Efficient Prompt Optimization for Comparative LLM-as-a-judge through Uncertainty Estimation

Yassir Fathullah
yf286@cam.ac.uk
University of Cambridge
Cambridge, United Kingdom

Mark J. F. Gales
mjfg@eng.cam.ac.uk
University of Cambridge
Cambridge, United Kingdom

Abstract

LLM-as-a-judge, through comparative prompting, is a powerful approach for Natural Language Generation evaluation. However, its quadratic computational cost makes iterative prompt optimization expensive. Instead, we propose leveraging uncertainty to select and re-evaluate only the most uncertain pairwise comparisons. Our framework significantly reduces the computational costs of iterative prompt optimization. Experiments on the SummEval dataset demonstrate that this approach can achieve up to 80% reduction in re-evaluation costs while maintaining or exceeding performance.

CCS Concepts

• **Computing methodologies** → **Discrete space search**; *Information extraction*; Natural language generation; • **Mathematics of computing** → Probabilistic inference problems.

Keywords

LLM–as-a-judge, Bradley-Terry, Ranking, Prompt Optimization

ACM Reference Format:
Yassir Fathullah and Mark J. F. Gales. 2025. Efficient Prompt Optimization for Comparative LLM-as-a-judge through Uncertainty Estimation . In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (KDD Workshop on Prompt Optimization).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Instruction-tuned Large Language Models (LLMs) have shown impressive zero-shot capabilities across a wide range of natural language processing and generation tasks [1, 5, 9, 11, 24, 36]. This has led to their increasing use as automated evaluators, or "LLM-as-a-judge," for language tasks, offering a scalable alternative to costly human judgments [6, 8, 20, 31, 35].

Within the LLM-as-a-judge paradigm, two main approaches exist: absolute scoring (assigning a numerical score to a single response) and comparative scoring (pairwise comparison of two responses) [18, 20, 35]. While absolute scoring is computationally efficient (requiring $O(N)$ LLM calls for N responses), its scores can be inconsistent and sensitive to prompt wording [18, 35]. Comparative scoring, in contrast, has shown higher correlations with human judgments and more reliable rankings. However, its primary drawback is the computational cost, which scales quadratically ($O(N^2)$) with the number of responses, making it prohibitively expensive when comparing many responses [21].

Furthermore, the efficacy of an LLM-as-a-judge, regardless of the scoring method, relies on well-designed prompts. These prompts typically include system-level instructions and task-specific instructions [7, 38]. Recent work like Optimization by PROmpting (OPRO) has shown that LLMs can also be used to iteratively refine and discover more effective prompts by generating new prompt candidates based on the performance of previous ones [33]. However, when applying OPRO to the comparative assessment, the quadratic complexity becomes a significant bottleneck. Each proposed prompt generated during the OPRO optimization process would require a full re-evaluation of all possible comparisons to determine its quality, making iterative optimization for comparative judges expensive [13, 21]. We observe that for many prompts, a large fraction of pairwise comparisons are made with high confidence by the LLM judge, and their outcomes are unlikely to change significantly. Re-evaluating these "easy" or stable pairs offers diminishing returns for the computational cost incurred.

This paper introduces an efficient method for optimizing prompts for comparative LLM-as-a-judge. By integrating uncertainty estimation into OPRO, we selectively re-evaluate only the most uncertain pairwise comparisons when assessing new candidate prompts. Our experiments on the SummEval dataset demonstrate that this approach can reduce the computational cost of prompt re-evaluation by up to 80% while maintaining performance.

2 Related Works

Our work builds upon recent advancements in ranking from pairwise comparisons [20], LLM-based evaluation [35] and automatic prompt optimization [33].

Ranking from Pairwise Comparisons. The problem of ranking a set of items based on pairwise comparisons has a long history, particularly in fields like sports and psychometrics [3, 10, 22, 23]. Traditional methods typically operate on binary outcomes, such as which candidate won a specific comparison. A foundational approach in this area is the Bradley-Terry (BT) model which posits that the probability of item x_i beating item x_j depends on the difference in their latent "skill" or "score" $s_i - s_j$ [4]. The probability is typically modelled using the logistic function:

$$\mathsf{P}(x_i \succ x_j | s_i, s_j) = \sigma(s_i - s_j) = \frac{1}{1 + \exp(s_j - s_i)}$$

KDD Workshop on Prompt Optimization, Toronto, ON

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

Given a set of observed binary comparison outcomes, the scores $s_{1:N} = (s_1, \dots, s_N)$ of N items can be estimated by maximizing the likelihood of the observed data. Extensions like the TrueSkill model further improve the modelling through a Bayesian framework [15].

