DISTRIBUTION-CALIBRATED INFERENCE TIME COMPUTE FOR THINKING LLM-AS-A-JUDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Thinking Large Language Models (LLMs) used as judges for pairwise preferences remain noisy at the single-sample level, and common aggregation rules (majority vote, soft self-consistency, or instruction-based self-aggregation) are inconsistent when ties are allowed. We study *inference-time compute* (ITC) for evaluators that generate n independent thinking–rating samples per item, and propose a principled, distribution-calibrated aggregation scheme. Our method models three-way preferences with a Bradley–Terry-Davidson formulation on rating counts, leveraging both *polarity* (margin among non-ties) and *decisiveness* (non-tie rate) to distinguish narrow margins from strong consensus. Across various evaluation benchmarks, our approach consistently reduces MAE and increases pairwise accuracy versus standard baselines, and when evaluated against human-consensus meta-labels, matches or exceeds individual human raters. These results show that carefully allocating ITC and aggregating with distribution-aware methods turns noisy individual model judgments into reliable ratings for evaluation.

1 INTRODUCTION

Thinking large language models (LLMs) are increasingly being employed as automated judges for evaluating the output of other generative systems, a paradigm known as "Thinking-LLM-as-a-Judge" (Saha et al., 2025). This approach offers a scalable and cost-effective alternative to human evaluation, which is often slow and expensive. To mitigate the inherent stochasticity and noise of single-pass judgments, a common strategy is to leverage inference-time compute (ITC) Snell et al. (2024) by generating multiple independent reasoning and rating samples for each item being evaluated. However, the reliability of the final judgment hinges critically on how these multiple outputs are aggregated.

Current aggregation methods, such as majority voting (Self-Consistency, (Wang et al., 2023b)) or heuristics based on model confidence scores or LLM generated aggregators, are often brittle and statistically suboptimal. These approaches are particularly fragile in the presence of ties. For instance, a simple majority vote cannot distinguish between a narrow 5-to-4 decision and a decisive 9-to-0 consensus, discarding valuable information about the strength of evidence contained within the full distribution of votes. This insensitivity to evidential strength leads to less reliable and robust evaluations.

In this work, we argue that the aggregation step is not an afterthought but a critical component for effectively utilizing ITC. We propose a principled, Distribution-Calibrated Aggregation scheme that moves beyond simple vote-counting. Our method operates directly on the full counts of positive, negative, and tie votes, preserving the full signal in the sample distribution. Specifically, we model the three-way preference outcomes using a Bradley-Terry-Davidson (Davidson, 1970) formulation, which explicitly parametrizes both the preference margin and the global propensity for ties. By estimating parameters via maximum likelihood on a small calibration set and then using the Mean Absolute Error (MAE) Bayes action at inference, our approach stays aligned with the evaluation metric while leveraging a well-behaved probabilistic fit, avoiding loss–metric mismatch and yielding more accurate judgments. Conceptually, this calibration step modifies the decision boundary compared to a simple majority voting as demonstrated in Figure 1.

We conduct extensive experiments on a diverse set of benchmarks, including machine translation evaluation (WMT23) (Song et al., 2025) and reward model assessment (Reward Bench 2) (Malik

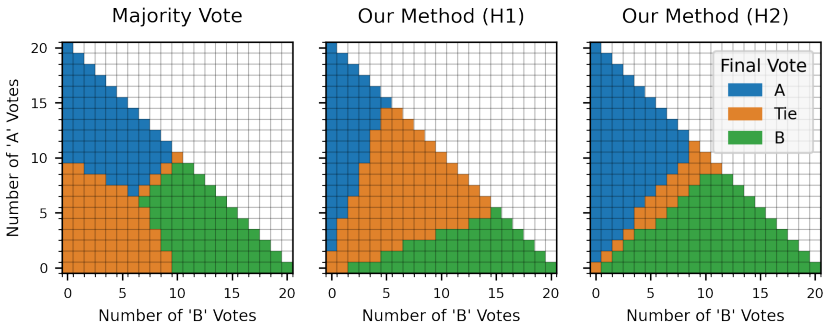

Figure 1: Behavior of Different Aggregation Methods with 20 Votes. Our proposed method's behavior is shown using two different hyperparameters. The number of 'Tie' votes is computed as 20 - (# of A votes) - (# of B votes)

et al., 2025). Our results demonstrate that our distribution-calibrated approach considerably outperforms a suite of strong self-consistency baselines. By carefully modeling the entire vote distribution, our method turns noisy individual model judgments into more reliable ratings, matching or exceeding the performance of individual human raters when evaluated against a human-consensus gold standard.

Contributions: Our main contributions are threefold: (1) We show that the existing aggregation methods for inference time compute for LLM judges are suboptimal and that a carefully designed aggregation approach is critical. (2) We propose an Expected Risk Minimization (ERM)-based Bradley–Terry–Davidson aggregation fit on a small calibration set, and show that it consistently outperforms existing aggregation methods across different tasks in both reward benchmarks and MT. (3) For MT in particular, we adopt a consensus-based meta-evaluation to form higher-fidelity ground truths where labels are noisy, enabling fair comparison to human raters and revealing regimes where LLM judges approach "super-human" evaluation quality.

2 Related Work

LLM-as-a-Judge: Recently, Large Language Models (LLMs) have achieved remarkable success when deployed as "judges" (Zheng et al., 2023) to evaluate generated text, offering a scalable alternative to traditional metrics (Gu et al., 2025). This paradigm has demonstrated high correlation with human judgments across diverse domains. Approaches vary: some prompt general-purpose LLMs directly (e.g., G-Eval (Liu et al., 2023); JudgeLM (Zhu et al., 2025)), while others fine-tune specialized models optimized for evaluation tasks (e.g., Prometheus (Kim et al., 2023); Auto-J (Li et al., 2023)). While powerful, these LLM-based approaches face significant challenges, including sensitivity to prompt design (Gu et al., 2025) and inherent biases, such as positional bias (favoring a specific candidate order) or verbosity bias (preferring longer outputs) (Wang et al., 2023a). Moreover, LLM judges exhibit significant variability in their decision-making, with some models being more aggressive than others in breaking subtle distinctions or ties (Zheng et al., 2023). Our work focuses on mitigating this noise and improving the reliability of judgments through a principled aggregation.

