Quadruple-Slit Experiment: Reliability Issues in Multiple-Choice Evaluation for Language Models

Anonymous ACL submission

Abstract

Multiple-choice evaluation has been commonly used for assessing language model capabilities. Current evaluation methods primarily employ a probability comparison approach. However, our study demonstrates overlooked reliability issues with this approach. The deterministic prediction comes at the cost of sacrificing core properties of multiple-choice questions—–order invariance, position independence and length independence. To perform reliability checking, we propose a test consistency checking method inspired by the double-slit experiment. Experiments across multiple LLMs and benchmarks reveal the shaky reliability of current implementations, uncovering severe position and length biases unintentionally introduced by these evaluation methods.

1 Introduction

Recent advances in artificial intelligence have been driven by the development of Large Language Models (LLMs). With expanding abilities to tackle a wide range of tasks, evaluating their capabilities becomes increasingly important. Researchers have made sustained efforts to construct comprehensive settings for evaluating LLMs. However, in examining one of the most straightforward and prevalent evaluation settings—multiple-choice evaluation—we uncover intrinsic reliability issues that have been overlooked in current implementations.

Multiple-choice question has become an important setting for assessing large language models due to its distinct structure. This structure presents models with a query and a constrained set of candidate choices, with one designated as correct. The specificity enables straightforward and grounded evaluation, allowing targeted assessment of model capabilities. For instance, the Open LLM Leaderboard (Beeching et al., 2023), a popular benchmark for evaluating LLMs, utilizes the multiple-choice format for 3 of its 4 evaluation tasks. LLama 2 (Touvron et al., 2023b), the successor model to LLama (Touvron et al., 2023a), evaluates its capabilities across 19 academic benchmarks, with 9 being multiple-choice settings, covering evaluation on language understanding, commonsense reasoning, and world knowledge.

However, implementing multiple-choice evaluation is not as straightforward as it may seem. Although LLMs can generate responses to queries, automatically evaluating these responses remains challenging. This requires either specially-designed prompts to elicit certain response forms (Zhang et al., 2023), or the utilization of robust language understanding tools to verify if responses match the choices (OpenAI, 2023). Both of these issues can affect the precision, stability, and consistency of the evaluation process.

Recent work has applied a two-step probability comparison approach for automatic multiple-choice evaluation, aided by predetermined choices. This first **adapts** the multiple-choice question into an evaluable format, then compares choice probabilities using **scoring** methods. While enabling definitive and automatic evaluation, the reliability of such methods has largely been overlooked. A recent study found high variability in results, with accuracy ranging from 30% to nearly 60% depending on the adaptation used (Liang et al., 2022). Given that numerous LLMs have been evaluated using probability comparison methods, the uncertainty around reliability underscores the core motivation of this work: *the need to validate the reliability of these methods under multiple-choice evaluation.*

When delving into the implementation details of these methods, we uncover three inherent issues that adversely impact the nature properties of multiple-choice questions:

1. **Order Invariance**: choices should be permuted randomly without altering the question itself. However, adaptation process uninten-

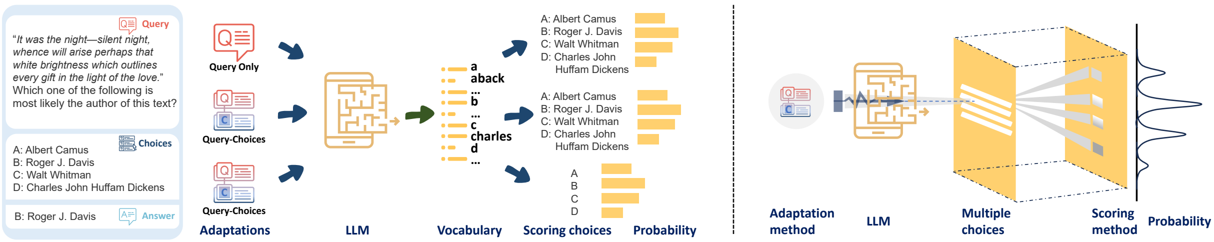

Figure 1: Illustration of the evaluation implementation and test consistency checking method for multiple-choice evaluation. Considering the evaluation input and scoring choice forms, three different adaptations are commonly used in probability comparison methods (**left**). Our objective is to uncover intrinsic reliability issues in these implementations. To achieve this, we propose test consistency checking method inspired by the famous double-slit experiment (**right**). This method treats each multiple-question evaluation as multiple trials, allowing us to bring out order invariance while revealing reliability issues related to position and length independence.

tionally disrupts the invariant property as it imposes an artificial order on the choices.

