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

Junjie Chen[1,2], Weihang Su[1], Zhumin Chu[1], Haitao Li[1], Yujia Zhou[1], Dingbo Yuan[3], Xudong Wang[3], Jun Zhou[3], Yiqun Liu[1], Min Zhang[1], Shaoping Ma[1], Qingyao Ai[2,1]*

[1]Department of Computer Science and Technology, Tsinghua University
[2]Quan Cheng Laboratory
[3]Ant Group
aiqy@tsinghua.edu.cn

Abstract

The rapid development of large language models (LLMs) has highlighted the need for efficient and reliable methods to evaluate their performance. Traditional evaluation methods often face challenges like high costs, limited task formats, dependence on human references, and systematic biases. To address these limitations, we propose Auto-PRE, an automatic LLM evaluation framework inspired by the peer review process. Unlike previous approaches that rely on human annotations, Auto-PRE automatically selects evaluator LLMs based on three core traits: consistency, pertinence, and self-confidence, which correspond to the instruction, content, and response stages, respectively, and collectively cover the entire evaluation process. Experiments on three representative tasks, including summarization, non-factoid QA, and dialogue generation, demonstrate that Auto-PRE achieves state-of-the-art performance while significantly reducing evaluation costs. Furthermore, the structured and scalable design of our automatic qualification exam framework provides valuable insights into automating the evaluation of LLMs-as-judges, paving the way for more advanced LLM-based evaluation frameworks.

Code — https://github.com/cjj826/Auto-PRE
Extended version — https://arxiv.org/pdf/2410.12265

Introduction

Recently, the rapid advancement of large language models (LLMs) has attracted significant attention from both academia and industry (Yang et al. 2025; Liu et al. 2024). As LLMs evolve rapidly, how to evaluate their performance effectively and efficiently has become a crucial question.

Existing evaluation methods for LLMs can be categorized into two types: manual evaluation (Zheng et al. 2023) and automatic evaluation (Chang et al. 2024). Manual evaluation is considered the most reliable and effective method, but it is usually suboptimal due to its high costs in practice. Automatic evaluation aims to reduce the cost by directly assessing model performance without human annotations. However, existing automatic 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 frameworks for open-ended task evaluation with LLMs, research (Zeng et al. 2024) has shown LLM-based evaluators (or reviewers), including the powerful GPT-4 (Achiam et al. 2023), may exhibit a preference for answers generated by models with the same origin. In our paper, we refer to this preference as a systematic bias, which could limit the reliability of the evaluation framework in practice.

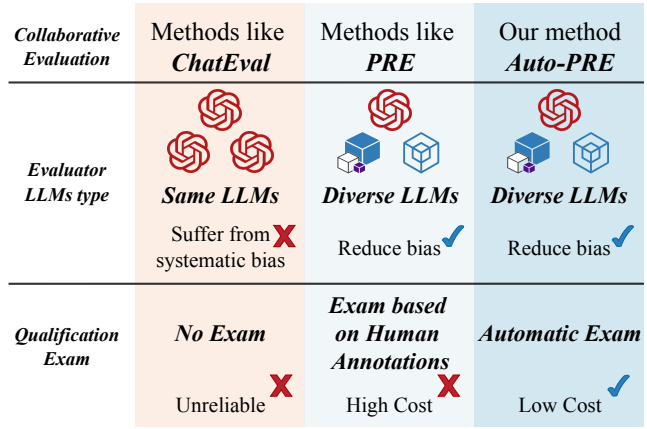

Figure 1: Comparison of existing collaborative evaluation methods. Our Auto-PRE offers advantages in reducing bias and lowering cost.

To develop more advanced evaluation methods, recent research has investigated the possibility of employing multiple LLMs to evaluate collaboratively, achieving notable performance. However, similar to human collaboration, when unqualified LLMs participate in collaboration, they often impair the method's performance. Therefore, selecting the appropriate LLMs as evaluators is a crucial issue. As shown in Figure 1, methods like ChatEval (Chan et al. 2024) directly select powerful LLMs such as GPT-3.5-turbo or 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 the systematic bias. Another method, PRE (Chu et al. 2024), simulates the academic peer-review mechanism by selecting qualified evaluator LLMs from a di-

*Corresponding author.