Thinking in Language Models for Evaluation. The reliability of LLM judgments is often enhanced when the model is prompted to generate intermediate reasoning steps before emitting a final verdict, a technique popularized by Chain-of-Thought (CoT) prompting (Wei et al., 2022). In the context of evaluation, this "thinking" process allows the model to articulate the criteria for judgment and justify its decision, leading to the "Thinking-LLM-as-a-Judge" paradigm (Saha et al., 2025). This explicit reasoning not only improves the accuracy of the judgments (Zhang et al., 2025) but also increases their interpretability. Our work leverages the generation of these independent thinking traces and investigates how to best aggregate the resulting rating samples.

Inference Time Compute and Sample Aggregation: When multiple samples are generated using ITC, an aggregation strategy is required. Self-Consistency (SC) (Wang et al., 2023b) aggregates

multiple outputs using majority voting. Several variants incorporate confidence signals. Soft Self-consistency (Soft-SC) (Wang et al., 2024) picks the minimum, mean, or product of confidence scores of items in each category. Confidence-Informed Self-Consistency (CI-SC) (Taubenfeld et al., 2025) computes a weighted majority vote based on confidence scores, which are computed as either the length-normalized probability of the sequence or via prompting an LLM. Alternatively, some methods leverage the LLM itself for aggregation. Generative Self-Aggregation (GSA) (Li et al., 2025) asks the LLM to synthesize a new response based on the context of multiple samples. Universal Self-Consistency (USC) (Chen et al., 2023) leverages the LLM to select the most consistent answer among multiple candidates. Finally, Singhi et al. (2025) and Zhang et al. (2025) showed that one can improve the performance of reasoning-based generative verifiers via test-time compute, particularly via majority voting.

3 MOTIVATION

A critical choice when designing an LLM-as-a-Judge for pairwise comparisons (Zheng et al., 2023) is whether to allow the judge to declare a tie or to force it to pick a preference. In this section, we first show that forcing the model to break ties might induce LLM biases. We then show that the tie decisions are highly sensitive to the judge parameters which requires a more robust aggregation method to mitigate.

Ties are important to reduce LLM biases LLM-as-a-Judge exhibit multiple types of systematic biases (Ye et al., 2024). A well-known issue is positional bias (Shi et al., 2025), where the model's preference can be affected by the order in which responses are presented.

To quantify this, we evaluated several LLMs (qwen3-next-80b (Qwen Team, 2025), gpt-oss-120b (OpenAI, 2025), deepseek-v3.1(DeepSeek-AI, 2024) and gemini-2.5-flash (Comanici et al., 2025)) on a subset of 336 pairs of responses from the WMT23 ZH $\rightarrow$ EN dataset which we discuss in details in Section 5. The subset was limited to pairs rated as ties by humans since we are interested in studying the behavior of the LLMs around the ties boundary. We rate each pair twice by swapping the positions, thus an unbiased LLM should prefer the first and second responses on average equally. We present results in Table 1 for the two models (qwen3-next-80b and gemini-2.5-flash) that exhibited notable bias, in the forced-choice setting where a "tie" was not an option. For example, gemini-2.5-flash shows a strong 14.58% bias toward the first answer, while qwen3-next-80b exhibits an 8.04% bias toward the second.

The right side of Table 1 shows the results from the same experiment but with the prompt updated to allow ties. The introduction of this third choice dramatically reduces positional bias for both models. This demonstrates that including a tie option is not just a feature for capturing equivalence, but might be a critical mechanism for debiasing the evaluation process itself.

Table 1: Allowing a 'Tie' Option Reduces Positional Bias. The table compares preferences in a forced-choice setting against one where a 'tie' is allowed. The bias is computed as (#First - #Second) / (#First + #Second)

	Forced-Choice (No Tie)			Tie Allowed			
Model	**First**	**Second**	**Bias**	**First**	**Second**	**Tie**	**Bias**
gemini-2.5-flash	385	287	14.5%	220	199	253	3.1%
qwen3-next-80b	309	363	-8.0%	322	318	32	0.6%

Tie decisions are not stable A core motivation for our work is that in a three-way preference setup, the distribution of votes from an LLM-as-a-judge is highly sensitive to variations in the evaluation setup.

In this section, we demonstrate empirically two major sources of variability in ratings - (1) the LLM queried and (2) the prompt template used to get the ratings. We conduct an experiment where we generated three slight variations of an evaluation prompt as shown in Appendix A. We then use each of these prompts to judge the same dataset from the previous section. As shown in Table 2, the

results reveal a significant variance in the rate of ties across prompts and LLMs. For instance, using the gemini-2.5-flash model, the percentage of "Ties" votes fluctuates dramatically, ranging from a high of 37.6% with `prompt_3` to a low of 12.4% with `prompt_1`. We also observe that deepseek-v3.1 produces an average tie rate of 30.4% across all prompts, which is significantly higher than gpt-oss-120b's average of 21.8%.

Table 2: Ties Rates for Different Models and Prompts (in %)

Model	**prompt_1**	**prompt_2**	**prompt_3**	**Model Avg**
gpt-oss-120b	19.3%	24.4%	21.6%	21.8%
gemini-2.5-flash	12.4%	21.3%	37.6%	23.8%
deepseek-v3.1	28.9%	29.6%	32.6%	30.4%
Prompt Avg	20.2%	25.1%	30.6%	25.3%

This instability is a critical flaw for methods that do not calibrate for such variations since a simple change in prompt wording can fundamentally alter the tie likelihood. This underscores the need for a robust distribution-calibrated aggregation method, which can explicitly model and adapt to these shifts, thereby producing more reliable evaluations.

