LLM-as-a-judge

Mokrov Semen
Amsterdam University of Applied Science
Amsterdam, The Netherlands
semen.mokrov@gmail.com

Deep Kayal
Amazon Research
London, The United Kingdom
_

Riccardo Pinosio
Amsterdam University of Applied Science
Amsterdam, The Netherlands
_

Debarati Bhaumik
Amsterdam University of Applied Science
Amsterdam, The Netherlands
_

Abstract

Here the abstract

ACM Reference Format:
Mokrov Semen, Deep Kayal, Riccardo Pinosio, and Debarati Bhaumik. 2025. LLM-as-a-judge. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Here the introduction

2 Related Work

2.1 Large Language Models as Evaluators

2.1.1 Introduction to LLM Evaluators Large Language Models (LLMs) are increasingly being used as automated evaluators for open-ended text generation tasks, providing a scalable alternative to slow and costly human judgment. Recent benchmarks and leaderboards for chatbots and text generation often rely on powerful LLMs (such as GPT-4) to rank or score model outputs in place of humans. For instance, the Vicuna Chatbot evaluation pipeline employs GPT-4 to compare responses in a pairwise manner, achieving evaluation agreements on par with human annotators on many queries. Researchers have even built interactive frameworks like *EvalGen* [13] that leverage LLMs to suggest evaluation criteria and generate test queries for model outputs, with a human-in-the-loop to refine these criteria. These approaches demonstrated the potential of LLM-as-a-judge systems: a strong LLM can approximate human evaluation reasonably well (e.g., GPT-4's judgments agree with human preferences roughly 75–80% of the time on multi-turn dialog tasks) [20]. As a result, LLM evaluators have quickly become a central tool for comparing models and tuning them to human-like performance.

2.1.2 Evaluation Paradigms LLM-as-a-Judge evaluations mostly employ two paradigms: pointwise and pairwise evaluations.

Pointwise Evaluation In pointwise evaluation, an LLM assesses a single response independently, assigning a score based on predefined criteria such as relevance, coherence, or factual accuracy. This method is straightforward and allows for fine-grained analysis of individual responses. However, it may suffer from inconsistencies due to the subjective nature of scoring and the lack of comparative context [13].

Pairwise Evaluation Pairwise evaluation involves presenting the LLM with two responses to the same prompt and asking it to determine which one is better according to specific criteria [11]. This comparative approach can mitigate some biases inherent in pointwise evaluations and often aligns more closely with human judgment. In score-based pairwise comparison, the LLM assigns each response a numerical score based on evaluation criteria such as coherence, helpfulness, or relevance, and then ranks them by comparing the scores [10]. This method allows for fine-grained differentiation between responses while retaining the benefits of comparative judgment.

2.1.3 Benchmarks Utilizing LLM-as-a-Judge Several benchmarks have been developed to standardize the evaluation of LLM outputs using the LLM-as-a-Judge paradigm.

MT-Bench MT-Bench is a multi-turn benchmark designed to assess the conversational abilities of LLMs. It evaluates models on their capacity to maintain context, follow instructions, and provide informative responses over multiple dialogue turns. MT-Bench employs GPT-4 as the judge, leveraging its advanced understanding to score model responses. The benchmark includes an Elo rating system derived from pairwise comparisons, offering a dynamic leaderboard that reflects the relative performance of various models[20].

AlpacaEval AlpacaEval focuses on single-turn instruction-following tasks, evaluating models based on their ability to generate helpful and accurate responses. It uses GPT-4 as an automatic evaluator to compare model outputs against reference responses. AlpacaEval provides win-rate statistics and supports a leaderboard that ranks models accordingly. The benchmark has been validated against a substantial set of human annotations, ensuring its reliability and relevance[6].

Chatbot Arena Chatbot Arena is a platform that combines human and LLM-based evaluations to rank conversational agents. It utilizes pairwise comparisons, where human annotators or LLM judges select the better response between two options. The results contribute to an Elo rating system, providing a comprehensive

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

leaderboard that reflects both human and automated assessments. Chatbot Arena serves as a valuable resource for understanding the comparative performance of various chatbots in real-world scenarios[20].