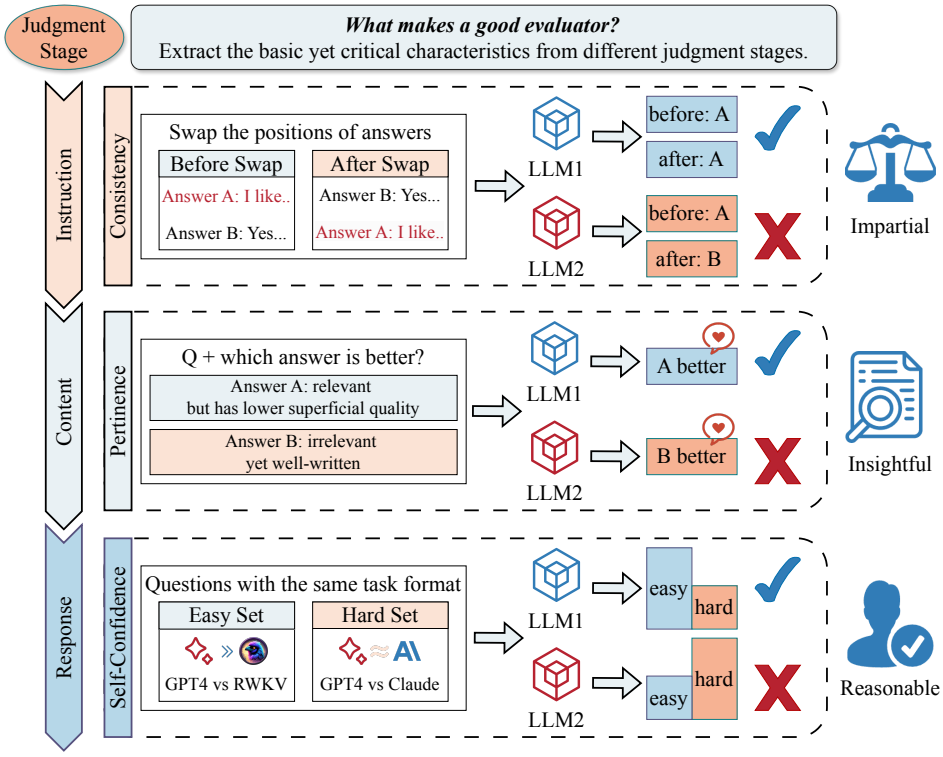

Figure 2: The framework of our automatic qualification exam. (1) *Consistency* measures the proportion of consistent outputs by the LLM after swapping answer positions in prompts; (2) *Pertinence* assesses whether the LLM evaluates based on the pertinence of answers to the question, unaffected by their superficial quality; (3) *Self-Confidence* determines if the LLM exhibits higher confidence on easier question sets when facing two sets of the same format but objectively different difficulties.

verse pool of candidates through a qualification exam, which measures the accuracy of their outputs against human annotations. The final evaluation scores are then computed via weighted aggregation of the selected evaluators' outputs. Although experiments indicate PRE can reduce the systematic bias, it still relies on human-annotated data for the exam and is therefore not fully automatic and high-cost.

In order to address the aforementioned limitations, we propose Auto-PRE, a peer-review framework that introduces an automatic qualification exam to select qualified evaluator LLMs without relying on human annotations. To achieve this, the key question we need to answer is: *What makes a good evaluator?* Our guiding principle is that qualified evaluator LLMs should exhibit characteristics similar to those of excellent human evaluators. However, the inherent complexity of human evaluators makes directly summarizing their characteristics challenging. To strike a balance between rationality and feasibility, we first structure the evaluation process into three stages: Instruction (evaluation prompt), Content (material to be assessed), and Response (assessment result generated by the evaluator). This division covers the entire evaluation process and ensures the completeness of our exam. We then extract basic yet critical characteristics from each stage: (1) Consistency: Upon receiving the judgment instruction, the evaluator should have no preset biases to ensure the objectivity; (2) Pertinence: When judging the specific content, the evaluator is expected to have a thorough understanding of the task and identify the core factors that truly impact the quality of the answers (e.g., pertinence to given question), rather than relying solely on secondary or superficial factors; (3) Self-Confidence: After providing the judgment response, the evaluator should have a reasonable confidence to reflect the reliability (Zhao, Chi, and van den Heuvel 2015). Based on these characteristics, we propose three automatic methods for selecting evaluator LLMs for peer review. All these methods require no human annotations, making the framework fully automatic, cost-efficient, scalable, and robust to LLM-introduced systematic bias.

Experimental results on three tasks, including summary generation, non-factoid question-answering, and dialogue generation, indicate that our Auto-PRE can achieve state-of-the-art performance at a much lower cost. Furthermore, our qualification exam covers the entire evaluation process and is easily adapted to support additional traits, serving as a potential guide for automating the evaluation of LLMs-as-judges and advancing the LLM-based evaluation methods.