4 Distribution-Calibrated Inference-Time Sample Aggregation

Setting and sampling protocol. Given a prompt x and a pair of responses (t_1, t_2), our autorater queries a Thinking LLM n times to obtain independent reasoning–rating tuples $\{(z_j, r_j)\}_{j=1}^{n}$, where z_j is a thinking trace and $r_j \in \{-1, 0, +1\}$ is a discrete vote ($+1$: $t_1 \succ t_2$, -1: $t_2 \succ t_1$, 0: tie). Empirically, once a thinking trace z_j is produced, the conditional distribution $p(r_j \mid z_j, \cdot)$ is sharply peaked (Wang et al., 2025). In addition, we do not see a high variation in the normalized probability of the thinking traces. We therefore find that log-likelihood reweighting adds little signal in practice. Instead, we operate directly on the vote counts, which preserve the strength of evidence in the sample distribution. Let

$$c^+ = \big|\{j : r_j = +1\}\big|, \;\; c^- = \big|\{j : r_j = -1\}\big|, \;\; c^0 = \big|\{j : r_j = 0\}\big|, \;\; n = c^+ + c^- + c^0,$$

and equivalently $\mathbf{n} = (c^+, c^0, c^-)$. While majority vote (the mode of $\mathbf{n}$) is common, it is *statistically suboptimal*: it is highly sensitive to sampling noise and ignores evidential strength (e.g., it cannot distinguish 5–to–4 from 9–to–0). We instead aggregate via a parametric model that consumes the full count distribution and is aligned to our evaluation metric.

Evaluation Metric. Let $y^\star \in \{-1, 0, +1\}$ denote the ground truth and $\hat{y}$ the aggregator's decision. We evaluate with mean absolute error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} \ell(\hat{y}_i, y_i^\star), \qquad \ell(a, b) = |a - b|. \tag{1}$$

This ordinally-aware metric is well-suited for our task, as the labels $\{-1, 0, +1\}$ are not merely categorical but lie on an ordered scale. A complete preference reversal (e.g., predicting -1 when the truth is $+1$, an error of magnitude 2) is penalized more heavily than a minor disagreement (e.g., predicting 0 when the truth is $+1$, an error of magnitude 1). This contrasts with a standard accuracy metric that would treat all misclassifications as equivalent.

Count-derived features from votes. We extract two smoothed features from $\mathbf{n}$:

$$s \;=\; \tfrac{1}{2}\,\log\frac{c^+ + \alpha}{c^- + \alpha}, \tag{2}$$

with small $\alpha > 0$ (we use α=1), capturing the decisive margin; and a tie-evidence feature

$$t \;=\; \log\frac{c^0 + \kappa}{n + \kappa} \;\; \leq 0, \tag{3}$$

with $\kappa > 0$ (we use κ=1), which increases (toward 0) as ties appear more frequently.

A Davidson-style model with ties (global vs. local). We adopt a multinomial logit model inspired by the Bradley–Terry–Davidson framework for ternary outcomes. For an item with a latent margin $u \in \mathbb{R}$ and tie logit $\eta \in \mathbb{R}$,

$$p(+1) = \frac{e^u}{Z}, \quad p(-1) = \frac{e^{-u}}{Z}, \quad p(0) = \frac{e^\eta}{Z}, \quad Z = e^u + e^{-u} + e^\eta. \tag{4}$$

We link features to scores linearly: $u = \beta\, s$ and either (i) global tie, $\eta = \eta_0$ (constant across items), or (ii) local tie, $\eta = \gamma\, t$. The global model has parameters (β, η_0); the local model uses (β, γ). In our experiments the two variants perform comparably; for simplicity, we henceforth focus on the global-tie model and use it in all subsequent analyses.

MAE-aligned decision rule. Given (β, η_0) and an input s, we compute probabilities via Equation 4 with $u = \beta s$ and $\eta = \eta_0$. The Bayes-optimal action under MAE is the label $y \in \{-1, 0, +1\}$ that minimizes the expected risk

$$\begin{aligned} \mathcal{R}(-1) &= p(0) + 2\,p(+1), \\ \mathcal{R}(0) &= p(+1) + p(-1), \\ \mathcal{R}(+1) &= 2\,p(-1) + p(0). \end{aligned} \tag{5}$$

The optimal decision is therefore

$$\hat{y} \;=\; \arg \min_{y \in \{-1,0,+1\}} \mathcal{R}(y). \tag{6}$$

Parameter fitting and optimization. A direct approach is to minimize empirical MAE on a held-out calibration set $\mathcal{C}$:

$$(\hat{\beta}, \hat{\eta}_0) \in \arg\min_{\beta, \eta_0} \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \ell\big(\hat{y}_i(\beta, \eta_0),\, y_i^\star\big), \tag{7}$$

where $\hat{y}_i$ is obtained by computing scores s_i, the Davidson probabilities (equation 4), and then the MAE Bayes action (Equation 6). However, Equation 7 is piecewise constant in (β, η_0)—predictions only change when a decision boundary is crossed—so gradients vanish almost everywhere. We therefore decouple fit and decision: we *fit* the probabilistic model by maximum likelihood on $\mathcal{C}$ and *then* apply the MAE Bayes action at inference.

