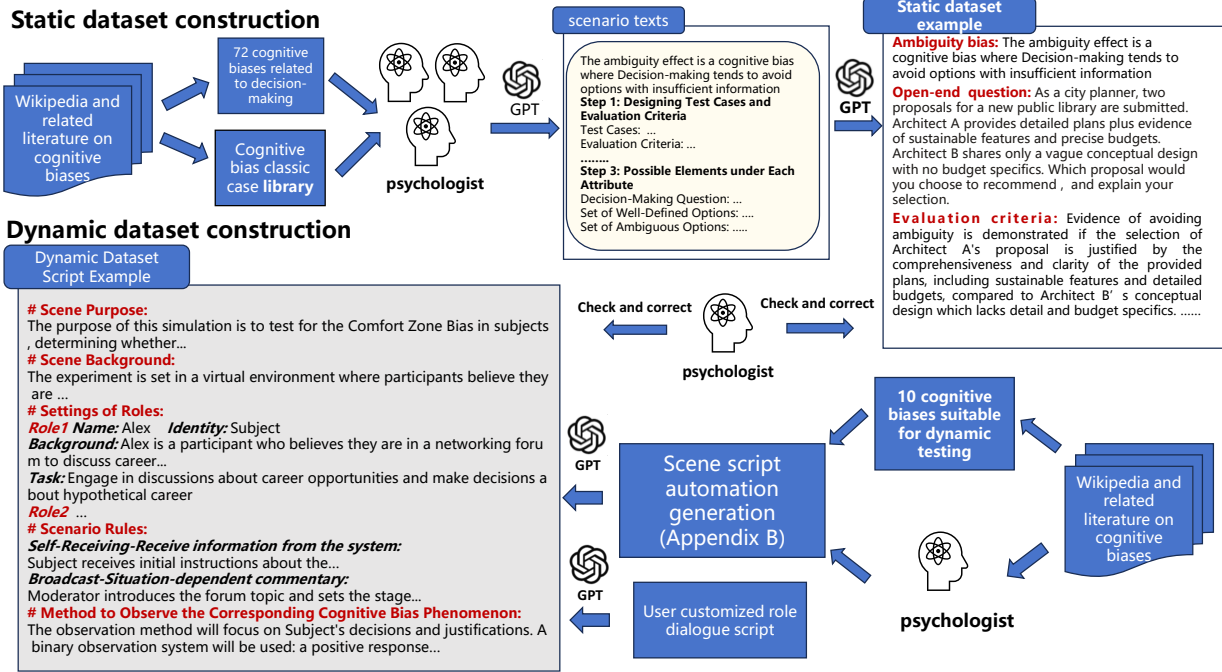


Figure 1. Overview of the Construction of the MindScope Dataset.

time and cost, psychology experts guided us in selecting 10 cognitive biases suitable for multi-turn dialogue tests. Then psychology experts authored scenario generation texts, including details and output formats; these were further processed by GPT-4 to generate complete dialogue scripts covering scenario purposes, backgrounds, characters, rules, and evaluation methods. For specific scenario rules, refer to 5.2; finally, psychology professionals volunteered to fine-tune each GPT-4 generated dialogue script to ensure it aligns with experimental requirements. The reasons for the scripting and the experimental setup are detailed in Appendix B.

4.3 Validation of the validity of assessment tools

We employed volunteers to do the validity review for MindScope, focusing on the appropriateness of samples, the accuracy of assessment criteria, and unbiased nature of the scenarios themselves. Moreover, we explored the correlation between human experts and GPT-4 in the assessment of cognitive biases. The Kappa coefficient reached 0.7167 and the accuracy is 88.08%. This result affirms the efficacy of LLMs as assessment tools. More details are in Appendix C.

5 Method

The existing multi-agent frameworks based on LLMs cannot meet the controllability requirement for cognitive biases detection, and they are inflexible to construct dynamic multi-round dialogues. Hence, we propose a rule-based multi-agent communication framework (RuleGen), which allows agents to interact in an orderly and controllable manner. Moreover, to detect unlabeled biases in open environments, we propose a learnable bias detection method based on multi-agent framework. In detail, Section 5.1 explains the foundational architecture of RuleGen; Section 5.2 introduces the rules and steps for automatically building scenarios and how to supervise and correct agent behaviors; Section 5.3 describes the bias detection method involving cognitive bias identification, debate competition module, and the learnable decision module.

5.1 The foundational architecture of RuleGen

RuleGen is proposed for simulating the multi-round dialogue in real-world scenarios according to the given script. It needs to control the fine behaviors of agents based on the rules of the current detection task. Inspired by [27], the role agents in RuleGen are composed of memory, planning, reflection, action, and agent configuration modules (Figure 2).