2. **Position Independence**: choices are elements without inherent positional properties. Here, positional biases are introduced when concatenating choices and possible answers.

3. **Length Independence**: a fair evaluation should avoid bias towards longer or shorter choices. We find that probability scoring methods introduce severe length bias, creating a dilemma where tendencies for both longer and shorter choices simultaneously hold true.

To perform reliability evaluation, inspired by the famous double-slit experiment in physics, we propose test consistency checking. By randomizing choice order across trials, this method enables consistency checks while preserving the fundamental order invariance. In experiments, we compare seven implementations, consisting of combinations of three adaptation methods and three probability scoring methods. We test both pre-trained and fine-tuned models on six multi-subject multiple-choice benchmarks. The results of our experiments reveal that all current probability comparison implementations suffer from inherent reliability issues.

This paper makes three contributions: (1) a systematic and focused study of multiple-choice evaluation; (2) an exploration of reliability issues in currently prevalent probability comparison methods; and (3) extensive comparison experiments that attempt to reveal underlying groundlessness in these evaluation methods. Additionally, through exploring this seemingly straightforward evaluation, we aim to spur rethinking the study of evaluation overall, as a fundamental discipline in AI development.

2 Background

2.1 Multiple-Choice Evaluation

Multiple-choice evaluation is a constrained evaluation setting where models are tested with multiple-choice questions. These questions have three key components. (1) The query: This provides context or poses a question for the model to consider. (2) The choices: Each candidate choice has a label (e.g, A, B, C) and a description that proposes a possible response to the query. (3)The answer: Only one choice is designated as the correct answer choice based on the query.

Multiple-choice evaluation constrains the output space with predetermined choices, allowing for targeted assessment of a model's abilities across domains (Zhong et al., 2023; Kung et al., 2023; Gao et al., 2021; Zhang et al., 2023). Currently, it has been used to assess in diverse fields such as mathematics, chemistry, medicine and humanities, spanning tests for safety (Lin et al., 2021), question answering (Clark et al., 2018), commonsense reasoning (Zellers et al., 2019), and multi-subject knowledge (Huang et al., 2023; Zeng, 2023).

2.2 Probability Comparison Methods

The constrained nature of multiple-choice evaluation enables more deterministic and automatic evaluation by comparing probability scores between choices, unlike open-ended evaluations where the LLM generates free-form responses. This type of methods generally involve two steps: first, adapting the multiple-choice question to an evaluable format; and second, calculating probability scores for each choice. By comparing these scores, a model can qualitatively predict which choice is more or less likely to be the correct answer.

Adaptation To evaluate language models using multiple-choice questions, adaptation methods have been applied to format the query and candidate choices in a way that allows the model to score each choice. These existing adaptation methods generally fall into three main categories.

- *joint-label*: This method concatenates all the choices together with the query to form an extended query. This extended query is fed to the model. The probability assigned to each label is then used to generate a score.
- *joint-desc* method: This method concatenates all the choices with the query first. Given the entire extended query into the language model at once, it uses the probability the model assigns to each choice, consisting of both the label and description, to generate a score.
- *separate* method: This method evaluates each choice individually by feeding only the original query into the language model. It then calculates the probability of each choice at a time to generate a score.

These adaptation methods have been applied in various works for multiple-choice evaluation. For example, the technical report of GPT-3 (Brown et al., 2020) indicates using the *separate* method in their evaluations. The Open LLM Leaderboard (Beeching et al., 2023) applies the *joint-desc* method by default when accessing models. Some evaluation frameworks utilize different methods depending on the benchmark. For instance, HELM (Liang et al., 2022) employs the *separate* method for the HellaSwag benchmark (Zellers et al., 2019) but uses the *joint-label* method for the MMLU benchmark (Hendrycks et al., 2021) instead.

Probability Scoring Probability scoring involves calculating probability scores for possible continuations (e.g, possible answer choices) given a query prompt (e.g, extended query). However, scoring for entire possible continuations poses challenges for language models, which only generate probabilities token-by-token (i.e, $P(x_i|x_{0:i})$) rather than for complete sequences. Given $x_{0:m}$ as the prompt and $x_{m:n}$ as a possible continuation to be scored, where m is the index of the first token in the continuation with a token length of $n - m$, previous work has developed several normalization methods to handle this issue of scoring (Gao, 2021).

