QUANTILES, NOT POINTS: STREAMING QUANTILE MATCHING FOR ORDINAL-AWARE LLM-AS-A-JUDGE CALIBRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-as-a-judge scores are often distributionally shifted from human ratings on any given evaluation task, and post-hoc calibration (mapping LLM to human scores in a small labeled anchor set) has become a popular method to correct for this shift. We present streaming quantile matching (SQM), a version of a classical method not previously benchmarked for LLMaJ calibration, which directly computes the optimal-transport map between the LLM and human score distributions. We develop an evaluation framework and within it we show that unlike point-wise calibrations, SQM respects the ordinality of the human anchor set and achieves the best distributional fidelity and the top overall rank across models and data. We also show that in general, post-hoc calibration narrows the raw error gap between smaller and larger judge models, which opens the possibility of more efficient LLM judges.

1 INTRODUCTION

Evaluating large language models (LLMs) outputs frequently relies on human annotation as a gold standard. Collection of these scores is slow and expensive, requires recruiting and training raters, and its per-item cost compounds linearly with every new model, prompt, or use case. Often, this becomes a bottleneck on how fast and safe models can be shipped and monitored. LLM-as-a-judge (LLMaJ) (Li et al., 2024) has emerged as a scalable substitute: prompting an LLM to score model outputs. These scores, however, often disagree with human ratings in systematic ways (Morandi, 2026; Zheng et al., 2023; Wu and Aji, 2025; Wang et al., 2024a).

Approaches to address this gap include finetuning the judge model to match human ratings (Zhu et al., 2025), using reference-free scoring without human input (Fu et al., 2024), or post-hoc calibration of the judge's output (Zhao et al., 2021; Kadavath et al., 2022). We focus on the last approach, and refer to it simply as calibration, as it retains some of the gold-standard human label quality while not requiring heavy processes like model fine-tuning. Because calibration keeps the weights of the judge model, the same calibrated judge can be reused as a fixed measurement instrument for any subsequent analysis.

At its core, we start from the idea that LLMaJ scores are ordinal ratings drawn from a distribution rather than isolated points, and enforce that any reasonable calibration should respect both the ordinality of the input and the distributions of the human and LLMaJ scores. The clasical (quantile matching) Kolen and Brennan (2014) method was developed with this in mind, and we adapt it for LLMaJ calibration as streaming quantile matching (SQM). As we lack a robust testing environment for post-hoc calibration, we first build a comparison framework that allows for testing across models, methods, and data budgets (Section 3). Upon testing, we conclude that SQM achieves the top overall rank and wins most often across models, datasets, and processing modes (Section 5.2), and we attribute this to its being both distribution- and ordinality-aware (Section 5.5). We also observe that calibration narrows the error gap between smaller and larger judge models, allowing effective use of less resource-intensive judges (Section 5.4), and that the biggest factor in calibration success ends up being the specific dataset being calibrated, with the most promising axes for calibration improvement being clear task definitions and attribute-aware calibration. (Section 5.3).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2 Related Work

LLMaJ emerged as a scalable alternative to human evaluation and is widely used, employing pairwise ranking or rubric absolute scoring (Zheng et al., 2023; Liu et al., 2023). Despite this success, LLMaJ's sensitivity to prompt design, model choice, and scoring protocol raises questions about its stability as an evaluation instrument. Recent surveys structure LLMaJ research along axes such as target attributes, evaluation protocols, alignment with human judgments, and bias analysis (Gu et al., 2024; Li et al., 2024; Gao et al., 2024; Chang et al., 2024). LLMaJ's limitations are well documented: judges systematically disagree with human raters through strictness offsets and score ranges (Morandi, 2026), positional and self bias (Wang et al., 2024a; Zheng et al., 2023), and sensitivity to surface features such as verbosity and style (Zheng et al., 2023; Wu and Aji, 2025).

Three broad strategies have been proposed to close the resulting gap to human judgment. The first fine-tunes the judge model itself on human preference data, producing dedicated evaluator models such as PandaLM (Wang et al., 2024b), Prometheus (Kim et al., 2024), and JudgeLM (Zhu et al., 2025); this improves alignment but forfeits the frozen-weight property, requiring a new fine-tuning run whenever the rubric, domain, or backbone model changes. The second replaces a single scalar judgment with a reference-free scoring function derived from the judge's own likelihoods or instruction-following behavior (Fu et al., 2024), sidestepping the need for a reference answer but not the need to correct the judge's own systematic bias. Although allowing for fast iteration, this approach has limitations in accuracy and explainability outlined by Subasic et al. (2026) and Bavaresco et al. (2025), and most recently Mukherjee et al. (2026). This paper studies a strategy which leaves the judge's weights untouched and instead learns a lightweight mapping from raw judge scores to human-aligned scores (Zhao et al., 2021; Kadavath et al., 2022); unlike fine-tuning, it requires no gradient updates to the judge model, and unlike reference-free scoring, it directly targets the judge's known bias structure using a small labeled anchor set.

A growing body of work studies calibration taking into account human input and operating under frozen weights regime. Morandi (2026) show that small data budgets lead to good calibration regime. And Hong et al. (2026) report that removing the calibration step alone collapses accuracy by 60%. Li (2026) recast calibration as a multi-judge aggregation problem, and Zhu et al. (2026) asks how much labeled human data a calibrator needs before multi-judge panel is trustworthy. A larger scale model comparison presented in Inoshita et al. (2026) find that isotonic regression best matches the human label distribution. Qi et al. (2026) and Wang et al. (2025) go beyond calibrating the output and calibrate a judge's reasoning trace or full output distribution rather than its point score, and Lai et al. (2025) calibrate using cross-layer internal representations rather than the final output at all. Concurrently, Steindl et al. (2026) align judges with per-item distributions of annotator labels using trained aligners, whereas we align population-level score distributions with a closed-form, training-free map. Each of these targets a different point in the space of (judge model, calibration method, data budget); none directly compares across all three, which is the gap this paper addresses.

Stripped of its LLMaJ-specific framing, calibration under frozen weights is an instance of the more general problem of matching one distribution to another using only a limited paired sample: shifting and rescaling a model's output distribution to align with a reference is the same underlying operation performed by Platt scaling (Platt, 1999) and temperature scaling (Guo et al., 2017) for classifier confidence, by (Zhao et al., 2021) for language model output probabilities more broadly. What varies across this literature is the family of transport map allowed (affine, monotone, or fully non-parametric) ((Ma et al., 2026) compares for reward functions), the loss used to measure distributional discrepancy (mean error, KL divergence, Wasserstein distance) (Niculescu-Mizil and Caruana, 2005), and how much paired data the map is fit on (Lee et al., 2026); the methods compared in this paper span this same design space, applied uniformly to the single problem of aligning a frozen LLM judge's scores with human ratings. Unlike this line of work, which treats calibration as fitting a mapping between two sets of point scores, we start from the view that LLM-judge scores are ordinal ratings inspired by (Kuleshov et al., 2018) drawn from a distribution and design methods around respecting both. We adapt classical quantile (equipercentile) matching (Kolen and Brennan, 2014), estimated online with the P^2 estimator (Jain and Chlamtac, 1985), to LLM-judge calibration (streaming quantile matching, SQM).

3 PROBLEM SETUP

We first fix the terminology used throughout the rest of the paper, then formalize the frozen-weight calibration problem introduced in Section 1 and describe the held-out evaluation protocol and metrics used to compare methods.

Terminology. A **model** is the underlying frozen LLM judge backbone that produces a raw score for a given input; its weights are never updated anywhere in this paper. A **prompt** is the fixed instruction and scoring rubric the model is given at inference time to produce that raw score (e.g. G-Eval (Liu et al., 2023) per-attribute rubrics used for SummEval); changing the prompt changes what the model is being asked to do, but never its weights. A **method** is a calibration function: a mapping from a model's raw score(s) to a human-aligned score, fit on a small set of labeled anchors and applied afterward with no further access to the model. A **mode** is the regime under which a method is fit to its anchor set: *streaming* fits and re-fits on one labeled example at a time; *batch* fits and re-fits on fixed-size batches of B examples; and *full-batch* fits once on the entire available calibration pool in a single shot, with no incremental updates. Streaming and batch modes are re-evaluated at every checkpoint as more data arrives, while full-batch is fit and evaluated exactly once.