Related Work

Evaluation methods for LLMs fall into manual and automatic types. Manual evaluation (e.g., Chatbot Arena (Zheng et al. 2023)) offers high reliability but is expensive and non-scalable. Automatic methods include: (1) reference-based

metrics such as BLEU (Papineni et al. 2002), ROUGE (Lin 2004), and BERTScore (Zhang et al. 2019), which are limited in capturing answer quality, and vulnerable to overfitting if references are leaked; (2) multiple-choice evaluation, which is simple but insufficient for open-ended tasks; and (3) LLM-based evaluation (Chan et al. 2024; Chu et al. 2024), where LLMs serve as evaluators. Although LLM-based evaluation methods have shown competitive evaluation performance, these evaluator LLMs still have many flaws, such as a preference for verbose answers or answers generated by similar LLMs (Zeng et al. 2024). Therefore, researchers have begun to explore whether these evaluator LLMs are truly qualified. Some of this work involves constructing evaluation benchmarks through random sampling output pairs and crowdsourced manual annotations, such as LLMEval (Zhang et al. 2023), 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. 2024) is designed to assess if evaluator LLMs follow instructions, with data samples that have been manually checked by experts to ensure objectivity. In contrast to these annotation-heavy evaluation methods, our methods automatically select evaluator LLMs based on objective traits across different judgment stages.

Methodology

As previously discussed, one of the key issues in collaborative evaluation methods is how to automatically, objectively, and cost-effectively select qualified evaluator LLMs. Based on the different judgment stages, we extract basic yet critical characteristics: Consistency, Pertinence, and Self-Confidence. Leveraging these three characteristics, we have designed an automatic qualification exam that incorporates three different selection methods, as shown in Figure 2. It is worth noting that the ability requirements for evaluators may vary across different tasks or even between datasets within the same task. However, the requirements tend to be similar for different instances within the same dataset of a given task. To balance effectiveness and cost-efficiency, our selection methods are designed to be performed per dataset. Next, we provide a detailed description of our methods.

Consistency

When receiving judgment instructions, an excellent evaluator should remain objective, impartial, and consistent, avoiding the influence of preset biases. Consider a task to evaluate the quality of n answers, denoted as $\{Y_1, Y_2, \ldots, Y_n\}$, to the same question Q. Within the judgment instructions, there exists a subset that should not affect the evaluation outcomes, denoted as NI. This subset includes factors such as the order of answers and the placement of question-answer pairs within the prompt. A qualified evaluator should demonstrate no bias towards specific NI. Formally, this means the evaluation score for any given answer Y_t should remain invariant under changes to NI:

$$Score(Y_t \mid NI_1) = Score(Y_t \mid NI_2), \tag{1}$$

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?

Table 1: Some cases of how GPT-4 modifies Q to Q'.

where NI_1 and NI_2 represent different NI.

Currently, numerous studies (Wang et al. 2023; Li et al. 2024) have shown that some LLMs exhibit various preset biases, so we can use the degree of biases of different LLMs for selection. Specifically, we implement the most common position bias and set $n = 2$. The candidate LLM L is given the tuple (Q, Y_1, Y_2) as input, and generates a preference relation T_1. Next, the positions of Y_1 and Y_2 are swapped to form the tuple (Q, Y_2, Y_1), which is then inputted to L to generate another preference relation T_2. We randomly sample m instances from the specific dataset, with $(Q_i, Y_{1,i}, Y_{2,i})$ $(i = 1, 2, \ldots, m)$. Then, the proportion of consistent outputs for the candidate LLM L is computed as:

$$P_c = \frac{\sum_{i=1}^{m} \mathbb{I}(T_{1,i} = T_{2,i})}{m}, \tag{2}$$

where $\mathbb{I}(\cdot)$ is the 0-1 indicator function, and this definition remains consistent throughout our paper. If the P_c exceeds the threshold η_c, the candidate LLM L is considered to have passed the exam. It should be noted that our framework is highly scalable and can be easily adapted to support the detection of other types of biases based on task requirements, such as token or prompt bias.

Pertinence

When judging specific contents, unqualified evaluators may struggle to identify the core factors that truly impact answer quality (e.g., the pertinence between answer and question) (Zeng et al. 2024). Instead, they tend to provide unreliable scores based on secondary or superficial factors such as the length or format of the answer. Therefore, we design a method to select insightful evaluator LLMs based on whether the candidate LLMs can effectively distinguish between the answer's pertinence to the given question and its superficial quality. To implement this method, we generate two types of answers: (1) answers that are highly pertinent to the given questions but of lower superficial quality, denoted as RA (Relevant Answers); and (2) answers that are less pertinent but exhibit higher superficial quality, denoted as IA (Irrelevant yet well-written Answers). Specifically, the process of constructing RA and IA involves two steps:

Step 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. We design 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

	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...