- Unnormalized method: A simple approach is to calculate the score of a continuation $x_{m:n}$ by summing the log likelihood of each token given the previous prompt. The formula is $\Sigma_{i=m}^{n-1} logP(x_i|x_{0:i})$, where higher scores indicate higher probability of being correct. However, this could introduce a length bias issue, as longer continuations typically have lower log likelihood, leading to a preference on shorter choices during evaluation.
- Token-length normalized method: The score of a continuation is calculated by taking the average log likelihood per *token* given the prompt, using the formula $\frac{1}{n-m}\Sigma_{i=m}^{n-1} logP(x_i|x_{0:i})$. This aims to normalize the score by the number of tokens. It is worth noting that the number of tokens is determined by the tokenizer used.
- Character-length normalized method: This method calculates the score by taking the average log likelihood per *character* given the prompt, using the formula $\frac{1}{L(x_{m:n})}\Sigma_{i=m}^{n-1} logP(x_i|x_{0:i})$ where $L(x_{m:n})$ is the number of characters in $x_{m:n}$. Using character length for normalization eliminates the impact of different tokenizers tokenizing the same text into varying length.

3 Reliability Issues

For multiple-choice evaluation, the prevailing methods primarily rely on the probability comparison approach, which consists of two key steps: an adaptation method and a probability scoring method. While numerous large language models have been evaluated on diverse multiple-choice questions using probability comparison methods (Liang et al., 2022; Beeching et al., 2023), the reliability of these methods has been largely overlooked. To address this gap, we closely examine the implementation of these methods, and our analysis reveals that there are three inherent reliability issues involved with these implementations.

Order Invariance What makes multiple-choice questions special? The pre-determined candidate choices. These choices constrain the output space, providing a set of options to select from. This constrained output space, represented abstractly as a finite and discrete set of choice elements, is the key differentiating factor that distinguishes multiple-choice questions from other types of evaluation settings. This makes multiple-choice questions intrinsically order invariant—the choices can be permuted without changing the nature of the question.

Current implementations of probability comparison methods adversely impact the order invariance. First and foremost, adaptation methods convert the representation of the choice set for language model evaluation. Specifically, these adaptation methods (e.g, *joint-desc* method) represent the choice set as ordered sequential text—a human-readable but ordered format. Through this process, order invariance is sacrificed unintentionally for human-friendly and controllable evaluation purposes. Secondly, current large language models, typically causal language models based on the Transformer architecture (Vaswani et al., 2017), fundamentally lack order invariance. Instead, one of the core design of the Transformer is the use of position embeddings to encode order information in text.

Position Independence Position independence is related to order invariance, but centers on the answer side more than the query. For a multiple-choice question, the answer is a selected choice from the candidate choice set. Position independence means that choices do not possess positional properties. In other words, there is an intrinsic position independence–there should be no positional bias when predicting the answer choice.

However, current implementations fail to achieve true position independence. First, probability scoring methods require concatenating each possible answer choice with the extended query for scoring. This concatenation establishes an implicit relation between the answer choice and candidate choices in the query, breaking position independence even if the choice order is randomized. Secondly, the self-attention mechanism in current language models also contributes to the destruction of position independence. It enables attention between the possible answer choice and other choices in the extended query, reminding the model of the unwanted existence of different positions when scoring based on causal language modeling.

Length Independence Length is an attribute attached to the text of choices. A fair evaluation implementation should not be impacted by the length of choices. In the abstract representation of such questions, the choices in the set are elements without an inherent length attribute. This marks an inherent length independence—evaluation results should not be biased by the length factor of choices.

However, current methods also fail to truly achieve length independence. This core issue stems from the core definition of language modeling. On one hand, longer text generally have lower log probabilities. Notably, this issue has been empirically observed by researchers. Prior work has proposed normalization methods to mitigate this bias by averaging the log likelihoods per token or character. However, averaged log probability increases as length grows. This leads to a dilemma where both “the longer, the more likely" and “the shorter, the more likely" can hold true when making selection.

4 Test Consistency

To test the reliability of these methods, a straightforward approach is to record prediction results across multiple trials and analyze their consistency. However, because probability comparison involves deterministic calculations, the prediction results will remain identical across trials.

To achieve this straightforward method of evaluation, our first goal is to bring out order invariance. We propose a simple “test consistency checking” method that evaluates a multiple-choice question multiple times, introducing randomness by varying the choice order across trials. Leveraging order invariance makes the evaluation method more reliable. Our experiments in the next section also demonstrate its effectiveness at revealing the reliability issues we aim to identify.

Inspiration We propose test consistency checking inspired by the famous double-slit experiment (Young, 1803; Green, 2005) in quantum physics. This classic physics experiment sends individual photons one at a time towards two parallel slits. Researchers then observe the resulting pattern on a detection screen. Surprisingly, the results show quantum particles can take both paths simultaneously from the source to the screen, producing an interference pattern on the screen. However, if detectors identify which slit each photon passes through first, the pattern will match the slit shape (Feynman et al., 1965). This reveals that light exhibits both wave and particle properties.