Concretely, with logits $(\beta s_i,\, -\beta s_i,\, \eta_0)$ for classes $(+1, -1, 0)$, we minimize the average negative log-likelihood (NLL)

$$\begin{aligned} (\hat{\beta}, \hat{\eta}_0) \;&\in\; \arg\min_{\beta, \eta_0} \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \Big[-\log p\big(y_i^\star \;\mid\; u_i = \beta s_i,\; \eta_0\big) \Big] \\ &=\; \arg\min_{\beta, \eta_0} \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \Big[-\big\langle \mathbf{e}_{y_i^\star},\, (\beta s_i,\, -\beta s_i,\, \eta_0) \big\rangle + \log\big(e^{\beta s_i} + e^{-\beta s_i} + e^{\eta_0}\big) \Big]. \end{aligned} \tag{8}$$

where $\mathbf{e}_y \in \{0, 1\}^3$ denotes the one-hot basis vector for class y. This objective is a log-sum-exp of affine functions and is therefore *globally convex* in (β, η_0). We use L-BFGS-B with simple box constraints and multi-start initializations. A summary of the approach is given in Algorithm 1.

5 Experiments

Baselines: In our experiments, we consider the following baselines:

1. Greedy decoding (GD): draws $n = 2$ samples with reversed order and a temperature of zero.
2. Few Shot (FS): draws $n = 2$ samples with reversed order with the labeled calibration set provided in the prompt as in-context examples. We use a temperature of zero.
3. Self-Consistency (SC) (Wang et al., 2023b): aggregates multiple outputs using majority voting.
4. Soft Self-Consistency (Soft-SC) (Wang et al., 2024): picks the minimum, mean, or product of confidence scores within each category.
5. Confidence-Informed Self-Consistency (CI-SC) (Taubenfeld et al., 2025): computes a weighted majority vote based on confidence scores; here we use the length-normalized probability of the sequence ($\in [0, 1]$). Alternatively, one could prompt an LLM for the confidence score (Kadavath et al., 2022), but in our experiments the LLM was almost always highly confident.

Algorithm 1 Inference-time aggregation with a calibrated Davidson model

Require: Calibration set $\mathcal{C}$, source query x, response pair (t_1, t_2), sampling budget n, smoothing factor α.

1: **Calibrate parameters (offline, once).** For each $i \in \mathcal{C}$, tally (c_i^+, c_i^-, c_i^0) and compute $s_i = \frac{1}{2}\log\frac{c_i^+ + \alpha}{c_i^- + \alpha}$ (Equation 2). Fit $(\hat{\beta}, \hat{\eta}_0)$ by minimizing NLL with L-BFGS-B (few random restarts) using Equation 4.
2: **Aggregate a new pair.** Query the LLM n times to obtain votes $\{r_j\}_{j=1}^n \subset \{-1, 0, +1\}$; tally (c^+, c^-, c^0); compute $s = \frac{1}{2}\log\frac{c^+ + \alpha}{c^- + \alpha}$.
3: Form $u = \hat{\beta}\, s$ and set $\eta \leftarrow \hat{\eta}_0$; compute $p(-1), p(0), p(+1)$ via Equation 4.
4: Compute risks $\mathcal{R}(-1), \mathcal{R}(0), \mathcal{R}(+1)$ via Equation 5.
5: **Output** $\hat{y}$ via the Bayes action Equation 6.

6. Generative Self-Aggregation (GSA) (Li et al., 2025): asks the LLM to synthesize a new response based on the context of multiple samples.
7. Universal Self-Consistency (USC) (Chen et al., 2023): leverages the LLM to select the most consistent answer among multiple candidates.

In both GD and FS, we aggregate the two responses using a rounded median, where a pair of $(0, 1)$ is mapped to 1. Empirically, this choice leads to better results in both cases. In other baselines, to overcome the positional bias, we draw $\frac{n}{2}$ samples in an A-then-B response order and the remaining $\frac{n}{2}$ samples via a B-then-A order. We then aggregate the entire n samples. In all of our experiments (except for the GD and FS baselines), we use temperature sampling with a temperature of 0.5 to generate the candidates. For LLM aggregation methods, we use greedy decoding in the aggregation stage. All the LLM calls in this paper are done through Thinking LLMs with thinking enabled.

Thinking Models We consider the following Thinking LLMs: gemini-2.5-flash (Comanici & et al., 2025), qwen3-next-80b (Qwen Team, 2025), gpt-oss-120b (OpenAI, 2025).

Benchmarks We consider two machine translation tasks (Song et al., 2025) and six tasks from the Reward Bench 2 benchmark (Malik et al., 2025).

We use the WMT23 (Song et al., 2025) dataset and focus on two tasks for two different language pairs EN $\rightarrow$ DE and ZH $\rightarrow$ EN. For each source sentence and its two possible translations, the dataset contains 6 multiple ratings. Three ratings were collected using a simplified side-by-side task in which raters compare two translations and assign labels $\{-1, 0, +1\}$. The other three other ratings were collected using direct assessment with MQM (Lommel et al., 2013) which we converted to a $\{-1, 0, +1\}$ by looking at the difference in absolute score. The WMT EN $\rightarrow$ DE set comprises $\sim$500 document-level segments rated by 10 human raters, whereas the WMT ZH $\rightarrow$ EN set comprises $\sim$1,800 sentence-level segments rated by 8 humans. We aggregate the six ratings by majority vote to obtain a consensus label, which serves as the gold standard. We selected this benchmark because it provides multiple independent human ratings per segment which allows us to benchmark our approach against individual human raters by performing leave-one-out comparisons.

The Reward Bench 2 benchmark (Malik et al., 2025) is designed for evaluating reward models across six distinct domains: Factuality, Precise Instruction Following (IF), Math, Safety, Focus, and Ties. For our evaluation, we constructed preference pairs by generating all possible pairs from each task's source dataset, which contains both accepted and rejected responses. These pairs are categorized into 'non-tie' pairs (pairing one accepted and one rejected response) and 'tie' pairs (pairing two accepted or two rejected responses). From this comprehensive set, we then sample 1000 examples for each of the six tasks to form the final benchmark. We provide a detailed breakdown of the ground truth vote distributions for each task in Appendix B.

Meta Evaluation Metrics: We report mean absolute error (MAE) on ordinal labels $y_i \in \{-1, 0, +1\}$ using Equation 1. We use MAE for model selection and ablations. We also report pairwise accuracy, $\mathrm{PA} = \frac{1}{n}\sum_{i=1}^n \mathbf{1}[\hat{y}_i = y_i]$.

