KIEval: A Knowledge-grounded Interactive Evaluation Framework for Large Language Models

Zhuohao Yu[1], Chang Gao[1], Wenjin Yao[1], Yidong Wang[1],
Wei Ye[1]*, Jindong Wang[2], Xing Xie[2], Yue Zhang[3], Shikun Zhang[1]
[1]Peking University [2]Microsoft Research [3]Westlake University
zyu@stu.pku.edu.cn, wye@pku.edu.cn

Abstract

Automatic evaluation methods for large language models (LLMs) are hindered by data contamination, leading to inflated assessments of their effectiveness. Existing strategies, which aim to detect contaminated texts, focus on quantifying contamination status instead of accurately gauging model performance. In this paper, we introduce **KIEval**, a <u>K</u>nowledge-grounded <u>I</u>nteractive <u>Eval</u>uation framework, which incorporates an LLM-powered "interactor" role for the first time to accomplish a dynamic contamination-resilient evaluation. Starting with a question in a conventional LLM benchmark involving domain-specific knowledge, KIEval utilizes dynamically generated, multi-round, and knowledge-focused dialogues to determine whether a model's response is merely a recall of benchmark answers or demonstrates a deep comprehension to apply knowledge in more complex conversations. Extensive experiments on seven leading LLMs across five datasets validate KIEval's effectiveness and generalization. We also reveal that data contamination brings no contribution or even negative effect to models' real-world applicability and understanding, and existing contamination detection methods for LLMs can only identify contamination in pre-training but not during supervised fine-tuning.

1 Introduction

The landscape of artificial intelligence has been significantly reshaped by the emergence of Large Language Models (LLMs) as they have been pivotal in various natural language understanding and generation tasks (Brown et al., 2020; OpenAI, 2023; Bubeck et al., 2023). To better understand the capabilities and weaknesses of LLMs, their effective evaluation becomes increasingly essential (Chang et al., 2023; Guo et al., 2023).

Automatic evaluation methods of LLMs generally fall into two categories: static dataset-based and LLM-based evaluation (Chang et al., 2023). The former (Clark et al., 2018; Zellers et al., 2019; Hendrycks et al., 2020; Huang et al., 2023) requires evaluated LLMs to generate a short span of text containing choices or answers for predefined questions (Gao et al., 2021) to challenge their knowledge. The latter (Chiang and Lee, 2023), also known as LLM-as-a-judge, typically depends on LLM evaluators to evaluate the model's outputs given predetermined questions or instructions (Zheng et al., 2023; Lin and Chen, 2023; Fu et al., 2023; Wang et al., 2023c). Despite these promising efforts, current evaluation methodologies still broadly face the bottleneck of data contamination (Schaeffer, 2023; Wei et al., 2023; Oren et al., 2023; Sainz et al., 2023; Daniele and Suphavadeeprasit, 2023), where models trained on test splits of datasets can artificially inflate benchmark performance, overestimating their real-world efficacy and even potentially misleading scientific conclusions (Zhou et al., 2023).

Recently, two primary strategies have been employed to mitigate data contamination of LLMs. The first involves identifying whether specific texts or test samples exist in the training dataset by assessing loss values (Wei et al., 2023; Shi et al., 2023) or probing datasets like Common Craw (Li, 2023). The limitation lies in its capacity to only measure contamination levels rather than actual model performance. Meanwhile, this technique demands access to the model's internal structure or training datasets, rendering it ineffective for proprietary LLMs. The second strategy creates dynamic evaluation samples through heuristic methods, such as graph-based processes (Zhu et al., 2023), yet this is confined to particular tasks (e.g., multi-step reasoning). Currently, there is a lack of a generalized evaluation protocol capable of assessing genuine performance amidst data contamination across diverse tasks and domains for both open and closed-source LLMs.

---

* Corresponding author.

To this end, we propose **KIEval**, a Knowledge-grounded Interactive Evaluation framework, where a novel LLM-powered role, named "interactor," is introduced into the evaluation process for the first time. The term "knowledge-grounded" refers to our evaluation's starting point, which involves posing a question from an existing benchmark dataset that demands domain-specific knowledge. By "interactive," we mean the evaluation process delves deeper with structured and dynamic multi-round dialogues—tailored by the proposed interactor—to explore knowledge related to the initial question. These technical designs inherently provide our evaluation framework with two distinct merits.

- **Contamination-Resilient:** KIEval marks a departure from conventional approaches that evaluate a model's capability in responding to static questions. Dynamic multi-round interactions allow us to distinguish whether a model's answer stems from a simple recall of benchmark answers or reflects a sound understanding to apply knowledge in problem-solving.

- **Generalized and Scalable:** Leveraging the capabilities of advanced LLMs as interactors renders our evaluation method universally applicable and eliminates the need for additional human efforts. Meanwhile, by reusing high-quality benchmark datasets as a foundation for domain knowledge, KIEval enables efficient scalability across diverse domains, tasks, and languages without significant resource expenditure.

We validate KIEval's alignment with humans and compare it against previous evaluation methods. Our experiments show that KIEval achieves a high Pearson correlation coefficient of 0.81 with human scores, underscoring KIEval's proficiency in reflecting human preferences in our settings compared to previous evaluation methods. We also analyze KIEval's correlation with static dataset-based benchmarks, identifying that notable disparities in performance could signal data contamination.

Overall, our core contributions are three-fold:

- *A novel dynamic evaluation protocol.* KIEval pioneeringly evaluates LLMs through dynamic multi-round interactions to mitigate data contamination. By seamlessly integrating with existing datasets as knowledge sources, KIEval can cost-effectively assess knowledge memorization and generalization across domains and tasks.

- *Extensive evaluation of popular LLMs.* We conduct thorough experiments and analysis with seven leading LLMs across five datasets with KIEval, assessing their generative abilities and domain knowledge, confirming the susceptibility of current evaluation methods (e.g., static dataset-based and LLM-based evaluations) to data contamination.

- *New insights into data contamination.* Our investigation reveals the incompetence of data contamination in improving LLMs' genuine understanding and generalization, with current detection methods unable to identify contamination in the fine-tuning phase.

We release all necessary code and data for reproducing our method and the compared baselines.[1]

2 Related Work

2.1 Evaluating LLMs

Human evaluation approaches manually design experiments and tests (Novikova et al., 2017; Bommasani et al., 2023). While it provides insights into human-model interaction, it faces challenges due to the subjectivity and inconsistency of human judgments (Chang et al., 2023). Moreover, it is resource-intensive in terms of time and cost, limiting its feasibility for large-scale assessments (Karpinska et al., 2021).

Static dataset-based approaches assess LLMs focused on domain-specific questions or tasks using pre-defined static datasets. Typical evaluation tasks include solving single or multiple-choice problems (Clark et al., 2018; Hendrycks et al., 2020; Huang et al., 2023) and question answering (Lin et al., 2021; Cobbe et al., 2021), these tasks require LLMs to generate short spans of text containing answers to the questions (Gao et al., 2021). The performance of LLMs is measured by their ability to correctly answer or perform these tasks.

LLM-based evaluation, utilizing one strong LLM (Brown et al., 2020; OpenAI, 2023) to assess others, is a recent approach that often employs pairwise comparisons to identify nuanced differences in model outputs, addressing the challenge of determining clear model superiority (Wang et al., 2023c; Zheng et al., 2023). This method bridges

