JudgeMoE: Distributional Aggregation for LLM-as-a-Judge

Yiqi Liu[1] **Joseph James**[2] **Yang Wang**[1]
Kun Zhao[3] **Chenghao Xiao**[4] **Chenghua Lin**[1]*

[1] The University of Manchester [2] The University of Sheffield
[3] University of Pittsburgh [4] Shanghai University of Finance and Economics

{yiqi.liu, chenghua.lin}@manchester.ac.uk

Abstract

When an LLM judge scores an output, its score distribution retains uncertainty and disagreement information that is lost after scalar compression. We introduce JudgeMoE, a lightweight aggregator that assigns example-specific weights to cached judge score distributions and fuses them before computing a final score. A protocol study shows that score-range choice is unstable across judge–dataset settings and that soft scoring usually outperforms hard decoding. On the original 10-cell benchmark, JudgeMoE improves mean Spearman over uniform log pooling by $+0.079$. Applying the same configuration to six additional cells yields a $+0.0393$ mean gain over the strongest local single judge across 16 cells, with positive differences in 12/16 cells and a one-sided Wilcoxon signed-rank $p = 0.0091$. Validation-based analyses further show that the preferred aggregation method depends on the task and judge pool.

1 Introduction

When an LLM judge assigns a score, the decoded label is only a compressed view of a richer score distribution. Two judges may both decode to 4 while expressing different uncertainty: one may concentrate mass on 4, while another spreads mass across 3, 4, and 5. Such differences matter for aggregation because confidence, entropy, and nearties affect whether judges provide consistent or uncertain evidence. We use *scoring protocol* to mean the score range, extraction rule, and prompt framing used to elicit this distribution.

LLMs are now widely used as automatic evaluators for summarization, dialogue, reasoning, and translation (Liu et al., 2023; Zheng et al., 2023). Their agreement with humans can change under protocol choices such as the score range and the rule used to extract a score (Wang et al., 2024a; Yamauchi et al., 2025; Li et al., 2026). Our protocol study shows that these choices change the measurement produced by the judge: across 30 dataset and model combinations where both soft and hard scores give usable rank correlations, each of the five tested score ranges is strongest in at least one setting, and soft-score extraction is stronger than hard decoding in most valid runs.

These findings motivate aggregation before score compression. Scalar averaging and scalar-gated aggregation combine decoded or expected scores after each judge distribution has already been reduced to one number. JudgeMoE instead pools the full judge score distributions and computes the final scalar score after pooling. Each judge distribution is treated as protocol-specific evidence over the benchmark's possible labels.

We use JudgeMoE as a lightweight testbed for this representation shift. With a frozen pool of small local judges and cached score distributions, the same outputs support several validation-guided deployment choices. A DeepSets gate assigns example-specific judge weights, and a tempered log-opinion pool fuses score buckets before final scoring. Our experiments focus on reusable pools of small local judges, while larger evaluators are included as standalone options in the validation-based selection procedure. We call one dataset paired with one score-scale condition a *cell*.

Our contributions are as follows:

- A protocol study showing that score-range choice is unstable across judge families and datasets, and that soft scoring is typically stronger than hard decoding.
- We introduce JudgeMoE, a lightweight conditional aggregator that builds on the protocol study by pooling cached judge score distributions before converting them to scalar scores.
- A validation-guided deployment analysis show-

*Corresponding author

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1016–1042
November 6-10, 2026 ©2026 Association for Computational Linguistics

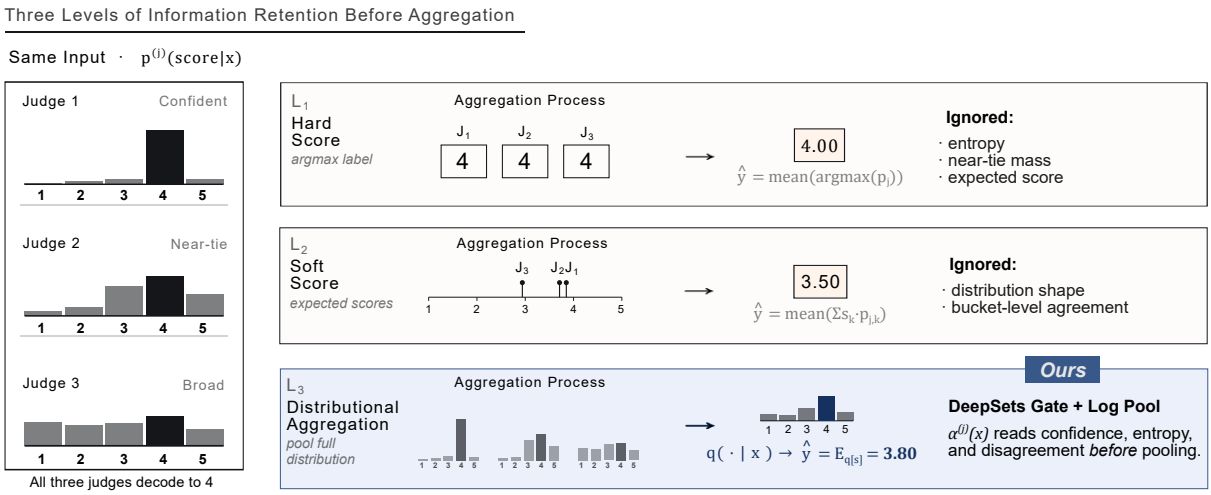


Figure 1: Three levels of information retention for the same judge score distributions. Hard-score aggregation uses only decoded labels, and soft-score aggregation uses expected scores. JudgeMoE pools the full score distributions before scoring, retaining confidence, entropy, near-tie mass, and bucket-level agreement during aggregation.

ing when fixed pooling, full JudgeMoE, sparse repair, or a standalone expert is supported by validation evidence.

2 Related Work

LLM-as-a-Judge. Recent work has established LLMs as scalable evaluators for open-ended generation (Liu et al., 2023; Zheng et al., 2023). Later studies show that positional bias and evaluator self-preference can affect agreement with humans (Wang et al., 2024a; Liu et al., 2024). Prompt design, safety artifacts, decoding choices, and grading scales can also change evaluation outcomes (Hong et al., 2026; Chen and Goldfarb-Tarrant, 2025; Yamauchi et al., 2025; Li et al., 2026). Other work improves individual judges through specialized evaluator models (Kim et al., 2024; Huang et al., 2025a), self-rationalization (Trivedi et al., 2024), checklist decomposition (Lee et al., 2025), and adaptive debiasing (Yang et al., 2026). Much of this line studies how to make a single judge more reliable; our work asks what can still be recovered by aggregating across multiple judges that produce different scores for the same example under a shared scoring protocol.

Evaluator ensembles and aggregation. Multi-model evaluation can use interaction, aggregation, or probability-level comparison across evaluators. Debate-style approaches let agents exchange rationales before producing a final answer (Chan et al., 2024; Du et al., 2024); jury-style approaches aggregate judgments from several models without explicit multi-turn interaction (Verga et al., 2024; Zhao et al., 2025b); and ContrastScore uses disagreement in model probabilities for generation evaluation (Wang et al., 2025b). Wang et al. (2025a) study inference from the judgment distribution of a single LLM evaluator and show that distributional information can improve over greedy decoded judgments. JudgeMoE studies conditional aggregation across multiple judges, using per-example score distributions under explicit scoring protocols and example-specific judge weights.

There is an established statistical tradition of aggregating noisy judgments under latent reliability models, from Dawid–Skene (Dawid and Skene, 1979) to recent LLM-specific probabilistic aggregation models (Zhao et al., 2025a). These models infer consensus labels and judge reliability from noisy annotations. They assume a shared discrete label space and a single latent consensus label, whereas we treat score distributions as protocol-dependent measurements and compare several deployment modes over cached outputs. Relatedly, QAEval uses the phrase "mixture of evaluators" for heterogeneous QA evaluators that mix rule-based and learned components (Yue et al., 2025); our study focuses on a homogeneous LLM pool and treats heterogeneous experts as optional task-aligned additions.