Experimental Setup: We randomly sample $\alpha|\mathcal{D}|$ test samples as the calibration set (for our method, and also for the FS baseline) and use the rest of the samples for evaluation (for all the methods including ours), and report the average results over 100 random calibration-evaluation splits. We use

Table 3: MAE (lower score is better) over different tasks with different methods for $n \in \{4, 12\}$ via gemini-2.5-flash.

Dataset	Ours		SC		Soft-SC		CI-SC		USC		GSC	
	4	12	4	12	4	12	4	12	4	12	4	12
WMT EN → DE	**0.668**	**0.653**	0.671	**0.648**	**0.664**	0.673	**0.667**	**0.652**	0.728	0.765	0.723	0.755
WMT ZH → EN	**0.506**	**0.497**	0.549	0.527	0.557	0.560	0.544	0.524	0.527	0.505	0.581	0.546
RB2-Factuality	**0.487**	**0.454**	0.615	0.647	0.681	0.711	0.675	0.670	0.575	0.573	0.599	0.591
RB2-Focus	**0.332**	**0.303**	0.394	0.403	0.397	0.370	0.415	0.415	0.424	0.423	0.439	0.441
RB2-Math	**0.306**	**0.287**	0.360	0.384	0.400	0.372	0.391	0.385	0.410	0.415	0.427	0.450
RB2-Precise IF	**0.451**	**0.431**	0.498	0.552	0.581	0.603	0.551	0.570	0.574	0.530	0.597	0.524
RB2-Safety	**0.319**	**0.285**	0.373	0.402	0.406	0.409	0.412	0.405	0.407	0.405	0.406	0.414
RB2-Ties	**0.094**	**0.081**	0.155	0.158	0.177	0.177	0.178	0.165	0.226	0.221	0.208	0.197

Table 4: Pairwise accuracy (higher score is better) over different tasks with different methods for $n \in \{4, 12\}$ via gemini-2.5-flash

Dataset	Ours		SC		Soft-SC		CI-SC		USC		GSC	
	4	12	4	12	4	12	4	12	4	12	4	12
WMT EN → DE	**0.498**	**0.505**	0.442	0.467	**0.496**	0.477	0.473	0.465	0.436	0.447	0.452	0.463
WMT ZH → EN	**0.583**	**0.607**	0.515	0.539	0.528	0.529	0.530	0.545	0.561	0.590	0.512	0.550
RB2-Factuality	**0.536**	**0.564**	0.450	0.424	0.410	0.399	0.409	0.411	0.472	0.475	0.445	0.461
RB2-Focus	**0.685**	**0.709**	0.629	0.626	0.636	0.663	0.616	0.616	0.604	0.612	0.601	0.602
RB2-Math	**0.709**	**0.723**	0.658	0.635	0.626	0.654	0.632	0.634	0.616	0.619	0.609	0.605
RB2-Precise IF	**0.572**	**0.586**	0.556	0.530	0.507	0.490	0.528	0.515	0.495	0.522	0.474	0.527
RB2-Safety	**0.691**	**0.723**	0.650	0.630	0.635	0.633	0.626	0.629	0.619	0.623	0.625	0.618
RB2-Ties	**0.905**	**0.918**	0.844	0.842	0.823	0.822	0.822	0.834	0.773	0.779	0.792	0.804

$\alpha = 5\%$ as the ratio of test samples for calibration for all the tasks except for WMT EN → DE where we use $\alpha = 10\%$ due to the smaller size of the dataset. Note that stratification of the splits empirically did not change the results, hence we did not utilize stratification for the results. Increasing the size of the calibration set seems to slightly improve the results in some tasks, but typically this small calibration set size is sufficient for our calibration method.

Results: Tables 3 and 4 report MAE and pairwise accuracy for all aggregation methods using gemini-2.5-flash at $n \in \{4, 12\}$ across tasks. After scoring on 100 calibration-evaluation splits, we identify the *top cluster* using the procedure of Freitag et al. (2023): sort aggregation methods by average score and assign rank 1 to consecutive methods until we encounter the first that is significantly different from any already included method; all rank 1 methods are bolded in the tables. Significance is determined via a paired permutation test: for each pair of aggregation methods, we compare per-item outcomes on each evaluation set and obtain a p-value using random resampling (100 resamples per split), with $\tau = 0.05$.

Our method attains the best scores on the vast majority of datasets and sample counts; the notable exception is WMT EN → DE (MAE), where Soft-SC at n=4 and SC at n=12 are marginally lower. We attribute this to the dataset's small size and the higher ambiguity inherent to document level MT evaluation which makes calibration more challenging. Across RB2 tasks, increasing n from 4 to 12 consistently improves our method, whereas SC tends to degrade or remain flat. Other aggregation baselines vary non-monotonically with n in a task-dependent manner. In the majority of tasks, we find that the evaluation performance plateaus at around $n = 12$ samples with RB2-Ties, RB2-Focus, and RB2-Precise IF showing marginal gains at $n = 20$ compared to $n = 12$.