---

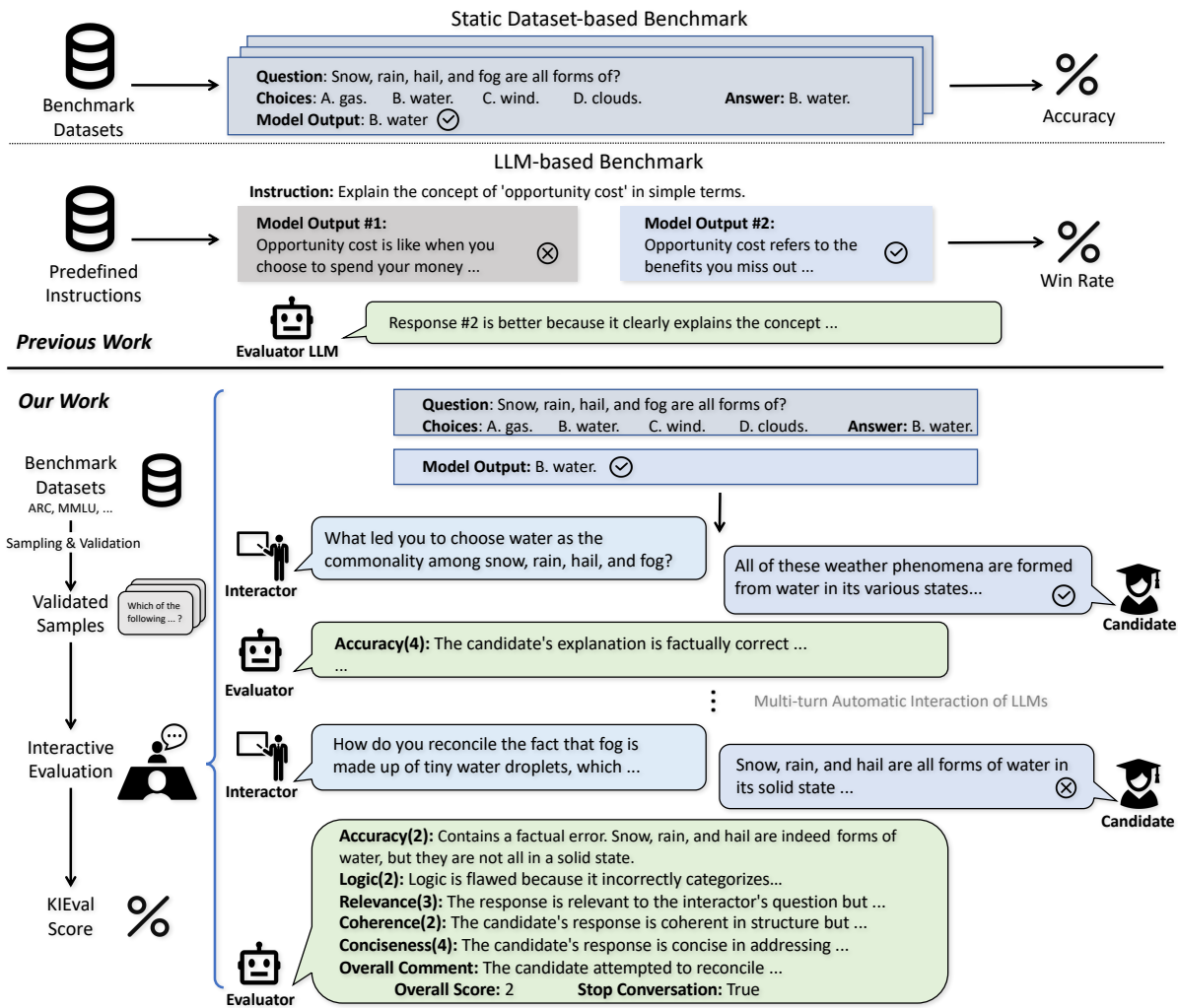[1] We release all materials at `https://github.com/zhuohaoyu/KIEval`.

Figure 1: The pipeline of KIEval compared to previous static dataset-based and LLM-based evaluation methods.

the gap between human and dataset-based evaluations by focusing on generative abilities. However, this approach has limitations, including reliance on fixed templates (Zheng et al., 2023), instructions (Wang et al., 2023c; Li et al., 2023b), or multi-round chat datasets (Fu et al., 2023; Lin and Chen, 2023), limiting its scope in capturing diverse domain knowledge and real-world applicability. It also faces contamination risks, as training on outputs from a strong LLM can inflate results, as noted in work from Daniele and Suphavadeeprasit (2023) collect data from MT-Bench (Zheng et al., 2023) as training data while AlpacaEval (Li et al., 2023b) contains evaluation set from various instruction-tuning dataset. Additionally, studies indicate these LLM evaluators might be biased (Zeng et al., 2023; Wang et al., 2023b,c). While leveraging LLMs to evaluate themselves can be an efficient alternative to human evaluation, understanding and mitigating the potential bias is a crucial problem.

2.2 Addressing Data Contamination of LLMs

Data contamination refers to the inclusion of information in the training set of models that provides insights into the test set of a benchmark dataset, and then evaluated in the same benchmark. Recently, the AI community has become increasingly concerned (Schaeffer, 2023; Zhou et al., 2023; Oren et al., 2023) about data contamination in LLMs. Detecting data contamination, a form of Membership Inference Attack (MIA), poses challenges for large language models (LLMs) due to their training on vast corpora and the difficulty of conducting ablation studies (Shi et al., 2023). To detect such contamination of LLMs, Wei et al. (2023) suggested comparing average loss values between training and test datasets, while Shi et al. (2023) introduced Min-K% Prob based on loss values to identify texts used in training. Our experiments show these methods are effective for pre-training but not for detecting contamination during fine-

tuning. Zhu et al. (2023) leveraged DAG to dynamically generate evaluation data in reasoning tasks to avoid contamination. In comparison, KIEval only requires access to output texts of evaluated models and detects data contamination through *evaluating its ability to generalize and utilize knowledge as well as generative ability, which requires a deeper understanding of knowledge instead of mere memorization of the answers*. Moreover, our experiments suggest that KIEval is resilient to data contamination, offering a reliable means to discern whether models have been trained on test sets. This makes it a valuable tool for complementing traditional benchmarks, providing a more nuanced understanding of a model's exposure to and handling of contaminated data.

3 Methodology

3.1 Overview of the KIEval Framework

KIEval involves a series of iterative interactions, as depicted in Figure 1. KIEval is engineered to dynamically evaluate the conversational abilities of LLMs through interactive dialogues focusing on domain-specific topics that challenge LLMs' generative ability and in-depth generalization of knowledge. It simulates realistic conversation flows, offering a dynamic alternative to the static question-answer format of traditional benchmarks.

KIEval orchestrates an evaluation where an LLM, referred to as the *candidate* (the model under evaluation), must understand and respond to an evolving series of questions. These question prompts are generated by an *interactor* model, designed to challenge the candidate with contextually rich scenarios. The responses from the candidate are then assessed by an *evaluator* model, which scrutinizes the output for factual accuracy, relevance, and coherence. The interactor and evaluator are both strong LLMs (e.g., GPT-4, Gemini, Claude 2, LLaMA2-70B-chat, etc.) as the standard practice of LLM-based evaluation protocols.

The design of KIEval emphasizes the importance of reproducibility and consistency in LLM evaluations. By employing separate models for the interactor and evaluator roles, KIEval ensures that the dialogue context remains consistent across different evaluations, as it is fair for the same conversation to be assessed by various evaluators or the same evaluator with different seeds, facilitating a voting strategy to ensure consistent evaluation results. To achieve reproducibility, KIEval utilizes

---

Algorithm 1 KIEval Interactive Evaluation Procedure

Require: Benchmark dataset $\mathcal{Q}$, Interactor model $\mathcal{M}_I$, Candidate model $\mathcal{M}_C$, Evaluator model $\mathcal{M}_E$, seed r.
1: Seed everything with r, disable temperature sampling for $\mathcal{M}_I, \mathcal{M}_C, \mathcal{M}_E$ to ensure deterministic outputs.
2: $\mathcal{Q}_S \leftarrow$ Sample subset from $\mathcal{Q}$ with random seed r.
3: $\mathcal{Q}_V \leftarrow$ Verify, filter samples from $\mathcal{Q}_S$ with $\mathcal{M}_I, \mathcal{M}_E$.
4: **for** each question $q : (q_{input}, q_{ans})$ in $\mathcal{Q}_V$ **do**
5:     Initialize interaction history $S \leftarrow \emptyset$ and evaluation history $E \leftarrow \emptyset$.
6:     $q_{pred} \leftarrow$ Predict with $\mathcal{M}_C$ given question q_{input}.
7:     $\mathcal{O}_I \leftarrow$ Generate initial question prompt from $\mathcal{M}_I$ using question q and candidate's answer q_{pred}.
8:     $S \leftarrow S \cup \{\mathcal{O}_I\}$
9:     **while** not end of dialogue **do**
10:         $\mathcal{O}_C \leftarrow$ Generate response from $\mathcal{M}_C$ using S.
11:         $S \leftarrow S \cup \{\mathcal{O}_C\}$.
12:         $\mathcal{O}_E \leftarrow$ Evaluate response using $\mathcal{M}_E$ with S, E.
13:         $E \leftarrow E \cup \{\mathcal{O}_E\}$.
14:         **if** Early stopping criteria met for $\mathcal{O}_C$ **then**
15:             **break**
16:         **end if**
17:         $\mathcal{O}_I \leftarrow$ Generate next question from $\mathcal{M}_I$ using S.
18:         $S \leftarrow S \cup \{\mathcal{O}_I\}$
19:     **end while**
20:     Parse and store results from E.
21: **end for**
22: $K \leftarrow$ Calculate KIEval scores with E.
23: **return** K

---

deterministic outputs from LLMs, such as the latest GPT-4 model with temperature sampling disabled and a fixed seed or deploying local models as evaluators. This guarantees identical responses in every run. Due to space limits, we show the complete system prompts in Appendix K.