In our proposed method, a multiple-choice query acts like a single photon. When sent to a large language model which generates arbitrary continuations, the query itself undergoes “self-interference”. The multiple choices are analogous to slits that the photon (query) can pass through, while probability comparison methods are like detectors tracking which "path" the query takes. Just as photons can take multiple paths but collapse to one after the measurement, a query may be consistent with multiple choices essentially, yet once the probability comparison method is applied, the query becomes consistent with only one choice. By analyzing the predictions across trials, we can check consistency, similar to observing patterns of photons.

Test Consistency Checking We propose test consistency checking to probe the reliability of different probability comparison methods. Specifically, we keep the original query unchanged while randomly shuffle the order of choices for each trial. We will record the evaluation results (e.g, predicted choices) of these trials under different methods. From the perspective of multiple-choice evaluation, the core idea is that a reliable evaluation method should predict the same choice regardless of how the choices are ordered. Therefore, we can check the consistency of the results across trials to indicate the reliability of these methods.

By preserving order invariance and using randomized order of choices, this method adapts the evaluation to align with the inherent nature of multiple-choice questions. It leverages inherent randomness in the choice order to assess model abilities across multiple trials. This approach can be seen as a revision to existing multiple-choice adaptation methods to better represent the core elements in multiple-choice questions. Furthermore, it can be used to reveal the other reliability issues we concern, testing whether evaluation results stem from genuine language comprehension or the biases introduced by evaluation methods.

5 Experiment

Below we conduct experiments to perform the evaluation through test consistency checking. Our experiments demonstrate that test consistency checking provides valuable insights for probing reliability issues we proposed.

5.1 Models and Benchmarks

We conduct test consistency checking on both pre-trained and fine-tuned large language models.
(1) *LLama* 7B/13B (Touvron et al., 2023a), a widely used pre-trained open-sourced LLM;
(2) *Alpaca* (Taori et al., 2023), a LLama-based language model fine-tuned on instruction data;
(3) *Falcon* (Almazrouei et al., 2023), a high performance language model pre-trained from scratch;
(4) *Falcon-Instruct*, a Falcon-based language model fine-tuned on chat and instruction data;
(5) *LLama 2* 7B/13B (Touvron et al., 2023b), the updated pre-trained successor to LLama;
(6) *LLama 2-Chat* 7B/13B, a LLama 2-based language model optimized on instruction datasets;
(7) *MPT* (Team, 2023), a language model pre-trained by MosaicML from scratch;
(8) *MPT-Chat*, the instruction fine-tuned MPT.
More details about these chosen LLMs families and sizes can be found in Appendix.

We select a diverse benchmark suite covering tests for commonsense reasoning, mathematics, logical reasoning and multidisciplinary knowledge. HellaSwag benchmark are used for commonsense reasoning. It presents story premises with four possible endings. Models must choose the most plausible ending. We select benchmarks from four distinct subjects in the MMLU suite: College Math, Formal Logic, Professional Law and Sociology. All of them contain 4-choice questions. We also use questions from the United States Medical Licensing Examination (USMLE) (Han et al., 2023) for testing medicine-related knowledge, which may contain up to 8 choices in one question.

5.2 Implementation Details

We test each multiple-choice question with m trials. Specifically, we perform 24 trials for 4-choice questions, which covers most of our benchmarks except for the USMLE. For the USMLE, we increase the number of trials to 100 and filter out questions with more than 6 choices. Given limited resources, we randomly select 100 questions for each benchmark.

We test all possible implementations of probability comparison approach, including three adaptation methods—*joint-label*, *joint-desc* and *separate* methods—with three probability scoring methods. The *joint-label* method relies solely on the choice label for probability scoring. As a result, the unnormalized scoring method that sums log likelihoods is equivalent to normalized methods that average log likelihoods. We therefore only use unnormalized scoring for *joint-label* method. In contrast, *joint-desc* and *separate* method utilizes the full text of choices. For these two methods, we conduct both unnormalized and normalized scoring.

