An Automatic and Cost-Efficient Peer-Review Framework for Language Generation Evaluation

Anonymous ACL submission

Abstract

With the rapid development of large language models (LLMs), how to efficiently evaluate them has become an important research question. Existing evaluation methods often suffer from high costs, limited test formats, the needs of human references, and systematic evaluation biases. To address these issues, our study introduces the Auto-PRE, an automatic LLM evaluation framework based on peer review. In contrast to previous studies that rely on human annotations, Auto-PRE selects evaluator LLMs automatically based on their inherent traits including consistency, self-confidence, and pertinence. We have conducted extensive experiments on both summary generation and non-factoid question-answering tasks. Results indicate our Auto-PRE achieves state-of-the-art performance at a lower cost. Moreover, our study highlights the impact of prompt strategies and evaluation formats on evaluation performance, offering guidance for method optimization in the future.

1 Introduction

Recently, the persistent advance of large language models (LLMs) has attracted a lot of attention in both academic and industry (Li et al., 2023; Yang et al., 2023). As LLMs evolve rapidly, how to evaluate their performance effectively and efficiently has become a crucial bottleneck.

Existing evaluation methods for LLMs can be categorized into two types: manual evaluation (Zheng et al., 2023) and automated evaluation (Chang et al., 2024). Manual evaluation is considered the most reliable and effective in general, but it is usually subject to high costs in practice. Automated evaluation aims to reduce the cost by directly assessing model performance without using human annotations. However, existing automated evaluation methods often support limited types of task formats (e.g., multiple-choice questions) and need human-created references for judgments. While recent studies have attempted to build reference-free for open-ended task evaluation with LLMs, research (Zeng et al., 2023) has shown LLM-based evaluators, including the powerful GPT-4 (Achiam et al., 2023), may often prefer answers generated from models sharing the same origin with them, which introduces systematic biases into the evaluation framework and thus has limited reliability in practice.

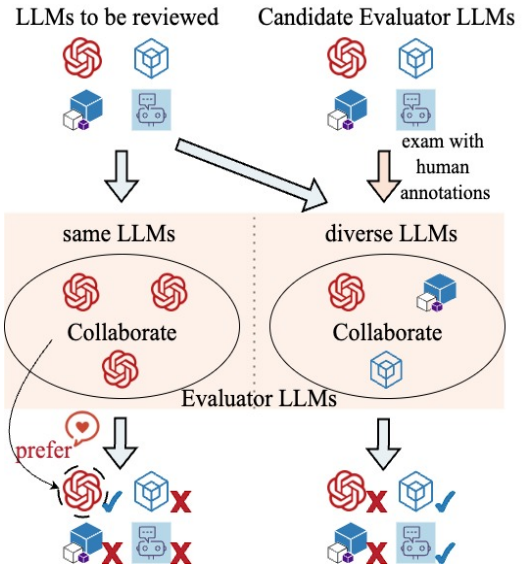

Figure 1: An illustration of existing peer-review methods for language generation evaluation. Methods like ChatEval construct evaluators with homogeneous LLMs, which may systematically prefer outputs generated by models from the same origins; Methods like PRE construct diverse evaluators for peer review but require a human annotated exam to select qualified evaluators.

To this end, recent research has investigated the possibility of employing multiple types of LLMs to evaluate collaboratively (as shown in Figure 1) and achieved notable performance. Similar to human peer review, when unqualified LLMs participate in peer review, they often impair the method's performance. Thus, how to select the appropriate

LLMs as evaluators (or reviewers) is a crucial issue. ChatEval (Chan et al., 2023), for instance, directly selects strong LLMs like GPT-4 to build multiple agents to debate and collaborate for evaluation. However, since it only utilizes LLMs from the same series, it still suffers from systematic biases. PRE (Chu et al., 2024) proposes to build a peer-review system with different types of evaluator LLMs selected with a well-crafted exam, which selects qualified evaluator LLMs by calculating the accuracy of the candidate LLMs' results compared to the manual annotations. Although this method can achieve superior evaluation performance, it still relies on human-annotated data for the exam and thus is not fully automated.

Inspired by PRE, we propose the Auto-PRE by designing an automated qualification exam. Previous studies (Zhao et al., 2015; Zeng et al., 2023) have shown that whether a human evaluator is excellent in the academic peer review system is influenced by three important factors: (1) **Consistency**: whether the reviewer produces consistent evaluation; (2) **Self-Confidence**: whether the reviewers can correctly estimate their evaluation confidence based on task difficulty; (3) **Pertinence**: whether the reviewer can capture the key information that distinguishes different candidates without affected by superficial factors that are not important for evaluation. Based on the above observations, we design three selection methods to filter evaluator LLMs automatically for peer review. All these methods involve no human annotations, thus making the whole framework a fully automated one that is open-ended, reference-free, cost-efficient, and robust to systematic biases introduced by LLMs.

Experiment results on both summary generation and non-factoid question-answering tasks indicate that our Auto-PRE can achieve state-of-the-art performance similar to PRE and ChatEval at a much lower cost. Additionally, we analyze how prompt strategies and evaluation formats could affect our framework to improve the generalizability of our work and provide more insights for future automated evaluation methods based on LLMs.

In conclusion, the contributions of this paper are as follows:

- We propose an automatic peer-review evaluation framework, i.e., Auto-PRE, by designing an automated qualification exam that selects qualified evaluator LLMs based on three LLM's traits.
- On both summary generation and non-factoid question-answering tasks, Auto-PRE achieves performance comparable with state-of-the-art methods at a much lower cost.

2 Related Work

2.1 Large Language Models

There have been numerous distinctive large language models (LLMs) designed by academic and industrial communities, which can be categorized into open-source and closed-source LLMs: Closed-source LLMs are represented by OpenAI's GPT series (OpenAI, 2022), Anthropic's Claude series (Anthropic, 2023), and Google's released Gemini (Team et al., 2023). Open-source LLMs are famously represented by Meta's Llama (Touvron et al., 2023). Based on Llama, many researchers have conducted extensive derivative work to enhance its performance. Representative works include Vicuna (Chiang et al., 2023), Alpaca (Taori et al., 2023), and et al. Apart from Llama, many researchers have also attempted to train their LLMs independently, including the Baichuan series (I, 2023), the ChatGLM series (Zeng et al., 2022), and FastChat-T5 (LMSYS, 2024).

2.2 Evaluation Methods For LLMs

Current evaluation methods for LLMs can be categorized into two types: manual evaluation and automated evaluation, the latter can be further divided into reference-based, multiple-choice questions, and LLM-based.

Manual evaluation has always been considered the most effective and reliable evaluation method (Frieder et al., 2023). For example, LMSYS establishes a platform, Chatbot Arena (Zheng et al., 2023). The platform randomly selects two different LLMs to chat with the users, who are unaware of the two LLMs' information and need to choose which one performs better. Though this method can be effective, as the number of LLMs and evaluation tasks grows rapidly, the cost becomes increasingly high and unsustainable.

Reference-based evaluation calculates similarity metrics between the reference answer text and the generated text to assess the quality of the generated text. Common metrics include BLEU(Papineni et al., 2002), ROUGE (Lin, 2004), BERTScore (Zhang et al., 2019) and so on. However, these metrics do not fully capture answer quality. Moreover, if reference answers are disclosed, LLMs can learn and optimize for them, making the test data useless.

Multiple-choice questions evaluation utilizes the format of multiple-choice questions with fixed answer outputs to simplify the evaluation process. GAOKAO (Zhang et al., 2023a) is a specific example. However, this format can't cover all real-world problem types, especially open-ended questions without fixed answers. Additionally, this format mainly assesses LLMs' understanding rather than their generation capabilities.