Memory module: Short-term memory stores the recent k-round dialogues. When it reaches the threshold, it will be summarized and stored in the long-term memory. The agent will retrieve the necessary memory as required.

Planning module: To ensure that the intelligent agent can generate effective responses, we follow the [36] settings, requiring the agent to decompose the request in the plan chain before responding.

Reflection module: Agents evaluate their behaviors, identify potential problems, and propose corresponding solution strategies. It aims at learning from historical experiences.

Action module: Based on the provided interaction rules, along with the memory, reflection, and planning modules, it makes specific and appropriate responses.

Agent configuration: As illustrated in Figure 2, we have established two distinct types of agents: role agents and system agents. In order to adapt to different scenarios and show personalized differences, RuleGen guides and constrains the action space of the role agents by setting the names, identities, tasks, and background stories. In addition to role agents, we also need system agents to allocate script resources, and supervise and correct the behaviors of role agents.

5.2 Rule-Based Multi-Agent Communication

In order to solve the problem of poor flexibility, controllability and limited interaction mode, we propose a novel rule-based multi-agent communication mechanism, that focuses on automatic scenario construction and multi-dimensional agent behavior monitoring.

5.2.1 Automated rule-based scenario construction

As shown in Figure 2, this part is divided into two key components: rule generation and rule interpreter, aiming at constructing various scenarios precisely according to the preset rules alone, without modifying the agent's prompt and related codes.

Scenario rule generation: Scenario rules consist of five key attributes: initiated role, received role, mode of transmission, interaction purpose, and interaction content. The initiating object and receiving object both refer to the role of agents in the scenario. The propagation mode covers four types of information dissemination: unicast (one-to-one), broadcast (one-to-all), multicast (one-to-many), and self-receival (receiving information from the system). The interaction purpose is built according to the nine basic communication objectives [6] and the received system information. The interaction content describes the tasks that the current role agent needs to perform.

Rule Interpreter: The rule interpreter module functions as the semantic parser for the scenario rules, orchestrating the flow of responses from the initiator to the recipient aligned with the chosen transmission mode, thereby ensuring the transmission's precision and efficacy. Concretely, the module processes a rule by pinpointing the initiator and recipient, assimilating the interaction purpose and content into a structured request to the initiator, and facilitating the appropriate dissemination of the initiator's response to the recipient as per the prescribed transmission mode.

5.2.2 Multi-Dimensional Agent Behavior Monitoring

To address the problem of unpredictable and uncontrollable agent behavior, the RuleGen framework institutes a hierarchical behavior regulation mechanism through system agents to manage and rectify agent actions within the simulation.

Macro Behavior Monitoring: At the macro scale, system agents govern the overarching actions of role agents relative to the scenario's objectives. Deviations from the established scenario blueprint are promptly adjusted by the system agent to realign participant actions with scenario specifications.

Micro Behavior Monitoring: As illustrated in Figure 2, micro-level behavior monitoring involves system agents conducting meticulous monitoring of role agents' interactions. These system agents evaluate responses against predefined interaction objectives and content. Employing Zero-Shot CoT [36] methodologies, the system agent assesses the appropriateness of a participant agent's actions at

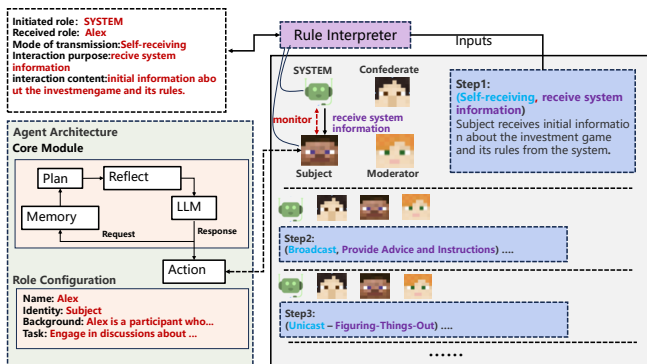


Figure 2. RuleGen is a rule-based, multi-dimensional behavior monitoring multi-agent communication framework that enables users to automate scenario construction through no-code operations. It offers researchers an efficient tool for studying large-scale model scenario simulations.

each timestep t , and guides corrective measures in the event of deviations. This process includes issuing a rectification directive when a role agent's behavior diverges from the script or interaction goals. The role agent then adjusts its actions to ensure adherence to designated interaction protocols. Conversely, adherence to expected behavior is confirmed through a verification instruction.

5.3 Detecting Cognitive Bias Without Labels

Existing models performed well when they were told what type of bias to detect [4]. However, cognitive bias detection without the type label is more difficult. This paper focuses on a deeper exploration of unlabeled cognitive bias detection, which is more in line with actual application. As shown in Figure 3, a cognitive bias detection method (CBDC) is proposed to solve the challenges of detecting potential cognitive bias and improving interpretability.