Frozen-weight calibration. For a fixed model m and prompt p, let $j = m_p(x) \in \{n \in \mathbb{Z} \mid a \leq n \leq b\}$ denote the raw judge score assigned to input x, and let $h(x) \in \{n \in \mathbb{Z} \mid a \leq n \leq b\}$ denote the corresponding human score. A method is a function $g_\phi : \{n \in \mathbb{Z} \mid a \leq n \leq b\} \to \mathbb{Z}$, parameterized by ϕ, that is fit on a labeled anchor set $\mathcal{A} = \{(j_i, h_i)\}_{i=1}^{n}$ so that $g_\phi(j)$ approximates h as closely as possible; at inference time g_ϕ is applied to $m_p(x)$ for new, unlabeled x, with m_p itself never updated. Following the distribution-matching view, if P_J is the distribution of raw scores j induced by m_p over a population of items and P_H is the corresponding distribution of human scores h over the same items, a method's goal is to find g_ϕ such that the pushforward distribution $g_\phi(P_J)$ is as close as possible to P_H, without access to P_H beyond the n paired anchors in $\mathcal{A}$.

Held-out evaluation protocol. For every (dataset, attribute, model) group, we reserve a fixed held-out fraction of the data (e.g. 0.3) that is never used to fit any method; the remaining pool is what full-batch mode fits on in one shot, and what streaming/batch modes replay incrementally. Every method and mode in a group is therefore scored against the exact same held-out population, at whatever amount of calibration data it has seen so far: full-batch mode is evaluated once, after its single fit on the entire non-held-out pool, while streaming/batch modes are re-scored against the held-out set at every checkpoint as they see more of the (non-held-out) data. This makes comparisons across modes and data budgets apples-to-apples: a streaming method that has seen 20% of the calibration pool and a full-batch method fit on exactly that same 20% are both scored on the identical held-out items, so differences in reported error reflect the fitting method and data budget, not which items happened to be evaluated.

Metrics. Our primary metric is **held-out mean absolute error**: the mean of $|g_\phi(j_i) - h_i|$ over the fixed held-out set, computed using whatever state a method has fit so far. This is distinct from, and preferred over, the naive **cumulative MAE**, the running average of a streaming/batch method's error on each new example *as it is seen*. Cumulative MAE is biased upward for as long as it includes a method's early, poorly-fit cold-start period, no matter how far into the stream it is measured. Held-out MAE has no such bias: it is always an independent, current-state evaluation of the method's present fit against data it has never been fit on, so it is comparable across methods, modes, and checkpoints without any correction. We additionally care about matching the *shape* of the score distribution rather than only its mean absolute deviation from human scores. For instance, whether a method reproduces human raters' full spread of scores rather than only their average. For this we report the **Earth Mover's Distance (EMD)** (Rubner et al., 2000), also known as the 1-Wasserstein distance, between the calibrated score distribution and the human score distribution on the held-out set, $W_1(\hat{P}_H, \hat{P}_{g_\phi(J)})$. We prefer EMD over Kullback–Leibler (KL) divergence (Kullback and Leibler, 1951) because it is an ordinal-aware metric: it weights each unit of probability mass by how far it must be moved along the score scale to match the target distribution, so a method that misplaces mass onto scores close to the correct one is penalized less than a method that misplaces the same mass onto scores far away. KL divergence, by contrast, treats scores as unordered categorical outcomes and penalizes any mismatch between the two distributions equally regardless of how close

Table 1: Dataset scale used for calibration. *Initial data points* is the ground-truth count of distinct underlying items before any candidate response or judge is involved. *Raw judgements* is the total number of individual judge scores our pipeline produces from a dataset.

Dataset	Initial data points	Raw judgements
BiGGen Bench	695	9,916
FLASK	168	6,429
HelpSteer2	519	17,304
SummEval	100	24,149
UPHELD	408	4,823
Total	1,890	62,621

the scores are on the underlying Likert scale. A method can achieve a low held-out MAE while still mismatching the human distribution's shape (e.g. by regressing every prediction toward the population mean), and EMD is what distinguishes that failure mode from genuine distributional alignment, while remaining sensitive to the ordinal structure that MAE alone discards at the level of aggregate shape. Finally, for clarity and understanding we report the **exact-match rate** on the held-out set, the fraction of items for which the calibrated and human score are identical.

3.1 DATASETS

We evaluate on five datasets spanning judge rubrics of differing granularity. **BiGGen Bench** (Kim et al., 2025) defines nine capabilities with instance-specific rubrics; in our pipeline these are scored under a single overall attribute using the benchmark's released absolute-scoring prompt template. **FLASK** (Ye et al., 2024) covers prompts annotated against named skill rubrics (e.g. logical robustness, factuality, commonsense understanding), using its released skill-specific judge prompts, collapsed to a single `quality` attribute in our pipeline. **HelpSteer2** (Wang et al., 2024c) provides 5 attributes (helpfulness, correctness, coherence, complexity, verbosity). HelpSteer2 ships no released LLM judge prompt; we reconstruct one from its annotator guidelines, and flag this dataset's judge-prompt source as reconstructed rather than released throughout. **SummEval** (Fabbri et al., 2021), scored via the **G-Eval** prompts (Liu et al., 2023), provides 4 attributes (coherence, consistency, fluency, relevance) on (model, summary) pairs from CNN/DailyMail articles. **UPHELD** (Subasic et al., 2026) is our fifth dataset, added to broaden coverage to multi-turn, free-form dialogue: human raters score a candidate assistant response given its preceding conversation turns. All five datasets use a five point integer Likert scale, making them directly comparable for calibration.

Table 1 reports, for each dataset, the number of *initial data points*, the ground-truth count of distinct items (instructions, source articles/conversations, or prompts) before any candidate response or judge is involved, and the total number of *raw judgements* our pipeline produces from them. The gap between the two columns comes from two multiplicative factors: (i) each initial item is paired with more than one candidate response to be scored (e.g. SummEval's 100 articles each have 16 candidate summaries, giving 1,600 pairs), and (ii) every one of those (item, response) pairs is then scored by four judge models (Section 3.2), and, for HelpSteer2 and SummEval, once per judged dimension on top of that, since each of the dimensions is an independently fit calibration target. A fraction ($\leq 2\%$) of judge calls fail to return a parseable score and are dropped rather than imputed.

3.2 MODELS

We calibrate four judge backbones, spanning both closed- and open-weight, and both larger and smaller parameter counts: **GPT-5** (OpenAI, 2025a), **Kimi K3** (Moonshot AI, 2026), a 2.8T-parameter open-weight MoE model from Moonshot AI, **GPT-OSS-20b** (OpenAI, 2025b), OpenAI's smaller open-weight model, and **Qwen3-30B** (specifically a3b-instruct-2507) (Qwen Team, 2025). All models are queried through a fixed inference gateway with weights entirely frozen throughout every experiment (we have no control over GPT-5's weights, but our experiments were run consecutively and averaged); GPT-OSS-20b and Qwen3-30B serve as the smaller judges compared against GPT-5 and Kimi K3 as the larger, frontier judges in the model-gap analysis of Section 5.

4 CALIBRATION METHODS

Table 2 reports each method we compare and their fitting mode (*Streaming* (fit and re-fit on one anchor at a time); *Batch* (fit and re-fit on fixed-size batches); *Full-batch* (fit once on the entire non-held-out pool)). Formal definitions of every method not given a formula below are in Appendix A.

4.1 SQM FOR LLMAJ CALIBRATION

Streaming quantile matching incrementally aligns the judge's empirical score quantiles to the human score quantiles, using the online P^2 quantile estimator (Jain and Chlamtac, 1985) so that neither distribution needs to be stored in full (on a K-point scale the empirical CDFs are maintained exactly with K running counts). With $\hat{F}_J, \hat{F}_H$ the empirical CDFs of the judge and human scores among anchors seen so far,

$$g_\phi(j) = \hat{F}_H^{-1}\Big(\hat{F}_J(j)\Big).$$

The point-wise methods fit g_ϕ against the paired anchors (j_i, h_i) directly.