5.3 Evaluation with Order Invariance

The initial results through test consistency checking are presented as categorical plots in Figure 2. This enables fair comparisons with preserving or-

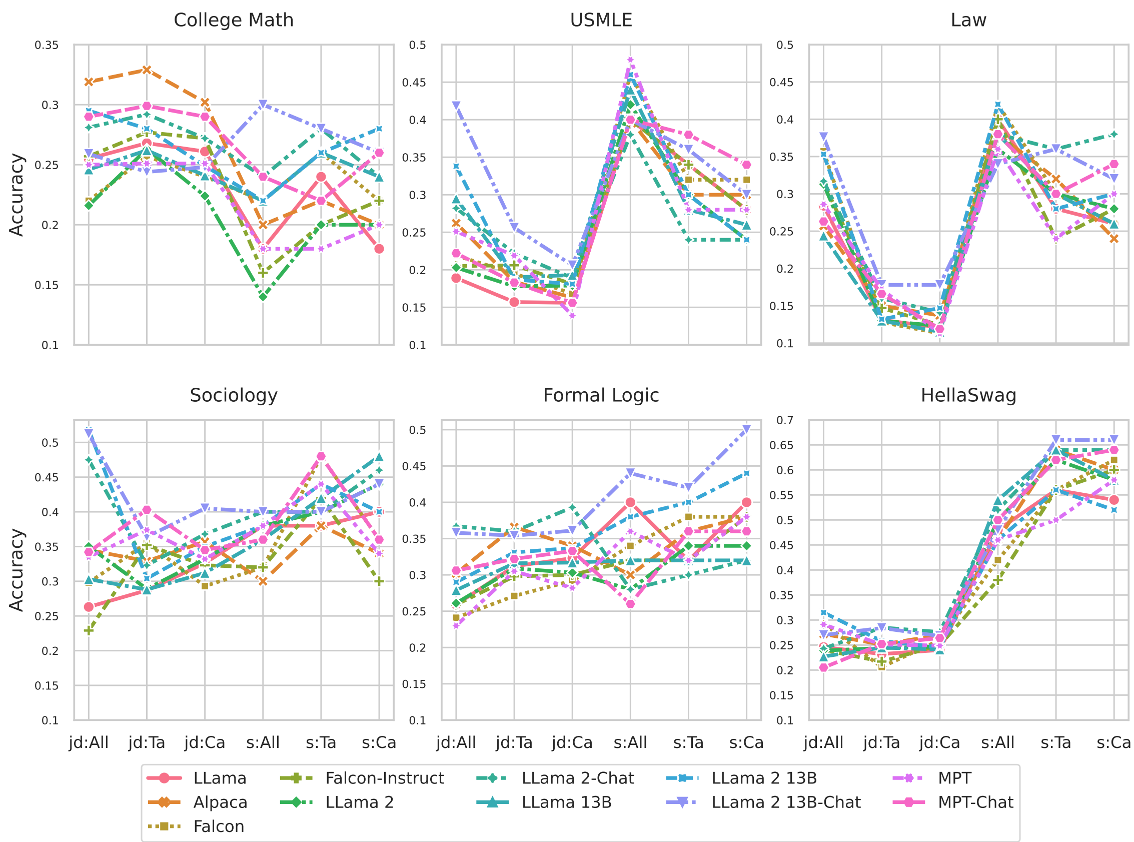

Figure 2: Evaluation of different implementations through test consistency checking. Accuracy scores are reported for comparison. The X-axis represents different implementations, where "jd:All" and "jd:Ca", for example, refer to the *joint-desc* adaptation with unnormalized and character-length normalized scoring method, respectively.

der invariance. By checking whether the predicted answer match the ground truth on every trial, an accuracy score is calculated for each implementation.

The first observation is that there is no consistent preferences for any implementation across models and benchmarks. The results for implementations are broadly sensitive to the benchmark used. For example, using *separate* adaptation always obtains higher accuracy compared to other implementations on the HellaSwag benchmark. However, on the College Math benchmark, *separate* adaptation obtains worse results.

Therefore, the results reveal challenges in comparing capabilities between models, which was partially discussed in Liang et al. (2022) . Varying the implementation can dramatically change measured accuracy between models on the same benchmark. For example, when comparing between LLama and Falcon on the HellaSwag benchmark, LLama achieves higher accuracy with the unnormalized *separate* method. However, with the character-length normalized *separate* method, Falcon outperforms LLama instead. Conclusions about relative model performance can be even more unclear across different benchmarks. Even with multiple trials, the evaluation results may not provide definitive conclusions. It seems that high variability is inevitable with current implementations.

However, certain implementations exhibit model-independent trends on specific benchmarks, suggesting potential underlying biases. The unnormalized *separate* method consistently outperforms others on the USMLE benchmark by a large margin. Conversely, some implementations yield notably low performance. The results obtained from the token-length normalized *joint-desc* method on the Professional Law benchmark are quite low, falling below 0.15. We argue that these results may stem from underlying biases in the implementations rather than genuine performance issues.