LLM-as-a-judge Comparative Assessment. Recent advances in instruction-tuned LLMs have enabled their use as evaluators for NLG outputs. Unlike traditional methods that rely on reference texts or bespoke models, LLM-as-a-judge approaches leverage the LLM's understanding of natural language instructions to assess quality zero-shot [20, 35]. Within this paradigm, comparative assessment has emerged as a robust method, where an LLM is prompted to compare two candidate texts x_i and x_j and provide a judgment on which is better according to a specified attribute [25, 27]. Crucially, LLMs can provide not just a binary outcome but also a probability p_{ij} that it thinks x_i is better than x_j:

$$p_{ij} = \mathsf{P}_{\mathsf{LLM}}(x_i \succ x_j), \quad C = (i, j, p_{ij})$$

where the indices of the texts and the outcome has been collated into a comparison C. This probabilistic output distinguishes LLM comparative assessment from traditional binary comparison settings and allows for richer modelling. The Product-of-Experts (PoE) framework provides a flexible way to combine information from multiple independent "experts" (in this case, individual pairwise comparisons) [16]. As shown by Liusie et al. [21], the PoE framework can be applied to LLM comparative assessment to estimate the scores $s_{1:N}$ from K comparisons $C_{1:K}$ where the probability density of the scores given the comparisons is modelled as:

$$\begin{aligned} \mathsf{p}(s_{1:N}|C_{1:K}) &\propto \prod_{i,j \in C_{1:K}} \mathsf{p}(s_i - s_j|C_k) && (1) \\ &\propto \prod_{i,j \in C_{1:K}} \sigma(s_i - s_j)^{p_{ij}} (1 - \sigma(s_i - s_j))^{1-p_{ij}} && (2) \end{aligned}$$

Each expert $\mathsf{p}(s_i - s_j|C_k)$ helps incorporate information in order to build up a density over the collection of scores $s_{1:N}$. This framework, a soft extension to the Bradley-Terry [4], allows for estimating scores:

$$\hat{s}_{1:N} = \arg\max_{s} \ln \mathsf{p}(s_{1:N}|C_{1:K}) \tag{3}$$

even when only a subset of all possible $N(N-1)$ comparisons is available.

Uncertainty in Comparative LLM-as-a-judge. Given that estimating scores from a partial set of comparisons inherently involves uncertainty, quantifying this uncertainty is valuable. The goal of uncertainty estimation in this context is to guide the selection of additional comparisons from the set of untested pairs, aiming to extract maximal information and efficiently improve the overall ranking accuracy [13]. Intuitively, some comparisons are less informative (e.g., comparing a clearly superior text against a clearly inferior one), while others are more likely to refine the ranking (e.g., comparing texts with similar estimated quality). Therefore, the aim is to quantify the uncertainty in untested comparisons and find the most informative pair to compare next. One metric is based on maximum variance:

$$\arg\max_{i,j} \ \mathbb{V}[s_i - s_j|C_{1:K}] \tag{4}$$

which would compare the pair with highest variance in their score difference. An alternative is identifying the pair that has the highest probability of reordering (assuming $\hat{s}_i > \hat{s}_j$):

$$\arg\max_{i,j} \ \mathsf{P}(s_i < s_j|C_{1:K}) \tag{5}$$

Both of these metrics allow one to iteratively find the most informative comparisons and reduce the number needed to robustly estimate the scores $\hat{s}_{1:N}$ and therefore the overall ranking. Since $\mathsf{p}(s_{1:N}|C_{1:K})$ is algebraically intractable, these metrics can be estimated through several approaches including Laplace's Approximation or Markov chain Monte Carlo sampling.