2.2 Problems in Current Systems

2.2.1 Position Bias Position bias refers to the tendency of LLM-based evaluators to favor responses based on their position in a prompt during pairwise comparisons. Studies have shown that models like GPT-4 often prefer the first response presented, regardless of content quality[14]. This bias can compromise the fairness and reliability of evaluations. To address this, frameworks such as PORTIA have been developed, which align similar content across candidate answers to mitigate position bias effectively[7].

2.2.2 Length Bias Length bias occurs when LLM evaluators disproportionately favor longer responses, associating verbosity with higher quality. This can lead to inflated evaluations for unnecessarily lengthy outputs. Research has demonstrated that longer responses often receive higher preference scores, even when shorter responses are equally or more informative[3].

2.2.3 Cost Problem Employing state-of-the-art large language models (LLMs) as evaluators in LLM-as-a-Judge frameworks introduces significant computational costs. High-performance models like GPT-4 entail substantial expenses per token, rendering large-scale evaluations financially heavy. Simpler queries are handled by smaller, cost-effective models, while more complex tasks are escalated to larger, more expensive models. This strategy can significantly reduce overall evaluation costs without compromising accuracy[1]. At the same time, ARJudge optimizes the trade-off between evaluation quality and computational expense by dynamically allocating resources based on the prompt characteristics[18].

2.3 Established Techniques for Biases Mitigation

This chapter presents established techniques for mitigating position bias, length bias, and cost-related issues.

PORTIA (Position-Oriented Response Transformation and Input Alignment) is an alignment-based system designed to mimic human comparison strategies to calibrate position bias in a lightweight yet effective manner. Specifically, PORTIA splits the answers into multiple segments, aligns similar content across candidate answers, and then merges them back into a single prompt for evaluation by LLMs. Experiments demonstrate that PORTIA markedly enhances the consistency rates for various models and comparison forms, achieving an average relative improvement of 47.46%. Furthermore, it rectifies around 80% of the position bias instances within the GPT-4 model, elevating its consistency rate up to 98%. However, PORTIA is not effective when the LLM evaluator refuses to provide a judgment, often seen in advanced models like GPT-3.5 and GPT-4 in sensitive domains (e.g., roleplay tasks). Additionally, it requires the merged input to fit within the model's context window, and it's not reproducible since it doesn't provide a working codebase at the time of writing[8].

FairEval introduces a calibration framework to address positional bias in LLM evaluations. Extensive experiments demonstrate that FairEval successfully alleviates evaluation bias, achieving a closer alignment with human judgment by improving consistency and reducing positional artifacts in comparative outputs. Specifically, the combined calibration strategy using all the components enhances the accuracy of GPT-4 and ChatGPT's evaluations by up to 14.3% and 26.9%, respectively, with a kappa correlation improvement of 0.25 and 0.46. However, despite its effectiveness, FairEval's reliance on multiple API calls and partial human intervention may lead to increased computational costs and scalability limitations when applied to large-scale model comparisons[15].

LLM-JudgeEval introduces a systematic framework for evaluating the reliability and bias of LLM-as-a-Judge methods by defining theoretically grounded metrics for accuracy, position bias, and length bias, and mitigating self-inconsistency. The authors propose interpretable formulations of position and length biases and quantify internal inconsistencies through a flipping noise model. However, while the study defines metrics to measure position and length bias, it does not propose any method to mitigate these biases, instead emphasizing their presence. Thus, this approach lies between a benchmark and a bias-solving solution[17].

G-EVAL proposes a novel evaluation framework that leverages GPT-4 in a form-filling paradigm with chain-of-thought prompting to assess NLG outputs on different dimensions. The system auto-generates step-by-step evaluation instructions and computes continuous quality scores by reweighting discrete ratings with token-level output probabilities. G-EVAL-4 achieves the highest human correlation across all benchmarks, with a Spearman correlation of *0.514* on SummEval and *0.599* on QAGS, outperforming state-of-the-art metrics such as UniEval and GPTScore. However, it reveals a systematic bias in favor of self-generated outputs[9].

