Improving LLM-as-a-Judge Inference with the Judgment Distribution

Victor Wang[1], Michael J.Q. Zhang[2], Eunsol Choi[2]
Department of Computer Science
[1]The University of Texas at Austin, [2]New York University
victorwang37@utexas.edu

Abstract

Using language models to scalably approximate human preferences on text quality (LLM-as-a-judge) has become a standard practice applicable to many tasks. A judgment is often extracted from the judge's textual output alone, typically with greedy decoding. However, LLM judges naturally provide *distributions* over judgment tokens, inviting a breadth of inference methods for extracting fine-grained preferences. We find that taking the mean of the judgment distribution consistently outperforms taking the mode (i.e. greedy decoding) in all evaluation settings (i.e. pointwise, pairwise, and listwise). We further explore novel methods of deriving preferences from judgment distributions, and find that methods incorporating risk aversion often improve performance. Lastly, we analyze LLM-as-a-judge paired with chain-of-thought (CoT) prompting, showing that CoT can collapse the spread of the judgment distribution, often harming performance. Our findings show that leveraging distributional output improves LLM-as-a-judge, as opposed to using the text interface alone.

1 Introduction

LLM-as-a-judge has emerged as a scalable framework for evaluating model outputs by approximating human annotation (Lin et al., 2024; Li et al., 2024b; Dubois et al., 2024). Typically, such systems prompt off-the-shelf LLMs to score a response or rank multiple responses to a given user prompt. LLM-as-a-judge methods boast strong agreement with human judgments across a breadth of domains and criteria (Zheng et al., 2023b; Ye et al., 2023), despite current limitations (Koo et al., 2023; Tan et al., 2024).

Most prior work involving LLM-as-a-judge elicits judgments through the LLM's text interface (Lin et al., 2024; Zhu et al., 2023; Ye et al., 2023), where the most likely token (i.e. the mode of the next token distribution) or a sampled token is taken to represent the LLM's judgment. Recent works (Lee et al., 2024a; Liu et al., 2023b; Yasunaga et al., 2024) have suggested that taking the mean of the score token distribution can better represent the LLM's judgment. In this work, we comprehensively evaluate design choices for leveraging LLM judges' distributional output.[1]

We show that the mean consistently outperforms the mode in the pointwise, pairwise, and listwise settings (i.e. evaluating one, two, and many responses at a time). Specifically, the mean achieves higher accuracy in 42 out of 48 cases on RewardBench (Lambert et al., 2024) and MT-Bench (Zheng et al., 2023b). We further explore novel methods of deriving preferences from score distributions (Section 4). For example, incorporating risk aversion often improves performance. Categorizing methods as discrete or continuous, where discrete methods (e.g. mode) are simple to interpret like rubric scores, we find that continuous methods outperform discrete methods, due to the latter often predicting ties and failing to capture slight preferences. In particular, the mode assigns ties more frequently than every other method, leading to the lowest accuracy even among discrete methods.

We further study how chain-of-thought (CoT) prompting (Wei et al., 2022) impacts the performance of LLM-as-a-judge. After the CoT reasoning, LLMs often exhibit sharper score distributions, making the mean judgment similar to the mode. Removing CoT increases the spread of the judgment distribution, often improving performance, and more so for taking the mean than taking the mode (e.g. absolute +6.5% for mean vs. +1.4% for mode, on average with pointwise scoring on RewardBench), demonstrating the synergy between eliciting and using distributional output.

Our findings stress the importance of leveraging

[1]We provide implementations of the evaluated methods at https://github.com/dubai03nsr/distributional-judge.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 23173–23199
November 4-9, 2025

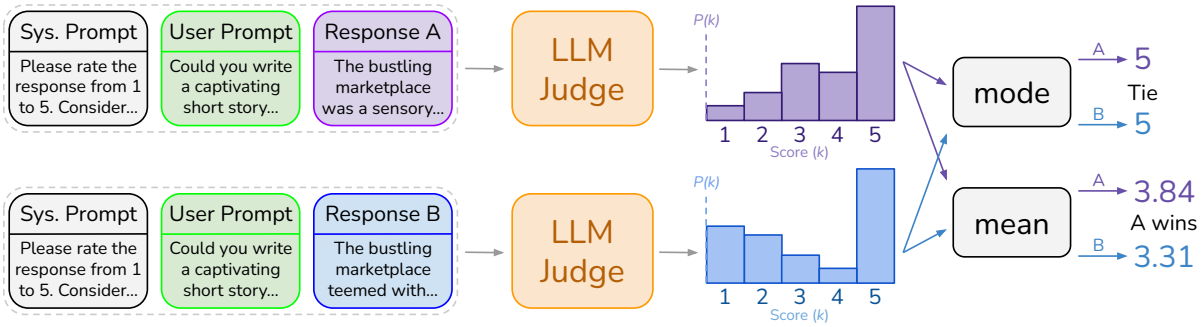

Figure 1: Pointwise LLM judge's logits produce a score distribution. We show two ways to compare two score distributions: (1) comparing the modes of the distributions and (2) comparing the means of the distributions.

distributional output to maximize the effectiveness of LLM-as-a-judge, as opposed to using the text interface alone. As LLM-as-a-judge paradigms are widely adopted for complex tasks, improving best practices for using LLM-as-a-judge can impact many end tasks' development and evaluation.

2 Background

2.1 LLM-as-a-Judge Settings

We briefly review three settings for LLM-as-a-judge; see Appendix A for more background.

Pointwise Scoring The LLM judge scores the two texts independently on a scale from 1 to some K, as shown in Figure 1 (Zheng et al., 2023b; Lin et al., 2024; Cui et al., 2023).

Pairwise Scoring The LLM judge scores both texts in a single prompt (Zhu et al., 2023; Saha et al., 2023; Chan et al., 2023). To account for position bias, we prompt the LLM judge twice, once for each order of presentation, and average the outputs (Lee et al., 2024a).

Pairwise Ranking The LLM judge states which of the two texts it prefers (Lin et al., 2024; Li et al., 2024b; Dubois et al., 2024). As with pairwise scoring, we prompt the LLM judge twice, once for each order of presentation.

2.2 Related Work

Mean Judgment Several prior works have used the mean of the judgment distribution, mostly in the pointwise setting. Liu et al. (2023b); Lee et al. (2024a); Saad-Falcon et al. (2024) note the benefits of the mean but do not empirically compare it with the mode. Zawistowski (2024), Hashemi et al. (2024), Lukasik et al. (2024) show that the mean outperforms the mode for summary scoring, dialogue scoring, and other regression tasks. Concurrent work (Yasunaga et al., 2024) shows that the mean outperforms the mode on RewardBench (Lambert et al., 2024), but the paper's focus is on data-efficient alignment.

Lee et al. (2024a); Zhai et al. (2024) use pairwise judgment distributions to train a student model, but do not empirically compare with distillation using one-hot judgments. In this work, we benchmark the mode, the mean, and newly proposed methods for leveraging distributional judgments across the pointwise, pairwise, and listwise settings.

CoT Zheng et al. (2023b) presented preliminary evidence that CoT benefits LLM-as-a-judge. Other LLM-as-a-judge systems have been proposed that take advantage of LLMs' ability to perform CoT reasoning (Ankner et al., 2024; Feng et al., 2024). On the other hand, Liu et al. (2024f) evaluate many evaluation protocols and find that CoT can hurt performance. However, their analysis assumes access only to the judges' text interface, not examining the effect of CoT on the judgment distribution. In this work, we analyze the interplay between CoT and the inference method (e.g. mode vs. mean).

Related phenomena on the effect of CoT have been studied in the literature (Chiang and Lee, 2023; Stureborg et al., 2024; Liu et al., 2024a; Lee et al., 2023; Sprague et al., 2024; Hao et al., 2024; Zheng et al., 2023b). Wang and Zhou (2024) show the sharpening effect of CoT, which improves performance on numerical reasoning tasks. In this work, we show that this sharpening effect can be harmful when the LLM is used as a judge.

Distributional Reward Models Using distributional judgment makes it possible for LLM judges to represent pluralistically aligned preferences (Sorensen et al., 2024; Siththaranjan et al., 2023; Kumar et al., 2024). Compared to existing work on

distributional reward models (Siththaranjan et al., 2023; Zhang et al., 2024b; Li et al., 2024a; Dorka, 2024; Poddar et al., 2024; Padmakumar et al., 2024), (1) our setting involves LLMs not trained or prompted for distributional judgment (Meister et al., 2024), and (2) LLM judges can produce arbitrary distributions over a flexibly chosen discrete judgment space.

3 Distributional Judgment

In this section, we present our findings comparing mode vs. mean inference and CoT vs. no-CoT prompting for LLM-as-a-judge systems.

3.1 Methods

To infer a judgment from the LLM's output distribution, we use the mode or the mean. With **mode**, we perform greedy decoding to produce a judgment token and discard the logits. With **mean**, we compute a weighted average of the judgment options, weighting each judgment option by the probability assigned to its token. See Appendix B for details.

3.2 Experimental Setup

Models As LLM judges, we use gpt-4o-2024-08-06 (shortened to GPT-4o) (OpenAI et al., 2024), Llama-3.1-8B-Instruct (Llama-3.1-8B) (Dubey et al., 2024), Mistral-7B-Instruct-v0.3 (Mistral-7B) (Jiang et al., 2023), and Prometheus-2-7B (Kim et al., 2024). We cover a commonly used closed-source LLM[2] (GPT-4o), as well as smaller open-source variants.

Inference Settings We prompt the LLM judge with or without **CoT** reasoning, i.e. to provide a brief explanation before stating the judgment. We use greedy decoding for CoT prompting. See Appendix C for prompts.

We softmax the judgment logits into judgment probabilities with temperature 1. We use the score space $\{1, \ldots, K = 9\}$ in this section.

Evaluation Datasets and Metrics We evaluate on RewardBench (Lambert et al., 2024) and MT-Bench (Zheng et al., 2023b), two canonical datasets for preference modeling with human annotations. Each data instance contains a prompt, a preferred response, and a dispreferred response.

[2]Some proprietary LLMs such as Anthropic Claude do not provide logit access, preventing us from including them in our experiments. In Appendix E.1, we provide partial results for DeepSeek-V3, whose trends match those of GPT-4o.

Model	Setting	Method	Reward Bench	MT-Bench
GPT-4o	point score	mode	85.1, 84.0	81.9, 80.5
		mean	87.4, **88.0**	**83.6**, 83.2
	pair score	mode	86.7, 87.4	86.2, 86.5
		mean	87.1, **87.6**	86.3, **86.8**
	pair rank	mode	88.4, 89.7	86.3, 85.6
		mean	88.6, **90.5**	**87.3**, 85.9
Llama -3.1-8B	point score	mode	69.6, 72.2	74.9, 71.9
		mean	72.7, **79.3**	78.7, **81.5**
	pair score	mode	71.7, 75.2	**82.6**, 82.4
		mean	72.1, **76.8**	82.3, 81.2
	pair rank	mode	68.9, 58.9	76.2, 63.0
		mean	**74.2**, 68.6	**80.0**, 76.5
Mistral-7B	point score	mode	60.4, 62.7	59.5, 66.2
		mean	63.8, **72.1**	62.6, **74.0**
	pair score	mode	67.3, 68.9	79.3, 79.8
		mean	68.1, **71.0**	80.0, **80.4**
	pair rank	mode	56.3, 53.8	51.5, 51.5
		mean	**63.9**, 59.1	**73.5**, 65.5
Prometheus -2-7B	point score	mode	64.3, 66.0	72.5, 73.5
		mean	64.6, **75.2**	72.1, **81.6**
	pair score	mode	**71.0**, 68.7	78.4, 80.8
		mean	70.5, 70.8	78.3, **80.9**
	pair rank	mode	59.6, 48.2	51.5, 43.0
		mean	**69.7**, 48.8	**75.4**, 33.4

Table 1: Mode vs. mean and CoT vs. no-CoT (comma-separated) accuracy results (%). For each base model+setting, we bold the best result and underline results not significantly worse ($\alpha = 0.05$). The mean outperforms the mode in 42 out of 48 cases. No-CoT outperforms CoT in 14 out of 16 cases when using the mean for pointwise or pairwise scoring.

We evaluate accuracy on the binary classification task; predicting the correct winner, a tie, or the wrong winner gets 1, 0.5, or 0 points, respectively (Lambert et al., 2024). RewardBench contains 2,985 (prompt, response 1, response 2) triplets, each labeled with the preferred response. Since MT-Bench has multiple human judgments per triplet, we compute accuracy using only triplets with unanimous human judgments (1,132 out of 1,814). See Appendix D for dataset details.

3.3 Results

Table 1 shows our main results, comparing mode vs. mean and CoT vs. no-CoT across various prompt settings and LLMs.

Mean outperforms mode The mean outperforms the mode in 42 out of 48 cases. In Table 10, we provide a subset breakdown of RewardBench and observe particularly large gains for pointwise scoring on the Reasoning subset.

CoT often harms LLM-as-a-judge For the scoring settings, no-CoT outperforms CoT in 14 out of

Model	Setting	RewardBench	MT-Bench
GPT-4o	point score	.039, .103	.041, .116
	pair score	.042, .066	.038, .064
	pair rank	.002, .065	.012, .114
Llama -3.1-8B	point score	.060, .101	.068, .093
	pair score	.054, .106	.047, .092
	pair rank	.215, .318	.186, .331

Table 2: Average standard deviation of judgment distribution, with judgment options rescaled to $[0, 1]$. Comma-separated values in each cell are with and without CoT. No-CoT always has a greater standard deviation.

16 cases when using the mean. For the pairwise ranking setting, CoT outperforms no-CoT, except with GPT-4o on RewardBench.

We interpret the harmful effect of CoT on pointwise scoring with the smaller models as being due to *sharpening*, whereby the initial entropy in the judgment is lost as the model commits to one instantiation of a reasoning trace. Table 2 confirms this trend by showing that the standard deviation of judgment distributions is lower for CoT than no-CoT. Moreover, removing CoT benefits the mean more than the mode (e.g. 69.6→72.2 for mode vs. 72.7→79.3 for mean, with Llama-3.1-8B on RewardBench), revealing the synergy between eliciting and utilizing distributional judgment.