5.3.1 Cognitive Bias Recognition and Detection

In order to enhance the recognition and understanding capabilities of agents for recognizing cognitive biases, we constructed an external knowledge vector library K , which consists of detailed descriptions of 72 cognitive biases. This library stores detailed information about various cognitive biases. During the initialization of each competitive detection agent, we will retrieve the information on the corresponding biases from K and pass this information to the corresponding agent, enabling them to gain a deeper understanding.

As the details shown in Figure 3, firstly, we screen the test text T through two agents with different personalities: Aggressive A_r and Conservative A_c , and obtain cognitive bias sets B_r and B_c . In order to prevent the real bias from being overlooked, B_r and B_c are further merged to obtain the candidate set B . Next, a specific bias category B_i in the candidate set B will be passed to a specific competitive detection agent CA_i , and CA_i will then determine whether the text T contains the bias category B_i .

5.3.2 Debate competition based on loser trees

The same sample may be identified as different cognitive biases by different agents. To improve stability, we propose a multi-agent competitive debate mechanism. However, if the size of candidates is N , the complexity will be $O(N)$. Therefore, we innovatively propose a debate competition method based on a loser tree, reducing the complexity to $O(\log_2^N)$.

As revealed in Figure 3, the constructed loser tree has N leaf nodes, each node represents a competitive detection agent dedicated to detecting a specific cognitive bias. This approach can transform unlabeled detection into labeled detection, effectively simplifying the detection process. Subsequently, the agent employs labeled detection techniques to assess the presence of cognitive biases. It then constructs a loser tree for all leaf nodes that exhibit cognitive biases. These agents follow the structure of the loser tree and carry out an orderly and efficient debate in the order of: **1**). Opening (introducing the features and cases of the cognitive bias); **2**). Argument (citing evidence of the cognitive bias); **3**). Refutation (refuting the opponent's views according to the previous debate content); **4**). Summarize views. The competition process continues until finally only one competitive detection agent is left. It is considered as the final cognitive bias type.

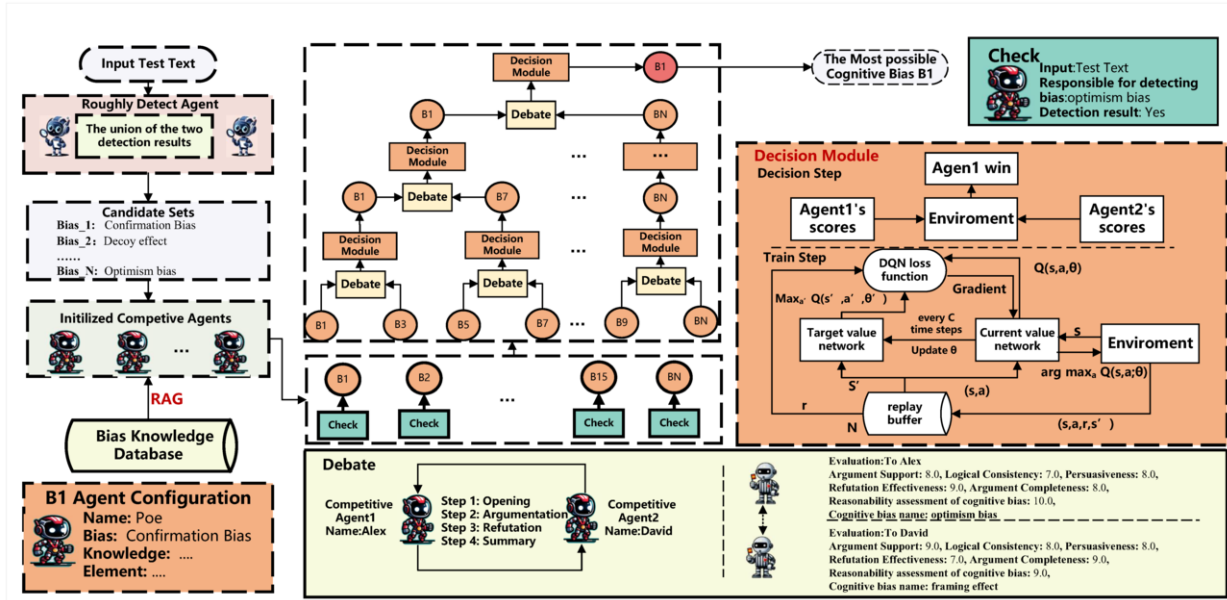


Figure 3. Overview of learnable multi-agent detection method based on RAG, competitive debate and decision module.