LLM-based evaluation uses LLMs to conduct evaluation tasks. ChatEval (Chan et al., 2023) uses several GPT series LLMs to play different roles and debate with each other and its findings suggest that the collaborative evaluation results produced by multiple LLMs have higher consistency with manual annotations. PRE (Chu et al., 2024) emulates the peer review mechanism in academia, selecting qualified evaluator LLMs from various candidate LLMs. The final evaluation results are aggregated from the original evaluation results from all evaluators. Its experiments indicate PRE can outperform various baselines, including the method that uses a single GPT-4 as an evaluator. It also effectively mitigates systematic biases associated with using a single type of LLMs as evaluators.

2.3 Meta-Evaluation For Evaluator LLMs

Although LLM-based evaluation methods have shown competitive evaluation performance, these methods still have many flaws, such as a preference for verbose answers or answers generated by similar LLMs (Zeng et al., 2023). Therefore, many researchers have begun to introduce meta-evaluation benchmarks to assess whether these LLM-based methods can achieve high consistency with manual annotations. Some of this work involves constructing evaluation benchmarks through random sampling output pairs and crowdsourced manual annotations, such as LLMEval (Zhang et al., 2023b), MT-Bench (Zheng et al., 2023), and FairEval (Wang et al., 2023), but these works overlook biases introduced by the subjective preferences of human annotators. To address this, some researchers attempt to create more objective meta-evaluation benchmarks; for instance, LLMBAR (Zeng et al., 2023) is an evaluation dataset designed to check if LLMs follow instructions in their outputs, with data samples that have been manually checked by the authors to ensure objectivity.

3 Proposed Method

3.1 Motivation

As discussed in Section 1, one of the key issues in peer-review-based evaluation is how to automatically, objectively, and cost-effectively select qualified evaluator LLMs. Inspired by the idea that qualified evaluator LLMs should possess traits similar to those of excellent human evaluators, we have designed selection methods based on these traits.

Due to the complex nature of humans, directly and comprehensively summarizing the traits of excellent human evaluators in an academic peer review system is not a trivial task. However, according to previous studies (Zhao et al., 2015; Zeng et al., 2023), we can find the following three important traits: (1) Excellent evaluators should not be influenced by the order in which papers are reviewed, treating each paper fairly and maintaining consistency throughout the evaluation; (2) Excellent evaluators should correctly estimate their confidence in their evaluation results based on their abilities and the difficulty of the evaluation task; (3) Excellent evaluators should objectively evaluate papers based on key information that distinguishes the quality of different papers, and avoid being influenced by superficial factors that are not important for evaluation. Based on the above three important traits that excellent human evaluators have, we design three methods including consistency, self-confidence, and pertinence to filter qualified evaluator LLMs.

3.2 Automated Qualification Exam

Figure 2 shows the framework of our automated qualification exam. This exam incorporates three different selection methods that reflect the traits evaluator LLMs should possess: (1) **Consistency**: evaluators should not be influenced by the content presentation order in the prompt; (2) **Self-confidence**: evaluators should have a reasonable self-confidence level on their evaluations based on their understanding of the task difficulty and their capabilities; (3) **Pertinence**: evaluators should objectively evaluate based on the key information that distinguishes different answers like the pertinence of the answer to the given question, rather than making subjective judgments based on superficial quality of the answer itself like answer length or tone. Next, we will introduce these three selection methods in detail.

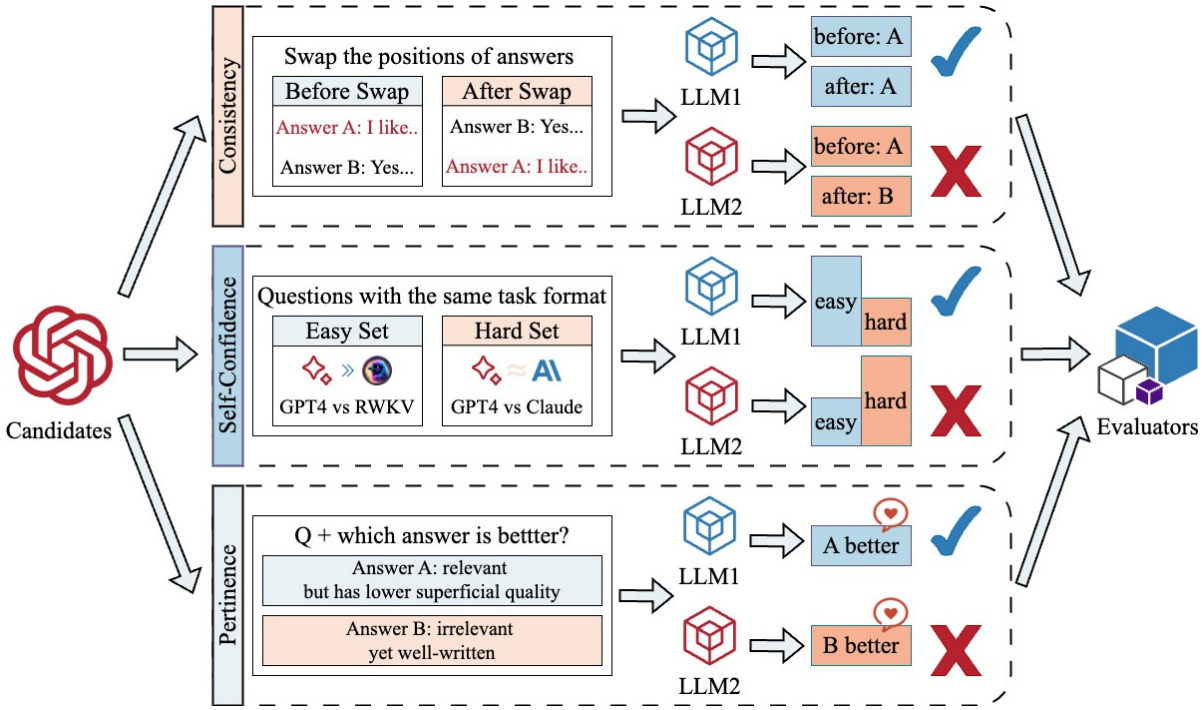

Figure 2: The framework of our automated qualification exam. (1) **Consistency** filters by swapping the positions of answers within prompts and calculating the proportion of consistent output by the LLM before and after the swap; (2) **Self-Confidence** selects based on whether the LLM shows higher confidence level on easier question set when it faces two question sets with the same task format but objectively different difficulties; (3) **Pertinence** selects based on whether the LLM can evaluate based on the pertinence of the answers to the question without affected by the superficial quality of the answers themselves.

3.2.1 Consistency

Similar to how human evaluators may be influenced by the order in which papers are reviewed, a lot of research has discovered that many LLMs also exhibit strong positional biases (Wang et al., 2023). For instance, when asked to evaluate the quality of two answers to the same question, the results provided by some LLMs may be influenced by the positions of the answers in the prompt. Based on this, we can design a selection method as follows:

Suppose there are a question Q and two different answers Y_1 and Y_2. Then the input to the candidate LLM L should include the tuple (Q, Y_1, Y_2), and L is required to provide a preference relation T_1 for Y_1 and Y_2. Subsequently, swap the positions of Y_1 and Y_2 in the prompt to form the tuple (Q, Y_2, Y_1), and input it again to L to obtain the preference relation T_2. If T_1 and T_2 are the same, the candidate L is considered to maintain consistent output before and after the swap; otherwise, it is considered inconsistent. When the proportion of consistent output by the candidate L exceeds a threshold η, it is considered to pass the qualification exam.