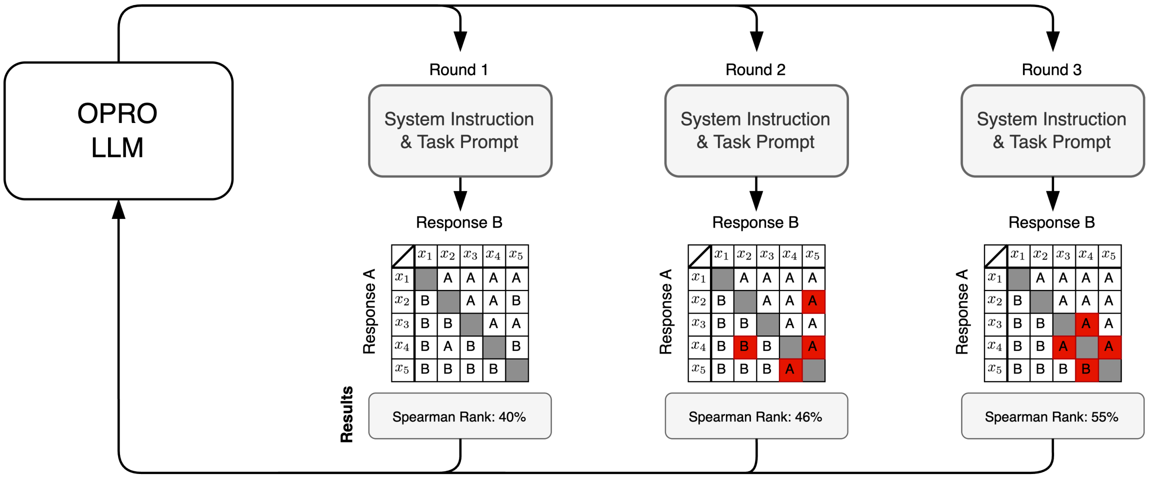
Automatic Prompt Optimization. The performance of LLMs is highly sensitive to the specific wording and structure of the input prompt [7, 29, 33, 38]. Manually engineering effective prompts can be time-consuming and requires significant expertise. Automatic prompt optimization methods aim to automate this process [33, 37]. Approaches range from gradient-based methods on continuous prompt representations to discrete search strategies [17, 32]. More recently, LLMs themselves have been used as optimizers to generate and refine prompts. Optimization by PROmpting is a notable example, where an LLM iteratively generates new prompts, evaluates their performance (e.g., task accuracy), and uses the history of prompts and their scores to guide the generation of better prompts [33]. Other related work explores similar ideas of using LLMs for prompt generation and refinement [14].

Positioning of Our Paper. While prior work has established the effectiveness of comparative LLM-as-a-judge and OPRO for prompt optimization, the integration of these concepts for efficient and optimized comparative LLM-as-a-judge remains an open challenge. The $O(N^2)$ cost of comparative assessment makes the iterative evaluation required by OPRO prohibitively expensive for large N. Our work bridges this gap by combining the PoE framework and uncertainty-based comparison selection to create an efficient OPRO optimization process. This allows us to achieve significant computational savings while simultaneously improving the performance of the LLM judge through prompt optimization, making scalable and high-quality LLM-based comparative evaluation practical.

3 Efficient Prompt Optimization

Our methodology adapts the Optimization by PROmpting (OPRO) framework to efficiently refine prompts for comparative LLM-as-a-judge systems [33]. The core idea is to iteratively improve both the system-level instructions and the task-specific prompts by using an OPRO LLM, while significantly reducing the computational cost of evaluating each new prompt candidate through uncertainty-guided selective re-evaluation. Figure 1 illustrates this iterative process.

The primary objective for the prompt optimization is to maximize the agreement between the LLM judge's predicted $\hat{s}_{1:N}$ and human-annotated ground truth $s_{1:N}$ scores. We quantify this using the Spearman Rank Correlation Coefficient. However, instead of evaluating all $N(N-1)$ possible comparisons we seek only to re-evaluate the fraction of most uncertain comparisons. Building on [13] we can re-run comparisons with the highest probability of



Round 1

	x_1	x_2	x_3	x_4	x_5
x_1		A	A	A	A
x_2	B		A	A	B
x_3	B	B		A	A
x_4	B	A	B		B
x_5	B	B	B	B	

Round 2

	x_1	x_2	x_3	x_4	x_5
x_1		A	A	A	A
x_2	B		A	A	A
x_3	B	B		A	A
x_4	B	B	B		A
x_5	B	B	B	A	

Round 3

	x_1	x_2	x_3	x_4	x_5
x_1		A	A	A	A
x_2	B		A	A	A
x_3	B	B		A	A
x_4	B	B	A		A
x_5	B	B	B	B	

Figure 1: The OPRO LLM iteratively optimizes the "System Instruction & Task Prompt" used by the LLM judge. Our proposed method leverages uncertainty estimation to identify the most uncertain comparisons (highlighted in red). In each optimization step, only these uncertain comparisons are reevaluated using the new prompt. The results from the best prompt found so far are retained for the remaining comparisons, enabling significant computational savings while driving performance improvement.