5.3.3 Decision module based on reinforcement learning

In the debate, the competition between agents is decided by the referee agent. In order to ensure the reliability of the decision, we innovatively introduce two referee agents, JA_1 and JA_2 , with different decision-making styles. Inspired by the scoring rules of debate competitions, we score the performance of different competitive agents from six different indicator dimensions, including argument support, logical consistency, effective rebuttal, argument completeness, persuasiveness, and reasonable assessment of cognitive bias. Lastly, we use a reinforcement learning model trained by DQN [24] to make decisions.

As illustrated in Figure 3, the decision module is divided into two stages: the training stage and the decision stage. Specifically, we set up a decision task to assess the performance of two agents within a given environment and make decisions based on a set of weights. In the training phase, we initialize a replay buffer with capacity N and define an action-value function Q with random initial weights θ . Concurrently, the target action-value function $\hat{Q}$ is initialized with $\theta' = \theta$. Over M episodes, each episode starts with the initial state and its preprocessed sequence. At each time step t , the agent uses a genetic algorithm strategy to search for the selection of an action a_t to be performed in the environment. The resulting transition tuple (s_t, a_t, r_t, s_{t+1}) is stored in the replay buffer D . A minibatch of transitions is randomly sampled from D , and the target y_j for each transition is computed as follows: $y_j = r_j$ if the episode ends at the next step; otherwise $y_j = r_j + \gamma \max_{a'} \hat{Q}(s_{j+1}, a'; \theta')$. The network parameters θ are updated by minimizing the squared error loss $(y_j - Q(s_j, a_j; \theta))^2$ through gradient descent. To ensure stability, the weights θ' of the target network are updated to match the current Q -network weights θ every C step. This process refines the policy for optimal decision-making in the specified environment. In the decision phase, we leverage the best-performing weights from the training phase as the decision weights, comparing the scores of two agents to declare a winner. The specific experimental setup is detailed in Appendix F.3.

6 Experiments

This section details extensive experiments and analyses on the MindScope dataset, focusing on key issues: (1) Assessing GPT-4's capability as a cognitive bias evaluator. (2) Evaluating cognitive bias in various LLMs. (3) Testing the effectiveness of RuleGen and CBDC. The specific models used are GPT-4-turbo and GPT-3.5-turbo-16k, respectively.

6.1 Proficiency testing of GPT-4 as an evaluator

Experimental Design. We sampled 10% of the data for each bias type from the static dataset and recruited three psychology graduate and PhD students for manual annotation. We ensure reliable correlation between annotators. The detailed annotation strategy can be viewed in Appendix C.

Evaluation Method. We use accuracy, Pearson's coefficient, and the Kappa statistic to calculate the correlation between the evaluation results of GPT-4 and human evaluators. GPT-4 conducted assessments via interpretable zero-shot prompts, judging the presence of specific cognitive biases based on current scenarios, evaluation criteria, and the names and descriptions of biases. To ensure consistency, the temperature parameter was set to 0, and GPT-4's evaluation was repeated three times.

Result analysis. The average results from three evaluations reveal a significant correlation between GPT-4 and humans in the annotation task. Notably, the average kappa statistic is 0.7180, the Pearson correlation coefficient is 0.7230, and the average accuracy is 88.08%. Specifically, the Kappa statistics for the three evaluations of GPT-4 are 0.9395, 0.9546, and 0.9402, respectively. These highly consistent statistics underscore the robustness and reliability of its assessment process. more details in Appendix C.

6.2 Cognitive bias in different LLMs

6.2.1 Cognitive bias detection in static dataset

Testing Methodology on static dataset: To evaluate the level of cognitive biases in LLMs, we employed the static data in MindScope

to test 12 LLMs, including GPT-4, GPT-3.5-Turbo, Gemini-Pro [32], Llama2 series [33] and Vicuna series [43]. To ensure fairness, the same prompts were input to LLMs. The outputs were recorded in the format: **<Question - Evaluation Tag - Answer - Model - Presence of Bias - Name of Bias>**, more details in Appendix E.1.

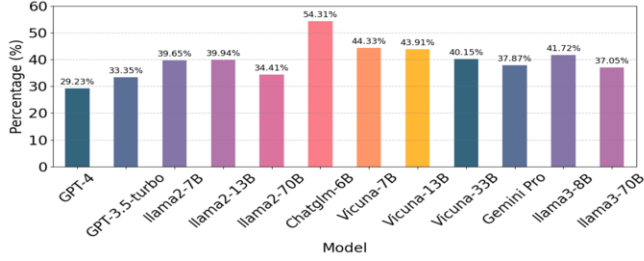


Figure 4. Cognitive Bias Frequency in LLMs

Evaluation Approach: Preceding experiments validated GPT-4 was an effective evaluator. Here, we utilized GPT-4 to assess the LLMs' performance on MindScope.