Adjacent settings. Pairwise preference learning provides relative supervision that can complement direct scores (Bradley and Terry, 1952; Jeong et al., 2025). RLHF work also uses reward-model en-

sembles or judge-based constraints inside policy optimization (Zhang et al., 2024; Zhai et al., 2024; Xu et al., 2025). Our setting is complementary: we aggregate frozen LLM judges directly at evaluation time, so one cached set of score distributions can support several validated aggregators without rerunning judge inference. This reuse objective distinguishes our setting from post-training evaluator ensembles.

3 Method

Setup and terminology. JudgeMoE takes each judge's probability distribution over possible scores as the input object. We use *bucket* as shorthand for one allowed score label, and a *bucket vector* is the normalized probability vector over those labels. A *protocol* is the score range, prompt, and score-extraction rule that produced that vector. A *cell* is one dataset paired with one score scale. A *gate* assigns an example-specific weight to each judge. A *log pool* fuses the weighted score distributions by taking a weighted geometric mean over buckets. *Sparse repair* means removing harmful judges on validation data and retraining the same aggregator on the selected subset. In short, JudgeMoE reads each judge's full distribution, decides how much to trust each judge for the current example, pools the distributions, and then computes one expected score.

3.1 Judge Representations

Each judge produces a normalized probability vector over the allowed score labels. Hard scores decode the argmax, soft scores take the expectation, and JudgeMoE uses the distribution itself. We optionally augment this bucket vector with scalar summaries, including hard and soft scores, confidence, entropy, and maximum bucket probability. Scores and targets are normalized to $[0, 1]$ for cross-scale comparison.

This is the main distinction from score-level aggregation. Scalar baselines aggregate decoded scores or expected scores, so the aggregator only sees compressed judge outputs. JudgeMoE keeps the bucket vector as the input object and lets the gate decide whether distributional differences matter for the current example. Feature ablations in the appendix show that much of the useful signal lies in the score distribution itself, with scalar uncertainty features adding a smaller gain over probabilities alone.

3.2 Aggregation Families

For deployment, we use validation labels to compare a small family of gating and pooling variants. The sweep includes feed-forward and contextual gates inspired by mixture-of-experts gating (Jacobs et al., 1991; Jordan and Jacobs, 1994), Deep Sets-style permutation-invariant gating (Zaheer et al., 2017), and probability fusion with log-opinion pooling (Heskes, 1997). We use this reproducible family to test whether aggregating judge score distributions before score compression improves evaluation quality.

The full validation comparison is shown in Appendix Table 13. The selected mainline uses a DeepSets gate with tempered log-opinion pooling ($T = 0.7$). For each example x, the gate assigns an example-specific weight to each judge from its distributional features. The following equations define the selected DeepSets gate and tempered log-opinion pool.

Let $\phi^{(j)}(x) \in \mathbb{R}^{d_\phi}$ be the feature vector of judge j, and let the shared encoder $\psi : \mathbb{R}^{d_\phi} \to \mathbb{R}^{d_h}$ map each judge feature vector into a common hidden space, $h^{(j)}(x) = \psi(\phi^{(j)}(x))$. With a shared scoring vector $a \in \mathbb{R}^{d_h}$, the judge weights are

$$\alpha^{(j)}(x) = \mathrm{softmax}_j\Big(a^\top h^{(j)}(x)\Big). \tag{1}$$

We use log-opinion pooling because multiplicative fusion rewards agreement across judges. The trade-off is increased sensitivity to any judge that assigns near-zero mass to the correct bucket: a single near-zero probability can push a bucket down unless that judge receives very small weight. The weights parameterize the tempered log-opinion pool:

$$q_k(x) \propto \exp\Big(\frac{1}{T}\sum_{j=1}^{m} \alpha^{(j)}(x) \cdot \log\Big(p_k^{(j)}(x) + \varepsilon\Big)\Big), \tag{2}$$

where $q_k(x)$ is the fused probability of score s_k, T is the pooling temperature, and ε is a small numerical constant added inside each judge log-probability for numerical stability. Lower temperatures sharpen the fused distribution toward buckets supported by the weighted pool, while higher temperatures flatten it. The final prediction is the expected score under the pooled distribution,

$$\hat{y}(x) = \sum_{k=1}^{K} s_k q_k(x). \tag{3}$$

Compared with linear pooling, this operator sharpens consensus and is often stronger. Its brittleness motivates the sparse-repair analysis in Section 6. In practice, the validation sweep over $T \in \{0.7, 1.0, 1.3\}$ selected $T = 0.7$. We use this configuration because it gives the best validation average while maintaining a small parameter count and judge-order invariance.

3.3 Training Objectives

Direct and auxiliary objectives. The primary evaluator is selected on direct-score quality; pairwise supervision is used only as an auxiliary analysis. We train primarily on direct-score labels with Huber loss:

$$\mathcal{L}_{\text{direct}} = \text{Huber}\big(\hat{y}(x), y\big). \tag{4}$$

In auxiliary experiments, we add a margin-ranking loss on pairwise preference data. For a pair (x_a, x_b) with human preference label $t \in \{-1, +1\}$ and scalar aggregator scores s_a, s_b, this loss is

$$\mathcal{L}_{\text{pair}} = \max(0, m - t\,(s_a - s_b))\,, \tag{5}$$

where m is a margin hyperparameter. The joint objective is

$$\mathcal{L} = \mathcal{L}_{\text{direct}} + \lambda \mathcal{L}_{\text{pair}}. \tag{6}$$

4 Experimental Setup

Our experiments consist of two parts. First, the protocol study examines how scoring protocols change judge output distributions. Second, the aggregation benchmark fixes the candidate judges and compares aggregation methods on direct-score cells. The original aggregation benchmark contains 10 dataset–scale cells. After the judge pool, architecture, feature construction, and training procedure had been fixed, we evaluated the same configuration on six additional cells from HelpSteer2 validation, FLASK, and WildBench. We report the original 10 cells and the six additional cells separately, and use all 16 cells for the expanded comparison.

4.1 Tasks and Datasets

Separate follow-ups then test heterogeneous experts, pairwise auxiliary training, and sparse repair. We evaluate JudgeMoE on direct-score and pairwise benchmarks spanning summarization (SummEval (Fabbri et al., 2021)), dialogue/helpfulness (TopicalChat-USR (Gopalakrishnan et al., 2019; Mehri and Eskenazi, 2020) and HelpSteer2 (Wang et al., 2024b)), reasoning/explanation quality (ROSCOE-eSNLI (Camburu et al., 2018; Golovneva et al., 2023)), translation quality estimation (WMT MQM (Freitag et al., 2021; Lommel et al., 2013)), and pairwise preference evaluation (MT-Bench-human (Zheng et al., 2023) and Summarize-from-Feedback validation sets (Stiennon et al., 2020)). These datasets include both high-resource cells where aggregation can be estimated reliably and smaller diagnostic cells where failure modes are visible. For HelpSteer2, Table 1 reports the main subset used in the original 10-cell benchmark; the held-out validation extension uses a separate source validation split summarized in the appendix.

Dataset	Task	Size
SummEval	Summarization	6,400
TopicalChat	Dialogue quality	2,160
ROSCOE-eSNLI	Reasoning quality	151
HelpSteer2	Helpfulness scoring	101,620 main
WMT MQM	Translation quality	150,347
MT-Bench-human	Pairwise pref.	2,575 eval
SFT validation	Pairwise pref.	5,000 eval / 20,000 train

Table 1: Datasets used in the paper. Direct-score datasets are evaluated with judge-score correlations, while pairwise datasets are used for auxiliary supervision and preference evaluation. The MT-Bench-human count is the filtered non-tie evaluation set; the SFT row lists the validation set used for pairwise evaluation and the training subset used for the auxiliary pairwise loss. Split-level counts appear in Appendix Table 9.

Split integrity. To prevent leakage across train, validation, and test splits, we preserve the original split labels for each dataset and use exact example-ID intersections only when constructing aligned subset comparisons. Model, judge-pool, and aggregation choices are made using validation labels only; test labels are used only for final reporting.