Table 2: Summaries by different LLMs.

a capable LLM (like GPT-4) to modify Q to obtain Q'. Table 1 shows that GPT-4 mainly achieves the transformation from Q to Q' by changing the keywords in the Q.

Step 2: Select one LLM's response to Q as the RA and another LLM's 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.

After obtaining m pairs of RA and IA, we calculate the proportion of candidate LLM outputs where the RA_i is rated better than the IA_i as a filtering metric:

$$P_p = \frac{\sum_{i=1}^{m} \mathbb{I}(RA_i > IA_i)}{m}. \quad (3)$$

If P_p exceeds a certain threshold η_p, the candidate LLM is considered qualified.

Self-Confidence

After providing judgment responses, like reliable human evaluators, qualified evaluator LLMs should have a reasonable self-confidence level based on their understanding of the task difficulty and their capabilities (Zhao, Chi, and van den Heuvel 2015). 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 for both questions must be identical to ensure that the LLM requires the same capabilities to solve them. Additionally, the difference in difficulties between the two questions must be based on objective criteria rather than human subjective judgment, to eliminate potential biases arising 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 the harder ones as evaluators. Implementing this requires addressing two key issues: (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 select evaluating the quality differences between two answers to the same question as our task format. For this task, an objective principle holds: the more similar the two answers in quality, the more difficult it becomes to distinguish between them. Based on this, we construct the easy and hard question sets by assuming that the similarity in answer quality is correlated with the capability gap between the two LLMs that generate them. Specifically, we pair LLMs with a large capability gap (e.g., GPT-4 vs. RWKV-7B) to form the easy set, and LLMs with similar capabilities (e.g., GPT-4 vs. Claude) to form the hard set. Table 2 presents a summarization example where GPT-4 and Claude generate valid summaries, while RWKV-7B produces an irrelevant dialogue. This highlights that comparing high-quality answers (e.g., GPT-4 vs. Claude) is objectively more challenging than answers with a clear quality gap (e.g., GPT-4 vs. RWKV-7B).

Regarding issue (2), we convert the probability of outputting a specific token into the uncertainty of the output (Duan et al. 2023; Manakul, Liusie, and Gales 2023), and assume that higher uncertainty represents lower self-confidence. Specifically, for the task we select, which involves evaluating the quality difference between two answers, 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 $-log(p)$, thereby obtaining self-confidence. However, this method requires access to the probabilities of the specific tokens, making it unsuitable for some closed-source LLMs. For this situation, we directly prompt these LLMs to output specific self-confidence level labels. Since these popular and closed-source LLMs (e.g., GPT-4) typically have an enormous number of parameters and exceptional capabilities, this straightforward and low-cost method shows good performance in our experiments.

If the candidate LLM's average confidence level on the easy set is S_{easy}, and its average confidence level on the hard set is S_{hard}, then P_s is defined as follows:

$$P_s = \mathbb{I}(S_{\text{easy}} > S_{\text{hard}}) \quad (4)$$

More details on the implementation and validation of our methods' effectiveness are in Appendix Section 3.

Experimental Setup

Tasks And Datasets

Unlike the automatic evaluation methods for multiple-choice questions, we focus on open-ended questions. To this end, we select three generative tasks along with representative datasets: (1) Summary generation: Xsum (Narayan, Cohen, and Lapata 2018), (2) Non-factual question-answering: NF_CATS (Bolotova et al. 2022), (3) Dialogue generation: DailyDialog (Li et al. 2017). For the above three datasets, we randomly sample 100 instances from each as the question set and select 7 different LLMs to generate answers, forming the answer set. Then, we employ human annotators to provide annotations over the answer set. We use these annotations as ground truth to evaluate the performance of each