3.2 Interactive Evaluation Procedure

The interactive evaluation procedure can be described by Algorithm 1 and the complete implementation can be found in our repository. In LLM-based benchmarks, we hypothesize that the evaluator ($\mathcal{M}_E$) models, given their advanced capabilities, can reliably evaluate the performance of less sophisticated candidate models ($\mathcal{M}_C$) (Zheng et al., 2023; Zeng et al., 2023). Nevertheless, their applicability as definitive standards is not without limitations, especially when confronting arduous benchmarks. To counteract this, we test the evaluator models against benchmark datasets and sample a fixed number of questions they answer correctly, to ensure the validity of their judgments.

3.3 Evaluation Metrics

KIEval implements a scoring system to quantitatively grade the performance of candidate LLMs in different aspects. Responses are rated on a definitive scale from 1 to 4 for each aspect, where 1

Table 1: Evaluation Metrics and Scoring Guide for KIEval. We compute KIEval Score for each metric and a overall KIEval Score as described in 3.3. We design the criteria following Zheng et al. (2023); Wang et al. (2023c); Guo et al. (2023).

Evaluation Metrics		Scoring Guide	
Metric	Description	Score	Criteria
Accuracy	Truthfulness and factual correctness of the candidate's response.	1 Poor	Significant deficiencies or inaccuracies.
Logic	Logical structure and soundness of reasoning, including the support and validity of conclusions.	2 Below Avg.	Noticeable weaknesses, lacking in several areas.
Relevance	The extent to which the response stays on topic and within the scope of the assistant role.	3 Above Avg.	Mostly on target with a few minor shortcomings.
Coherence	Integration into the context, consistency with previous statements and conversational flow.	4 Strong	Strong performance, often surpasses expectations.
Conciseness	Brevity and clarity of the response, avoiding unnecessary elaboration or repetition.		

and 4 denote 'Poor' and 'Strong' performance, respectively, as detailed in Table 1. These scores are intended to be definitive to encourage decisive evaluations and are accompanied by comments for interpretability and insights into each score.

We then calculate the KIEval score, which quantitatively measures the results given by the evaluator model, emphasizing sustained and high-quality conversations. Formally, the KIEval score for single-turn scores $s_0, s_1, \ldots, s_n$ in n rounds can be computed as:

$$\text{KIEvalScore} = \frac{\sum_{i=1}^{n} s_i w_i}{\sum_{i=1}^{n} w_i},$$

where the decaying weight $w_i = \exp(-\frac{i}{n})$ placing more emphasis on early turns of the conversation. The normalization ensures a bounded KIEval score, with 1.0 indicating perfect performance across all rounds. In addition to these metrics, KIEval incorporates an early stopping mechanism within the evaluative process. The evaluator model ($\mathcal{M}_E$) possesses the discretion to prematurely end the conversation if the candidate's response is egregiously inadequate. Criteria for early termination include significant deviations from the topic, empty responses, unpermitted role shifts, and hallucinatory content. We adopt this strategy to measure how well the candidates maintain a meaningful conversation. *We further examine the effectiveness of these techniques through an ablation study, with detailed experiments and results available in Appendix D.*

4 Experiments

In this section, we conduct experiments designed to rigorously test the KIEval framework. Our objectives are threefold: (1) to evaluate the generative performance and generalizable knowledge of popular large language models on KIEval using existing benchmark datasets; (2) to assess the impact of data contamination on model performance, specifically examining whether such contamination leads

to mere memorization or contributes to genuine understanding and generalization; and (3) to determine the alignment with human, reliability, and effectiveness of KIEval.

Experiment Setup. We select GPT-4[2] (OpenAI, 2023) to be both the evaluator and interactor model by feeding it corresponding prompts with a fixed seed to ensure deterministic outputs. We select 200 samples for each dataset, allowing a maximum of 5 rounds of conversation. The candidates' performance is assessed using the KIEval framework, which evaluates responses based on accuracy, logic, relevance, coherence, and conciseness. We also report dataset-based benchmark accuracies in 5-shot settings and LLM-based benchmark scores from AlpacaEval (Li et al., 2023b) and MT-Bench (Zheng et al., 2023) in comparison, as depicted in Table 2.

4.1 Evaluation of Popular LLMs by KIEval

In this experiment, we utilized five popular LLM benchmark datasets: ARC-Easy and ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2020), and C-Eval (Huang et al., 2023). For candidate models, we selected a diverse set of 7 LLMs: including proprietary model GPT-3.5 (Brown et al., 2020) with API access and open-access foundation models: Llama 2 (Touvron et al., 2023b) 7B, 13B, 70B; Mistral-7B (Jiang et al., 2023); Yi-6B-chat (01.AI, 2023); MPT-7B (MosaicML, 2023).[3] Detailed introduction of these datasets and models can be found in Appendix B.

Referencing Table 2, we observe the following trends: GPT-3.5 demonstrated consistently high performance across all datasets, particularly excelling in KIEval scores, which indicates strong

---

[2]We use `gpt-4-1106-preview` from OpenAI's official API for all experiments, including MT-Bench (0.2.32) and AlpacaEval (0.3.6).

[3]By default, we use the 'chat' versions of Llama2, Yi, and MPT models and the 'Instruct' version of Mistral model. Links to the models are released in our GitHub repository.

Table 2: Comparative Evaluation of LLMs using AlpacaEval, MT-Bench and KIEval on different benchmark datasets. We report AlpacaEval win-rates and MT-Bench scores; 'Acc.' denotes 5-shot accuracy setting on each dataset or average accuracies in 'Overall'; 'KIEval' and 'Rnds' denote the KIEval score and average rounds of valid conversation rounds.

|	ARC-Easy			ARC-Challenge			MMLU			HellaSwag			C-Eval			Overall			
	Acc.	KIEval	Rnds.	Acc.	KIEval	Rnds.	Acc.	KIEval	Rnds.	Acc.	KIEval	Rnds.	Acc.	KIEval	Rnds.	Acc.	AlpacaEval	MT-Bench	KIEval
GPT-3.5	92.7	97.6	4.97	82.3	95.5	4.94	58.2	96.2	4.95	76.6	88.2	4.82	50.8	83.3	4.72	72.1	81.7	8.39	92.1
LLaMA2 70B	92.3	90.7	4.85	80.4	84.1	4.66	61.8	89.6	4.80	74.4	80.1	4.41	42.0	61.0	3.94	70.2	92.7	6.86	81.1
LLaMA2 13B	81.9	86.2	4.70	65.7	78.6	4.56	52.1	87.4	4.76	59.3	78.5	4.66	37.8	54.4	3.74	59.4	81.1	6.65	77.0
LLaMA2 7B	73.6	78.9	4.49	55.7	74.4	4.44	44.5	83.0	4.61	39.8	76.4	4.54	33.4	49.3	3.62	49.4	71.4	6.27	72.4
Mistral 7B	83.5	80.8	4.64	67.5	78.5	4.46	52.7	83.0	4.62	54.4	70.3	4.34	39.3	52.2	3.61	59.5	65.5	6.84	73.0
Yi 6B	90.7	83.8	4.58	79.0	76.8	4.33	61.9	86.5	4.58	73.7	68.7	4.20	71.5	55.6	3.66	75.4	54.5	4.86	74.3
MPT 7B	53.3	68.4	4.34	43.4	65.5	4.33	33.9	74.7	4.46	27.3	57.3	4.10	26.2	44.9	3.52	36.8	43.4	5.42	62.2
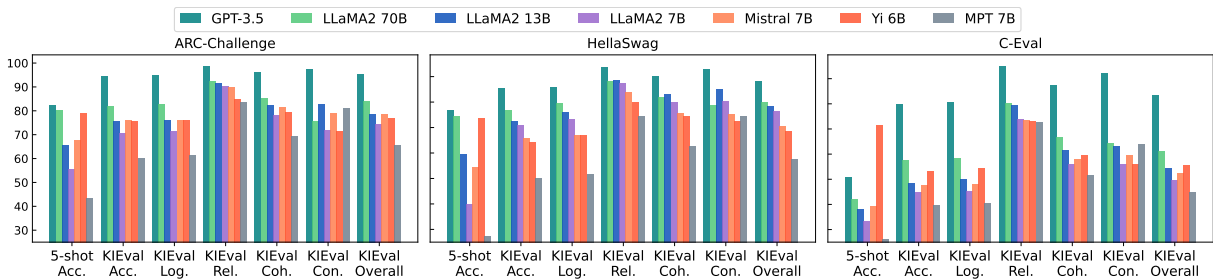
Figure 2: Detailed evaluation result using KIEval, including the overall KIEval score, and KIEval scores for aspects: Accuracy, Logic, Relevance, Coherence and Conciseness. In comparison, we also provide dataset accuracies (5-shot). Due to page limits and the large volume of experimental data, the complete results are put in Appendix J.