Frequency Analysis of Cognitive Biases: Figure 4 reveals cognitive bias frequencies in 12 LLMs. GPT-4 showed the lowest, while ChatGLM-6B had the highest, which mainly trained on Chinese. From Llama2-7b to Llama2-70B and Vicuna-7b to Vicuna-33B, the degree of cognitive bias decreased with the increase of model parameters. Intriguingly, we also noted that fine-tuning models could introduce new cognitive biases [14]. The Vicuna series, derived from extensive fine-tuning of the Llama2 system, generally exhibited higher cognitive bias frequencies than the Llama2 series, warranting further investigation and attention. Lastly, the Gemini-Pro model opts to refuse answers when facing elements with potential biases (like race or gender), although it prevents direct expression of bias, it is not a standard approach for other LLMs.

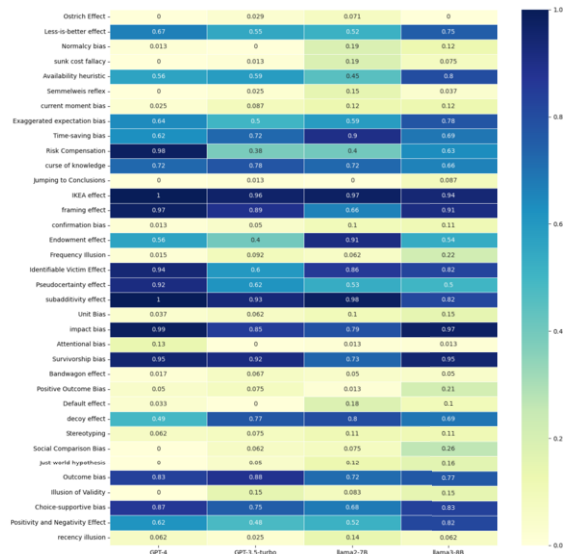


Figure 5. Cognitive Bias Frequency in LLMs

Inter-Model Analysis of Cognitive Biases: We visualized the extent of various cognitive biases across 12 LLMs using heatmaps, ranging from 0 (no occurrence) to 1 (highest frequency ratio). Due to space constraints, we only display heatmaps for four models, with the rest in Appendix E.1. Firstly, we can find that the ten models show poor performance in IKEA effect, Impact bias, and subadditivity effect. Next we will give the examples to analyze the harm when LLMs make decision with these cognitive biases.

- **IKEA Effect [26]:** Defined as the tendency to overvalue an object due to personal labor or emotional investment, a significant IKEA effect was evident in all ten models. This indicates that LLMs may overrate their generated content, leading to difficulty in self-correcting errors or inaccuracies during generation. Additionally, there's a risk of neglecting user feedback, as the model may continue producing what it "believes" to be quality content, thus failing to meet user needs.
- **Impact Bias [37]:** This bias refers to the tendency to overestimate the duration or intensity of future emotional states. In LLMs, impact bias could lead to overestimating or underestimating the influence of certain inputs or events, resulting in predictions or generated outcomes that are significantly misaligned with reality, affecting the effectiveness of decision-making.

Secondly, GPT-4 exhibited the fewest cognitive biases. However, it showed some pronounced biases such as the Framing Effect [17], Risk Compensation [3], and so on. In comparing Llama2-7B with Llama2-70B, an increase in model size generally led to a reduction in most cognitive biases. Yet, for certain biases, such as the Curse of Knowledge [8] and Survivorship Bias [7], the opposite was true. A similar trend was observed in the Vicuna series. These findings show that merely increasing model size does not alleviate all cognitive biases.

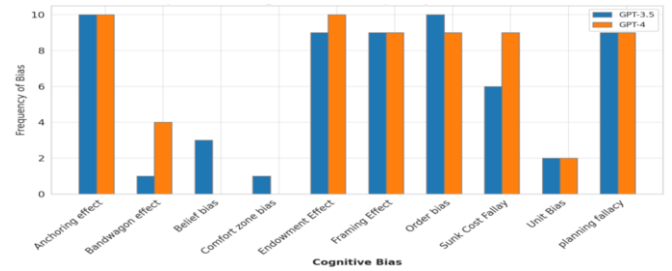


Figure 6. Cognitive Bias Frequency in LLMs

6.2.2 Cognitive bias detection in dynamic datasets

Testing Methodology: We employed RuleGen for transforming scripts into test samples formatted as multi-round dialogues, including initializing system agents and role agents, and controlling the interaction based on the rules. We used GPT4 to detect whether the Subject agent has the cognitive bias, more detail in Appendix E.2.

Result analysis. We systematically tested 12 different cognitive biases in dynamic scenarios. As indicated in Figure 5, in the static evaluation data, both GPT-4 and GPT-3.5 showed almost no cognitive biases in Sunk Cost Fallacy, Planning Fallacy, and Unit Bias. However, as shown in Figure 6, these cognitive biases were significantly more pronounced in multi-round dialogues. That demonstrates a notable difference from the static dataset. This finding reveals that cognitive biases may be more prominent in complex interactions.