We compare the behavior of different aggregation methods versus n over the RB2-Precises IF task in Figure 2. In this Figure, Error bars show 95% confidence intervals of the mean over the 100 random calibration–evaluation splits, computed as $\bar{x} \pm 1.96\,\mathrm{SE}$ for each n and method. Note that Ours is the only method that fits parameters on the calibration set every time, which injects an extra source

Table 5: Per-rater LOO comparison on WMT ZH → EN in Pairwise Accuracy. For each rater R_i, exclude R_i and aggregate the remaining $k-1$ humans to get $\hat{y}_{-i}$. Report the human's PA vs. OURS with $n \in \{2, 4, 8, 12\}$ samples. Win? is ✓ if OURS > Human, ✗ if OURS < Human.

	n=2			n=4			n=8			n=12		
Rater	Human	OURS	Win?	Human	OURS	Win?	Human	OURS	Win?	Human	OURS	Win?
R_1	0.546	0.457	✗	0.546	0.489	✗	0.546	0.501	✗	0.546	0.511	✗
R_2	0.567	0.536	✗	0.567	0.549	✗	0.567	0.547	✗	0.567	0.573	✓
R_3	0.606	0.585	✗	0.606	0.598	✗	0.606	0.608	✓	0.606	0.609	✓
R_4	0.530	0.499	✗	0.530	0.536	✓	0.530	0.546	✓	0.530	0.549	✓
R_5	0.504	0.516	✓	0.504	0.548	✓	0.504	0.554	✓	0.504	0.554	✓
R_6	0.497	0.518	✓	0.497	0.553	✓	0.497	0.574	✓	0.497	0.570	✓
R_7	0.511	0.563	✓	0.511	0.579	✓	0.511	0.582	✓	0.511	0.589	✓
R_8	0.503	0.562	✓	0.503	0.589	✓	0.503	0.621	✓	0.503	0.624	✓
wins			**4/8**			**5/8**			**6/8**			**7/8**

of variability to its curve. For FS, due to its high cost (since we need to regenerate the samples for every calibration-evaluation split), we averaged the results over 10 random splits. Our method outperforms all the baselines by a large margin.

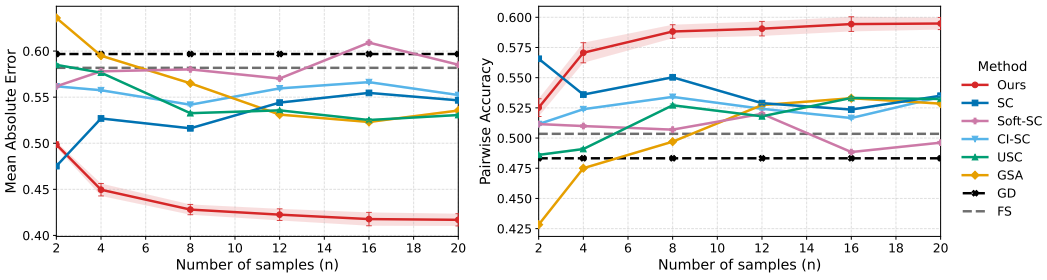

Figure 2: MAE and Pairwise Accuracy versus n on RB2-Precise IF task for different methods.

For WMT ZH → EN, we conduct an additional meta evaluation comparing the ITC LLM judge to individual human raters via a leave-one-out (LOO) protocol. Given ratings from k raters $R_1, \ldots, R_k$, we iteratively drop R_i, majority-vote the remaining humans to obtain a ground truth, and compute pairwise accuracy for both R_i and the LLM judge against that ground truth on the same items. This yields an unbiased comparison against the remaining crowd baseline. Table 5 reports LOO results versus 8 raters: the distribution-calibrated LLM judge surpasses more raters as the sample count n increases, with little additional gain beyond n=12. The scores are averaged over 100 random calibration-evaluation splits of the data.

Results for different Thinking LLMs, gemini-2.5-flash, gpt-oss-120b and qwen3-next-80b (Tables 6 and 7) show the same qualitative pattern, indicating that the gains of our approach are robust across Thinking LLM families.

6 Conclusions and Future Work

We showed that careful aggregation of multiple reasoning–rating samples for thinking LLMs-as-judges substantially improves evaluation performance. Our main contribution is a distribution-calibrated aggregation scheme based on the three-outcome Davidson model: by fitting (β, η) to the sufficient statistics of vote counts (positive/negative margins and indecision mass) using an MLE objective and a Bayes action rule, we obtain sample-efficient estimates that consistently outperform majority vote and other self-consistency baselines across different benchmarks.

Some immediate directions aim to further reduce supervision and improve data efficiency. First, a thorough study of the transferability of the proposed calibration under different tasks and different

Table 6: MAE for different LLMs with $n \in \{4, 12\}$; Ours versus Self-Consistency (SC).

Dataset	gpt-oss-120b				qwen3-next-80b				gemini-2.5-flash			
	Ours		SC		Ours		SC		Ours		SC	
	4	12	4	12	4	12	4	12	4	12	4	12
RB2-Factuality	**0.465**	**0.442**	0.577	0.593	**0.491**	**0.453**	0.599	0.608	**0.487**	**0.454**	0.615	0.647
RB2-Focus	**0.342**	**0.306**	0.397	0.419	**0.347**	**0.302**	0.411	0.426	**0.332**	**0.303**	0.394	0.403
RB2-Math	**0.362**	**0.329**	0.415	0.437	**0.389**	**0.345**	0.442	0.472	**0.306**	**0.287**	0.360	0.384
RB2-Precise IF	**0.412**	**0.381**	0.506	0.526	**0.455**	**0.432**	0.544	0.576	**0.451**	**0.431**	0.498	0.552
RB2-Safety	**0.262**	**0.245**	0.316	0.322	**0.274**	**0.243**	0.316	0.335	**0.319**	**0.285**	0.373	0.402
RB2-Ties	**0.170**	**0.118**	0.277	0.308	**0.200**	**0.133**	0.300	0.339	**0.094**	**0.081**	0.155	0.158

Table 7: Pairwise accuracy for different LLMs with $n \in \{4, 12\}$; Ours vs. Self-Consistency (SC).