ZEPO is a meta-evaluation framework designed to improve human alignment in LLM-as-a-judge evaluations by directly targeting preference bias, including position bias, through zero-shot prompt optimization. The authors did not limit their focus to the LLM-as-a-judge problem; rather, the proposed framework is designed to enhance the evaluation capabilities of LLMs across all types of responses, including those generated by humans. Rather than relying on labeled data or manual prompt design, ZEPO formulates a zero-shot fairness objective based on the uniform distribution of LLM preferences. It uses an LLM-based optimizer to iteratively paraphrase prompts, searching for those that yield the fairest (most balanced) decisions. Experimental results show that ZEPO achieves up to 29% relative gains in Spearman correlation with human judgment over strong pairwise baselines in terms of coherence, relevancy, informativeness, etc. When combined with position debias (e.g., Balanced Position Calibration[16]), ZEPO further improves performance, achieving up to 8% improvement in Spearman correlation comparing to the pairwise comparison combined with the debias technique. However, the current implementation uses a basic greedy search with GPT-3.5 for prompt optimization, leaving room for performance gains through more advanced search techniques and optimizer models[21].

LC AlpacaEval addresses length bias issue by training a logistic regression model on prior annotations from AlpacaEval, factoring in model identity, prompt length, and instruction difficulty. By neutralizing the response length component during score prediction, this method improves evaluation fairness. As a result, the automatic

evaluation match human opinions better (raising the Spearman correlation with Chatbot Arena from 0.94 to 0.98). Moreover, their system is more robust to the fluctuations in the win rate: before, models could change their win rate by as much as 41.4 percentage points just by changing how long their answers were; now, it is 9.7. Also, during adversarial attack (attempt to deceive the system by purpose), the win rate only goes up by 8.5 points, instead of 22.2. However, LC AlpacaEval's performance is bounded by the scope of its training data; it struggles to generalize effectively to more complex multi-turn tasks and open-ended domains, where human judgment nuances are more difficult to approximate[2].

AdapAlpaca tackles the length bias problem by adjusting the evaluation setup itself rather than correcting scores post hoc. Specifically, it ensures that each model response is compared to a reference response of similar length, by dividing the word count range into intervals (e.g., 0–200, 200–400 words) and prompting GPT-4 to generate responses within those ranges. This approach works under the assumption that the final response quality that a judge assign to an answer consist of two elements: **desirability** (length-independent) and **information mass** (length-dependent). The presented benchmark shows that the average difference between its win rates and human preferences is just *0.99%*, compared to *+24.35%* for concise responses and *+4.22%* for detailed responses under the standard AlpacaEval metric. However, its reliance on fixed-length generation assumes that quality can be preserved across word counts. Moreover, the method still depends on GPT-4 for reference generation and annotation[4].

Tuned LLM-judges introduces a multi-fidelity, multi-objective framework to systematically optimize hyperparameters of zero-shot LLM judges, including *model choice*, *prompt design*, *output format*, and *inference settings*. 4480 judge configurations were evaluated using open-weight models and progressively filtered through a three-stage evaluation, reducing costs by leveraging **human agreement** as a cheaper yet effective metric. Their best-performing judge achieves a human agreement score of *0.49* at the lower cost on the LMSys dataset, outperforming JudgeLM and PandaLM, and corresponding Arena-Hard's accuracy. On the Arena-Hard benchmark, the tuned judge achieves a Spearman correlation of *0.93*, surpassing even GPT-4 and Claude-based judges in ranking alignment. Although, the introduced method still illustrates LLM judge limitations such as prompt sensitivity, stylistic bias, and variability across model sizes[12].

CascadeEval proposes a cascade-based evaluation mechanism with early abstention, allowing smaller, cheaper LLMs to abstain from answering difficult queries and defer them to stronger models. This leads to a 13.0% average reduction in cost and a 5.0% drop in error rates across six benchmarks, with only a 4.1% increase in abstention rates, demonstrating strong performance-cost trade-offs. However, the method requires careful calibration of confidence thresholds and may depend on benchmark-specific tuning[5].