having the wrong ordering:

$$\arg\max_{i,j} \; \mathsf{P}(s_i < s_j | C_{1:K}) \tag{6}$$

Alternatively, we introduce a different strategy of choosing the most anomalous comparisons:

$$\arg\max_{i,j} \; \mathsf{KL}\left(p_{ij} || \mathsf{P}(x_i \succ x_j | \hat{s}_i, \hat{s}_j)\right) \tag{7}$$

$$= \arg\max_{i,j} \; p_{ij} \ln \frac{p_{ij}}{\sigma(\hat{s}_i - \hat{s}_j)} + (1 - p_{ij}) \ln \frac{1 - p_{ij}}{\sigma(\hat{s}_j - \hat{s}_i)} \tag{8}$$

based on the KL-divergence between the predicted LLM probability p_{ij} and the overall probability $\sigma(\hat{s}_i - \hat{s}_j)$ from the predicted scores $\hat{s}_{1:N}$. The more the LLM prediction deviates from the overall predicted scores, the more likely it is to be re-run.

Furthermore, an ideal LLM judge should also be positionally consistent [19, 34]. LLMs often unfairly favor the first or second candidate presented, irrespective of their actual quality [26]. This bias manifests as an inconsistency between the probability $p_{ij} = \mathsf{P}_{\mathsf{LLM}}(x_i \succ x_j) \neq 1 - \mathsf{P}_{\mathsf{LLM}}(x_j \succ x_i) = 1 - p_{ji}$. Therefore, we can task the OPRO LLM to generate candidates that not only improve the Spearman rank correlation but also minimize positional bias:

$$\mathsf{Bias} = \sum_{i \neq j} \mathsf{KL}\left(p_{ij} || 1 - p_{ji}\right) \tag{9}$$

$$= \sum_{i \neq j} p_{ij} \ln \frac{p_{ij}}{1 - p_{ji}} + (1 - p_{ij}) \ln \frac{1 - p_{ij}}{p_{ji}} \tag{10}$$

By cutting down the number of re-evaluations and tasking the OPRO LLM with optimizing positional bias, we can obtain improved prompts at a fraction of the cost needed in a standard OPRO setup.

4 Experimental Evaluation

We perform experiments on the summary evaluation SummEval dataset [12] containing 100 articles, each with 16 machine-generated summaries evaluated on four different attributes: coherency (COH), consistency (CON), fluency (FLU), and relevancy (REL). We optimize on 30 training articles, validate the generated prompts on 10 articles and report the performance of the best found prompt on the remaining 60 test articles. The OPRO LLM will be based on Gemini 2.5 Flash Preview 04-17 [2, 30], with structured outputs to ensure it generates a system instruction and a task prompt, and temperature set to 0 for deterministic outputs. Each OPRO run will consist of $S = 25$ rounds and we generate an initial of 5 prompt pairs as a starting point, see Appendix A for details. The LLM-as-a-judge will be based on the Qwen2.5 family [28]. We will optimise the prompts for the 3B model and benchmark it against the 7B and 14B models. Furthermore, we will investigate using both probability of reordering (PoR, Eq. 6) and kl-divergence (PKL, Eq. 7) to perform efficient selection. Finally, we will task the OPRO to optimise Spearman rank (SRC) and additionally positional bias (PB).

Results. Table 1 shows baseline performance of Qwen2.5 models, alongside performance after OPRO optimization using full evaluation. The OPRO-optimized prompts for the 3B model consistently improve SRC across all attributes, outperforming the 7B model and reaching 14B level. Notably, when optimizing for both SRC and PB, we observe a reduction in positional bias while largely maintaining or even slightly improving Spearman correlation on some attributes. This demonstrates the capability of the OPRO framework to handle multi-objective prompt optimization effectively.

Table 2 investigates the core contribution of our work: efficient prompt optimization. Here, 'All' refers to full re-evaluation after 5 OPRO rounds. The subsequent rows show performance when only re-evaluating the top 20% or 40% of uncertain comparisons selected

Table 1: Baseline performance: Spearman rank performance (↑)/positional bias (↓).