SQM over predictive score distributions extends SQM to judges that expose a full per-example predictive distribution over ratings rather than a single collapsed score. This can be done either as explicit categorical (v, w) over the rating enum (e.g. the softmax over the score token's logits) or an unweighted sample from a rollout of repeated stochastic completions. Each example's entire distribution, not just its mean or argmax, is pooled as weighted mass into the running judge distribution, so $\hat{F}_J$ reflects the judge's predictive uncertainty at every score level and is a lower-variance estimate of the judge's marginal than an equal-count sample of point scores would give. A query's own predictive distribution is then calibrated pointwise, support point by support point,

$$g_\phi(v) = \hat{F}_H^{-1}\Big(\hat{F}_J(v)\Big) \quad \text{for each } v \in \mathrm{supp}(j),$$

with the original weights w preserved, so the returned calibrated distribution keeps the query's original shape (spread, skew, multi-modality) instead of collapsing it to a point before calibration. Taking the weighted mean of this calibrated distribution recovers a single point estimate, and a bare scalar judge score degenerates to a one-point distribution of weight 1 – so this method is a strict generalization of, and a drop-in replacement for, plain streaming quantile matching.

4.2 OTHER CDF METHODS

Mini-batch sliced Wasserstein optimal transport aligns the judge and human score distributions batch-by-batch (Bonneel et al., 2015). For a mini-batch of size B, sorting both sides to order statistics $j_{(1)} \le \cdots \le j_{(B)}$ and $h_{(1)} \le \cdots \le h_{(B)}$ gives the closed-form 1-D optimal-transport plan,

$$\{(j_{(i)}, h_{(i)})\}_{i=1}^{B} = \arg\min_{\sigma \in S_B} \sum_{i=1}^{B} \big(j_{(i)} - h_{\sigma(i)}\big)^2;$$

matched pairs from every mini-batch accumulate into a growing reference set $\mathcal{R}$, and $g_\phi(j)$ is a piecewise-linear interpolation of $\mathcal{R}$ at j.

Cross-entropy matching restricts g_ϕ to an affine map $g_\phi(j) = aj + b$ and fits (a, b) against the two marginals' histograms $p(a, b)$ and q over the score range, rather than against paired anchors,

$$(a, b) = \arg\min_{a,b} \mathrm{KL}\left(p(a, b) \,\|\, q\right).$$

SQM's nonparametric quantile-to-quantile map is the closed-form solution to the 1-D optimal-transport problem (Peyré and Cuturi, 2019) under the L_1/EMD cost – exactly the discrepancy Earth Mover's distance measures (Section 3) – while cross-entropy matching restricts the map to an affine rescaling under KL divergence, a non-ordinal-aware objective. Both restrict themselves to the two marginal distributions, but only one of them is optimizing the same discrepancy our evaluation rewards; Section 5.5 returns to this contrast and why it decides which method wins.

Table 2: Methods' MAE, Wasserstein distance, and exact-match rate, pooled across all datasets, models, and prompts. *Median* is the median calibrated value across held-out groups; *Best* is that method's single best-performing held-out group. **Bold** marks the best value in each column and underline the second best; (Δ %) is the relative improvement over the raw scores; *W/L* is the number of dataset$\times$metric combinations in which the method finished first / last by pairwise majority vote over metrics, with ties counted.

		Held-out MAE ↓		Wasserstein ↓		Exact match ↑		
Method (Reference)	Mode	Median (Δ +/-%)	Best (Δ+/- %)	Median (Δ+/- %)	Best (Δ+/- %)	Median (Δ+/- %)	Best (Δ+/- %)	Rank (W/L)
SQM methods (Our Proposal)								
Streaming quantile matching	Streaming	0.53 (+31.37)	0.19 (+75.24)	0.26 (+27.87)	**0.06 (+83.87)**	0.55 (+30.86)	**0.91 (+117.27)**	**1 (7W/0L)**
SQM over predictive score distributions	Streaming	0.53 (+30.81)	0.20 (+74.10)	**0.24 (+33.80)**	**0.06 (+82.75)**	0.55 (+30.73)	0.90 (+115.64)	6 (2W/0L)
CDF methods								
Mini-batch sliced Wasserstein OT (Bonneel et al., 2015)	Batch	0.60 (+22.45)	**0.18 (+76.29)**	0.29 (+20.52)	0.07 (+79.85)	0.55 (+30.82)	**0.91 (+117.27)**	5 (5W/0L)
Cross-entropy matching	Full-batch	0.67 (+13.36)	0.28 (+64.11)	0.32 (+11.47)	0.09 (+75.58)	0.47 (+13.49)	0.84 (+100.50)	17 (0W/3L)
Contrastive rollout calibration inspired by (Rimsky et al., 2024)	Streaming	0.65 (+16.07)	0.23 (+69.71)	0.34 (+7.71)	0.12 (+65.71)	0.48 (+15.48)	0.89 (+112.74)	15 (0W/1L)
Other methods								
Dawid–Skene EM (Dawid and Skene, 1979)	Batch	0.54 (+29.79)	0.24 (+68.87)	0.35 (+4.00)	0.08 (+77.65)	**0.57 (+35.41)**	**0.91 (+116.68)**	2 (1W/0L)
Few-shot exemplar (inspired by (Zhao et al., 2021))	Streaming	**0.52 (+32.86)**	**0.18 (+76.47)**	0.31 (+15.27)	0.09 (+76.28)	0.56 (+35.16)	0.90 (+115.52)	3 (5W/1L)
Streaming Bayesian posterior update (Bishop, 2006)	Streaming	0.55 (+28.87)	0.27 (+65.06)	0.34 (+5.97)	0.08 (+77.65)	0.56 (+33.80)	0.83 (+98.16)	4 (1W/0L)
Causal judge evaluation (Landesberg and Narayan, 2025)	Batch	0.55 (+28.25)	0.24 (+68.39)	0.34 (+6.17)	0.12 (+67.35)	0.54 (+29.94)	0.83 (+97.51)	7 (1W/0L)
Neural ODE calibration (inspired by) (Morandi, 2026)	Batch	0.55 (+29.35)	0.24 (+68.34)	0.35 (+4.58)	**0.06 (+82.37)**	0.54 (+30.38)	0.83 (+97.51)	8 (1W/0L)
Streaming linear regression (Young, 1984)	Streaming	0.55 (+28.47)	0.28 (+63.62)	0.36 (+0.82)	0.16 (+55.34)	0.56 (+34.96)	0.83 (+97.51)	9 (2W/0L)
Conformal prediction (derived from (Vovk et al., 2005))	Batch	0.56 (+28.19)	0.26 (+66.67)	0.34 (+5.69)	0.11 (+68.55)	0.56 (+33.66)	0.84 (+100.50)	10 (2W/0L)
Median scaling	Streaming	0.71 (+8.43)	0.26 (+66.67)	0.43 (-18.19)	0.08 (+77.65)	0.45 (+8.54)	0.84 (+100.50)	11 (1W/3L)
Isotonic regression (Zadrozny and Elkan, 2002)	Full-batch	0.57 (+26.17)	0.28 (+63.59)	0.40 (-8.61)	0.15 (+58.72)	**0.57 (+35.41)**	0.84 (+100.50)	12 (0W/1L)
SCOPE(Badshah et al., 2026)	Full-batch	0.57 (+26.38)	0.28 (+63.46)	0.40 (-10.13)	0.16 (+57.31)	0.54 (+30.02)	0.84 (+100.50)	13 (0W/0L)
Score-debiased KDE (SD-KDE) (Epstein et al., 2025)	Full-batch	0.57 (+26.19)	0.28 (+63.59)	0.39 (-8.23)	0.15 (+58.72)	**0.57 (+35.41)**	0.84 (+100.50)	14 (0W/1L)
Platt scaling (Platt, 1999)	Full-batch	0.62 (+20.08)	0.29 (+63.12)	0.41 (-11.27)	0.20 (+46.32)	0.55 (+30.83)	0.83 (+99.21)	16 (2W/5L)

5 RESULTS

All our experiments are run on a 30% held-out dataset, running each setup three times to account for randomness in LLM-judge scores and potential parsing errors. We structure the experiments through a set of questions we pose and answers we distill from the experimental results.