3.2.2 Self-Confidence

Like excellent human evaluators, qualified evaluator LLMs should have a reasonable self-confidence level in their evaluations based on their understanding of the task difficulty and their capabilities. As for what counts as a reasonable level of self-confidence, a suitable prior assumption is that when the same LLM encounters **two questions with the same task format but objectively different difficulties**, it should have more self-confidence in solving the easier question. It is noteworthy that in the above assumption, the task formats of the two questions must be the same to ensure that the capabilities required by the LLM to solve the two questions are the same. Additionally, the difference in difficulties between the two questions must be based on objective criteria rather than human subjective judgment, to eliminate the bias that may arise from the disagreement between humans and LLMs.

Based on this assumption, we select those LLMs that show higher self-confidence on the easier set than on harder ones as evaluator LLMs. Two issues need to be addressed when it comes to implemen-

Table 1: Summaries by different LLMs

	Content
Original News	The man's body was discovered in a field near Belsyde Avenue...
GPT-4's Summary	A man's body was found in a field near Belsyde Avenue...
Claude's Summary	The body of an unidentified man was discovered in a field near...
RWKV's Summary	Bob: Hey Alice, have you heard about the death of a man...

tation: (1) How to construct two question sets with the same task format but objectively different difficulties? (2) How to extract the self-confidence of LLMs?

Regarding issue (1), we initially selected the evaluation task format to be evaluating the quality difference between two answers to the same question. Then we construct the easy and hard question sets based on the following assumption: The smaller the capability gap between the two LLMs that generate the two answers, the more similar the quality of the answers. Consequently, the task of evaluating the quality difference between the two answers is objectively harder; conversely, it is easier. Based on this, we construct the easy and hard sets by controlling the capability gap between the two LLMs which generate the two answers, using LLMs with a significant capability gap to form the easy set and those with minimal gap to form the hard set.

Table 1 shows a specific example where the task is summary generation. In the ELO leaderboard of LMSYS released in September 2023 (Zheng et al., 2023), GPT-4 ranks 1st and Claude 2nd, while RWKV (Peng et al., 2023) ranks 21st. This suggests a significant capability gap between GPT-4 and RWKV and a minimal gap between GPT-4 and Claude. As Table 1 shows, both GPT-4 and Claude generated summaries that meet the task requirements, whereas RWKV produced a dialogue irrelevant to the task. **Objectively**, this makes the task of evaluating the quality difference between the answers from GPT-4 and Claude harder, while it is easier to evaluate between GPT-4 and RWKV.

Regarding issue (2), we design two methods for extracting the self-confidence of LLMs: The first method is **Direct Prompting**: this method prompts the LLM to directly output specified self-confidence level labels. It is straightforward and intuitive but is influenced by the prompt strategies. Additionally, some LLMs may struggle to directly output self-confidence labels due to lesser comprehension abilities. Thus, we propose the second method, **Probability Transformation**: this method converts the probability of outputting a specific token into the uncertainty of the output (Duan et al., 2023; Manakul et al., 2023), and assumes that higher uncertainty represents lower self-confidence. It aligns more closely with the inherent nature of LLM outputs, avoiding biases introduced during output. However, it requires access to the probabilities of the specific tokens, making it unsuitable for some closed-source LLMs. We argue that these two methods can complement each other to better extract the self-confidence of LLMs: For some closed-source LLMs that do not provide access to specific token probabilities but generally have larger parameter sizes and stronger capabilities, self-confidence can be obtained by direct prompting. Conversely, for open-source LLMs with relatively smaller parameter sizes and weaker capabilities, which cannot extract their self-confidence by direct prompting, we can extract self-confidence by probability transformation.

To validate the effectiveness of our selection method, we initially select some LLMs as candidates and filter unqualified evaluator LLMs whose confidence is higher on hard set than easy set (we call these LLMs show a reversal of confidence). Then we compare our selection results with those relying on manual annotations. The results indicate that LLMs that show a reversal of confidence also show lower accuracy in the qualification tests based on manual annotations, which demonstrates the effectiveness of our selection method. Detailed experiment results and analysis are in Appendix D.

3.2.3 Pertinence

When humans act as evaluators, they may sometimes focus only on the **superficial quality** of the answers, such as the length and format of the answer, neglecting their **pertinence** to given question and thus making incorrect evaluations (Zeng et al., 2023). Similarly, we can select evaluator LLMs based on whether the candidate LLMs can distinguish between the pertinence of the answer to the given question and its superficial quality.

To implement this selection method, we generate answers that are highly pertinent to given questions

Table 2: Some cases of how GPT-4 modifies Q to Q'

Q	Q'
Could someone define **Christian** for me?	Can anyone explain the concept of **Buddhism** to me?
What is the best Fantasy **Football** Platform?	What is the top Fantasy **Baseball** App?
What do **photojournalists** do?	What do **marine biologists** do?

but of lower superficial quality as **RA: Relevant Answers** and answers that are less pertinent but of higher superficial quality as **IA: Irrelevant yet well-written Answers**. Specifically, the process of constructing IA involves the following two steps:

(1) Generate a variant of the original question Q, denoted as Q', where Q and Q' are similar but sufficiently different to ensure that answers generated based on each have significantly different pertinence to the original question Q. The difference in pertinence decreases as the similarity between Q and Q' increases. There are two methods for constructing Q': one is to search other questions from the same dataset as Q to find a suitable Q', and the other is to prompt a capable LLM (such as GPT-4) to modify Q to obtain Q'. Table 2 shows that GPT-4 mainly achieves the transformation from Q to Q' by changing the keywords in the question Q.

(2) Select the answers from another LLM in response to Q' as the IA. Here, the LLM generating the IA can be a more capable LLM than the LLM generating the RA, or it can be the candidate LLM itself. The former is based on the assumption that a more capable LLM is likely to produce answers with higher superficial quality, while the latter assumes that the candidate LLM considers its own answers to be of sufficient superficial quality.

To verify our assumption and the effectiveness of our selection method, we conduct extensive experiments and the results show that our selection method has a significant ability to discriminate different candidate LLMs. What's more, we explore the robustness of the *pertinence* by varying the variables within it. From the results, we can observe that the performance of smaller parameterized candidate LLMs is more significantly affected, while the performance of larger parameterized candidate LLMs remains almost unchanged. The more detailed analysis can be found in Appendix E.

4 Experimental Setup

4.1 Tasks And LLMs Selection

Unlike the automated evaluation methods for multiple-choice questions mentioned in Section 1, we focus on open-ended questions. To this end, we select two representative generative tasks: summary generation and non-factual question-answering, and choose typical datasets for each.

Summary generation involves generating a summary for a given text. For this task, we utilize the Extreme Summary (Xsum) dataset (Narayan et al., 2018), which consists of over 220,000 real single-document news and summaries collected by the British Broadcasting Corporation (BBC).

Non-factual question-answering refers to providing answers to questions that do not have fixed responses. For this task, we choose the NF-CATS dataset (Bolotova et al., 2022). It contains about 12,000 non-factual questions and these questions are categorized into eight classes to differentiate the difficulty levels.