Methods	Xsum			NF_CATS			DailyDialog		
ROUGE-L		0.5798			/			0.6984	
BERTScore		0.5901			/			0.6143	
GPTScore		0.6910			0.5940			0.3445	
	5-level	100-level	pairwise	5-level	100-level	pairwise	5-level	100-level	pairwise
Vicuna-7b-v1*	0.5100	0.5215	0.5106	0.5503	0.5735	0.5036	0.5712	0.5302	0.5000
ChatGLM3-6B*	0.6564	0.6288	0.5127	0.5518	0.5551	0.5244	0.5806	0.6093	0.5213
Baichuan2-13b*	0.5745	0.6247	0.6057	0.5521	0.5500	0.5515	0.5483	0.6055	0.6260
FastChat-t5-3b*	0.6180	0.6553	0.6921	0.5411	0.5708	0.6537	0.5669	0.5759	0.6614
GPT-3.5-turbo*	0.6840	0.6695	0.6470	0.5586	0.5592	0.6080	0.6814	0.6542	0.6812
ChatGLM-Pro*	0.6553	0.7033	0.6951	0.6485	0.6887	0.7042	0.6001	0.6497	0.7412
GPT-4*	0.6893	0.7005	0.7369	0.6330	0.6801	$0.7815^{\dagger\dagger}$	0.6732	0.6752	0.8088
PandaLM	/	/	0.6350	/	/	0.7205	/	/	0.7039
DeepSeek-R1-0528	0.6809	0.7131	0.7119	0.6589	0.7095	0.7159	0.5923	0.6668	0.7742
ChatEval	0.5694	0.5747	0.6584	0.6009	0.6435	0.7366	0.6080	0.6725	0.6820
PRE (w/o Filter)	0.7055	0.7002	0.7401	0.6804	0.6711	0.7542	0.7258	0.7295	0.7413
PRE (Auto-Exam)	0.7064	0.7133	0.7381	0.6795	0.6905	0.7664	0.7248	0.7129	0.8048
PRE (Human Filter)	$0.7211^{\dagger\dagger}$	0.7192	0.7423	0.6824	$0.7104^{\dagger\dagger}$	$0.7801^{\dagger\dagger}$	0.7255	$0.7318^{\dagger\dagger}$	0.8085
Auto-PRE (ours)	$\mathbf{0.7231}^{\dagger\dagger}$	$\mathbf{0.7195}^{\dagger}$	$\mathbf{0.7462}^{\dagger\dagger}$	$\mathbf{0.6887}^{\dagger\dagger}$	$\mathbf{0.7146}^{\dagger\dagger}$	$\mathbf{0.7821}^{\dagger\dagger}$	$\mathbf{0.7305}^{\dagger}$	$\mathbf{0.7469}^{\dagger\dagger}$	$\mathbf{0.8161}^{\dagger\dagger}$

Table 3: The overall performance (accuracy) of Auto-PRE and other baselines. The best result is highlighted in bold. †/†† indicates p-value of paired sample t-test where the method outperforms PRE (Auto-Exam) is less than 0.05/0.01.

evaluation method. More dataset details can be found in Appendix Section 1.

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. We use accuracy as the main evaluation metric, defined as the agreement rate between the manual preference annotations and the method's evaluation results. When using pointwise evaluation, we will convert each answer's individual scores into pairwise rankings. In addition, for pointwise, we also use Spearman correlation coefficient (Lehman et al. 2013) (S) to measure the consistency between the model's outputs and manual annotations. We calculate S for each instance of the task, and report the mean of them as the overall performance.

Baselines

We compare our Auto-PRE with different baselines:

1. ROUGE-L and BERTScore: These are widely used reference-based metrics (not applicable to the NF_CATS due to the lack of reference answers). For BERTScore, we use deberta-xlarge-mnli (He et al. 2020) as the base model. We report the F1 score for both metrics.

2. GPTScore (Fu et al. 2023): This metric assesses the quality of generated text by computing the log-probability of the output given the prompt under a specific LLM. We use text-davinci-003 (Brown et al. 2020) as the base model.

3. Single LLM: Use a single general LLM as evaluator. The LLMs used are marked with an asterisk (*) in Table 3.

4. PandaLM (Wang et al. 2024): A fine-tuned variant of LLaMA-7B specifically designed for preference judgment.

5. Deepseek-R1-250528 (Guo et al. 2025): A strong representative of recently popular reasoning-specialized LLMs.

6. ChatEval (Chan et al. 2024): Use two GPT-3.5-turbo to build two agents as evaluators to debate in two rounds with a one-by-one communication strategy.

7. PRE (Human Filter): Use all 7 LLMs in Single LLM as candidates and select evaluators based on human annotations. Unless noted otherwise, PRE refers to this version.

8. PRE (w/o Filter): Use all candidates as evaluators without any filtering.

9. PRE (Auto-Exam): The original PRE offers a preliminary and simplified version of the automatic qualification exam, using only consistency for selection.

Implementation Details

We set *temperature* to 0 and *do_sample* to False for reproducibility. In our Auto-PRE, η_c and η_p are respectively defined as the mean P_c and P_p across all candidate LLMs. The fusion weight for each evaluator LLM is the average of P_c, P_p, and P_s. Evaluation prompts and hyperparameter details are in Appendix Sections 2 and 5, respectively.