5.1 WHICH METHODS REDUCE ERROR RELATIVE TO THE RAW JUDGE?

Prior work has shown the effectiveness of calibration (Morandi, 2026; Hong et al., 2026; Li, 2026; Inoshita et al., 2026). We first confirm this holds in our setup, and Table 2 reports the improvement over the raw scores, pooled across datasets, models, and prompts. Using a paired Wilcoxon signed-rank test (Holm-corrected within each method's family of comparisons), we find a statistically significant improvement over the raw judge in 14 of the 17 methods, and the best held-out checkpoint of every method beats the raw judge on at least one metric. Appendix B contains detailed per-method and metric view. Calibration left rank correlation with human judgments unchanged: the median Spearman $\rho = 0.494$ and Kendall $\tau_b = 0.450$ are the same before and after calibration, pooled across methods and datasets. For monotone calibrators (SQM, isotonic regression, Platt scaling, affine maps) this holds by construction, since a non-decreasing map cannot reorder items. That it also holds empirically for the non-monotone methods indicates that post-hoc calibration corrects the judge's scale and distribution rather than its ranking of items. This is analogous to temperature scaling, which reduces calibration error while leaving accuracy unchanged (Guo et al., 2017).

5.2 WHICH METHODS REACH THEIR BEST SCORE FASTER?

We ask whether the SQM incremental distribution-matching methods reach a good calibration faster than fitting a method once on the full non-held-out pool. Figure 1 shows an early-stop view, the first percentile at which each method's per-dataset curve enters a $+5\%$ band around the full-batch average (0.65; best full-batch method, SCOPE, at 0.63). SQM is among the methods which cross the threshold the earliest, entering the band on SummEval at roughly the first percentile of the pool and on UPHELD by roughly the ninth. Mini-batch sliced Wasserstein OT does not visibly share this advantage: because it only updates once per fixed-size batch rather than refining a single running estimate after every anchor, its data-efficiency is structurally coarse.

In terms of actual run time, SQM falls in the top-3 methods, with only Streaming linear regression and Isotonic regression beating it at 0.7 and 0.8$\times$ the time needed for the SQM to finish. The slowest method was SD-KDE which on average took $\approx 9\times$ more than SQM. Note that all the times for a single calibrations are recorded in less than 10ms (run on a MacBook Pro M5, 48GB RAM laptop).

The predictive-distribution variant of SQM is slower, taking about $50\times$ as long as single-score SQM on average, because it must parse token probabilities.

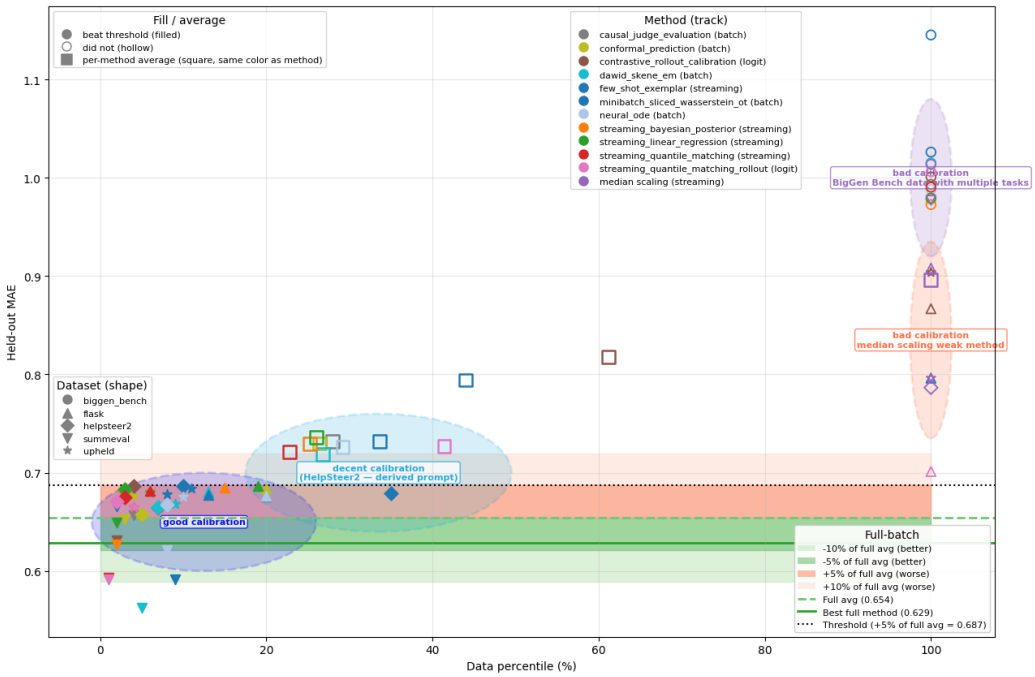


Figure 1: For each (method, dataset) pair: held-out MAE at the first point in the stream where it falls within +5% of the full-batch reference (filled = reached; hollow = never reached, plotted at 100%; squares = per-method averages). Shaded bands mark $\pm 5\%$ (darker) and $\pm 10\%$ (lighter) around the full-batch average.

A related question is whether methods need all of its data: fitting it on just the first 10/25/50/75% of the training pool shows how much held-out MAE improves with a tenfold increase in data. Figure 2 (Appendix B) reports this comparison; streaming quantile matching's own pooled trajectory (average 0.61) beats every one of these data-limited full-batch checkpoints, including the checkpoint fit on 100% of the data, using far less labeled data to get there.

A practitioner running a live streaming or batch fit does not get to see the curves in the Figures above, in advance and needs an automatic rule for when to stop collecting labels. We compared five loss-aware stopping criteria against two fixed-horizon baselines (stop after exactly 100 or 200 examples); none of the adaptive criteria beat the simple fixed horizon, which is closest to both the best checkpoint a run ever reaches and to running the full stream while consuming a fraction of the labeled pool. We report the full comparison in Appendix B.1.

5.3 Which methods are more stable across datasets?

Section 5.1's pooled numbers can hide a method that wins big on one metric or dataset and especially badly on another. Figure 1 makes this visible directly: plotting every (method, dataset) pair's early, and stopping percentile against its held-out MAE does not separate by method. Instead they form into four clusters that are mostly explained by the *dataset* a point belongs to.

The majority of (method, dataset) pairs FLASK, HelpSteer2, SummEval, and UPHELD, across essentially every method we test fall into one *good calibration* cluster, reaching the full-batch reference's $+5\%$ band within the first 0–25% of the labeled pool. A second cluster, *decent calibration (HelpSteer2 – derived prompt)*, is not made of individual points but of several methods' per-dataset *averages* sitting right at that threshold; this is a property of HelpSteer2, which ships no released LLM-judge prompt and is scored with one we reconstruct (Section 3.1), not a weakness of the methods themselves, whose individual points on the other three datasets sit well inside *good calibration*.

Table 3: Pairwise MAE gap between judge models, raw vs. average-calibrated across all methods.

Model pair	Raw MAE gap	Avg. calibrated MAE gap (% reduction)
GPT-5 – Kimi-K3 (large – large)	0.106	0.060 (43%)
GPT-OSS-20b – Qwen3-30b (small – small)	0.093	0.064 (32%)
GPT-5 – GPT-OSS-20b (large – small)	0.150	0.022 (85%)
GPT-5 – Qwen3-30b (large – small)	0.243	0.086 (65%)
Kimi-K3 – GPT-OSS-20b (large – small)	0.255	0.082 (68%)
Kimi-K3 – Qwen3-30b (large – small)	0.348	0.146 (58%)

A third cluster, *bad calibration – BigGen Bench data on multiple tasks*, contains essentially every method's BigGen Bench point, SQM's included: none of them ever reaches the threshold on this dataset, suggesting this is a property of the dataset, not of any one method's fit. Only the fourth cluster is a genuine method-level exception to this pattern, and it cuts against the same two methods in two different ways. Median-scaling's points on FLASK, HelpSteer2, and UPHELD land in a cluster of their own, *bad calibration – median scaling weak method*, well outside *good calibration*: it is the one method that fails to calibrate well even on the datasets every other method finds easy.