For the above two datasets, we randomly sample 100 examples from each as the question sets. Then, we select eleven representative LLMs to generate the answers to these question sets. The basic information and all the uses of these LLMs are summarized in Table 5 (in Appendix A). We use the score and rank from the leaderboard published by LMSYS in September 2023 (Zheng et al., 2023) as a fundamental reference for the capabilities of these LLMs. Additionally, we reuse the manual preference annotations over seven LLMs' outputs from the PRE as ground truth to evaluate the performance of each evaluation method. We also use the experimental settings from the original PRE, utilizing the answers and their corresponding manual annotations from three LLMs as PRE's qualification exam data. Moreover, we select seven LLMs as candidate LLMs. To ensure the stability and reproducibility of the evaluation results, we set the *temperature* as 0 and *do_sample* as False.

4.2 Prompt Strategies

Section 3.2.2 introduces the method for extracting the self-confidence of LLMs by direct prompting. When implementing it, we designed four different prompt strategies as shown in Table 6 (in Appendix B) by varying two aspects: the self-confidence level labels and the granularity.

What's more, Section 3.2.2 also introduces the method for extracting the self-confidence of LLMs

Table 3: The overall performance of our Auto-PRE and other baselines. 'pass' records the qualified LLMs (Vicuna-7b-v1, ChatGLM3-6B, Baichuan-2-13b, FastChat-t5-3b, GPT-3.5-turbo, ChatGLM-Pro, and GPT-4 are abbreviated as integers 1-7, respectively). The best result is highlighted in bold and the second-best result is underlined.

methods	Xsum						NF-CATS					
	pairwise		5-level		100-level		pairwise		5-level		100-level	
	pass	acc	pass	acc	pass	acc	pass	acc	pass	acc	pass	acc
ChatEval	–	0.6584	–	0.5694	–	0.5747	–	0.7366	–	0.6009	–	0.6435
GPT-4	–	0.7369	–	0.6893	–	0.7005	–	0.7815	–	0.6330	–	0.6801
PRE	[3,4,5,6,7]	**0.7423**	[2,3,4,5,6,7]	0.7211	[2,3,4,5,6,7]	0.7192	[4,5,6,7]	0.7801	[6,7]	0.6824	[6,7]	**0.7104**
wo-filter PRE	all	0.7401	all	0.7055	all	0.7002	all	0.7542	all	0.6804	all	0.6711
Auto-PRE (C)	[4,5,6,7]	0.7381	[4,5,6,7]	0.7064	[4,5,6,7]	0.7133	[5,6,7]	0.7664	[5,6,7]	0.6795	[5,6,7]	0.6905
Auto-PRE (S)	[3,4,5,6,7]	0.7398	[3,4,5,6,7]	0.7086	[3,4,5,6,7]	0.7114	[3,4,6,7]	0.7598	[3,4,6,7]	0.6735	[3,4,6,7]	0.6702
Auto-PRE (P)	[2,4,5,6,7]	0.7379	[2,4,5,6,7]	**0.7231**	[2,4,5,6,7]	**0.7195**	[4,5,6,7]	0.7702	[4,5,6,7]	**0.6836**	[4,5,6,7]	0.6774
Auto-PRE (A)	[4,5,6,7]	0.7412	[4,5,6,7]	0.7047	[4,5,6,7]	0.7144	[6,7]	**0.7821**	[6,7]	0.6777	[6,7]	**0.7104**

by probability transformation. In practical implementation, to minimize the influence of irrelevant tokens, we choose to simplify the output content. Specifically, we set the task format as pairwise (evaluating the quality difference of two answers to the same question). The candidate LLMs are only required to output the specific token 'one' or 'two' to indicate which answer is better. Through this simplification, we can directly convert the probability p of an LLM outputting 'one' or 'two' into the LLM's uncertainty ($-logp$) (Duan et al., 2023; Manakul et al., 2023), thereby obtaining the self-confidence of the LLM. The output restriction statement we designed is: "You only need to output 'one' or 'two' directly to indicate which answer is better." Then we design two types of prompts based on the position of the output restriction statement within the prompt:

(1) **prompt1**: The output restriction statement is placed at the beginning of the prompt, i.e., "Output restriction statement + Question + Two answers."

(2) **prompt2**: The output restriction statement is placed at the end of the prompt, i.e., "Question + Two answers + Output restriction statement."

4.3 Evaluation Formats And Metrics

We compare two evaluation formats: pointwise and pairwise. For pointwise, we design two implementation approaches: 5-level and 100-level. For pairwise evaluation, we minimize the bias introduced by the positioning of answers by calculating the mean of the evaluation results before and after swapping the positions of the two answers. The specific information can be found in Appendix C.

We analyze the experimental results using various metrics and methods such as accuracy (the agreement rate between the manual preference annotations and the evaluation results. When using pointwise evaluation, we will convert each answer's individual scores into pairwise rankings), Kendall's tau coefficient (Kendall, 1938), Spearman's rank correlation coefficient (Lehman et al., 2013), t-test (Student, 1908), and rank-sum test (McKnight and Najab, 2010).

4.4 Baselines

We compare Auto-PRE with state-of-the-art methods including PRE, wo-filter PRE (all candidate LLMs are treated as evaluator LLMs), ChatEval (uses two GPT-3.5-turbo to build two agents as evaluators to debate in two rounds with one-by-one communication strategy) and GPT-4 (uses a single GPT-4 as the evaluator). See more detailed settings in Appendix F.

5 Results And Analysis

In this section, we present the experimental results and attempt to answer the following two research questions (RQs):

(1) How does the performance of our Auto-PRE compare to these baseline methods?

(2) What advantages does our Auto-PRE have in balancing cost and performance?

5.1 Overall Results (RQ1)

In this section, we compare Auto-PRE with various baseline methods. Table 3 shows the overall performance of our Auto-PRE and other baseline methods on the Xsum and NF-CATS datasets. The results indicate that under various task settings, Auto-PRE achieves performance that is comparable to state-of-the-art methods and on the NF-CATS, Auto-PRE performs the best. Additionally, the evaluation format has a significant impact on the performance of the methods and the experiments show that pairwise generally outperforms point-

Table 4: The detailed settings and cost of various variants of evaluation methods. In Auto-PRE, if two same LLMs are used, the difference is the prompt strategy. Take the ChatEval-3 as an example, testing a single sample requires calling GPT-3.5-turbo's API four times, so its cost is approximately 4 ($/1 M tokens).

Method	Settings	Cost ($/1 M tokens)
ChatEval-1 (C1)	one GPT-3.5-turbo as evaluator; one role; one round of debate; one-by-one communication strategy	1
ChatEval-2 (C2)	two GPT-3.5-turbo as evaluators; two roles; one round of debate; one-by-one communication strategy	2
ChatEval-3 (C3)	two GPT-3.5-turbo as evaluators, two roles; two rounds of debate; one-by-one communication strategy	4
Auto-PRE-1 (A1)	open-source LLMs; one ChatGLM-Pro	1
Auto-PRE-2 (A2)	open-source LLMs; one ChatGLM-Pro; one GPT-3.5-turbo	2
Auto-PRE-3 (A3)	open-source LLMs; one ChatGLM-Pro; two GPT-3.5-turbo	3
Auto-PRE-4 (A4)	open-source LLMs; two ChatGLM-Pro; two GPT-3.5-turbo	4
Auto-PRE-5 (A5)	open-source LLMs; one ChatGLM-Pro; one GPT-3.5-turbo ; one GPT-4	42