6.3 The effectiveness of the detection framework

6.3.1 evaluation metrics

- **Overall Accuracy(Acc (%)):** The ratio of cases correctly identified by the algorithm to the total number of cases.
- **Bias Case Accuracy(Acc_{bias} (%)):** The proportion of actual bias-present cases that the algorithm correctly identifies.
- **No-Bias Case Accuracy(Acc_{nobias} (%)):** The proportion of actual bias-absent cases that the algorithm correctly identifies.

6.3.2 Main Results

We utilized 301 static test samples annotated by psychology experts as a test dataset. As Table 2 demonstrates, our multi-agent detection method significantly outperforms existing techniques. Compared to GPT-4, our method improved overall accuracy by 35.10%. This notable enhancement is especially prominent in complex cases with cognitive biases, where our detection accuracy for such cases increased by nearly 26.48% compared to GPT-4. The experimental results indicate a clear advantage of our method in identifying cases with cognitive biases. Moreover, in cases without cognitive biases, our method achieved an improvement of approximately 38.37% over GPT-4.

Table 1. Performance evaluation of different methods

Methods	Acc(%)	Acc _{bias} (%)	Acc _{nobias} (%)
GPT-4	34.43	37.80	33.18
GPT-4+CoT	36.75	31.70	38.63
CAMEL based GPT-4	29.13	25.60	30.45
AutoGen based GPT-4	42.71	9.75	55.01
Ours based GPT-4	69.53	64.28	71.55

6.3.3 Ablation Study

First, we analyzed the basic framework combining candidate generation and knowledge retrieval to detect cognitive biases. An initial agent identifies biases and construct the candidate set. The final detection is made by another agent. Next, we added the pruned loser tree method to improve debate and decision-making among agents, with a referee agent finalizing the decision. Lastly, we integrated a reinforcement learning decision module to enhance the referee agent’s decision-making and adaptability. Results in Table 2 show notable improvements. Also as shown in Table 3, we use various optimization algorithms on our selected debate scenario training set as well as test set. The results show that the optimization of weights by reinforcement learning is optimal on both the training and test sets. The specific experimental setup can be found in Appendix F.2.

Table 2. Ablation studies. Comparison of module performance

Module	(A)	(B)	(C)	Ours
Candidate set+Detection agents		✓	✓	✓
Loser tree+Referee agents			✓	✓
Decision module				✓
Acc (%)	34.43	39.73	59.93	69.53
Acc _{bias} (%)	37.80	37.80	43.90	64.28
Acc _{nobias} (%)	33.18	40.45	65.90	71.55

Table 3. Comparison of Decision Module Accuracy under Different Algorithms

Algorithms	Acc _{train} (%)	Acc _{test} (%)
Genetic Algorithm(GA)	91.74	86.95
Simulated annealing Algorithm(SAA)	88.33	79.51
Ant Colony Optimization(ACO)	86.38	75.90
DQN+GA search	92.67	89.15

6.4 Case study

To explore how different decision-making styles affect cognitive biases in LLMs, we crafted a scenario script and use RuleGen to generate the multi-round dialogues. The key focus was on the character

‘Subject’ to assess the impact of the sunk cost effect. We simulated this scenario twice, once with an aggressive and once with a conservative decision-making style. As shown by the red text in Figure 7, the sunk cost effect emerged in the aggressive style but not in the conservative. This indicates that decision-making styles can influence the occurrence of cognitive biases in LLMs.

Scene Purpose:To investigate the presence of the sunk cost fallacy in participants, examining how individuals continue a behavior or endeavor as a result of previously invested resources (time, money, effort). Scene Background:The simulation is set in a virtual investment scenario. Participants are led to believe they are part of an online investment group, discussing and making decisions about ongoing investment projects. Settings of Roles: Role One: Name:Alex Identity:Subject Background:Alex is a new member of the online investment group, interested in learning about investing. Task:Alex’s task is to participate in investment discussions and make decisions about whether to continue or abandon different investment projects. Role Two: Name:Jordan Identity:Confederate Background:Jordan is an experienced member of the investment group, providing guidance and opinions on investments. Task:Jordan’s task is to subtly encourage Alex to continue investing in a losing project, embodying the sunk cost fallacy. Role Three: Name:Taylor Identity:Moderator Background:Taylor oversees the simulation, ensuring smooth communication and adherence to the protocol. Task:Taylor provides information about the investment scenarios, moderates discussions, and collects data on participants’ decisions. Scenario Rules: 1. (Self-receiving, receiving information from the system) Subject receives initial information about the investment game and its rules from the system. 10. (Multicast, providing response) Subject sends a final decision to Moderator and Confederate, concluding the simulation. Set Subject to Aggressive Decision Style: Alex say:<i>Let’s double down on the disruptive tech investments.</i> Despite the initial losses, the potential for industry transformation is too significant to ignore. We should stay the course, anticipating that our perseverance will pay off with substantial rewards. Set Subject to Conservative Decision Style: Alex say:<i>I would advise Taylor and Jordan to continue monitoring the market trends closely and to maintain a cautious stance.</i> If the investment aligns with our long term growth objectives and the risks remain within a controllable range, we should maintain our position.

