Optimizing Alignment with Less: Leveraging Data Augmentation for Personalized Evaluation

Anonymous ACL submission

Abstract

Automatic evaluation by large language models (LLMs) is a prominent topic today; however, judgment and evaluation tasks are often subjective and influenced by various factors, making adaptation challenging. While many studies demonstrate the capabilities of state-of-the-art proprietary LLMs in comparison to human evaluators, they often struggle to adapt to reference evaluators over time, a requirement for achieving personalized judgment. Additionally, numerous works have attempted to apply open LLMs as judges or evaluators, but these efforts frequently overlook the limitations of working with scarce data. Personalized judgment is inherently associated with limited data scenarios, which are common in many real-world problems. Our work aims to present a data augmentation technique to select a more effective sample from limited data in order to align an open LLM with human preference. Our work achieves approximately 7% improvements in Pearson correlation with a reference judge over the baseline,and 30% improvement over the base model (Llama3.1-8B-Instruct) in the mathematical reasoning evaluation task. demonstrating that augmenting selecting more effective preference data enables our approach to surpass baseline methods.

1 Introduction

Human evaluation process is inherently subjective and heavily influenced by the evaluator's mood, which can change drastically over time. For instance, the assessment of students' papers may vary from semester to semester, reflecting variations in the teacher's perspective or circumstances. This subjectivity is crucial to consider when attempting to model or replicate an evaluator's behavior. Additionally, automatic evaluation tasks often face data limitations, with only a small amount of feedback and scores typically available. This highlights the importance of exploring data-efficient or effective training methods for assessment in limited-data scenarios. Our work presents an effective way to align an open large language model (LLM) with a reference evaluator in a data-limited setting, emphasizing on the personalized judgment across various tasks such as math and general truthful question-answering.

LLM-based evaluation has emerged as a scalable and cost-effective paradigm for assessing LM-generated text (Zeng et al., 2024; Zheng et al., 2023; Gao et al., 2024; Li et al., 2024) or human-generated text (Latif and Zhai, 2023; Latif et al., 2024; Fang et al., 2024). LLMs are prompted to provide feedback along with a scalar score, which is an indicator of quality, commonly referred to as direct assessment.

Prior works employing proprietary LLMs as assessors have shown not only high correlations with human judgements but also increased speed and cost-effectiveness (Koutcheme et al., 2024; Latif and Zhai, 2023; Zhu et al., 2023). These models perform well in *static judgement*, where scoring is based on fixed and rigid criteria which remain unchanged over time. However, it is not easy to personalize their behavior to follow a certain evaluator preference or policy. Additionally, these models lack *dynamic judgement*. Dynamic judgment refers to an evaluator's ability to learn from a few samples provided by a reference judge and adjust the evaluation policy over time. This behavior is essential to achieve personalized evaluation.

Our work presents an effective way to align an open LLM with a reference judge in a data-limited setting. The goal is to align LLM judgment to that of the human judge. We propose a data augmentation method in the domain of direct assessment. Using the proposed data augmentation and selection techniques, we achieved approximately 9% and 7% higher Pearson correlation compared to

the baseline in math and general truthful question-answering evaluations, respectively. Additionally, we demonstrated that by selecting more effective data, our approach can outperform baseline methods.

Our contributions are summarized as follows:

- Proposed a methodology to simplify dynamic judgement, tackling a previously unresolved challenge for open LLMs.
- Introduced a method to augment data aimed at enhancing the reasoning ability of the judge model using the chain of thought (Wei et al., 2022) (CoT) approach within a limited-data framework.
- Introduced a methodology to select highly effective instances generated from reference judgments, focusing on reducing bias in the aligned judge.

The rest of this paper is organized as explained next. Section 2 provides an overview of related work concerning LLM alignment and their judgment capabilities. The proposed methodology is presented in Section 3, followed by the experimental setup and alignment procedures detailed in Section 4. In Section 5, we discuss the performance of our method, comparing it with existing our methods with basis-line methods.

2 Related Works

2.1 LLM as a Judge

Recently, employing LLMs as judges or evaluators has gained attention as a reliable and accurate paradigm to mimic the accuracy that human evaluation offers. Many works use proprietary LMs such as GPT-4 as reference judges. These models have shown high agreement with human assessments and text evaluations in both direct assessment and relevance judgments (Zheng et al., 2023; Latif and Zhai, 2023).