Which setting works the best? Comparing different LLMs, we find GPT-4o performs better with pairwise judgment (e.g. 88.0 for pointwise scoring vs. 90.5 for pairwise ranking on RewardBench) as in prior work, but the smaller models often do better with pointwise judgment and rely heavily on CoT for pairwise ranking (e.g. with Prometheus-2-7B on MT-Bench, 75.4→33.4 when removing CoT from pairwise ranking, compared to 81.6 with no-CoT pointwise scoring). We believe this is because pairwise judgment demands a more powerful judge to leverage the context. Thus, in pairwise ranking with the smaller models, the reasoning gained by CoT often outweighs the distributional signal lost in the process. Nonetheless, using pairwise scoring (where assigning individual scores can be viewed as an intermediate reasoning step) rather than pairwise ranking can eliminate the need for CoT, and we recover much of the gap on RewardBench, and match or exceed pointwise performance on MT-Bench.

4 Study on Pointwise Scoring

Beyond the mode and mean discussed in prior work and the previous section, we further explore the design space of utilizing distributional output from LLM scorers.

Discrete vs. Continuous We say a method is *discrete* if it compares two score distributions by their independently assigned scores that take values in $\{1, \ldots, K\}$. Otherwise, we say it is *continuous*. Discrete scores are often desirable for interpretability (e.g. simple rubrics) but, by the pigeonhole principle, can often result in tied comparisons and fail to capture slight preferences.

Additional Metric: Mean Squared Error For our further analysis, we report mean squared error (MSE) in addition to accuracy. For target labels in $\{0, 1\}$ (a unanimously preferred response), MSE is equivalent to the Brier score. Accuracy incentivizes predicting a winner instead of a tie as long as oracle confidence is over 50%. In contrast, expected MSE is optimized by exactly predicting the oracle confidence, thus serving as a measure of a method's calibration given the judge's distributional output.

On MT-Bench, we generalize the label space to $[0, 1]$ by averaging the human judgments, thus allowing us to evaluate MSE on the full dataset. In Appendix F.1, we analyze alignment between the judgment distributions of LLMs and those of humans (as opposed to the average or majority vote).

4.1 Methods

Table 3 lists our extended methods for comparing two score distributions. We motivate the newly introduced methods below and provide details in Appendix B.1.

Users often prefer discrete methods (e.g. mode) because they are simple to interpret, even if they have lower accuracy than continuous methods (e.g. mean). This motivates the question of where the mode (the status quo method) ranks among discrete methods. To answer this question, we compare the mode to other discrete methods: rounded mean, median, and first percentile (discussed in the next paragraph).

Humans exhibit risk aversion when making decisions. They often disprefer negative outcomes more strongly than they prefer positive outcomes (Holt and Laury, 2002). However, this disposition is not captured by the measures of central tendency

Name	Description	Definition of NAME$(X_1, X_2) \in [-1, 1]$ (higher says X_1 is better, lower says X_2 is better)	Discrete or Continuous
MODE	Mode	$\text{sgn}(r_1 - r_2)$ with $r_i = \arg\max_k P(X_i = k)$	Discrete
MEAN	Mean	$\frac{\mathbb{E}(X_1 - X_2)}{\mathbb{E}\lvert X_1 - X_2\rvert + \sigma(X_1 - X_2)}$	Continuous
[MEAN]	Rounded mean	$\text{sgn}(r_1 - r_2)$ with $r_i = \arg\min_k \lvert\mathbb{E}X_i - k\rvert$	Discrete
MEDI	Median	$\text{sgn}(r_1 - r_2)$ with $r_i = Q_{X_i}(0.5)$	Discrete
1P	1st percentile	$\text{sgn}(r_1 - r_2)$ with $r_i = Q_{X_i}(0.01)$	Discrete
RAM	Risk-averse mean	MEAN$(X_1 - \sigma_-(X_1), X_2 - \sigma_-(X_2))$	Continuous
QT	Quantiles	$\int_0^1 \text{sgn}(Q_{X_1}(p) - Q_{X_2}(p))\,\mathrm{d}p$	Continuous
PS	Probability of superiority	$P(X_1 > X_2) - P(X_1 < X_2)$	Continuous

Table 3: Methods of comparing two score distributions X_1, X_2 over K score options. sgn is the sign function. $Q_X(p)$ denotes the value at the p-quantile. $\sigma(X)$ denotes the standard deviation; $\sigma_-(X) = \sqrt{\mathbb{E}[\max(\mathbb{E}X - X, 0)^2]}$ denotes the lower semi-deviation, a risk measure (Bond and Satchell, 2002).

Model	Method	RewardBench		MT-Bench	
		Acc ↑	MSE ↓	Acc ↑	MSE ↓
GPT-4o	MODE	84.0	.118	80.5	.145
	MEAN	88.0	.102	83.2	.097
	[MEAN]	85.2	.109	80.2	.146
	MEDI	84.6	.112	80.2	.142
	1P	84.3	.116	81.0	.138
	RAM	**88.4**	.100	**83.4**	**.096**
	QT	87.9	**.096**	83.2	.118
	PS	87.8	**.096**	83.3	.103
Llama -3.1-8B	MODE	72.2	.192	71.9	.142
	MEAN	79.3	.155	**81.5**	.104
	[MEAN]	75.0	.186	75.0	.145
	MEDI	73.6	.191	73.9	.142
	1P	76.0	.183	79.2	.147
	RAM	**79.9**	**.152**	81.4	**.102**
	QT	79.0	.164	81.1	.116
	PS	78.9	.161	81.4	.110

Table 4: Pointwise results over methods. No-CoT (see Table 11 for CoT). Text styling follows Table 1.

discussed so far. Thus, we investigate whether incorporating the human disposition of risk aversion into LLM-as-a-judge inference methods improves alignment with human preferences. The methods 1P (discrete) and RAM (continuous) reflect risk aversion. 1P takes an approach contrary to MODE; instead of focusing on where the most mass lies, 1P assigns a low score if there is even a 1% chance of such a low score (Siththaranjan et al., 2023). RAM is MEAN but with each distribution shifted down by its risk σ_-.

We have used MODE to represent the status quo LLM-as-a-judge inference method, which uses greedy decoding to obtain a judgment token. However, some prior works use a positive temperature, e.g. to obtain varied CoT chains (Zhang et al., 2024a), in which case a sampled judgment token is decoded rather than the mode. To account for the random nature of sampling, we design the method PS as the difference in winrates over repeated pairs of samples from the LLM judge (Siththaranjan et al., 2023). QT generalizes MEDI and 1P by averaging the comparisons over all quantiles, and can be viewed as PS but with X_1 and X_2 positively monotonically correlated.

4.2 Results

Main Takeaways

- Table 4 shows that the top pointwise methods are the continuous ones (MEAN, RAM, QT, PS), in both accuracy and MSE, indicating that they should be chosen over discrete methods.
- Even among discrete methods, MODE has the lowest accuracy in 3 out of the 4 cases, indicating that the mode is a suboptimal choice even if discrete scores are desired.
- 1P often outperforms MEDI (e.g. 79.2 vs 73.9 accuracy with Llama-3.1-8B on MT-Bench), and RAM slightly outperforms MEAN (e.g. 79.9 vs. 79.3 accuracy with Llama-3.1-8B on RewardBench), suggesting that risk aversion can be helpful for preference modeling.

Study: Score Granularity and Ties We show here that ties explain the finding above that the discrete methods fall behind the continuous ones, and we experiment with score granularity as a remedy.

Table 5 shows that the discrete methods predict ties on a significant number of instances, on which MEAN is still able to achieve nontrivial accuracy. On the other hand, we find that on instances where a discrete method does not predict a tie, it has similar accuracy to MEAN (not shown; see Table 12), indicating that the performance gap is well explained by ties. Nonetheless, tie behavior varies by method;

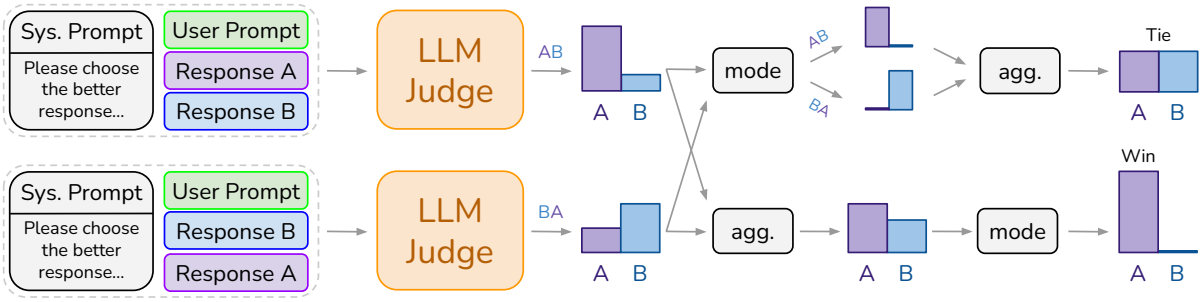

Figure 2: Comparing pairwise LLM-as-a-judge prediction based on **when** to aggregate the two judgments, one from each response pair presentation order. Pre- vs. post-aggregation (bottom vs. top in figure) can be likened to mean vs. mode, as the former aggregates at the distribution level while the latter aggregates at the text level (if mode is used).

Model	Method	Tie rate		MEAN's accuracy	
		$K = 9$	$K = 99$	$K = 9$	$K = 99$
GPT-4o	MODE	.17	.20	72	73
	[MEAN]	.16	.03	67	53
	MEDI	.17	.09	70	62
	1P	.16	.08	66	60
Llama -3.1-8B	MODE	.35	.24	69	70
	[MEAN]	.26	.07	64	61
	MEDI	.29	.11	67	67
	1P	.23	.08	65	57

Table 5: Tie analysis for discrete pointwise methods on RewardBench using no-CoT (see Table 12 for CoT and Table 13 for MT-Bench). We report results with two score granularity levels (K). Tie rate is the proportion of instances where the method predicts a tie, over which we report MEAN's accuracy (%); excess of 50% or 75% indicates room for improving accuracy or MSE, respectively.

	Method	RewardBench		MT-Bench	
		Acc ↑	MSE ↓	Acc ↑	MSE ↓
GPT-4o	MODE	$81.7_{-2.3}$	$.134_{+.016}$	$78.4_{-2.1}$	$.158_{+.013}$
	MEAN	$\mathbf{86.7}_{-1.3}$	$.108_{+.006}$	$\underline{82.9}_{-0.3}$	$.099_{+.002}$
	[MEAN]	$\underline{86.5}_{+1.3}$	$.127_{+.018}$	$\underline{82.7}_{+2.5}$	$.182_{+.036}$
	MEDI	$85.2_{+0.6}$	$.126_{+.014}$	$81.5_{+1.3}$	$.170_{+.028}$
	1P	$\underline{86.4}_{+2.1}$	$.116_{+.000}$	$\underline{82.7}_{+1.7}$	$.165_{+.027}$
	RAM	$\mathbf{86.7}_{-1.7}$	$\mathbf{.104}_{+.004}$	$\mathbf{83.0}_{-0.4}$	$\mathbf{.098}_{+.002}$
	QT	$\underline{86.6}_{-1.3}$	$.114_{+.018}$	$\underline{82.7}_{-0.5}$	$.147_{+.029}$
	PS	$\underline{86.6}_{-1.2}$	$\underline{.105}_{+.009}$	$82.4_{-0.9}$	$.107_{+.004}$
Llama-3.1-8B	MODE	$72.0_{-0.2}$	$.221_{+.029}$	$75.1_{+3.2}$	$.169_{+.027}$
	MEAN	$\underline{79.3}_{+0.0}$	$.156_{+.001}$	$\underline{81.3}_{-0.2}$	$\underline{.103}_{-.001}$
	[MEAN]	$78.5_{+3.5}$	$.198_{+.012}$	$80.7_{+5.7}$	$.180_{+.035}$
	MEDI	$76.5_{+2.9}$	$.207_{+.016}$	$80.1_{+6.2}$	$.161_{+.019}$
	1P	$78.5_{+2.5}$	$.195_{+.012}$	$\underline{81.5}_{+2.3}$	$.177_{+.030}$
	RAM	$\mathbf{79.7}_{-0.2}$	$\mathbf{.152}_{+.000}$	$\underline{81.1}_{-0.3}$	$\mathbf{.102}_{+.000}$
	QT	$78.7_{-0.3}$	$.177_{+.013}$	$81.3_{+0.2}$	$.143_{+.027}$
	PS	$78.6_{-0.3}$	$.163_{+.002}$	$\mathbf{81.8}_{+0.4}$	$.111_{+.001}$

Table 6: Pointwise results over methods ($K = 99$). No-CoT (see Table 14 for CoT). Subscripts denote change from $K = 9$ (Table 4). Text styling follows Table 1.

MODE has the most ties and the highest MEAN accuracy, amounting to the most untapped signal for determining the better response.

Table 5 further shows that granularizing the score space from $K = 9$ to $K = 99$ improves the expressivity of the discrete methods (except for MODE), drastically reducing the rate of ties, while MEAN accuracies remain similar or decrease.

Table 6 expands on the comparison between $K = 99$ and $K = 9$, reporting results from the same setting in Table 4 except for the granularity scale. Consistent with our motivation, the discrete methods (except for MODE) improve in accuracy, rivaling the continuous methods. Although MODE somewhat makes up for its low accuracy with a lower MSE than most other discrete methods on MT-Bench, it suffers the highest MSE on RewardBench.

Taken together, Tables 4-6 show that even in use cases where discrete scores are desired, one should consider alternatives to the mode.

Sensitivity to Score Granularity In Appendix F.2, we analyze the sensitivity of different methods to score granularity, and find theoretically and empirically that the mode is the most sensitive method.

5 Study on Pairwise Ranking

The judgment styles in Section 3's overview were scoring (Section 4) and ranking. In this section, we analyze design decisions for pairwise ranking, and in Section 6 listwise ranking.

5.1 Design Decisions

As we explain below, the pairwise ranking experiments in Table 1 used Likert-2, post-aggregation for the mode, and pre-aggregation for the mean. We now consider alternative choices (see Appendix B.2.2 for details).