4.2 Judge Pool

Our candidate pool includes six small judges from multiple model families: Qwen2.5-7B-Instruct (Qwen et al., 2024), Qwen3.5-9B (Qwen Team, 2026), Gemma-2-9B-IT (Gemma Team et al., 2024), Gemma-3-12B-IT (Gemma Team et al., 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Ministral-3-8B-Instruct-2512 (Liu et al., 2026). The mainline comparison uses all six judges. The compact 3-judge configuration retains Gemma-2-9B-IT, Gemma-3-12B-IT, and Qwen2.5-7B-Instruct; this pool is selected once from the

protocol study and then held fixed for the aggregation benchmark. A family-diverse 3-judge control enforces three different model families across the selected judges; it is weaker and appears in the appendix. The pool is homogeneous in evaluator type because all members are local instruct-tuned LLM judges; model families vary. Heterogeneous follow-ups append one task-aligned expert at a time to stress-test whether validation keeps, rejects, or isolates a candidate expert.

4.3 Score Ranges and Prompting

We evaluate score ranges [1–3], [1–5], [1–7], [1–10], and [1–100] as a diagnostic study of protocol sensitivity. The protocol study checks whether scoring choices change the measurement; the aggregation experiments then freeze a shared protocol to isolate pooling from repeated scale search. The main JudgeMoE protocol uses a shared scale across judges, with 1–5 as the default and 1–10 as the primary ablation. These scales serve as fixed deployment settings for aggregation comparisons. We also run a held-out protocol study over 24 dataset–model cells in which the protocol is selected on validation and evaluated once on test.

For each allowed score label, we extract or score the corresponding label probability, normalize across the allowed labels, and use the expectation as the soft score; hard decoding takes the highest-probability label. Soft-vs-hard counts exclude runs where rank correlation is undefined, such as constant prediction or target vectors; the appendix gives the exact filtering rule.

Prompts are task-specific but structurally aligned so the comparison stays focused on scale and aggregation. The appendix summarizes prompt skeletons and the shared output contract; the released artifact contains the exact prompt strings and parsing scripts.

4.4 Baselines

Baselines include the best single judge, uniform average or log-opinion pooling, covariance-aware aggregation, aligned strong single-judge follow-ups on selected cells, and JudgeMoE variants. Unless stated otherwise, "best single" denotes the strongest single judge within the evaluated cell; validation-selected single-judge results are reported with the deployment-selection diagnostics. The covariance-aware baseline is the strongest fixed pool: it uses validation error covariance to set non-uniform judge weights, giving less influence to judges with large or highly correlated errors, while keeping those weights fixed across examples.

4.5 Implementation Details

All aggregation experiments are script-driven and use fixed split manifests, cached judge score distributions, and reproducible checkpoint configurations. Appendix Tables 10 and 40 summarize the hyperparameters, seed settings, prompt templates, runner scripts, and result files needed to reproduce the experiments.

4.6 Metrics

Before aggregation and evaluation, direct-score outputs are normalized to a common $[0, 1]$ range to make scores comparable across datasets with different scales. For direct-score cells from SummEval, TopicalChat-USR, HelpSteer2, ROSCOE-eSNLI, and WMT MQM, we use Spearman correlation with human labels as the primary metric. Pairwise preference cells are evaluated with pairwise accuracy. We report MAE and RMSE only when absolute score error is directly relevant, such as in calibration or error-magnitude analyses. Main-paper tables report equal-cell averages, where a cell is one dataset paired with one score scale and is treated as one deployment condition. Sample-weighted summaries use the number of aligned evaluation examples in each cell as weights. For comparisons between aggregation methods, we report the mean cell-level difference and apply a Wilcoxon signed-rank test to the cell-level differences. One-sided tests evaluate whether JudgeMoE improves over the baseline, and two-sided tests evaluate whether the differences are nonzero. Some cells share datasets and examples across score scales, so these tests summarize variation across evaluation conditions with dependence between related cells.

5 Main Results

We first show that scoring protocols change judge measurements, then test example-dependent weighting against fixed and single-judge baselines. The failures identify when validation should prefer fixed pooling, sparse repair, or a standalone expert.

5.1 Protocol Choices Change the Measurement

The protocol study shows that score range and extraction rule change judge behavior. Figure 2 shows two patterns: the best score range varies

Comparison	Baseline	JudgeMoE	Δ	95% CI	one-sided p	two-sided p
Uniform log	0.304	**0.383**	**+0.079**	[0.039, 0.119]	0.005	0.010
Cov-log	0.341	**0.383**	+0.043	[-0.001, 0.085]	0.053	0.105
Best single	0.341	**0.383**	+0.042	[-0.007, 0.088]	0.053	0.105

Table 2: Main 10-cell baseline comparisons for the selected 6-judge JudgeMoE checkpoint. JudgeMoE shows its largest and statistically significant gain over uniform log pooling. Comparisons to covariance-aware fixed pooling and the best single-judge baseline have positive mean gaps, but their confidence intervals cross zero and are not statistically significant. Confidence intervals are paired bootstrap intervals over cells. The last two columns report one-sided and two-sided paired Wilcoxon signed-rank test p-values over cells, without multiplicity correction. "Best single" is the strongest single judge within each evaluated cell.

across dataset–model combinations, and soft scoring improves over hard decoding in most valid runs. These results support aggregating score distributions rather than scalar summaries alone. Across 24 dataset–model cells, validation-selected protocols also outperform fixed 1–5 or 1–10 soft scoring on test (Appendix Table 15).

5.2 Conditional Aggregation

Table 2 compares the selected configuration with equal-weight distributional pooling, fixed scalar baselines, and the best individual judges on the original 10-cell benchmark.

Gain over uniform pooling. The most direct comparison is uniform log pooling, which uses the same six judges with equal weights. JudgeMoE gains +0.079 mean Spearman, with Wilcoxon signed-rank tests giving $p = 0.005$ for the one-sided improvement test and $p = 0.010$ for the two-sided difference test. This supports example-specific weighting over equal-weight distributional pooling for this judge pool.

Comparison with stronger baselines. Against covariance-aware log pooling and the best single judge, JudgeMoE has positive mean gains, but the confidence intervals cross zero (Table 2). We therefore treat these comparisons as inconclusive and use validation data to choose among learned weighting, fixed pooling, and single-judge deployment.

Expanded evaluation. We applied the same JudgeMoE configuration to six additional cells from HelpSteer2 validation, FLASK, and WildBench after the judge pool, architecture, features, and training procedure had been fixed. Across all 16 cells, JudgeMoE improves over the strongest single judge in the local pool by $+0.0393$ mean Spearman, with positive differences in 12/16 cells and a one-sided Wilcoxon signed-rank $p = 0.0091$ ($p = 0.0182$ two-sided). The mean difference from uniform log pooling is $+0.0601$ with 12/16 positive cells ($p = 0.0017$ one-sided), while the difference from covariance-aware log pooling is $+0.0250$ with 9/16 positive cells ($p = 0.0964$ one-sided). The six additional cells are reported individually in Appendix Tables 32, 33, and 47.

Cell-level variation. Appendix Table 27 shows that JudgeMoE improves over the best single judge in most cells, but fails on both TopicalChat score ranges and WMT MQM@1–5. Section 6 uses these failures diagnostically: TopicalChat favors fixed pooling, while WMT MQM@1–5 motivates sparse judge selection and retraining.

Supporting analyses. The appendix gives three additional analyses. First, excluding ROSCOE, the 8-cell average remains higher for JudgeMoE than for the best single judge (0.418 vs. 0.384). On the original 10 cells, the sample-weighted mean differences are $+0.0413$ relative to the best single judge and $+0.0549$ relative to covariance-aware log pooling; excluding ROSCOE gives $+0.0413$ and $+0.0548$, respectively (Appendix Table 26). Second, ablations show that the compact 3-judge setting remains competitive and that full score distributions carry most of the useful signal (Appendix Tables 13 and 18). In a separate 2×2 comparison using the same cells, judge manifests, feature construction, pooling temperature, 30-epoch schedule, and three random seeds, DeepSets with log pooling achieves the highest mean Spearman of 0.4068 (Appendix Table 14). Third, JudgeMoE obtains lower equal-cell ECE for the final scalar predictions than single-judge and fixed-pooling baselines (0.059 vs. 0.179–0.221; Appendix Table 28). This ECE compares normalized scalar predictions with human scores and does not measure calibration of the underlying score-token probabilities.