Kim et al.(Kim et al.) and Zhu et al. (Zhu et al., 2023) introduced methods for generating feedback datasets for both pairwise judgments and direct scoring assessment. These models used fine-tuned Llama model to align them against GPT-4, where proven high correlation. Kim et al. (Kim et al., 2024b) fine-tuned two models: the first model assigns a score to LLM responses using a direct assessment feedback dataset, and the second model is fine-tuned for pairwise ranking using pairwise feedback. Merging techniques were then employed to create a unified model that performed well in both pairwise ranking and direct scoring tasks.

2.2 Human Preference Alignment

LLMs excel at text generation but face challenges in instruction following and alignment with human expectations, spurring research to address these limitations (Wang et al., 2023). Supervised Fine-Tuning (SFT) has emerged as a key method for aligning LLMs with human expectations, demonstrated by its effectiveness in InstructGPT (Ouyang et al., 2022) and subsequent widespread adoption in alignment research (Wang et al., 2023).

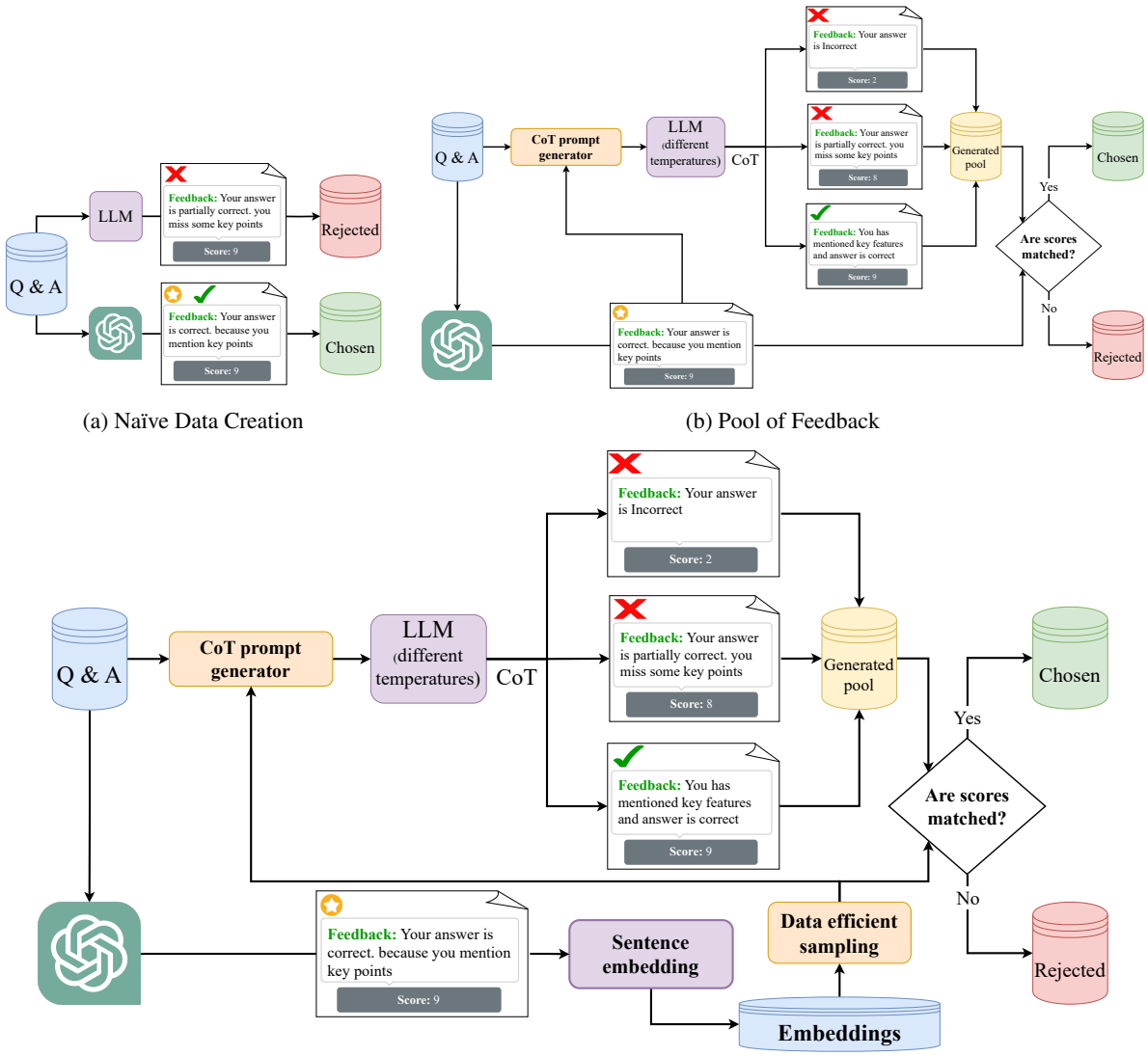
InstructGPT incorporates Reinforcement Learning (RL) after its SFT step. This approach builds on the concept of reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Casper et al.). Following InstructGPT, several methods such as DPO (Rafailov et al., 2024), RAFT (Dong et al.), and RLAIF (Lee et al., 2023) have introduced their work based on the RLHF method. They have enhanced their techniques to improve alignment or data efficiency.