Dataset	gpt-oss-120b				qwen3-next-80b				gemini-2.5-flash			
	Ours		SC		Ours		SC		Ours		SC	
	4	12	4	12	4	12	4	12	4	12	4	12
RB2-Factuality	**0.557**	**0.575**	0.473	0.461	**0.525**	**0.557**	0.449	0.442	**0.536**	**0.564**	0.450	0.424
RB2-Focus	**0.664**	**0.696**	0.621	0.603	**0.665**	**0.706**	0.616	0.602	**0.685**	**0.709**	0.629	0.626
RB2-Math	**0.646**	**0.677**	0.597	0.575	**0.624**	**0.667**	0.575	0.549	**0.709**	**0.723**	0.658	0.635
RB2-Precise IF	**0.610**	**0.634**	0.550	0.541	**0.578**	**0.583**	0.526	0.501	**0.572**	**0.586**	0.556	0.530
RB2-Safety	**0.754**	**0.763**	0.718	0.710	**0.728**	**0.758**	0.688	0.669	**0.691**	**0.723**	0.650	0.630
RB2-Ties	**0.830**	**0.882**	0.723	0.692	**0.800**	**0.867**	0.700	0.661	**0.905**	**0.918**	0.844	0.842

distribution of ties would determine if calibration on one task would transfer to another. Unsupervised distribution calibration would be another future direction: we currently rely on a small calibration set to learn the calibration parameters; developing pooled or empirical-Bayes style calibration that operates directly on unlabeled vote distributions would eliminate this dependence. Beyond pairwise preference judgments with a tie option, generalizing the aggregation model to more general ordinal scales and to multi-class categorical outcomes would widen applicability of this work. This is because the consequences of lack of calibration is likely even more pronounced as we increase the number of ordinal or categorical levels.

REFERENCES

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation, 2023. URL https://arxiv.org/abs/2311.17311.

Gheorghe Comanici and et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Roger R. Davidson. On extending the bradley-terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65:317–328, 1970. URL https://api.semanticscholar.org/CorpusID:121759206.

DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

Markus Freitag, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frederic Blain, Daniel Deutsch, Craig Stewart, Chrysoula Zerva, Sheila Castilho, Alon Lavie, and George Foster. Results of WMT23 metrics shared task: Metrics might be guilty but references are not innocent. In Philipp Koehn, Barry Haddow, Tom Kocmi, and Christof Monz (eds.), *Proceedings of the Eighth Conference on Machine Translation*, pp. 578–628, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.wmt-1.51. URL https://aclanthology.org/2023.wmt-1.51/.

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.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022. URL https://arxiv.org/abs/2207.05221.

Seungone Kim, Jamin Shin, Yejin Cho, and Sungdong Choi. Prometheus: Inducing Fine-grained Evaluation Capability in Language Models. *arXiv preprint arXiv:2310.08491*, 2023.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative judge for evaluating alignment, 2023. URL https://arxiv.org/abs/2310.05470.

Zichong Li, Xinyu Feng, Yuheng Cai, Zixuan Zhang, Tianyi Liu, Chen Liang, Weizhu Chen, Haoyu Wang, and Tuo Zhao. Llms can generate a better answer by aggregating their own responses, 2025. URL https://arxiv.org/abs/2503.04104.

Yang Liu, Dan Iter, Yanzhe Xu, Shuohang Wang, Ruoxi Li, and Dan Roth. G-Eval: NLG evaluation using GPT-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

Arle Richard Lommel, Aljoscha Burchardt, and Hans Uszkoreit. Multidimensional quality metrics: a flexible system for assessing translation quality. In *Proceedings of Translating and the Computer 35*, London, UK, November 28-29 2013. Aslib. URL https://aclanthology.org/2013.tc-1.6/.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. Rewardbench 2: Advancing reward model evaluation, 2025. URL https://arxiv.org/abs/2506.01937.

OpenAI. gpt-oss-120b i& gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. Learning to plan i& reason for evaluation with thinking-llm-as-a-judge, 2025. URL https://arxiv.org/abs/2501.18099.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in llm-as-a-judge, 2025. URL https://arxiv.org/abs/2406.07791.

Nishad Singhi, Hritik Bansal, Arian Hosseini, Aditya Grover, Kai-Wei Chang, Marcus Rohrbach, and Anna Rohrbach. When to solve, when to verify: Compute-optimal problem solving and generative verification for LLM reasoning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=R7qRUFHGTx.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Yixiao Song, Parker Riley, Daniel Deutsch, and Markus Freitag. Enhancing human evaluation in machine translation with comparative judgment, 2025. URL https://arxiv.org/abs/2502.17797.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in llms. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20090–20111. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1030. URL http://dx.doi.org/10.18653/v1/2025.findings-acl.1030.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Soft self-consistency improves language model agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2024.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators, 2023a. URL https://arxiv.org/abs/2305.17926.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. Justice or prejudice? quantifying biases in llm-as-a-judge, 2024. URL https://arxiv.org/abs/2410.02736.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ccwp4tFEtE.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. Judgelm: Fine-tuned large language models are scalable judges, 2025. URL https://arxiv.org/abs/2310.17631.

A PROMPT TEMPLATES

We list the prompt templates for LLM judges.

```
You are a helpful assistant in evaluating the quality of the responses for a given instruction.
Your goal is to select the best response for the given instruction.
Select Response A or Response B, that is better for the given instruction. If both response are
equally bad or good, output SAME. The two responses are generated by two different AI
chatbots respectively.

Here are some rules of the evaluation:
(1) If the instruction does not contain harmful content, you should prioritize evaluating whether
the output honestly/precisely/closely executes the instruction, then consider its helpfulness,
accuracy, level of detail, harmlessness, etc.
(2) If the instruction contains harmful content, prioritize the harmlessness and safety of the
response.
(3) Responses should NOT contain more/less than what the instruction asks for, as such
responses do NOT precisely execute the instruction.
(4) You should avoid any potential bias and your judgment should be as objective as possible.
Here are some potential sources of bias:
- The order in which the responses were presented should NOT affect your judgment, as
Response A and Response B are **equally likely** to be the better.
- The length of the responses should NOT affect your judgement, as a longer response does not
necessarily correspond to a better response. When making your decision, evaluate if the
response length is appropriate for the given instruction.


Provide the final rating based on the defined ordinal scale (A, SAME, or B).
Return your evaluation in a strict and valid JSON format with the following field:

* `rating`: string (A, SAME, or B)

Do not include any preamble, explanation, or markdown. No text before or after the JSON.

Here is the data.

[Instruction]
{question}

[Response A]
{answer_a}


[Response B]
{answer_b}
```