Center	Agg. Time	RewardBench		MT-Bench	
		Acc ↑	MSE ↓	Acc ↑	MSE ↓
mode	post	56.7	**.240**	57.5	**.192**
	pre	**73.1**	.265	**78.1**	.222
median	post	56.8	**.240**	57.5	**.192**
	pre	**72.9**	.261	**78.0**	.218
mean	post	73.2	**.207**	**78.2**	**.144**
	pre	**73.2**	.222	78.1	.155

Table 7: Pairwise ranking results over methods using Likert-3 comparing pre- and post-aggregation. All methods use Llama-3.1-8B, CoT (see Table 15 for GPT-4o and no-CoT). Text styling follows Table 1.

K	RewardBench		MT-Bench	
	Acc ↑	MSE ↓	Acc ↑	MSE ↓
2	**74.2**, 68.6	**.187**, .214	**80.0**, 76.5	**.126**, .135
3	73.2, 66.3	.222, .240	78.1, 70.8	.155, .155
5	70.0, 58.5	.215, .234	77.1, 64.8	.142, .153

Table 8: Pairwise ranking results over Likert-K scales, using pre-aggregation mean. Llama-3.1-8B, CoT (see Table 16 for GPT-4o and no-CoT). Text styling follows Table 1.

Timing of aggregation and measure of central tendency Pairwise judgment suffers from position bias, i.e. the LLM judge's sensitivity to the order in which the evaluated texts are presented, which is usually addressed by prompting the LLM judge twice, once for each order of presentation (Lee et al., 2024a). We examine the remaining question of whether to aggregate the two judgments *before or after* computing the measure of central tendency (mode, median, or mean), as shown in Figure 2. Pre- vs. post-aggregation can be likened to mean vs. mode, as the former aggregates at the distribution level while the latter aggregates at the text level (if mode is used).

Granularity We prompt the judge to express its preference on a K-point Likert scale: $[>, <]$ (Likert-2), $[>, =, <]$ (Likert-3), or $[\gg, >, =, <, \ll]$ (Likert-5) (Liu et al., 2024b). In the prompt (Appendix C.3), the symbols are assigned descriptions such as "significantly better" and "slightly better".

5.2 Methods Results

Table 7 shows that accuracy depends little on the measure of central tendency and mostly on when we aggregate, with aggregating first leading to higher accuracy (as much as 56.7→73.1 using the mode on RewardBench). Considering that the timing of aggregation does not affect accuracy if the two runs agree, this shows that **even for inconsistent judgments caused by position bias, there is still valuable signal in the relative magnitudes of preference that we can leverage by aggregating first**.

On the other hand, an intuitive explanation for why the measure of central tendency has little effect on accuracy is that the judgment space is small, so there is high correlation between the signs of the measures of central tendency. In fact, they are equivalent in the pre-aggregation Likert-2 setting.

Although aggregating first improves accuracy, it harms MSE for mode and median, which we attribute to the volatile prediction of a binary winner when faced with the uncertain situation of positional inconsistency. Nevertheless, the mean (with either pre- or post-aggregation) is among the top accuracy methods while outperforming all other methods on MSE. This demonstrates the calibration benefit of using the judgment distribution to produce a continuous prediction.

With GPT-4o (not shown; see Tables 15, 16), MSE is always minimized with no-CoT, highlighting the discord between CoT's sharpening effect and calibration. In Appendix F.3, we further analyze position bias and find that CoT increases the occurrence of severe position bias.

5.3 Granularity Results

Table 8 compares the Likert scales used in the pairwise ranking prompt. We find that **Likert-2 performs the best overall**, in line with the AlpacaEval methodology (Dubois et al., 2024) but deviating from WB-Reward and Arena-Hard-Auto (Lin et al., 2024; Li et al., 2024b), which use Likert-5.

6 Listwise Judgment

Listwise judgment is not as prevalent as pointwise or pairwise judgment, but it offers efficiency (Zhu et al., 2024) while granting the judge the maximal context for comparison (Buyl et al., 2023).

6.1 Judgment Spaces and Methods

We consider two prompts for eliciting listwise preferences over N texts (Appendix C.4). Prompt 1 is the one proposed by Zhu et al. (2024), which prompts to produce all $\binom{N}{2}$ pairwise preferences and then aggregate them into a sorted list. Prompt 2 skips the intermediate pairwise step and asks to directly produce the list (Liu et al., 2023a; Qin

et al., 2023). We can then extract all pairwise[3] preferences from one of the following judgment spaces using the mode (textual output) or the mean (distributional output).

- INTERM (Prompt 1): Intermediate pairwise preferences (Likert-3, no-CoT, only one of the two presentation orders), which we view as the reasoning process leading to the list. This efficiently extends pairwise ranking to the listwise setting, similar to batch prompting (Cheng et al., 2023).
- LIST (Prompt 1): Final list. For MEAN, we use the probability distribution over text identifiers at each rank, inspired by Zhuang et al. (2023); Reddy et al. (2024). Specifically, at rank r, denote $p_r(i)$ as the probability of decoding text (identifier) i. Decoding text i at rank r implies that any text j not yet decoded will be decoded at a later rank and is thus worse than text i, and vice versa. Hence, we define $\text{MEAN}(i, j) \in [0, 1]$ as the average of $\frac{p_r(i)}{p_r(i)+p_r(j)}$ over the ranks r until i or j is decoded.
- DIRECT LIST (Prompt 2): LIST but with Prompt 2 (no intermediate pairwise step).

6.2 Experimental Setup

Models Due to the context length required for listwise ranking and the difficulty of the task, we limit our evaluation to GPT-4o. In preliminary experiments, we found poor performance with the smaller models, but in Appendix E.1 we show that DeepSeek-V3 exhibits similar trends to GPT-4o.

Datasets We evaluate on Nectar (Zhu et al., 2024), RM-Bench (Liu et al., 2024c), and MT-Bench (Zheng et al., 2023b).

From Nectar, we use a random subset of 1,000 prompts, each with 7 responses. We discard the GPT-4 judgments included in the dataset and collect our own silver labels using GPT-4o with pairwise ranking (Likert-5, no-CoT, pre-aggregation, mean). RM-Bench contains 1,327 prompts, each with 3 chosen and 3 rejected responses, yielding 9 pairwise preference labels. MT-Bench contains 160 prompts, each with 6 responses. See Appendix D for dataset details.

6.3 Results

Table 9 compares mode and mean in the listwise judgment spaces. The two methods have similar accuracy, but the mean has much lower MSE.

[3]We retain the pairwise evaluation setup from previous sections; see Appendix D.1 for discussion.

Space	Method	Nectar		RM-Bench		MT-Bench	
		Acc	MSE	Acc	MSE	Acc	MSE
interm	mode	**80.4**	.155	62.1	.339	**80.8**	.201
	mean	**80.4**	**.048**	**62.5**	**.243**	80.7	**.121**
list	mode	**82.2**	.156	**62.4**	.376	**83.7**	.189
	mean	82.0	**.105**	61.7	**.317**	83.5	**.157**
direct list	mode	86.1	.138	**69.9**	.301	**86.8**	.168
	mean	**86.4**	**.087**	69.4	**.267**	85.9	**.133**

Table 9: Listwise results (GPT-4o). Text styling follows Table 1.

We find DIRECT LIST to be the most accurate judgment space (notably, outperforming pointwise scoring on MT-Bench; see Table 4), while INTERM has the lowest MSE. We hypothesize that DIRECT LIST outperforms LIST due to the intermediate pairwise comparisons playing a similar role to CoT in the pointwise and pairwise settings, where distributional output is captured most intactly without it. Even so, in Appendix F.3 we find DIRECT LIST to suffer the most position bias, consistent with Zhu et al. (2024), while INTERM has the least.

7 Conclusion and Recommendations

We comprehensively evaluated design choices for leveraging LLM judges' distributional output. For pointwise scoring, we showed that continuous methods (e.g. mean) outperform discrete methods (especially the mode) due to ties. For pairwise ranking, we related the mean vs. mode comparison to pre- vs. post-aggregation of the two presentation orders' judgments. Although smaller LLM judges suffer heavily from inconsistent judgments due to position bias, pre-aggregation effectively leverages the relative magnitudes of preference.

We showed that CoT collapses the spread of the judgment distribution, often hurting performance. This applies even to the challenging setting of listwise ranking, where accuracy was maximized by directly predicting the list without an intermediate pairwise step. We hope that highlighting this limitation of CoT encourages the development of reasoning mechanisms that preserve output diversity and calibration for judgment and other subjective or open-ended tasks.

Recommendations We summarize our findings into guidelines for choosing judgment settings. Large judges like GPT-4o should use pairwise ranking no-CoT, or direct listwise ranking as an efficient alternative. Smaller judges like Llama-3.1-8B should use pointwise scoring no-CoT. The mean

should be used instead of the mode, but these setting guidelines apply even if one uses the mode.

Limitations

Downstream Performance In this paper, we evaluate LLM-as-a-judge design decisions by their performance on preference modeling datasets. However, this setup may not reveal downstream impacts. We do not explore the impact of distributional judgments on reinforcement learning from AI feedback (RLAIF) (Lee et al., 2024a) or human decision making.

Training Our experiments involve off-the-shelf LLMs as judges without specific tuning. We do not explore training LLM judges to express distributional judgments (Saad-Falcon et al., 2024). Similarly, we exclude distributional reward models (Dorka, 2024) from the scope of our study.

CoT We conclude from our results that CoT often hurts judgment performance. However, we only consider one prompt design per setting for eliciting CoT reasoning (Appendix C) and do not perform prompt optimization. Furthermore, we do not consider more extensive test-time scaling, such as asking the judge to produce its own reference response (Zheng et al., 2023b) or aggregating many CoT judgment runs (Zhang et al., 2024a; Stureborg et al., 2024).

Natural Language Judgments A valuable aspect of LLM-as-a-judge is its ability to augment judgments with interpretable rationales (Mahan et al., 2024; Byun et al., 2024; Ye et al., 2024b; Cao et al., 2024). However, the distributional judgments we consider here are limited to those that are easily quantifiable, and we do not propose methods for leveraging distributional output over natural language feedback. While it is possible to continue decoding a rationale after the judgment, the rationale will be conditioned on the decoded judgment and not reflect the distribution over the unchosen judgment options. One approach could be to decode several rationales, each conditioned on a different judgment option.

References

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova Dassarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv*, abs/2204.05862.

Shaun A. Bond and Stephen E. Satchell. 2002. Statistical properties of the sample semi-variance. *Applied Mathematical Finance*, 9:219 – 239.

Maarten Buyl, Paul Missault, and Pierre-Antoine Sondag. 2023. Rankformer: Listwise learning-to-rank using listwide labels. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Ju-Seung Byun, Jiyun Chun, Jihyung Kil, and Andrew Perrault. 2024. Ares: Alternating reinforcement learning and supervised fine-tuning for enhanced multi-modal chain-of-thought reasoning through diverse ai feedback. *ArXiv*, abs/2407.00087.

Maosong Cao, Alexander Lam, Haodong Duan, Hongwei Liu, Songyang Zhang, and Kai Chen. 2024. Compassjudger-1: All-in-one judge model helps model evaluation and evolution.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shan Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *ArXiv*, abs/2308.07201.

Zhoujun Cheng, Jungo Kasai, and Tao Yu. 2023. Batch prompting: Efficient inference with large language model apis. In *Conference on Empirical Methods in Natural Language Processing*.

Cheng-Han Chiang and Hunghuei Lee. 2023. A closer look into automatic evaluation using large language models. *ArXiv*, abs/2310.05657.

Brian Conrey, James Gabbard, Katie Grant, Andrew Liu, and Kent E. Morrison. 2013. Intransitive dice. *Mathematics Magazine*, 89:133 – 143.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *ArXiv*, abs/2310.01377.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, et al. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Nicolai Dorka. 2024. Quantile regression for distributional reward models in rlhf. *ArXiv*, abs/2409.10164.

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. *Preprint*, arXiv:2407.21783.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

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

Xidong Feng, Ziyu Wan, Mengyue Yang, Ziyan Wang, Girish A. Koushiks, Yali Du, Ying Wen, and Jun Wang. 2024. Natural language reinforcement learning. *ArXiv*, abs/2402.07157.

Mark Finkelstein and Edward O. Thorp. 2006. Nontransitive dice with equal means.

Nate Gillman, Daksh Aggarwal, Michael Freeman, Saurabh Singh, and Chen Sun. 2024. Fourier head: Helping large language models learn complex probability distributions.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *Preprint*, arXiv:2412.06769.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 2024. Llm-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Annual Meeting of the Association for Computational Linguistics*.

Charles A. Holt and Susan K. Laury. 2002. Risk aversion and incentive effects. *American Economic Review*, 92(5):1644–1655.

Zhengyu Hu, Jieyu Zhang, Zhihan Xiong, Alexander J. Ratner, Hui Xiong, and Ranjay Krishna. 2024. Language model preference evaluation with multiple weak evaluators. *ArXiv*, abs/2410.12869.

Hawon Jeong, ChaeHun Park, Jimin Hong, and Jaegul Choo. 2024. Prepair: Pointwise reasoning enhance pairwise evaluating for robust instruction-following assessments. *arXiv preprint arXiv:2406.12319*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

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

Alexander Y Klimenko. 2015. Intransitivity in theory and in the real world. *Entropy*, 17(6):4364–4412.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2023. Benchmarking cognitive biases in large language models as evaluators. *arXiv preprint arXiv:2309.17012*.

Sachin Kumar, Chan Young Park, Yulia Tsvetkov, Noah A. Smith, and Hanna Hajishirzi. 2024. Compo: Community preferences for language model personalization.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

Gyeong-Geon Lee, Ehsan Latif, Xuansheng Wu, Ninghao Liu, and Xiaoming Zhai. 2023. Applying large language models and chain-of-thought for automatic scoring. *ArXiv*, abs/2312.03748.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. 2024a. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. In *Forty-first International Conference on Machine Learning*.

Noah Lee, Jiwoo Hong, and James Thorne. 2024b. Evaluating the consistency of llm evaluators.