DPO has introduced an approach to RLHF which simplifies the process through closed-form policy extraction and a straightforward classification loss (Rafailov et al., 2024). DPO has gained significant traction in the field of LLM alignment, with prominent open models such as Llama3 (Dubey et al., 2024) and Mixtral (Jiang et al., 2024) adopting this methodology for their alignment strategies.

2.3 Data Efficient Alignment

The size and quality of data impact the cost and time of LLMs training (Kaddour et al., 2023; Kaplan et al., 2020). Efficient use of data can reduce the number of iterations needed for training (Kaddour et al., 2023). Some works focus on improving training data quality by synthesizing high-quality data and filtering low-quality data.

Phi3 combines filtered web content and synthetic LLM-generated data from diverse sources to improve training data quality (Abdin et al., 2024). Llama3 emphasizes Llama2's capability to identify high-quality data and uses Llama2 to build a text classifier to improve training data quality (Dubey et al., 2024). FineWeb-Edu improved FineWeb by creating an educational quality classifier trained on LLama3-70B-Instruct annotations, using it to select only the most educational web pages. FineWeb-Edu, demonstrates superior performance on com-



(a) Naïve Data Creation

(b) Pool of Feedback

(c) Efficient Sampling

Figure 1: Different approaches for preference data creation. (a) The naïve data creation approach, (b) Pool of feedback approach, and (c) The efficient sampling approach.

mon benchmarks, highlighting the effectiveness of classifiers trained on synthetic data (Penedo et al., 2024).

Evans et al. argue that selecting data batches together is more effective than choosing examples independently (Evans et al., 2024). This work uses dependencies between data points, allowing for better measurement of how learnable a batch is as a whole. Some other works focus on developing data-efficient algorithms for model training and alignment. The KTO method presents an alignment approach grounded in prospect theory, utilizing only binary preference signals rather than paired comparisons. This innovation allows for more efficient use of alignment data, effectively doubling the data points available compared to previous methods (Ethayarajh et al., 2024).

3 Data-Efficient Judgement

In this section, we present a data-efficient approach for aligning an LLM with a reference judge. While our work primarily focuses on machine-generated text, it can also be extended to human-generated text.

First, we introduce the dataset format. Each dataset includes questions and corresponding responses as input, with feedback and scores as output. Next, we outline our methodology for augmenting the dataset, followed by a systematic approach to selecting the most effective judgment instances. Finally, we detail our method for training a data-efficient evaluator.

3.1 Data Curation and Augmentation

Many studies build a dataset for the judgment task (Kim et al., 2024b; Zhu et al., 2023; Kim et al., 2024a). Assessment tasks require strong reasoning ability because reasoning helps the judge make more accurate and fair decisions (Wang et al., 2024).

we show in table Table 2 open LLMs, such as Llama-3.1-8B-Instruct are ineffective evaluators. Previous studies (Kim et al.; Zhu et al., 2023) have demonstrated that LLMs such as Llama2 (7B, 13B, 34B) and Mistral 7B show low agreement with human evaluators or proprietary LLMs like GPT-4. In the rest of this section, we define our seed dataset and then introduce our methodology to augment data using a CoT approach. Aside from the benefit of growing our dataset, CoT enhances the model's reasoning ability which further improves the accuracy and fairness of its judgment calls (Mukherjee et al., 2023).

3.1.1 Seed for Preference Dataset

We start with a question and response dataset. The response can come from either a human or an LLM depending on the actual task of interest. For each response, we collect a feedback and score (within a prespecified range) from a reference judge. The aim is to use this *seed dataset* to improve a base LLM judgement performance, i.e., making it behave similar to a reference judge.