Experimental Results

In this section, we present the experimental results and aim to address the following four research questions (RQs):

1. How does the performance of Auto-PRE compare to other baseline methods?
2. Does Auto-PRE help mitigate the systematic bias caused by relying on LLMs from the same series?
3. What are the benefits of Auto-PRE in reducing costs?
4. How do the three selection methods in Auto-PRE interact and complement each other?

Methods	Xsum		NF_CATS		DailyDialog	
ROUGE-L	0.2329		/		0.4057	
BertScore	0.2715		/		0.5137	
GPTScore	0.4203		0.1966		0.4589	
	5-l	100-l	5-l	100-l	5-l	100-l
GPT-4	0.4801	0.4701	0.3318	0.3287	0.5044	0.4684
ChatEval	0.1767	0.1942	0.2142	0.2560	0.3865	0.4333
PRE (w/o Filter)	0.4650	0.4312	0.3555	0.3103	0.5297	0.5294
PRE (Auto-Exam)	0.4809	0.4595	0.3729	0.3515	0.5342	0.5229
PRE (Human Filter)	0.4991	0.4633	**0.3939**	0.3706	0.5347	0.5547
Auto-PRE	**0.5087**	**0.4937**	0.3931	**0.3818**	**0.5382**	**0.5599**

Table 4: The spearman correlation coefficient (S) of Auto-PRE and other baselines. Here, 5-l and 100-l denote 5-level and 100-level annotations, respectively.

Main Results (RQ1)

Table 3 and 4 show the overall results, leading to the following observations:

Across various settings, Auto-PRE consistently outperforms existing methods and achieves the best overall performance on average. In contrast, reference-based metrics such as ROUGE-L and BERTScore exhibit significant performance gaps relative to the top-performing methods, while GPTScore also falls notably behind. PandaLM performs reasonably well, but only on the NF_CATS. Deepseek-R1-0528 demonstrates strong competitiveness, particularly under the pointwise format, highlighting the potential of reasoning LLMs for evaluation tasks. Additionally, the comparison with ChatEval in Table 3 is not entirely fair (as we don't use GPT-4 as the base for the agent due to cost considerations), but we will provide a detailed comparison of the performance between Auto-PRE and ChatEval under equivalent cost conditions in the cost analysis section.

Compared to GPT-4, Auto-PRE exhibits significant improvements in pointwise format, with an average increase of 4.53% in accuracy. In terms of pairwise, Auto-PRE performs comparably with GPT-4. Moreover, we believe that GPT-4's systemic bias may not be apparent in the context of the overall dataset, as 70% of the answer pairs in our experimental setup do not include LLMs from the GPT series. In the bias analysis section, we will test on instances containing LLMs from the GPT series to further discuss the advantages of Auto-PRE in mitigating bias compared to GPT-4.

Compared to PRE (w/o Filter) and PRE (Auto-Exam), Auto-PRE achieves significantly better performance while keeping low costs, with an average improvement of 2.44% in accuracy and 0.0325 in S over PRE (w/o Filter), and an average improvement of 1.45% in accuracy and 0.0256 in S over PRE (Auto-Exam). This underscores the necessity of our more well-designed qualification exam.

Compared to PRE (Human Filter), Auto-PRE achieves comparable performance while significantly reducing costs, which will be discussed in the cost analysis section. Interestingly, Auto-PRE even outperforms PRE (Human Filter) to some extent, which might be because of the more comprehensive coverage of different judgment stages in our automatic qualification exam. In contrast, while manual annotation-based filtering emphasizes evaluator accuracy, it does not fully account for all judgment stages. This will be discussed in detail in the ablation study section.

Methods	Xsum		NF_CATS	
	accuracy	rate (%)	accuracy	rate (%)
GPT-4	0.5366	**83.64**	0.8618	**87.88**
PRE (w/o Filter)	0.5610	68.18	0.8553	84.85
PRE (Auto-Exam)	0.5671	65.45	0.8750	86.36
PRE (Human Filter)	0.5549	77.27	**0.8816***	71.21
Auto-PRE	**0.5854***	65.45	**0.8816***	74.24

Table 5: The bias analysis of GPT-4 (pairwise). * indicates p-value of paired sample t-test, where the method outperforms GPT-4 is less than 0.05.

Bias Analysis (RQ2)

To demonstrate that Auto-PRE effectively reduces bias in single-evaluator LLM (e.g., GPT-4), we conduct experiments using GPT-3.5-turbo and ChatGLM2-6B as two answer generators, tested on the Xsum and NF_CATS datasets in the pairwise evaluation format.