ARJudge introduces a robust, multi-faceted evaluation framework that combines adaptive criteria generation with both text-based and code-driven analysis to assess LLM outputs. Fine-tuned on the Composite Analysis Corpus, ARJudge achieves strong performance across benchmarks like PandaLM Eval and MTBench, with an average accuracy of 77.7% and even outperforming tuning-free models like Qwen2.5-7B by up to 26.7% on challenging datasets such as LLMBar. However, ARJudge still relies on a moderately large model (Qwen2.5-7B) and incurs non-trivial fine-tuning costs, limiting its accessibility for lightweight deployment. Additionally, its performance gains rely heavily on code execution capabilities, which may not be feasible in constrained environments[18].

LLM Cascades with Mixture of Thought Representations introduce a cost-efficient evaluation framework that strategically combines weak and strong LLMs using answer consistency as a routing signal. The system leverages weaker models like GPT-3.5 to produce multiple intermediate reasoning traces—such as Chain-of-Thought (CoT) and Program-of-Thought (PoT)—and accepts their answers when the outputs are sufficiently consistent. Experiments on six reasoning benchmarks show that this method maintains accuracy close to full GPT-4 evaluation while reducing total cost by up to 60%. However, the approach assumes that consistency among weaker LLMs correlates with correctness, which may not always hold in complex or adversarial tasks. Moreover, all the experiments were aimed to testing the capabilities of the system, but not in terms of LLM-as-a-judge [19].

2.4 Final Comparison

Table 1 summarizes the key characteristics of the reviewed approaches with respect to the bias they address, their core methodology, and notable limitations.

Table 1: Comparison of mitigation approaches across evaluation biases

Approach	Mitigation Target	Core Technique	Limitations
PORTIA	Position bias	Input alignment with merged prompt	Not robust when LLM refuses judgment; no codebase; limited by context window
FairEval	Position bias	Score calibration using contextual references	High API usage; partial human dependency; limited scalability
LLM-JudgeEval	Position bias & Length bias	Definitions of accuracy and bias metrics and their calculations	The mitigation approach isn't introduced; the framework only emphasis the biases
ZEPO	Not specified	Zero-shot prompt optimization for fairness-based preference calibration	Basic optimizer limits performance gains
G-Eval	Not specified	Usage of chain-of-thought prompting and a form-filling paradigm	Favor bias of self-generated outputs
LC AlpacaEval	Length bias	Logistic regression debiasing verbosity artifacts	Annotator dependency; the same quality assumption
AdapAlpaca	Length bias	Match the model's reply with the similar size reference	Fixed-length responses must preserve quality, relies on the state-of-art-model as annotator
CascadeEval	Cost problem	Cascaded abstention with fallback to strong models	Needs careful confidence threshold tuning; benchmark-dependent
Tuned LLM-judges	Cost problem	Multi-fidelity search over 4,480 judge configurations using human-agreement	Prompt sensitivity, stylistic bias, model sizes variability
ARJudge	Cost problem	Code-based adaptive prompting	Fine-tuning and larger model needed; less cost-efficient and portable
LLM Cascades + MoT	Cost problem	Consistency check across CoT/PoT reasoning	Assumes agreement = correctness; may fail on adversarial examples

As observed, current solutions are optimized for specific bias types but fall short of offering a unified remedy. PORTIA and SOD address position bias but suffer from deployment or scalability limitations. LC AlpacaEval effectively mitigates length bias but lacks adaptability for multi-turn scenarios. CascadeEval emerges as a practical framework for reducing evaluation cost with minimal performance degradation but does not address other biases.

Given these insights, the goal of the present study is to design a new evaluation framework that addresses all three issues—position

bias, length bias, and cost—by integrating the core strengths of **CascadeEval**, **FairEval**, and **LC AlpacaEval**.