SQM's stability, read this way, is not that it stands apart from every other method. What sets it apart is that it never becomes an outlier in the bad direction: its one weak dataset, BigGen Bench, fails exactly as much as the dataset forces every method to fail, never more, unlike few-shot exemplar's catastrophic BigGen Bench point or median scaling's separate cluster of failures on all datasets.

5.4 Which models are calibrated better?

Previously (Lail and Markham, 2026) have shown that "smaller" judges can achieve "larger" judges equivalence by ensemble, and we are interested in whether the same can be achieved by calibration. We compare four judge models, two "large" (GPT-5, Kimi-K3) and two "small" (GPT-OSS-20b, Qwen3-30b). Table 3 shows that the reduction of MAE is not uniform. Raw MAE separates the four models by a wide margin (0.69 for Kimi-K3; 1.04 for Qwen3-30b), and the two small models have the highest raw MAE. Averaged across all methods calibration shrinks cross-tier (small-vs-large) gaps 58–85%, roughly double the 32–43% reduction seen within each tier. Most extreme we notice that GPT-OSS-20b's average-calibrated MAE (0.666) ends up only 0.022 above GPT-5's (0.644).

Whether this gap closure is uniform, or driven by a handful of standout methods, is a fair question given Section 5.3's unstable calibration across datasets. Using each small model's best calibration method rather than the tier average, GPT-OSS-20b closes 97.4% of its raw gap to the best large model (Kimi-K3) and Qwen3-30b closes 76.3% of its gap – versus 67.8% and 58.1% respectively under the tier-average method mix. Under the worst calibration method for each (median scaling for both), the picture reverses: GPT-OSS-20b's raw MAE barely moves (a 1.7% reduction) and its gap to Kimi-K3's typical calibrated performance actually *widens* by about 36% relative to the raw gap; Qwen3-30b's worst method similarly leaves it essentially uncalibrated (1.6% raw MAE reduction) and about 26% further from Kimi-K3 than the raw gap. This suggests that method, and not model, choice is what determines the similarity of small and big models calibrated judgments.

5.5 Why does streaming quantile matching win?

SQM ranks first overall and its two variants achieve the lowest median Wasserstein distances of all methods (0.24 for the predictive-distribution variant and 0.26 for single-score SQM, vs. 0.29–0.43 for all others; Table 2). Direct-fit streaming baselines constrain the correction g_ϕ to fixed parametric families (e.g., affine maps, quadratic features, or local exemplars). In contrast, SQM's nonparametric quantile-to-quantile map directly aligns empirical CDFs—the exact closed-form optimal-transport solution under the L_1/EMD objective used to measure distributional success. This distributional fit also carries over to pointwise metrics: SQM is within 0.01 of the best median MAE and 0.02 of the best median exact-match rate. We hypothesize that matching the full score distribution, rather than fitting a fixed parametric form, makes SQM less sensitive to rubric differences across tasks.

Extending this nonparametric mapping to full example score distributions rather than a single collapsed value (Section 4.1) yields the lowest median EMD (0.24), while the single-score variant is

within 0.01 of the best median MAE. Both variants leading on distributional fidelity confirms that the quantile-matching mechanism drives performance, not the output representation.

Other distributional methods fall short due to specific structural limitations:

- **Cross-Entropy Matching:** Although it targets score distributions and improves EMD over full-batch baselines (Section 5.1), it trails SQM significantly (median MAE 0.67 vs. 0.53; Wasserstein 0.32 vs. 0.26). Its KL objective is distance-blind. It penalizes misplaced probability mass regardless of bin distance and its affine restriction prevents true reshaping.
- **Mini-Batch Sliced Wasserstein OT:** While solving 1-D optimal transport, it operates only within isolated, fixed-size mini-batches rather than maintaining a continuously refined global estimate, leading to middling MAE and no early-stopping benefit.

Comparing the baselines reveals two axes governing performance: ordinal awareness and distributional alignment. Pointwise accuracy requires ordinal awareness ("ordinal-aware beats ordinal-blind"), as shown by isotonic regression beating cross-entropy matching on MAE and exact match despite ignoring score distributions entirely. Conversely, explicit distributional alignment is a direct route to distributional shape fidelity, as the three lowest Wasserstein distances all belong to distributional methods. SQM's advantage holds at a matched data budget: its streaming trajectory at the 100% checkpoint, having seen the same data both baselines are fit on in full-batch mode, still leads both (Figure 2). This rules out data budget as a primary confounder. We note this functions as a quasi-ablation at the property level rather than model internals, disentangling the impact of loss function objectives (e.g., L_1/EMD vs. isotonic regression's L_2 loss) and modeling capacity (e.g., nonparametric flexibility vs. cross-entropy matching's affine restriction).

6 Limitations, Future Work, and Conclusions

Data scale and language. All datasets use a 5-point Likert scale; human alignment can vary on coarser or finer scales (Li et al., 2026). HelpSteer2's native 0–4 scale is mapped to 1–5 via a standard offset. All prompts and data are English-only due to resource constraints, though the core calibration benefit should be language-agnostic.

Judge models. We hold every model's weights frozen and vary only the post-hoc calibration function (Section 3); how the judge was trained, prompted, or fine-tuned, all known to affect judge-human alignment (Zhu et al., 2025; Kim et al., 2024; Wang et al., 2024b; Xia et al., 2025), is out of scope. Whether a better trained or prompted judge needs less calibration data is left to future work. Two frontier-scale (GPT-5, Kimi-K3) and two smaller judges (GPT-oss-20b, Qwen3-30b) are enough to check that the frontier-vs-small gap and calibration's effect on it (Section 5.4) are not an artifact of one model, but too few to characterize which model families are more calibration-friendly.

Regardless of limitations, the results presented make the case for SQM as the most appropriate calibration method tested. We showed that distributionally-aware SQM is best on distribution shape, competitive on per-item accuracy all across models and data budgets, and allows for robust judge calibration within a fixed horizon of 100–200 human labeled examples. Our work thus shows that robust LLMaJ performance can be boosted by a modest set of human-labeled data, which can allow for more confident scaling of LLM applications. Beyond SQM specifically, this paper set out to answer, under one shared experimental protocol, how much human-labeled data a frozen LLM judge needs to be calibrated well, and whether calibration lets smaller judges compete with larger ones.

Reproducibility Statement. All resources sufficient to reproduce every result reported are included as anonymized supplementary material. We withhold public release until after review to preserve anonymity, and will release the repository after review period has finished. Exact reproduction of raw judge scores may not be possible for closed or updated endpoints the calibration methods themselves are fully deterministic given any fixed anchor set.

Ethical Statement. While we recognize that LLMaJs risk reducing traditional data annotation roles, calibration techniques enable reliable human-in-the-loop workflows that preserve essential human oversight while mitigating sudden displacement. By keeping human expertise central to quality control, these approaches help transform annotation jobs rather than eliminate them outright.

AI USE DISCLOSURE

In this work, we used generative AI tools for:

- Polishing the writing, fixing LaTeX formatting, correcting typos, improving unclear or ambiguous sentences, and formatting tables, figures, and equations.
- Drafting select sections based on initial human outlines and concepts, followed by iterative human review, correction, and refinement.
- Implementing certain calibration methods using AI coding assistants (Claude Code and Cursor), followed by thorough human code review and test-case verification.
- Help with literature review searches (paired with standard search engines) to identify related works within the rapidly growing body of literature.
- Data analysis and chart generation assistance using Cursor and Google NotebookLM.

We did not use generative AI tools for:

- Conceptualizing or designing the research ideas and experimental frameworks.
- Autonomous agentic workflows where AI generated content from broad goals without direct human oversight.
- Generating synthetic data (all datasets used are publicly available).
- Fully autonomous section writing. No text was generated without initial human input and extensive subsequent editing of AI suggestions or edits.

We take full responsibility for the content of this work, including all text, claims, and artifacts produced with the aid of generative AI tools.