Table 5 presents the results. As Equation 5 shows, $rate$ is defined as the proportion of instances where, despite human annotators judging the two LLMs as tied or preferring ChatGLM2-6B, the method still favors GPT-3.5-turbo.

$$rate_m = \frac{\sum \mathbb{I}(T_h \in \{0, -1\} \wedge T_m = 1)}{\sum \mathbb{I}(T_h \in \{0, -1\})} \times 100\%, \quad (5)$$

where T_h and T_m denote human and method preferences, respectively; 1 indicates a preference for GPT-3.5-turbo, -1 for ChatGLM2-6B, and 0 for a tie.

From the results, we can observe a significant performance gap between GPT-4 and Auto-PRE, with an average difference of 3.43% in accuracy. Furthermore, GPT-4 demonstrates a notably higher $rate$ compared to Auto-PRE, with an average disparity of 15.92% in $rate$. This suggests that GPT-4's preference for GPT-3.5-turbo could compromise its performance and reliability. In contrast, Auto-PRE improves both overall performance and reliability by effectively leveraging collaboration across diverse LLMs.

Cost Analysis (RQ3)

In this section, we analyze the cost-effectiveness of Auto-PRE and compare its performance with ChatEval under equivalent costs. To achieve this, we implement several variants of the above methods. ChatEval includes three variants (C1, C2, C3), while Auto-PRE includes five variants (A1, A2, A3, A4, A5). Their detailed configurations are provided in Appendix Section 4. We use pairwise as the evaluation format on Xsum and NF_CATS. Each task has 4200 instances, and each instance has about 1K tokens, so completing each task requires approximately 4.2 M tokens. Based on the official pricing released (AI 2025; OpenAI 2025), the

Methods	Xsum	NF_CATS	DailyDialog
	acc [pass]	acc [pass]	acc [pass]
PRE (Auto-Exam)	0.7381 [4,5,6,7]	0.7664 [5,6,7]	0.8048 [4,5,6,7]
Auto-PRE (P)	0.7379 [2,4,5,6,7]	0.7702 [4,5,6,7]	0.8065 [3,4,5,6,7]
Auto-PRE (S)	0.7398 [3,4,5,6,7]	0.7658 [3,4,6,7]	0.7900 [2,5,6,7]
PRE (Human Filter)	0.7423 [3,4,5,6,7]	0.7801†† [4,5,6,7]	0.8085 [5,6,7]
Auto-PRE (C+P+S)	**0.7462**†† [4,5,6,7]	**0.7821**†† [6,7]	**0.8161**†† [5,6,7]

Table 6: The performance (accuracy) of different Auto-PRE variants (pairwise). The meaning of †† is the same as in Table 3. 'pass' records the qualified LLMs in specific exams (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).

costs of ChatGLM_Pro and GPT-3.5-turbo are estimated to be similar at $1 per million tokens. The cost of GPT-4 is estimated at $40 per million tokens. Open-source LLMs 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 automatic qualification exam (less than $1) can be neglected compared to the total costs. The GPU cost is also negligible: processing 4.2M tokens with a 13B open model requires approximately 0.12 GPU-hours on an A100, i.e., about $0.5.

Detailed experimental results, illustrating the relationship between total cost and accuracy, are provided in Appendix (Figures 3 and 5). The results show that at the same cost, our Auto-PRE (A1, A2, A3, A4, A5) can achieve higher performance than all baselines, including ChatEval (C1, C2, C3). Compared to PRE (Human Filter), our methods significantly reduce costs (about $115) while maintaining performance without notable differences. Compared to GPT-4, our methods can reduce costs by 90%, while keeping nearly the same performance (with only a 0.54% decrease in accuracy). Overall, our methods can achieve performance comparable to state-of-the-art methods at a much lower cost.

Ablation Studies (RQ4)

In this section, we delve into the contributions of each selection method of Auto-PRE to the overall performance. We explore the performance of several variants of Auto-PRE that combine different selection methods on three tasks in the pairwise task format, including PRE (Auto-Exam): using only Consistency; Auto-PRE (P): using only Pertinence; Auto-PRE (S): using only Self-Confidence; and Auto-PRE (C+P+S): using all three methods.

The results in Table 6 indicate that Auto-PRE (C+P+S), which integrates all three selection methods, achieves the best performance, significantly outperforming PRE (Auto-Exam), Auto-PRE (P), and Auto-PRE (S) that use only a single selection method, with an average improvement of 1.33% in accuracy. This demonstrates that the three selection methods exhibit a synergistic effect, complementing each other to enhance overall performance.