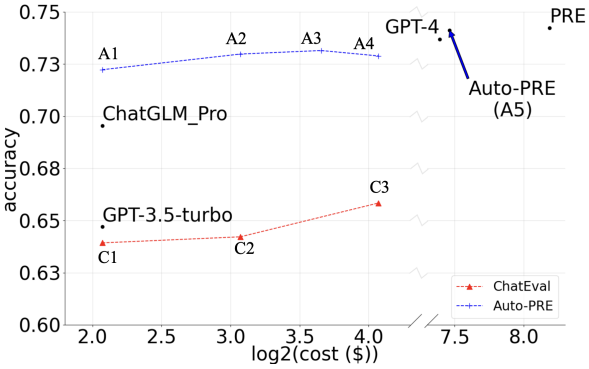

Figure 3: The performance on the Xsum (The labels on the line represent different variants (Table 4)

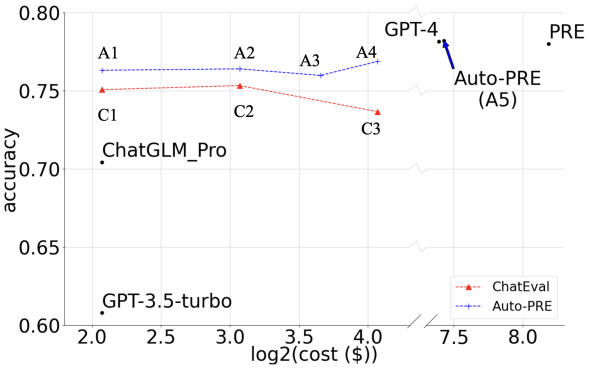

Figure 4: The performance on the NF-CATS

wise. More analysis and discussion can be found in Appendix G.

5.2 Cost Analysis (RQ2)

In addition to comparing the performance of Auto-PRE with various baselines, we are also interested in the impact of evaluation cost on performance. Hence, we implement various variants of the above methods to compare the evaluation performance at different costs (as shown in Table 4). We use pairwise as the evaluation format, each task has 4200 samples and each sample has about 1K tokens, so completing each task requires approximately 4.2 M tokens. Based on the official pricing released (glm; gpt), the costs of ChatGLM_Pro and GPT-3.5-turbo are estimated to be similar at 1 ($/1 M tokens). The cost of GPT-4 is estimated at 40 ($/1 M tokens). Open-source LLMs (like FastChat-t5-3b) are considered cost-free. Additionally, the cost of the qualification exam of PRE based on human annotations is about $115 while the cost of our automated qualification exam (less than 1$) can be neglected compared to the total costs.

Figures 3 and 4 show the relationship between the total cost and accuracy. The results show that at the same cost, Auto-PRE can achieve higher performance than all baselines. Moreover, Auto-PRE can achieve performance comparable with state-of-the-art methods at a much lower cost.

6 Conclusion

This paper develops the Auto-PRE by designing an automated qualification exam based on three LLMs' inherent traits: (1) **Consistency**: swapping the positions of answers within prompts, whether the LLM produces consistent output before and after the swap; (2) **Self-Confidence**: when facing two question sets with the same task format but objectively different difficulties, whether the LLM shows more confidence on the easier set. (3) **Pertinence**: whether the LLM can differentiate the pertinence of the answers to the question and the superficial quality of the answers themselves. Experiment results indicate that, across various scenarios, our Auto-PRE can achieve comparable performance to state-of-the-art at a much lower cost. Moreover, we analyze how prompt strategies and evaluation formats could affect evaluation performance to explore optimization methods for future automated evaluation methods based on LLMs.

7 Limitations

We think our work has two main limitations:

1. Due to cost considerations, the number of LLMs used in our experiments remains limited, for example, LLMs such as Claude-2 and Llama-70b have not been set as candidate LLMs. However, we believe our Auto-PRE has good generality and scalability. Therefore, future work can increase the number of LLMs to conduct larger-scale experimental validations.

2. We need to explore deeper into some experimental phenomena that have been observed, such as designing and validating more efficient prompt strategies.

In summary, we will continue striving towards developing an automated evaluation framework for LLMs that is low-cost, reference-free, capable of addressing diverse real-world scenarios, and minimizes systemic biases as much as possible.

8 Ethics Statement

Throughout this research, ethical considerations have been integral to ensuring the responsible development and application of AI technologies. We are committed to the principles of open research and the scientific value of reproducibility. Therefore, we have made all code from our study publicly accessible on GitHub. This transparency allows the community to validate our results and promotes the use of our methods in various settings.

References

The price of chatglm-pro. https://open.bigmodel.cn/pricing. Accessed: 2024-06-13.

The price of gpt-3.5-turbo. https://openai.com/api/pricing/. Accessed: 2024-06-13.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2023. Introducing claude. Online. Available: https://www.anthropic.com/index/introducing-claure.

Valeriia Bolotova, Vladislav Blinov, Falk Scholer, W Bruce Croft, and Mark Sanderson. 2022. A non-factoid question-answering taxonomy. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1196–1207.

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, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

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)*, 2(3):6.

Zhumin Chu, Qingyao Ai, Yiteng Tu, Haitao Li, and Yiqun Liu. 2024. Pre: A peer review based large language model evaluator. *arXiv preprint arXiv:2401.15641*.

Jinhao Duan, Hao Cheng, Shiqi Wang, Chenan Wang, Alex Zavalny, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, Alexis Chevalier, and J J Berner. 2023. Mathematical capabilities of chatgpt. *ArXiv*, abs/2301.13867.

Technology B I. 2023. Baichuan-7b. Online. Accessed: 2023-9-6.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Ann Lehman, Norm O'Rourke, Larry Hatcher, and Edward Stepanski. 2013. *JMP for basic univariate and multivariate statistics: methods for researchers and social scientists*. Sas Institute.

Beibin Li, Konstantina Mellou, Bo Zhang, Jeevan Pathuri, and Ishai Menache. 2023. Large language models for supply chain optimization. *arXiv preprint arXiv:2307.03875*.

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

LMSYS. 2024. Fastchat. Online. Accessed: 2024-2-11.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Patrick E McKnight and Julius Najab. 2010. Mann-whitney u test. *The Corsini encyclopedia of psychology*, pages 1–1.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

OpenAI. 2022. Introducing chatgpt. Online. Available: https://openai.com/blog/chatgpt.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. 2023. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*.

Student. 1908. The probable error of a mean. *Biometrika*, pages 1–25.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

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

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

David F Williamson, Robert A Parker, and Juliette S Kendrick. 1989. The box plot: a simple visual method to interpret data. *Annals of internal medicine*, 110(11):916–921.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, P. Zhang, Yuxiao Dong, and Jie Tang. 2022. Glm-130b: An open bilingual pre-trained model. *ArXiv*, abs/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*.

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

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023a. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*.

Xinghua Zhang, Bowen Yu, Haiyang Yu, Yangyu Lv, Tingwen Liu, Fei Huang, Hongbo Xu, and Yongbin Li. 2023b. Wider and deeper llm networks are fairer llm evaluators. *arXiv preprint arXiv:2308.01862*.

Yun Wei Zhao, Chi-Hung Chi, and Willem-Jan van den Heuvel. 2015. Imperfect referees: Reducing the impact of multiple biases in peer review. *Journal of the Association for Information Science and Technology*, 66(11):2340–2356.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Haotong Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685.