Dexun Li, Cong Zhang, Kuicai Dong, Derrick-Goh-Xin Deik, Ruiming Tang, and Yong Liu. 2024a. Aligning crowd feedback via distributional preference reward modeling. *ArXiv*, abs/2402.09764.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024b. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Weixian Waylon Li, Yftah Ziser, Yifei Xie, Shay B. Cohen, and Tiejun Ma. 2024c. Tsprank: Bridging pairwise and listwise methods with a bilinear travelling salesman model.

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`.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *ArXiv*, abs/2305.20050.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2024. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*.

Minqian Liu, Ying Shen, Zhiyang Xu, Yixin Cao, Eunah Cho, Vaibhav Kumar, Reza Ghanadan, and Lifu Huang. 2023a. X-eval: Generalizable multi-aspect text evaluation via augmented instruction tuning with auxiliary evaluation aspects. In *North American Chapter of the Association for Computational Linguistics*.

Ryan Liu, Jiayi Geng, Addison J. Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L. Griffiths. 2024a. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse.

Shang Liu, Yu Pan, Guanting Chen, and Xiaocheng Li. 2024b. Reward modeling with ordinal feedback: Wisdom of the crowd.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2024c. Rm-bench: Benchmarking reward models of language models with subtlety and style.

Yinhong Liu, Zhijiang Guo, Tianya Liang, Ehsan Shareghi, Ivan Vuli'c, and Nigel Collier. 2024d. Aligning with logic: Measuring, evaluating and improving logical consistency in large language models. *ArXiv*, abs/2410.02205.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulic, Anna Korhonen, and Nigel Collier. 2024e. Aligning with human judgement: The role of pairwise preference in large language model evaluators. *arXiv preprint arXiv:2403.16950*.

Yixin Liu, Kejian Shi, Alexander Fabbri, Yilun Zhao, Peifeng Wang, Chien-Sheng Wu, Shafiq Joty, and Arman Cohan. 2024f. Reife: Re-evaluating instruction-following evaluation. *ArXiv*, abs/2410.07069.

Adian Liusie, Potsawee Manakul, and Mark JF Gales. 2023. Zero-shot nlg evaluation through pairware comparisons with llms. *arXiv preprint arXiv:2307.07889*.

Adian Liusie, Vatsal Raina, Yassir Fathullah, and Mark J. F. Gales. 2024. Efficient llm comparative assessment: A product of experts framework for pairwise comparisons. *ArXiv*, abs/2405.05894.

Charles Lovering, Michael Krumdick, Viet Dac Lai, Nilesh Kumar, Varshini Reddy, Rik Koncel-Kedziorski, and Chris Tanner. 2024. Are language model logits calibrated? *ArXiv*, abs/2410.16007.

Michal Lukasik, Harikrishna Narasimhan, Aditya Krishna Menon, Felix X. Yu, and Sanjiv Kumar. 2024. Regression aware inference with llms. In *Conference on Empirical Methods in Natural Language Processing*.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. 2024. Generative reward models. *arXiv preprint arXiv:2410.12832*.

Nicole Meister, Carlos Guestrin, and Tatsunori Hashimoto. 2024. Benchmarking distributional alignment of large language models.

Niklas Muennighoff, Qian Liu, Qi Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro von Werra, et al. 2023. Octopack: Instruction tuning code large language models. *ArXiv*, abs/2308.07124.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, et al. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Vishakh Padmakumar, Chuanyang Jin, Hannah Rose Kirk, and He He. 2024. Beyond the binary: Capturing diverse preferences with reward regularization. *ArXiv*, abs/2412.03822.

John W. Payne. 1976. Task complexity and contingent processing in decision making: An information search and protocol analysis. *Organizational Behavior and Human Performance*, 16(2):366–387.

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. 2024. Personalizing reinforcement learning from human feedback with variational preference learning. *ArXiv*, abs/2408.10075.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Zhen Qin, Junru Wu, Jiaming Shen, Tianqi Liu, and Xuanhui Wang. 2024. Lampo: Large language models as preference machines for few-shot ordinal classification. *ArXiv*, abs/2408.03359.

Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is llm-as-a-judge robust? investigating universal adversarial attacks on zero-shot llm assessment. *arXiv preprint arXiv:2402.14016*.

Revanth Gangi Reddy, JaeHyeok Doo, Yifei Xu, Md Arafat Sultan, Deevya Swain, Avirup Sil, and Heng Ji. 2024. First: Faster improved listwise reranking with single token decoding. *arXiv preprint arXiv:2406.15657*.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2023. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *ArXiv*, abs/2308.01263.

Jon Saad-Falcon, Rajan Vivek, William Berrios, Nandita Shankar Naik, Matija Franklin, Bertie Vidgen, Amanpreet Singh, Douwe Kiela, and Shikib Mehri. 2024. Lmunit: Fine-grained evaluation with natural language unit tests. *ArXiv*, abs/2412.13091.

Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2023. Branch-solve-merge improves large language model evaluation and generation. *ArXiv*, abs/2310.15123.

Richard P. Savage. 1994. The paradox of nontransitive dice. *American Mathematical Monthly*, 101:429–436.

Markus Schulze. 2011. A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method. *Social Choice and Welfare*, 36(2):267–303.

Ammar Shaikh, Raj Abhijit Dandekar, Sreedath Panat, and Raj Abhijit Dandekar. 2024. Cbeval: A framework for evaluating and interpreting cognitive biases in llms. *ArXiv*, abs/2412.03605.

Lin Shi, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the judges: A systematic investigation of position bias in pairwise comparative assessments by llms. *arXiv preprint arXiv:2406.07791*.

Anand Siththaranjan, Cassidy Laidlaw, and Dylan Hadfield-Menell. 2023. Distributional preference learning: Understanding and accounting for hidden context in rlhf. *arXiv preprint arXiv:2312.08358*.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell L Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, et al. 2024. Position: A roadmap to pluralistic alignment. In *Forty-first International Conference on Machine Learning*.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *ArXiv*, abs/2409.12183.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *ArXiv*, abs/2405.01724.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. 2024. Judgebench: A benchmark for evaluating llm-based judges. *arXiv preprint arXiv:2410.12784*.

Raphael Tang, Xinyu Crystina Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2023. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. In *North American Chapter of the Association for Computational Linguistics*.

T. N. Tideman. 1987. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *ArXiv*, abs/2402.10200.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2023. Do-not-answer: A dataset for evaluating safeguards in llms. *ArXiv*, abs/2308.13387.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2024a. Helpsteer2-preference: Complementing ratings with preferences. *ArXiv*, abs/2410.01257.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024b. Helpsteer2: Open-source dataset for training top-performing reward models. *ArXiv*, abs/2406.08673.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Michihiro Yasunaga, Leonid Shamis, Chunting Zhou, Andrew Cohen, Jason Weston, Luke Zettlemoyer, and Marjan Ghazvininejad. 2024. Alma: Alignment with minimal annotation.

Chen Ye, Wei Xiong, Yuheng Zhang, Nan Jiang, and Tong Zhang. 2024a. Online iterative reinforcement learning from human feedback with general preference model.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2023. Flask: Fine-grained language model evaluation based on alignment skill sets. *arXiv preprint arXiv:2307.10928*.

Ziyi Ye, Xiangsheng Li, Qiuchi Li, Qingyao Ai, Yujia Zhou, Wei Shen, Dong Yan, and Yiqun Liu. 2024b. Beyond scalar reward model: Learning generative judge from preference data. *ArXiv*, abs/2410.03742.

Krystian Zawistowski. 2024. Unused information in token probability distribution of generative llm: improving llm reading comprehension through calculation of expected values. *arXiv preprint arXiv:2406.10267*.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2023. Evaluating large language models at evaluating instruction following. *arXiv preprint arXiv:2310.07641*.

Yuanzhao Zhai, Zhuo Zhang, Kele Xu, Hanyang Peng, Yue Yu, Dawei Feng, Cheng Yang, Bo Ding, and Huaimin Wang. 2024. Online self-preferring language models. *ArXiv*, abs/2405.14103.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024a. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*.

Michael JQ Zhang, Zhilin Wang, Jena D Hwang, Yi Dong, Olivier Delalleau, Yejin Choi, Eunsol Choi, Xiang Ren, and Valentina Pyatkin. 2024b. Diverging preferences: When do annotators disagree and do models know? *arXiv preprint arXiv:2410.14632*.

Xu Zhang, Xunjian Yin, and Xiaojun Wan. 2024c. Contrasolver: Self-alignment of language models by resolving internal preference contradictions. *ArXiv*, abs/2406.08842.

Yifan Zhang, Ge Zhang, Yue Wu, Kangping Xu, and Quanquan Gu. 2024d. General preference modeling with preference representations for aligning language models. *ArXiv*, abs/2410.02197.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, et al. 2023a. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *ArXiv*, abs/2309.11998.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, Karthik Ganesan, Wei-Lin Chiang, Jian Zhang, and Jiantao Jiao. 2024. Starling-7b: Improving helpfulness and harmlessness with rlaif. In *First Conference on Language Modeling*.

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

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and G. Zuccon. 2023. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

A Related Work: LLM-as-a-judge Settings

LLM-as-a-judge has been used in pointwise (evaluating one response at a time), pairwise (two), and listwise (many) settings.

Pairwise judgment has the advantage of grounding each evaluated response in the other, creating for a more calibrated task and leading to better agreement with humans (Liusie et al., 2023). However, due to intransitivity in pairwise preferences (Liu et al., 2024e), the cost to sort N texts is $O(N^2)$ rather than $O(N \log N)$, compared to $O(N)$ in the pointwise setting. In addition, pairwise comparisons are susceptible to position bias (Shi et al., 2024), which often must be addressed by running both orders and aggregating the results (Zeng et al., 2023; Li et al., 2024b). Pairwise comparisons have also been shown to be more biased toward superficial traits such as verbosity and tone, in both LLM and human judges (Payne, 1976; Jeong et al., 2024), although pointwise scoring more easily falls victim to adversarial responses (Raina et al., 2024).

The listwise setting provides the maximal amount of context to the judge while keeping the same compute complexity as the pointwise setting. However, the judgment task becomes much more challenging (Qin et al., 2023; Koo et al., 2023), especially due to the amplified position bias (Zhu et al., 2024), and the combinatorially many orders makes it severely more daunting to address than in the pairwise case (Tang et al., 2023; Qin et al., 2024). To mitigate position bias, Zhu et al. (2024) leverage intermediate pairwise preferences for aggregation into a sorted list. Zhuang et al. (2023); Reddy et al. (2024) use the distribution from a single output token for listwise passage reranking, a related task to LLM-as-a-judge.

B Methods

Let A_1 and A_2 be two texts to compare. We describe the methods of predicting a value in $[-1, 1]$ that signifies the advantage of A_1 over A_2. For accuracy, we take the sign of the prediction. For MSE, we rescale predictions from $[-1, 1]$ to $[0, 1]$.

The prompts for the various settings are in Appendix C.

B.1 Pointwise Methods

We elaborate on the pointwise methods introduced in Section 4.1. The LLM judge independently judges A_1 and A_2, producing score distributions over $\{1, \ldots, K\}$ for an integer K that define independent random variables X_1 and X_2, which are used to compare A_1 and A_2.

The methods are invariant to scaling and translating the judgment space, and all methods that do not take expectations $\mathbb{E}$ (which assumes linearity) are invariant to applying a positive monotone trans-

formation to the judgment space. The methods are all equivalent if the distributions are deterministic, thus our experiments evaluate their ability to leverage the LLM judge's *distributional* output.

The denominator in MEAN normalizes it into $[-1, 1]$, similar to $\text{sgn}(x) = \frac{x}{|x|}$, taking $\frac{0}{0}$ to be 0. The σ term lowers the magnitude of the prediction in the presence of uncertainty in a continuous manner. Specifically, let $k, k' \in \{1, \ldots, K\}$ with $k \neq k'$. For $\epsilon \in [0, 1]$, let X_1 have a two-point distribution $(1-\epsilon)\delta_k + \epsilon\delta_{k'}$ and let X_2 have a deterministic distribution δ_k. Then $\text{MEAN}(X_1, X_2)$ as a function of ϵ is continuous at $\epsilon = 0$.

For MEAN, RAM, and PS, we assume X_1 and X_2 to be independent, but QT can be viewed as PS but with X_1 and X_2 positively monotonically correlated. By incorporating the sign function, QT and PS are less sensitive to extremal values than MEAN. In addition, QT and PS can model intransitive preferences, e.g. $\text{PS}(X_1, X_2)$, $\text{PS}(X_2, X_3) > 0 \not\Rightarrow \text{PS}(X_1, X_3) > 0$, which we analyze in Appendix F.4.

B.2 Pairwise Methods

In the pairwise setting, we consider two prompting approaches for jointly evaluating the two texts A_1 and A_2: scoring both texts (§B.2.1) and expressing a preference (§B.2.2).

To account for position bias, we prompt the LLM judge once for each order of presentation. For an order $\mathbf{o} \in \mathbf{O} := \{(1, 2), (2, 1)\}$, we use $\mathbf{o}$ to denote dependence on the order (A_{o_1}, A_{o_2}) in which the texts appear in the prompt.

B.2.1 Pairwise Scoring

For a given order $\mathbf{o}$, the LLM judge scores the two texts jointly in the same run. If we could obtain the joint distribution $P(X^{\mathbf{o}}_{o_1}, X^{\mathbf{o}}_{o_2})$, we could compute the marginals and use any method in Table 3. However, the judge first outputs the score for A_{o_1} and conditions on it when outputting the score for A_{o_2}, i.e. $X^{\mathbf{o}}_{o_1}$ and $X^{\mathbf{o}}_{o_2}$ are not independent. Thus, the full joint distribution $P(x_{o_1}, x_{o_2}) = P(x_{o_1})P(x_{o_2} \mid x_{o_1})$ can only be obtained by injecting each $x_{o_1} \in \{1, \ldots, K\}$ into the context to access $P(x_{o_2} \mid x_{o_1})$. This is feasible with local models but not with API-access models where inference cost scales with K. Hence, we stick to a single run and condition on the greedily decoded $x_{o_1} = \arg\max_k P(X^{\mathbf{o}}_{o_1} = k)$, giving us

$$X^{\mathbf{o}}_{\Delta} \overset{d}{=} (X^{\mathbf{o}}_1 - X^{\mathbf{o}}_2) \mid (X^{\mathbf{o}}_{o_1} = x_{o_1})$$

as a proxy for the score difference $X^{\mathbf{o}}_1 - X^{\mathbf{o}}_2$. Semantically, $X^{\mathbf{o}}_{\Delta}$ is symmetric (i.e. there should be no prior preference for A_1 or A_2), so we would like our scalar judgment to be some measure of *central* tendency (mode, median, or mean). As shown in Figure 2, we also have the choice of whether to aggregate the judgments from the two orders of presentation *before or after* computing the measure of central tendency.

For pre-aggregation, we simply take the mixture distribution,

$$P(X_{\Delta} = \delta) := \frac{1}{|\mathbf{O}|} \sum_{\mathbf{o} \in \mathbf{O}} P(X^{\mathbf{o}}_{\Delta} = \delta)$$

for all $\delta \in \{-(K-1), \ldots, K-1\}$, leaving more sophisticated approaches such as the convolution and Wasserstein barycenter for future study:

$$\begin{aligned}
\text{AGG-MODE} &:= \text{sgn}(\text{mode}(X_{\Delta})) \\
\text{AGG-MEDI} &:= \text{sgn}(\text{median}(X_{\Delta})) \\
\text{AGG-MEAN} &:= \text{MEAN}(X_{\Delta}),
\end{aligned}$$

where MEAN is defined as in Table 3, overloaded to take a single argument representing $X_1 - X_2$.

For post-aggregation, we sum the two scalar judgments from the two orders and normalize:

$$\begin{aligned}
\text{MODE-AGG} &:= \frac{\sum_{\mathbf{o} \in \mathbf{O}} \text{mode}(X^{\mathbf{o}}_{\Delta})}{\sum_{\mathbf{o} \in \mathbf{O}} |\text{mode}(X^{\mathbf{o}}_{\Delta})|} \\
\text{MEDI-AGG} &:= \frac{\sum_{\mathbf{o} \in \mathbf{O}} \text{median}(X^{\mathbf{o}}_{\Delta})}{\sum_{\mathbf{o} \in \mathbf{O}} |\text{median}(X^{\mathbf{o}}_{\Delta})|} \\
\text{MEAN-AGG} &:= \frac{1}{|\mathbf{O}|} \sum_{\mathbf{o} \in \mathbf{O}} \text{MEAN}(X^{\mathbf{o}}_{\Delta}),
\end{aligned}$$

taking $\frac{0}{0} := 0$.

B.2.2 Pairwise Ranking

We prompt the LLM judge to express its preference on a K-point Likert scale: $[>, <]$ (Likert-2), $[>, =, <]$ (Likert-3), or $[\gg, >, =, <, \ll]$ (Likert-5). Assigning the symbols $[\gg, >, =, <, \ll]$ the numerical values $[2, 1, 0, -1, -2]$, the methods for pairwise ranking then follow those above for pairwise scoring. We remark that the 'mode' and 'mean' for pairwise scoring and pairwise ranking in Table 1 are with post-aggregation and pre-aggregation, respectively.

B.3 Listwise Methods

The listwise methods are introduced in Section 6.1.

C Prompts

We present representative example prompts to illustrate the different settings. The prompts are adapted from MT-Bench (Zheng et al., 2023b). Auxiliary modifications are not shown, such as the prompt for second-turn evaluation in MT-Bench.

C.1 Judgment Extraction Details

To identify the token position containing the judgment, we use the specified format when available (e.g. "Rating A: `{rating_a}`." in pairwise scoring). Otherwise, we use the latest token position with more than 0.5 total probability assigned to judgment tokens. If no valid token is found, we default the judgment to the minimum score of 1 in the scoring setting, and to a tie in the ranking setting. (For Nectar experiments, we exclude instances with invalid silver-label judgments.)

For the local models (Llama-3.1-8B, Mistral-7B, Prometheus-2-7B) in no-CoT prompting, we force a prefix of the assistant's response (e.g. "Rating A: ") and use a single output token as the judgment token position.

C.2 Pointwise Prompts

System prompt for pointwise scoring (CoT, $K = 9$)

Please act as an impartial judge and
↪ evaluate the quality of the response
↪ provided by an AI assistant to the user
↪ prompt displayed below. Your
↪ evaluation should consider factors
↪ such as the helpfulness, relevance,
↪ accuracy, depth, creativity, level of
↪ detail, and ethicality of the response.
↪ Begin your evaluation by providing a
↪ short explanation. Be as objective as
↪ possible. After providing your
↪ explanation, please rate the response
↪ with an integer score from 1 to 9,
↪ without further explanation.

System prompt for pointwise scoring (no-CoT, $K = 9$)

Please act as an impartial judge and
↪ evaluate the quality of the response
↪ provided by an AI assistant to the user
↪ prompt displayed below. Your
↪ evaluation should consider factors
↪ such as the helpfulness, relevance,
↪ accuracy, depth, creativity, level of
↪ detail, and ethicality of the response.
↪ Be as objective as possible. Please rate
↪ the response with an integer score
↪ from 1 to 9, without further
↪ explanation.

User prompt for pointwise judgment

[User Prompt]
`{User Prompt}`
[End User Prompt]

[Start of Assistant's Answer]
`{Assistant's Answer}`
[End of Assistant's Answer]

C.3 Pairwise Prompts

System prompt for pairwise scoring (CoT, $K = 9$)

```
Please act as an impartial judge and
↪ evaluate the quality of the responses
↪ provided by two AI assistants to the
↪ user prompt displayed below. Your
↪ evaluation should consider factors
↪ such as the helpfulness, relevance,
↪ accuracy, depth, creativity, level of
↪ detail, and ethicality of their responses.
↪ Begin your evaluation by comparing
↪ the two responses and provide a short
↪ explanation. Avoid any position biases
↪ and ensure that the order in which the
↪ responses were presented does not
↪ influence your decision. Do not allow
↪ the length of the responses to
↪ influence your evaluation. Do not
↪ favor certain names of the assistants.
↪ Be as objective as possible. After
↪ providing your explanation, output
↪ your final verdict by strictly following
↪ this format: "Rating A: {rating_a}.
↪ Rating B: {rating_b}.", where
↪ "{rating_a}" and "{rating_b}" are
↪ integer scores from 1 to 9.
```

For pairwise ranking with the local models, we use a different prompt from the one below. We found that they would often fail to include the braces specified in the judgment format, so we omit them when prompting these models.

System prompt for pairwise ranking (Likert-5, CoT)

```
Please act as an impartial judge and
↪ evaluate the quality of the responses
↪ provided by two AI assistants to the
↪ user prompt displayed below. You
↪ should choose the assistant that
↪ follows the user's instructions and
↪ answers the user's question better.
↪ Your evaluation should consider
↪ factors such as the helpfulness,
↪ relevance, accuracy, depth, creativity,
↪ level of detail, and ethicality of their
↪ responses. Begin your evaluation by
↪ comparing the two responses and
↪ provide a short explanation. Avoid any
↪ position biases and ensure that the
↪ order in which the responses were
↪ presented does not influence your
↪ decision. Do not allow the length of
↪ the responses to influence your
↪ evaluation. Do not favor certain names
↪ of the assistants. Be as objective as
↪ possible. After providing your
↪ explanation, output your final verdict
↪ by strictly following this format:
↪ "[[>>]]" if assistant A is significantly
↪ better, "[[>]]" if assistant A is slightly
↪ better, "[[=]]" for a tie, "[[<]]" if
↪ assistant B is slightly better, and
↪ "[[<<]]" if assistant B is significantly
↪ better.
```

User prompt for pairwise judgment

```
[User Prompt]
{User Prompt}
[End User Prompt]

[Start of Assistant A's Answer]
{Assistant A's Answer}
[End of Assistant A's Answer]

[Start of Assistant B's Answer]
{Assistant B's Answer}
[End of Assistant B's Answer]
```

C.4 Listwise Prompts

The listwise prompts are adapted from Nectar (Zhu et al., 2024).

System prompt for listwise judgment ($N = 7$), with intermediate pairwise preferences

We are interested in ranking different large language model chat completions to a conversation. Please act as an impartial judge and evaluate the quality of the completions provided by the 7 AI assistants. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, level of detail, and ethicality of their responses.

After the conversation and assistant responses, the section "PAIRWISE EVALUATION ORDER" will specify the order in which to perform pairwise comparisons. Output an array in which, for each pairwise comparison, you choose the letter of the better response, or '=' for a tie. The array should be comma-separated and enclosed in double square brackets.

Then, considering these pairwise rankings, please rank all 7 responses from best to worst (breaking ties randomly), strictly in the following format: [[_, _, _, _, _, _, _]] where '_' contains an assistant's letter name.

Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible.

System prompt for listwise judgment ($N = 7$), without intermediate pairwise preferences

We are interested in ranking different large language model chat completions to a conversation. Please act as an impartial judge and evaluate the quality of the completions provided by the 7 AI assistants. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, level of detail, and ethicality of their responses.

Please rank all 7 responses from best to worst (breaking ties randomly), strictly in the following format: [[_, _, _, _, _, _, _]] where '_' contains an assistant's letter name.

Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible.

User prompt for listwise judgment ($N = 7$). The presentation order is randomized. The pairwise evaluation order is randomized every instance for the prompt with intermediate pairwise preferences, and omitted for the prompt without intermediate pairwise preferences.

```
[CONVERSATION START]
{Conversation}
[CONVERSATION END]

[MODEL A RESPONSE START]
{Model A's response}
[MODEL A RESPONSE END]

[MODEL B RESPONSE START]
{Model B's response}
[MODEL B RESPONSE END]

[MODEL C RESPONSE START]
{Model C's response}
[MODEL C RESPONSE END]

[MODEL D RESPONSE START]
{Model D's response}
[MODEL D RESPONSE END]

[MODEL E RESPONSE START]
{Model E's response}
[MODEL E RESPONSE END]

[MODEL F RESPONSE START]
{Model F's response}
[MODEL F RESPONSE END]

[MODEL G RESPONSE START]
{Model G's response}
[MODEL G RESPONSE END]

PAIRWISE EVALUATION ORDER: [(G,
↪ C), (B, G), (C, D), (A, E), (G, A), (A,
↪ D), (B, A), (B, E), (B, F), (A, C), (E,
↪ C), (E, F), (B, D), (F, A), (G, E), (F,
↪ C), (F, D), (C, B), (F, G), (D, G), (E,
↪ D)]
```

D Datasets

RewardBench (Lambert et al., 2024) is a reward model benchmark spanning chat, reasoning, and safety. Each instance consists of a prompt, a chosen response, and a rejected response, all manually verified. The dataset categories are *Chat*, with 358 instances sourced from AlpacaEval (Li et al., 2023) and MT-Bench (Zheng et al., 2023b); *Chat Hard*, with 456 instances sourced from MT-Bench and LLMBar (Zeng et al., 2023); *Safety*, with 740 instances sourced from XSTest (Röttger et al., 2023), Do-Not-Answer (Wang et al., 2023), and original data; and *Reasoning*, with 1431 instances sourced from PRM800k (Lightman et al., 2023) and HumanEvalPack (Muennighoff et al., 2023). Except for excluding the prior sets category, we follow the original work and compute the final score as the average of the category scores.

MT-Bench (Zheng et al., 2023b) is a dataset of multi-turn questions spanning writing, roleplay, extraction, reasoning, math, coding, knowledge I (STEM), and knowledge II (humanities/social science). There are 3,355 (prompt, model pair, human judge, turn) tuples, 1,814 unique (prompt, model pair, turn) tuples, and 80 unique prompts each with two turns of interaction. To evaluate accuracy, we use the 1,132 instances with unanimous non-tie human judgments. To evaluate MSE, we use all 1,814 instances and set the label of an instance to the average of the human judgments, where a 0 or 1 represents the evaluated winner, and a 0.5 represents a tie.

Nectar (Zhu et al., 2024) is a dataset of 183k prompts each with 7 model responses. The prompts are sourced from Anthropic-HH (Bai et al., 2022), LMSYS-Chat-1M (Zheng et al., 2023a), UltraFeedback (Cui et al., 2023), and ShareGPT. We use a random subset of size 1,000.

RM-Bench (Liu et al., 2024c) is a reward model benchmark focusing on sensitivity to subtle content differences and resistance to style biases. There are 1,327 instances spanning chat, code, math, and safety. Similar to RewardBench, we follow the original work and average the 4 category scores. For each prompt, there are 3 pairs of (chosen, rejected) responses, where each pair is written with a particular style regarding concision and whether formatted as plain text or markdown.

The HelpSteer2 dataset (Wang et al., 2024b) contains multiple human ratings on a 0-4 scale for five attributes (helpfulness, correctness, coherence, complexity, verbosity) for each (prompt, response) instance. We use a random subset of size 1,000.

D.1 Listwise Evaluation

For the listwise setting, we use the same evaluation setup as with the pointwise and pairwise setting.[4] We concern ourselves with agreement at the pair level rather than the list level because pairwise preferences are sufficient to produce a total order, such as by choosing the maximum likelihood order (Liu et al., 2024e; Liusie et al., 2024) or with graph-theoretic methods (Tideman, 1987; Schulze, 2011; Li et al., 2024c). Thus, pairwise preferences are an adequate unit at which to measure agreement, and the aggregation into a total order may be modularized away for experimental simplicity.

To compute accuracy on Nectar with silver labels (Section 6.2), we take the sign of the silver label as the silver label for accuracy.

E Additional Results

Table 10 is an expanded version of Table 1, providing a subset breakdown of RewardBench. We observe particularly large gains for pointwise scoring on the Reasoning subset, e.g. absolute +7.7% and +17.1% for GPT-4o and Llama-3.1-8B.

Tables 11 ($K = 9$) and 14 ($K = 99$) show pointwise results over methods (expanded versions of Tables 4 and 6). Tables 12 and 13 show expanded tie analyses on RewardBench (simplified in Table 5) and MT-Bench.

Table 15 shows pairwise ranking results over methods, extending Table 7. Table 16 compares the Likert scales used for pairwise ranking, extending Table 8. In Table 16, the most calibrated setting on MT-Bench is (GPT-4o) Likert-5 no-CoT, achieving a 31% lower MSE than the most accurate setting, Likert-2 CoT, suggesting that a finer granularity has potential to improve calibration (Liu et al., 2024b). With GPT-4o in Tables 15 and 16, MSE is always minimized with no-CoT, highlighting the discord between CoT's sharpening effect and calibration. This result is in line with AlpacaEval (Dubois et al., 2024), which uses no-CoT and judgment probabilities, but deviating from WB-Reward and Arena-Hard-Auto (Lin et al., 2024; Li et al., 2024b), which use CoT and decoded judgments.

E.1 DeepSeek-V3 Results

We provide partial results for DeepSeek-V3 (DeepSeek-AI et al., 2025), a model of comparable size to GPT-4o. Tables 17 and 18 contain pointwise and listwise results, respectively. The trends for DeepSeek-V3 match those of GPT-4o.

[4]This means that our MT-Bench results are directly comparable across settings.

F Analysis

F.1 Heterogenous Preferences

We investigate whether LLM judges can represent pluralistically aligned preferences (i.e. reflect diverse human opinions) (Sorensen et al., 2024; Siththaranjan et al., 2023; Kumar et al., 2024) through their judgment distribution, without explicit training or prompting.

F.1.1 Multimodality

We begin by quantifying the degree of multimodality in the judgment distributions. An implicit assumption behind the conventional method of using the mode judgment is that the judgment distribution is unimodal and thus the mode is a representative judgment. However, in cases where humans disagree, we would like LLM judges to reflect the heterogeneity in the human population with a multimodal distribution.

We quantify multimodality as the minimum amount of probability mass that must be added to make an unnormalized unimodal distribution, divided by the total mass of the unnormalized unimodal distribution to obtain a value in $[0, 1)$, where a distribution is unimodal if the probability mass function is non-decreasing and then non-increasing. For example, if the judgment distribution is $[0.5, 0.2, 0.3]$, the minimum additional mass is 0.1 to obtain the unimodal distribution $[0.5, 0.3, 0.3]$ with total mass 1.1, so we compute the multimodality as $0.1/1.1 \approx 0.091$.

Table 19 presents the results. We find that more granularity leads to more multimodality (note that $K = 2$ always has multimodality 0), and no-CoT is more multimodal than CoT. The case of extreme multimodality for pointwise scoring $K = 99$ can be largely attributed to token bias (Lovering et al., 2024; Shaikh et al., 2024). For example, GPT-4o $K = 99$ CoT on MT-Bench assigns on average 0.036 probability to a single token that is a multiple of 5, but only 0.002 to a single token that differs by 1 from one of those multiples of 5.

F.1.2 Annotator Disagreement

We next examine whether human annotator disagreement is correlated with the uncertainty in the LLM's judgment distribution. On datasets with multiple human judgments per instance, we compute Spearman's ρ between the standard deviation

Model	Setting	Method	RewardBench					MT-Bench
			Chat	Chat Hard	Safety	Reasoning	Total	
GPT-4o	point score	mode	95.8, 89.7	76.0, 77.4	89.3, 88.5	79.5, 80.3	85.1, 84.0	81.9, 80.5
		mean	**97.1**, 94.3	75.2, **79.8**	**90.3**, 89.7	87.0, **88.0**	87.4, **88.0**	**83.6**, 83.2
	pair score	mode	97.3, **97.9**	69.0, 70.7	89.1, **89.5**	91.3, 91.3	86.7, 87.4	86.2, 86.5
		mean	97.2, 97.8	69.7, **70.8**	**89.5**, **89.5**	91.9, **92.4**	87.1, **87.6**	86.3, **86.8**
	pair rank	mode	96.9, 97.6	76.4, 79.1	89.0, **90.9**	91.4, 91.3	88.4, 89.7	86.3, 85.6
		mean	96.2, **98.3**	76.6, **79.4**	88.5, 90.8	93.0, **93.6**	88.6, **90.5**	**87.3**, 85.9
Llama-3.1-8B	point score	mode	83.8, 87.6	57.6, 58.0	76.2, 78.2	60.8, 64.8	69.6, 72.2	74.9, 71.9
		mean	89.0, **95.8**	58.6, **58.8**	73.0, **80.8**	70.2, **81.9**	72.7, **79.3**	78.7, **81.5**
	pair score	mode	92.0, 94.3	**45.4**, **45.4**	69.5, 78.8	79.9, 82.4	71.7, 75.2	**82.6**, 82.4
		mean	92.6, **95.8**	44.6, 45.0	69.3, **78.9**	81.7, **87.6**	72.1, **76.8**	82.3, 81.2
	pair rank	mode	76.7, 65.2	**52.3**, 48.1	71.0, 66.4	75.6, 55.8	68.9, 58.9	76.2, 63.0
		mean	90.5, **93.0**	50.0, 44.1	**78.1**, 72.7	**78.3**, 64.6	**74.2**, 68.6	**80.0**, 76.5
Mistral-7B	point score	mode	52.4, 66.2	51.5, 50.5	**79.9**, 75.7	57.8, 58.4	60.4, 62.7	59.5, 66.2
		mean	54.5, **82.1**	**53.5**, 49.1	**79.9**, 79.6	67.2, **77.5**	63.8, **72.1**	62.6, **74.0**
	pair score	mode	87.6, 89.9	40.2, 40.4	74.0, 73.0	67.4, 72.4	67.3, 68.9	79.3, 79.8
		mean	89.2, **91.1**	**41.2**, 39.3	**74.1**, 73.4	67.8, **80.2**	68.1, **71.0**	80.0, **80.4**
	pair rank	mode	51.0, 51.5	**51.0**, 46.2	62.2, 66.8	61.0, 50.8	56.3, 53.8	51.5, 51.5
		mean	79.5, **81.7**	39.3, 36.3	**73.1**, 67.7	**63.8**, 50.6	**63.9**, 59.1	**73.5**, 65.5
Prometheus-2-7B	point score	mode	81.3, 81.7	50.5, 50.8	65.9, 73.4	59.2, 58.2	64.3, 66.0	72.5, 73.5
		mean	82.4, **92.2**	48.9, **54.4**	65.7, **76.6**	61.3, **77.6**	64.6, **75.2**	72.1, **81.6**
	pair score	mode	91.2, 92.0	**44.1**, 43.6	**75.9**, 69.4	72.7, 69.6	**71.0**, 68.7	78.4, 80.8
		mean	91.3, **93.0**	42.7, 43.0	74.9, 72.0	73.0, **75.1**	70.5, 70.8	78.3, **80.9**
	pair rank	mode	55.6, 45.4	**51.0**, 50.0	66.6, 49.7	65.3, 47.8	59.6, 48.2	51.5, 43.0
		mean	**90.5**, 45.0	44.3, 50.7	**74.2**, 55.5	**69.8**, 44.1	**69.7**, 48.8	**75.4**, 33.4

Table 10: Mode vs. mean and CoT vs. no-CoT (comma-separated) accuracy results (%). Expanded version of Table 1.

of the human judgments and that of the LLM's judgment distribution.

For MT-Bench, we take the 961 instances with multiple human judgments. Table 20 reports weak correlation in all settings except no correlation in pairwise ranking with Llama-3.1-8B. Remarkably, *pointwise* score distributions encode sufficient information to predict if humans will disagree on a *pairwise* comparison of the texts.

The HelpSteer2 dataset (Wang et al., 2024b) contains multiple human ratings on a 0-4 scale for five attributes for each (prompt, response) instance. We use a random subset of size 1,000. We prompt with the provided annotation guidelines and have the model rate all attributes in a single run. Table 21 reports weak correlation on helpfulness, correctness, and coherence but no correlation on complexity and verbosity. We suspected this to be due to that conditioning on the earlier attributes' scores may reduce uncertainty for the later attributes (Stureborg et al., 2024; Hashemi et al., 2024), but we found that the average standard deviation is similar across attributes for both LLM and human judgments.

F.1.3 Pluralistic Alignment

We finally evaluate the alignment between predicted judgment distributions and human judgment distributions. We quantify the distance between two distributions μ and ν with the Wasserstein p-distance for $p \in \{1, 2\}$:

$$W_p(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \left(\mathop{\mathbb{E}}_{(x,y)\sim\gamma} |x - y|^p \right)^{\frac{1}{p}}, \quad (1)$$

where $\Gamma(\mu, \nu)$ is the set of couplings of μ and ν. A higher p more heavily punishes large point distances $|x - y|$. We scale the judgment spaces to $[0, 1]$ so that $W_p(\mu, \nu) \in [0, 1]$.

As baselines, we consider deterministic distributions that place probability 1 on a measure of central tendency.

Table 22 shows that using a distributional prediction has little success in improving alignment with the MT-Bench human pairwise preferences, but Table 23 shows success for HelpSteer2 human pointwise scores.

We also experimented with the HelpSteer2-Preference dataset, prompting with the provided annotation guidelines (Wang et al., 2024a). However, we found severe position bias in our experiments with GPT-4o and Llama-3.1-8B (no-CoT). The analysis showed no correlation between predicted distribution variance and annotator disagreement, and poor pluralistic alignment compared to the deterministic baselines.

Model	Method	RewardBench		MT-Bench	
		Acc ↑	MSE ↓	Acc ↑	MSE ↓
GPT-4o	MODE	85.1, 84.0	.116, .118	81.9, 80.5	.152, .145
	MEAN	87.4, 88.0	.099, .102	83.6, 83.2	.115, .097
	[MEAN]	85.1, 85.2	.116, .109	82.0, 80.2	.150, .146
	MEDI	85.0, 84.6	.116, .112	82.0, 80.2	.150, .142
	1P	84.8, 84.3	.120, .116	82.6, 81.0	.141, .138
	RAM	87.4, **88.4**	.099, .100	**83.9**, 83.4	.115, **.096**
	QT	87.4, 87.9	.107, **.096**	83.5, 83.2	.139, .118
	PS	87.4, 87.8	.106, **.096**	83.5, 83.3	.136, .103
Llama -3.1-8B	MODE	69.6, 72.2	.237, .192	74.9, 71.9	.177, .142
	MEAN	72.7, 79.3	.198, .155	78.7, **81.5**	.129, .104
	[MEAN]	70.1, 75.0	.238, .186	75.7, 75.0	.172, .145
	MEDI	69.8, 73.6	.238, .191	75.2, 73.9	.176, .142
	1P	70.2, 76.0	.238, .183	76.8, 79.2	.172, .147
	RAM	72.7, **79.9**	.200, **.152**	78.8, 81.4	.130, **.102**
	QT	72.8, 79.0	.220, .164	78.7, 81.1	.154, .116
	PS	72.8, 78.9	.216, .161	78.6, 81.4	.149, .110

Table 11: Pointwise results over methods. Comma-separated values are with and without CoT (expanded version of Table 4). Text styling follows Table 1.

Model	Method	Tie rate		MEAN's accuracy		Non-tie accuracy Δ ↑	
		$K = 9$	$K = 99$	$K = 9$	$K = 99$	$K = 9$	$K = 99$
GPT-4o	MODE	.13, .17	.09, .20	64, 72	61, 73	+0.0, –0.1	+0.0, –0.3
	[MEAN]	.13, .16	.02, .03	65, 67	61, 53	+0.0, +0.0	+0.0, –0.0
	MEDI	.13, .17	.06, .09	65, 70	58, 62	–0.0, +0.0	–0.0, –0.1
	1P	.13, .16	.05, .08	66, 66	58, 60	+0.1, +0.1	+0.0, +0.5
Llama -3.1-8B	MODE	.27, .35	.18, .24	60, 69	63, 70	+0.2, –0.2	+0.2, –2.0
	[MEAN]	.25, .26	.07, .07	58, 64	56, 61	+0.0, +0.0	+0.0, +0.0
	MEDI	.26, .29	.11, .11	60, 67	59, 67	+0.1, –0.5	–0.2, –0.6
	1P	.24, .23	.08, .08	61, 65	55, 57	+0.3, +0.6	+0.8, +0.1

Table 12: Tie analysis for discrete pointwise methods on RewardBench (expanded version of Table 5). Tie rate is the proportion of instances where the method predicts a tie, over which we report MEAN's accuracy (%); excess of 50% or 75% indicates room for improving accuracy or MSE, respectively. Non-tie accuracy Δ (%) is the method's accuracy minus MEAN's accuracy over the non-tie instances. Comma-separated values are with and without CoT. We find that the mode has the most ties, the highest MEAN accuracy, and the lowest non-tie accuracy delta (i.e. poor recall without better precision), especially for no-CoT $K = 99$.

F.2 Sensitivity to Score Granularity

Adopting the view that LLMs latently encode a continuous distribution but output a discretization of it (Gillman et al., 2024), we analyze how faithfully functions of the (latent) continuous distribution can be approximated by those functions computed on the (observed) discretization. For practical interest, this manifests as robustness to the choice of K, with convergence in distribution to the continuous distribution as $K \to \infty$. Thus, independently of the "principledness" of certain functions of a ground-truth continuous distribution, it is appropriate to examine the effect of discretization on our ability to approximate them to begin with. Our theoretical result is stated in Proposition 1 (see Appendix G.1 for full statement, proof, and discussion).

Proposition 1. *Among the discrete methods in Table 3,* MODE *computed on continuous distributions may fail to be approximated by the same function computed on their discretizations, even under regularity conditions. Meanwhile,* [MEAN], MEDI, *and* 1P *admit an approximation error bound.*

We empirically assess the robustness to K of the score distributions produced by the LLM judge as well as the functions computed on them. The former is not addressed by Proposition 1, which assumes the score distributions to be errorless discretizations and thus consistent across granularities.

F.2.1 Sensitivity of Score Distributions

For an evaluated text, let μ^K denote the score distribution with granularity K, with the score space scaled to $[0, 1]$. We coarsify μ^{99} into $\hat{\mu}^{99}$

Model	Method	Tie rate		MEAN's accuracy		Non-tie accuracy Δ ↑	
		$K = 9$	$K = 99$	$K = 9$	$K = 99$	$K = 9$	$K = 99$
GPT-4o	MODE	.13, .21	.08, .22	62, 62	64, 67	+0.0, –0.2	+0.1, –0.7
	[MEAN]	.13, .21	.02, .04	61, 64	42, 48	+0.0, +0.0	+0.0, +0.0
	MEDI	.13, .22	.05, .11	61, 64	64, 55	+0.0, +0.1	+0.0, –0.6
	1P	.14, .19	.06, .09	56, 62	67, 56	–0.1, +0.1	+0.3, +0.7
Llama -3.1-8B	MODE	.25, .45	.14, .26	65, 71	61, 65	–0.1, –1.0	–0.4, –3.0
	[MEAN]	.24, .36	.06, .09	63, 68	49, 55	+0.0, +0.0	+0.0, –0.1
	MEDI	.25, .40	.10, .18	65, 69	54, 58	+0.0, –0.3	–0.4, +0.5
	1P	.20, .23	.07, .07	60, 59	53, 50	+0.1, –0.3	+1.8, +0.3

Table 13: Tie analysis for discrete pointwise methods on MT-Bench, mirroring Table 12.

Model	Method	RewardBench		MT-Bench	
		Acc ↑	MSE ↓	Acc ↑	MSE ↓
GPT-4o	MODE	$86.1_{+1.0}$, $81.7_{-2.3}$	$.118_{+.002}$, $.134_{+.016}$	$83.8_{+1.9}$, $78.4_{-2.1}$	$.152_{+.000}$, $.158_{+.013}$
	MEAN	$\mathbf{87.4}_{+0.0}$, $\underline{86.7}_{-1.3}$	$\mathbf{.097}_{-.002}$, $.108_{+.006}$	$\underline{84.8}_{+1.2}$, $82.9_{-0.3}$	$.105_{-.010}$, $.099_{+.002}$
	[MEAN]	$87.0_{+1.9}$, $\underline{86.5}_{+1.3}$	$.124_{+.008}$, $.127_{+.018}$	$\mathbf{85.1}_{+3.1}$, $82.7_{+2.5}$	$.167_{+.017}$, $.182_{+.036}$
	MEDI	$86.7_{+1.7}$, $85.2_{+0.6}$	$.119_{+.003}$, $.126_{+.014}$	$84.1_{+2.1}$, $81.5_{+1.3}$	$.160_{+.010}$, $.170_{+.028}$
	1P	$86.6_{+1.8}$, $86.4_{+2.1}$	$.121_{+.001}$, $.116_{+.000}$	$84.2_{+1.6}$, $82.7_{+1.7}$	$.159_{+.018}$, $.165_{+.027}$
	RAM	$87.1_{-0.3}$, $\underline{86.7}_{-1.7}$	$.098_{-.001}$, $.104_{+.004}$	$\mathbf{85.1}_{+1.2}$, $83.0_{-0.4}$	$.106_{-.009}$, $\mathbf{.098}_{+.002}$
	QT	$\underline{87.3}_{-0.1}$, $\underline{86.6}_{-1.3}$	$.112_{+.005}$, $.114_{+.018}$	$\underline{84.8}_{+1.3}$, $82.7_{-0.5}$	$.149_{+.010}$, $.147_{+.029}$
	PS	$\underline{87.3}_{-0.1}$, $\underline{86.6}_{-1.2}$	$.105_{-.001}$, $.105_{+.009}$	$\underline{84.8}_{+1.3}$, $82.4_{-0.9}$	$.130_{-.006}$, $.107_{+.004}$
Llama -3.1-8B	MODE	$73.4_{+3.8}$, $72.0_{-0.2}$	$.222_{-.015}$, $.221_{+.029}$	$77.3_{+2.4}$, $75.1_{+3.2}$	$.191_{+.014}$, $.169_{+.027}$
	MEAN	$75.9_{+3.2}$, $\underline{79.3}_{+0.0}$	$.183_{-.015}$, $.156_{+.001}$	$79.3_{+0.6}$, $\underline{81.3}_{-0.2}$	$.125_{-.004}$, $\underline{.103}_{-.001}$
	[MEAN]	$75.3_{+5.2}$, $78.5_{+3.5}$	$.229_{-.009}$, $.198_{+.012}$	$79.3_{+3.6}$, $80.7_{+5.7}$	$.201_{+.029}$, $.180_{+.035}$
	MEDI	$74.4_{+4.6}$, $76.5_{+2.9}$	$.228_{-.010}$, $.207_{+.016}$	$78.4_{+3.2}$, $80.1_{+6.2}$	$.198_{+.022}$, $.161_{+.019}$
	1P	$76.2_{+6.0}$, $78.5_{+2.5}$	$.218_{-.020}$, $.195_{+.012}$	$\underline{80.6}_{+3.8}$, $\underline{81.5}_{+2.3}$	$.187_{+.015}$, $.177_{+.030}$
	RAM	$76.1_{+3.4}$, $\mathbf{79.7}_{-0.2}$	$.179_{-.021}$, $\mathbf{.152}_{+.000}$	$79.7_{+0.9}$, $\underline{81.1}_{-0.3}$	$.123_{-.007}$, $\mathbf{.102}_{+.000}$
	QT	$75.7_{+2.9}$, $78.7_{-0.3}$	$.214_{-.006}$, $.177_{+.013}$	$78.8_{+0.1}$, $81.3_{+0.2}$	$.179_{+.025}$, $.143_{+.027}$
	PS	$75.7_{+2.9}$, $78.6_{-0.3}$	$.203_{-.013}$, $.163_{+.002}$	$78.6_{+0.0}$, $\mathbf{81.8}_{+0.4}$	$.151_{+.002}$, $.111_{+.001}$

Table 14: Pointwise results over methods ($K = 99$). Comma-separated values are with and without CoT (expanded version of Table 6). Subscripts denote change from $K = 9$ (Table 11). Text styling follows Table 1.

by binning into 9 blocks of 11 scores. We then quantify sensitivity as the Wasserstein 1-distance $W_1(\mu^9, \hat{\mu}^{99}) \in [0, 1]$ (Eq. 1) averaged over the pointwise instances in the dataset.

F.2.2 Sensitivity of Pointwise Methods

For a dataset $\mathcal{D}$ of paired responses, we denote $\mathbf{a}^K$ as the $|\mathcal{D}|$-length vector containing the value of a method computed on each pair using granularity K. We then quantify sensitivity as the normalized flip rate

$$\mathrm{FR} := \frac{\|\mathrm{sgn}(\mathbf{a}^9) - \mathrm{sgn}(\mathbf{a}^{99})\|_1}{\|\mathrm{sgn}(\mathbf{a}^9)\|_1 + \|\mathrm{sgn}(\mathbf{a}^{99})\|_1} \in [0, 1]. \quad (2)$$

F.2.3 Results

Table 24 presents the results on sensitivity to granularity. The discrete metrics are more sensitive than the continuous metrics. Furthermore, consistent with Proposition 1, we find that the mode is the most sensitive among the discrete methods, particularly with no-CoT.

The effect of CoT differs between the models: GPT-4o is less sensitive with CoT, and Llama-3.1-8B is less sensitive with no-CoT. Similar to Lee et al. (2024b), it would appear that although GPT-4o is a more capable judge than Llama-3.1-8B, it is not as robust to granularity (in each model's CoT/no-CoT of choice). However, this is partially because a limitation with setting K as large as 99 for GPT-4o is that no-CoT distributions tend to have high spread (Table 2), resulting in nontrivial probability mass falling outside of the top 20 tokens provided by the OpenAI API. Concretely, the average total mass on the top score tokens is 0.88/0.90 on RewardBench/MT-Bench for no-CoT, but over 0.99 for CoT.

F.3 Position Bias

We compare the degree of position bias (i.e. the LLM judge's sensitivity to the order in which the evaluated texts are presented (Zheng et al., 2023b)) between various settings.

Model	Center	Agg. Time	RewardBench		MT-Bench	
			Acc ↑	MSE ↓	Acc ↑	MSE ↓
GPT-4o	mode	post	88.1, 89.3	.099, **.090**	86.1, 84.9	**.139**, .142
		pre	88.4, **90.3**	.112, .094	**86.5**, 85.2	.154, .154
	median	post	88.1, 89.3	.099, **.091**	86.1, 84.9	**.138**, .142
		pre	88.4, **90.0**	.111, .094	**86.6**, 85.4	.153, .146
	mean	post	88.9, **90.4**	.098, **.077**	**86.5**, 85.4	.132, .100
		pre	88.9, **90.4**	.098, .078	**86.6**, 85.4	.132, **.097**
Llama -3.1-8B	mode	post	56.7, 52.4	**.240**, .279	57.5, 53.4	.192, **.176**
		pre	**73.1**, 66.1	.265, .337	**78.1**, 70.9	.222, .268
	median	post	56.8, 52.5	**.240**, .279	57.5, 53.5	.192, **.176**
		pre	**72.9**, 65.3	.261, .319	**78.0**, 69.1	.218, .238
	mean	post	73.2, 65.6	**.207**, .229	**78.2**, 70.5	**.144**, .146
		pre	**73.2**, 66.3	.222, .240	78.1, 70.8	.155, .155

Table 15: Pairwise ranking results over methods, using Likert-3 (expanded version of Table 7). Comma-separated values are with and without CoT. Text styling follows Table 1.

Model	K	RewardBench		MT-Bench	
		Acc ↑	MSE ↓	Acc ↑	MSE ↓
GPT-4o	2	88.6, **90.5**	.094, **.077**	**87.3**, 85.9	.136, .101
	3	88.9, 90.4	.098, **.078**	86.6, 85.4	.132, .097
	5	88.8, 89.5	.099, .106	84.7, 85.8	.129, **.087**
Llama -3.1-8B	2	**74.2**, 68.6	**.187**, .214	**80.0**, 76.5	**.126**, .135
	3	73.2, 66.3	.222, .240	78.1, 70.8	.155, .155
	5	70.0, 58.5	.215, .234	77.1, 64.8	.142, .153

Table 16: Pairwise ranking results over Likert-K scales, using pre-aggregation mean (expanded version of Table 8). Comma-separated values are with and without CoT. Text styling follows Table 1.

Method	Acc ↑	MSE ↓
MODE	84.7, 82.5	0.123, 0.128
MEAN	84.8, 84.2	0.119, 0.120
[MEAN]	84.7, 82.7	0.123, 0.127
MEDI	84.7, 82.6	0.123, 0.128
1P	84.5, 82.9	0.124, 0.126
RAM	**84.9**, 84.1	0.120, **0.118**
QT	**85.0**, 83.9	0.122, 0.125
PS	**85.0**, 83.9	0.122, 0.125

Table 17: Pointwise results with DeepSeek-V3 on RewardBench. $K = 9$. Comma-separated values are with and without CoT. Text styling follows Table 1.

Evaluation Metrics For the pairwise setting (scoring or ranking), we measure mean absolute error (MAE) and mean squared error (MSE) between the two judgments from the two orders, using pre-aggregation mean. Compared to MAE, MSE punishes a few large errors more than many small errors.

For the listwise setting, we measure Spearman's ρ between the difference in the presented positions of two responses and the judgment.

[5]In every judgment space, GPT-4o tends to favor responses that are presented earlier.

Space	Method	Nectar		RM-Bench	
		Acc	MSE	Acc	MSE
direct list	mode	83.6	0.149	**67.8**	0.322
direct list	mean	**84.0**	**0.129**	67.6	**0.307**

Table 18: Listwise results with DeepSeek-V3. Text styling follows Table 1.

Model	Setting	K	RewardBench	MT-Bench
GPT-4o	point score	9	.000, .008	.000, .012
	point score	99	.362, .409	.357, .440
	pair rank	3	.000, .018	.000, .019
	pair rank	5	.014, .049	.021, .041
Llama -3.1-8B	point score	9	.009, .040	.013, .025
	point score	99	.356, .379	.382, .365
	pair rank	3	.044, .091	.051, .081
	pair rank	5	.107, .194	.107, .245

Table 19: A study on multimodality (see Appendix F.1.1). Comma-separated values are with and without CoT.

Results Tables 25 and 26 report position bias in the pairwise settings. We find that no-CoT always improves MSE, even when it hurts MAE, showing that no-CoT reduces cases of extreme position bias.

Table 27 reports listwise position bias. We find that DIRECT LIST exhibits the most position bias, consistent with Zhu et al. (2024), despite achieving the highest accuracy (Table 9). On the other hand, INTERM has the least position bias. As the intermediate pairwise preferences can be likened to CoT, this suggests that intermediate reasoning can mitigate bias in challenging judgment settings. However, since an ideal judge should be able to simultaneously maximize accuracy and minimize bias, we believe current methods have ample room

Model	Setting	MT-Bench
GPT-4o	point score	**+0.21**, **+0.24**
	pair score	**+0.19**, **+0.27**
	pair rank	**+0.19**, **+0.27**
Llama -3.1-8B	point score	**+0.21**, **+0.14**
	pair score	**+0.20**, **+0.24**
	pair rank	+0.02, –0.04

Table 20: Spearman's ρ between standard deviation of human judgments and that of LLM's judgment distribution. Comma-separated values are with and without CoT. Bold denotes significant correlation ($\alpha = 0.01$). Ranking uses Likert-3; scoring uses $K = 9$ converted to a Likert-3 distribution $[P(X_1 > X_2), P(X_1 = X_2), P(X_1 < X_2)]$.

for improvement.

F.4 Transitivity

We say a comparison method $a(\cdot, \cdot) \in [-1, 1]$ is transitive if $a(A_1, A_2) > 0$ and $a(A_2, A_3) \geq 0$ imply $a(A_1, A_3) > 0$ for all triplets of texts (A_1, A_2, A_3). For example, a score distribution comparison function that reduces to the comparison of two real numbers derived from the two score distributions independently (e.g. mode or mean) is transitive. On the other hand, QT, PS, and the pairwise ranking methods are intransitive.

Human preferences have been shown to exhibit intransitivity (Klimenko, 2015), motivating the question of whether LLM judges do so too and how this depends on the method used. Several prior works have proposed methods incorporating awareness of the intransitivity in LLM or human preferences (Liu et al., 2024e; Ethayarajh et al., 2024; Zhang et al., 2024d; Ye et al., 2024a; Hu et al., 2024; Zhang et al., 2024c; Liu et al., 2024d). We adopt the view in Liu et al. (2024e) that transitivity is generally desirable and indicative of a more capable judge, especially in the absence of a curated dataset of intransitive human preferences. Nevertheless, we remark that the ability to model intransitivity is essential to preference modeling in its full generality (Ethayarajh et al., 2024; Zhang et al., 2024d; Ye et al., 2024a), which, among pointwise methods, is achieved by QT and PS but not by mode and mean used in prior work.

Table 28 presents the intransitivity rates of different methods. Despite the capacity of QT and PS to model intransitive preferences (Savage, 1994; Finkelstein and Thorp, 2006; Conrey et al., 2013), we find that they exhibit negligible intransitivity compared to the pairwise ranking methods. Similar to Liu et al. (2024e), we observe that a stronger judge (GPT-4o) exhibits less intransitivity than a weaker judge (Llama-3.1-8B). Pre-aggregation mean exhibits less intransitivity than post-aggregation mode. Notably, for pairwise ranking, we observe more intransitivity with CoT than without CoT, even though CoT achieves higher accuracy (Table 1).

G Derivations

G.1 Approximability of Discrete Pointwise Functions Under Discretization

Proposition 1. *We analyze the discrete methods in Table 3. Specifically, we examine the score function r rather than $\operatorname{sgn}(r_1 - r_2)$.*

Let X be a random variable with support $S \subset [\frac{1}{2}, K + \frac{1}{2})$ for an integer K. Define its discretization $\hat{X}$ by $P(\hat{X} = \hat{x}) \coloneqq P([X] = \hat{x})$ for $\hat{x} \in \hat{S} \coloneqq \{1, \ldots, K\}$, where $[\cdot]$ denotes rounding to the nearest integer.

1. MODE *may fail to be approximated: Suppose X has a density f_X that is L-Lipschitz with $L \leq 1$ and achieves its supremum at $x^* \in \arg\max_{x \in S} f_X(x)$. Let $\hat{x}^* \in \arg\max_{\hat{x} \in \hat{S}} P(\hat{X} = \hat{x})$. Suppose some $\hat{x} \in \hat{S}$, with arbitrarily large $|\hat{x} - \hat{x}^*| > 1$, satisfies $P(\hat{X} = \hat{x}^*) \geq P(\hat{X} = \hat{x}) + \frac{L}{4}$. The above is consistent with $[x^*] = \hat{x}$.*

2. [MEAN] *can be approximated: $|[\mathbb{E}X] - [\mathbb{E}\hat{X}]| \leq 1$.*

3. MEDI *and* 1P *can be approximated: For $p \in (0, 1)$, $|Q_X(p) - Q_{\hat{X}}(p)| \leq \frac{1}{2}$.*

Proof.

1. We present a construction.

   If $L = 0$, the claim is immediate; assume not. Define $d \coloneqq \frac{L}{4}(\sqrt{1 + 8/L} - 2) \geq \frac{L}{4}$. Let $f_X(x) = (d - \frac{L}{4}) + L(x - \hat{x} + \frac{1}{2})$ for $x \in [\hat{x} - \frac{1}{2}, \hat{x})$, and $f_X(x) = (d - \frac{L}{4}) + L(\hat{x} - x + \frac{1}{2})$ for $x \in [\hat{x}, \hat{x} + \frac{1}{2})$, and $f_X(x) = d + \frac{L}{4}$ for $[x] = \hat{x}^*$.

   Around the regions $[\hat{x} - \frac{1}{2}, \hat{x} + \frac{1}{2})$, $[\hat{x}^* - \frac{1}{2}, \hat{x}^* + \frac{1}{2})$, we let f_X decrease to 0 with slope $\pm L$, or until reaching the domain boundary or each other. Continuity is maintained at the junction because, supposing $\hat{x} < \hat{x}^*$ without loss of generality, the nearest endpoints $\hat{x} + \frac{1}{2}, \hat{x}^* - \frac{1}{2}$

Model	Helpfulness	Correctness	Coherence	Complexity	Verbosity
GPT-4o	**+0.24**	**+0.36**	**+0.32**	+0.02	–0.01
Llama-3.1-8B	**+0.14**	**+0.22**	**+0.22**	–0.00	+0.01

Table 21: Spearman's ρ between standard deviation of human judgments and that of LLM's judgment distribution. HelpSteer2, no-CoT. Bold denotes significant correlation ($\alpha = 0.01$).

Model	Setting	Method	W_1	W_2
GPT-4o	point score	mode	.229, .246	.406, .419
		mean	.229, .247	.388, **.349**
		distr	**.219**, .222	.395, .386
	pair score	mode	.229, .230	.419, .419
		mean	.218, **.215**	.399, **.387**
		distr	.220, **.215**	.408, .401
	pair rank	mode	.228, .226	.420, .412
		mean	.221, .212	.396, **.362**
		distr	.215, **.203**	.405, .385
Llama -3.1-8B	point score	mode	.274, .267	.438, .405
		mean	.277, .267	.412, **.359**
		distr	.261, **.246**	.425, .391
	pair score	mode	.268, .276	.460, .470
		mean	.241, .244	.404, **.400**
		distr	**.239**, .243	.426, .433
	pair rank	mode	**.296**, .336	.490, .531
		mean	.356, .370	.423, **.420**
		distr	.347, .356	.540, .548

Table 22: Pluralistic alignment error (↓, Eq. 1) from MT-Bench human pairwise preferences. Comma-separated values are with and without CoT. Text styling follows Table 1. The method 'distr' uses the predicted distribution, while the other methods place probability 1 on a measure of central tendency.

satisfy $|(\hat{x}+\frac{1}{2})-(\hat{x}^*-\frac{1}{2})| \geq 1$ and $|f_X(\hat{x}+\frac{1}{2})-f_X(\hat{x}^*-\frac{1}{2})| = \frac{L}{2}$.

We verify that $P(\hat{X}=\hat{x}^*) = d+\frac{L}{4} = P(\hat{X}=\hat{x})+\frac{L}{4}$ and $\hat{x} \in \{\hat{x}\} \cup [\hat{x}^*-\frac{1}{2}, \hat{x}^*+\frac{1}{2}) = \arg\max_{x\in S} f_X(x)$.

It remains to check that we have a valid distribution. The total $\int f_X$ is bounded by the case if f_X is allowed to reach 0 everywhere possible above:

$$\begin{aligned}\int f_X &\leq P(\hat{X}=\hat{x}) + P(\hat{X}=\hat{x}^*) \\ &\quad + \frac{1}{L}\left(d-\frac{L}{4}\right)^2 + \frac{1}{L}\left(d+\frac{L}{4}\right)^2 \\ &= 1-\frac{L}{4} < 1,\end{aligned}$$

so f_X can be made a valid density by adding an appropriately scaled uniform density, not affecting the desired properties.

2. Denote the measures of $X, \hat{X}$ as $\mu_X, \mu_{\hat{X}}$. The definition of X and $\hat{X}$ is equivalent to the existence of a coupling $\gamma \in \Gamma(\mu_X, \mu_{\hat{X}})$ with samples defined by $(x, \hat{x}) \sim \gamma$ for $x \sim \mu_X$ and $\hat{x} = [x]$.

$$\begin{aligned}|\mathbb{E}X - \mathbb{E}\hat{X}| &= \left|\int (x-\hat{x})\,\mathrm{d}\gamma(x,\hat{x})\right| \\ &\leq \int |x-\hat{x}|\,\mathrm{d}\gamma(x,\hat{x}) \leq \int \frac{1}{2}\,\mathrm{d}\gamma(x,\hat{x}) = \frac{1}{2}\end{aligned}$$

Thus, $|[\mathbb{E}X] - [\mathbb{E}\hat{X}]| \leq 1$.

3. Let $q \coloneqq Q_X(p)$.

$$\begin{aligned}&P(\hat{X} < [q]-\frac{1}{2}) = P(X < [q]-\frac{1}{2}) < p \\ &\leq P(X < [q]+\frac{1}{2}) = P(\hat{X} < [q]+\frac{1}{2}),\end{aligned}$$

implying $Q_{\hat{X}}(p) = [q]$ where $|q-[q]| \leq \frac{1}{2}$.

□

Remark. The suppositions in (1) are to impose regularity and show even then approximation may not hold. For an example of their omission, without requiring absolutely continuous X, it could place atoms at arbitrary x, preventing any margin $P(\hat{X}=\hat{x}^*) - P(\hat{X}=\hat{x})$ less than 1 from producing an error bound. The crucial case that causes the mode to be unstable to approximate is the case of multimodality.

In (3), it is crucial that we assumed no discretization error, i.e. $|P(\hat{X}=\hat{x}) - P([X]=\hat{x})| = 0$. With any discretization error, we would have no bound on approximation error.

H Licensing

Our usage of the artifacts below complies with their licenses.

Model Licensing GPT-4o[6] has a proprietary license. Llama-3.1-8B[7] is licensed under the Llama

[6] https://platform.openai.com/docs/models#gpt-4o

[7] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

Model	Method	Helpfulness		Correctness		Coherence		Complexity		Verbosity	
		W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2
GPT-4o	mode	.218	.311	.219	.332	.149	.252	.211	.273	.186	.257
	mean	.221	.297	.217	.318	.151	.240	.213	.262	.197	**.244**
	distr	**.188**	**.279**	**.194**	**.301**	**.134**	**.233**	**.199**	**.255**	**.179**	.249
Llama -3.1-8B	mode	.259	.369	.250	.377	.154	.280	.227	.290	.182	.255
	mean	.255	.339	.249	.347	.158	.253	.224	.274	.174	**.223**
	distr	**.219**	**.328**	**.215**	**.334**	**.134**	**.250**	**.209**	**.270**	**.164**	.234

Table 23: Pluralistic alignment error (↓, Eq. 1) from HelpSteer2 human pointwise scores. No-CoT. Text styling follows Table 1. The method 'distr' uses the predicted distribution, while the other methods place probability 1 on a measure of central tendency.

Model	Method	Reward-Bench	MT-Bench
GPT-4o	–	.091, .105	.093, .111
	MODE	.103, .150	.128, .214
	MEAN	.066, .080	.105, .136
	[MEAN]	.104, .115	.144, .199
	MEDI	.101, .113	.137, .185
	1P	.096, .117	.137, .196
	RAM	.074, .084	.111, .138
	QT	**.064**, .078	**.104**, .133
	PS	**.064**, .078	**.104**, .137
Llama -3.1-8B	–	.136, .063	.117, .076
	MODE	.213, .201	.223, .247
	MEAN	.149, .042	.131, .048
	[MEAN]	.213, .139	.219, .218
	MEDI	.219, .160	.224, .218
	1P	.223, .105	.183, .133
	RAM	.168, .037	.156, .068
	QT	.151, **.034**	.130, .048
	PS	.151, .037	.129, **.046**

Table 24: Sensitivity to granularity (↓) of the score distributions (Eq. 1) and of the pointwise methods computed on them (Eq. 2). Comma-separated values are with and without CoT. Text styling follows Table 1.

Model	Setting	K	MAE	MSE
GPT-4o	score	9	.090, **.076**	.057, **.031**
	score	99	.094, .095	.049, .032
	rank	2	.086, .087	.083, .037
	rank	3	.085, .089	.078, .035
	rank	5	.141, .182	.079, .053
Llama -3.1-8B	score	9	.199, .163	.125, .066
	score	99	.188, **.160**	.114, **.060**
	rank	2	.357, .329	.193, .154
	rank	3	.683, .518	.547, .340
	rank	5	.506, .342	.334, .164

Table 25: Pairwise position bias (↓, see Appendix F.3) on RewardBench (see Table 26 for MT-Bench). Comma-separated values are with and without CoT. Text styling follows Table 1. We find that no-CoT always maintains or improves MSE, even when it hurts MAE.

Model	Setting	K	MAE	MSE
GPT-4o	score	9	.108, **.091**	.075, **.038**
	score	99	.111, .108	.066, **.039**
	rank	2	.108, .132	.100, .056
	rank	3	.108, .134	.093, .051
	rank	5	.187, .172	.120, .047
Llama -3.1-8B	score	9	.211, .148	.145, .056
	score	99	.193, **.141**	.129, **.049**
	rank	2	.312, .355	.174, .172
	rank	3	.618, .532	.466, .337
	rank	5	.458, .298	.293, .129

Table 26: Pairwise position bias (↓) on MT-Bench, mirroring Table 25.

3.1 Community License Agreement. Mistral-7B[8] and Prometheus-2-7B[9] are licensed under the Apache License 2.0.

Dataset Licensing The datasets contain English language data. RewardBench[10] and RM-Bench[11] are licensed under the ODC-By license. MT-Bench[12] and HelpSteer2[13] are licensed under the CC BY 4.0 license. Nectar[14] is licensed under the Apache License 2.0.

I Ethical Considerations

LLMs can exhibit unwanted biases. Relying on their judgments for downstream applications can propagate these biases. Nevertheless, our findings in this paper promote practices for improving alignment with human preferences.

[8] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

[9] https://huggingface.co/prometheus-eval/prometheus-7b-v2.0

[10] https://huggingface.co/datasets/allenai/reward-bench

[11] https://huggingface.co/datasets/THU-KEG/RM-Bench

[12] https://huggingface.co/datasets/lmsys/mt_bench_human_judgments

[13] https://huggingface.co/datasets/nvidia/HelpSteer2/tree/main/disagreements

[14] https://huggingface.co/datasets/berkeley-nest/Nectar

Space	Nectar	RM-Bench	MT-Bench
interm	**.086**	**.079**	**.033**
list	.092	.100	.041
direct list	.118	.105	.056

Table 27: Listwise position bias (↓) with GPT-4o. We report the absolute value[5] of Spearman's ρ between the difference in the presented positions of two responses and the judgment. Text styling follows Table 1.

Model	Setting	Method	MT-Bench
GPT-4o	point score	QT	.000, .000
	point score	PS	.006, .002
	pair rank	MODE-AGG	.026, .022
	pair rank	AGG-MEAN	.007, .003
Llama -3.1-8B	point score	QT	.000, .000
	point score	PS	.001, .000
	pair rank	MODE-AGG	.234, .218
	pair rank	AGG-MEAN	.040, .023

Table 28: A study on transitivity. In each cell, we report the proportion of triplets that exhibit intransitivity, with and without CoT. (Pointwise scoring uses $K = 9$; pairwise ranking uses Likert-2.) In addition, our Nectar silver labels (GPT-4o, Likert-5, no-CoT, mean) have an intransitivity rate of 0.020.