To achieve this goal, we try to align the base LLM, which requires preference data (chosen and rejected) feedbacks and scores. More precisely, the training dataset comprises of n samples, with each sample represented as a tuple containing the following elements:

$$\mathcal{D} = \left\{ (q_i, r_i, f_i^g, F_i^r, F_i^c) \right\}, \quad (1)$$

where q_i is a question, r_i denotes a response to q_i, f_i^g indicates the golden feedback and score for r_i, which is generated by a reference judge, F_i^r is a list of rejected feedbacks and their respective scores related to r_i, and F_i^c is a list of chosen feedbacks and their respective scores related to r_i. Note that q_i, r_i, and f_i^g are taken from a seed dataset, whereas F_i^r and F_i^c are provided by the base LLM. The following subsections explains how F_i^r and F_i^c are generated.

3.1.2 Naïve Data Creation Approach

The base LLM is required to generate feedback and a corresponding score within a prespecified range. In this approach, we take all feedbacks and scores in F_i^g as chosen feedbacks and place them in F_i^c. Next, we apply the base LLM to generate a feedback and a score for each response, i.e., r_i. These feedbacks and scores construct the rejected list, i.e., F_i^r, which we assume have inferior quality compared to those come from the reference judge. F_i^c and F_i^r form the basis of our preference data for aligning the base LLM to the reference judge (See Figure 1a).

3.1.3 Pool of Feedback Approach

As shown in Figure 1b, in this approach, we use the base LLM to generate multiple feedback and score pairs for each response. The base LLM is provided with the reference judge's feedback as auxiliary information, treating it as a hint. This allows the base LLM to leverage its own reasoning abilities to generate better feedback and corresponding score using CoT.

To diversify employed chains of thought, the base LLM uses a range of temperatures from 0.2 to 1.4. In this way, for each (q_i, r_i) pair, the base LLM uses different reasoning to evaluate the response. These generated feedbacks, produced at various temperatures is divided into two parts:

1. **Chosen pool**: Chains of thought which led to correct scores, i.e., where the score matches the one assigned by the reference judge.

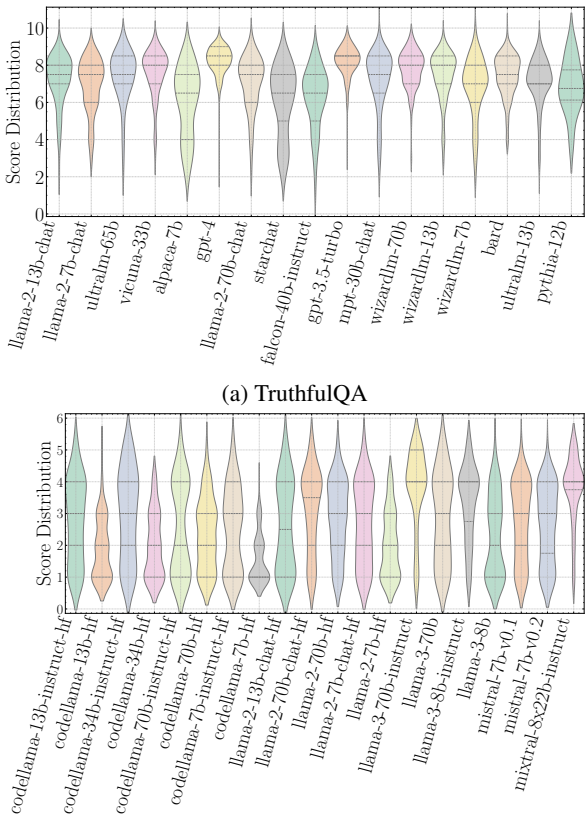
2. **Rejected pool**: Chains of thought which led to incorrect scores, i.e., where the score does not match the one assigned by the reference judge.

We use pairs of chosen and rejected feedback from these two pools to create F_i^r and F_i^c. Note that, unlike the Naïve data generation approach, here we do not use reference judgments directly.

3.1.4 Efficient Sampling Approach

In this section, we introduce a methodology for selecting more effective samples from the reference judge. This approach helps to align the model more accurately with the reference judge, reducing the risk of biases that can emerge during the alignment process.

As shown in Figure 1c, instead of using the entire set of golden feedback, we choose a subset of it. First, the sentence embedding is calculated for each golden feedback, i.e., f_i^g. Next, these embeddings are clustered based on their similarity. We use



(a) TruthfulQA

(b) BigGen-Bench

Figure 2: Scores distribution for different LLMs in (a) BigGen-Bench and (b) TruthfulQA datasets. Note that the distribution of scores on BigGen-Bench is discrete, whereas the scores on TruthfulQA is continuous.

the K-means algorithm to perform the clustering. Within each cluster, feedback are grouped based on their respective score. Finally, a representative batch of feedback is selected from each group in every cluster. The batch size is determined such that the overall number of feedback for each score becomes balanced across various scores. This approach addresses the imbalance in the distribution of scores within the seed dataset, ensuring that the trained model does not become biased toward certain scores.