Model	Method	Rounds	COH	CON	FLU	REL
Qwen2.5-3B-Instruct	-	1.0	46.45 ± 2.43/1.95 ± 2.13	45.66 ± 3.72/1.28 ± 0.88	42.96 ± 2.18/2.27 ± 1.98	46.72 ± 2.10/2.40 ± 2.08
Qwen2.5-7B-Instruct	-	1.0	49.34 ± 3.21/3.53 ± 2.31	50.50 ± 2.40/1.59 ± 1.20	39.30 ± 1.19/4.85 ± 5.46	49.56 ± 1.17/1.68 ± 1.01
Qwen2.5-14B-Instruct	-	1.0	51.74 ± 1.03/3.31 ± 1.95	51.30 ± 1.57/2.31 ± 1.55	45.41 ± 0.91/6.30 ± 4.58	50.17 ± 0.82/2.72 ± 3.63
Qwen2.5-3B-Instruct	SRC	5 + S	50.84/2.80	50.75/1.95	45.51/0.94	50.61/1.61
	SRC + PB	5 + S	50.64/1.72	51.22/0.82	45.61/0.95	49.74/1.04

by either PoR or PKL. Using PoR to select just 20% of comparisons for re-evaluation achieves Spearman correlations remarkably close to full re-evaluation across most attributes for both SRC and SRC+PB optimization objectives. For instance, under SRC optimization, COH drops only slightly from 50.84 to 50.51, while REL even improves from 50.61 to 51.25. Increasing the re-evaluation to 40% often further closes the gap or even surpasses the full re-evaluation performance, particularly for COH and REL, suggesting that not all comparisons need to be rerun. The PKL selection mechanism also shows strong performance, often comparable to PoR, especially at the 40% selection rate.

Furthermore, Figure 2 shows the performance of various prompts when given a partial set of comparisons. It shows that efficient OPRO generates prompts that perform similarly to the full OPRO runs, again showcasing redundancies in having to perform full re-evaluation. Furthermore, prompts that take into account positional bias showcase much better performance when relying on a smaller number of comparisons. This is since the lower positional bias implies that once A vs B has been compared, B vs A provides very little information. On the other hand, prompts with higher positional bias still benefit from making both comparisons. This efficiency makes iterative prompt optimization for comparative LLM-as-a-judge a much more practical and scalable.

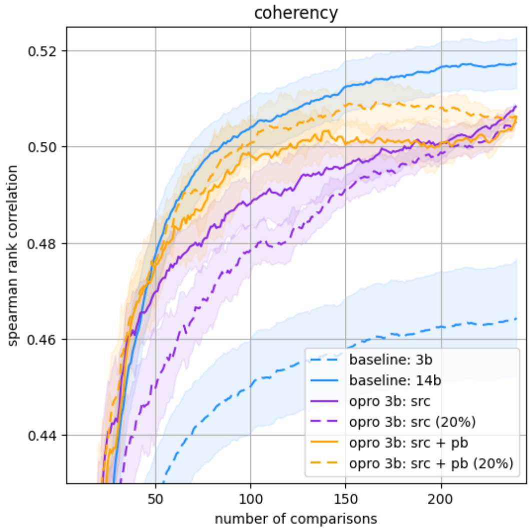


Figure 2: Coherency: Spearman rank performance when selecting a random set of comparisons.

5 Conclusion

This work introduced an efficient method for optimizing prompts (both system and task-level) for comparative LLM-as-a-judge. By integrating uncertainty estimation–specifically using Probability of Reordering and KL-divergence metrics–into OPRO, we selectively re-evaluate only the most uncertain pairwise comparisons when assessing new candidate prompts. Our experiments on the SummEval dataset demonstrate that this approach can reduce the computational cost of prompt re-evaluation by up to 80% while achieving Spearman rank correlations and positional bias levels comparable to, and sometimes exceeding, those obtained with full re-evaluation. This significant efficiency gain makes iterative prompt optimization for comparative LLM-as-a-judge more practical and scalable.

Table 2: Efficient performance: Spearman rank performance (↑)/positional bias (↓). The selection mechanisms only pick the top 20% or 40% of uncertain comparisons instead of re-evaluating all.

Model	Method	Selection	Rounds	COH	CON	FLU	REL
Qwen2.5-3B-Instruct	SRC	All	5.0 + S	50.84/2.80	50.75/1.95	45.51/0.94	50.61/1.61
		PoR (20%)	5.0 + 0.2S	50.51/2.94	50.74/2.14	45.54/1.06	51.25/2.30
		PoR (40%)	5.0 + 0.4S	53.13/2.32	51.15/1.88	45.90/0.82	51.40/1.87
		PKL (20%)	5.0 + 0.2S	50.37/2.39	50.97/1.73	45.93/1.02	50.37/2.09
		PKL (40%)	5.0 + 0.4S	50.23/2.70	51.12/1.69	45.92/0.79	50.23/2.40
Qwen2.5-3B-Instruct	SRC + PB	All	5.0 + S	50.64/1.72	51.22/0.82	45.61/0.95	49.74/1.04
		PoR (20%)	5.0 + 0.2S	50.60/1.67	50.97/0.62	45.33/0.93	50.80/1.00
		PoR (40%)	5.0 + 0.4S	52.58/1.68	50.94/0.50	45.71/0.68	50.96/1.14
		PKL (20%)	5.0 + 0.2S	50.29/1.49	51.50/0.68	44.78/0.88	50.29/1.29
		PKL (40%)	5.0 + 0.4S	50.77/1.30	51.60/0.97	45.49/0.57	50.77/1.03