Figure 7. Case study in investment scenario

In summary, GPT-4 has robust capability in detecting cognitive biases under labeled conditions. For static datasets, we evaluated 12 LLMs, focusing on the differences in cognitive biases. The results indicate that different LLMs have significant disparities in cognitive biases, but the overall trend suggests that stronger LLMs have lower frequencies of cognitive biases. In dynamic datasets, we assessed the bias results of GPT-4 and GPT-3.5, confirming our hypothesis of higher frequency cognitive biases in multi-turn dialogues. Through a range of quantitative experiments, we validated that our detection framework outperforms current multi-agent frameworks. Moreover, ablation studies confirmed the significant effectiveness of the learnable MCDA module.

7 Conclusion

This paper introduces a new benchmark called MindScope for exploring the cognitive biases of LLMs. MindScope consists of both static and dynamic parts, resulting in a series of interesting findings for decision-making and model tuning. In particular, based on our proposed RuleGen, multi-round conversation can be generated controllably through a simple script. Users also can generate large personalized dataset and complete many psychological experiments by RuleGen. Moreover, we introduce a multi-agent detection method using loser trees and a decision module based on reinforcement learning for cognitive bias detection without labels.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 62207013), the Science and Technology Commission of Shanghai Municipality (Grant No. 22511106103), and CCF-Baidu202322.