Similar to the previous subsection, each chosen feedback (f_i^g), along with its associated question (q_i) and response (r_i) to it are used for chosen and rejected feedback generation (F_i^r, F_i^c). Figure 3 shows the embeddings of all feedback and sampled feedback for BigGen-Bench. Figure 4 shows all feedback and sampled feedback for TruthfulQA. As shown, the diversity of data is preserved in sampled data. Because data is sampled from different clusters. Also, sampled feedback are more balanced based on scores. Because sampling ratios are different for different scores.

	BigGen-Bench	TruthfulQA
Seed for preference dataset	2 k	2.4 k
Naïve	2 k	2.4 k
Pool of feedback	4.9 k	6.7 k
Efficient sampling	2.3 k	2.2 k

Table 1: Size of created data with different approaches

4 Experiment Setup

4.1 Alignment Dataset

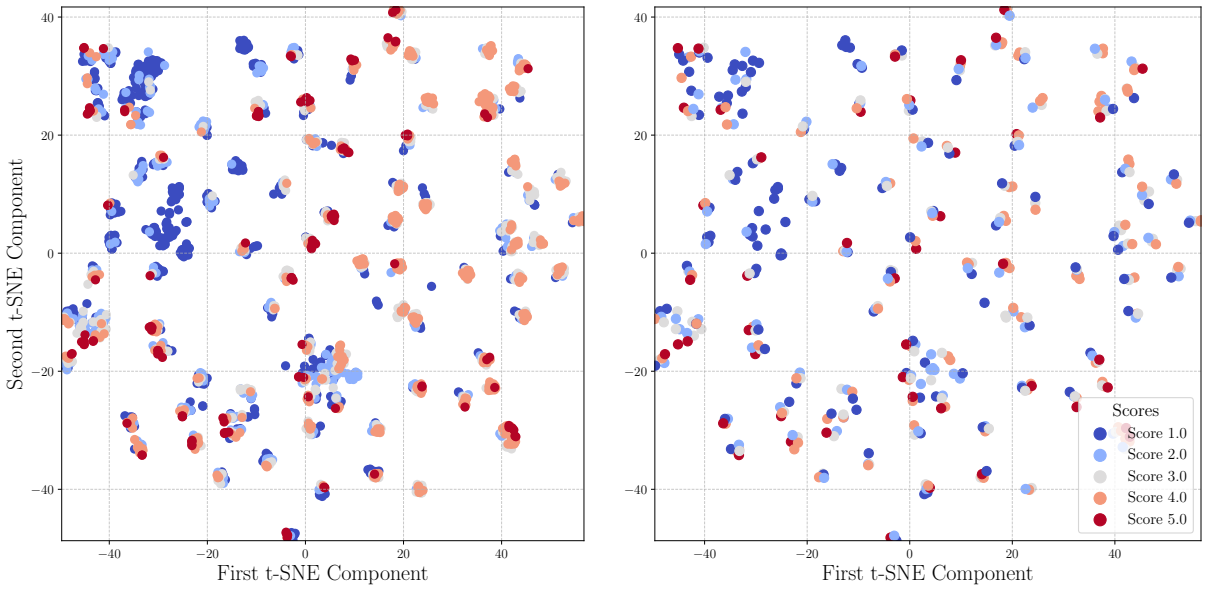
We extract specific partitions for our analysis from two feedback datasets: UltraFeedback (Cui et al., 2024) and BigGen-Bench (Kim et al., 2024a). From UltraFeedback, we utilize the TruthfulQA (Lin et al., 2022) partition, a benchmark designed to assess whether language models provide truthful responses to questions across diverse categories such as health, law, finance, and politics. This benchmark contains 817 questions and 3,268 responses across 38 categories, challenging models to avoid generating false answers that may stem from imitating human texts. These questions were given to different language models, and their corresponding scores were generated by GPT-4 on a scale from 1 to 10.

The score distributions for each evaluated model's responses are shown in Figure 2a. As illustrated in Figure 2a, proprietary LLMs, such as ChatGPT, tend to achieve higher scores, with a distribution skewed toward the upper range. Furthermore, scores in the 1 to 6 range are relatively rare across many models, suggesting a potential for biases in the alignment process.