REFERENCES

Sher Badshah, Ali Emami, and Hassan Sajjad. SCOPE: Selective conformal optimized pairwise LLM judging. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2602.13110.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-252-7. doi: 10.18653/v1/2025.acl-short.20. URL https://aclanthology.org/2025.acl-short.20/.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):Article 39, 2024.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

Elliot L. Epstein, Rajat Dwaraknath, Thanawat Sornwanee, John Winnicki, and Jerry Weihong Liu. SD-KDE: Score-debiased kernel density estimation, 2025. URL https://arxiv.org/abs/2504.19084. ICLR 2025 Workshop on Frontiers in Probabilistic Inference.

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

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2024. URL https://arxiv.org/abs/2302.04166.

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.

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, 2024. URL https://arxiv.org/abs/2411.15594.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330, 2017.

Yihan Hong, Huaiyuan Yao, Bolin Shen, Wanpeng Xu, Hua Wei, and Yushun Dong. From rubrics to reliable scores: Evidence-grounded text evaluation with LLM judges, 2026. URL https://arxiv.org/abs/2601.08654.

Keito Inoshita, Xiaokang Zhou, Akira Kawai, and Katsutoshi Yada. LLMs capture emotion labels, not emotion uncertainty: Distributional analysis and calibration of human-LLM judgment gaps, 2026. URL https://arxiv.org/abs/2604.27345.

Raj Jain and Imrich Chlamtac. The P^2 algorithm for dynamic calculation of quantiles and histograms without storing observations. *Communications of the ACM*, 28(10):1076–1085, 1985.

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, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.08491.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee, Namgyu Ho, Se June Joo, Miyoung Ko, Yoonjoo Lee, Hyungjoo Chae, Jamin Shin, Joel Jang, Seonghyeon Ye, Bill Yuchen Lin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. The BiGGen bench: A principled benchmark for fine-grained evaluation of language models with language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5877–5919, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.303. URL https://aclanthology.org/2025.naacl-long.303/.

Michael J. Kolen and Robert L. Brennan. *Test Equating, Scaling, and Linking: Methods and Practices*. Springer, New York, 3rd edition, 2014. ISBN 978-1-4939-0316-0. doi: 10.1007/978-1-4939-0317-7.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pages 2796–2804, 2018.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Peng Lai, Jianjie Zheng, Sijie Cheng, Yun Chen, Peng Li, Yang Liu, and Guanhua Chen. Beyond the surface: Enhancing LLM-as-a-judge alignment with human via internal representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2508.03550.

Ryan Lail and Luke Markham. On cost-effective LLM-as-a-judge improvement techniques, 2026. URL https://arxiv.org/abs/2604.13717.

Eddie Landesberg and Manjari Narayan. Causal judge evaluation: Calibrated surrogate metrics for LLM systems, 2025. URL https://arxiv.org/abs/2512.11150.

Junghyun Lee, Sanghwa Kim, Yassir Jedra, Alexandre Proutiere, and Se-Young Yun. Instance-optimal estimation with multiple LLM judges on a budget, 2026. URL https://arxiv.org/abs/2605.23362. ICML 2026 Workshop on Combining Theory and Benchmarks.

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.

Weiyue Li, Minda Zhao, Weixuan Dong, Jiahui Cai, Yuze Wei, Michael Pocress, Yi Li, Wanyan Yuan, Xiaoyue Wang, Ruoyu Hou, et al. Grading scale impact on LLM-as-a-judge: Human-LLM alignment is highest on 0–5 grading scale, 2026. URL https://arxiv.org/abs/2601.03444.

Yanran Li. Calibrate, don't curate: Label-efficient estimation from noisy LLM judges, 2026. URL https://arxiv.org/abs/2605.09702.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Rachel Ma, Dylan Hadfield-Menell, and Kristjan Greenewald. Distributional process reward models: Calibrated prediction of future rewards via conditional optimal transport, 2026. URL https://arxiv.org/abs/2605.06785.

Moonshot AI. Kimi K3: Open frontier intelligence. Technical report, Moonshot AI, 2026. URL https://github.com/MoonshotAI/Kimi-K3/blob/main/k3_tech_report.pdf.

Andrea Morandi. Two ways to de-bias an LLM-as-a-judge: A continuous-score comparison of hierarchical Bayesian calibration and neural-ODE score transport, 2026. URL https://arxiv.org/abs/2605.09227.

Sourabrata Mukherjee, H. Hamna, Kalika Bali, and Sunayana Sitaram. The geometry of llm-as-judge: Why inter-llm consensus is not human alignment. *arXiv preprint arXiv:2606.03043*, 2026. URL https://arxiv.org/abs/2606.03043.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 625–632, 2005.

OpenAI. GPT-5 system card. Technical report, OpenAI, 2025a. URL https://cdn.openai.com/gpt-5-system-card.pdf.

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

Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport: With Applications to Data Science*. Foundations and Trends in Machine Learning. Now Publishers, 2019. ISBN 9781680835502.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Alexander J. Smola, Peter Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

Jasmine Qi, Danylo Dantsev, and Muyang Sun. VERDI: Single-call confidence estimation for verification-based LLM judges via decomposed inference, 2026. URL https://arxiv.org/abs/2605.11334.

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

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 15504–15522, 2024. URL https://aclanthology.org/2024.acl-long.828/.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

Sebastian Steindl, Nikos Voskarides, Alberto Gasparin, and Diego Marcheggiani. Post-hoc alignment of LLM-judges to human judgment distribution, 2026. URL https://arxiv.org/abs/2609.01073. To appear at EMNLP 2026.

Ilija Subasic, Andrew Rabinovich, and Zhao Chen. Evaluating language models in realistic conversational contexts. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=5kv79PBBy3.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024a. 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. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23173–23199, 2025. URL https://aclanthology.org/2025.findings-emnlp.1259/.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. PandaLM: An automatic evaluation benchmark for LLM instruction tuning optimization. In *International Conference on Learning Representations (ICLR)*, 2024b. URL https://arxiv.org/abs/2306.05087.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Egert Daniel, Jimmy Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. Helpsteer2: Open-source dataset for training top-performing reward models. In *Advances in Neural Information Processing Systems*, volume 37, pages 1474–1501. Curran Associates, Inc., 2024c.

Minghao Wu and Alham Fikri Aji. Style over substance: Evaluation biases for large language models. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 297–312, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.21/.

Yuxi Xia, Pedro Henrique Luz de Araujo, Klim Zaporojets, and Benjamin Roth. Influences on LLM calibration: A study of response agreement, loss functions, and prompt styles. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. URL https://aclanthology.org/2025.acl-long.188/.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *International Conference on Learning Representations (ICLR)*, 2024.

Peter C. Young. *Recursive Estimation and Time-Series Analysis: An Introduction*. Springer-Verlag, 1984.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 694–699, 2002.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. URL https://arxiv.org/abs/2102.09690.

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. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Bin Zhu, Yi Xie, and Yanghui Rao. A finite-calibration regime map for LLM judge panels, 2026. URL https://arxiv.org/abs/2606.01034. To appear at WISE 2026.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned large language models are scalable judges. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2310.17631.

A Formal Definitions of Calibration Methods

Section 4 introduces calibration methods and already gives formal definitions of methods directly (Section 4.1). Of the remaining, judge evaluation (Landesberg and Narayan, 2025), SCOPE (Badshah et al., 2026), and Neural ODE calibration (Morandi, 2026) are cited directly against a paper that already formalizes an LLM-judge calibration method, so we do not re-derive them here. For the remaining methods, whose citations (where they exist) are to classifier calibration or image-retrieval techniques from outside the LLM-judge setting, we give the exact correction g_ϕ each one fits.