Table 5: The basic information and all the uses of LLMs. ChatGLM2-6B has evolved to ChatGLM3-6B during our work; hence, while reusing the manual annotations of ChatGLM2-6B, we choose ChatGLM3-6B as a candidate LLM

LLM	ELO score (rank)	LLM to be reviewed	manual annotation	PRE qualification exam data	candidate LLM
GPT-4	1193 (1/28)	✓			✓
Claude-1	1161 (2/28)	✓	✓		
GPT-3.5 -turbo	1118 (5/28)	✓	✓	✓	✓
Llama-2 -70b-chat	1060 (7/28)	✓			
Vicuna -7b-v1	1003 (14/28)	✓	✓		✓
ChatGLM 2(3)-6B	965 (18/28)	✓	✓		✓
RWKV-4-Raven-7B	14B:939 (21/28)	✓	✓		
Alpaca-7b	13B:919 (22/28)	✓	✓	✓	
FastChat -t5-3b	888 (25/28)	✓	✓	✓	✓
ChatGLM -Pro	–	✓			✓
Baichuan2 -13b	–	✓			✓

A The Basic Information And All Uses of LLMs

In this section, we show the basic information and all the uses of LLMs, which can be found in Table 5. The specific meaning of each column refers to section 4.1.

B Four Different Prompt Strategies For Extracting The Self-Confidence Of LLMs

In this section, we show the four different prompt strategies for extracting the self-confidence of LLMs by direct prompting, which can be found in Table 6.

C Evaluation Formats

In this section we show the three different evaluation formats, which can be found in Table 6.

D Self-Confidence Results

D.1 Is The Self-Confidence Of LLM Sensitive To Different Prompt Strategies?

In this section, we investigate how prompt strategies affect the self-confidence extracted from the

Table 6: Four different prompt strategies for extracting the self-confidence of LLMs. Fine-grained prompts are achieved by adding specific explanations for each level label, and coarse-grained prompts are unexplained.

Prompt Strategies	**self-confidence level labels**	**granularity**
num	[1,2,3,4,5]	coarse-grained
num _explanation	[1,2,3,4,5]	fine-grained
doubtful	[‘doubtful’, ‘uncertain’, ‘moderate’, ‘confident’, ‘absolute’]	fine-grained
null	[‘null’,‘low’, ‘medium’, ‘high’,‘expert’]	fine-grained

Table 7: Three evaluation formats

Evaluation Format	**Template**
5-level	###Task Description###. Directly output a number between 1 and 5 to indicate the quality score of this answer: - 1 means the answer is irrelevant to the question - 2 means the answer is related to the question, but does not solve the question - 3 means the answer only solves a part of the question - 4 means the answer solves majority aspects of the question, but not perfect - 5 means the answer is perfect to solve the question ###Question+Answer###
100-level	###Task Description### Directly output a number between 0 and 100 to indicate the score of this answer. The higher the score, the higher the quality of the answer. ###Question+Answer###
pairwise	Prompt1 (default) and Prompt2 described in section 4.2

LLM. We compare four different prompt strategies as shown in Table 6 and select ChatGLM-Pro as the evaluator LLM; GPT-4, Claude-1, and RWKV-4-Raven-7B as LLMs to be reviewed. Using these three LLMs, we generate answers to the Xsum question set and paired them in a pairwise evaluation format to create a total of 300 paired examples. To minimize the positional bias of answers within the prompts, we swap the positions of the two answers in each sample, resulting in a total of 600 paired examples. The experiment results are shown in Table 8. To further verify whether the consistency of the results for ‘doubtful’ and ‘null’ is general, we expand the number of LLMs to be reviewed, including Alpaca-7b, Vicuna-7b-v1, GPT-3.5-turbo, Claude-1, FastChat-t5-3b, GPT-4, Llama2-70b-chat, and RWKV-4-Raven-7B, with other settings remaining unchanged. The results are shown in Figure 5. We can find that when varying the pairs of the LLMs being tested, the consistency between the two sets of results under the ‘doubtful’ and ‘null’ settings appears random, and the degree of consistency does not exhibit a clear relationship with the capability gap between the two LLMs being tested. Therefore, we can know that the self-confidence of LLMs is sensitive to prompt strategies, highlighting the importance of designing appropriate prompt strategies.

Table 8: The impact of prompt strategies on the self-confidence of LLM. Only the results for ‘doubtful’ and ‘null’ in the first row show significant consistency (p-value less than 0.05), while the results of the other groups all exhibit inconsistency.

Comparison group	Spearman’s coefficient (p-value)	Kendall’s coefficient (p-value)
doubtful vs null	**0.2198(0.003)**	**0.2144(0.003)**
doubtful vs num	0.0377(0.604)	0.0372(0.603)
doubtful vs num_explanation	-0.0356(0.622)	-0.0350(0.621)
null vs num	-0.0205(0.781)	-0.0202(0.780)
num vs num_explanation	-0.0143(0.842)	-0.0141(0.842)
null vs num_explanation	-0.0035(0.962)	-0.0034(0.962)

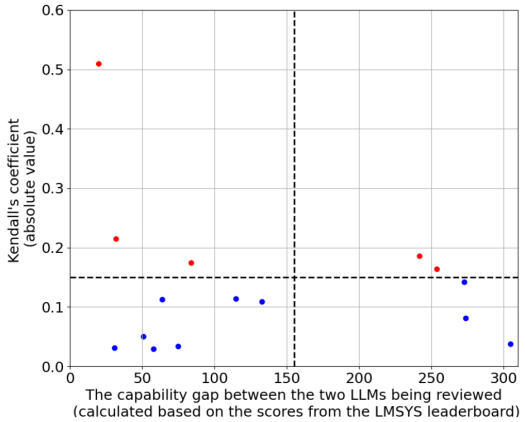

Figure 5: The impact of different pairs of LLMs to be reviewed on the consistency between ‘doubtful’ and ‘null’, where red dots indicate significant consistency (p-value less than 0.05), and blue dots indicate non-significant consistency (p-value greater than 0.05).

D.2 Does The Self-Confidence Of LLM Reverse When Performing Two Question Sets With Same Task Format But Objectively Different Difficulties?

Table 9: The self-confidence level of LLMs extracted by direct prompting on the Xsum dataset. The 'doubtful' and 'null' are two prompt strategies. After obtaining self-confidence level labels on both question sets, we converted them into integers from 1 to 5 (with higher numbers indicating greater confidence) and then calculated the mean and standard deviation as the results. The two data in each row can be considered to have significant differences (p-value less than 0.05 in t_test and rank_sum test).

different settings	easy	hard
doubtful_ChatGLM-Pro	4.0824±0.5333	3.7474±0.4796
null_ChatGLM-Pro	3.9738±0.4020	3.8883±0.4293
doubtful_ChatGLM3-6B	4.0331±0.8913	3.8101±0.9322
null_ChatGLM3-6B	**3.9206±1.0979**	**4.0964±1.0155**
doubtful_GPT-3.5-turbo	4.2050±0.5683	3.7450±0.4359
null_GPT-3.5-turbo	4.0500±0.2179	3.9600±0.2417

We initially select some candidate LLMs for filtering tests, and then compare our selection results with those relying on manual annotations to examine whether the LLMs exhibiting a reversal of self-confidence inversion also perform poorly in the qualification tests based on manual annotations. Table 9 and 10 (in Appendix D) show the self-confidence level of LLMs extracted by two methods on the Xsum dataset. The easy set is formed by GPT-4 and RWKV, the hard set is formed by GPT-4 and Claude. Both experiment results show that the self-confidence level of ChatGLM3-6B on the easy set is lower than on the hard set, indicating a reversal of confidence. Therefore, it is filtered out. Then we conduct the filtering test based on manual annotation and the results are shown in Figure 6. It indicates that LLMs exhibiting a reversal of confidence also show lower accuracy in the qualification tests based on manual annotations, which demonstrates the effectiveness of our selection method.