5.3 Focused Heterogeneous Stress Tests

For each heterogeneous follow-up, we append one task-aligned expert to the compact homogeneous pool. Validation labels choose among the homogeneous pool, the heterogeneous pool, a repaired subset, and the standalone expert when applicable. Table 3 reports test Spearman correlations and

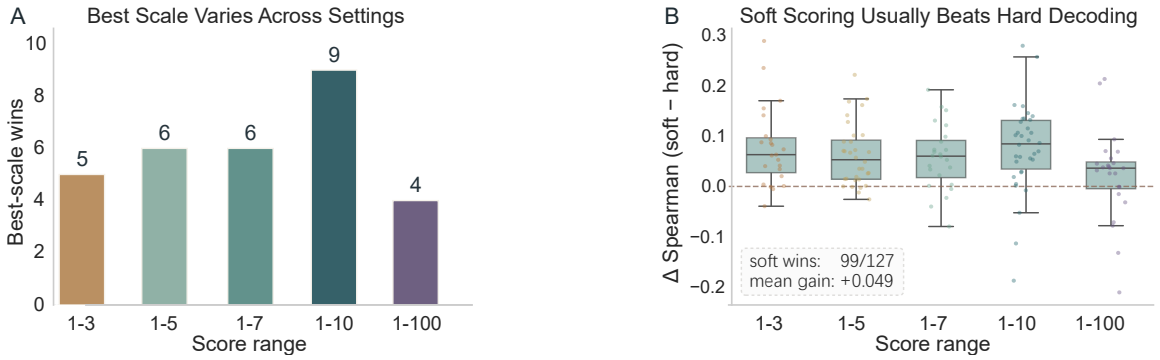


Figure 2: Protocol sensitivity. Left: the best score range varies across dataset–model cells. Right: soft scoring usually improves over hard decoding across valid runs and scales.

Follow-up	Hom. JudgeMoE	Het. JudgeMoE	Best single	Validation decision
SummEval + BERTSc.	0.646	**0.670**	0.517	keep expert
HelpSteer2 + RM	0.533	**0.545**	0.390	repair pool
WMT + trans. BERTSc.	**0.239**	0.133	0.190	keep LLM pool
WMT + COMET	0.247	0.162	**0.4326**	use standalone

Table 3: Focused heterogeneous follow-ups on compact deployment settings. All entries are test Spearman correlations with human labels; higher is better. BERTSc. denotes BERTScore and RM denotes the reward model. The validation decision is chosen using validation labels before test reporting. "Keep expert" selects the heterogeneous pool, "repair pool" selects a validation-repaired subset, "keep LLM pool" rejects the added expert, and "use standalone" keeps the external metric outside JudgeMoE.

Observed pattern	Choice
Protocol variation	Select protocol
Strong fixed pool	Fixed pooling
Learned-weight gain	Full JudgeMoE
Weak judge	Subset + retraining
Strong external evaluator	Standalone evaluator

Table 4: Validation-guided deployment choices supported by the protocol study, main aggregation experiments, TopicalChat and WMT analyses, and heterogeneous-expert experiments.

validation-selected deployment choices.

Positive additions. BERTScore helps on SummEval@1–10, and a Skywork reward model helps on held-out HelpSteer2 after train-split subset repair. These cases show that task-aligned experts can help when validation confirms complementary signal.

Rejected additions. Translation metrics give the stronger warning. On WMT MQM@1–10, translation-BERTScore is rejected by subset search. COMET is the strongest standalone evaluator at 0.4326, while mixed-pool JudgeMoE reaches 0.1620. Validation selects COMET as the standalone evaluator. Strong standalone performance therefore does not necessarily translate into strong performance after adding the evaluator to the learned pool.

6 Validation-Guided Deployment

6.1 Deployment Decision Matrix

The previous section reports the evidence used for deployment decisions: protocol choice changes judge behavior, learned weights improve over uniform pooling, TopicalChat favors fixed pooling, and WMT MQM@1–5 improves after pruning and retraining. Table 4 summarizes how these validation patterns guide the deployment choice for a new judge pool.

Shared-error check. Judges often make similar mistakes, which leaves less room for ensembling to help. Across the 10 direct-score cells, the mean judge-error correlation is 0.7775 for the full 6-judge pool and 0.7071 for the strong 3-judge pool (Appendix Table 20). Values around 0.7 to 0.8 indicate that much of the remaining error is shared across judges.

Validation selector. The shared-error check motivates a validation-based choice among full, fixed, sparse, and single-judge modes. A thresholded selector matches the test-time best aggregation family in 6/8 held-out direct-score cells (Appendix Table 21). A stronger validation-only selector reaches 0.3732 average test Spearman across the eight held-out direct-score cells, close to the 0.3789 obtained by selecting the best family using test performance. Its selected fixed subsets use 3.38 judges on average (Appendix Table 23).

Calibration checks also favor JudgeMoE over the strongest single and fixed-pool baselines on

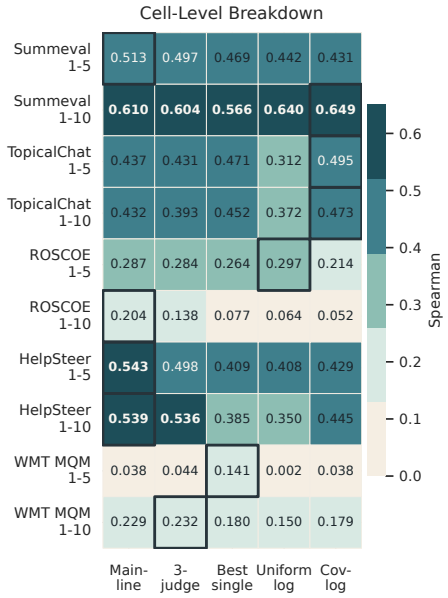


Figure 3: Per-cell pattern for the main direct-score benchmark. JudgeMoE improves over the best single judge in most direct-score cells, but the per-cell breakdown also identifies failures on TopicalChat and WMT MQM@1–5 (Appendix Table 27). TopicalChat is a case where covariance-aware fixed pooling already outperforms JudgeMoE, while WMT MQM@1–5 motivates sparse repair: retraining JudgeMoE on the validation-selected sparse subset improves Spearman from 0.0376 to 0.1648 (Tables 11 and 53).

dataset averages.

The focused heterogeneous follow-ups support the same conclusion. Task-aligned experts can help when they add new evidence to the homogeneous pool and pass train-split diagnostics, while prompt-diverse expansions of the same base models hurt, and both translation-BERTScore and COMET degrade the WMT mixed pool. This supports validating candidate experts before they enter the pool.

6.2 Cell-Level Patterns

The cell-wise breakdown explains why deployment should be selected on validation data. Most cells favor JudgeMoE, but TopicalChat and WMT MQM@1–5 show different failure patterns. TopicalChat is a fixed-pooling case: covariance-aware log pooling is already strong and residual judge-error correlation is extremely high. WMT MQM@1–5 is different because a weak judge contaminates the pool, so removing that judge and retraining the aggregator improves the result. These two cells motivate the fixed-pooling and sparse-repair rows in Table 4.

6.3 WMT Failure and Sparse Repair

The WMT MQM diagnostic isolates a concrete failure mode: coarse score ranges compress ranking information, weak judges can hurt log pooling, and sparse task-specific composition can outperform all-judge fusion. At WMT 1–5, the full-6 log pool degrades sharply and the learned mainline only partially recovers (Table 11). Leave-one-out validation identifies Qwen2.5-7B as a harmful weak judge. Keeping only the validation-selected top-2 subset and retraining sparse JudgeMoE raises WMT 1–5 to 0.1648, above both the best single judge at 0.1413 and the fixed sparse pool at 0.1473 (Table 53). The full gate already assigns negligible weight to Qwen2.5 (Appendix Table 52). Validation and test subset ranks are tightly aligned across all 120 WMT fixed-pool candidates, which consist of all single judges plus all non-single subsets under uniform-log and cov-log pooling (Appendix Table 51).