References

[1] 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* (2023).

[2] 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* (2023).

[3] David Beaudoin and Tim Swartz. 2018. A computationally intensive ranking system for paired comparison data. *Operations Research Perspectives* 5 (2018), 105–112.

[4] Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika* 39, 3/4 (1952), 324–345.

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

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

[7] Bowen Cao, Deng Cai, Zhisong Zhang, Yuexian Zou, and Wai Lam. 2024. On the Worst Prompt Performance of Large Language Models. In *Advances in Neural Information Processing Systems*.

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

[9] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research* (2024).

[10] László Csató. 2013. Ranking by pairwise comparisons for Swiss-system tournaments. *Central European Journal of Operations Research* 21 (2013), 783–803.

[11] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[12] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics* 9 (2021), 391–409.

[13] Yassir Fathullah and Mark J. F. Gales. 2025. Generalised Probabilistic Modelling and Improved Uncertainty Estimation in Comparative LLM-as-a-judge. *arXiv preprint arXiv:2505.15240* (2025).

[14] Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024. Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers. In *International Conference on Learning Representations*.

[15] Ralf Herbrich, Tom Minka, and Thore Graepel. 2007. TrueSkill™: A Bayesian Skill Rating System. In *Advances in Neural Information Processing Systems 19*. MIT Press, 569–576.

[16] Geoffrey E. Hinton. 1999. Products of experts. In *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, Vol. 1. IET, 1–6.

[17] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 4582–4597. doi:10.18653/v1/2021.acl-long.353

[18] 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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 2511–2522. doi:10.18653/v1/2023.emnlp-main.153

[19] Adian Liusie, Yassir Fathullah, and Mark JF Gales. 2024. Teacher-Student Training for Debiasing: General Permutation Debiasing for Large Language Models. *arXiv preprint arXiv:2403.13590* (2024).

[20] Adian Liusie, Potsawee Manakul, and Mark Gales. 2024. LLM Comparative Assessment: Zero-shot NLG Evaluation through Pairwise Comparisons using Large Language Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Yvette Graham and Matthew Purver (Eds.). Association for Computational Linguistics, St. Julian's, Malta, 139–151. https://aclanthology.org/2024.eacl-long.8

[21] Adian Liusie, Vatsal Raina, Yassir Fathullah, and Mark Gales. 2024. Efficient LLM Comparative Assessment: a Product of Experts Framework for Pairwise Comparisons. *arXiv preprint arXiv:2405.05894* (2024).

[22] Jordan J Louviere, David A Hensher, and Joffre D Swait. 2000. *Stated choice methods: analysis and applications*. Cambridge university press.

[23] Charles F Manski. 1977. The structure of random utility models. *Theory and decision* 8, 3 (1977), 229.

[24] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[25] ChaeHun Park, Minseok Choi, Dohyun Lee, and Jaegul Choo. 2024. PairEval: Open-domain Dialogue Evaluation with Pairwise Comparison. *arXiv preprint arXiv:2404.01015* (2024).

[26] Pouya Pezeshkpour and Eduardo R Hruschka. 2024. Large Language Models Sensitivity to The Order of Options in Multiple-Choice Questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*. 2006–2017.

[27] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. *arXiv preprint arXiv:2306.17563* (2023).

[28] Qwen Team. 2024. Qwen2.5: A Party of Foundation Models. https://qwenlm.github.io/blog/qwen2.5/

[29] Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. 2025. Benchmarking prompt sensitivity in large language models. In *European Conference on Information Retrieval*. Springer, 303–313.

[30] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530* (2024).

[31] 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]

[32] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems* 36 (2023), 51008–51025.

[33] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large Language Models as Optimizers. In *International Conference on Learning Representations (ICLR)*.

[34] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large Language Models Are Not Robust Multiple Choice Selectors. In *International Conference on Learning Representations*.