Table 10: The self-confidence level of LLMs extracted by probability transformation on the Xsum dataset. The 'prompt1' and 'prompt2' are two prompt strategies. The results show represent the mean and standard deviation of uncertainty, **with higher values indicating lower self-confidence**. The two data in each row can be considered to have significant differences (p-value less than 0.05 in t_test and rank_sum test), except that marked with *.

different settings	easy	hard
prompt1 + ChatGLM3-6B*	0.6090±0.2027	0.6407±0.1960
prompt2 + ChatGLM3-6B	**0.5573±0.1932**	**0.5015±0.1671**
prompt1 + Baichuan2-13b	0.3198±0.2288	0.4088±0.1776
prompt2 + Baichuan2-13b	0.3631±0.2137	0.4308±0.1881
prompt1 + GPT-3.5-turbo	0.2871±0.3008	0.3776±0.2844
prompt2 + GPT-3.5-turbo	0.2102±0.2444	0.3269±0.2516

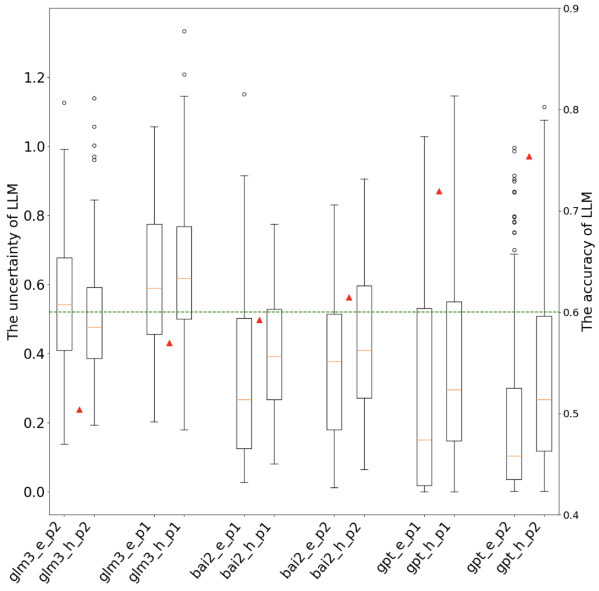

Figure 6: The left vertical axis shows the uncertainty of LLM and higher uncertainty indicates lower self-confidence. The right vertical axis shows the accuracy of LLM in the qualification test based on manual annotations. The horizontal axis represents different experimental groups (each group is further divided into the easy and hard sets, for example, 'glm3_e_p2' means: LLM is ChatGLM3-6B, prompt is 'prompt2', and the question set is 'easy'). The accuracy of the LLM in the manual annotation-based selection is marked by red triangles, while the uncertainty data are depicted as box plots (Williamson et al., 1989). If a paired box's median (represented by the yellow line) shows the left higher than the right, it indicates a reversal of confidence.

D.3 More Self-Confidence Results On The NF-CATS

In this section, we show some self-confidence results on the NF-CATS (Table 11). Despite the task difficulty of the NF-CATS being quite different from that of the Xsum, our selection method still proved effective, validating that our method is general.

Table 11: The self-confidence level of LLMs extracted by probability transformation on the NF-CATS dataset. The 'prompt1' and 'prompt2' are two prompt strategies. The results show represent the mean and standard deviation of uncertainty, **with higher values indicating lower self-confidence**.

different settings	easy	hard
prompt1 + ChatGLM3-6B	**0.2638±0.2015**	**0.1730±0.1313**
prompt2 + ChatGLM3-6B	0.4271±0.3709	0.4763±0.3666
prompt1 + GPT-3.5-turbo	**0.2146±0.2630**	**0.1957±0.2206**
prompt2 + GPT-3.5-turbo	**0.2308±0.2237**	**0.2304±0.2514**
prompt1 + Baichuan2-13b	0.5051±0.2101	0.5189±0.1852
prompt2 + Baichuan2-13b	0.4186±0.2854	0.4287±0.2585
prompt1 + Fastchat-t5-3b	0.1775±0.2163	0.1960±0.1945
prompt2 + Fastchat-t5-3b	0.2808±0.2179	0.4112±0.1898
prompt1 + Vicuna-7b-v1	0.3820±0.1541	0.4197±0.1510
prompt2 + Vicuna-7b-v1	**0.5214±0.2806**	**0.5098±0.3034**

Table 12: L1 represents the LLM that generates RA, and L2 represents the LLM that generates IA. The Xsum dataset is used as the question Q, with GPT-4 employed to modify Q into Q'. We choose **'prompt1'** as prompt and calculate the accuracy of candidate LLMs as the result (the proportion where the RA is rated better than the IA).

L1	L2	ChatGLM 3-6B	FastChat -t5-3b	Baichuan2 -13b	ChatGLM -Pro	GPT-3.5 -turbo
GPT-4	Baichuan2 -13b	0.765	0.865	0.670	0.995	0.980
Vicuna-7b-v1		0.740	0.855	0.670	0.990	0.975
ChatGLM2-6B		0.730	0.855	0.670	0.975	0.970
GPT-4	GPT-3.5 -turbo	0.770	0.865	0.630	0.990	0.955
Vicuna-7b-v1		0.700	0.860	0.640	0.970	0.945
ChatGLM2-6B		0.705	0.820	0.650	0.970	0.950
GPT-4	self -generate	0.740	0.835	0.670	0.995	0.960
Vicuna-7b-v1		0.715	0.800	0.670	0.985	0.935
ChatGLM2-6B		0.695	0.765	0.670	0.985	0.960

E Pertinence Results

E.1 The Impact Of LLMs That Generate RA (L1) And IA (L2)

In our experiments, we select ChatGLM3-6B, FastChat-t5-3b, Baichuan2-13b, ChatGLM-Pro, and GPT-3.5-turbo as candidate LLMs. Table 12 (in Appendix E) shows that the LLMs used for generating RA (L_1) and IA (L_2) significantly impact the results. Moreover, the results generally show a trend of decreasing as the capability of L_1 diminishes and that of L_2 increases. This is because when we choose a weaker LLM L_1 and a relatively stronger LLM L_2, the surface quality of the RA declines while that of the IA improves, leading the candidate LLMs to be more easily misled by the IA, thus lowering their accuracy. It is worth noting that when L_2 is the candidate LLM itself, referred to as 'self-generate', the performance of the candidate LLMs does not significantly differ from when L_2 is another LLM, indicating that using the candidate itself as a source for generating IA is feasible.

Table 13: L1 represents the LLM that generates RA, and L2 represents the LLM that generates IA. The Xsum dataset is used as the question Q, with GPT-4 employed to modify Q into Q'. We choose **'prompt2'** as prompt and calculate the accuracy of candidate LLMs as the result (the proportion where the RA is rated better than the IA).

L1	L2	ChatGLM 3-6B	FastChat -t5-3b	Baichuan2 -13b	ChatGLM -Pro	GPT-3.5 -turbo
GPT-4	Baichuan2 -13b	0.505	0.705	0.750	0.990	0.985
Vicuna-7b-v1		0.505	0.675	0.710	0.980	0.960
ChatGLM2-6B		0.500	0.650	0.750	0.975	0.970
GPT-4	GPT-3.5 -turbo	0.500	0.680	0.730	0.980	0.955
Vicuna-7b-v1		0.500	0.655	0.700	0.965	0.945
ChatGLM2-6B		0.500	0.640	0.735	0.970	0.930
GPT-4	self -generate	0.500	0.590	0.750	0.990	0.960
Vicuna-7b-v1		0.500	0.590	0.710	0.970	0.930
ChatGLM2-6B		0.500	0.565	0.750	0.965	0.935

E.2 The Impact Of Prompt Strategies

We modify the prompt used to 'prompt2', keeping other settings the same with the previous experiments. Table 13 shows the results. We can observe that the performance of smaller parameterized candidate LLMs is more significantly affected, while the performance of larger parameterized candidate LLMs remains almost unchanged. Specifically, ChatGLM3-6B and FastChat-t5-3b show a decrease in performance under the prompt2 setting compared to the prompt1 setting, with ChatGLM3-6B's accuracy even dropping to 0.5 in many cases under prompt2, which is nearly equivalent to the accuracy of randomly generated answers. On the other hand, Baichuan2-13B seems to handle prompt2 better, showing improved performance over prompt1; ChatGLM-Pro and GPT-3.5-turbo exhibit robust performance across both prompt strategies. This underscores the importance of careful design in prompt strategies again.