The heterogeneous translation follow-ups show the same failure mode. Appending translation-BERTScore drops the strong 3-judge JudgeMoE baseline from 0.2389 to 0.1328, and train-split subset search ranks the pure [Gemma-2, Gemma-3] subset first. COMET is more instructive: it becomes the strongest single expert at 0.4326 and uniform log pooling rises to 0.3926, while mixed-pool JudgeMoE still falls to 0.1620. Adding a strong external metric can therefore degrade the learned aggregator when its distributional behavior mismatches the LLM judges it was trained with. The conservative WMT recommendation is either a small clean LLM pool or the standalone metric itself. The appendix reports the full failure-and-recovery numbers.

The practical value comes from the reusable evidence layer as well as the full six-judge model. Once frozen judge distributions are cached, validation can choose among single, fixed, sparse, and gated modes without rerunning a monolithic evaluator.

6.4 Implications for Evaluation Practice

The results suggest a shift in how LLM judges are reported and reused. Individual judge quality remains important, yet the scoring protocol is part of the measurement and belongs with the judge output. A minimal benchmark report includes at least the score range, the score-extraction rule, and whether expected scores were computed from to-

ken probabilities or from decoded labels.

Probability access. We also compare variants using different levels of judge output information. Hard-score-only features obtain 0.2634 mean Spearman, soft-score-only features obtain 0.3100, hard and soft scores obtain 0.3671, and probability-derived scalar summaries obtain 0.3776. The difference between hard and soft scores together and hard scores alone is $+0.1036$ with a one-sided Wilcoxon signed-rank $p = 0.0068$, while the additional gain from probability-derived summaries over hard and soft scores is $+0.0106$ with $p = 0.0801$. Full results appear in Appendix Table 19.

7 Conclusion

LLM-as-a-judge evaluation is best understood as protocol-conditioned measurement. The final score is only a compressed view of a distribution shaped by the scoring protocol. Our results show that score ranges and extraction rules change judge behavior and that example-specific distributional aggregation improves over equal-weight log pooling on the original 10-cell benchmark. Across the expanded 16-cell evaluation, JudgeMoE improves over the strongest local single judge by $+0.0393$ mean Spearman, with positive differences in 12/16 cells and a one-sided Wilcoxon signed-rank $p = 0.0091$. The WMT-COMET case further shows that validation can favor a strong standalone evaluator when adding it to the learned pool reduces performance. The practical question is which cached measurements to trust, pool, prune, or keep standalone.

Limitations

The original aggregation benchmark contains 10 dataset–scale cells, followed by six additional cells evaluated with the same JudgeMoE configuration. The resulting 16-cell analysis covers more evaluation conditions, while several cells still share datasets, score scales, and judge families. Evaluation on additional datasets and criteria would further establish how the observed gains transfer across applications.

This paper studies learned aggregation over a fixed pool of small local judges, so results depend on candidate quality, task match, and score resolution. Larger evaluators are included as standalone comparison points, while aggregation over pools of larger or proprietary judges remains for future study. Broader comparisons across code execution, factuality checking, and other evaluation settings would also require careful cost normalization (Huang et al., 2025b).

The feature design is intentionally conservative. Current ablations suggest that judge distributions carry much of the useful signal. Comparisons with hidden states, explicit cross-judge interaction features, full natural-language explanations, and a broader same-capacity scalar-gated control remain useful next steps for separating representation gains from model-capacity gains.

The distributional approach also depends on access to informative judge score probabilities. The restricted-access experiments show that expected scores recover substantial signal beyond decoded scores, while probability-derived summaries provide a smaller additional improvement. The reported ECE measures the calibration of the final scalar predictions against human scores and does not establish calibration of the underlying score-token probabilities. In addition, the present benchmarks provide scalar human targets; annotator disagreement and multimodal human preferences remain outside the current scope.

Ethics Statement

Automatic evaluation systems can influence model selection, training feedback, and deployment decisions, so errors or biases in judge models may propagate directly into research conclusions. Judge aggregation adds its own risks: protocol choices can privilege some score distributions over others, and weak-judge contamination can distort pooled rankings. We therefore emphasize comparison against human annotations, cross-dataset evaluation, explicit discussion of failure modes, and clear separation between architecture selection, held-out checkpoint expansion checks, and validation-only deployment heuristics. The appendix summarizes benchmark provenance and split counts; the released artifact records original license constraints and external API use where applicable. JudgeMoE is best understood as an efficiency-oriented evaluation aid, with careful human assessment still required in high-stakes settings.

Acknowledgements

Joseph James was supported by the UKRI AI Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded

by UK Research and Innovation [grant number EP/S023062/1].

References

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. Chateval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*.

Hongyu Chen and Seraphina Goldfarb-Tarrant. 2025. Safer or luckier? LLMs as safety evaluators are not robust to artifacts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

A. P. Dawid and A. M. Skene. 1979. 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.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *The Twelfth International Conference on Learning Representations*.

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

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Riviere, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Leonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pages 1891–1895.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tom Heskes. 1997. Selecting weighting factors in logarithmic opinion pools. In *Advances in Neural Information Processing Systems*, volume 10.

Hanhua Hong, Chenghao Xiao, Yang Wang, Yiqi Liu, Wenge Rong, and Chenghua Lin. 2026. Beyond one-size-fits-all: Inversion learning for highly effective NLG evaluation prompts. *Transactions of the Association for Computational Linguistics*, 14:689–710.

Hui Huang, Xingyuan Bu, Hongli Zhou, Yingqi Qu, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. 2025a. An empirical study of LLM-as-a-judge for LLM evaluation: Fine-tuned judge model is not a general substitute for GPT-4. In *Findings of the Association for Computational Linguistics: ACL 2025*.

Tzu-Heng Huang, Harit Vishwakarma, and Frederic Sala. 2025b. Time to impeach LLM-as-a-judge: Programs are the future of evaluation. *arXiv preprint arXiv:2506.10403*.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Hawon Jeong, ChaeHun Park, Jimin Hong, Hojoon Lee, and Jaegul Choo. 2025. The comparative trap: Pairwise comparisons amplifies biased preferences of LLM evaluators. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*.

Michael I. Jordan and Robert A. Jacobs. 1994. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353.

Yukyung Lee, JoongHoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. 2025. Checkeval: A reliable LLM-as-a-judge framework for evaluating text generation using checklists. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15771–15798.

Weiyue Li, Minda Zhao, Weixuan Dong, Jiahui Cai, Yuze Wei, Michael Pocress, Yi Li, Wanyan Yuan, Xiaoyue Wang, Ruoyu Hou, Kaiyuan Lou, Wenqi Zeng, Yutong Yang, Yilun Du, and Mengyu Wang. 2026. Grading scale impact on LLM-as-a-judge: Human-LLM alignment is highest on 0-5 grading scale. *arXiv preprint arXiv:2601.03444*.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, and 1 others. 2026. Ministral 3. *arXiv preprint arXiv:2601.08584*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Yiqi Liu, Nafise Moosavi, and Chenghua Lin. 2024. LLMs as narcissistic evaluators: When ego inflates evaluation scores. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12688–12701.

Arle Richard Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2013. Multidimensional quality metrics: a flexible system for assessing translation quality. In *Proceedings of Translating and the Computer 35*, London, UK. Aslib.

Shikib Mehri and Maxine Eskenazi. 2020. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707, Online. Association for Computational Linguistics.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents. `https://qwen.ai/blog?id=qwen3.5`. Accessed: 2026-05-25.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021.

Prapti Trivedi, Aditya Gulati, Oliver Molenschot, Meghana Arakkal Rajeev, Rajkumar Ramamurthy, Keith Stevens, Tanveesh Singh Chaudhery, Jahnavi Jambholkar, James Zou, and Nazneen Rajani. 2024. Self-rationalization improves LLM as a fine-grained judge. *arXiv preprint arXiv:2410.05495*.

Pat Verga, Sebastian Hofstätter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024a. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450.

Victor Wang, Michael JQ Zhang, and Eunsol Choi. 2025a. Improving llm-as-a-judge inference with the judgment distribution. In *Findings of the Association for Computational Linguistics: EMNLP 2025*.

Xiao Wang, Daniil Larionov, Siwei Wu, Yiqi Liu, Steffen Eger, Nafise Sadat Moosavi, and Chenghua Lin. 2025b. ContrastScore: Towards higher quality, less biased, more efficient evaluation metrics with contrastive evaluation. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 3045–3060.