[35] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685* (2023).

[36] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less Is More for Alignment. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 55006–55021. https://proceedings.neurips.cc/paper_files/paper/2023/file/ac662d74829e4407ce1d126477f4a03a-Paper-Conference.pdf

[37] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large Language Models are Human-Level Prompt Engineers. In *The Eleventh International Conference on Learning Representations*.

[38] Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. ProSA: Assessing and Understanding the Prompt Sensitivity of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 1950–1976.

Figure 3: The meta prompt used to generate a collection of prompts. This is formulated for evaluating SummEval coherency.

We have a collection of summaries for a given article and want to prompt a large language model to compare each pair of summaries against each other to determine which one is more coherent. From this, the plan is to obtain a full ranking of all the summaries using for example Bradley-Terry. The performance of the overall predicted ranking from the language model will be compared with the true ranking and be measured using Spearman Rank Correlation (higher is better).

However, the language model suffers from positional bias. The positional bias, as measured by KL-divergence (lower is better), measures the discrepancy in the prediction by the language model when the two options are reversed. This information could be relevant to the language and can be used in the prompt.

Therefore, we need to write a well-designed prompt that can trigger the language model to produce a high Spearman Rank while keeping the biases low. Furthermore, we can use the optional "System Instruction" to decide the tone and style instructions of the model. As a starting point, write 5 different prompts (with associated system instructions) using the Summary 1/2 label set and the fillers <context>, <A> and <B>, do not ask the language model to reason. Follow a similar format and style to the examples below:

System Instruction: "You are an expert summary assessment system."
Prompt: "Which Summary is more coherent, Summary 1 or Summary 2?\n\nArticle: <context>\n\nSummary 1: <A>\n\nSummary 2: <B>"

System Instruction: ""
Prompt: "Article: <context>\n\nSummary 1: <A>\nSummary 2: <B>\n\nWhich summary is more coherent as a summary of the article? Answer ’1’ or ’2’."

A Experimental Setup

In all SummEval experiments, we start with 5 prompts generated by Gemini 2.5 Flash Preview 04-17 with a temperature of 0 to ensure deterministic outputs, see Figure 3. This collection of prompts will then be used as a starting point for the OPRO process. Once the prompts have been evaluated, the meta prompt for OPRO then incorporates the scores into the process, see Figure 4. To ensure that the OPRO LLM generates new prompts in the correct format, structured outputs are used to ensure a string field for both a system instruction and a task prompt. Furthermore, note that we use the fillers "<context>", "<A>" and "<B>" as fields that are used to populate the prompt with articles and two different summaries being compared against each other.

B Additional Results

Figure 5 shows additional results enforcing the observations made in Figure 2. Two observations can be made: (1) efficient OPRO can perform as well as standard OPRO even when more than 80% of comparisons are saved from previous rounds; (2) additionally optimising for positional bias yields prompts that perform well under a limited number of randomly chosen comparisons. This is consistently observed across all 4 attributes of summarisation in SummEval.

Figure 4: The meta prompt used for the OPRO process. This is formulated for evaluating SummEval coherency.

We have a collection of summaries for a given article and want to prompt a large language model to compare each pair of summaries against each other to determine which one is more coherent. From this, the plan is to obtain a full ranking of all the summaries using for example Bradley-Terry. The performance of the overall predicted ranking from the language model will be compared with the true ranking and be measured using Spearman Rank Correlation (higher is better).

However, the language model suffers from positional bias. The positional bias, as measured by KL-divergence (lower is better), measures the discrepancy in the prediction by the language model when the two options are reversed. This information could be relevant to the language and can be used in the prompt.

Therefore, we need to write a well-designed prompt that can trigger the language model to produce a high Spearman Rank while keeping the biases low. Furthermore, we can use the optional 'System Instruction' to decide the tone and style instructions of the model.

Example 1
System Instruction: "You are a detail-oriented text analyst."
Prompt: "Compare the following summaries for coherence:\n\nArticle: <context>\n\nSummary 1: <A>\n\nSummary 2: <B>\n\nIndicate the number of the more coherent summary:"
Spearman Rank Correlation: 44.47
Positional Bias: 1.94

Example 2
System Instruction: "You are a neutral evaluator of text quality."
Prompt: "Evaluate the coherence of the following summaries relative to the article.\n\nArticle: <context>\n\nSummary 1: <A>\n\nSummary 2: <B>\n\nSelect the more coherent summary: 1 or 2."
Spearman Rank Correlation: 46.04
Positional Bias: 3.05

…

The above is a list of different prompts and associated system instructions (ranked from worst to best Spearman Rank) used to compare the summaries (in coherency) along with their associated performance score. First, analyze each and every prompt and what makes a prompt perform well or badly with attention to the length of the prompt. Then, write a new prompt which has to include the Summary 1/2 label set and the fillers <context>, <A> and <B>. The proposed prompt should primarily achieve higher Spearman Rank Correlation and lower positional bias.