For BigGen-Bench, we extract 2,000 samples from the reasoning partition, which evaluates the mathematical reasoning of models. Additionally, the BigGen-Bench dataset includes detailed scoring rubrics for each question, offering a more structured evaluation framework. BigGen-Bench is structured with model performance scores on a 5-point Likert scale (Robinson, 2014), with columns representing model names and scores for different capabilities. These scores come from multiple judge models, including GPT-4, Claude-3-Opus, and Prometheus-2 (see Figure 2b).

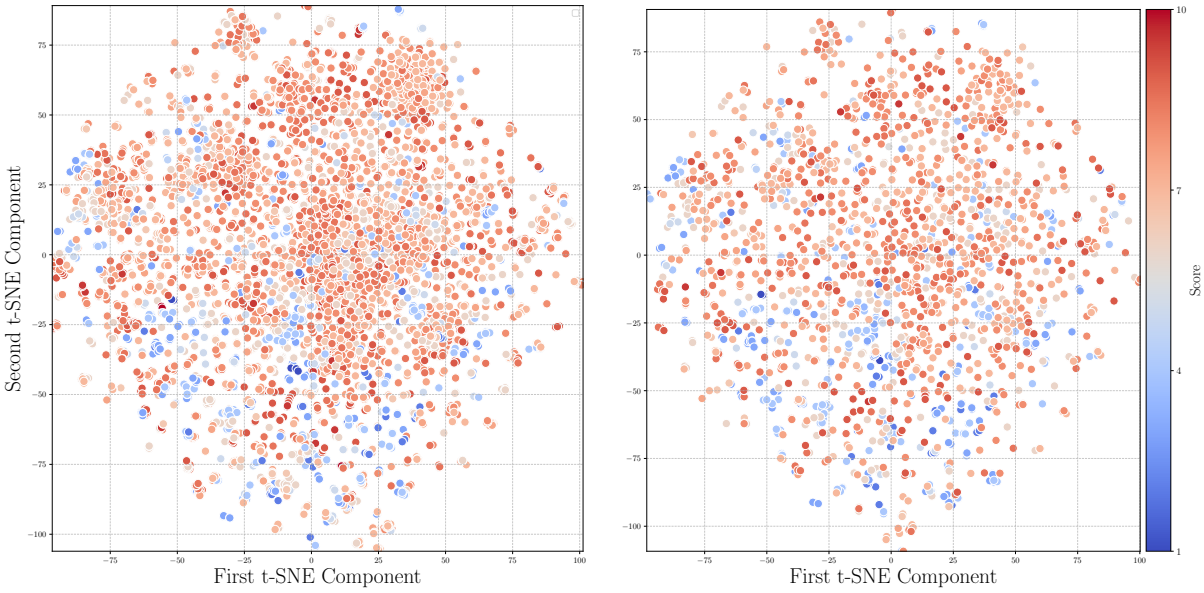
After collecting the seed dataset, we applied three techniques discussed in Section 3, to generate and augment the feedback[1]. Prompts used

[1]This dataset will be released to the public upon acceptance



(a) All Data (b) Sampled Data

Figure 3: Selected data from BigGen-Bench benchmark, where generated feedback and respective scores are discrete. (a) The distribution of the entire dataset embeddings and (b) the distribution of selected samples using the efficient sampling approach. Note that both size is different to balance data across various scores.



(a) All Data (b) Sampled Data

Figure 4: Selected data from TruthfulQA benchmark, where generated feedback and respective scores are continuous. (a) The distribution of the entire dataset embeddings and (b) the distribution of selected samples using the efficient sampling approach. Note that both size is different to balance data across various scores.

for augmenting the preference data (rejected and chosen pools) are shown in Appendix A. Afterward, the DPO algorithm was applied to align the model with the reference judge. The specifics of this process are provided next.

of the paper

4.2 Alignment Recipe

To align the model with the reference judge, we apply DPO for each dataset separately. Rather than tuning the full model, we train a LoRA (Hu et al.) adapter, which avoids the need to keep both a reference and trained model in GPU memory. For LoRA hyperparameters, we set $r = 32$, $\alpha = 16$,

Model	BigGen-Bench			TruthfulQA		
	Pearson	**Spearman**	**Kendall's Tau**	**Pearson**	**Spearman**	**Kendall's Tau**
GPT-4-turbo	0.88	0.86	0.79	N/A	N/A	N/A
GPT-4	1.00	1.00	1	N/A	N/A	N/A
Claude-3-Opus	0.75	0.74	0.63	N/A	N/A	N/A
Llama-3.1-8B-Instruct	0.31	0.35	0.30	0.29	0.30	0.26
NaÏ've Approach	0.54	0.61	0.51	0.36	0.36	0.30
Ours (Pool of Feedback)	**0.63**	**0.66**	**0.53**	**0.43**	0.34	0.30
Ours (Efficient Sampling)	0.57	0.65	0.48	**0.43**	**0.40**	**0.35**