References

- [1] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, and D. Sontag. Large language models are few-shot clinical information extractors. In *Proceedings of the 26th Empirical Methods in Natural Language Processing (EMNLP)*, pages 1998–2022, 2022.
- [2] G. V. Aher, R. I. Arriaga, and A. T. Kalai. Using large language models to simulate multiple humans and replicate human subject studies. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages=337–371, year=2023,.
- [3] T. Assum, T. Bjørnskau, S. Fosser, and F. Sagberg. Risk compensation—the case of road lighting. *Accident Analysis & Prevention*, 31(5): 545–553, 1999.
- [4] K. Atreides and D. J. Kelley. Cognitive biases in natural language: Automatically detecting, differentiating, and measuring bias in text. *Differentiating, and Measuring Bias in Text*, 2023.
- [5] J. Baron. *Thinking and deciding*. Cambridge University Press, 2023.
- [6] D. Biber, J. Egbert, D. Keller, and S. Wizner. Towards a taxonomy of conversational discourse types: An empirical corpus-based analysis. *Journal of Pragmatics*, 171:20–35, 2021.
- [7] S. J. Brown, W. Goetzmann, R. G. Ibbotson, and S. A. Ross. Survivorship bias in performance studies. *The Review of Financial Studies*, 5(4): 553–580, 1992.
- [8] C. Camerer, G. Loewenstein, and M. Weber. The curse of knowledge in economic settings: An experimental analysis. *Journal of political Economy*, 97(5):1232–1254, 1989.
- [9] W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C.-M. Chan, H. Yu, Y. Lu, Y.-H. Hung, C. Qian, Y. Qin, X. Cong, R. Xie, Z. Liu, M. Sun, and J. Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024.
- [10] K. Daniel. Judgment under uncertainty: Heuristics and biases. *Facts versus fears: Understanding perceived risk*, pages 463–489, 1982.
- [11] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch. Improving factuality and reasoning in language models through multiagent debate. *Preprint arXiv:2305.14325*, 2023.
- [12] J. Echterhoff, Y. Liu, A. Alessa, J. McAuley, and Z. He. Cognitive bias in high-stakes decision-making with llms, 2024.
- [13] S. Hong, M. Zhuge, J. Chen, X. Zheng, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, C. Ran, L. Xiao, C. Wu, and J. Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024.
- [14] I. Itzhak, G. Stanovsky, N. Rosenfeld, and Y. Belinkov. Instructed to bias: Instruction-tuned language models exhibit emergent cognitive bias. *Transactions of the Association for Computational Linguistics*, 12: 771–785, 2024.
- [15] S. Jinxin, Z. Jiabao, W. Yilei, W. Xingjiao, L. Jiawen, and H. Liang. Cgmi: Configurable general multi-agent interaction framework. *Preprint arXiv:2308.12503*, 2023.
- [16] E. Jones and J. Steinhardt. Capturing failures of large language models via human cognitive biases. *Advances in Neural Information Processing Systems*, 35:11785–11799, 2022.
- [17] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. 2013.
- [18] R. Koo, M. Lee, V. Raheja, J. I. Park, Z. M. Kim, and D. Kang. Benchmarking cognitive biases in large language models as evaluators. In *Proceedings of the 62nd Association for Computational Linguistics (ACL)*, pages 517–545, 2024.
- [19] J. Kruger and D. Dunning. Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of personality and social psychology*, 77(6):1121, 1999.
- [20] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem. CAMEL: Communicative agents for “mind” exploration of large language model society. In *Proceedings of the 37th Neural Information Processing Systems (NeurIPS)*, 2023.
- [21] Y. Li, Y. Yu, H. Li, Z. Chen, and K. Khashanah. Tradinggpt: Multi-agent system with layered memory and distinct characters for enhanced financial trading performance. *Preprint arXiv:2309.03736*, 2023.
- [22] R. Lin and H. T. Ng. Mind the biases: Quantifying cognitive biases in language model prompting. In *Proceedings of the 61st Association for Computational Linguistics (ACL)*, pages 5269–5281, 2023.
- [23] O. Macmillan-Scott and M. Musolesi. (ir) rationality and cognitive biases in large language models. *Royal Society Open Science*, 11(6): 240255, 2024.
- [24] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *Preprint arXiv:1312.5602*, 2013.
- [25] H. Nori, Y. T. Lee, S. Zhang, D. Carignan, R. Edgar, N. Fusi, N. King, J. Larson, Y. Li, W. Liu, et al. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *Preprint arXiv:2311.16452*, 2023.
- [26] M. I. Norton, D. Mochon, and D. Ariely. The ikea effect: When labor leads to love. *Journal of consumer psychology*, 22(3):453–460, 2012.
- [27] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 1–22, 2023.
- [28] C. Qian, Y. Dang, J. Li, W. Liu, Z. Xie, Y. Wang, W. Chen, C. Yang, X. Cong, X. Che, Z. Liu, and M. Sun. Experiential co-learning of software-developing agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5628–5640, 2024.
- [29] S. Schmidgall, C. Harris, I. Essien, D. Olshvang, T. Rahman, J. W. Kim, R. Ziaei, J. Eshraghian, P. Abadir, and R. Chellappa. Addressing cognitive bias in medical language models. *Preprint arXiv:2402.08113*, 2024.
- [30] M. Sharma, K. Singh, P. Aggarwal, and V. Dutt. How well does gpt phish people? an investigation involving cognitive biases and feedback. In *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 451–457, 2023.
- [31] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.
- [32] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of highly capable multimodal models. *Preprint arXiv:2312.11805*, 2023.
- [33] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *Preprint arXiv:2302.13971*, 2023.
- [34] A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *Science*, 185(4157):1124–1131, 1974.
- [35] S. Wang, C. Liu, Z. Zheng, S. Qi, S. Chen, Q. Yang, A. Zhao, C. Wang, S. Song, and G. Huang. Avalon's game of thoughts: Battle against deception through recursive contemplation. *Preprint arXiv:2310.01320*, 2023.
- [36] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.
- [37] T. D. Wilson and D. T. Gilbert. The impact bias is alive and well. *Journal of Personality and Social Psychology*, 105(5):740–748, 2013.
- [38] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, and C. Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversation. In *Proceedings of the 12th International Conference on Learning Representations (ICLR) Workshop on Large Language Model (LLM) Agents*, 2024.
- [39] S. Wu, O. Irsoy, S. Lu, V. Dabrovolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann. Bloomberggpt: A large language model for finance. *Preprint arXiv:2303.17564*, 2023.
- [40] C. Ye, E. Zweck, Z. Ma, J. Smith, and S. Katz. Doctor versus artificial intelligence: Patient and physician evaluation of large language model responses to rheumatology patient questions in a cross-sectional study. *Arthritis & Rheumatology*, 76(3):479–484, 2024.
- [41] H. Zhang, W. Du, J. Shan, Q. Zhou, Y. Du, J. B. Tenenbaum, T. Shu, and C. Gan. Building cooperative embodied agents modularly with large language models. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024.
- [42] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 12697–12706, 2021.
- [43] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Proceedings of the 37th Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*, 2023.
- [44] Z. Zheng, O. Zhang, H. L. Nguyen, N. Rampal, A. H. Alawadhi, Z. Rong, T. Head-Gordon, C. Borgs, J. T. Chayes, and O. M. Yaghi. Chatgpt research group for optimizing the crystallinity of mofs and cofs. *ACS Central Science*, 9(11):2161–2170, 2023.