5.4 Evaluation on Position

In this section, we track the impact of choice position and analyze how it interacts with implementations. Figure 3 shows categorical plots summarizing the position of predicted choices for testing

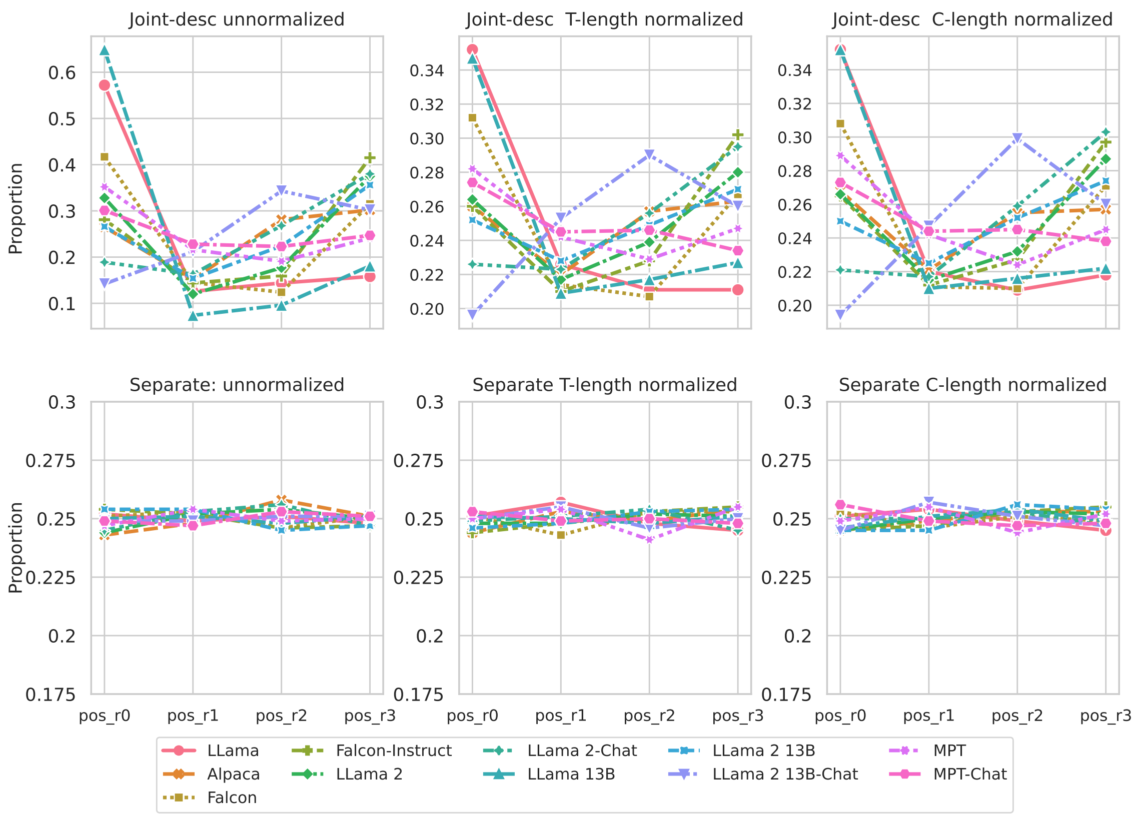

Figure 3: Summarization of the position rank of predicted choices across different implementations and models. Proportions are reported for comparison. "pos_rx" refers to the predicted choices ranked at position x, with "pos_r1" being the choices with the top-ranked position among all candidates.

different models and implementations. The y-axis represents the proportion of times each choice position rank is selected. Since the USMLE benchmark can have more than 4 choices, we exclude results on it, thus an unbiased implementation should yield an expected value of 0.25.

The first observation is a significant contrast between the high variability of results from *joint-desc* methods and the relatively stable tendency on *separate* methods. The *separate* methods exhibit ideal position independence, with near identical proportions around 0.25. The difference between the *joint-desc* method and the *separate* method is that the former uses an extended query that includes the original query and choices, while the latter only uses the original query before concatenating each choice. The inclusion of choices in the query prompt clearly introduces positional bias.

Applying normalization methods reduces the variance in position bias exhibited, but does not eliminate it completely. This is evident from the lower variability in results obtained from the normalized *joint-desc* method compared to unnormalized one. These results confirm our analysis, demonstrating that this bias cannot be eliminated solely by bringing out order invariance. On the other hand, such position bias can partly reveal the relationship between different models, the impact of pre-training and fine-tuning processes, and the influence of different model sizes.

5.5 Evaluation on Length

Figure 4 shows categorical plots summarizing the length of predicted choices for testing different models and implementations. The y-axis represents the proportion of times each length rank is selected, excluding questions where all choices have equal length. The statistical analysis reveals the distribution of the golden choice length rank as follows: {0: 0.298, 1: 0.303, 2: 0.18, 3: 0.219}. For example, the golden option with the highest length rank occurs approximately 29.8% of the time.