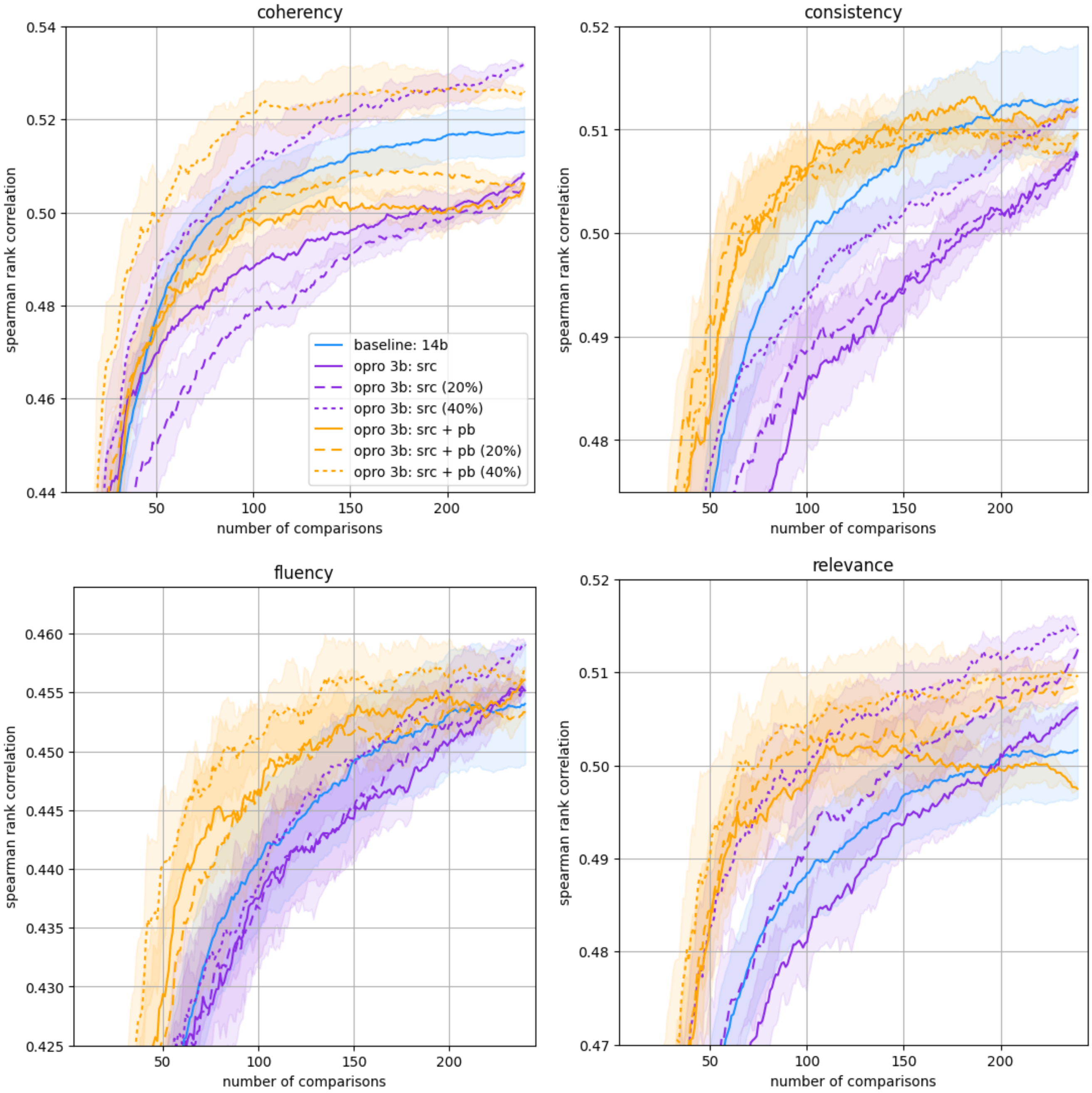


Figure 5: Spearman rank performance when selecting a random set of comparisons.