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

Tengyu Xu, Eryk Helenowski, Karthik Abinav Sankararaman, Di Jin, Kaiyan Peng, Eric Han, Shaoliang Nie, Chen Zhu, Hejia Zhang, Wenxuan Zhou, Zhouhao Zeng, Yun He, Karishma Mandyam, Arya Talabzadeh, Madian Khabsa, Gabriel Cohen, Yuandong Tian, Hao Ma, Sinong Wang, and Han Fang. 2025. The perfect blend: Redefining RLHF with mixture of judges. In *The Thirteenth International Conference on Learning Representations*.

Yusuke Yamauchi, Taro Yano, and Masafumi Oyamada. 2025. An empirical study of LLM-as-a-judge: How design choices impact evaluation reliability. *arXiv preprint arXiv:2506.13639*.

Bo Yang, Lanfei Feng, Yunkui Chen, Yu Zhang, Xiao Xu, and Shijian Li. 2026. Fairjudge: An adaptive, debiased, and consistent LLM-as-a-judge. *arXiv preprint arXiv:2602.06625*.

Tan Yue, Rui Mao, Xuzhao Shi, Shuo Zhan, Zuhao Yang, and Dongyan Zhao. 2025. Qaeval: Mixture

of evaluators for question-answering task evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14717–14730, Vienna, Austria. Association for Computational Linguistics.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander J. Smola. 2017. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30.

Yuanzhao Zhai, Han Zhang, Yu Lei, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang. 2024. Uncertainty-penalized reinforcement learning from human feedback with diverse reward LoRA ensembles. *arXiv preprint arXiv:2401.00243*.

Shun Zhang, Zhenfang Chen, Sunli Chen, Yikang Shen, Zhiqing Sun, and Chuang Gan. 2024. Improving reinforcement learning from human feedback with efficient reward model ensemble. *arXiv preprint arXiv:2401.16635*.

Jitian Zhao, Changho Shin, Tzu-Heng Huang, Satya Sai Srinath Namburi GNVV, and Frederic Sala. 2025a. From many voices to one: Statistically principled aggregation of LLM judges. *OpenReview*.

Justin Zhao, Flor Miriam Plaza-del Arco, Benjamin Genchel, and Amanda Cercas Curry. 2025b. Language model council: Democratically benchmarking foundation models on highly subjective tasks. In *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 12395–12450.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *NeurIPS 2023 Datasets and Benchmarks Track*.

A Background and Problem Setup

We consider an evaluation instance x consisting of task-specific context and a candidate output, with human target signal $y \in [0, 1]$. Judge j produces a categorical distribution $p^{(j)}(\cdot \mid x)$ over a finite score range $\mathcal{S} = \{s_1, \ldots, s_K\}$, where the score values are normalized to $[0, 1]$ before aggregation and evaluation.

Hard score. The hard score is the decoded label:

$$\hat{y}^{(j)}_{\text{hard}} = \arg\max_{s \in \mathcal{S}} p_j(s \mid x). \tag{7}$$

Soft score. The soft score uses the expected score under the judge distribution:

$$\hat{y}^{(j)}_{\text{soft}} = \sum_{s \in \mathcal{S}} s\, p_j(s \mid x). \tag{8}$$

Our experiments compare both protocols across several score ranges. The empirical motivation for JudgeMoE is that the optimal score range is unstable across judges and tasks, while each judge carries complementary information through its full score distribution, confidence, and entropy. This motivates an aggregation layer that consumes score distributions directly while preserving information that would be discarded after decoding a single label.

Goal. Given multiple judges $\{j_1, \ldots, j_m\}$, we want to learn an aggregation function

$$\hat{y}(x) = f\big(z^{(1)}(x), \ldots, z^{(m)}(x)\big), \tag{9}$$

where each $z^{(j)}(x)$ includes score-distribution features and optional metadata. The desired aggregator is lightweight, easy to train, and compatible with both direct-score and pairwise supervision.

B Additional Experimental Settings and Results

Roadmap. The appendix first gives prompt templates and artifact accounting, then reports the full per-cell and deployment-rule tables behind the main results. It next provides implementation and selection details, pairwise and out-of-domain checks, qualitative and calibration diagnostics, compute proxies, and finally the theoretical propositions used to interpret the operating conditions.

Component	Prompt content
Instruction	Evaluator role + criterion
Context	Source / prompt / dialogue
Candidate	Output to score
Reference	Optional task reference
Output	Scalar score in specified range

Table 5: Shared prompt components for direct-score judging. Dataset-specific wording varies, while all prompts use the same scalar-score output format.

B.1 Prompt Templates

Table 5 summarizes the prompt policy used throughout the project, and Table 6 gives the default direct-score template skeletons. The exact source and candidate fields differ across summarization, dialogue, reasoning, and translation, but the output contract is deliberately uniform so that scale and extraction effects can be isolated cleanly.

B.2 Additional Tables

Table 7 lists the experiment artifacts used to construct the paper tables. We keep the full benchmark matrices in the released artifacts and include compact summaries in the paper due to space constraints.

Architecture variants. The validation sweep compares several gates for computing judge weights $w_j(x)$. The linear gate uses

$$w(x) = \text{softmax}(W\phi(x) + b),$$

where $\phi(x)$ concatenates the per-judge features. The low-rank gate factorizes the weight matrix as $W = UV^\top$ to reduce parameters. The DeepSets gate computes permutation-invariant judge weights by applying a shared encoder to each judge feature vector and normalizing the resulting scores:

$$h_j = \rho(\phi^{(j)}(x)), \qquad w_j(x) = \text{softmax}_j(a^\top h_j).$$

For distributional variants, the final pooled distribution is computed with tempered log-opinion pooling,

$$p_{\text{pool}}(y \mid x) \propto \prod_{j=1}^{J} p_j(y \mid x)^{w_j(x)/T}.$$

B.3 Implementation Details

Table 40 summarizes the implementation choices that matter for reproducibility.

Task family	Default direct-score template skeleton
Summarization	`You are an expert summarization evaluator. Evaluate the [aspect] of the generated summary. [Aspect-specific criterion.] SOURCE DOCUMENT: {source} GENERATED SUMMARY: {candidate} Return only a score from {low} to {high}.`
Dialogue	`You are an expert dialogue response evaluator. Evaluate the [aspect] of the assistant response. CONTEXT: {source} ASSISTANT RESPONSE: {candidate} Return only a score from {low} to {high}.`
Assistant response	`You are an expert evaluator of assistant responses. Evaluate the [aspect] of the response. USER REQUEST: {source} ASSISTANT RESPONSE: {candidate} Return only a score from {low} to {high}.`
Reasoning	`You are an expert evaluator of reasoning chains. Evaluate the overall quality of the reasoning chain. Consider correctness, completeness, and coherence. PROBLEM OR CONTEXT: {source} REASONING CHAIN: {candidate} Return only a score from {low} to {high}.`
Translation	`You are an expert translation evaluator. Evaluate the overall translation quality. Consider adequacy and fluency. SOURCE SEGMENT: {source} TRANSLATION: {candidate} [optional HUMAN REFERENCE] Return only a score from {low} to {high}.`
FLASK-style	`You are an expert evaluator of assistant responses. Evaluate the candidate response using the provided criterion and scoring rubric. INSTRUCTION: {source} CANDIDATE RESPONSE: {candidate} CRITERION: {criterion} [reference/rubric blocks] Return only a score from {low} to {high}.`

Table 6: Default direct-score prompt templates. All task families use the same scalar-score output format; prompt-style variants are evaluated separately in Table 16.

Experiment	Paper content
Protocol study	Scale + soft/hard scoring
Aggregation	Main results + ablations
Seed sweep	6-judge / 3-judge variation
Pairwise study	Direct + pairwise training
Large judges	Quality + efficiency
WMT MQM	Pool failure + sparse retraining

Table 7: Experiment groups and their corresponding analyses in the paper.