Median scaling. With running median $\bar{j}$ over anchors seen so far and $u_i = j_i - \bar{j}$, $v_i = h_i - \bar{j}$,

$$T = \frac{\sum_i u_i^2}{\sum_i u_i v_i}, \qquad g_\phi(j) = \bar{j} + \frac{j - \bar{j}}{T}.$$

Few-shot exemplar calibration. Given the current exemplar pool $\mathcal{A} = \{(j_i, h_i)\}$ and a new score j, let $N_k(j) \subseteq \mathcal{A}$ be the k exemplars with smallest $|j - j_i|$; with weights $w_i = 1/(|j - j_i| + \epsilon)$,

$$g_\phi(j) = \frac{\sum_{i \in N_k(j)} w_i h_i}{\sum_{i \in N_k(j)} w_i}.$$

Streaming linear regression. An affine map fit by ordinary least squares,

$$g_\phi(j) = \alpha + \beta j, \qquad \beta = \frac{\sum_i (j_i - \bar{j})(h_i - \bar{h})}{\sum_i (j_i - \bar{j})^2}, \qquad \alpha = \bar{h} - \beta \bar{j},$$

with (α, β) recomputed after every anchor from running first/second moments and the cross-moment (recursive least squares), not refit from scratch.

Streaming Bayesian posterior update. With feature map $\phi(j) = (j^2,\, j,\, 1)^\top$, weight vector w, prior precision $\Lambda_0 = \tau I$, and observation noise variance σ^2, each new anchor (j_i, h_i) updates the posterior precision and precision-weighted mean in closed form,

$$\Lambda_i = \Lambda_{i-1} + \frac{\phi(j_i)\phi(j_i)^\top}{\sigma^2}, \qquad \Lambda_i \mu_i = \Lambda_{i-1}\mu_{i-1} + \frac{\phi(j_i)\, h_i}{\sigma^2},$$

and $g_\phi(j) = \phi(j)^\top \mu_n$ after n anchors.

Dawid–Skene EM. Scores are discretized into K classes. Let π_k be the prior over the latent true class k, and $\pi^J_{k,c}$, $\pi^H_{k,c}$ the judge's and (during fitting) the human's confusion matrices, $P(\text{observed class} = c \mid \text{true} = k)$. EM alternates

$$\gamma_i(k) \propto \pi_k\, \pi^J_{k,c_i^J}\, \pi^H_{k,c_i^H} \quad \text{(E-step)}, \qquad \pi_k \propto \sum_i \gamma_i(k), \quad \pi^J_{k,c} \propto \sum_i \gamma_i(k)\, \mathbb{1}[c_i^J = c] \quad \text{(M-step)},$$

with the human update analogous. At prediction time only the judge's class $c(j)$ is observed, and

$$g_\phi(j) = \sum_k \gamma(k \mid j)\, v_k, \qquad \gamma(k \mid j) \propto \pi_k\, \pi^J_{k,c(j)},$$

where v_k is class k's score midpoint.

Contrastive Rollout Calibration. Each judge output is a rollout over K rating values $r_1, \ldots, r_K$, snapped onto the fixed rating enum as a probability vector $p \in \Delta^{K-1}$ and mapped to logit space $z = \log(p + \epsilon)$. During fitting the (j, h) pairing is used directly: example i is *agreeing* if $|\sum_k r_k p_{ik} - h_i| \le \tau$ and *disagreeing* otherwise. Two streaming means accumulate logits over the two groups,

$$\mu_{\text{pos}} = \mathbb{E}[z \mid \text{agree}], \qquad \mu_{\text{neg}} = \mathbb{E}[z \mid \text{disagree}], \qquad v = \mu_{\text{pos}} - \mu_{\text{neg}}.$$

At prediction time only the judge rollout is observed. Let $\hat{v} = v/\|v\|$, $\text{target} = \mu_{\text{pos}} \cdot \hat{v}$, and $s = z \cdot \hat{v}$. The calibrated logits are

$$z' = z + \max(0,\ \text{target} - s)\,\hat{v},$$

leaving examples that already project at or beyond the agreeing cluster unchanged. The calibrated score is

$$g_\phi(j) = \sum_k r_k \,\text{softmax}(z')_k,$$

a convex combination of the rating enum and therefore always in $[\min_k r_k, \max_k r_k]$.

Conformal prediction. A streaming-linear-regression point estimate $\hat{h}(j) = \alpha + \beta j$ is maintained as above; over a sliding window of the W most recent nonconformity scores $r_i = |\hat{h}(j_i) - h_i|$, the margin is their $(1-\alpha)$-quantile,

$$g_\phi(j) = \hat{h}(j), \qquad \text{margin} = \text{Quantile}_{1-\alpha}\big(\{r_i\}_{i \in \text{window}}\big),$$

with the returned interval $[\hat{h}(j) - \text{margin},\ \hat{h}(j) + \text{margin}]$ used only for coverage evaluation, not for the point-estimate error metrics.

Platt scaling. Human scores are min-max normalized to soft targets $t_i = (h_i - h_{\min})/(h_{\max} - h_{\min}) \in [\epsilon, 1-\epsilon]$, and (A, B) are fit by minimizing cross-entropy against t_i,

$$(A, B) = \arg\min_{A,B} -\sum_i \Big[t_i \log \sigma(Aj_i + B) + (1 - t_i) \log\big(1 - \sigma(Aj_i + B)\big)\Big],$$

with σ the logistic sigmoid, and $g_\phi(j) = h_{\min} + (h_{\max} - h_{\min})\,\sigma(Aj + B)$.

Isotonic regression. The non-parametric monotone least-squares fit,

$$g_\phi = \arg\min_{g \text{ monotone non-decreasing}} \sum_i \big(h_i - g(j_i)\big)^2,$$

solved via the pool-adjacent-violators algorithm.

Score-debiased KDE (SD-KDE). At query j, with Gaussian kernel weights $w_i(j) = K\big((j - j_i)/h\big)$, a weighted local-linear fit $(\beta_0(j), \beta_1(j)) = \arg\min_{\beta_0,\beta_1} \sum_i w_i(j)\big(h_i - \beta_0 - \beta_1(j_i - j)\big)^2$ is solved at every query, and

$$g_\phi(j) = \beta_0(j),$$

the fitted local intercept – this removes the leading-order boundary/curvature bias of a plain Nadaraya–Watson weighted average $\sum_i w_i(j) h_i / \sum_i w_i(j)$.

B Supporting Material: Held-Out Metrics per Calibration Method

Table 2 in Section 5.1 reports the per-method, per-track results and the average, pooled effect of calibration across all methods. This section gives the full distributional view behind that table.

Figure 2 supports the data-limited full-batch comparison in Section 5.2.

Figures 3 and 4 show the same held-out MAE, Wasserstein distance, and exact-match rate, broken out per method (grouped by mode: streaming, batch, full-batch) rather than pooled. For each method, the raw-judge and calibrated distributions are shown as boxplots across all (dataset, model, attribute) held-out groups. Figure 3 reports each metric in its native units; because MAE, Wasserstein distance, and exact-match rate live on different scales, comparing methods across metrics is easier in Figure 4, where each metric is min–max normalized to $[0, 1]$ and exact-match rate is flipped to a mismatch rate ($1 -$ exact match), so that lower is always closer to human on every metric.

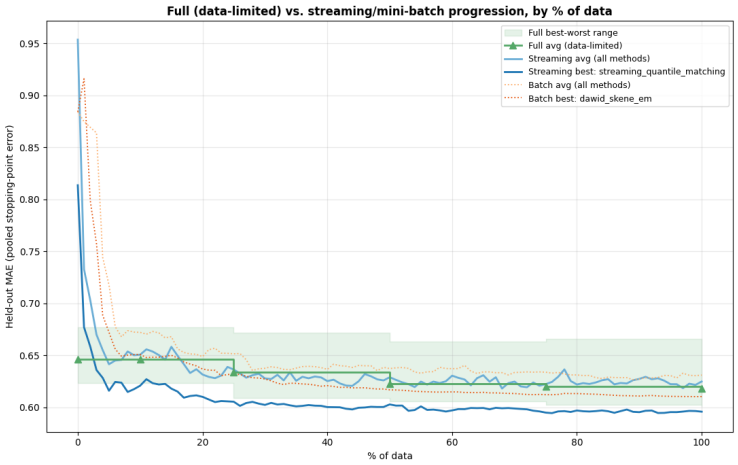


Figure 2: Full-batch method fit on limited amounts of training data (10/25/50/75/100%, green step function with best–worst range shaded) versus streaming's and batch's own pooled stopping-point curves across every percentile of data (best method and across-method average for each track). Held-out MAE, pooled across all four judge models.