We observe that length independence is severely violated in these implementations. Instead, they exhibit consistent yet distinct length-dependent tendencies. On one side, the "the shorter, the more likely" tendency indicates a preference for shorter choices. On the other side, the "the longer, the

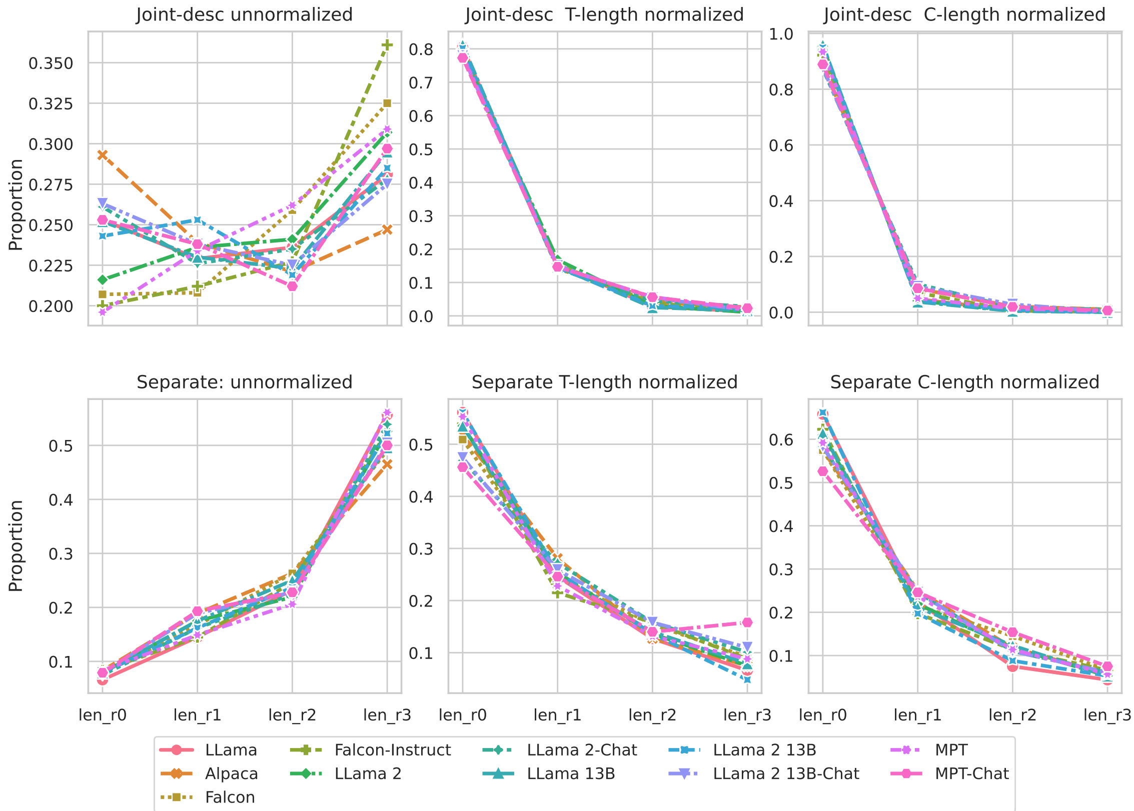

Figure 4: Summarization of the length rank of predicted choices across different implementations and models. Proportions are reported for comparison. “len_r0” refers to the longest candidate choice among the choice set.

more likely" tendency shows a reversed preference.

The unnormalized methods exhibit “the shorter, the more likely" tendency. Since longer choices tend to have lower probabilities, unnormalized methods are biased towards selecting shorter choices intuitively. However, the results from both the *joint-desc* and *separate* methods show that prepending a long prompt can mitigate this biased tendency, or through the fine-tuning process.

The normalized methods exhibit the “the longer, the more likely” tendency. As the choice length increases, the selection probability monotonically rises. Applying normalized *joint-desc* methods tends to predict the longest option as the answer with a probability over 0.75. Using character-length normalized *joint-desc* can push this probability even to 0.95 for some models.

Our findings confirm the implicit connection between length and position, as described in previous research (Kaplan et al., 2020). As the length increases, the total log likelihood decreases, but the per-token/character log likelihood increases. The opposing preferences observed in our results, with unnormalized methods favoring shorter choices and normalized methods favoring longer choices, present a dilemma where the tendencies of “the shorter, the more likely" and “the longer, the more likely" coexist. We conjecture that this partially stems from the core definition of language modeling and becomes ingrained during pre-training, making it fundamentally difficult to solve.