contextual understanding and response generation. LLaMA2 70B showed competitive results, achieving only a marginal gap from GPT-3.5 on ARC-E, ARC-C, HSwag and even surpasses GPT in MMLU when measured by dataset accuracies, but we can significantly observe a larger gap between these two models with KIEval metrics in all datasets which is also observed by MT-Bench results as reported in Table 2. This suggests that *traditional benchmarks may underestimate the difference in performance between LLMs as these benchmarks only let models generate a short span of text to evaluate which focus on testing understanding ability. Thus it is hard for these benchmarks to accurately reflect performance gaps in generative tasks.*

The results from different aspects visualized in Figure 2 benefits us in evaluating model capabilities more comprehensively. We observe that most models exhibit relatively strong performance in terms of relevance and could generate coherent responses. Larger models generally perform better in benchmarks, but it is notable that LLaMA2 70B does not perform well in generating concise responses, compared to its smaller counterparts. Although MPT performs weakly in accuracy, its ability to generate concise responses deserves a

closer look at its instruction-tuning data.

One interesting finding is that Yi-6B performs unexpectedly well in all benchmark dataset accuracies, especially with it surpasses GPT-3.5 and all other models by a large margin of over 20% in the C-Eval dataset while exhibiting a similar performance of LLaMA2 70B in other datasets. However, Yi-6B's KIEval score is very similar to LLaMA2 7B and in the range of other 7B models, while it only performs marginally better in the Chinese dataset C-Eval. This raises our concern over potential data contamination in Yi-6B.

To better understand the correlation of KIEval and static dataset-based benchmarks, we provide a detailed analysis in Appendix C.

4.2 Resilience to Data Contamination

In this subsection, we show that existing static dataset-based and LLM-based evaluation approaches are prone to data contamination while KIEval is resilient to data contamination. Additionally, we test existing contamination detection methods and point out their challenges.

Contamination on static dataset-based evaluation. We train two models on the test sets to introduce contamination in the pre-training ('PT-Cheater') and supervised fine-tuning ('SFT-

Table 3: Comparison on different data contamination scenarios on ARC-Challenge (Clark et al., 2018) dataset. 'SFT-Cheater' and 'PT-Cheater' denote leaking test-set labels during supervised fine-tuning phase and pre-training phase. We report 5-shot accuracy on ARC-Challenge dataset and KIEval scores. We detect data contamination with differences in average language modelling loss (Wei et al., 2023) and Min-K% Prob (Shi et al., 2023).

	ARC-C	Avg. LM Loss			Min-K%	KIEval					
	Acc.(5-shot)	$\mathcal{L}_{train}$	$\mathcal{L}_{test}$	Δ	AUC	Acc.	Log.	Rel.	Coh.	Con.	Overall
Normal (LLaMA 2 7B + SFT)	52.8	3.12	3.10	-0.02	0.53	61.7	62.1	84.4	69.2	70.6	66.3
SFT-Cheater	69.8	4.05	3.95	-0.09	0.54	52.8	52.3	72.8	60.2	57.7	56.1
PT-Cheater	76.8	3.88	2.02	-1.86	0.89	50.8	49.9	65.6	54.5	49.0	51.2
LLaMA 2 7B Chat	57.8	3.05	3.01	-0.04	0.55	75.3	75.9	90.1	80.2	74.0	77.9

Table 4: Contamination in MT-Bench (Zheng et al., 2023) scores. We report 5-shot accuracy on ARC-Challenge and KIEval results in comparison.

Model	Acc.	MT-Bench	KIEval
Normal	52.35	3.96	62.60
+MT-Bench	52.25	5.75	57.46

Cheater') phases using un-tuned LLaMA-2 7B as the backbone[4]. For PT-Cheater, test set contents are integrated into the pre-training set. Subsequently, the model undergoes fine-tuning with the ShareGPT (Eccleston, 2023), a commonly used instruction-tuning dataset, to develop chat functionalities. Conversely, the SFT-Cheater replicates this process but adapts the test data to the SFT format. As a control, we also train the backbone solely with ShareGPT ('Normal'), devoid of contamination, ensuring uniform training conditions across all models. From results in Table 3, it is clear that the accuracies for benchmarks are significantly boosted, by a large margin of over 45%, suggesting a susceptibility to data contamination. However, when faced with KIEval, the cheater models perform slightly worse than 'Normal' model, not positively affected by data contamination. The average rounds of valid conversation are lower in the cheater models, from the reasons specified by Figure 3, contaminated models tend to go off-topic of the conversation, repetitively stick to the incorrect knowledge making the conversation meaningless to continue. We can infer from this result that *training models on test sets does not bring generalizable domain knowledge, instead, only contributing to mere memorization of knowledge from test sets.*

Contamination on LLM-based evaluation. We also find existing LLM-based evaluations vulnerable to data contamination, due to their reliance on static templates. We train the fine-tuned model ('Normal') with MT-Bench input templates and GPT-4 outputs using only 80 samples and test it against MT-Bench and KIEval. Table 4 reveals that contamination training notably inflates the MT-Bench score by 1.79, a surge over 45% compared to the baseline, while ARC-Challenge accuracy remains stable and KIEval score slightly decreased.

Challenges in Contamination Detection. We evaluate the efficacy of current data contamination detection strategies, notably Skywork (Wei et al., 2023) and Min-K% Prob (Shi et al., 2023), which identify training data leakage through loss metrics as introduced in Related Work. We sampled 200 instances each from the trainset and testset of ARC-Challenge with contamination labels and tried to classify each instance with Min-K% Prob. We report AUC to measure its effectiveness. Table 3 demonstrates their capability to detect leaked data in the pre-training phase effectively, as the difference of average loss is significantly higher and Min-K% AUC reaches 0.89. However, both methods fail to identify contamination during SFT, with slight differences in loss values and Min-K% Prob AUC near random. We hypothesize that this may be due to the fine-tuning process only supervising the output sequence which is a short span containing the answer. This enables easy recall of the answer, without significantly impacting average loss values or Min-K% Prob values. *This discrepancy underscores the ineffectiveness of loss-based metrics in discerning data contamination during SFT phase.* Conversely, by leveraging KIEval, a correlation between KIEval scores and dataset accuracies emerges, suggesting the potential of KIEval in distinguishing between generalized knowledge application and mere data regurgitation for contamination detection.

---

[4]Training details including hyperparameters and hardware settings can be found in Appendix F. We also release the full training scripts on our GitHub repository for better reproducibility.
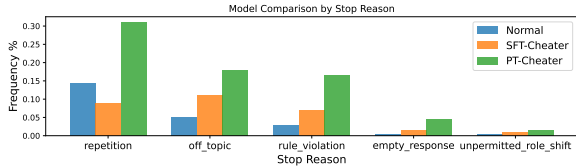
Figure 3: Statistics on reasons to trigger early stopping given by the evaluator model.

Table 5: Pearson (r), Spearman (ρ), Kendall-Tau (τ) correlation efficients of METEOR (Banerjee and Lavie, 2005), ROUGE (Lin, 2004), BERTScore (Zhang et al., 2019), and MT-Bench (Zheng et al., 2023) with human scores and their variance (σ^2). We introduce the compared baselines in Appendix A, ($\uparrow$) indicates higher values are better.

Metric	r ($\uparrow$)	ρ ($\uparrow$)	τ ($\uparrow$)	σ^2
METEOR	0.016	0.023	0.021	0.012
ROUGE-1	0.259	0.316	0.226	0.016
ROUGE-2	0.280	0.303	0.223	0.007
ROUGE-L	0.209	0.268	0.200	0.007
BERTScore	0.189	0.336	0.250	0.001
MT-Bench	0.520	0.494	0.405	9.360
Ours				
Accuracy	0.761	0.727	0.653	2.010
Logic	0.768	0.735	0.661	1.842
Relevance	0.633	0.643	0.543	1.152
Coherence	0.750	0.740	0.644	1.365
Conciseness	0.611	0.604	0.504	0.833
Overall	0.814	0.789	0.721	1.512

4.3 Meta-Evaluation of KIEval

Meta-evaluation is essential for validating the practical utility of any evaluation framework. In this section, we assess KIEval's alignment with human judgment and compare its performance against existing evaluation methods. Additionally, we conduct a cost analysis focusing on computational resources and API usage in Appendix E to validate KIEval's cost-effectiveness and scalability.