References

[1] Lingjiao Chen, Matei Zaharia, and James Zou. 2023. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. arXiv:2305.05176 [cs.LG] https://arxiv.org/abs/2305.05176

[2] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2025. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators. arXiv:2404.04475 [cs.LG] https://arxiv.org/abs/2404.04475

[3] Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. 2024. Explaining Length Bias in LLM-Based Preference Evaluations. arXiv:2407.01085 [cs.LG] https://arxiv.org/abs/2407.01085

[4] Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. 2024. Explaining Length Bias in LLM-Based Preference Evaluations. arXiv:2407.01085 [cs.LG] https://arxiv.org/abs/2407.01085

[5] Hui Huang, Yingqi Qu, Xingyuan Bu, Hongli Zhou, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. 2024. An Empirical Study of LLM-as-a-Judge for LLM Evaluation: Fine-tuned Judge Model is not a General Substitute for GPT-4. arXiv:2403.02839 [cs.CL] https://arxiv.org/abs/2403.02839

[6] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaEval: An Automatic Evaluator of Instruction-following Models. https://github.com/tatsu-lab/alpaca_eval.

[7] Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2024. Split and Merge: Aligning Position Biases in LLM-based Evaluators. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 11084–11108. doi:10.18653/v1/2024.emnlp-main.621

[8] Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2024. Split and Merge: Aligning Position Biases in LLM-based Evaluators. arXiv:2310.01432 [cs.CL] https://arxiv.org/abs/2310.01432

[9] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. arXiv:2303.16634 [cs.CL] https://arxiv.org/abs/2303.16634

[10] Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2024. Aligning with Human Judgement: The Role of Pairwise Preference in Large Language Model Evaluators. In *First Conference on Language Modeling*. https://openreview.net/forum?id=9gdZI7c6yr

[11] Adian Liusie, Potsawee Manakul, and Mark J. F. Gales. 2024. LLM Comparative Assessment: Zero-shot NLG Evaluation through Pairwise Comparisons using Large Language Models. arXiv:2307.07889 [cs.CL] https://arxiv.org/abs/2307.07889

[12] David Salinas, Omar Swelam, and Frank Hutter. 2025. Tuning LLM Judge Design Decisions for 1/1000 of the Cost. arXiv:2501.17178 [cs.CL] https://arxiv.org/abs/2501.17178

[13] Shreya Shankar, J. D. Zamfirescu-Pereira, Björn Hartmann, Aditya G. Parameswaran, and Ian Arawjo. 2024. Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences. arXiv:2404.12272 [cs.HC] https://arxiv.org/abs/2404.12272

[14] Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2025. Judging the Judges: A Systematic Study of Position Bias in LLM-as-a-Judge. arXiv:2406.07791 [cs.CL] https://arxiv.org/abs/2406.07791

[15] 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:2305.17926 [cs.CL] https://arxiv.org/abs/2305.17926

[16] 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:2305.17926 [cs.CL] https://arxiv.org/abs/2305.17926

[17] Hui Wei, Shenghua He, Tian Xia, Fei Liu, Andy Wong, Jingyang Lin, and Mei Han. 2025. Systematic Evaluation of LLM-as-a-Judge in LLM Alignment Tasks: Explainable Metrics and Diverse Prompt Templates. arXiv:2408.13006 [cs.CL] https://arxiv.org/abs/2408.13006

[18] Kaishuai Xu, Tiezheng Yu, Wenjun Hou, Yi Cheng, Liangyou Li, Xin Jiang, Lifeng Shang, Qun Liu, and Wenjie Li. 2025. Learning to Align Multi-Faceted Evaluation: A Unified and Robust Framework. arXiv:2502.18874 [cs.CL] https://arxiv.org/abs/2502.18874

[19] Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. 2024. Large Language Model Cascades with Mixture of Thoughts Representations for Cost-efficient Reasoning. arXiv:2310.03094 [cs.CL] https://arxiv.org/abs/2310.03094

[20] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. arXiv:2306.05685 [cs.CL] https://arxiv.org/abs/2306.05685

[21] Han Zhou, Xingchen Wan, Yinhong Liu, Nigel Collier, Ivan Vulić, and Anna Korhonen. 2024. Fairer Preferences Elicit Improved Human-Aligned Large Language Model Judgments. arXiv:2406.11370 [cs.CL] https://arxiv.org/abs/2406.11370