E.3 The Impact Of Methods For Getting Q'

We also explore the effects of different methods for constructing Q'. Table 14 shows the results. Firstly, the impact of prompt strategies on the candidate LLMs is consistent with the observations from section G.2. Secondly, because the Q' obtained by random searching often exhibits a greater from Q, the pertinence scores of the RA are easily higher than those of the IA. Consequently, all candidate LLMs, except for FastChat-t5-3b, achieve higher accuracy under this setting than before. The anomalous performance of FastChat-t5-3b suggests that it

Table 14: L1 represents the LLM that generates RA, and L2 represents the LLM that generates IA. The Xsum dataset is used as the question Q. We randomly search a question that is different from Q as Q' from the Xsum dataset. We calculate the accuracy of the candidate LLMs as the result (the proportion where the RA is rated better than the IA).

L1	prompt	ChatGLM 3-6B	FastChat -t5-3b	Baichuan2 -13b	ChatGLM -Pro	GPT-3.5 -turbo
GPT-4	prompt1	0.855	0.595	0.730	1.000	1.000
Vicuna-7b-v1		0.795	0.600	0.755	1.000	1.000
ChatGLM2-6B		0.775	0.595	0.720	1.000	1.000
GPT-4	prompt2	0.500	0.505	0.820	1.000	1.000
Vicuna-7b-v1		0.495	0.515	0.825	1.000	1.000
ChatGLM2-6B		0.500	0.530	0.845	0.995	1.000

Table 15: L1 represents the LLM that generates RA, and L2 represents the LLM that generates IA. The NF-CATS dataset is used as the question Q, with GPT-4 employed to modify Q into Q'. We choose **'prompt1'** as prompt and calculate the accuracy of candidate LLMs as the result (the proportion where the RA is rated better than the IA).

L1	L2	ChatGLM 3-6B	FastChat -t5-3b	Baichuan2 -13b	ChatGLM -Pro	GPT-3.5 -turbo
GPT-4	FastChat -t5-3b	0.573	0.853	0.735	1.000	0.895
Vicuna-7b-v1		0.565	0.888	0.670	1.000	0.913
ChatGLM2-6B		0.540	0.858	0.665	0.975	0.810
GPT-4	Baichuan2 -13b	0.500	0.858	0.525	0.938	0.828
Vicuna-7b-v1		0.495	0.873	0.505	0.755	0.773
ChatGLM2-6B		0.495	0.868	0.520	0.845	0.600
GPT-4	GPT-3.5 -turbo	0.500	0.845	0.540	0.918	0.803
Vicuna-7b-v1		0.485	0.873	0.530	0.885	0.728
ChatGLM2-6B		0.490	0.848	0.525	0.800	0.725

Table 16: L1 represents the LLM that generates RA, and L2 represents the LLM that generate IA. The NF-CATS dataset is used as the question Q, with GPT-4 employed to modify Q into Q'. We choose **'prompt2'** as prompt and calculate the accuracy of candidate LLMs as the result (the proportion where the RA is rated better than the IA).

L1	L2	ChatGLM 3-6B	FastChat -t5-3b	Baichuan2 -13b	ChatGLM -Pro	GPT-3.5 -turbo
GPT-4	FastChat -t5-3b	0.560	0.898	0.725	1.000	0.935
Vicuna-7b-v1		0.553	0.910	0.730	0.993	0.950
ChatGLM2-6B		0.553	0.888	0.700	0.965	0.888
GPT-4	Baichuan2 -13b	0.523	0.885	0.595	0.920	0.905
Vicuna-7b-v1		0.513	0.870	0.515	0.855	0.815
ChatGLM2-6B		0.493	0.865	0.515	0.790	0.738
GPT-4	GPT-3.5 -turbo	0.468	0.890	0.535	0.905	0.835
Vicuna-7b-v1		0.510	0.880	0.535	0.860	0.730
ChatGLM2-6B		0.455	0.843	0.465	0.775	0.665

may be less sensitive to changes in the pertinence score difference between positive and IA.

E.4 More Pertinence Results On The NF-CATS

Table 15 and 16 show more 'pertinence' results on the NF-CATS. The observations are almost the same as those on the Xsum, which proves our selection method is general. Specifically, we observe that the candidate LLMs exhibit greater robustness to 'prompt1' and 'prompt2' on the NF-CATS dataset than on the Xsum dataset. This may be due to the shorter length of samples on the NF-CATS compared to the Xsum, which reduces the differences between 'prompt1' and 'prompt2'.

Table 17: The specific details of different settings.

method	selection method	threshold	weight
PRE	manual annotations	0.6	accuracy in the exam
wo-filter PRE	–	–	1
Auto-PRE (C)	Consistency	0.55	1
Auto-PRE (S)	Self-confidence	–	1
Auto-PRE (P)	Pertinence	0.7	1
Auto-PRE(A)	All-three	–	average

F The Detailed Setting Of Different Methods

In this section, we show the detailed setting of different methods, which can be found in Table 17.

The threshold is served as the pass line for candidate LLMs in the exam. The 0.6 threshold for PRE and the 0.55 threshold for Auto (C) are both referenced from the PRE work, while the 0.7 threshold for Auto (P) is based on the results in the experiments of Section E. In practical applications, these parameters can be flexibly adjusted based on their performance across validation sets for different tasks.

The weight refers to the fusion weight of each evaluator LLM when merging all evaluation results in the final stage. For Auto-PRE(A), the weight is 'average' which means the average accuracy of LLM in three selection methods: Consistency (the proportion of consistent output), Self-Confidence (default as 1), and Pertinence (the proportion where the RA is rated better than the IA).

G More Analysis and Discussion About The Overall Result

Firstly, we find that although Auto-PRE (A) always outperforms the other variants of Auto-PRE under pairwise evaluation, it is less effective than Auto-PRE with only one selection method under pointwise evaluation. As for the possible reasons, on the one hand, this may be due to the inherently larger result biases associated with the pointwise evaluation format; on the other hand, it may be because the three selection methods we currently designed are all organized in a pairwise format rather than a pointwise format, which may compromise their

effectiveness in pointwise evaluation. In the future, we will further design pointwise selection methods to enhance our framework.

What's more, it is interesting that sometimes, the wo-filter PRE results in superior final evaluation performance compared to outcomes from stringent selections. A plausible explanation for this could be that, in the wo-filter PRE, more unqualified LLMs are incorporated into the peer-review system. While these LLMs indeed have a negative impact on performance due to their inaccurate evaluations, they also contribute positively by reducing biases introduced by powerful but high-bias evaluator LLMs, such as GPT-4, during the merging stage. This inspires us to consider two directions for future research: on the one hand, exploring how to integrate the biases of LLMs (preferring answers generated from similar LLMs) into our selection methods; on the other hand, beyond designing an effective qualification exam, it is also worth investigating how to merge the evaluation results to achieve more unbiased results under the peer review framework.