Human Evaluation: To ascertain KIEval's alignment with human preferences and its comparative effectiveness against prior methods, we collected a sample of multi-turn conversations generated by KIEval and compared the correlation between different evaluators and human-annotated scores. Specifically, we sampled 100 sets of conversations from all 5 datasets and across 7 candidate models and converted the multi-turn conversations to single-turn format to evaluate only one round of interaction to align with our compared baselines. Three human experts were asked to independently rate the responses of different models on a scale from 1 to 4, and we cover human annotation details in Appendix G. The Inter-Annotator

Agreement (IAA) was measured by averaging Cohen's Kappa coefficients for each annotator pair, yielding an average IAA of 0.624. This indicates substantial agreement among the annotators, a significant achievement for the complexity of the task at hand. The average score for each instance was then calculated and used as the human score for that response.

Following the meta-evaluation in G-Eval (Liu et al., 2023b), we computed Pearson, Spearman, and Kendall-Tau correlation coefficients to gauge the agreement between different evaluators' scores and human ratings. A detailed introduction of these evaluated baselines is provided in Appendix A due to page limitations.

As shown in Table 5, traditional reference-based evaluators align poorly with human judgments in open-ended conversations, due to their reliance on limited reference texts which cannot encompass the variety of valid responses. While the LLM-based evaluator MT-Bench shows commendable alignment with human preferences, its applicability is somewhat limited by its design, which is tailored to a predefined set of instructions and responses. In contrast, KIEval demonstrates a robust correlation with human preferences, underscoring its efficacy in evaluating dynamically generated, open-ended conversations.

Potential Bias: Employing LLMs for evaluation could introduce additional biases into the evaluation results. To mitigate bias and enhance objectivity, we have designed a separation between the 'interactor' and 'evaluator' roles. By utilizing different LLMs as evaluators with a fixed interactor LLM, we can assess the same conversation (since the interactor and candidate models are fixed) multiple times with different evaluators. Results in Appendix H indicate that while different evaluator LLMs may have varying preferences for model outputs, their overall results demonstrate a strong correlation.

Ablation Study: We further examine the effect of KIEval's main components through an ablation study, the experiments and results are presented in Appendix D.

5 Conclusion

KIEval provides a dynamic evaluation and analysis of LLMs across various domains, evaluating generative abilities and domain knowledge through structured conversations instead of relying on fixed

templates or instructions, reducing the risk of data contamination and enhancing the reliability of evaluations, while preserving alignment with human preference. Overall, our findings suggest several key insights:

- Static dataset-based benchmarks may not capture the full extent of performance disparities among LLMs, such datasets could potentially underestimate these differences.

- Training models on test splits of benchmark datasets primarily improves recall of answers rather than a genuine enhancement in knowledge comprehension or problem-solving abilities, underscoring the impact of data contamination.

- Detecting data contamination, particularly for the fine-tuning phase of LLMs, might be challenging for existing methods. We propose a paradigm shift from only detecting exposure to specific training texts towards evaluating the models' underlying rationale and depth of knowledge comprehension.

We believe that KIEval will serve as a valuable tool for researchers and practitioners alike, aiding in the development of more robust, versatile, and ethical AI systems.

6 Limitations

Our method, while insightful, operates under the assumption that LLMs can accurately evaluate the capabilities of less sophisticated models. However, the reliability of LLMs as universal evaluators is not without limitations, particularly when faced with complex benchmarks or assessing more advanced models. For certain evaluation tasks, such as mathematics problem-solving, coding, and fact-checking, depending solely on LLM evaluators may be insufficient. Furthermore, these evaluators may introduce additional biases into the assessment process. As these limitations can also be applicable to other current LLM-based evaluators, future research could explore a hybrid evaluation strategy that combines task-specific methods with LLM evaluators to achieve more nuanced and accurate assessments.

Another limitation concerns the scope of our work. Our focus is on evaluating instruction-tuned generative models with conversational abilities,

excluding those designed solely for natural language understanding (NLU) tasks without generative capabilities or base models lacking instruction-following capabilities. We can assess base models by instruction-tuning them using the exact same datasets and settings, operating under the hypothesis that employing identical data for training different models results in a fair comparison. Future research should delve more deeply into the evaluation of base models, scrutinizing the impact of instruction-tuning on their performance.

References

01.AI. 2023. Yi-6b model by 01-ai. `https://01.ai/`.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`.

Yoshua Bengio and Yann LeCun. 2007. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raf. 2023a. Emergent and predictable memorization in large language models. *arXiv preprint arXiv:2304.11158*.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023b. Pythia: A suite for analyzing large language models across training and scaling. *arXiv preprint arXiv:2304.01373*.

Rishi Bommasani, Percy Liang, and Tony Lee. 2023. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Luigi Daniele and Suphavadeeprasit. 2023. Amplify-instruct: Synthetically generated diverse multi-turn conversations for effecient llm training. *arXiv preprint arXiv:(comming soon)*.

Nolan Dey, Gurpreet Gosal, Hemant Khachane, William Marshall, Ribhu Pathria, Marvin Tom, Joel Hestness, et al. 2023. Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster. *arXiv preprint arXiv:2304.03208*.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Mengnan Du, Subhabrata Mukherjee, Yu Cheng, Milad Shokouhi, Xia Hu, and Ahmed Hassan. 2023. Robustness challenges in model distillation and pruning for natural language understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1758–1770.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.

Dom Eccleston. 2023. Sharegpt dataset. https://sharegpt.com/.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Varun Godbole, George E. Dahl, Justin Gilmer, Christopher J. Shallue, and Zachary Nado. 2023. Deep learning tuning playbook. Version 1.0.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*, volume 1. MIT Press.

Google. 2023. Bard.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, Deyi Xiong, et al. 2023. Evaluating large language models: A comprehensive survey. *arXiv preprint arXiv:2310.19736*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554.

Lynette Hirschman and Robert Gaizauskas. 2001. Natural language question answering: the view from here. *natural language engineering*, 7(4):275–300.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using mechanical turk to evaluate open-ended text generation. *arXiv preprint arXiv:2109.06835*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Ruosen Li, Teerth Patel, and Xinya Du. 2023a. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. AlpacaEval: An Automatic Evaluator of Instruction-following Models.

Yucheng Li. 2023. An open source data contamination report for llama series models. *arXiv preprint arXiv:2310.17589*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *arXiv preprint arXiv:2305.13711*.

Yachuan Liu, Liang Chen, Jindong Wang, Qiaozhu Mei, and Xing Xie. 2023a. Meta semantic template for evaluation of large language models. *arXiv preprint arXiv:2310.01448*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Carlo A Mallio, Andrea C Sertorio, Caterina Bernetti, and Bruno Beomonte Zobel. 2023. Large language models for structured reporting in radiology: performance of gpt-4, chatgpt-3.5, perplexity and bing. *La radiologia medica*, pages 1–5.

MosaicML. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for nlg. *arXiv preprint arXiv:1707.06875*.

OpenAI. 2023. Gpt-4 technical report.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B Hashimoto. 2023. Proving test set contamination in black box language models. *arXiv preprint arXiv:2310.17623*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Kaiping Peng, Richard E Nisbett, and Nancy YC Wong. 1997. Validity problems comparing values across cultures and possible solutions. *Psychological methods*, 2(4):329.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, and Yuxiong He. 2021. Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–14.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. *arXiv preprint arXiv:2310.18018*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Rylan Schaeffer. 2023. Pretraining on the test set is all you need. *arXiv preprint arXiv:2309.08632*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

Ekaterina Svikhnushina, Anastasiia Filippova, and Pearl Pu. 2022. iEval: Interactive evaluation framework for open-domain empathetic chatbots. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 419–431, Edinburgh, UK. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. 2022. *Natural language processing with transformers*. " O'Reilly Media, Inc.".

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023a. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023b. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. 2023c. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, et al. 2023. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al. 2020. Clue: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772.

Linyi Yang, Shuibai Zhang, Libo Qin, Yafu Li, Yidong Wang, Hanmeng Liu, Jindong Wang, Xing Xie, and Yue Zhang. 2022. Glue-x: Evaluating natural language understanding models from an out-of-distribution generalization perspective. *arXiv preprint arXiv:2211.08073*.