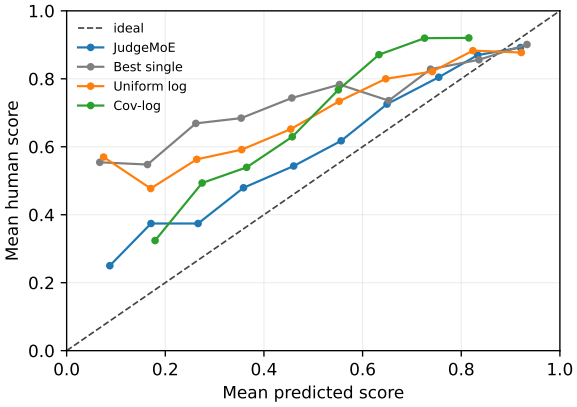


Figure 4: Equal-cell reliability diagram for the main direct-score benchmark. For each system and each bin, we average the bin-wise predicted score and human score across the 10 benchmark cells, so the plot complements the equal-cell ECE table and the example-weighted summary.

Selection differs across the paper's components, so Appendix Table 31 summarizes the main procedures. The protocol study evaluates scoring choices separately from aggregation. The main aggregation benchmark uses one shared 6-judge configuration across the original 10 direct-score cells, with 1–5 as the default scale and 1–10 as the main alternative. The six additional cells from HelpSteer2 validation, FLASK, and WildBench were evaluated after the judge pool, architecture, feature construction, and training procedure had been fixed. The compact 3-judge pool is selected once from the protocol-study screening and then held fixed. Appendix Table 42 reports a separate 3-judge pairwise experiment. For the WMT subset experiment, judge subsets are selected on WMT validation data and then retrained and evaluated on the corresponding test split.

The revision-only interpretability and calibration diagnostics are generated directly from the released per-example artifacts by the diagnostic-table generation program in the artifact packet. That program regenerates Appendix Tables 28, 29, 50, 51, and 52 together with Figure 4.

B.4 Seed Stability

Table 41 summarizes seed stability for the two main settings.

B.5 Pairwise Joint-Training Details

Table 43 gives the per-dataset pairwise improvements for the best joint setting (pairwise loss weight 0.5). These results support the main claim that pairwise supervision is best used as an auxiliary signal alongside direct-score supervision.

B.6 Additional WMT Diagnostic Visualization

B.7 Focused Theoretical Perspective

This section keeps only the theory that ties directly to the JudgeMoE formulation and the empirical storyline. The propositions below explain three con-

Family	Aggregation input	Full distribution	Example-specific weights	Interaction
Single-judge methods	Score / rationale / checklist	No	No	No
Jury / debate	Votes / answers / rationales	Usually no	Sometimes	Yes
Latent-reliability models	Labels / preferences	No	Judge-level	No
Scalar weighting	Scalar scores	No	Yes	No
Fixed opinion pooling	Score distributions	Yes	No	No
JudgeMoE	Cached score distributions	Yes	Yes	No

Table 8: Comparison of related evaluator aggregation methods. JudgeMoE combines full score distributions with example-specific judge weighting.

Dataset	Train	Dev	Valid	Test	Total	Note
Direct-score corpus						
All direct-score data	212,953	15,912	5,190	31,813	265,868	–
HelpSteer2	–	–	–	–	106,810	Train / validation preserved
SummEval	–	–	–	–	6,400	Human scores retained
TopicalChat	–	–	–	–	2,160	Dialogue aspects retained
ROSCOE	–	–	–	–	151	Diagnostic subset
WMT MQM	–	–	–	–	150,347	MQM metadata retained
Pairwise corpus						
All pairwise data	724,238	6,878	104,522	48,916	884,554	MT-Bench ties removed

Table 9: Dataset composition and split counts for the released evaluation data. Licenses and use restrictions follow the original datasets and are recorded in the release manifest.

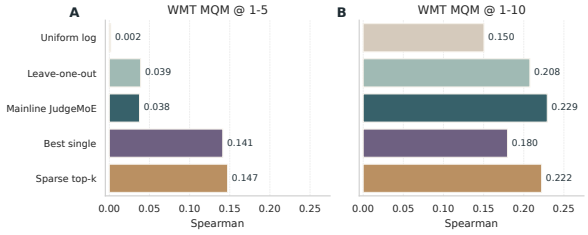


Figure 5: Compact visualization of the WMT MQM failure mode. At 1–5, sparse composition and dropping one weak judge both improve over naive full-pool aggregation, and the sparse retraining follow-up shows that a cleaned judge pool can repair the failure. At 1–10, better score resolution lifts both sparse pooling and the learned mainline model, with sparse retraining retaining most of the mainline gain.

crete design and deployment choices: why score distributions can contain more information than compressed scores, why the best pooling temperature is often $T<1$, and when sparse repair helps a contaminated log pool. Several small judges can still lose to a larger judge, as the WMT failure case shows. Proposition 1 together with Propositions 3 and 4 are retained as background facts; Propositions 6–8 are specific to the JudgeMoE setup and connect directly to the main empirical findings.

Notation. Let the ordered human score take values in $\{v_1, \ldots, v_K\} \subset \mathbb{R}$. A judge outputs a categorical distribution $P = (P_1, \ldots, P_K)$ over these values. For multi-judge pooling, let $w \in \Delta^{m-1}$ denote a pooling weight vector. When comparing two candidate score buckets a and b, write the per-judge likelihood ratio as $r_j(a, b) = p_j(a)/p_j(b)$.

Proposition 1 (Conditional mean minimizes squared loss). Let the ordered human score take values in $\{v_1, \ldots, v_K\} \subset \mathbb{R}$, let a judge output a categorical distribution $P = (P_1, \ldots, P_K)$ over these values, and define $m(P) = \mathbb{E}[Y \mid P]$. Then for any measurable scalar summary $a(P)$,

$$\mathbb{E}\big[(Y - a(P))^2 \mid P\big] = \mathrm{Var}(Y \mid P) + \big(m(P) - a(P)\big)^2. \tag{10}$$

Hence the unique minimizer of conditional squared loss is

$$a^\star(P) = m(P). \tag{11}$$

If the judge distribution is calibrated with respect to the human score distribution, so that

$$m(P) = \sum_{k=1}^{K} v_k P_k \equiv \mu(P), \tag{12}$$

then the soft score $\mu(P)$ is the optimal scalar summary under squared loss. Under the same calibration condition, the hard score $h(P) = v_{\arg\max_k P_k}$ is a Bayes action under exact-class 0–1 loss.

Item	Setting	Reference
Architecture	DeepSets, d_h=128, dropout 0.1, log pool, T=0.7, ε=10^{-6}	Table 13
Optimization	Adam, lr 8×10^{-4}, batch 128, 20 epochs	Released logs
Objective	Huber loss; auxiliary ranking loss with margin 0.0	Section 3.3, Table 42
Seeds	13; sweep: 13, 17, 29	Table 41
Features	K score probabilities + scalar summaries	Table 18
Splits	Original split labels; exact example-ID intersection	Table 9
Release	Runners, manifests, prompts, configs, results	Release package

Table 10: Key settings and reproducibility references for the main aggregation experiments. The selected configuration is unchanged for the six additional evaluation cells.

Dataset	Scale	Mainline	3-judge	Best single	Best judge	Uniform log	Cov-log
SummEval	1–5	0.5134	0.4967	0.4687	Gemma-2-9B	0.4420	0.4313
SummEval	1–10	0.6101	0.6043	0.5657	Gemma-3-12B	0.6401	0.6494
TopicalChat	1–5	0.4357	0.4297	0.4683	Gemma-2-9B	0.3108	0.4928
TopicalChat	1–10	0.4270	0.3925	0.4458	Gemma-2-9B	0.3700	0.4708
ROSCOE	1–5	0.2870	0.2841	0.2641	Gemma-3-12B	0.2975	0.2142
ROSCOE	1–10	0.2041	0.1381	0.0773	Ministral-3-8B	0.0644	0.0516
HelpSteer2	1–5	0.5426	0.4977	0.4087	Gemma-2-9B	0.4076	0.4285
HelpSteer2	1–10	0.5386	0.5356	0.3849	Gemma-2-9B	0.3497	0.4449
WMT MQM	1–5	0.0376	0.0443	0.1413	Gemma-2-9B	0.0016	0.0383
WMT MQM	1–10	0.2293	0.2316	0.1797	Gemma-2-9B	0.1504	0.1793

Table 11: Per-dataset breakdown for the main aggregation systems. The gains vary across dataset and scale cells: the main gains come from SummEval at 1–5, HelpSteer, and WMT MQM at 1–10, while TopicalChat and SummEval at 1–10 remain challenging against strong fixed baselines. ROSCOE is reported as a small-sample diagnostic cell and is interpreted cautiously because it contains only 151 examples.