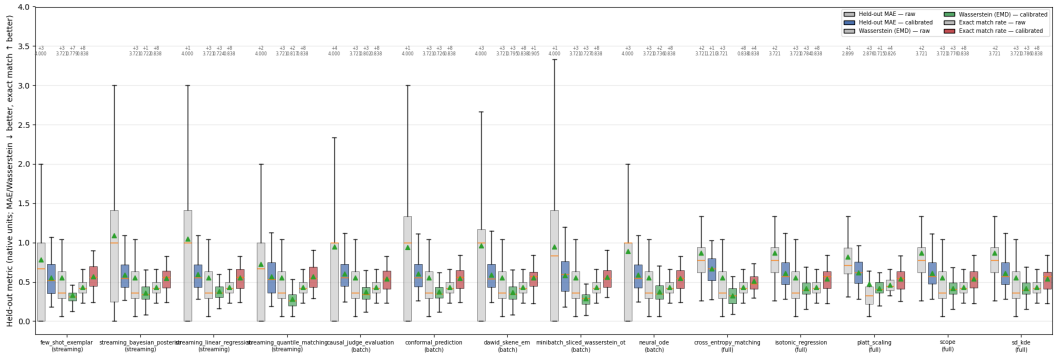


Figure 3: Held-out MAE, Wasserstein distance, and exact-match rate per calibration method (raw vs. best-checkpoint calibrated), in native units. Methods are grouped by track (streaming, batch, full-batch); outlier counts and the most extreme outlier value are annotated above each box.

B.1 Supporting Material: Automatic Stopping Criteria

Section 5.2 reports only the headline conclusion that fixed-horizon stopping outperforms adaptive stopping; this section gives the full comparison behind that statement. We compare five loss-aware adaptive stopping rules – an EMA-slope detector (`ema_slope`), a Page–Hinkley change-point test (`page_hinkley`), the ADWIN sliding-window drift detector (`adwin`), a champion-challenger holdout comparison (`champion_challenger`), and a reservoir-validation check (`reservoir_validation`) – against two fixed-horizon baselines that simply stop after exactly 100 or 200 examples (`stop_at_100`, `stop_at_200`), for every streaming and batch method, dataset, and judge model.

Figure 5 scores each rule two ways: how much worse its stopping point's held-out score is than that same run's own best-ever checkpoint (left), and how much worse it is than the score the run eventually reaches if left to consume the full pool (right; negative is better than the eventual final score). On both measures, the two fixed horizons dominate: median regret is smallest for `stop_at_200` (roughly 5–6% worse than the best checkpoint, and effectively on par with the eventual final score) and `stop_at_100` (roughly 9% and 2% respectively), while every adaptive rule has a higher median regret and a much heavier tail of outliers. `page_hinkley` is the least reliable of all seven rules, with a mean regret an order of magnitude above its own median (driven by runs where it fires

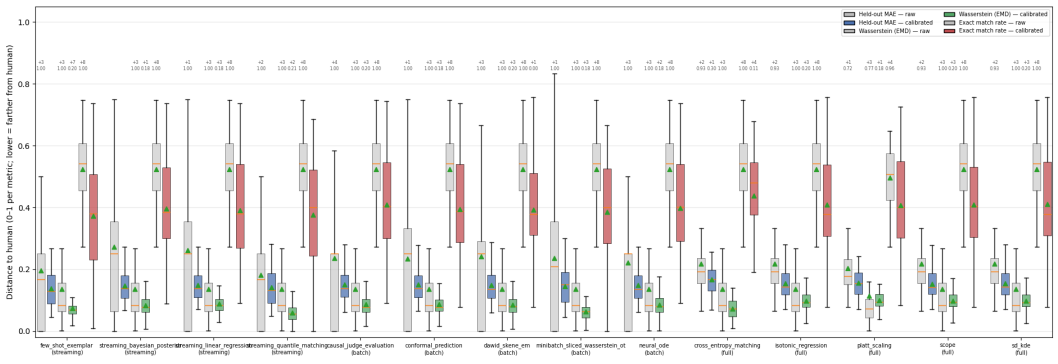


Figure 4: Same comparison as Figure 3, with each metric min–max normalized to $[0, 1]$ and exact-match rate expressed as a mismatch rate, so that lower is always closer to human on every metric.

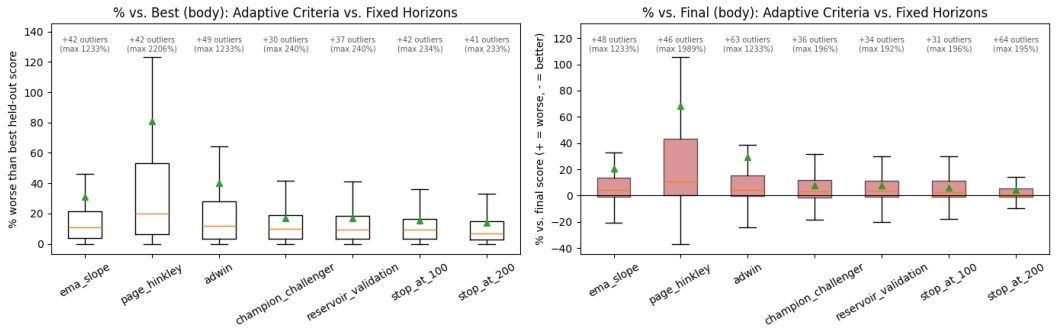


Figure 5: Automatic stopping rules compared on two regret measures, pooled across all streaming and batch methods, datasets, and judge models. **Left:** % worse than that run's own best-ever held-out score (lower is better). **Right:** % vs. the run's eventual final score after consuming the full pool (negative means the stopping point already beat the eventual final score). Orange line: median; green triangle: mean; outlier counts and the most extreme outlier are annotated above each box. The two fixed-horizon rules (`stop_at_100`, `stop_at_200`) have the lowest median and mean regret of all seven rules on both measures.

far too early or never fires at all) and outliers exceeding 1200–2200% worse than the best checkpoint. `champion_challenger` and `reservoir_validation` are the best-behaved of the five adaptive rules but still do not beat either fixed horizon on median or mean regret. In short, knowing the loss curve's shape in advance (a fixed horizon) is a more reliable stopping strategy in our setting than trying to detect convergence from the stream itself.

C Supporting Material: Per-Dataset and Per-Model Method Comparison

Section 5.3 looks at each method's held-out MAE reduction split out by judge model, and at its stability using the win/last-place record from Table 2. Figure 6 gives the per-model breakdown

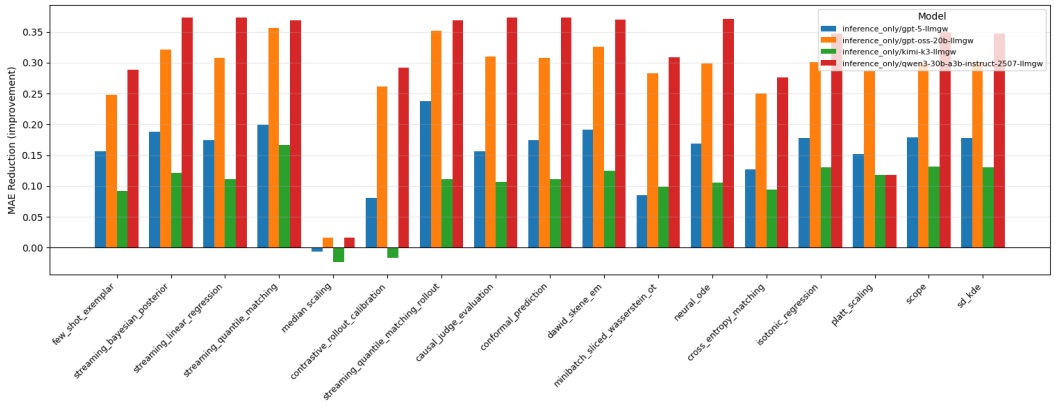


Figure 6: Held-out MAE reduction (raw judge minus calibrated final MAE) per calibration method, one bar cluster per judge model. Larger is better. Improvement is consistently largest for the two weaker judges (gpt-oss-20b, qwen3-30b) and smallest for kimi-k3; median scaling is the exception, with near-zero improvement for every model.