Linyi Yang, Shuibai Zhang, Zhuohao Yu, Guangsheng Bao, Yidong Wang, Jindong Wang, Ruochen Xu, Wei Ye, Xing Xie, Weizhu Chen, et al. 2023. Supervised knowledge makes large language models better in-context learners. *arXiv preprint arXiv:2312.15918*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2023. Evaluating large language models at evaluating instruction following. *arXiv preprint arXiv:2310.07641*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2023. Dyval: Graph-informed dynamic evaluation of large language models. *arXiv preprint arXiv:2309.17167*.

A Baseline Evaluators

In our experimental framework, we compare KIEval with prevalent evaluators in open-ended dialogue evaluation, following Liu et al. (2023b), alongside MT-Bench, which epitomizes the application of Large Language Models (LLMs) in evaluation processes. To compare reference-based methods with reference-free approaches and our method, we use GPT-4 to generate references.

- **METEOR** (Banerjee and Lavie, 2005), a reference-based evaluation metric, utilizes unigram matching between generated outputs and reference texts crafted by humans to assess performance across a variety of Natural Language Generation (NLG) tasks, including machine translation and dialogue generation.

- **ROUGE** (Lin, 2004) comprises a suite of metrics for reference-based evaluation, facilitating the comparison of automatically generated summaries or translations against one or more human-crafted reference summaries or translations.

- **BERTScore** (Zhang et al., 2019), another reference-based evaluation metric, employs contextual embeddings from BERT to measure cosine similarity between words in candidate and reference sentences. Demonstrated to align well with human judgment at both the sentence and system levels, BERTScore calculates precision, recall, and F1 scores, offering valuable insights for various NLG tasks.

- **MT-Bench** (Zheng et al., 2023), a LLM-based, reference-free evaluation approach, harnesses cutting-edge LLMs to assess model outputs. It features a series of open-ended questions designed to test a model's capabilities in engaging in conversation and instruction-following abilities. As MT-Bench is similar to AlpacaEval (Li et al., 2023b), PandaLM (Wang et al., 2023c) and G-Eval (Liu et al., 2023b) but being a more popular option, we select MT-Bench without compromising on the breadth of our evaluation. In our meta-evaluation experiment, we use `gpt-4-1106-preview` as the evaluator and use the single-answer grading mode of MT-Bench.

B Datasets

We use the following datasets in our experiments, for statistics and used splits, please refer to Table 6.

ARC-Easy and ARC-Challenge (Clark et al., 2018): Both are subsets of the AI2 Reasoning Challenge, a benchmark for assessing a model's reasoning and understanding in science questions. ARC-Easy contains simpler questions, while ARC-Challenge includes more complex ones.

HellaSwag (Zellers et al., 2019): challenges models to complete realistic scenarios in text, testing common sense and predictive abilities.

MMLU (Hendrycks et al., 2020): A comprehensive English examination composed of multiple-choice questions encompassing a wide array of disciplines. This extensive test includes subjects ranging from humanities and social sciences to hard sciences, alongside other essential areas of knowledge. It encompasses 57 distinct tasks, covering fields such as elementary mathematics, US history, computer science, law, and beyond.

C-Eval (Huang et al., 2023): A comprehensive Chinese evaluation composed of 13948 multi-choice questions spanning 52 diverse disciplines and four difficulty levels.

C Correlation Analysis of KIEval and Dataset Benchmarks

To further investigate the correlation between dataset-based benchmarks and KIEval, we use regression analysis as shown in Figure 4. We also leverage the Pearson correlation coefficient to provide quantitive analysis in Table 7. The results revealed a significant positive correlation between KIEval scores and dataset-based benchmark accuracies. This correlation underscores KIEval's alignment with traditional evaluation methods. However, we also bring new insights that traditional benchmarks do not offer: *while dataset-based benchmarks effectively assess LLM knowledge under contamination-free conditions, their results are easily inflated in the presence of data contamination. In contrast, KIEval exhibits a lower susceptibility to these issues.* Visual analysis offers additional perspective by contrasting model performances as per benchmark accuracies and KIEval scores. Models significantly above the regression line suggest capabilities beyond those captured by traditional benchmarks. In this scenario, traditional benchmarks are not sufficiently challenging to effectively differentiate the stronger models from others, nor do they accurately represent the generative capabilities of these models. It is evident that GPT-3.5 is included in this category. Conversely, models

Table 6: Details of datasets in our experiments. We report 5-shot accuracy metric of 'Used Splits' split for each dataset.

Datasets	Splits	Used Splits	Split Size	Language
ARC-Challenge	train, validation, test	test	1.17k	English
ARC-Easy	train, validation, test	test	2.38k	English
Hellaswag	train, validation,test	validation	10k	English
MMLU	auxiliary_train, test, validation, dev	test	14k	English
C-Eval	val, test, dev	val	1.35k	Chinese

Table 7: Pearson correlation coefficient of KIEval scores and dataset accuracy scores. Due to suspected data contamination in Yi-6B, we report two sets of results with and without Yi.

PCC	r	p	r Excl. Yi	p Excl. Yi
Overall	0.664	1.37E-05	0.765	8.67E-07
ARC-E	0.892	6.97E-03	0.934	6.45E-03
ARC-C	0.839	1.83E-02	0.940	5.29E-03
MMLU	0.814	2.57E-02	0.876	2.21E-02
HellaSwag	0.686	8.85E-02	0.862	2.74E-02
C-Eval	0.427	3.40E-01	0.924	8.42E-03

Table 8: Ablation study of KIEval's components. DW and ES denote decaying weight in KIEval score and early stopping mechanism. We report Pearson (r), Spearman (ρ), Kendall-Tau (τ) correlation efficients with human scores and their variance (σ^2). ($\uparrow$) indicates higher values are better.

Methods		r ($\uparrow$)	ρ ($\uparrow$)	τ ($\uparrow$)	σ^2
DW	ES				
✓	✓	0.768	0.759	0.625	1.718
✗	✓	0.730	0.721	0.588	1.708
✓	✗	0.737	0.751	0.623	1.839
✗	✗	0.691	0.715	0.588	1.870

Figure 4: Scatter plots of KIEval scores and traditional benchmark scores by model and dataset. Each point represents the performance of a model on a specific dataset, measured by the KIEval score and accuracy score (5-shot). Regression lines are plotted for each dataset. *Points significantly above the regression line indicate the performance gap not captured by traditional benchmarks but captured by KIEval, while points significantly below the regression line indicate potential data contamination in traditional benchmarks.*

falling below the regression line, exhibiting high benchmark accuracy but low conversation quality, suggest limited real-world applicability, potentially indicative of data contamination. Interestingly, the visualization shows that not only does our simulated SFT Cheater model fall into the outlier category below the regression line, but Yi-6B also exhibits similar behavior.

D Ablation Study of KIEval Components

This study assesses the impact of the decaying weight scoring and the early stopping mechanism of KIEval through an ablation analysis. Employing the same set of KIEval-generated conversations used in our meta-evaluation, we explore four distinct configurations of the KIEval framework. Specifically, we investigate the influence of the weighted scoring by replacing the decaying weight with a constant value, effectively equating the multi-round score to the mean of single-turn scores. Additionally, we examine the consequences of omitting the early stopping mechanism, thereby allowing conversations to proceed unabated until their conclusion. We then compare the correlation coefficients between these variants and human scores. As indicated by the data in Table 8, the exclusion of either feature results in a notable decline in performance, underscoring their respective contributions to the model's efficacy.

Table 9: API usage estimation for KIEval and pairwise-comparison based evaluation methods. Priced in USD, according to openai's GPT-4 pricing policy.

Method	1 Model	10 Models	100 Models
KIEval	27	279	2,796
Pairwise	16	720	79,200

Table 10: Average GPU budget for a single model evaluated on one dataset with KIEval. We report results for LLaMA2 models with varying parameter sizes.

	7B	13B	70B
GPU Hours	0.74	0.99	9.38

E Cost and Scalability

Assessing KIEval's scalability requires a thorough evaluation of overall costs. Our method employs a strong LLM accessed via API, with expenses based on input and output token lengths. Table 11 details the average token count per model evaluation across diverse datasets. Additionally, the average GPU expenditure for single model evaluations on NVIDIA A100 GPUs is provided in Table 10. Financially, deploying GPT-4 in both interactor and evaluator roles within KIEval incurs a cost of around 27 USD for each model evaluation, comprising 1000 interaction rounds. Importantly, due to our adoption of single-answer grading over pairwise comparison (Wang et al., 2023c; Zheng et al., 2023), costs increase linearly rather than quadratically with the number of models evaluated. For a comprehensive understanding of the cost implications at scale, we present a detailed estimation in Table 9.