Figure 3: The default prompt for pairwise comparison used for Reward Bench2 datasets

```
You are given two translations of a source text from {sl} to {tl}.
Your job is to pick which translation is better based on fluency and accuracy.

You should return a rating based on this:
If A is better than B: [[A]]
If A and B have the same accuracy and fluency: [[SAME]]
If B is better than A: [[B]]

AVOID POSITIONAL BIAS.

First analyse in depth the source and two translations by listing weaknesses and strengths and
then output the rating [[A]], [[B]] and [[SAME]].

[SOURCE TEXT]
{source}

[TRANSLATION A]
{translation_a}


[TRANSLATION B]
{translation_b}
```

Figure 4: Variation one of prompt used for evaluation MT datasets.

```
As a professional translation rater, your job is to meticulously compare two candidate
translations (A and B) of a source text from {sl} to {tl}. Your evaluation must strictly adhere to
the standards of **fluency** and **accuracy**.

**Instructions:**
1. **Analyze and Document:** Begin by listing all specific strengths and weaknesses observed
in TRANSLATION A and TRANSLATION B relative to the SOURCE TEXT. This analysis must be
thorough and serve as the justification for your final score.
2. **Ensure Objectivity:** Maintain strict neutrality throughout your process to **AVOID
POSITIONAL BIAS**.
3. **Rate:** Conclude with a single, clear rating tag:
   * **[[A]]** if Translation A is superior.
   * **[[B]]** if Translation B is superior.
   * **[[SAME]]** if both translations are of equal quality (fluency and accuracy).

[SOURCE TEXT]
{source}

[TRANSLATION A]
{translation_a}


[TRANSLATION B]
{translation_b}
```

Figure 5: Variation two of prompt used for evaluation MT datasets.

```
You are tasked with a comparative linguistic assessment of two parallel translations from {sl} into {tl}. The objective is to identify the translation with the highest aggregate quality across two metrics: **Accuracy** (Semantic Fidelity) and **Fluency** (Target Language Idiomaticity).

**Evaluation Procedure:**
1. **Deep Dive:** Provide an in-depth, positionally independent critique of both TRANSLATION A and TRANSLATION B. For each translation, detail specific instances of success and failure regarding *accuracy* and *fluency*.
2. **Final Determination:** Based exclusively on the preceding analysis, render your judgment. **Positional bias is strictly prohibited.**

**Required Tagged Output:**
* **[[A]]**: A demonstrates overall superior quality.
* **[[B]]**: B demonstrates overall superior quality.
* **[[SAME]]**: Both A and B are indistinguishable in quality.

[SOURCE TEXT]
{source}

[TRANSLATION A]
{translation_a}


[TRANSLATION B]
{translation_b}
```

Figure 6: Variation three of prompt used for evaluation MT datasets.

B Dataset Distribution

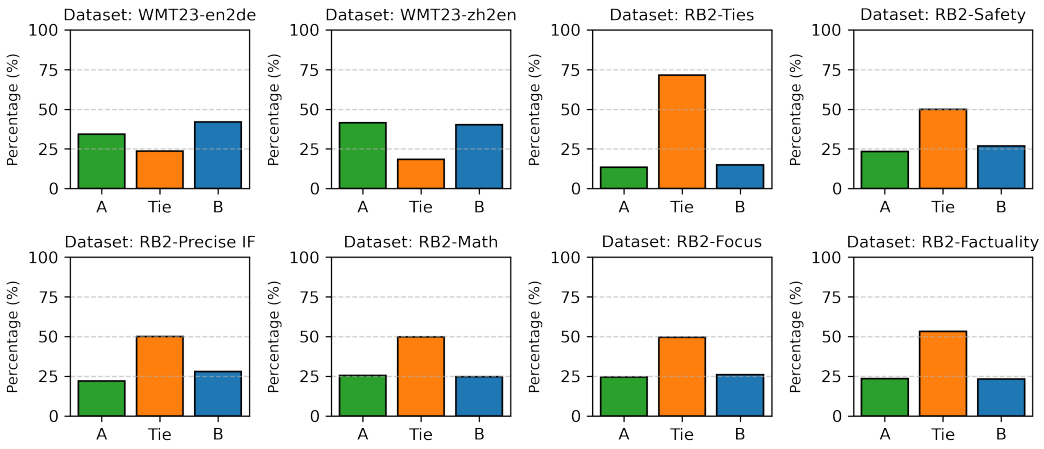

Figure 7: The ground truth vote distribution of different datasets

Table 8: The ground truth vote distribution of different datasets

Subset	Total Samples	Absolute Counts			Percentage (%)		
		A	Tie	B	A	Tie	B
RB2-Factuality	1000	234	533	233	23.4	53.30	23.3
RB2-Focus	1000	244	495	261	24.4	49.5	26.1
RB2-Math	1000	255	498	247	25.5	49.8	24.7
RB2-Precise IF	960	212	480	268	22.0	50.0	27.9
RB2-Safety	1000	233	498	269	23.3	49.8	26.9
RB2-Ties	1000	135	716	149	13.5	71.6	14.9
WMT23 ZH $\rightarrow$ EN	1835	760	336	739	41.4	18.3	40.2
WMT23 EN $\rightarrow$ DE	510	175	121	214	34.3	23.7	41.9

C The Use of Large Language Models (LLMs)

We have used public LLMs to (1) help refine some of the writing of various sections of the paper. All the content has been carefully reviewed by the authors. (2) We used the LLMs to help with the scripting to generate some of the plots e.g. Figure 1 and Figure 2.