Interestingly, we observe Auto-PRE (C+P+S) even outperforms PRE (Human Filter) to some extent, showing the potential of our automatic methods. Taking NF_CATS as an example, in our automatic qualification exam, GPT-3.5-turbo is filtered out in the self-confidence test due to exhibiting unreasonable confidence levels. In contrast, this issue is not detected by the manual annotation-based qualification exam. This suggests our method covers a broader range of judgment stages: it not only considers the model's unbiased understanding of the judgment instruction but also emphasizes its insightful comprehension of the judgment content and its reasonable self-confidence level after generating a judgment response. In comparison, the manual annotation-based qualification exam places greater emphasis on the accuracy of evaluation results while neglecting aspects such as the judge's self-confidence in its generated responses.

Conclusion

This paper develops the Auto-PRE by designing an automatic qualification exam based on three characteristics: (1) Consistency, (2) Pertinence, (3) Self-Confidence extracted from different judgment stages. Experimental results indicate that Auto-PRE achieves state-of-the-art performance while significantly reducing cost. By providing a scalable and efficient qualification exam, our work lays a foundation for automating the evaluation of LLMs-as-judges and improving the reliability of LLM-based evaluation methods.

Acknowledgments

We thank Ant Group and our Ant Group co-authors for their support and contributions.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI, Z. 2025. The Price of ChatGLM-Pro. https://open.bigmodel.cn/pricing.

Bolotova, V.; Blinov, V.; Scholer, F.; Croft, W. B.; and Sanderson, M. 2022. A non-factoid question-answering taxonomy. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1196–1207.

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

Chan, C.-M.; Chen, W.; Su, Y.; Yu, J.; Xue, W.; Zhang, S.; Fu, J.; and Liu, Z. 2024. ChatEval: Towards Better LLM-Based Evaluators through Multi-Agent Debate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3): 1–45.

Chu, Z.; Ai, Q.; Tu, Y.; Li, H.; and Liu, Y. 2024. PRE: A Peer Review Based Large Language Model Evaluator. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*. ACM.

Duan, J.; Cheng, H.; Wang, S.; Wang, C.; Zavalny, A.; Xu, R.; Kailkhura, B.; and Xu, K. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*.

Fu, J.; Ng, S.-K.; Jiang, Z.; and Liu, P. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

He, P.; Liu, X.; Gao, J.; and Chen, W. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Lehman, A.; O'Rourke, N.; Hatcher, L.; and Stepanski, E. 2013. *JMP for basic univariate and multivariate statistics: methods for researchers and social scientists*. Sas Institute.

Li, H.; Chen, J.; Ai, Q.; Chu, Z.; Zhou, Y.; Dong, Q.; and Liu, Y. 2024. Calibraeval: Calibrating prediction distribution to mitigate selection bias in llms-as-judges. *arXiv preprint arXiv:2410.15393*.

Li, Y.; Su, H.; Shen, X.; Li, W.; Cao, Z.; and Niu, S. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

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

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Manakul, P.; Liusie, A.; and Gales, M. J. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Narayan, S.; Cohen, S. B.; and Lapata, M. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1797–1807. Brussels, Belgium: Association for Computational Linguistics.

OpenAI. 2025. The Price of GPT-3.5-turbo. https://openai.com/api/pricing/.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Wang, P.; Li, L.; Chen, L.; Cai, Z.; Zhu, D.; Lin, B.; Cao, Y.; Liu, Q.; Liu, T.; and Sui, Z. 2023. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Wang, Y.; Yu, Z.; Zeng, Z.; Yang, L.; Wang, C.; Chen, H.; Jiang, C.; Xie, R.; Wang, J.; Xie, X.; Ye, W.; Zhang, S.; and Zhang, Y. 2024. PandaLM: An Automatic Evaluation Benchmark for LLM Instruction Tuning Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zeng, Z.; Yu, J.; Gao, T.; Meng, Y.; Goyal, T.; and Chen, D. 2024. Evaluating Large Language Models at Evaluating Instruction Following. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zhang, X.; Yu, B.; Yu, H.; Lv, Y.; Liu, T.; Huang, F.; Xu, H.; and Li, Y. 2023. Wider and deeper llm networks are fairer llm evaluators. *arXiv preprint arXiv:2308.01862*.

Zhao, Y. W.; Chi, C.-H.; and van den Heuvel, W.-J. 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.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J.; and Stoica, I. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *ArXiv*, abs/2306.05685.