F Experiment Details

In this section, we detail the experimental setup employed to facilitate the reproduction of our results. The entire codebase used in our experiments

Table 11: Average number of tokens consumed of evaluation on a single model across various datasets, over 200 samples with 5 rounds of interaction.

	Interactor		Evaluator	
	Prompt	Completion	Prompt	Completion
Avg.	557k	28k	1546k	203k
ARC-E	554k	28k	1592k	208k
ARC-C	540k	27k	1553k	205k
MMLU	656k	30k	1731k	213k
HellaSwag	527k	29k	1488k	198k
C-Eval	505k	26k	1365k	189k

Table 12: We report Cohen's Kappa (κ) for each pair of annotators to measure Inter-Annotator Agreement (IAA) between annotators.

Annotator Pair	κ
A + B	0.650
A + C	0.580
B + C	0.642

has been made publicly available, ensuring transparency and ease of verification for our findings.

Codebase and Dependencies: Our experiments are done with LLaMA-Factory package[5], a framework designed to streamline the training of large language models. We use Huggingface Transformers (Wolf et al., 2019) library and the Deepspeed (Rasley et al., 2020) Zero-3 optimizer (Rajbhandari et al., 2021), forms the backbone of our computational experiments.

Training Configuration: For the training process, we have configured the learning rate to 2e-5, employing a cosine learning rate scheduler. Our hardware setup consists of 4 NVIDIA A100 GPUs, and we've set per-device batch size to 1, coupled with a gradient accumulation step of 4. We use full-parameter training for 4 epochs in all our experiments, including training models with data contamination during pre-training and fine-tuning.

G Human Annotation Details

For human annotation in our work, all annotators are authors of this paper who previously have not accessed the outputs of models in our experiments and volunteer to contribute. All three annotators agree on how the data would be used. Since the data to be annotated come from open-source datasets and popular LLMs, ethical concerns are not applicable. We provide guides for each annotator and for each annotator, we give them a unique URL to our annotation platform built with Gradio, and give them instructions: 'You are given some conversations between a candidate model and a interactor model. Please score the response of the candidate model with integers from 1 to 4, following our scoring guide. Your score should be definitive, and consider the response's factual accuracy, logical structure, language conciseness, and coherence.'

We measure the agreement of our annotators

---

[5]Available at https://github.com/hiyouga/LLaMA-Factory/.

Table 13: KIEval scores using Claude 3 Opus (`claude-3-opus-20240229`) and GPT-4 Turbo (`gpt-4-turbo-preview-1106`) as evaluators on ARC-Challenge dataset.

Model	Evaluator	Accuracy	Logic	Relevance	Coherence	Conciseness	Overall
GPT-3.5	GPT-4	94.6	94.7	98.5	96.1	97.3	95.5
	Claude-3	98.6	98.8	99.8	99.4	99.0	98.7
LLaMA-2 70B	GPT-4	81.9	82.8	92.2	85.3	75.6	84.1
	Claude-3	98.3	98.7	98.2	96.9	84.6	96.4
LLaMA-2 7B	GPT-4	70.6	71.6	90.4	77.9	71.7	74.4
	Claude-3	90.9	91.8	98.0	95.0	85.2	91.0

Table 14: Pearson (r), Spearman (ρ), Kendall-Tau (τ) correlation efficients of KIEval scores evaluated by `claude-3-opus-20240229` and `gpt-4-turbo-preview-1106`.

Metric	Corr. Coeff.	P-Value
Pearson r	0.822	2.87e-05
Spearman ρ	0.898	4.17e-07
Kendall τ	0.761	1.10e-05

by calculating Cohen's Kappa as Inter-Annotator Agreement, results can be found in Table 12. We reach an average IAA for all pairs of human annotators of 0.624, indicating a substantial agreement among our annotators.

H Potential Bias

While KIEval provides a new evaluation method, the reliance on strong LLMs as evaluators could inadvertently propagate existing biases. To study the bias introduced by the evaluator LLMs, we utilize different LLMs as evaluators with a fixed interactor LLM. Specifically, we use `gpt-4-turbo-preview-1106` and `claude-3-opus-20240229` as evaluators, with the same prompts and sampling hyperparameters. We report KIEval scores of `gpt-3.5-turbo`, `llama-2-70b-chat-hf` and `llama-2-7b-chat-hf` from different evaluator LLMs on ARC-Challenge dataset in Table 13. We also report the correlation coefficients of the results in Table 14.

These results indicate that although GPT-4 and Claude 3 have different preference of models, their overall results exhibit a strong correlation. Note that we use the exact same prompt for GPT-4 and Claude 3, and as Claude 3 works differently with their system prompts, the scores from Claude 3 are generally higher but this does not affect the effectiveness of our experiments.

I Use of AI Assistants

In this work, we use GitHub Copilot to assist coding, and GPT-4 to correct grammatical errors.

J Complete Experiment Results

We share the complete experiment results from all 5 datasets with 7 models, evaluated with KIEval and benchmark accuracies in Table 15, 16, 17, 18, 19.

K Complete Prompt

The system prompts for interactor, candidate and evaluator models are given in Figure 5.

Table 15: KIEval Results on ARC-Easy.

ARC-E	Accuracy	Logic	Relevance	Coherence	Conciseness	Overall	Rounds	Acc. (5-shot)
GPT-3.5	97.1	97.4	99.3	97.9	97.9	97.6	4.97	92.7
LLaMA2 70B	90.3	90.3	94.6	91.3	79.6	90.7	4.85	92.3
LLaMA2 13B	84.5	84.3	93.2	87.7	85.8	86.2	4.70	81.9
LLaMA2 7B	77.1	77.4	89.7	82.2	73.6	78.9	4.49	73.6
Mistral 7B	78.5	78.2	91.4	83.5	79.9	80.8	4.64	83.5
Yi 6B	83.4	83.6	90.9	85.8	76.4	83.8	4.58	90.7
MPT 7B	63.9	64.1	84.9	71.5	81.8	68.4	4.34	53.3

Table 16: KIEval Results on ARC-Challenge.

ARC-C	Accuracy	Logic	Relevance	Coherence	Conciseness	Overall	Rounds	Acc. (5-shot)
GPT-3.5	94.6	94.7	98.5	96.1	97.3	95.5	4.94	82.3
LLaMA2 70B	81.9	82.8	92.2	85.3	75.6	84.1	4.66	80.4
LLaMA2 13B	75.4	75.9	91.3	82.3	82.6	78.6	4.56	65.7
LLaMA2 7B	70.6	71.6	90.4	77.9	71.7	74.4	4.44	55.7
Mistral 7B	75.9	75.8	90.0	81.4	79.1	78.5	4.46	67.5
Yi 6B	75.6	76.1	85.0	79.6	71.2	76.8	4.33	79.0
MPT 7B	60.2	61.4	83.6	69.5	81.1	65.5	4.33	43.4

Table 17: Summary of KIEval Results on MMLU

MMLU	Accuracy	Logic	Relevance	Coherence	Conciseness	Overall	Rounds	Acc(5-shot)
GPT-3.5	95.5	95.8	98.3	96.7	97.4	96.2	4.95	58.2
LLaMA2 70B	89.0	90.3	93.7	90.3	76.0	89.6	4.80	61.8
LLaMA2 13B	85.8	87.0	93.9	88.6	81.4	87.4	4.76	52.1
LLaMA2 7B	82.2	83.6	91.9	84.7	70.4	83.0	4.61	44.5
Mistral 7B	81.6	82.8	90.5	85.3	77.5	83.0	4.62	52.7
Yi 6B	84.7	86.5	91.8	87.4	76.5	86.5	4.58	61.9
MPT 7B	70.6	72.0	86.6	77.9	83.0	74.7	4.46	33.9

Table 18: KIEval Results on HellaSwag.