6 Conclusion

This paper primarily focuses on the reliability of multiple-choice evaluation methods. We uncover three intrinsic issues ingrained in current methods that negatively impact reliability. The adaptation and probability scoring processes undermine the fundamental nature of multiple-choice questions: order invariance, position independence, and length independence. To perform reliability checking, we propose a test consistency checking method inspired by the double-slit experiment. The method first brings out the order invariance by leverages multiple trials evaluation through the choice shuffling. Experiments covering 6 benchmarks and different LLMs reveal severe reliability issues harbored within these methods, demonstrating the need for further efforts in evaluation study.

Limitations

In this paper, we study the reliability of implementations used in current automatic multiple-choice evaluation. We uncover the overlooked reliability issues and introduce a method inspired by double-slit experiment to conduct the reliable multiple-choice evaluation. Still, our work is limited in:

- More novel evaluation method: We put our focus on pointing out the reliability issues harbored within current evaluation methods in this work. On the other hand, we consider the test consistency checking method to be a viable approach for multiple-choice evaluation. This method has several advantages: (1) it can reflect the genuine performance of models with preserving order invariance, (2) it can be easily applied to the numerous models that have already been evaluated, and (3) it has the potential to uncover reliability issues related to choice position and length. In the future, we are actively working on developing more novel evaluation methods to further enhance the assessment of multiple-choice tasks.

- Computational resources concerns: The basic algorithm is built on evaluating multiple trials for one multiple-choice question, which can be resource-intensive, especially as the number of choices increases. Studying new strategies that can be applied in limited resources is an important direction in the future, and we limit our work to 4-choices evaluation.

Ethical Statement

In this work, we conduct experiments focusing on testing reliability of LLM evaluation methods. All data and models are open-source and raise little ethical concerns. Further, our work is beneficial for ethical problems in LLM evaluation. By introducing the test consistency checking method, claims about the state-of-the-art performance or the effectiveness of newly released LLMs should be approached with caution. This is because the test consistency checking method may reveal significant differences in performance compared to traditional evaluation methods, offering a more reliable and trustworthy assessment of LLM capabilities.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

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.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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.

Richard P Feynman, Robert B Leighton, and Matthew Sands. 1965. The feynman lectures on physics; vol. i. *American Journal of Physics*, 33(9):750–752.

Leo Gao. 2021. Multiple choice normalization in lm evaluation.

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.

Brian Green. 2005. The fabric of the cosmos: Space, time, and the texture of reality. *Journal of chemical education*, 82(6):822–823.

Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressem. 2023. Medalpaca–an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on*

Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *CoRR*, abs/2305.08322.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. 2023. Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models. *PLoS digital health*, 2(2):e0000198.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yüksekgönül, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic evaluation of language models. *CoRR*, abs/2211.09110.

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

OpenAI. 2023. Gpt-4 technical report.

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`.

MosaicML NLP Team. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. Accessed: 2023-05-05.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

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*.

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.

T Young. 1803. Experiments and calculations relative to physical optics, the philosophical transaction, 1803. *Young, T.(Ed.), A Course of Lectures on Natural Philosophy and the Mechanical Arts II*, pages 639–648.

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

Hui Zeng. 2023. Measuring massive multitask chinese understanding. *arXiv preprint arXiv:2304.12986*.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023. Evaluating the performance of large language models on GAOKAO benchmark. *CoRR*, abs/2305.12474.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *CoRR*, abs/2304.06364.

A Appendix

LLMs in Experiment We selected LLMs based on representativeness, fairness, and performance. Given these criteria, we initially chose LLama due to its common use. Moreover, LLama 2 was recently released with better performance. We also selected the Falcon model, claimed as the best LLM upon its release on the Open LLM Leaderboard. We also considered MPT models from MosaicML, but visualization challenges with more models led us to limit our selection. We believe these models sufficiently support our conclusion.

To maintain a fair comparison, we opted to use the 7B versions of LLMs as default. It was challenging to ensure that all LLM families had the same version of the model. For instance, LLama has versions of 7/13/33/65B, while LLama 2 has versions of 7/13/70B. The Falcon model has 7/40B versions. By standardizing our use of the 7B versions as default, we aimed to keep the comparison as equitable as possible. Due to resource constraints, model versions exceeding 13B were deemed too costly for our project.

Why use 100 questions? In our work, we believe that using 100 questions is sufficient to illustrate the reliability issues we aim to uncover. However, when it comes to reflecting the performance of models on a specific test set, more samples may be required. It is important to note that our proposed method is not impacted by the number of data samples. This means that our evaluation approach can effectively assess model performance regardless of the number of samples available, providing a reliable and consistent evaluation method.