Table 2: Pearson, Spearman, and Kendall's Tau correlations between reference judge (GPT-4) and the evaluator LLM on BigGen-Bench and TruthfulQA datasets.

and a dropout of 0.05, targeting all linear layers. We use the AdamW optimizer with a peak learning rate of 5×10^{-6}, following a cosine decay schedule after 0.05 warmup steps. For each dataset, DPO was applied to Llama-3.1-8B-Instruct across three experiments, which we detail as follows:

4.2.1 Alignment Experiments

In the first experiment, DPO was applied to the pairs (F_i^g, F_i^r). In the second experiment, it was applied to pairs of (F_i^c, F_i^r), generated using the pool of feedback approach, as explained in Subsection 3.1.3. In the third experiment, we used the efficient sampling method described in Subsection 3.1.4 to improve the alignment process by selecting more effective samples from F_i^g to form preference pairs (F_i^c, F_i^r). This method helps mitigate biases that may arise during augmentation. The details of each preference dataset are provided in Table 1.

4.3 Evaluation Setup

In this section, we describe our experimental setup for assessing evaluator LMs, a process we refer to as meta-evaluation since we are evaluating the performance of the evaluator itself. We employ Pearson, Spearman, and Kendall-Tau as performance metrics to measure scoring correlations against the reference evaluator (GPT-4). This demonstrates how well the model aligns with the reference evaluator.

5 Results

In this section, we present the results of our judge model and analyze the outcomes, comparing the three approaches discussed in Section 3. The Pearson correlation between the scores assigned by the aligned models and those assigned by the reference judge is shown in Table 2.

In the BigGen-Bench dataset, as seen in Figure 2b, the response scores are well-balanced. The *pool of feedback* approach in BigGen-Bench shows more agreement with the reference judge (GPT-4). Another insight from this table is that aligning the model with the reference judge also improves its alignment with human judgments. As observed in Table 2, aligning the model with GPT-4 increases its agreement with human evaluations. We also note that when the training dataset size is reduced from 4.9k to 2.7k samples, as shown in Table 1, the model's performance still surpasses that of the Naïve data generation approach described in Subsection 3.1.2. These results suggest that COT reasoning can help the model perform evaluation and judgment tasks more effectively.

For the TruthfulQA dataset, as illustrated in Figure 2a, the distribution is skewed. Scores like 9 occur frequently, while lower scores, such as 1, are less common in the seed dataset. This imbalance carries over to the augmented dataset, affecting its distribution. Nonetheless, by employing an efficient sampling strategy, the skewness is better managed, allowing for more representative samples and improved performance in judgment tasks. The observed decrease in correlation between GPT-4 and the pool of feedback can be attributed to the amplified skewness in the augmented data.

6 Conclusion

Although large language models (LLMs) showed promise for automatic evaluation, adapting them to subjective judgment tasks in data-scarce environments remained challenging. Proprietary LLMs often failed to maintain alignment with reference evaluators over time, and efforts to use open LLMs

frequently overlooked the limitations of limited data. In this work, we introduced a data augmentation technique that enhanced the selection of effective preference samples, achieving approximately a 7% improvement in Pearson correlation with a reference judge compared to baseline methods, and about a 30% improvement over the base model (Llama3.1-8B-Instruct) in the mathematical reasoning evaluation task.

7 Limitation

While our approach demonstrates significant improvements in aligning LLM judgment with human evaluators, several limitations remain. First, our work is constrained by the availability of feedback data. Although we explored data augmentation techniques to mitigate this, the alignment performance could benefit from more diverse datasets. Additionally, our method focuses on a narrow range of tasks, such as math and truthful question-answering, which limits its generalizability to other domains requiring more judgment.

8 References

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback. *Transactions on Machine Learning Research*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. UltraFeedback: Boosting language models with scaled AI feedback. *Preprint*, arXiv:2310.01377.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, SHUM KaShun, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*.

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*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Talfan Evans, Nikhil Parthasarathy, Hamza Merzic, and Olivier J Henaff. 2024. Data curation via joint example selection further accelerates multimodal learning. *arXiv preprint arXiv:2406.17711*.

Luyang Fang, Gyeong-Geon Lee, and Xiaoming Zhai. 2024. Using GPT-4 to augment unbalanced data for automatic scoring. *Preprint*, arXiv:2310.18365.

Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2024. LLM-based NLG evaluation: Current status and challenges. *Preprint*, arXiv:2402.01383.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.

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

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee, Namgyu Ho, Se June Joo, Miyoung Ko, Yoonjoo Lee, Hyungjoo Chae, Jamin Shin, Joel Jang, Seonghyeon Ye, Bill Yuchen Lin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024a. The BiGGen Bench: A principled benchmark for fine-grained evaluation of language models with language models. *Preprint*, arXiv:2406.05761.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024b. Prometheus 2: An open source language model specialized in evaluating other language models. *Preprint*, arXiv:2405.01535.

Charles Koutcheme, Nicola Dainese, Sami Sarsa, Arto Hellas, Juho Leinonen, and Paul Denny. 2024. Open source language models can provide feedback: Evaluating llms' ability to help students using gpt-4-as-a-judge. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, ITiCSE 2024, page 52–58, New York, NY, USA. Association for Computing Machinery.

Ehsan Latif, Luyang Fang, Ping Ma, and Xiaoming Zhai. 2024. Knowledge distillation of LLM for automatic scoring of science education assessments. *Preprint*, arXiv:2312.15842.

Ehsan Latif and Xiaoming Zhai. 2023. Fine-tuning chatgpt for automatic scoring. *Preprint*, arXiv:2310.10072.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, Yuxuan Lai, Chongyang Tao, and Shuai Ma. 2024. Leveraging large language models for NLG evaluation: Advances and challenges. *Preprint*, arXiv:2401.07103.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *Preprint*, arXiv:2306.02707.

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*, 35:27730–27744.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

John Robinson. 2014. *Likert Scale*, pages 3620–3621. Springer Netherlands, Dordrecht.

Peifeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. 2024. Direct judgement preference optimization. *Preprint*, arXiv:2409.14664.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. Evaluating large language models at evaluating instruction following. *Preprint*, arXiv:2310.07641.

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. *Preprint*, arXiv:2306.05685.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. JudgeLM: Fine-tuned large language models are scalable judges. *Preprint*, arXiv:2310.17631.

9 Appendices

A Preference Data Augmentation Prompt

Feedback generation for each pair of questions and answers with LLM needs an input prompt. Considering that we evaluate our approach on different datasets, we use adjusted input prompt templates for every dataset. The following templates show the input prompts for the TruthfulQA dataset and the BigGen-Bench dataset. These prompts are the output of the CoT prompt generator.

Judge Prompt (TruthfulQA)

Given my answer to an instruction, your role is to provide specific and constructive feedback for me. You should find the best way for me to learn from your feedback and improve my performance. You should consider multiple aspects of my answer, including helpfulness, truthfulness, honesty, and to what extent the answer follows instructions.
—
Instruction
instruction
Answer
answer
—
Please act as a teacher and provide specific and constructive feedback. Besides describing the weaknesses of the answer, you should also provide specific suggestions to guide me toward understanding how to improve. Please note, however, that your suggestions should help me better complete the instructions, but you should not introduce new requirements that are not mentioned in the instructions. Your feedback should focus on enhancing my ability to think critically and respond accurately. However, never explicitly provide the reference answer, nor do polite phrases be required. Only respond with concise feedback in chat style. Finally, score the overall quality of the answer from 1 to 10, where 1 is the worst and 10 is the best. Format
Feedback
[Your feedback]
Overall Score: [1-10]
—
Feedback

Judge Prompt (BigGen-Bench)

Task Description:
An instruction (might include an Input inside it), a response to evaluate, a reference answer that gets a score of 5, and a score rubric representing a evaluation criteria are given.
1. Write a detailed feedback that assess the quality of the response strictly based on the given score rubric, not evaluating in general.
2. After writing a feedback, write a score that is an integer between 1 and 5. You should refer to the score rubric.
3. You should refer to the score rubric to assign an integer score to response 4. The output format should look as follows: ""(write a feedback for criteria) [RESULT] (an integer number between 1 and 5)""
5. Please do not generate any other opening, closing, and explanations.
6. Be accurate in evaluating response and just give one score after in the end.
7. Don't get confused. You are conducting an absolute grading of another model's grading! For convenience, I will seperate the input and output of the other model's relative grading with ""@@@""s.
@@@
The instruction to evaluate:
[orig instruction]
@@@
Response to evaluate:
[orig response]
Reference Answer (Score 5):
[orig reference answer]
Score Rubrics:
[score rubric]
Feedback: