Analyzing Uncertainty of LLM-as-a-Judge: Interval Evaluations with Conformal Prediction

Anonymous Author(s)
Affiliation
Address
email

Abstract

LLM-as-a-judge has become a promising paradigm for evaluating model generations, but the lack of reliability limits its deployment in applications. It has been very common to use LLMs in model evaluation while uncertainty quantification for rating evaluation remains underexplored. This work presents the first analysis framework to offer interval evaluations in LLM-based scoring via conformal prediction. Conformal prediction constructs continuous prediction intervals from a single evaluation run and we design an ordinal boundary adjustment for discrete rating tasks. We also suggest a midpoint-based score within the interval as a low-bias alternative to raw model score and weighted average. Extensive experiments and analysis across evaluators and conformal predictors show that our framework provides reliable uncertainty quantification for LLM-as-a-judge.

1 Introduction

Large language models (LLMs) have become powerful automatic evaluators for natural language generation (NLG) tasks, known as LLM-as-a-judge. Its consistency with human judgments results in strong performance with respect to metrics like ROUGE [24] and BLEU [31]. Besides, LLM judges can flexibly adapt to diverse evaluation criteria and provide scalable, cost-effective assessments compared to expert annotation [8, 11]. These advantages make the LLM-as-a-judge useful in various scenarios, such as cyberattack detection [60] and wildlife trafficking identification [3].

However, a single evaluation from a LLM judge might be biased [52, 21] and uncertain due to inherent randomness [35], thus undermining its reliability in scenarios like healthcare [4] and finance [17]. Though a LLM judge can express its confidence with well-designed prompt or via fine-tuning [56, 26, 43], it may still suffer from overconfidence [55] or dishonesty [23]. We ask: *How can a LLM judge provide reliable evaluation given the user request?*

Conformal prediction [47] is a distribution-free method to quantify the uncertainty of an LLM judge [58]. It outputs a prediction interval (or set for classification) to a model output with statistically guaranteed coverage using only a calibration step, as long as the data is exchangeable. In this paper, we comprehensively evaluate nine conformal prediction methods in quantifying the uncertainty of scoring evaluation by the average width and coverage rate of prediction intervals.

In summary, our contributions are

- We are the first to analyze the uncertainty of LLM-as-a-judge using conformal prediction in rating-based evaluation, which uses the output from a single evaluation run.
- We design a boundary adjustment, which improves the efficiency empirically without compromising the coverage. The interval points suggest better alignment to human evaluation.
- We analyze factors affecting the interval quality, including judge framework, the choice of LLM, and the size of calibration, and offer practical insights or recommended choices.

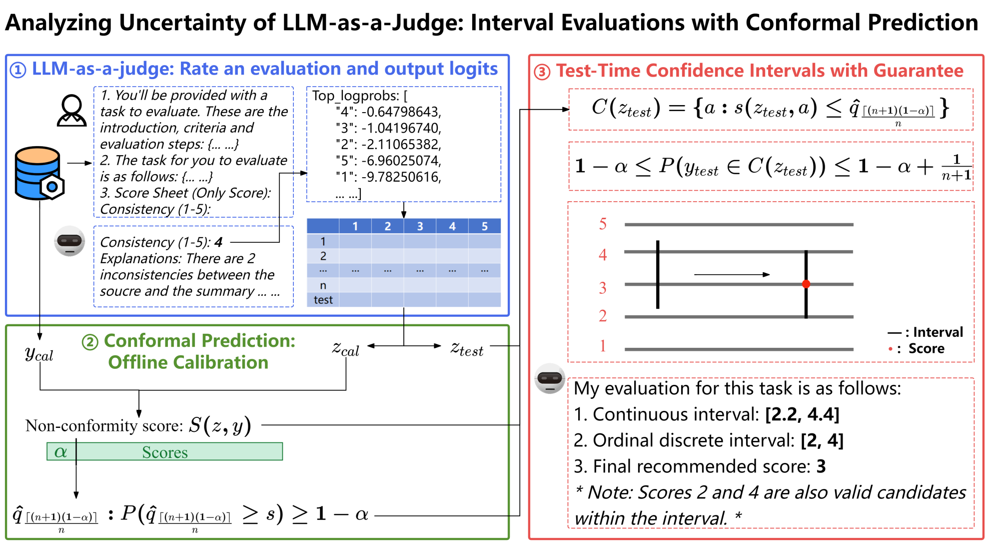

Figure 1: Overview of uncertainty quantification in rating. More details can be seen in Appendix A.2.

2 Analyzing Uncertainty of LLM Judges

Framework. As shown in Figure 1, we evaluate a task following G-Eval [27] and extract the logits of certain tokens (e.g., 1 - 5) at the position of rating token only. Then these logits compose the input for conformal predictors, which output intervals with theoretical coverage guarantee of real label through an offline calibration. For each test point, our framework provides a continuous interval, a discrete interval adjusted by rounding the boundaries and the middle point as a recommended score.

Evaluations We run evaluations on benchmarks in text summarization (SummEval [6]), dialogue summarization (DialSumm [7]) and reasoning (ROSCOE [10]). We primarily adopt G-Eval [27] and SocREval [14] with a CoT prompt (Appendix A.5). Evaluations are mainly conducted using GPT-4o mini (2024-07-18), DeepSeek-R1-Distill-Qwen-32B [5], and Qwen2.5-72B-Instruct [33].

Conformal Prediction. In our experiments, we employ 9 conformal predictors (CQR [34], Asymmetric CQR [36], CHR [37], LVD [25], Boosted CQR and Boosted LCP [54], R2CCP [13], Ordinal APS [28] and Ordinal Risk Control [57]). Detailed introduction with setting is provided in Appendix A.7. For each method, we target on 90% coverage, split of dataset into 50% calibration set and 50% test set with 30 random seeds (1-30) and report average interval width and coverage rate.

3 Experiments and Analysis

In tables of continuous intervals and discret intervals, Gray marks coverage <85%, <u>underline</u> marks coverage between 85%–90%, and **bold** highlights the smallest interval width among methods achieving $\geq$90% coverage for each evaluator–dimension. Across all experimental settings (Table 1 and Table 10), most conformal predictors consistently generate prediction intervals with coverage close to the 90%. However, some methods show unsatisfying coverage especially in small-sample conditions (ROSCOE), further highlighting the importance of proper calibration (see Figure 2).

Table 2 and 11 show the effect of boundary adjustment to mitigate miscoverage: all conformal regressors exhibit increased coverage after adjustment, with the majority achieving stable coverage at or above the 90%. We provide a theoretical explanation for this (Theorem 1). Moreover, a mild adjustment (e.g. adjust to nearest label within 0.1 radius) is effective (Table 12 and 13).

Our analysis shows that DeepSeek-R1-Distill-Qwen-32B provides the most consistent coverage; under the G-Eval framework we observe higher coverage (at the cost of slightly wider bands) versus SocREval; among predictors, R2CCP strikes the best balance between coverage and width.

Method	SummEval Evaluated with G-Eval				ROSCOE Evaluated with SocREval			
	Consistency	Coherence	Fluency	Relevance	CosmosQA	DROP	e-SNLI	GSM8K
				GPT-4o mini				
Boosted CQR	1.01 / 87.75%	2.73 / 87.80%	1.54 / 88.68%	2.00 / 87.42%	3.15 / 80.07%	2.63 / 78.57%	1.82 / 80.26%	3.08 / 82.50%
Boosted LCP	0.76 / 89.22%	2.67 / 87.34%	0.92 / 89.18%	1.91 / 87.19%	3.60 / 83.91%	2.92 / 85.40%	1.88 / 81.23%	3.36 / 85.93%
CHR	0.67 / 88.99%	2.41 / 82.96%	0.94 / 88.86%	1.74 / 82.62%	2.54 / 73.06%	1.86 / 68.92%	1.36 / 72.24%	1.98 / 78.67%
Asym CQR	1.25 / 94.97%	2.91 / 93.76%	1.60 / 93.75%	2.13 / 91.42%	3.90 / 98.71%	3.91 / 98.60%	**2.87/ 96.67%**	3.89 / 98.80%
Sym CQR	1.15 / 94.16%	**2.87 / 93.15%**	1.44 / 92.92%	**2.09 / 90.92%**	**3.53 / 95.27%**	**3.82/96.70%**	3.04 / 96.62%	**3.53 / 95.67%**
LVD	1.01 / 92.35%	2.73 / 89.76%	**1.11 / 90.59%**	2.02 / 89.55%	3.10 / 83.95%	2.49 / 83.05%	2.17 / 86.18%	3.08 / 89.57%
R2CCP	**0.69 / 90.88%**	2.62 / 89.63%	0.92 / 89.36%	1.97 / 89.70%	2.96 / 85.85%	2.43 / 84.73%	1.75 / 84.02%	2.15 / 85.07%
				DeepSeek-R1-Distill-Qwen-32B				
Boosted CQR	1.10 / 89.30%	2.36 / 88.98%	1.16 / 89.46%	2.00 / 88.98%	3.17 / 82.72%	2.47 / 81.11%	1.79 / 80.96%	2.94 / 79.83%
Boosted LCP	0.77 / 89.20%	2.32 / 86.70%	0.93 / 89.10%	1.91 / 86.89%	3.48 / 81.60%	2.79 / 85.46%	1.84 / 80.61%	3.43 / 85.23%
CHR	0.82 / 91.17%	2.23 / 87.07%	0.90 / 89.24%	1.87 / 86.38%	2.66 / 76.50%	1.95 / 78.06%	1.38 / 71.97%	2.01 / 81.60%
Asym CQR	1.30 / 95.13%	2.72 / 92.86%	1.49 / 94.52%	2.21 / 92.06%	3.84 / 99.08%	3.95 / 99.27%	2.86 / 96.05%	3.85 / 98.43%
Sym CQR	1.16 / 93.88%	2.67 / 92.50%	1.31 / 93.01%	2.13 / 91.05%	**3.48 / 96.70%**	**3.83 / 96.35%**	2.97 / 96.36%	3.46 / 95.60%
LVD	0.97 / 92.93%	2.43 / 91.10%	1.00 / 91.10%	**2.04 / 90.14%**	3.25 / 88.10%	2.62 / 88.06%	**2.24 / 90.96%**	**3.02/90.63%**
R2CCP	**0.69 / 90.44%**	**2.30 / 90.12%**	**0.89 / 90.09%**	2.00 / 89.84%	2.94 / 86.97%	2.29 / 86.35%	1.85 / 87.87%	1.88 / 85.33%
				Qwen2.5-72B-Instruct				
Boosted CQR	0.80 / 88.28%	2.46 / 87.82%	1.24 / 89.22%	1.88 / 87.17%	3.05 / 79.08%	2.56 / 81.17%	1.51 / 77.11%	2.81 / 80.67%
Boosted LCP	0.67 / 88.81%	2.43 / 86.92%	0.94 / 89.26%	1.86 / 87.51%	3.46 / 80.41%	2.81 / 85.75%	1.74 / 77.50%	3.38 / 86.23%
CHR	0.61 / 89.04%	2.14 / 80.93%	0.98 / 88.93%	1.61 / 79.61%	2.44 / 72.65%	2.08 / 75.87%	1.22 / 69.69%	1.81 / 77.50%
Asym CQR	1.11 / 94.47%	2.80 / 93.13%	1.63 / 94.79%	2.17 / 92.21%	3.86 / 99.01%	3.89 / 98.67%	**2.77 / 96.84%**	3.87 / 98.97%
Sym CQR	0.98 / 93.10%	2.73 / 92.25%	1.44 / 93.73%	2.11 / 91.30%	**3.37 / 94.80%**	3.79 / 97.02%	3.01 / 97.37%	3.35 / 95.33%
LVD	0.85 / 92.82%	2.55 / 90.49%	1.09 / 90.94%	1.94 / 89.27%	3.05 / 84.29%	**2.67 / 90.57%**	1.91 / 85.96%	**2.83/ 90.13%**
R2CCP	**0.61 / 90.73%**	2.44 / 89.54%	**0.95 / 90.17%**	**1.98 / 90.45%**	2.90 / 85.34%	2.39 / 86.25%	1.59 / 84.50%	2.00 / 86.73%

Table 1: Continuous intervals: SummEval evaluated by G-Eval and ROSCOE evaluated by SocREval.

Method	SummEval Evaluated with G-Eval				ROSCOE Evaluated with SocREval			
	Consistency	Coherence	Fluency	Relevance	CosmosQA	DROP	e-SNLI	GSM8K
				GPT-4o mini				
Boosted CQR	0.99 / 92.81%	2.73 / 93.02%	1.54 / 94.38%	2.00 / 92.93%	3.20 / 93.40%	2.63 / 89.65%	1.82 / 92.15%	3.09 / 91.17%
Boosted LCP	0.74 / 91.90%	2.68 / 93.53%	**0.90 / 90.88%**	**1.91 / 92.70%**	3.60 / 95.48%	3.01 / 91.27%	1.90 / 91.80%	3.26 / 92.17%
CHR	0.70 / 91.79%	2.41 / 87.78%	0.94 / 90.60%	1.74 / 88.10%	2.56 / 82.45%	1.87 / 78.86%	1.34 / 83.46%	1.94 / 83.23%
Asym CQR	1.25 / 96.02%	2.90 / 95.41%	1.60 / 94.57%	2.14 / 94.14%	3.90 / 98.84%	3.91 / 98.73%	2.87 / 96.89%	3.89 / 98.80%
Sym CQR	1.15 / 95.45%	2.87 / 94.94%	1.44 / 93.80%	2.09 / 93.56%	3.53 / 95.34%	3.82 / 97.05%	3.04 / 96.89%	3.53 / 95.67%
LVD	1.01 / 94.11%	2.73 / 93.72%	1.12 / 92.70%	2.03 / 93.82%	**3.13 / 91.53%**	**2.52 / 90.22%**	2.17 / 94.82%	**3.09/93.37%**
R2CCP	**0.68 / 92.15%**	**2.62 / 92.81%**	0.91 / 90.99%	1.97 / 93.38%	2.93 / 89.46%	2.41 / 89.21%	**1.71 / 90.11%**	2.09 / 86.93%
OrdinalAPS	2.28 / 71.48%	1.88 / 64.84%	1.78 / 13.65%	2.36 / 87.94%	0.73 / 47.52%	0.83 / 55.08%	0.72 / 52.76%	0.58 / 73.90%
OrdinalRC	2.41 / 75.19%	2.02 / 67.38%	1.93 / 14.58%	2.51 / 90.30%	0.82 / 49.46%	0.91 / 57.11%	0.80 / 54.61%	0.60 / 74.43%
				DeepSeek-R1-Distill-Qwen-32B				
Boosted CQR	1.08 / 93.55%	2.37 / 93.96%	1.15 / 93.48%	2.01 / 93.72%	3.20 / 95.71%	**2.52 / 93.30%**	**1.79 / 93.25%**	2.94 / 92.23%
Boosted LCP	0.76 / 92.03%	2.32 / 92.37%	0.93 / 91.34%	1.92 / 92.81%	3.46 / 95.95%	2.80 / 91.94%	1.87 / 92.89%	3.36 / 93.63%
CHR	0.87 / 93.96%	**2.23 / 91.42%**	0.91 / 91.98%	**1.87 / 90.84%**	2.69 / 86.80%	1.97 / 85.90%	1.39 / 85.96%	2.01 / 86.60%
Asym CQR	1.31 / 95.99%	2.72 / 94.83%	1.49 / 95.57%	2.21 / 94.53%	3.84 / 99.08%	3.95 / 99.27%	2.88 / 96.45%	3.84 / 98.47%
Sym CQR	1.15 / 95.02%	2.67 / 94.34%	1.32 / 94.44%	2.13 / 93.67%	3.48 / 96.80%	3.82 / 96.54%	2.99 / 96.80%	3.46 / 95.63%
LVD	0.97 / 95.01%	2.44 / 94.58%	1.00 / 93.21%	2.04 / 94.12%	3.28 / 95.27%	2.67 / 93.75%	2.24 / 96.36%	3.03 / 94.40%
R2CCP	**0.68 / 91.57%**	2.30 / 93.22%	**0.89 / 91.80%**	1.99 / 92.96%	**2.91 / 90.58%**	2.25 / 89.97%	**1.80 / 92.35%**	1.82 / 86.93%
OrdinalAPS	2.51 / 90.06%	2.52 / 90.64%	3.76 / 91.08%	2.13 / 89.98%	1.32 / 60.00%	1.26 / 78.22%	1.46 / 87.85%	1.50 / 85.67%
OrdinalRC	2.54 / 90.11%	2.56 / 91.18%	3.73 / 89.53%	2.14 / 90.07%	1.44 / 62.35%	1.33 / 78.22%	1.52 / 88.33%	1.55 / 86.07%
				Qwen2.5-72B-Instruct				
Boosted CQR	0.81 / 92.36%	2.47 / 93.06%	1.25 / 93.66%	1.88 / 92.81%	3.10 / 94.01%	2.56 / 90.79%	**1.49 / 92.11%**	2.82 / 92.03%
Boosted LCP	0.65 / 91.26%	2.44 / 92.26%	0.93 / 91.20%	**1.86 / 92.57%**	3.40 / 94.90%	2.84 / 92.41%	1.79 / 91.84%	3.33 / 92.90%
CHR	0.66 / 92.21%	2.14 / 86.10%	0.98 / 91.16%	1.61 / 85.78%	2.49 / 82.14%	2.05 / 82.89%	1.18 / 84.56%	1.79 / 85.27%
Asym CQR	1.10 / 95.47%	2.79 / 94.70%	1.64 / 95.63%	2.17 / 94.85%	3.85 / 99.18%	3.89 / 98.67%	2.77 / 97.06%	3.87 / 98.97%
Sym CQR	0.98 / 94.35%	2.72 / 94.18%	1.45 / 94.79%	2.10 / 94.02%	3.36 / 95.07%	3.79 / 97.08%	3.01 / 97.68%	3.34 / 95.33%
LVD	0.85 / 95.11%	2.56 / 94.05%	1.09 / 93.45%	1.95 / 93.86%	**3.07 / 92.01%**	2.67 / 93.87%	1.91 / 95.53%	2.87 / 93.43%
R2CCP	**0.59 / 91.83%**	**2.43 / 92.78%**	**0.95 / 92.12%**	1.98 / 93.72%	2.88 / 89.29%	**2.34 / 90.00%**	1.55 / 90.20%	1.96 / 88.57%
OrdinalAPS	2.86 / 90.18%	3.01 / 90.59%	3.05 / 45.43%	2.75 / 90.29%	0.71 / 55.99%	0.25 / 56.83%	0.67 / 77.68%	0.46 / 70.87%
OrdinalRC	2.85 / 90.00%	2.96 / 89.35%	3.21 / 53.31%	2.75 / 90.14%	0.75 / 57.28%	0.29 / 56.83%	0.80 / 79.74%	0.49 / 71.37%

Table 2: Discrete intervals: SummEval evaluated by G-Eval and ROSCOE evaluated by SocREval.

As an application of intervals, we select the best interval types for midpoint evaluation and compare against the raw score in LLM response and the weighted average derived from token probabilities. Table 3 and 14 show that, while the midpoint estimates achieve comparable or even slightly better correlation with the ground-truth scores, they significantly reduce more than 90% of prediction error.

Using R2CCP, we construct continuous intervals under four calibration regimes—25%, 50%, 75%, and 100% of the whole calibration set, and show the tendency in Figure 2, which highlights the importance of sufficiently large calibration sets to stabilize coverage around the required threshold.

We also explore the potential of intervals in decision-making by reprompting judges with intervals information (See Appendix A.11.5). Figs. 6 to 11 show the examples of responses in reprompting.

Method	Coherence				Consistency				Fluency				Relevance			
	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ
GPT-4o mini																
Raw Score	1.729	1.055	0.446	0.373	1.674	1.073	0.480	0.437	3.907	1.977	0.219	0.197	1.009	0.786	0.512	0.427
Weighted Sum	1.643	1.037	0.514	0.379	1.548	1.066	0.478	0.383	3.412	1.733	0.319	0.250	0.865	0.737	0.567	0.419
Con_R2CCP	**0.791**	**0.716**	0.512	0.373	**0.510**	**0.432**	0.455	0.371	**0.442**	**0.491**	**0.330**	**0.261**	**0.418**	**0.509**	0.546	0.403
Dis_R2CCP	**0.794**	**0.715**	0.508	**0.386**	**0.512**	**0.428**	**0.506**	**0.468**	**0.443**	**0.488**	**0.336**	**0.300**	**0.423**	**0.509**	0.540	0.423
DeepSeek-R1-Distill-Qwen-32B																
Raw Score	1.010	0.775	0.549	0.457	1.229	0.770	0.467	0.425	2.843	1.549	0.387	0.355	0.763	0.682	0.520	0.437
Weighted Sum	0.869	0.734	0.599	0.447	1.439	1.065	0.468	0.375	2.783	1.564	0.420	0.332	0.646	0.632	0.565	0.419
Con_R2CCP	**0.599**	**0.619**	**0.663**	**0.492**	**0.564**	**0.446**	0.445	0.361	**0.373**	**0.455**	0.391	0.311	**0.431**	**0.513**	0.555	0.412
Dis_R2CCP	**0.602**	**0.619**	**0.661**	**0.508**	**0.566**	**0.441**	0.462	0.423	**0.375**	**0.454**	0.393	0.351	**0.434**	**0.512**	0.548	0.431
Qwen2.5-72B-Instruct																
Raw Score	1.432	0.981	0.426	0.358	2.068	1.237	0.458	0.416	4.476	1.958	0.310	0.281	1.188	0.903	0.498	0.420
Weighted Sum	1.282	0.932	0.539	0.395	1.847	1.213	0.483	0.387	4.236	1.928	0.363	0.285	1.091	0.885	0.555	0.412
Con_R2CCP	**0.675**	**0.659**	**0.603**	**0.444**	**0.469**	**0.396**	0.465	0.378	**0.414**	**0.486**	0.340	0.269	**0.407**	**0.502**	**0.571**	**0.425**
Dis_R2CCP	**0.678**	**0.659**	**0.600**	**0.456**	**0.469**	**0.387**	**0.538**	**0.498**	**0.416**	**0.485**	0.342	**0.306**	**0.411**	**0.501**	**0.566**	**0.444**

Table 3: Comparison of interval midpoints with LLM scoring baselines on SummEval.

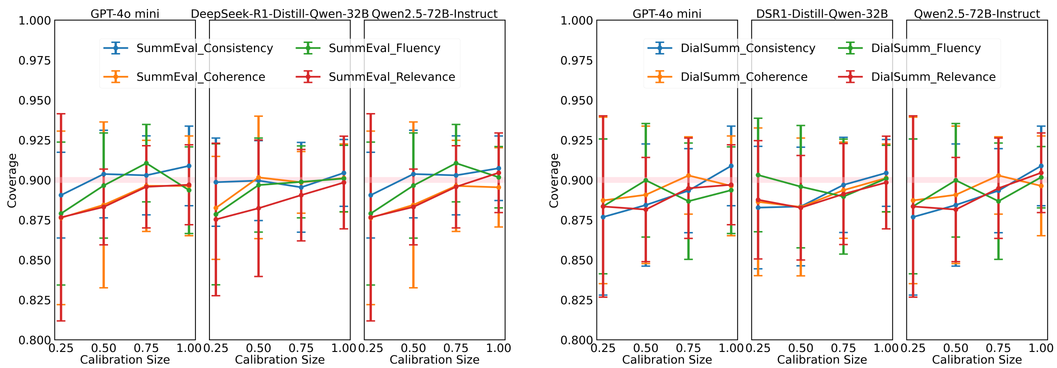

Figure 2: Coverage increase to 90% and error bars shrink as calibration set expands.

4 Conclusion and Discussion

This work introduces the application of Conformal Prediction methods for quantifying LLM scoring uncertainty based on single-output logits. We provide the first analysis of applying CP in estimating LLM scoring uncertainty: by employing nine distinct CP methods across three LLM judge models, two evaluation frameworks, and multiple datasets, we construct continuous and discrete prediction intervals that achieve or approximate 90% confidence coverage. Moreover, we design a theoretically grounded boundary adjustment technique that transforms continuous intervals to discrete rating scales, yielding a global improvement in coverage and enhancing the reliability and interpretability of the intervals. Finally, we explore using interval midpoints as calibrated scores to assess the utility of interval estimation within the LLM-as-a-judge paradigm, which offers a more accurate and robust alternative for score estimation directly from rating-type LLM-as-a-judge. Experimental results demonstrate that this strategy matches or slightly surpasses baselines on correlation metrics while significantly outperforming direct scoring on error metrics, thereby achieving higher accuracy.

To make use of the intervals, the quantified uncertainty by prediction intervals is the key. As long as an LLM judge is used for evaluation, an interval evaluation helps users to determine when they can trust the judgment. On the one hand, a wider prediction interval serves as a warning signal of unreliability with the score, which is particularly beneficial in high-risk environments where uncertainty-induced errors must be minimized, such as in medical diagnosis [28, 41]. On the other hand, a narrower prediction interval suggests a higher degree of certainty in the score, thereby indicating selective prediction and reducing the need for manual review in automated evaluation, such as in essay scoring [39]. We believe our framework might be helpful in example selection to avoid the model collapse when trained on LLM generated data [38], since wide intervals could contribute to active learning. We also believe that interval-based evaluations could provide a principled foundation for building uncertainty-aware pairwise comparison and listwise ranking frameworks, potentially enabling more reliable and generalizable evaluation methodologies [49, 51].

References

[1] Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2022. URL https://arxiv.org/abs/2107.07511.

[2] Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control, 2022. URL https://arxiv.org/abs/2110.01052.

[3] Juliana Barbosa, Ulhas Gondhali, Gohar A. Petrossian, Kinshuk Sharma, Sunandan Chakraborty, Jennifer Jacquet, and Juliana Freire. A cost-effective llm-based approach to identify wildlife trafficking in online marketplaces. 2025. URL https://api.semanticscholar.org/CorpusID:278208290.

[4] Philip Chung, Akshay Swaminathan, Alex J. Goodell, Yeasul Kim, S. Momsen Reincke, Lichy Han, Ben Deverett, Mohammad Amin Sadeghi, Abdel-Badih Ariss, Marc Ghanem, David Seong, Andrew A. Lee, Caitlin E. Coombes, Brad Bradshaw, Mahir A. Sufian, Hyo Jung Hong, Teresa P. Nguyen, Mohammad R. Rasouli, Komal Kamra, Mark A. Burbridge, James C. McAvoy, Roya Saffary, Stephen P. Ma, Dev Dash, James Xie, Ellen Y. Wang, Clifford A. Schmiesing, Nigam Shah, and Nima Aghaeepour. Verifact: Verifying facts in llm-generated clinical text with electronic health records, 2025. URL https://arxiv.org/abs/2501.16672.

[5] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[6] Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation, 2021. URL https://arxiv.org/abs/2007.12626.

[7] Mingqi Gao and Xiaojun Wan. DialSummEval: Revisiting summarization evaluation for dialogues. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, pages 5693–5709, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.418. URL https://aclanthology.org/2022.naacl-main.418/.

[8] Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. Llm-based nlg evaluation: Current status and challenges, 2024. URL https://arxiv.org/abs/2402.01383.

[9] Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Jingyu Sun, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024. URL https://arxiv.org/abs/2406.12793.

[10] Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Roscoe: A suite of metrics for scoring step-by-step reasoning, 2023. URL https://arxiv.org/abs/2212.07919.

[11] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on llm-as-a-judge, 2025. URL https://arxiv.org/abs/2411.15594.

[12] Leying Guan. Localized conformal prediction: A generalized inference framework for conformal prediction, 2022. URL https://arxiv.org/abs/2106.08460.

[13] Etash Kumar Guha, Shlok Natarajan, Thomas Möllenhoff, Mohammad Emtiyaz Khan, and Eugene Ndiaye. Conformal prediction via regression-as-classification. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rulxyXjf46.

[14] Hangfeng He, Hongming Zhang, and Dan Roth. Socreval: Large language models with the socratic method for reference-free reasoning evaluation, 2024. URL https://arxiv.org/abs/2310.00074.

[15] Sameer Jain, Vaishakh Keshava, Swarnashree Mysore Sathyendra, Patrick Fernandes, Pengfei Liu, Graham Neubig, and Chunting Zhou. Multi-dimensional evaluation of text summarization with in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, page 8487–8495. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.537. URL http://dx.doi.org/10.18653/v1/2023.findings-acl.537.

[16] Jaehun Jung, Faeze Brahman, and Yejin Choi. Trust or escalate: Llm judges with provable guarantees for human agreement, 2024. URL https://arxiv.org/abs/2407.18370.

[17] Kiran Kamble, Melisa Russak, Dmytro Mozolevskyi, Muayad Ali, Mateusz Russak, and Waseem AlShikh. Expect the unexpected: Failsafe long context qa for finance, 2025. URL https://arxiv.org/abs/2502.06329.

[18] Klaus-Rudolf Kladny, Bernhard Schölkopf, and Michael Muehlebach. Conformal generative modeling with improved sample efficiency through sequential greedy filtering, 2025. URL https://arxiv.org/abs/2410.01660.

[19] Bhawesh Kumar, Charlie Lu, Gauri Gupta, Anil Palepu, David Bellamy, Ramesh Raskar, and Andrew Beam. Conformal prediction with large language models for multi-choice question answering, 2023. URL https://arxiv.org/abs/2305.18404.

[20] Haitao Li, Junjie Chen, Qingyao Ai, Zhumin Chu, Yujia Zhou, Qian Dong, and Yiqun Liu. Calibraeval: Calibrating prediction distribution to mitigate selection bias in llms-as-judges, 2024. URL https://arxiv.org/abs/2410.15393.

[21] Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. Llms-as-judges: A comprehensive survey on llm-based evaluation methods, 2024. URL https://arxiv.org/abs/2412.05579.

[22] Minzhi Li, Zhengyuan Liu, Shumin Deng, Shafiq Joty, Nancy F. Chen, and Min-Yen Kan. Dna-eval: Enhancing large language model evaluation through decomposition and aggregation, 2024. URL https://arxiv.org/abs/2405.15329.

[23] Siheng Li, Cheng Yang, Taiqiang Wu, Chufan Shi, Yuji Zhang, Xinyu Zhu, Zesen Cheng, Deng Cai, Mo Yu, Lemao Liu, Jie Zhou, Yujiu Yang, Ngai Wong, Xixin Wu, and Wai Lam. A survey on the honesty of large language models, 2024. URL https://arxiv.org/abs/2409.18786.

[24] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*, 2004. URL https://api.semanticscholar.org/CorpusID:964287.

[25] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Locally valid and discriminative prediction intervals for deep learning models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8378–8391. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/46c7cb50b373877fb2f8d5c4517bb969-Paper.pdf.

[26] Shudong Liu, Zhaocong Li, Xuebo Liu, Runzhe Zhan, Derek F. Wong, Lidia S. Chao, and Min Zhang. Can LLMs learn uncertainty on their own? expressing uncertainty effectively in a self-training manner. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21635–21645, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1205. URL https://aclanthology.org/2024.emnlp-main.1205/.

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

[28] Charles Lu, Anastasios N. Angelopoulos, and Stuart Pomerantz. Improving trustworthiness of ai disease severity rating in medical imaging with ordinal conformal prediction sets, 2022. URL https://arxiv.org/abs/2207.02238.

[29] Christopher Mohri and Tatsunori Hashimoto. Language models with conformal factuality guarantees, 2024. URL https://arxiv.org/abs/2402.10978.

[30] Tomoko Nemoto and David Beglar. Likert-scale questionnaires. In *JALT 2013 conference proceedings*, pages 1–8, 2014.

[31] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, 2002. URL https://api.semanticscholar.org/CorpusID:11080756.

[32] Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling, 2024. URL https://arxiv.org/abs/2306.10193.

[33] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[34] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

[35] Kayla Schroeder and Zach Wood-Doughty. Can you trust llm judgments? reliability of llm-as-a-judge, 2024. URL https://arxiv.org/abs/2412.12509.

[36] Matteo Sesia and Emmanuel J. Candès. A comparison of some conformal quantile regression methods. *Stat*, 9, 2019. URL https://api.semanticscholar.org/CorpusID:202565594.

[37] Matteo Sesia and Yaniv Romano. Conformal prediction using conditional histograms, 2021. URL https://arxiv.org/abs/2105.08747.

[38] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022): 755–759, jul 2024. doi: 10.1038/s41586-024-07566-y. URL https://doi.org/10.1038/s41586-024-07566-y.

[39] Yishen Song, Qianta Zhu, Huaibo Wang, and Qinhua Zheng. Automated essay scoring and revising based on open-source large language models. *IEEE Transactions on Learning Technologies*, 17:1880–1890, 2024. doi: 10.1109/TLT.2024.3396873.

[40] Jiayuan Su, Jing Luo, Hongwei Wang, and Lu Cheng. Api is enough: Conformal prediction for large language models without logit-access, 2024. URL https://arxiv.org/abs/2403.01216.

[41] Ting Fang Tan, Kabilan Elangovan, Liyuan Jin, Yao Jie, Li Yong, Joshua Lim, Stanley Poh, Wei Yan Ng, Daniel Lim, Yuhe Ke, Nan Liu, and Daniel Shu Wei Ting. Fine-tuning large language model (llm) artificial intelligence chatbots in ophthalmology and llm-based evaluation using gpt-4, 2024. URL https://arxiv.org/abs/2402.10083.

[42] Vianney Taquet, Vincent Blot, Thomas Morzadec, Louis Lacombe, and Nicolas Brunel. Mapie: an open-source library for distribution-free uncertainty quantification, 2022. URL https://arxiv.org/abs/2207.12274.

[43] Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in llms, 2025. URL https://arxiv.org/abs/2502.06233.

[44] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback, 2023. URL https://arxiv.org/abs/2305.14975.

[45] Gerrit J. J. van den Burg, Gen Suzuki, Wei Liu, and Murat Sensoy. Aligning black-box language models with human judgments, 2025. URL https://arxiv.org/abs/2502.04997.

[46] Harit Vishwakarma, Alan Mishler, Thomas Cook, Niccolo Dalmasso, Natraj Raman, and Sumitra Ganesh. Prune 'n predict: Optimizing LLM decision-making with conformal prediction. In *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI*, 2025. URL https://openreview.net/forum?id=gENDUvkytD.

[47] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

[48] Nico Wagner, Michael Desmond, Rahul Nair, Zahra Ashktorab, Elizabeth M. Daly, Qian Pan, Martín Santillán Cooper, James M. Johnson, and Werner Geyer. Black-box uncertainty quantification method for llm-as-a-judge, 2024. URL https://arxiv.org/abs/2410.11594.

[49] Victor Wang, Michael J. Q. Zhang, and Eunsol Choi. Improving llm-as-a-judge inference with the judgment distribution, 2025. URL https://arxiv.org/abs/2503.03064.

[50] Zhiyuan Wang, Jinhao Duan, Lu Cheng, Yue Zhang, Qingni Wang, Xiaoshuang Shi, Kaidi Xu, Hengtao Shen, and Xiaofeng Zhu. Conu: Conformal uncertainty in large language models with correctness coverage guarantees, 2024. URL https://arxiv.org/abs/2407.00499.

[51] Tianjun Wei, Wei Wen, Ruizhi Qiao, Xing Sun, and Jianghong Ma. Rocketeval: Efficient automated llm evaluation via grading checklist, 2025. URL https://arxiv.org/abs/2503.05142.

[52] Minghao Wu and Alham Fikri Aji. Style over substance: Evaluation biases for large language models, 2023. URL https://arxiv.org/abs/2307.03025.

[53] Qiujie Xie, Qingqiu Li, Zhuohao Yu, Yuejie Zhang, Yue Zhang, and Linyi Yang. An empirical analysis of uncertainty in large language model evaluations, 2025. URL https://arxiv.org/abs/2502.10709.

[54] Ran Xie, Rina Foygel Barber, and Emmanuel J. Candès. Boosted conformal prediction intervals, 2024. URL https://arxiv.org/abs/2406.07449.

[55] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms, 2024. URL https://arxiv.org/abs/2306.13063.

[56] Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. Sayself: Teaching llms to express confidence with self-reflective rationales, 2024. URL https://arxiv.org/abs/2405.20974.

[57] Yunpeng Xu, Wenge Guo, and Zhi Wei. Conformal risk control for ordinal classification, 2024. URL https://arxiv.org/abs/2405.00417.

[58] Fanghua Ye, Mingming Yang, Jianhui Pang, Longyue Wang, Derek F Wong, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu. Benchmarking llms via uncertainty quantification. *arXiv preprint arXiv:2401.12794*, 2024.

[59] Gal Yona, Roee Aharoni, and Mor Geva. Can large language models faithfully express their intrinsic uncertainty in words?, 2024. URL https://arxiv.org/abs/2405.16908.

[60] Javier Yong, Haokai Ma, Yunshan Ma, Anis Yusof, Zhenkai Liang, and Ee-Chien Chang. Attackseqbench: Benchmarking large language models' understanding of sequential patterns in cyber attacks, 2025. URL https://arxiv.org/abs/2503.03170.

[61] Mozhi Zhang, Mianqiu Huang, Rundong Shi, Linsen Guo, Chong Peng, Peng Yan, Yaqian Zhou, and Xipeng Qiu. Calibrating the confidence of large language models by eliciting fidelity, 2024. URL https://arxiv.org/abs/2404.02655.

[62] Qiyuan Zhang, Yufei Wang, Yuxin Jiang, Liangyou Li, Chuhan Wu, Yasheng Wang, Xin Jiang, Lifeng Shang, Ruiming Tang, Fuyuan Lyu, and Chen Ma. Crowd comparative reasoning: Unlocking comprehensive evaluations for llm-as-a-judge, 2025. URL https://arxiv.org/abs/2502.12501.

[63] Han Zhou, Xingchen Wan, Yinhong Liu, Nigel Collier, Ivan Vulić, and Anna Korhonen. Fairer preferences elicit improved human-aligned large language model judgments, 2024. URL https://arxiv.org/abs/2406.11370.

A Appendix

A.1 Related Work

Uncertainty Quantification for LLM-as-a-Judge. Uncertainty quantification for LLM-as-a-judge is an important yet less explored area. Wagner et al. [48] prompt the judge to justify each rating option as if it were correct and then construct a confusion matrix from token-level probabilities of these assessments to derive confidence scores. Xie et al. [53] use token probabilities to estimate the confidence of judgments, and demonstrate that such measures exhibit bias and instability through extensive experiments. Similar conclusions are also found when applying other two common paradigms: (1) prompting LLMs to self-report confidence [59, 56], which can suffer from overconfidence [55] or dishonesty [23]; and (2) consistency-based approaches that rely on multiple generations [44, 55], which, like the confusion matrix-based method, are computationally expensive. To our best knowledge, Jung et al. [16] is the most relevant work to us, which applies conformalized risk control [2] to ensure agreement with human preferences in pairwise response comparison [63, 20, 22, 62, 45]. In contrast, we focus on using conformal prediction to quantify uncertainty in rating tasks instead of pairwise preference modeling.

Conformal Prediction for LLMs. Conformal prediction [47] has drawn interest for uncertainty quantification in LLMs [58] due to its distribution-free and post-hoc nature with provable statistical guarantee. Owing to these advantages, recent works primarily apply conformal prediction to classification tasks, such as multiple-choice question answering [19, 61, 40, 46] and response selection for factual consistency [32, 29, 50, 18]. These studies typically focus on ensuring that the correct answer is included in a unordered prediction set. However, we focus on providing intervals that reflect the variability in LLM judgments in rating tasks, which has ordinal preference.

A.2 Preliminaries and Details of our Framework

LLM-as-a-Judge. In recent years, LLMs have been widely adopted as evaluators to score NLG tasks, which commonly yields a predicted score $\hat{y}$ on a Likert scale [30]. Following G-Eval [27], given a prompt p and a generated text x to be evaluated, an LLM judge M is expected to produce a response

$$M(p, x) = (z, \hat{y}), \tag{1}$$

where z denotes the logits and $\hat{y}$ is a scalar score according to a predefined scale. Note that, for the logits, we only need to extract the logits of certain tokens (e.g., 1, 2, 3, 4, 5 if in a Likert scale) at the position of rating token only. Other than rating tasks, LLM-as-a-judge can also be applied to other evaluation paradigms [11], such as pairwise comparison or ranking, in which candidate outputs are first scored by the LLM judge and then compared or ordered based on those scores [49, 51].

Conformal Prediction. Conformal prediction [47] is a model-agnostic uncertainty quantification method. It constructs a prediction interval (or a set for classification) with coverage guarantee, free of training or prompting the judge model or assumptions about the underlying data distribution. In our work, we adopt split conformal prediction [47], which quantify the uncertainty with a held-out calibration set. A non-conformity score function $s(z, y)$ is computed for each point in the calibration set, to measure how "unusual" a prediction $\hat{y}$ is to a ground truth y. For regression tasks, the non-conformity score is often defined as

$$s(z, y) = |\hat{y} - y|. \tag{2}$$

Given a user-desired miscoverage rate α, the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$-quantile $\hat{q}$ of these scores is then used to construct the prediction interval for the prediction $\hat{y}_{test}$ of any test point

$$\mathcal{C}(z_{test}, \hat{y}_{test}) = [\hat{y}_{test} - \hat{q},\ \hat{y}_{test} + \hat{q}], \tag{3}$$

or equivalently

$$\mathcal{C}(z_{test}, \hat{y}_{test}) = \{a : s(z_{test}, a) \leq \hat{q}\}. \tag{4}$$

Such a prediction interval satisfies the coverage guarantee [1]

$$1 - \alpha \leq \mathbb{P}\left(y_{test} \in \mathcal{C}\left(z_{test}, \hat{y}_{test}\right)\right) \leq 1 - \alpha + \frac{1}{n+1}, \tag{5}$$

as long as the calibration set and test set are exchangeable, i.e., the joint distribution remain the same after any permutations on these two sets.

A.2.1 From Logits to Intervals

We focus on quantifying the uncertainty using conformal prediction in rating tasks (e.g., in Likert scale). An overview of the workflow is presented in Figure 1.

Extract Logits as Feature As our framework targets uncertainty estimation in discrete rating tasks, the token-level logits corresponding to Likert-scale scores (e.g., 1–5) are used as features.

As shown in Figure 1, an LLM judge, prompted with a chain-of-thought (CoT) instruction (Example in Appendix A.5) that specifies an output format, generates a response containing its rating. After accurately locating the target score token "4", we extract the log probabilities of all potential score tokens (e.g., 1–5). To ensure semantic consistency, we aggregate the probabilities of tokens with equivalent meanings(e.g., "two" vs. 2).

As a result, we obtain a K-dimensional feature vector z representing the logits associated with each candidate score token in $\{1, 2, \ldots, K\}$[1], which composes the input for conformal prediction, i.e. $\{z, y\}$, which is found to have these properties: (1) Independent and identically distributed, at least exchangeable[2]; (2) $\mathbb{E}[\Sigma_{i=1}^{k} e^{z_i}] = 1$, which causes the interdependence among variables; (3) heteroskedasticity[3]; (4) Isolated distribution of label caused by rating nature. These properties inspires our method choice for interval construction.

Notably, token probabilities are frequently seen in early works but they would cause multicollinearity in regression. Thus they are only used in ordinal predictors (because they need) and cause unstable performance due to well-known bias.

Interval Estimation As discussed in Section A.2, modern conformal prediction methods vary in how they define non-conformity scores and construct intervals, yet they share a unified structure that ensures valid coverage. In our framework, we go beyond the basic absolute-error formulation and adopt a diverse set of nine conformal predictors. Each is designed to handle specific data characteristics—such as asymmetry, heteroskedasticity, or ordinal outputs—and offers complementary strengths. This diversity allows our framework to remain robust and adaptable across different evaluation scenarios.

The selected methods include quantile regression–based approaches (e.g., CQR [34], Asymmetric CQR [36]), histogram-based estimators (e.g., CHR [37]), kernel regression variants (e.g., LVD [25]), boosted methods [54] (e.g., Boosted CQR and Boosted LCP), and ordinal classification–based predictors (e.g., R2CCP [13], Ordinal APS [28] and Ordinal Risk Control [57]). A complete summary of their non-comformity score functions and construction principles is provided in Appendix A.7.

A.2.2 Boundary Adjustment

In addition to an interval in Equation (3), our framework further transform the regression problem to a ordinal classification problem by a boundary adjustment, due to the ordinal and discrete nature of rating. Therefore the interval boundaries will be aligned with potential labels, instead of continuous scores that might have no exact meaning.

For conformal prediction, we redefine the non-conformity score function as:

$$s'(z, y) = s(z, y') = \begin{cases} s(z, \lceil y \rceil) & \text{if } y \leq \lfloor \hat{y} \rfloor, \\ s(z, y) & \text{otherwise}, \\ s(z, \lfloor y \rfloor) & \text{if } y \geq \lceil \hat{y} \rceil. \end{cases} \tag{6}$$

Because all potential labels y' are integers in rating evaluation, this new function ensures the scores consistent on calibration set. However, it transforms the interval from Equation 4 to

$$\mathcal{C}(z_{test}) = \{a : s'(x_{test}, a) \leq \hat{q}\} = [l, u] \to [l', u'], \tag{7}$$

[1] We use $K = 5$ for the standard Likert scale or GPA-like settings, but K can be adapted to other granularities (e.g., 7 or 10) depending on the evaluation scale.

[2] It depends on the generation task to evaluate. If the dateset comprises summaries and annotations generated by multiple models from the same source document, the features extracted by the LLM judge are unlikely to be i.i.d. However, exchangeability is guaranteed for permutation invariance of evaluations.

[3] Hypothesis testing results are shown in Appendix A.6.

where $l' = \lceil l \rceil$ and $u' = \lfloor u \rfloor$.

We shrink the boundaries to interior labels by cutting excessive areas because they cover no potential labels, while y in these areas share a same $\hat{q}$ with the labels, which means this adjustment has no influence to coverage. For example, $[2.9, 4.2]$ will be shrunk to $[3, 4]$ with the same coverage since $s'(z, 2.9) = s'(z, 3)$ and $s'(z, 4.2) = s'(z, 4)$.

On the other hand, we can also expand an interval to mitigate the marginal miscoverage caused by isolated label distribution and limited calibration size. For example, the interval $[2.2, 3.9]$ only cover 3 but can be expanded to $[2, 4]$, then a pitiful miscoverage can be avoided if the ground truth is 2 or 4. This improvement of coverage can be explained by a larger $\hat{q}$ for each boundary, which theoretically ensures more abnormal results to be covered.

The following theorem shows the non-decreasing coverage after boundary adjustment, with its proof shown in Appendix A.2.3. Other discrete ganularities (e.g. GPA scale) are also applicable after linear transformation to integers.

Theorem 1 (Non-decreasing Coverage After Boundary Adjustment)**.** *Based on coverage guarantee in Equation (5), we transform the non-conformity score function $s(x, y)$ by Equation (6) and adjust an continuous interval by Equation (7).*

Then, if the adjustment is performed by shrinking ($l' = \lceil l \rceil$ and $u' = \lfloor u \rfloor$), coverage preserves:

$$\mathbb{P}\Big(Y_{test} \in \mathcal{C}'(x_{test})\Big) \geq 1 - \alpha.$$

And if at least one boundary is expanded ($l' = \lfloor l \rfloor$ or $u' = \lceil u \rceil$), coverage is expected to increase:

$$\mathbb{P}\Big(Y_{test} \in \mathcal{C}'(x_{test})\Big) > 1 - \alpha.$$

A.2.3 Proof of Boundary Adjustment Non-decreasing Coverage Guarantee

Proof. By the standard split conformal prediction procedure with the nonconformity score $s(x, y) = |\hat{y} - y|$, the prediction set

$$\mathcal{C}(x_{test}) = \{z \in \mathbb{R} : s(x_{test}, z) \leq \hat{q}_{1-\alpha}\}$$

satisfies

$$\mathbb{P}(Y_{test} \in \mathcal{C}(x_{test})) \geq 1 - \alpha.$$

In our discrete setting, every potential label is an element of a predetermined ordered set (e.g., $\{1, 2, 3, 4, 5\}$). The adjusted score $s'(x, y)$ is defined such that for each y,

$$s'(x, y) = s\Big(x, y'\Big),$$

where y' is the label nearest to y from the appropriate side.

In regions where the original interval $\mathcal{C}(x_{test}) = [l, u]$ already contains some labels, the shrinking adjustment leads to

$$s'(x_{test}, l) = s\Big(x_{test}, \lceil l \rceil\Big) \leq \hat{q}_{1-\alpha}$$

or

$$s'(x_{test}, u) = s\Big(x_{test}, \lfloor u \rfloor\Big) \leq \hat{q}_{1-\alpha}.$$

Thus, every label that was originally covered (i.e., satisfying $s(x_{test}, y) \leq \hat{q}$) remains covered, ensuring that the coverage remains unchanged.

On the other hand, suppose that an expanding adjustment is performed, we have

$$s'(x_{test}, l) \leq \hat{q}_{1-\alpha} \leq s'\Big(x_{test}, \lfloor l \rfloor\Big) \leq \hat{q}_{1-\alpha_0}$$

or

$$s'(x_{test}, u) \leq \hat{q}_{1-\alpha} \leq s'\Big(x_{test}, \lceil u \rceil\Big) \leq \hat{q}_{1-\alpha_0},$$

where $0 \leq \alpha_0 < \alpha$.

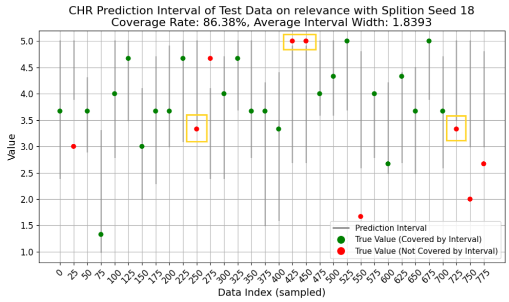

Figure 3: Red points mean the labels lying outside the intervals, which could turn green (inside) if the interval just extend to nearest labels (e.g. 3.33 and 5).

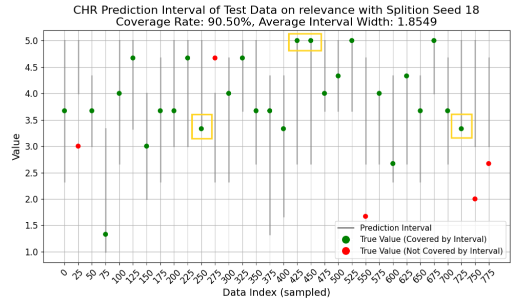

Figure 4: After applying boundary adjustment, the coverage in this instance improves from 86.38% to 90.50%, while the average width increases slightly to 1.8549.

In this case, for any $z \notin \mathcal{C}(x_{test})$, it is possible that $z \in \mathcal{C}'(x_{test})$ for z could be $\lfloor l \rfloor$ or $\lceil u \rceil$. As a consequence, if the original interval barely missed covering the label, the expansion guarantees that these outcomes are now covered.

Hence, the event

$$\{Y_{test} \in \mathcal{C}(x_{test})\} \subseteq \{Y_{test} \in \mathcal{C}'(x_{test})\},$$

which implies

$$\mathbb{P}\Big(Y_{test} \in \mathcal{C}'(x_{test})\Big) \geq \mathbb{P}\Big(Y_{test} \in \mathcal{C}(x_{test})\Big) \geq 1 - \alpha.$$

Moreover,

$$\begin{aligned}
&\mathbb{P}\Big(Y_{\text{test}} \in \mathcal{C}'(x_{\text{test}})\Big) - \mathbb{P}\Big(Y_{\text{test}} \in \mathcal{C}(x_{\text{test}})\Big) \\
&= \mathbb{P}\Big(q_{1-\alpha} \leq s'\Big(x_{test}, \lfloor l \rfloor \text{ or } \lceil u \rceil\Big) \leq q_{1-\alpha_0}\Big) \\
&= (1 - \alpha_0) - (1 - \alpha) = \alpha - \alpha_0 > 0.
\end{aligned}$$

Thus

$$\mathbb{P}\Big(Y_{test} \in \mathcal{C}'(x_{test})\Big) > 1 - \alpha.$$

$\square$

During the transition from continuous to discrete intervals, we observe consistent improvements in empirical coverage across all experimental settings. A concrete example is illustrated in Figure 3 and 4, where certain ground-truth scores fall just outside the estimated intervals. In such cases, a marginal upward shift in the estimated quantiles would suffice to restore coverage. This demonstrates that a relatively modest increase in interval width can lead to a substantial gain in calibration, successfully achieving the 90% coverage.

A.2.4 Midpoints as Calibrated Scores

To make use of prediction interval, one can take its midpoint as a suggested score. The interval provides coverage guarantees but offers no indication of the direction toward the true label. Even if biased, the midpoint is the minimum-variance estimator of the true label given the endpoints.

A.3 Limitations

This paper has main limitations in tasks for LLM to judge. Our experimental results are primarily based on summarization and reasoning in NLG tasks, with a focus on the SummEval, DialSumm and ROSCOE. Additionally, we acknowledge that there are numerous other tasks that we have yet to explore, including but not limited to machine translation, multimodal generation, etc.

A.4 Ethics

Our work analyzes conformal prediction for reliable LLM-based evaluation. While it can quantify uncertainty and mitigate bias, there could also be potential ethical concerns. First, our framework relies on human annotations for calibration. If these annotations contain subjective or biased judgments (e.g., gender bias), the resulting intervals may reflect such biases and distort model evaluations. For example, our work could reduce workload and mitigate bias in essay scoring [1] with some calibration from teachers. But biased annotations may systematically underrate certain linguistic styles, leading to unfairly low and wide prediction intervals for some students. Second, our interval estimates are nonparametric and should not be interpreted as classical confidence intervals in statistics. The midpoint is a heuristic and convenient choice, not a statistical mean or mode, and does not imply symmetric or continuous uncertainty. Misinterpreting this could result in misleading conclusions. We will add these discussions in the revised version.

A.5 Prompt Used in Our Analysis

In our analysis, we adopted the LLM-as-a-judge frameworks G-Eval [27] across all tasks and SocREval [14] specifically for reasoning tasks, making only minimal prompt adjustments to suit each evaluation. Below we provide three representative prompt examples: the relevance evaluations on SummEval, and the ROSCOE evaluations under both G-Eval and SocREval.

Prompt on Relevance of SummEval

You'll be handed a summary of a news article.

Your challenge is to rate how well the summary captures the essence of the article.

Make sure to thoroughly read and understand these instructions before diving in. Keep this guide handy as you work through the task, so you can refer back to it if needed.

Evaluation Criteria:

Relevance (1–5): Does the summary hit the mark by including the most important content from the original article? It should focus on the key details without wandering into irrelevant or repetitive information. If the summary strays or over-explains, it should be rated lower.

How to Evaluate:

1. Read both the source article and the summary attentively.
2. Compare the two, identifying the critical points of the article.
3. Judge how well the summary captures these important points and avoids unnecessary details.
4. Give the summary a relevance score between 1 and 5.

Source Article:

`{{Document}}`

Summary:

`{{Summary}}`

Evaluation Form (ENTER A SCORE BETWEEN 1–5):

Relevance:

Prompt on ROSCOE by G-Eval

You will receive a generated response based on the question.

Your mission is to assess whether the generated response answers the question in a well-justified manner.

Please pay close attention to the instructions and keep this guide handy while completing your review. Feel free to refer back to it as needed.

Evaluation Criterion:
Quality (1–5): 1=incomprehensible and wrong, 5=clear and correct

Steps for Evaluation:

1. Carefully read the Situation (Premise) and the Claim (Hypothesis) to understand their key elements.
2. Evaluate the response to determine if it logically connects the Situation to the Claim and sufficiently explains whether the Claim is supported (or not) based on the details provided.
3. Score the response's quality on a scale of 1 to 5, with 1 being the lowest and 5 the highest, based on clarity, correctness, and the justification provided.

Situation (Premise):
`{{Premise}}`

Claim (Hypothesis):
`{{Hypothesis}}`

Question:
`{{Question}}`

Generated response:
`{{Response}}`

Evaluation Form (Only provide the overall quality score from 1–5 without any text):
Quality:

It is worth noting that, through oversight, we applied the SummEval prompt template directly to the DialSumm task. Even though DialSumm is a dataset for dialogue summarization rather than news summarization, the resulting intervals on DialSumm still exhibited great performance.

Prompt on ROSCOE by SocREval

Instruction:

Does the generated response answer the question in a well-justified manner? Please generate your own response for the question first, then conduct a qualitative analysis on the generated response by taking into account your own response, and finally give me an overall quality score in [1, 2, 3, 4, 5] (1=incomprehensible and wrong, 5=clear and correct). Note that you need to consider both the explanation and the answer in the generated response.

Example:

- **Situation (Premise):** Two women are embracing while holding to go packages.
- **Claim (Hypothesis):** Two women are holding packages.
- **Question:** Is the Claim supported by the Situation?
- **Generated response:** The two women are most likely embracing because they are either friends or family. If they were just holding packages, there would be no need for them to embrace. The answer is Yes.
- **Example representation:**
  - "own response": "Yes, the Claim is supported by the Situation. The Situation states that two women are ëmbracing while holding to go packages,ẅhich means they are holding packages. Therefore, the Claim is accurate.",
  - "qualitative analysis": "The generated response starts by making an assumption about why the two women might be embracing, suggesting they could be friends or family. While this could be true, it is not directly relevant to the question asked. The core of the question is about the Claim's accuracy given the Situation. The latter part of the generated response does correctly determine that the answer is Ÿes.Ḧowever, the earlier assumption makes the response longer and potentially more confusing.",
  - "overall quality": 4

Actual Input:
- **Situation (Premise):** `{{Premise}}`
- **Claim (Hypothesis):** `{{Hypothesis}}`
- **Question:** `{{Question}}`
- **Generated response:** `{{Response}}`

Evaluation Request:

Please generate your own response for the question first, then conduct a qualitative analysis on the generated response by taking into account your own response, and finally give me the overall quality of the given generated response for the question by taking into account both your own response and the qualitative analysis based on the instruction and the format of the example representation.

Evaluation Form (Only provide the overall quality score from 1–5 without any text):

Quality:

A.6 Hypothesis Testing of Heteroskedasticity

In order to assess the validity of regression-based conformal prediction and to guide our choice of conformal prediction (CP) methods, we perform two classical tests for heteroskedasticity: the Breusch–Pagan (BP) test and the White test. This phenomenon indicates a non-constant residual variance, which causes deviation in coverage rates and inefficiency in interval widths. This data property also motivates the development of modern CP algorithms such as CQR [34], LCP [12], and R2CCP [13].

Breusch–Pagan Test. The BP test regresses the squared OLS residuals $\hat{e}_i^2$ on the original covariates X. Under the null hypothesis of homoskedasticity,

$$H_0 : \mathrm{Var}(\varepsilon_i) = \sigma^2 \quad \text{vs.} \quad H_1 : \mathrm{Var}(\varepsilon_i) = \sigma^2 h(X_i),$$

the test statistic

$$\mathrm{LM}_{BP} \;=\; n\,R^2_{\hat{e}^2 \sim X} \;\dot{\sim}\; \chi^2_k,$$

where n is the sample size and $k = \dim(X)$. A small p-value indicates rejection of homoskedasticity.

White Test. The White test extends BP by including not only X but also their squares and pairwise interactions $Z = \{X,\, X^2,\, X_i X_j\}$ in the auxiliary regression of $\hat{e}_i^2$. The statistic

$$\mathrm{LM}_{White} \;=\; n\,R^2_{\hat{e}^2 \sim Z} \;\dot{\sim}\; \chi^2_m,$$

with $m = \dim(Z)$. Unlike BP, White's method does not require specifying the form of $h(\cdot)$.

Test Results. Table 4 reports both BP and White p-values across our datasets and evaluators.

- **SummEval / DialSumm (G-Eval)**: All four metrics and all models exhibit highly significant heteroskedasticity ($p < 10^{-12}$).
- **ROSCOE by G-Eval**: CosmosQA remains homoskedastic, whereas DROP, e-SNLI and GSM8k show $p < 0.05$.
- **ROSCOE by SocREval**: Heteroskedasticity is confined to DROP (for DSR1-Qwen-32B and GPT-4o-mini) and to CosmosQA/e-SNLI (for Qwen2.5-72B).

SummEval by G-Eval

Evaluator	Test	Consistency				Coherence				Fluency				Relevance			
		LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value
GPT-4o-mini	BP	372.121	3.02e-78	96.615	4.52e-89	147.034	5.71e-30	32.261	2.07e-31	144.954	1.58e-29	31.759	6.35e-31	102.860	1.32e-20	21.903	2.85e-21
	White	446.359	4.68e-82	30.547	2.61e-97	204.285	1.60e-32	11.556	4.96e-35	187.021	4.08e-29	10.450	3.71e-31	132.282	1.45e-18	7.116	1.85e-19
DSR1-Qwen-32B	BP	332.234	1.17e-69	83.545	4.44e-78	64.602	1.36e-12	13.414	7.81e-13	209.266	2.95e-43	47.970	2.46e-46	78.494	1.73e-15	16.447	7.41e-16
	White	406.728	8.21e-74	26.910	4.32e-86	142.666	1.58e-20	7.729	1.33e-21	242.606	3.52e-40	14.111	6.53e-44	92.448	2.76e-11	4.841	1.19e-11
Qwen2.5-72B	BP	351.775	7.26e-74	89.844	2.03e-83	82.248	2.84e-16	17.276	1.11e-16	227.917	2.99e-47	52.956	5.94e-51	83.830	1.32e-16	17.627	4.96e-17
	White	407.695	5.17e-74	26.996	2.33e-86	142.134	1.99e-20	7.697	1.71e-21	245.423	9.55e-41	14.304	1.40e-44	100.688	9.49e-13	5.302	3.34e-13

DialSumm by G-Eval

Evaluator	Test	Consistency				Coherence				Fluency				Relevance			
		LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value
GPT-4o-mini	BP	70.220	9.22e-14	14.723	4.30e-14	199.050	4.54e-41	46.209	2.85e-44	250.633	4.02e-52	60.796	2.06e-57	87.825	1.92e-17	18.664	5.49e-18
	White	96.250	5.87e-12	5.091	2.01e-12	238.533	2.32e-39	14.160	1.56e-43	271.824	4.40e-46	16.613	9.80e-52	170.231	7.82e-26	9.548	9.86e-28
DSR1-Qwen-32B	BP	100.158	4.90e-20	21.483	9.27e-21	126.174	1.54e-25	27.616	9.87e-27	169.680	8.54e-35	38.451	4.63e-37	177.728	1.64e-36	40.540	5.12e-39
	White	169.039	1.33e-25	9.468	1.83e-27	196.532	5.45e-31	11.260	1.21e-33	225.735	8.54e-37	13.255	1.83e-40	250.758	8.03e-42	15.045	1.65e-46
Qwen2.5-72B	BP	88.782	1.21e-17	18.877	3.37e-18	209.551	2.57e-43	49.076	6.76e-47	199.737	3.23e-41	46.395	1.92e-44	125.827	1.83e-25	27.532	1.19e-26
	White	123.892	5.40e-17	6.694	7.13e-18	228.974	1.92e-37	13.482	3.09e-41	235.628	8.89e-39	13.953	7.83e-43	175.737	6.61e-27	9.897	6.01e-29

ROSCOE by G-Eval

Evaluator	Test	CosmosQA				DROP				e-SNLI				GSM8k			
		LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value
GPT-4o-mini	BP	5.839	0.3222	1.167	0.3270	11.334	0.0451	2.328	0.0440	26.074	0.0001	6.053	0.0000	7.586	0.1806	1.530	0.1822
	White	17.194	0.6404	0.841	0.6609	23.456	0.2669	1.188	0.2681	35.174	0.0192	1.974	0.0124	26.151	0.1609	1.346	0.1556
DSR1-Qwen-32B	BP	8.042	0.1539	1.626	0.1550	20.313	0.0011	4.369	0.0008	24.209	0.0002	5.537	0.0001	15.828	0.0074	3.335	0.0065
	White	17.670	0.6092	0.867	0.6290	40.833	0.0039	2.281	0.0022	58.598	0.0000	4.122	0.0000	33.872	0.0270	1.825	0.0210
Qwen2.5-72B	BP	7.883	0.1628	1.592	0.1641	22.042	0.0005	4.785	0.0004	22.554	0.0004	5.092	0.0002	27.782	0.0000	6.259	0.0000
	White	25.904	0.1690	1.333	0.1640	31.326	0.0510	1.657	0.0438	49.770	0.0002	3.196	0.0000	56.739	0.0000	3.545	0.0000

ROSCOE by SocREval

Evaluator	Test	CosmosQA				DROP				e-SNLI				GSM8k			
		LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value	LM Stat.	p-value	F Stat.	p-value
GPT-4o-mini	BP	7.256	0.20231	1.461	0.20457	4.016	0.54705	0.796	0.55399	7.637	0.17742	1.545	0.17954	3.577	0.61180	0.707	0.61918
	White	20.130	0.26762	1.199	0.26973	37.301	0.01077	2.041	0.00732	12.766	0.75172	0.722	0.77546	11.661	0.82022	0.663	0.83626
DSR1-Qwen-32B	BP	5.659	0.34085	1.130	0.34606	29.404	0.00002	6.643	0.00001	6.994	0.22105	1.409	0.22457	4.244	0.51487	0.841	0.52199
	White	13.283	0.86492	0.636	0.88162	38.105	0.00860	2.095	0.00561	18.955	0.52477	0.933	0.54694	8.983	0.98311	0.421	0.98678
Qwen2.5-72B	BP	13.470	0.01935	2.805	0.01810	8.464	0.13245	1.714	0.13293	16.545	0.00545	3.569	0.00450	2.321	0.80313	0.456	0.80886
	White	34.356	0.00755	2.227	0.00499	22.780	0.19917	1.291	0.19706	29.926	0.03818	1.813	0.02981	12.387	0.77613	0.707	0.79345

Table 4: Breusch-Pagan (BP) and White tests detect pervasive heteroscedasticity in SummEval and DialSumm: both tests yield highly significant p-values (p < 1e-12) across all four metrics and all evaluators. By contrast, in ROSCOE by G-Eval only DROP, e-SNLI and GSM8k exhibit significant heteroscedasticity (p < 0.05) while CosmosQA remains homoscedastic; in ROSCOE by SocREval heteroscedasticity is confined to DROP for DeepSeek-R1-Qwen-32B and GPT-4omini and to CosmosQA and e-SNLI for Qwen2.5-72B-Instruct.

A.7 Summary of CP Methods

In our analysis, we employ a total of seven regression-based conformal prediction (CP) methods to generate continuous prediction intervals (e.g., [3.2,4.1]), as well as two ordinal classification-based CP methods to produce ordered discrete intervals (e.g., [3,4]). In this subsection, we provide a detailed discussion of these CP approaches, including the motivation behind our choice to focus on regression and ordinal formulations rather than commonly used risk-control-based methods. We further elaborate on how each method computes nonconformity scores and constructs predictive intervals accordingly.

A.7.1 Why not use classification methods?

As mentioned, early work has primarily applied conformal prediction to classification-style tasks, which produces non-ordered prediction set, e.g. {A, C} in multiple choice question answering. Admittedly, the rating scale {1,2,3,4,5} can be cast as a multiple-choice classification problem. However, it is unclear how to interpret a predicted set such as {1,5}: what does it mean for both the lowest and highest scores to be both plausible, and nothing in between? As Wang et al. [49] have shown, judgment distributions from LLMs can be irregular or even bimodal, making such fragmented prediction sets not only difficult to interpret, but also problematic for downstream decision-making.

In contrast, regression-based and ordinal conformal predictors generate ordered prediction intervals, offering a coherent and interpretable depiction of score variability. These intervals communicate not just inclusion, but range—what is the highest plausible score, and what is the lowest? In high-stakes applications such as medical diagnosis, this becomes crucial. For example, if an LLM evaluator assigns a rating of 3 (e.g., "moderate condition"), a disjoint set like {1,5} offers confusing insight. On the other hand, a calibrated interval such as [3,5] conveys that the case might be severe, thus signaling the need for a more cautious and proactive treatment plan.

A.7.2 Continuous CP methods

The following gives a brief description of each CP method used in our experiments, including its nonconformity score, interval construction procedure and how we employ.

Conformalized Quantile Regression (CQR) [34]

- *Nonconformity score:*

$$s_i = \max\{\hat{q}_{\alpha/2}(x_i) - y_i,\ y_i - \hat{q}_{1-\alpha/2}(x_i)\},$$

  where $\hat{q}_\tau$ is the τ–quantile regression estimator.
- *Interval construction:* Compute s_i on calibration set and let $Q_{1-\alpha}$ be the $(1-\alpha)$-quantile of $\{s_i\}$. For a test input x, form

$$\big[\hat{q}_{\alpha/2}(x) - Q_{1-\alpha},\ \hat{q}_{1-\alpha/2}(x) + Q_{1-\alpha}\big].$$

- *Deployment:* We implement Conformalized Quantile Regression (CQR) using the `MapieQuantileRegressor` from the `mapie` package [42], with a `GradientBoostingRegressor` (configured for quantile loss) as the base estimator for quantile regression.

Asymmetric CQR [36]

- *Nonconformity scores:*

$$s_i^\ell = \hat{q}_\alpha(x_i) - y_i, \quad s_i^u = y_i - \hat{q}_{1-\alpha}(x_i).$$

- *Interval construction:* Let Q_ℓ and Q_u be the $(1-\alpha)$-quantiles of $\{s_i^\ell\}$ and $\{s_i^u\}$, respectively. Then

$$\big[\hat{q}_\alpha(x) - Q_\ell,\ \hat{q}_{1-\alpha}(x) + Q_u\big].$$

- *Deployment:* Same with CQR but the asymmetric variant.

Conditional Histogram Regression (CHR) [37]

- *Distribution estimation:* Partition the target range into bins and estimate $\Pr(Y \in \text{bin} \mid X = x)$ via a black-box model.
- *Nested set series:* Based on conditional probability, construct a series of nested set $\{C_t\}_{t=0}^T$, where T is the length of the series and C_t expands as t increases.
- *Compute conformity score*

$$s_i = \min\{t \in \{0, ..., T\} : y \in C_t\}$$

  on calibration set and obtain estimated quantile $s_{1-\alpha} = \hat{t}$.
- *Interval construction:* Find the $\hat{t}$-th set $C_{\hat{t}}$ in $\{C_t(x_{test})\}_{t=0}^T$ for the test point.
- *Deployment:* We estimate the conditional distributions by QNet estimator with two hidden layers of 256 units each, a batch size of 32, learning rate 5×10^{-4} and 1000 epochs.

Locally Valid and Discriminative (LVD) [25]

- *Nonconformity scores:* For each calibration example (x_{n+i}, y_{n+i}), compute the absolute residual
$$R_i = \left| y_{n+i} - \hat{y}_{n+i} \right|,$$
where $\hat{y}_{n+i}$ is the model's point prediction (e.g. from a deep network or kernel regression).
- *Interval construction:* For a test input x, assign similarity weights
$$w_i \propto K_f\big(x_{n+i},\, x\big) \quad \text{and} \quad w_\infty \propto K_f(x, x)$$
(normalized so $\sum_i w_i + w_\infty = 1$), form the weighted empirical distribution of $\{R_i\}$ with a "safe-guard" atom at ∞, take its $(1-\alpha)$–quantile Q, and output
$$\big[\hat{y}(x) - Q,\ \hat{y}(x) + Q\big].$$
- *Deployment:* We train the kernel similarity function using KernelMLKR with parameters `d=10, seed=0, n_iters=500, norm=True, lr=1e-3` that used in their demo notebook.

Boosted Conformal Prediction [54]

- *Boosting the conformity score:* BoostedCP optimizes conformity score functions from baselines like CQR [34] or LCP [12] via gradient boosting. This is guided by a tailored loss function, aiming for enhanced conditional coverage or reduced interval length. It operates post-model training, solely relying on model predictions.
- *Interval Estimation:* The boosted score function is used for calibration to compute empirical quantiles. These quantiles, with the boosted score, construct final intervals for testing points. This approach improves prediction interval statistical properties while maintaining valid marginal coverage.
- *Deployment:* We set `n_rounds_cv` as 500 and `learning_rate` as 0.02.

R2CCP (Regression-to-Classification Conformal Prediction) [13]

- *Two-stage approach:* Partition the continuous response range $[y_{\min}, y_{\max}]$ into K equally spaced bins with midpoints $\{\hat{y}_k\}_{k=1}^{K}$. Train a softmax-output neural network to classify.
- *Non-conformity score:* On the calibration set, compute for each pair the interpolated probability $\sigma_j = \bar{q}_\theta(y_j \mid x_j)$ by linearly interpolating q_θ between adjacent bin midpoints.
- *Interval construction:* Obtain $(1-\alpha)$-quantile $\hat{q}_{1-\alpha}$ of non-conformity scores and generate intervals by
$$C_{1-\alpha}(x_{test}) = \{\, z \in \mathbb{R} : \bar{q}_\theta(z \mid x) \geq \hat{q}_{1-\alpha} \,\}.$$
- *Deployment:* We train R2CCP model with `max_epoches = 100`. In practice, this method might yields fragmental intervals. We merge those intervals into one by taking minimum and maximum. Moreover, the range of labels in calibration determines the bin split in testing. Thus there would be error if two ranges are inconsistent, which causes that the trials of random experiments are sometimes slightly less than 30.

A.7.3 Ordinal CP methods

Ordinal CP methods generate intervals by softmax probabilities, which derives from judgment distribution. For GPA-scale tasks, we obtain probabilities of fractional labels (e.g. 1.33, 1.67, ...) by linear interpolation. For ordinal CP methods, since they directly produce discrete intervals, boundary adjustment has no influence to their intervals for the boundaries are already on the potential labels{1.00, 1.33, ..., 4.67, 5.00}

Ordinal APS [28]

- *Nonconformity score:* Nonconformity score equals to 1 if the true label lies in the interval and 0 if not. Obtain an empirical quantile λ as the threshold of accumulated probability mass.
- *Interval construction:* Start from the label with highest probability, and then extend to both directions until the accumulated probability mass reach the quantile.

Ordinal Risk Control [57]

- *Nonconformity score:* Similar to Ordinal APS but calculate the empirical risk by weighted average of nonconformity scores. Select a smallest quantile λ to control the empirical risk.

- *Interval Estimation:* Similar to Ordinal APS, start from point estimation and extend to both directions until the miscoverage risk is higher than λ.
- *Deployment:* We deploy the `WeightedCRPredictor` variant for better performance in our tasks.

A.7.4 Setting and Comparison

Table 5 shows the hyperparameter setting of each method as a complement to the method introduction above.

Method	Hyperparameter
Boosted LCP	len_local_boost: n_rounds_cv=500, learning_rate=0.02, store=True, verbose=False
Boosted CQR	len_cqr_boost: same with Boosted LCP
CHR	QNet estimator, batch_size=32, hidden_dim=256, lr=5e-4, epoch=1000
Asymmetric CQR	MAPIE, gradient boosting regressor with quantile loss
Symmetric CQR	same as Asymmetric CQR
LVD	DNN_model, readout_layer = pretrain_general(seed=0, quiet=True, model_setting=0), kernel_model = KernelMLKR(d=10, seed=0, n_iters=500, norm=True, lr=1e-3)
R2CCP	Default setting but max_epochs=100
OrdinalAPS	Default setting
OrdinalRC	Default setting with WeightedCRPredictor

Table 5: Hyperparameters for Different Methods

Table 6 shows the runtime and memory analysis of each method. Boosted CQR and Boosted LCP are having higher cost potentially due to boosting. For LVD, the main reason for higher cost might be due to computing the kernel matrix for hundreds of iterations to quantify pairwise similarities.

Method	Time Mean (s)	Time Std. (s)	Memory Mean (MB)	Memory Std. (MB)
Boosted CQR	91.43	3.24	0.89	0.05
Boosted LCP	87.27	2.57	2.05	0.01
CHR	9.54	0.25	0.62	0.01
Symmetric CQR	0.83	0.03	0.35	0.00
Asymmetric CQR	0.82	0.03	0.35	0.00
LVD	93.67	2.82	0.55	0.01
R2CCP	9.25	0.53	1.35	0.00
OrdinalAPS	0.01	0.00	0.20	0.00
OrdinalRC	0.03	0.00	0.19	0.00

Table 6: Comparison of Methods in Terms of Time and Memory

A.8 Prompt sensitivity

Prompt	Consistency	Fluency	Coherence	Relevance
		GPT-4o		
CoT 0	$0.86 \pm 0.24/89.65\% \pm 1.48\%$	$1.03 \pm 0.14/89.45\% \pm 1.71\%$	$3.10 \pm 0.23/89.08\% \pm 4.29\%$	$2.46 \pm 0.11/90.20\% \pm 2.09\%$
CoT 1	$0.83 \pm 0.35/88.90\% \pm 1.89\%$	$1.09 \pm 0.16/90.22\% \pm 1.74\%$	$2.99 \pm 0.16/88.52\% \pm 3.62\%$	$2.37 \pm 0.10/89.08\% \pm 2.75\%$
CoT 2	$0.92 \pm 0.41/89.42\% \pm 2.37\%$	$1.14 \pm 0.19/90.17\% \pm 1.52\%$	$2.98 \pm 0.15/89.15\% \pm 3.52\%$	$2.41 \pm 0.10/89.08\% \pm 2.39\%$
CoT 3	$0.88 \pm 0.36/89.30\% \pm 2.06\%$	$1.10 \pm 0.21/89.92\% \pm 2.20\%$	$3.02 \pm 0.10/89.55\% \pm 2.18\%$	$2.35 \pm 0.09/89.03\% \pm 2.51\%$
CoT 4	$0.81 \pm 0.33/89.35\% \pm 1.56\%$	$1.11 \pm 0.13/90.30\% \pm 1.78\%$	$3.06 \pm 0.13/89.50\% \pm 3.03\%$	$2.42 \pm 0.11/90.42\% \pm 2.38\%$
		GPT-4o mini		
CoT 0	$0.74 \pm 0.23/88.78\% \pm 1.72\%$	$1.12 \pm 0.19/90.12\% \pm 1.69\%$	$2.96 \pm 0.09/88.33\% \pm 2.27\%$	$2.38 \pm 0.11/89.65\% \pm 1.85\%$
CoT 1	$0.87 \pm 0.23/89.58\% \pm 1.66\%$	$1.18 \pm 0.28/90.50\% \pm 1.66\%$	$3.02 \pm 0.11/89.12\% \pm 2.13\%$	$2.39 \pm 0.11/89.15\% \pm 2.22\%$
CoT 2	$0.88 \pm 0.29/89.75\% \pm 1.70\%$	$1.06 \pm 0.15/89.85\% \pm 1.37\%$	$3.01 \pm 0.12/89.83\% \pm 2.31\%$	$2.42 \pm 0.15/89.98\% \pm 3.06\%$
CoT 3	$0.82 \pm 0.34/89.45\% \pm 1.80\%$	$1.08 \pm 0.14/88.85\% \pm 2.64\%$	$2.96 \pm 0.08/90.35\% \pm 1.69\%$	$2.41 \pm 0.12/89.03\% \pm 2.47\%$
CoT 4	$0.82 \pm 0.31/88.88\% \pm 1.87\%$	$1.13 \pm 0.22/89.95\% \pm 1.98\%$	$3.04 \pm 0.08/89.10\% \pm 1.70\%$	$2.50 \pm 0.13/90.97\% \pm 2.32\%$

Table 7: GPT-4o and GPT-4o mini on 5 seeds comparison

GPT-4o mini provides a budget-friendly option. Our preliminary experiments on GPT-4o-mini vs. GPT-4o showed that their interval quality are similar. These pieces of evidence show that GPT-4o-mini is a high-quality LLM judge. Regarding Qwen2.5-72B-Instruct and DeepSeek-R1-Distill-Qwen-32B, we choose them because they are all widely used open-source models with strong performance and could also fit with our GPU limitations. Besides, we specifically choose DeepSeek-R1-Distill-Qwen-32B to investigate how reasoning models would impact the LLM-based evaluation and interval quality in both summarization and reasoning tasks.

A.9 Human-based baseline in summarization tasks

Due to lack of reference, we design a human-based baseline by Equation (2) and (3) after randomly choosing one annotation as prediction. Table 8 demonstrates that R2CCP consistently matches or outperforms the human baseline across both SummEval and DialSumm.

Dataset	Evaluator	Method	Metrics			
			Consistency	Coherence	Fluency	Relevance
SummEval	Human-based	Baseline	0.667 (91.4%)	2.000 (95.6%)	1.333 (96.3%)	2.000 (92.8%)
	GPT-4o-mini	R2CCP	0.621 (90.1%)	2.652 (89.9%)	1.135 (93.4%)	2.076 (91.5%)
	DSR1-Qwen-32B	R2CCP	0.598 (89.3%)	2.168 (85.8%)	0.850 (90.1%)	2.142 (93.3%)
	Qwen-2.5-72B	R2CCP	0.491 (88.9%)	2.429 (88.0%)	0.812 (88.0%)	1.969 (91.4%)
DialSumm	Human-based	Baseline	2.667 (95.9%)	2.000 (96.9%)	2.000 (95.1%)	2.667 (95.6%)
	GPT-4o-mini	R2CCP	1.799 (91.99%)	1.701 (91.00%)	1.215 (89.71%)	1.580 (85.2%)
	DSR1-Qwen-32B	R2CCP	1.912 (88.7%)	1.283 (89.3%)	0.812 (88.0%)	1.805 (89.9%)
	Qwen-2.5-72B	R2CCP	1.591 (87.0%)	1.494 (90.3%)	1.136 (90.3%)	1.653 (91.9%)

Table 8: Comparison of human-based baseline and R2CCP (seed = 42) on SummEval and DialSumm

A.10 In Context Learning G-Eval

Following ICE [15] and G-Eval [27], we designed the prompts by example selection with three sampling methods. For each test sample, we randomly select examples with difference source test as the exsample pool. There are 100 source in the SummEval dataset, so the sample size of example pool is $99 \times 16 = 1584$.

As for sampling method, ICE [15] has introduced uniform sampling and stratified sampling to in-context LLM evaluation on SummEval. We modified the later to quantile-based sampling to stratify bins by distribution quantiles. For example, assume we need K examples in a prompt, uniform sampling is to randomly select K samples in the example pool, stratified sampling is to stratify the range of scores into K bins and then randomly select 1 from each bin, and quantile based sampling is to stratify the distribution of scores into K quantile bins and select 1 from each bin.

After prompt design, we obtained evaluations from GLM 4-flash [9], which is free to use API. Then we calculate the correlation with the expert average. Here we present several results of different number of shots and different sampling methods, comparing with results of G-Eval and ICE. We found that ICL-G-Eval based on GLM 4-flash is equivalent to GPT3.5-Eval.

Adding more samples leads to varying effects on correlation across dimensions. For coherence and fluency, the impact is minimal or slightly negative. In contrast, consistency and relevance benefit, particularly under the quantile method. Among evaluation methods, quantile performs best in relevance, while stratified excels in other three dimensions.

A.11 Supplementary Results and Analysis

A.11.1 Continuous Intervals

In Table 10, we observe that some methods such as Boosted CQR and Boosted LCP consistently fall short of the 90 % coverage target on the DialSumm dataset, achieving only 86%–88%. In contrast, R2CCP maintains coverage in the 89%–91% range while yielding the narrowest intervals among methods with comparable performance, thus offering an optimal trade-off between coverage and efficiency. LVD achieves slightly higher coverage (around 90 %–92 %) but at the cost of wider intervals, making it suitable for scenarios that prioritize coverage over interval compactness. Both

Metric	Coherence		Consistency		Fluency		Relevance	
	ρ	τ	ρ	τ	ρ	τ	ρ	τ
G-EVAL-3.5	0.440	0.335	0.386	0.318	0.424	0.347	0.401	0.320
G-EVAL-4	**0.582**	**0.457**	**0.507**	**0.425**	**0.547**	**0.433**	**0.514**	**0.418**
ICE (Uniform Sampling)	0.476	**0.388**	**0.486**	**0.466**	0.366	0.328	0.467	0.384
ICE (Stratified Sampling)	**0.497**	0.387	0.298	0.263	**0.397**	**0.348**	**0.485**	**0.396**
ICL-G-Eval 0-shot	0.411	0.354	**0.441**	**0.417**	**0.404**	**0.384**	0.428	**0.375**
ICL-G-Eval 1-shot uniform	0.411	0.339	0.405	0.374	0.302	0.282	0.406	0.341
ICL-G-Eval 2-shot quantile	0.445	0.365	0.422	0.387	0.345	0.321	0.439	0.367
ICL-G-Eval 3-shot stratified	**0.447**	**0.367**	0.424	0.385	0.356	0.335	0.430	0.361
ICL-G-Eval 4-shot stratified	0.409	0.334	0.421	0.382	0.362	0.340	0.412	0.340
ICL-G-Eval 5-shot stratified	0.393	0.323	0.419	0.381	0.356	0.334	0.414	0.339
ICL-G-Eval 5-shot quantile	0.417	0.343	0.404	0.364	0.302	0.280	**0.448**	**0.375**

Table 9: Summary-level Spearman and Kendall-Tau correlations of different metrics on the SummEval benchmark

Asymmetric and Symmetric CQR reliably guarantee or exceed the 90 % coverage, but the cost is larger interval widths (mostly larger than 3 on ROSCOE). Across evaluators, we find that the intervals produced by the DSR1-Distill-Qwen-32B model achieve marginally higher average coverage rates. And those generated by Qwen2.5-72B-Instruct are generally shorter with lower coverage.

Method	DialSumm Evaluated with G-Eval				ROSCOE Evaluated with G-Eval			
	Consistency	Coherence	Fluency	Relevance	CosmosQA	DROP	e-SNLI	GSM8K
				GPT-4o mini				
Boosted CQR	1.85 / 86.81%	1.61 / 87.26%	1.03 / 86.33%	1.65 / 87.06%	3.12 / 77.99%	2.58 / 78.32%	2.13 / 75.79%	3.20 / 80.03%
Boosted LCP	1.83 / 87.45%	1.59 / 88.30%	1.00 / 87.53%	1.76 / 87.20%	3.45 / 79.66%	2.94 / 86.41%	1.94 / 80.26%	3.42 / 83.53%
CHR	1.54 / 80.01%	1.48 / 83.03%	0.99 / 84.01%	1.40 / 80.53%	2.47 / 70.78%	1.82 / 68.44%	1.28 / 55.66%	2.27 / 70.27%
Asym CQR	2.43 / 92.30%	1.87 / 94.00%	1.18 / 94.40%	2.09 / 92.38%	3.95 / 99.56%	3.89 / 98.19%	**2.98 / 96.67%**	3.94 / 99.27%
Sym CQR	2.41 / 91.99%	1.77 / 92.41%	**1.08 / 93.38%**	2.06 / 91.57%	**3.60 / 96.43%**	**3.77 / 96.54%**	3.35 / 95.31%	**3.58 / 94.83%**
LVD	1.90 / 89.20%	1.75 / 90.67%	1.20 / 88.40%	1.79 / 89.23%	3.18 / 83.44%	2.33 / 79.11%	3.00 / 91.18%	3.10 / 84.50%
R2CCP	**1.84 / 90.13%**	**1.63 / 90.15%**	1.14 / 89.64%	**1.72 / 90.11%**	3.09 / 86.77%	2.54 / 86.70%	2.20 / 88.01%	2.43 / 84.67%
				DeepSeek-R1-Distill-Qwen-32B				
Boosted CQR	1.89 / 87.48%	1.31 / 88.61%	1.11 / 88.07%	1.71 / 87.39%	3.40 / 80.92%	2.84 / 85.02%	2.23 / 83.68%	3.27 / 80.53%
Boosted LCP	1.88 / 86.05%	1.32 / 86.77%	1.02 / 87.28%	1.82 / 87.24%	3.49 / 81.73%	2.94 / 86.06%	1.99 / 83.33%	3.38 / 81.13%
CHR	1.76 / 86.09%	1.26 / 87.80%	1.06 / 87.79%	1.53 / 85.22%	2.64 / 78.10%	2.19 / 80.13%	1.80 / 77.24%	2.77 / 81.90%
Asym CQR	2.52 / 91.92%	1.58 / 93.20%	1.22 / 94.04%	2.42 / 92.51%	3.89 / 98.95%	3.88 / 97.78%	2.95 / 97.15%	3.90 / 99.27%
Sym CQR	2.50 / 91.30%	1.51 / 91.85%	**1.11 / 92.83%**	2.33 / 91.69%	**3.62 / 96.29%**	**3.82 / 96.22%**	3.33 / 97.85%	3.54 / 95.27%
LVD	**2.03 / 90.19%**	**1.41 / 90.29%**	1.22 / 90.41%	**1.87 / 90.01%**	3.31 / 89.52%	2.81 / 88.98%	**2.86 / 94.82%**	**3.39 / 90.07%**
R2CCP	1.86 / 89.22%	1.31 / 89.92%	1.19 / 90.57%	1.70 / 89.39%	3.05 / 86.84%	2.44 / 85.87%	1.96 / 85.43%	2.51 / 86.77%
				Qwen2.5-72B-Instruct				
Boosted CQR	1.69 / 86.57%	1.35 / 87.06%	1.05 / 87.38%	1.52 / 87.08%	3.35 / 81.02%	2.55 / 83.52%	1.90 / 81.84%	3.18 / 82.70%
Boosted LCP	1.75 / 86.08%	1.35 / 86.29%	0.96 / 87.77%	1.63 / 86.88%	3.45 / 80.41%	2.79 / 83.05%	1.85 / 80.79%	3.42 / 83.47%
CHR	1.48 / 81.50%	1.25 / 81.88%	1.00 / 85.97%	1.33 / 81.15%	2.59 / 74.97%	1.76 / 68.38%	1.39 / 66.84%	1.76 / 72.57%
Asym CQR	2.40 / 91.92%	1.61 / 92.88%	1.14 / 93.87%	2.04 / 92.38%	3.93 / 99.12%	3.92 / 99.17%	2.96 / 97.41%	3.90 / 99.00%
Sym CQR	2.37 / 91.51%	**1.52 / 91.57%**	**1.06 / 93.11%**	**2.02 / 91.55%**	**3.62 / 96.67%**	3.78 / 96.86%	3.37 / 98.29%	**3.58 / 95.30%**
LVD	**1.84 / 90.43%**	1.47 / 89.91%	1.20 / 90.26%	1.74 / 89.99%	3.34 / 89.01%	2.41 / 84.57%	**2.65 / 92.41%**	2.83 / 88.50%
R2CCP	1.74 / 89.97%	1.41 / 89.67%	1.14 / 89.70%	1.61 / 89.80%	3.07 / 87.11%	**3.10 / 93.40%**	1.68 / 84.80%	2.38 / 87.53%

Table 10: Comparison of interval width and coverage across conformal methods on DialSumm and ROSCOE tasks with G-Eval. Gray marks coverage <85%, underline marks coverage between 85%–90%, and **bold** highlights the smallest interval width among methods achieving ≥90% coverage for each evaluator–dimension. Asymmetric CQR yields the highest coverage but with wider intervals; R2CCP and LVD can offer narrower intervals while still meeting the coverage target, making them preferable when efficiency matters.

A.11.2 Discrete Intervals

Overall, with the aid of boundary adjustment, nearly all continuous-interval methods achieve average coverage rates of approximately 90% (Table 11). With coverage guarantee, the gap between BoostedCP (Boosted CQR and Boosted LCP) and R2CCP become narrower, all of which now offer similarly optimal trade-offs between coverage and interval width.

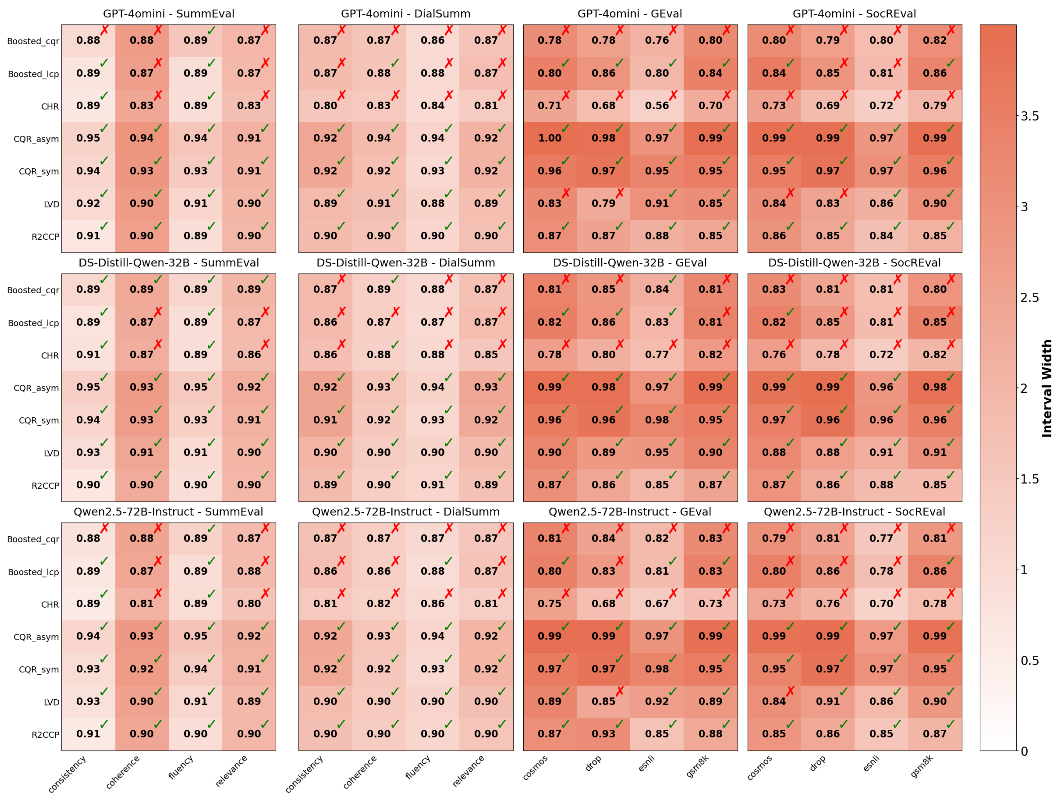

Figure 5: This is the summary and comparison on all experiments for continuous intervals. Each cell displays the mean coverage of its corresponding CP method over 30 trials on the given dataset. The cell's shading encodes the average interval width, with lighter hues denoting narrower intervals. A ✓ or ✗ in the cell's upper-right corner denotes whether the coverage criterion is met: specifically, if the mean coverage plus one standard deviation exceeds 90%, the cell is marked with ✓; otherwise, it is marked with ✗.

When comparing across evaluators, we observe heterogeneity in method performance: on GPT-4o mini, Boosted CQR and Boosted LCP typically attain the best balance, with R2CCP slightly behind; in contrast, under DSR1-Distill-Qwen-32B and Qwen2.5-72B-Instruct, the ordinal methods (OrdinalAPS, OrdinalRC), which generally underperform on GPT-4o mini, excel on DSR1. Notably, OrdinalAPS produces interval with markedly smaller intervals than those of the other methods while the coverage is around 90%.

A.11.3 With Only 0.1 Adjustment is Effective to Mitigate Miscoverage

In Section A.2.2, we introduced boundary adjustment, whereby a continuous prediction interval on a Likert scale is rounded to the nearest integer endpoints, i.e. any true label falling within half an ordinal step of a boundary is adjusted to that boundary. In practice, we observed that purely continuous intervals sometimes underperform the nominal 90% coverage target (Table 1 and 10), owing to the heteroskedastic, and correlated nature of LLM-generated judgments and to calibration set sizes that are insufficient. Full boundary adjustment reliably remedies this miscoverage by converting continuous intervals into ordinal discrete intervals, but it may introduce bias or fail to satisfy users' preference for continuous outputs. To strike a balance, we propose a partial boundary adjustment with threshold λ (e.g. $\lambda = 0.1$), meaning that only those interval endpoints within λ of an integer are rounded. For instance, [3.2,4.9] becomes [3,5] under full adjustment, but under $\lambda = 0.1$ it becomes [3.2,5], which increases coverage if the true label is 5.

As our theorem certifies, this outward adjustment effectively shifts the quantile levels to include more potential labels within the interval. Empirically, larger λ yields greater coverage gains, while

Method	DialSumm Evaluated with G-Eval				ROSCOE Evaluated with GEval			
	Consistency	Coherence	Fluency	Relevance	CosmosQA	DROP	e-SNLI	GSM8K
GPT-4o mini								
Boosted CQR	1.85 / 93.33%	1.60 / 93.95%	1.00 / 93.32%	**1.66 / 92.66%**	3.16 / 93.40%	2.60 / 90.51%	2.16 / 89.39%	**3.22 / 90.50%**
Boosted LCP	**1.83 / 92.94%**	1.60 / 93.01%	**0.96 / 93.50%**	1.76 / 91.85%	3.39 / 94.46%	2.97 / 91.71%	**1.96 / 92.32%**	3.32 / 92.87%
CHR	1.54 / 86.65%	**1.47 / 90.28%**	0.96 / 89.86%	1.40 / 87.49%	2.48 / 80.31%	1.82 / 78.06%	1.30 / 72.41%	2.25 / 78.90%
Asym CQR	2.43 / 94.34%	1.86 / 95.87%	1.18 / 95.97%	2.08 / 94.68%	3.95 / 99.69%	3.89 / 98.54%	2.99 / 97.54%	3.94 / 99.27%
Sym CQR	2.40 / 94.09%	1.76 / 94.50%	1.07 / 95.01%	2.05 / 94.33%	3.60 / 96.60%	3.77 / 96.86%	3.42 / 97.50%	3.57 / 95.10%
LVD	1.90 / 93.81%	1.75 / 94.43%	1.21 / 93.81%	1.80 / 93.77%	3.20 / 91.70%	2.33 / 86.63%	3.01 / 96.89%	3.11 / 89.53%
R2CCP	1.84 / 93.32%	1.63 / 93.38%	1.15 / 93.28%	1.72 / 93.65%	**3.06 / 90.31%**	**2.52 / 90.48%**	2.16 / 92.35%	2.42 / 86.77%
OrdinalAPS	2.24 / 90.39%	2.03 / 35.69%	1.87 / 60.61%	2.07 / 79.40%	1.79 / 70.61%	1.44 / 78.57%	1.75 / 70.13%	1.36 / 75.03%
OrdinalRC	2.33 / 91.49%	3.17 / 39.46%	2.01 / 64.66%	2.21 / 83.29%	1.94 / 73.16%	1.52 / 80.73%	1.84 / 72.32%	1.44 / 75.47%
DeepSeek-R1-Distill-Qwen-32B								
Boosted CQR	1.89 / 92.91%	1.31 / 94.36%	1.08 / 94.16%	1.71 / 92.95%	3.44 / 94.69%	2.88 / 94.41%	2.27 / 95.44%	3.29 / 93.60%
Boosted LCP	1.87 / 91.83%	1.32 / 93.31%	**0.98 / 93.27%**	1.82 / 91.52%	3.49 / 94.76%	3.03 / 91.27%	2.01 / 92.59%	3.31 / 91.70%
CHR	**1.76 / 90.78%**	**1.25 / 93.09%**	1.03 / 92.48%	**1.53 / 90.65%**	2.66 / 85.95%	2.21 / 87.94%	1.83 / 90.57%	2.75 / 89.17%
Asym CQR	2.51 / 93.90%	1.58 / 95.41%	1.22 / 95.61%	2.43 / 94.35%	3.89 / 98.95%	3.88 / 97.87%	2.94 / 97.28%	3.90 / 99.37%
Sym CQR	2.49 / 93.67%	1.51 / 94.80%	1.11 / 94.58%	2.31 / 93.54%	3.61 / 96.43%	3.83 / 96.95%	3.32 / 97.98%	3.54 / 95.80%
LVD	2.04 / 93.87%	1.41 / 95.07%	1.23 / 94.74%	1.87 / 93.86%	3.34 / 94.69%	2.82 / 93.40%	2.87 / 98.55%	3.42 / 95.47%
R2CCP	1.85 / 92.87%	1.31 / 93.84%	1.19 / 93.79%	1.70 / 93.23%	3.04 / 91.29%	2.40 / 89.84%	**1.90 / 90.79%**	2.49 / 88.87%
OrdinalAPS	2.05 / 90.05%	3.17 / 90.40%	3.43 / 90.25%	2.17 / 89.90%	**2.90 / 90.99%**	**2.27 / 91.24%**	3.20 / 91.93%	2.98 / 91.93%
OrdinalRC	2.05 / 90.04%	3.17 / 90.30%	3.42 / 89.93%	2.17 / 89.80%	2.79 / 89.59%	2.22 / 90.73%	3.15 / 90.48%	**2.86 / 90.83%**
Qwen2.5-72B-Instruct								
Boosted CQR	1.70 / 92.85%	**1.35 / 93.90%**	1.05 / 93.82%	1.52 / 92.95%	3.40 / 94.56%	2.57 / 93.30%	1.92 / 94.47%	3.22 / 92.03%
Boosted LCP	1.76 / 92.50%	**1.35 / 93.38%**	**0.90 / 92.39%**	1.62 / 92.52%	3.45 / 95.24%	2.85 / 90.95%	**1.91 / 92.02%**	3.38 / 92.73%
CHR	1.48 / 87.99%	1.25 / 89.46%	0.97 / 91.59%	1.34 / 88.06%	2.62 / 83.74%	1.77 / 82.92%	1.43 / 85.31%	1.78 / 82.10%
Asym CQR	2.41 / 95.10%	1.61 / 95.21%	1.14 / 95.08%	2.04 / 95.00%	3.93 / 99.42%	3.92 / 99.24%	2.95 / 97.50%	3.89 / 99.07%
Sym CQR	2.37 / 94.62%	1.51 / 94.42%	1.06 / 94.40%	2.05 / 94.91%	3.63 / 97.14%	3.78 / 97.14%	3.38 / 98.42%	3.58 / 95.73%
LVD	1.84 / 94.48%	1.48 / 94.77%	1.20 / 95.10%	1.73 / 94.04%	3.36 / 94.90%	**2.41 / 92.44%**	2.65 / 98.25%	2.83 / 92.47%
R2CCP	**1.73 / 93.55%**	1.41 / 93.72%	1.15 / 93.83%	1.60 / 93.17%	3.05 / 90.71%	3.08 / 95.65%	1.59 / 89.67%	2.39 / 89.60%
OrdinalAPS	2.57 / 89.84%	2.88 / 63.10%	2.95 / 75.48%	2.83 / 90.10%	2.78 / 89.42%	**2.02 / 90.95%**	2.79 / 93.25%	**2.46 / 90.30%**
OrdinalRC	2.57 / 89.85%	3.01 / 68.81%	3.09 / 78.40%	2.81 / 89.81%	**2.85 / 90.95%**	1.93 / 89.62%	2.71 / 91.40%	2.60 / 91.63%

Table 11: Comparison of narrow discrete intervals and coverage across methods on DialSumm and ROSCOE with G-Eval. Gray marks coverage $< 85\%$, underline marks coverage between 85%–90%, and **bold** highlights the smallest interval width among methods achieving $\geq 90\%$ coverage for each evaluator–dimension. Comparing with Table 10, we could find that all coverage rates improve due to boundary adjustment, while the interval widths remain comparable. For ordinal CP methods, since they directly produce discrete intervals, boundary adjustment has no influence to their intervals for the boundaries are already on the potential labels{1.00, 1.33, ..., 4.67, 5.00}

the average interval width does not increase too much and can even shrink. This is because our adjustment simultaneously cut redundant fractional parts that fail to cover the true label (e.g. [1.05,2.1] becomes [1,2], removing the excessive [2,2.1] segment). If the frequencies and sums of shrinking and expanding adjustments across intervals are approximately balanced, the average interval width remains unchanged. A sufficient condition for this result is that, within each integer bin, boundaries' fractional parts (e.g. 0.3 of 4.3, 0.7 of 1.7) are symmetrically distributed. But due to unknown distributions of model output, formally verifying this theorem remains challenging.

A.11.4 Midpoints

Table 3, 14, 15 and 16 show that midpoints are less-biased score evaluations than baselines from LLM judgments.

A.11.5 Reprompt and Regrade

Assuming LLM judge could mimic human to decide, we explore the potential of intervals in decision-making by reprompting judges with intervals information (Figure 6). We reprompt our best intervals among 30 experiments (Seed 1, 18, 9 and 30 respectively) of ROSCOE (R2CCP + DSR1) to the judge and find that intervals strengthen its confidence in initial ratings , which mostly lies within the intervals (Table 17, Figure 7, 8 and 9). And it also retains its rating even it's outside the interval (Figure 10 and 11).

Table 17 shows that there is trivial difference if we try to reprompt LLM judges to regrade. However, Figure 6, 7, 8, 9, 10, 11 show the examples of responses in reprompting.

Judge	Dataset	0.5	0.1	0
GPT-4o mini				
G-Eval	Cosmos	3.0612 ± 0.5594 / $90.31\% \pm 7.07\%$	3.0847 ± 0.5371 / $87.31\% \pm 8.17\%$	3.0864 ± 0.5335 / $86.77\% \pm 7.98\%$
	DROP	2.5230 ± 0.4804 / $90.48\% \pm 5.53\%$	2.5410 ± 0.4402 / $87.30\% \pm 6.03\%$	2.5431 ± 0.4363 / $86.70\% \pm 5.87\%$
	e-SNLI	2.1562 ± 0.4732 / $92.35\% \pm 6.88\%$	2.1932 ± 0.4282 / $88.30\% \pm 7.15\%$	2.1953 ± 0.4264 / $88.01\% \pm 7.17\%$
	GSM8K	2.4205 ± 0.7782 / $86.77\% \pm 7.63\%$	2.4283 ± 0.7639 / $85.10\% \pm 7.50\%$	2.4298 ± 0.7626 / $84.67\% \pm 8.00\%$
SocREval	Cosmos	2.9294 ± 0.4597 / $89.46\% \pm 7.01\%$	2.9586 ± 0.4396 / $86.73\% \pm 7.80\%$	2.9618 ± 0.4350 / $85.85\% \pm 7.79\%$
	DROP	2.4125 ± 0.8208 / $89.21\% \pm 9.21\%$	2.4271 ± 0.7631 / $85.40\% \pm 10.04\%$	2.4300 ± 0.7600 / $84.73\% \pm 9.97\%$
	e-SNLI	1.7076 ± 0.5804 / $90.11\% \pm 8.41\%$	1.7467 ± 0.4878 / $84.99\% \pm 8.23\%$	1.7480 ± 0.4842 / $84.02\% \pm 8.62\%$
	GSM8K	2.0943 ± 1.1782 / $86.93\% \pm 8.15\%$	2.1452 ± 1.0893 / $85.70\% \pm 7.95\%$	2.1452 ± 1.0866 / $85.07\% \pm 7.87\%$
DeepSeek-R1-Distill-Qwen-32B				
G-Eval	Cosmos	3.0357 ± 0.4873 / $91.29\% \pm 5.71\%$	3.0474 ± 0.4629 / $87.35\% \pm 6.04\%$	3.0489 ± 0.4601 / $86.84\% \pm 5.87\%$
	DROP	2.4000 ± 0.6406 / $89.84\% \pm 7.80\%$	2.4360 ± 0.5746 / $86.48\% \pm 7.94\%$	2.4385 ± 0.5714 / $85.87\% \pm 8.05\%$
	e-SNLI	1.8952 ± 0.4948 / $90.79\% \pm 7.39\%$	1.9532 ± 0.4343 / $86.06\% \pm 6.95\%$	1.9585 ± 0.4315 / $85.43\% \pm 7.02\%$
	GSM8K	2.4865 ± 0.8454 / $88.87\% \pm 9.31\%$	2.5067 ± 0.8065 / $87.07\% \pm 9.04\%$	2.5078 ± 0.8053 / $86.77\% \pm 8.89\%$
SocREval	Cosmos	2.9094 ± 0.6323 / $90.58\% \pm 7.92\%$	2.9365 ± 0.5718 / $87.76\% \pm 8.45\%$	2.9378 ± 0.5689 / $86.97\% \pm 8.37\%$
	DROP	2.2457 ± 0.6021 / $89.97\% \pm 8.61\%$	2.2906 ± 0.5369 / $86.92\% \pm 9.06\%$	2.2931 ± 0.5342 / $86.35\% \pm 9.08\%$
	e-SNLI	1.7965 ± 0.5139 / $92.35\% \pm 7.77\%$	1.8413 ± 0.4481 / $88.45\% \pm 8.00\%$	1.8450 ± 0.4443 / $87.87\% \pm 7.92\%$
	GSM8K	1.8238 ± 1.2189 / $86.93\% \pm 7.45\%$	1.8767 ± 1.1507 / $85.67\% \pm 7.30\%$	1.8796 ± 1.1480 / $85.33\% \pm 7.02\%$
Qwen2.5-72B-Instruct				
G-Eval	Cosmos	3.0529 ± 0.5262 / $90.71\% \pm 6.80\%$	3.0624 ± 0.5089 / $87.82\% \pm 8.11\%$	3.0652 ± 0.5059 / $87.11\% \pm 8.10\%$
	DROP	3.0765 ± 0.9169 / $95.65\% \pm 5.33\%$	3.0954 ± 0.8907 / $93.68\% \pm 7.50\%$	3.0964 ± 0.8894 / $93.40\% \pm 7.74\%$
	e-SNLI	1.5885 ± 0.4282 / $89.67\% \pm 6.42\%$	1.6737 ± 0.3734 / $85.28\% \pm 6.56\%$	1.6792 ± 0.3694 / $84.80\% \pm 6.89\%$
	GSM8K	2.3922 ± 0.6387 / $89.60\% \pm 4.34\%$	2.3778 ± 0.6328 / $87.93\% \pm 5.71\%$	2.3782 ± 0.6323 / $87.53\% \pm 5.95\%$
SocREval	Cosmos	2.8786 ± 0.5572 / $89.29\% \pm 7.43\%$	2.8999 ± 0.5210 / $86.16\% \pm 8.76\%$	2.8996 ± 0.5176 / $85.34\% \pm 8.46\%$
	DROP	2.3446 ± 0.6459 / $90.00\% \pm 7.90\%$	2.3832 ± 0.5883 / $86.92\% \pm 8.14\%$	2.3852 ± 0.5854 / $86.25\% \pm 8.22\%$
	e-SNLI	1.5461 ± 0.6282 / $90.20\% \pm 8.42\%$	1.5854 ± 0.5467 / $85.43\% \pm 8.44\%$	1.5897 ± 0.5397 / $84.50\% \pm 8.75\%$
	GSM8K	1.9602 ± 1.0970 / $88.57\% \pm 7.01\%$	2.0026 ± 1.0422 / $87.07\% \pm 7.14\%$	2.0015 ± 1.0391 / $86.73\% \pm 7.09\%$

Table 12: R2CCP interval width and coverage under boundary adjustments 0.5, 0.1, and 0 for three models, two judge frameworks, and four reasoning datasets (width±std / coverage%± std) based on 30 random trials: all coverages improve.

Dataset	Dimension	0.167 (Full Adjustment)	0.1	0
GPT-4o mini				
SummEval	Consistency	0.6753 ± 0.2026 / $92.15\% \pm 2.25\%$	0.6800 ± 0.1951 / $91.68\% \pm 2.33\%$	0.6858 ± 0.1859 / $90.88\% \pm 2.49\%$
	Coherence	2.6186 ± 0.1522 / $92.81\% \pm 2.37\%$	2.6201 ± 0.1497 / $91.54\% \pm 2.69\%$	2.6243 ± 0.1466 / $89.63\% \pm 3.12\%$
	Fluency	0.9116 ± 0.1673 / $90.99\% \pm 2.06\%$	0.9166 ± 0.1657 / $90.49\% \pm 2.29\%$	0.9213 ± 0.1641 / $89.36\% \pm 2.71\%$
	Relevance	1.9688 ± 0.1288 / $93.38\% \pm 1.96\%$	1.9693 ± 0.1244 / $91.90\% \pm 2.19\%$	1.9705 ± 0.1215 / $89.70\% \pm 2.50\%$
DialSumm	Consistency	1.8443 ± 0.1299 / $93.32\% \pm 1.85\%$	1.8425 ± 0.1298 / $92.03\% \pm 1.96\%$	1.8436 ± 0.1287 / $90.13\% \pm 2.39\%$
	Coherence	1.6264 ± 0.1363 / $93.38\% \pm 2.68\%$	1.6274 ± 0.1354 / $92.10\% \pm 3.08\%$	1.6256 ± 0.1337 / $90.15\% \pm 3.38\%$
	Fluency	1.1504 ± 0.1187 / $93.28\% \pm 2.03\%$	1.1484 ± 0.1237 / $91.87\% \pm 2.56\%$	1.1357 ± 0.1226 / $89.64\% \pm 2.92\%$
	Relevance	1.7161 ± 0.1398 / $93.65\% \pm 2.06\%$	1.7178 ± 0.1391 / $92.15\% \pm 2.32\%$	1.7209 ± 0.1395 / $90.11\% \pm 2.75\%$
DeepSeek-R1-Distill-Qwen-32B				
SummEval	Consistency	0.6804 ± 0.1521 / $91.57\% \pm 2.17\%$	0.6876 ± 0.1437 / $91.02\% \pm 2.11\%$	0.6941 ± 0.1343 / $90.44\% \pm 2.09\%$
	Coherence	2.2972 ± 0.1161 / $93.22\% \pm 1.65\%$	2.2994 ± 0.1169 / $91.91\% \pm 1.88\%$	2.3042 ± 0.1172 / $90.12\% \pm 2.13\%$
	Fluency	0.8886 ± 0.1605 / $91.80\% \pm 1.82\%$	0.8907 ± 0.1561 / $91.09\% \pm 2.00\%$	0.8926 ± 0.1512 / $90.09\% \pm 2.08\%$
	Relevance	1.9935 ± 0.1557 / $92.96\% \pm 2.09\%$	1.9951 ± 0.1514 / $91.72\% \pm 2.35\%$	1.9984 ± 0.1482 / $89.84\% \pm 2.90\%$
DialSumm	Consistency	1.8534 ± 0.1426 / $92.87\% \pm 2.01\%$	1.8574 ± 0.1397 / $91.43\% \pm 2.10\%$	1.8601 ± 0.1371 / $89.22\% \pm 2.62\%$
	Coherence	1.3113 ± 0.1082 / $93.84\% \pm 1.85\%$	1.3126 ± 0.1077 / $92.28\% \pm 2.14\%$	1.3138 ± 0.1068 / $89.92\% \pm 2.76\%$
	Fluency	1.1903 ± 0.1368 / $93.79\% \pm 1.78\%$	1.1915 ± 0.1349 / $92.58\% \pm 2.03\%$	1.1859 ± 0.1348 / $90.57\% \pm 2.35\%$
	Relevance	1.6952 ± 0.1660 / $93.23\% \pm 1.89\%$	1.6982 ± 0.1639 / $91.70\% \pm 2.16\%$	1.7043 ± 0.1601 / $89.39\% \pm 2.57\%$
Qwen2.5-72B-Instruct				
SummEval	Consistency	0.5876 ± 0.1520 / $91.83\% \pm 1.92\%$	0.5973 ± 0.1447 / $91.47\% \pm 1.88\%$	0.6122 ± 0.1341 / $90.73\% \pm 2.02\%$
	Coherence	2.4308 ± 0.1457 / $92.78\% \pm 2.07\%$	2.4331 ± 0.1444 / $91.53\% \pm 2.19\%$	2.4367 ± 0.1426 / $89.54\% \pm 2.48\%$
	Fluency	0.9494 ± 0.1180 / $92.12\% \pm 1.46\%$	0.9500 ± 0.1216 / $91.38\% \pm 1.76\%$	0.9527 ± 0.1218 / $90.17\% \pm 1.92\%$
	Relevance	1.9765 ± 0.1257 / $93.72\% \pm 1.78\%$	1.9776 ± 0.1253 / $92.50\% \pm 2.06\%$	1.9789 ± 0.1237 / $90.45\% \pm 2.49\%$
DialSumm	Consistency	1.7319 ± 0.1106 / $93.55\% \pm 1.59\%$	1.7350 ± 0.1083 / $92.16\% \pm 1.91\%$	1.7368 ± 0.1050 / $89.97\% \pm 2.18\%$
	Coherence	1.4060 ± 0.1115 / $93.72\% \pm 1.97\%$	1.4079 ± 0.1086 / $92.07\% \pm 2.36\%$	1.4094 ± 0.1076 / $89.67\% \pm 2.81\%$
	Fluency	1.1518 ± 0.1265 / $93.83\% \pm 2.29\%$	1.1475 ± 0.1345 / $92.37\% \pm 2.74\%$	1.1376 ± 0.1398 / $89.70\% \pm 3.42\%$
	Relevance	1.5966 ± 0.1742 / $93.17\% \pm 2.02\%$	1.6015 ± 0.1714 / $91.82\% \pm 2.43\%$	1.6071 ± 0.1682 / $89.80\% \pm 2.85\%$

Table 13: R2CCP interval width and coverage under boundary adjustments of 0.5, 0.1, and 0 for three models on SummEval and DialSumm across four dimensions (width±std / coverage%±std): all coverages improve.

Method	Coherence				Consistency				Fluency				Relevance			
	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ
GPT-4o mini																
Raw Score	3.787	1.711	0.205	0.172	1.000	0.772	0.656	0.547	2.111	1.171	0.400	0.344	1.278	0.874	0.668	0.564
Weighted Sum	3.701	1.699	0.218	0.162	0.825	0.704	0.702	0.546	1.688	1.066	0.434	0.338	1.175	0.855	0.703	0.549
Con_R2CCP	**0.344**	**0.454**	**0.396**	**0.300**	**0.391**	**0.489**	0.688	0.532	**0.173**	**0.309**	0.433	0.340	**0.338**	**0.445**	**0.716**	0.563
Dis_R2CCP	**0.348**	**0.453**	**0.385**	**0.313**	**0.395**	**0.489**	0.684	**0.553**	**0.178**	**0.309**	0.418	**0.364**	**0.342**	**0.446**	**0.714**	**0.584**
DeepSeek-R1-Distill-Qwen-32B																
Raw Score	2.908	1.412	0.396	0.329	1.422	0.952	0.589	0.497	2.454	1.383	0.414	0.356	1.214	0.829	0.555	0.461
Weighted Sum	2.149	1.241	0.456	0.343	0.652	0.614	0.642	0.491	2.115	1.287	0.452	0.347	0.674	0.625	0.621	0.476
Con_R2CCP	**0.211**	**0.348**	**0.627**	**0.488**	**0.451**	**0.509**	**0.668**	**0.512**	**0.185**	**0.315**	**0.460**	0.356	**0.348**	**0.450**	**0.721**	**0.563**
Dis_R2CCP	**0.215**	**0.347**	**0.615**	**0.511**	**0.455**	**0.508**	**0.665**	**0.534**	**0.188**	**0.314**	**0.455**	**0.389**	**0.352**	**0.450**	**0.716**	**0.581**
Qwen2.5-72B-Instruct																
Raw Score	3.934	1.775	0.321	0.267	1.344	0.897	0.704	0.599	2.796	1.420	0.478	0.406	1.812	1.070	0.609	0.521
Weighted Sum	3.693	1.746	0.358	0.266	1.076	0.819	0.737	0.577	2.575	1.335	0.499	0.386	1.552	1.014	0.660	0.516
Con_R2CCP	**0.241**	**0.381**	**0.583**	**0.450**	**0.370**	**0.467**	0.737	0.577	**0.169**	**0.306**	0.489	0.380	**0.311**	**0.424**	**0.727**	**0.578**
Dis_R2CCP	**0.245**	**0.380**	**0.574**	**0.471**	**0.373**	**0.467**	0.734	0.594	**0.174**	**0.306**	0.485	**0.419**	**0.315**	**0.424**	**0.722**	**0.594**

Table 14: Midpoints experiment on DialSumm: The midpoints substantially reduce MSE and MAE while boosting Spearman's ρ and Kendall's τ across all evaluators and dimensions, outperforming both Raw Score and Weighted Sum. **Bold** indicates better performance than baselines, underlined denotes comparable performance, and gray indicates worse performance.

Method	CosmosQA				DROP				e-SNLI				GSM8k			
	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ
GPT-4o mini																
Raw Score	1.780	1.044	0.483	0.406	1.843	0.951	0.490	0.411	2.719	1.210	0.340	0.288	2.216	0.909	0.586	0.516
Weighted Sum	1.704	1.065	0.490	0.371	1.651	0.894	0.516	0.391	2.610	1.221	0.357	0.273	2.169	0.936	0.577	0.458
Con_R2CCP	2.035	1.223	0.366	0.282	**1.509**	1.034	0.458	0.353	**1.045**	**0.865**	0.239	0.189	2.307	1.282	0.493	0.396
Dis_R2CCP	2.044	1.220	0.348	0.293	**1.526**	1.024	0.469	0.402	**1.061**	**0.854**	0.231	0.206	2.317	1.277	0.501	0.434
DeepSeek-R1-Distill-Qwen-32B																
Raw Score	2.353	1.166	0.396	0.335	2.156	0.977	0.478	0.419	3.090	1.466	0.225	0.199	2.300	0.906	0.596	0.538
Weighted Sum	1.805	1.157	0.462	0.348	1.281	0.913	0.551	0.422	2.144	1.286	0.279	0.214	1.907	1.045	0.602	0.476
Con_R2CCP	1.931	1.172	0.440	0.344	1.485	1.003	0.491	0.380	**0.904**	**0.802**	**0.423**	**0.334**	2.232	1.283	0.540	0.432
Dis_R2CCP	1.936	**0.999**	0.407	0.345	1.518	0.999	0.478	0.406	**0.916**	**0.792**	**0.405**	**0.355**	2.256	1.281	0.539	0.472
Qwen2.5-72B-Instruct																
Raw Score	1.964	1.179	0.420	0.364	1.797	0.928	0.498	0.421	1.920	1.173	0.359	0.304	1.911	0.820	0.653	0.589
Weighted Sum	1.840	1.166	0.484	0.367	1.381	0.867	0.569	0.437	1.615	1.101	0.388	0.292	1.767	0.857	0.662	0.529
Con_R2CCP	1.992	1.207	0.429	0.329	1.789	1.124	**0.584**	**0.453**	**0.796**	**0.727**	**0.471**	**0.372**	2.021	1.174	0.566	0.460
dis_R2CCP	2.004	1.200	0.390	0.327	1.801	1.121	**0.573**	**0.487**	**0.816**	**0.716**	**0.455**	**0.400**	2.044	1.173	0.578	0.511

Table 15: Midpoints experiment on ROSCOE evaluated by G-Eval. **Bold** indicates better performance than baselines, underlined denotes comparable performance, and gray indicates worse performance.

Method	CosmosQA				DROP				e-SNLI				GSM8k			
	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ	MSE	MAE	ρ	τ
GPT-4o mini																
Raw Score	1.780	1.044	0.483	0.406	2.969	1.284	0.202	0.168	1.096	0.841	0.551	0.496	4.103	1.613	0.173	0.148
Weighted Sum	1.704	1.065	0.490	0.371	1.408	0.905	0.563	0.429	1.054	0.849	0.574	0.460	1.612	0.771	0.649	0.523
Con_R2CCP	1.904	1.170	0.430	0.330	1.560	1.017	0.495	0.386	**0.725**	**0.716**	0.509	0.408	2.061	1.154	0.569	0.470
Dis_R2CCP	1.917	1.165	0.415	0.348	1.578	1.013	0.493	0.421	**0.753**	**0.711**	0.505	0.453	2.095	1.144	0.589	0.527
DeepSeek-R1-Distill-Qwen-32B																
Raw Score	2.130	1.128	0.500	0.432	1.443	0.803	0.630	0.564	0.693	0.629	0.581	0.531	1.445	0.628	0.707	0.640
Weighted Sum	2.016	1.107	0.525	0.398	1.446	0.825	0.639	0.503	0.668	0.632	0.622	0.496	1.425	0.645	0.664	0.522
Con_R2CCP	**1.853**	1.151	0.468	0.362	**1.264**	0.914	0.602	0.476	0.717	0.708	0.615	0.490	1.849	1.048	0.595	0.477
Dis_R2CCP	**1.875**	1.146	0.595	0.515	**1.290**	0.907	0.595	**0.515**	0.734	0.695	0.580	0.517	1.891	1.045	0.637	0.577
Qwen2.5-72B-Instruct																
Raw Score	1.737	0.975	0.533	0.444	1.313	0.730	0.610	0.536	0.590	0.488	0.651	0.591	1.387	0.653	0.730	0.663
Weighted Sum	1.688	0.986	0.527	0.407	1.290	0.735	0.603	0.475	0.558	0.499	0.665	0.540	1.388	0.659	0.681	0.557
Con_R2CCP	1.897	1.162	0.446	0.348	1.378	0.968	0.556	0.456	0.609	0.656	0.597	0.482	1.823	1.063	0.648	0.556
dis_R2CCP	1.910	1.156	0.442	0.375	1.403	0.965	0.541	0.487	0.632	0.652	0.595	0.533	1.849	1.057	0.653	0.596

Table 16: Midpoints experiment on ROSCOE evaluated by SocREval. **Bold** indicates better performance than baselines, underlined denotes comparable performance, and gray indicates worse performance.

Reprompt on ROSCOE by DeepSeek-R1-Distill-Qwen-32B with SocREval

Let me show you our evaluation record. Based on all these information, make decision and give me final score.

Initial Prompt:
`{{Prompt in 1st round}}`

Initial Response:
`{{Response in 1st round}}`

Reprompt and Regrade:

Thank you for your initial evaluation!

To help you arrive at a final score that more closely aligns with human expert judgment, we have constructed a 90% confidence interval for this task using conformal prediction based on your past scoring records. This interval is provided to help you gauge the uncertainty in your recent assessment, which we hope will enhance your evaluation.

Interval Information: The confidence interval we have provided is `{{Interval}}`. Please keep in mind that there is approximately a 90% probability that the expert's score lies within this interval, and a 10% probability that it lies outside.

— **Your Objective**: Acting as a human expert, use the interval information along with the recent evaluation task to decide whether and how to adjust the initial score.

— Below are some decision-making suggestions for your reference, but we also encourage you to apply your own independent thinking to align as closely as possible with human expert judgment.

Decision-Making Suggestions:

1. **Key Dimensions to Consider**:
    - **Original Score Confidence**: Your confidence level in the score you just assigned;
    - **Interval Position**: Whether the original score falls inside or outside the interval;
    - **Interval Width**: Whether the interval is narrow (e.g., ≤ 2.0) or wide (e.g., ≥ 2.0);
    - **Potential Labels**: What specific label options lie within the interval (e.g., {3.00, 4.00} for an example interval [3,4]).

2. **Advice on Decision, Reasoning, and Explanation**:
    - **High Confidence & Score Within a Narrow Interval**: If the interval is narrow and your score is validated by the interval boundaries, you may confidently retain your original score, provided you believe your evaluation and explanation are seamless. You may also make minor adjustments within the interval where you think the score is most plausible.
    - **High Confidence & Score Outside a Narrow Interval**: Although there is a small probability (<10%) that you are correct, we encourage you to question your initial judgment, reconsider the evaluation, and consider adjusting the score to the most probable point within the interval, or retain the original score with a brief justification.
    - **Low Confidence & Any Interval**: Use the interval to guide a careful re-examination of the task. For example, consider why an expert rating might take a certain value within the interval and whether that reasoning is sound. After reflection, if you find a value most reasonable, you may choose that score.
    - **Challenging the Interval**: You have the right to firmly believe that the true score cannot possibly fall within the provided interval. However, since we guarantee that the interval covers the expert score 90% of the time, your challenge likely indicates an error in expert judgment. If your explanation convinces us, this would be a valuable discovery. Generally, though, we prefer to treat the expert judgment as ground truth.

Please use the suggestions above to produce a **new final score** through a step-by-step chain of thought:
1. Your confidence level in the original score (high/medium/low) and the reason;
2. How the interval width, potential labels and the position of the original score influence your judgment;
3. Your adjustment action (retain/minor adjustment/re-examination/other) and the rationale;
4. The final score you assign.

Finally, please first state your final evaluated score (1–5), followed by your explanation:

Final Score:

Figure 6: In our reprompting, the dialogue in 1st round are fed into the 2nd-round re-evaluation, and the judge is supplied with explicit guidance on how to leverage the interval for decision-making. For example, verbalize its confidence, assessing the initial score's position relative to the interval,

Dataset	Width / Coverage	Method	MSE	MAE	ρ	τ
CosmosQA	2.60 / 89.80%	Initial Raw	2.204082	1.163265	0.480293	0.419364
		Reprompt Raw	2.193877	1.153061	0.476310	0.417798
		Initial Weighted	2.052884	1.133203	0.508314	0.390947
		Reprompt Weighted	2.111918	1.167847	0.499106	0.377264
DROP	1.67 / 89.52%	Initial Raw	1.371429	0.800000	0.603949	0.551028
		Reprompt Raw	1.380952	0.809524	0.603821	0.550921
		Initial Weighted	1.333399	0.800079	0.612075	0.485937
		Reprompt Weighted	1.345206	0.814889	0.634605	0.503989
e-SNLI	1.26 / 89.47%	Initial Raw	0.684211	0.631579	0.561363	0.517585
		Reprompt Raw	0.657895	0.631579	0.595460	0.548320
		Initial Weighted	0.610842	0.623154	0.639802	0.512257
		Reprompt Weighted	0.605095	0.638822	0.646223	0.517462
GSM8K	1.14 / 92.00%	Initial Raw	0.860000	0.420000	0.816251	0.747567
		Reprompt Raw	0.850000	0.410000	0.819313	0.755605
		Initial Weighted	0.840141	0.437920	0.763763	0.599531
		Reprompt Weighted	0.833816	0.463369	0.738347	0.590121

Table 17: Reprompting the LLM with prediction intervals reinforces its original judgments since initial scores already lie within those intervals, the model makes trivial adjustments, revealing that it might be hard for interval alone to correct inherent bias since there is no significant difference in each metric for comparison.

Reprompt Response of LLM Judge: Ground truth is 1 in this example (DROP of ROSCOE).

Final Score: 2

Explanation:

1. **Confidence Level**: High confidence in the original score of 2. The generated response incorrectly stated the number of losses, which was a clear error.
2. **Interval Consideration**: The confidence interval [1.00, 3.00] suggests that the score is likely within this range. The original score of 2 falls within this interval, aligning with the expert judgment.
3. **Adjustment Action**: Retain the original score of 2. The reasoning remains sound as the generated response had a factual error, leading to a lower quality score.
4. **Rationale**: The generated response's error in interpreting the team's record significantly impacts its quality, justifying the score of 2.

Figure 7: Example 1 of an LLM judge's reprompted scoring and rationale. In this example, its initial raw score is 2 and initial weighted score is 1.65. After reprompting, the judge more firmly believe that 2 is right since its weighted score become 2.04, which representing a shift in judgment distributions. However, the ground truth of this is 1.

Reprompt Response of LLM Judge: Ground truth is 2 in this example (GSM8K of ROSCOE).

Final Score: 2

Explanation: After reviewing the generated response and considering the confidence interval [1.00, 2.00], I remain confident that the original score of 2 is appropriate. The generated response incorrectly calculated the water intake rate and the time taken, leading to an overestimation of the water taken on. The correct approach, as detailed in my own response, results in 16 liters, whereas the generated response arrived at 160 liters. The interval suggests that the score is likely within 1 to 2, and since the response is incorrect but shows some understanding, a score of 2 is justified.

Figure 8: Example 2 of an LLM judge's reprompted scoring and rationale. In this example, its initial raw score is 2 and initial weighted score is 1.81. After reprompting, the judge more firmly believe that 2 is right since its weighted score become 2.02, which representing a shift in judgment distributions. In this example, the ground truth is 2.

Reprompt Response of LLM Judge: Ground truth is 3 in this example (CosmosQA of ROSCOE).

Final Score: 4

Explanation: 1. Confidence in Original Score: High. The initial evaluation correctly identified that Sam's understanding supports the Claim, though it could have elaborated more on the context.
2. Interval Consideration: The interval [1.00, 5.00] is wide, allowing for a range of scores. The original score of 4 falls comfortably within this interval, suggesting it is plausible.
3. Adjustment Action: Retain. The reasoning aligns well with the provided information, and the score accurately reflects the quality of the generated response.
4. Rationale: The generated response correctly answers the question but could have provided more depth. However, it sufficiently supports the Claim, warranting a score of 4.

Figure 9: Example 3 of an LLM judge's reprompted scoring and rationale. In this example, its initial raw score is 4 and initial weighted score is 3.82. After reprompting, the judge more firmly believe that 4 is right since its weighted score become 3.97, which representing a shift in judgment distributions. In this example, the ground truth is 3.

Reprompt Response of LLM Judge: Ground truth is 5 in this example (Consistency of SummEval).

Final Score: 4

Explanation:

1. **Confidence in Original Score**: High. The initial evaluation correctly assessed the summary's accuracy without factual errors.
2. **Omitted Details**: The summary fails to mention Vanessa's profession and the police appeal, which are key elements of the source text.
3. **Interval Consideration**: The confidence interval [4.67, 5.00] indicates that a higher score is possible under the uncertainty model.
4. **Adjustment Action**: Retain. Despite the interval's suggestion, the omission of critical details justifies keeping the score at 4.

Figure 10: Example of an LLM judge's reprompted scoring and rationale. This response shows that though we guarantee the help of prediction interval, the LLM judge still insist its choice thus fails to reduce the bias.

Reprompt Response of LLM Judge: Ground truth is 5 in this example (Consistency of SummEval).

Final Score: 4

Explanation:

1. **Confidence in Original Score**: High. The summary accurately reflects the main points without introducing inaccuracies, though it omits some details.
2. **Interval Consideration**: The interval [4.67, 5.00] suggests a higher score is expected. However, the summary's omission of Pereira's contract and other clubs' interest justifies a slightly lower score.
3. **Adjustment Action**: Retain the original score of 4. The summary's focus on Orlando City's interest makes the omissions acceptable, maintaining consistency without misleading information.
4. **Final Score**: 4. The summary is truthful and consistent, though less comprehensive, warranting a score within the lower end of the interval.

Figure 11: Example of an LLM judge's reprompted scoring and rationale. This response shows that though we guarantee the help of prediction interval, the LLM judge still insist its choice thus fails to reduce the bias.