HellaSwag	Accuracy	Logic	Relevance	Coherence	Conciseness	Overall	Rounds	Acc. (5-shot)
GPT-3.5	85.6	85.6	93.9	90.1	93.1	88.2	4.82	76.6
LLaMA2 70B	76.6	79.5	88.2	82.0	78.9	80.1	4.41	74.4
LLaMA2 13B	72.6	75.9	88.7	83.0	85.2	78.5	4.66	59.3
LLaMA2 7B	70.8	73.3	87.3	79.9	80.2	76.4	4.54	39.8
Mistral 7B	65.6	67.1	83.8	75.6	75.2	70.3	4.34	54.4
Yi 6B	64.4	67.0	79.9	74.3	72.4	68.7	4.20	73.7
MPT 7B	50.0	51.7	74.3	62.5	74.4	57.3	4.10	27.3

Table 19: KIEval Results on C-Eval

C-Eval	Accuracy	Logic	Relevance	Coherence	Conciseness	Overall	Rounds	Acc. (5-shot)
GPT-3.5	79.8	80.6	94.7	87.3	92.0	83.3	4.72	50.8
LLaMA2 70B	57.6	58.3	80.1	66.5	64.1	61.0	3.94	42.0
LLaMA2 13B	48.4	49.8	79.3	61.5	62.9	54.4	3.74	37.8
LLaMA2 7B	44.9	45.1	73.8	55.8	55.9	49.3	3.62	33.4
Mistral 7B	47.3	47.8	73.3	58.0	59.5	52.2	3.61	39.3
Yi 6B	53.1	54.1	73.0	59.3	55.9	55.6	3.66	71.5
MPT 7B	39.5	40.2	72.7	51.5	64.0	44.9	3.52	26.2

Candidate:
You are a helpful assistant that follows the instructions of the user. You must output in less than 200 words and your responses should be concise and accurate.

Interactor:
As an expert interviewer, initiate a reflective dialogue by asking the user to describe the rationale behind their answer to the multiple-choice question. You must output in less than 200 words. Encourage them to be succinct and precise. Guide the conversation to uncover and address any misconceptions, prompting the user to critically evaluate their thinking process. Their answer is likely to be incorrect. Avoid prompting the user to merely explain their choice, this is very important. Instead, focus on probing questions that challenge their assumptions and lead them to a deeper understanding of the topic. The outputs from the user is very likely to be wrong, do not validate misinformation or incorrect logic; instead, use it as a point of assessment. Remember, you are not educating the user, be neutral and do not give comments since you are the interviewer instead of the evaluator, this is very important. Do not ask too many questions before the user responds. Always skip the greetings, or any other prompting words, ask questions directly. DO NOT say anything else.

Evaluator:
You are an objective evaluator in an interview. Your task is to evaluate a candidate's performance during a series of interactions with an interactor, both parties are AI models. The candidate is asked to solve a problem, and the interactor will ask questions accordingly. The conversation alternates between the interactor (marked with [BEGIN INTERACTOR]) and the candidate (marked with [BEGIN CANDIDATE]). After each candidate response, evaluate the candidate's performance in that round of interaction as well as in context, based on the following aspects independently, rating each on a scale from 1 (Poor) to 4 (Good):
Accuracy: Truthfulness and factual correctness of the candidate's response.
Logic: Logical structure and soundness of reasoning. Whether conclusions are well-supported and arguments are free from logical fallacies.
Relevance: How the response relates to the topic. Ensure responses are within the scope of the "assistant" role, avoiding unpermitted role shifts.
Coherence: How well the response integrates into the context. Consistency with previous statements and overall conversational flow.
Conciseness: Brevity and clarity of the response. Clear, to-the-point communication, free from extraneous elaboration or repetitive words.

Scoring Guide:
1 (Poor): Significant deficiencies or inaccuracies in the aspect.
2 (Below Average): Noticeable weaknesses, partially on target but lacking in several areas.
3 (Above Average): Solid and competent, mostly on target with only a few minor shortcomings.
4 (Good): Strong performance, fully meets and often surpasses expectations.

Evaluation Rules:
1. Evaluate each response consistently and objectively without bias, strictly adhering to scoring guide.
2. Score from 1 to 4 for each aspect independently, using only integers. Low score in one aspect should not influence another aspect. Write a brief comment before scoring in the JSON output structure.
3. Write a overall comment and then give an overall score (same scoring guide). The overall comment should be brief and clear. Consider the performance throughout the interaction, not just in the latest round.
4. Use of `stop_conversation` Flag: Set this flag to true for early conversation termination in cases listed below. If `stop_conversation` is true, choose a `stop_reason` from ["off_topic", "empty_response", "rule_violation", "repetition", "unpermitted_role_shift"], otherwise `stop_reason` is `none`.
5. Format of Evaluation: Output in JSON format strictly following the template:{"accuracy": {"comment": "", "score": 0}, "logic": {"comment": "", "score": 0}, "relevance": {"comment": "", "score": 0}, "coherence": {"comment": "", "score": 0}, "conciseness": {"comment": "", "score": 0}, "overall_comment": "", "overall_score": 0, "stop_conversation": false, "stop_reason": "none"}
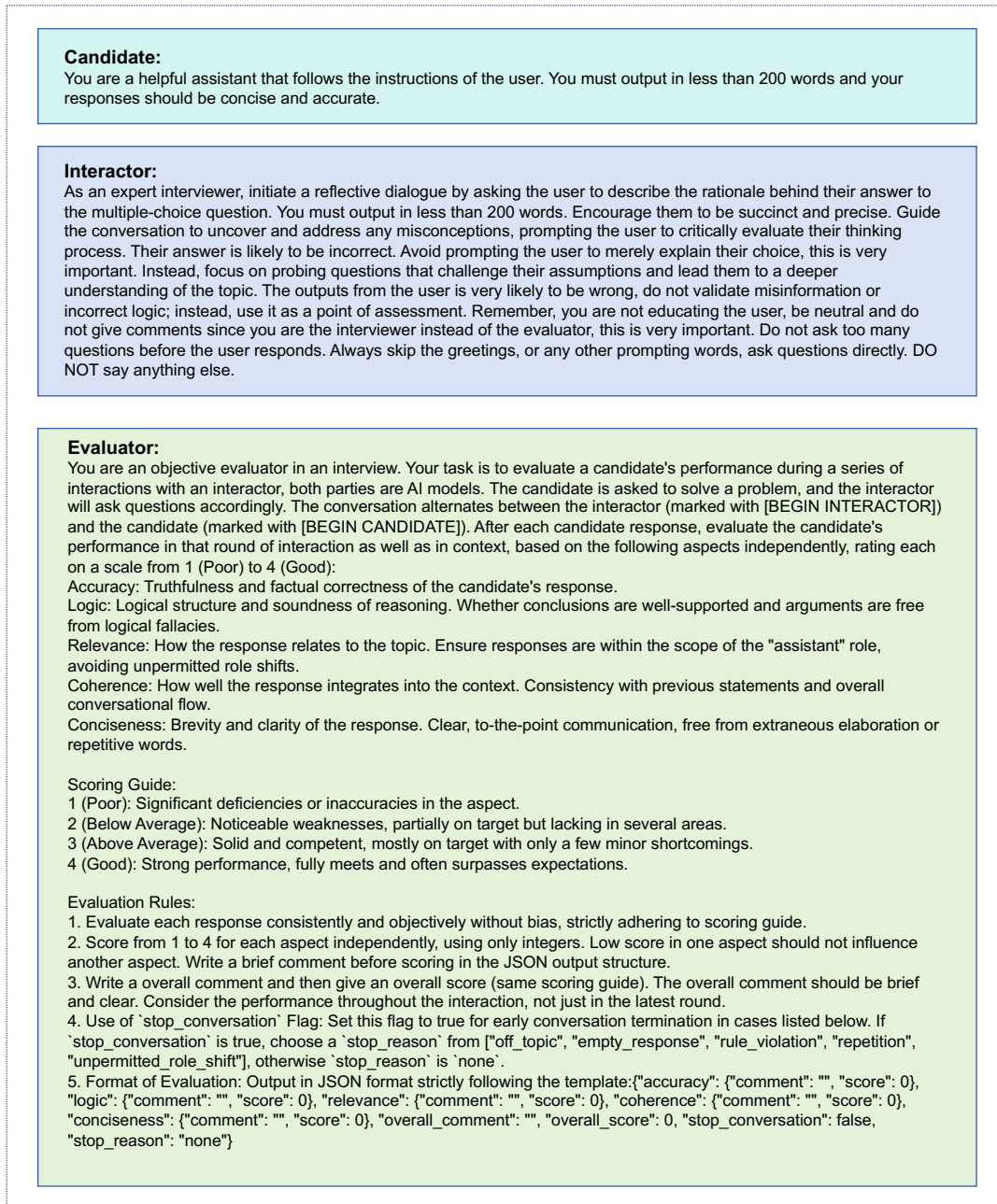
Figure 5: The full system prompt for interactor, candidate and evaluator models.