Temperature	Avg. Sp.	Δ vs. single	Wins / 10
T=1.0	0.3828	0.0414	2
T=0.7	**0.3832**	**0.0419**	**4**
T=1.3	0.3729	0.0315	4

Table 12: Temperature sensitivity for DeepSets + log pooling. The values are taken from the 3-way temperature comparison on the same 10 dataset–scale cells.

Learned aggregation configuration	Avg. Sp.
Linear + log pool	0.3511
FFN + linear pool	0.3505
Low-rank linear + log	0.3546
DeepSets + log pool	0.3828
DeepSets + log, T=0.7	**0.3832**
3-judge efficient	0.3656

Table 13: Learned-aggregation selection on the original 10-cell suite. The selected mainline is DeepSets gating with tempered log pooling at T=0.7; Table 2 uses this fixed configuration for the main baseline comparisons.

Proof sketch. Apply the bias–variance decomposition around the conditional mean $m(P) = \mathbb{E}[Y \mid P]$. The variance term is independent of $a(P)$, so the conditional squared loss is minimized at $a(P) = m(P)$. Under score-distribution calibration, $m(P) = \mu(P)$.

Interpretation. Expected scoring is the natural one-dimensional reduction when the judge distribution is aligned with the human-score distribution. The empirical protocol study examines how well this reduction works with extracted LLM score probabilities in practice.

Proposition 3 (Correlated error floor for averaging). Assume judge score errors ε_j are zero mean with

$$\mathrm{Var}(\varepsilon_j) = \sigma^2, \qquad \mathrm{Corr}(\varepsilon_i, \varepsilon_j) = \rho \text{ for } i \neq j,$$

where $\rho \in [0, 1)$. For uniform averaging,

$$\bar{\varepsilon} = \frac{1}{m} \sum_{j=1}^{m} \varepsilon_j,$$

the error variance is

$$\mathrm{Var}(\bar{\varepsilon}) = \sigma^2 \left(\rho + \frac{1-\rho}{m} \right). \tag{13}$$

Therefore,

$$\lim_{m \to \infty} \mathrm{Var}(\bar{\varepsilon}) = \sigma^2 \rho. \tag{14}$$

Proof sketch. Expand $\mathrm{Var}(\bar{\varepsilon})$ into diagonal and off-diagonal covariance terms. Under the

Gate	Linear pooling	Log pooling
FFN	0.3623 ± 0.0016	0.3751 ± 0.0004
DeepSets	0.3831 ± 0.0026	$\mathbf{0.4068 \pm 0.0016}$

Table 14: 2×2 gate and pooling comparison on the original 10 cells. All four settings use the same judge manifests, feature construction, pooling temperature, 30-epoch training schedule, and three random seeds. Values are mean Spearman $\pm$ standard deviation. This experiment uses a separate training schedule from the reported mainline checkpoint, so the absolute values are interpreted within this table.

Selector	Equal-cell Sp.	Weighted Sp.
Validation-selected protocol	0.1865	0.1674
Fixed 1–5 soft	0.1486	0.0269
Fixed 1–10 soft	0.1813	0.1002
Best protocol by test performance	0.2569	0.1681

Table 15: Held-out protocol study over 24 dataset and model cells. The validation-selected protocol outperforms fixed 1–5 and 1–10 soft scoring on equal-cell Spearman. The selected protocols span all five tested score ranges, showing that protocol sensitivity is not limited to one fixed range.

exchangeable-correlation assumption, the diagonal terms contribute σ^2/m and the off-diagonal terms contribute $\sigma^2\rho(1 - 1/m)$.

Interpretation. Ensembling removes only the idiosyncratic component of judge noise. Shared bias or highly correlated error remains as an irreducible floor, which formalizes why judge diversity matters more than simply adding more judges.

Proposition 4 (Weak-judge veto threshold under log pooling). Consider a weighted log pool

$$q(y) \propto \exp\left(\sum_{j=1}^{m} \alpha_j \log p_j(y)\right), \qquad \alpha_j \geq 0, \qquad \sum_j \alpha_j = 1. \tag{15}$$

For two competing score buckets a and b,

$$\log \frac{q(a)}{q(b)} = \sum_{j=1}^{m} \alpha_j \log \frac{p_j(a)}{p_j(b)}. \tag{16}$$

Hence, if one weak judge $j = 0$ prefers b over a while the remaining judges prefer a over b, the pooled ranking flips in favor of b if and only if

$$\alpha_0 \log \frac{p_0(b)}{p_0(a)} > \sum_{j \neq 0} \alpha_j \log \frac{p_j(a)}{p_j(b)}. \tag{17}$$

Style	Rows	Avg. soft Sp.	Wins vs. default
Default	12	0.3230	0/12
Rubric	12	0.2275	0/12
Critical	12	0.2968	4/12

Table 16: Reduced prompt-style protocol check over two judges, three datasets, two scales, and three phrasing variants. A non-default prompt is best in 4/12 dataset–model–scale cells. The rubric wording is uniformly weaker than the default prompt, while the critical wording helps in a small subset of cases, including Qwen on WMT MQM (+0.1158 at 1–5 and +0.3023 at 1–10).

Pool	Judges	Mean ρ	JudgeMoE Sp.	Wins vs. default
Default 2	2	0.7955	0.4049	–
Protocol-diverse 6	6	0.8472	0.3883	3 / 3

Table 17: Protocol-diverse pool pilot on `summeval`, `topical_chat`, and `wmt_mqm` at `1-5` and `1-10`. The 6-judge pool augments a default 2-judge pool (`Qwen2.5-7B`, `gemma-2-9B`) with `rubric` and `critical` prompt variants of the same two models. Mean residual correlation increases, and average JudgeMoE performance drops slightly, indicating that added protocol diversity can fail to yield useful complementarity.

In the homogeneous equal-weight case with m good judges, $p_j(a) = p$, $p_j(b) = p'$, and one weak judge with $p_0(a) = \delta$, $p_0(b) = \delta'$, this reduces to

$$\log(\delta'/\delta) > m \log(p/p'). \tag{18}$$

Proof sketch. Take the log ratio of the pooled probabilities for a and b; the result is exactly the weighted sum of per-judge log-likelihood ratios. The flip condition is simply the sign change of this quantity.

Interpretation. Log pooling rewards agreement, but it also gives every judge multiplicative influence. A weak judge that places too little mass on the correct bucket can therefore veto the remaining judges. This is the mechanism behind the WMT weak-judge pollution result.

Linear-pool contrast. Under equal-weight linear pooling in the same homogeneous setting, the ranking flips only when

$$\delta' - \delta > m(p - p').$$

The weak judge enters additively in probability space, so single-judge pollution is less explosive than in log pooling.

Feature set	Avg. Sp.	Δ vs. single	Wins / 10
Full current	**0.3832**	**0.0419**	**6**
Distribution + scores	0.3811	0.0398	2
Distribution only	0.3746	0.0332	2

Table 18: Feature ablation for the main DeepSets + log-pool configuration. Most of the signal already lies in the judge score distributions, but the full scalar summary bundle provides a small and consistent gain over using distributions alone.

Input features	Mean Sp.	SD
Hard score only	0.2634	0.0028
Soft score only	0.3100	0.0088
Hard and soft scores	0.3671	0.0027
Probability-derived scalar summaries	0.3776	0.0053

Table 19: Performance with different levels of access to judge outputs on the original 10 cells.

Proposition 6 (Optimal pooling temperature as a calibration diagnostic). Suppose the ground-truth posterior over K ordered scores given instance x is $p^*(\cdot \mid x)$. Model each judge j as producing a temperature-scaled version of that posterior:

$$p_k^{(j)}(x) = \frac{\big(p_k^*(x)\big)^{1/\tau_j}}{Z_j(x)}, \qquad Z_j(x) = \sum_{\ell=1}^{K} \big(p_\ell^*(x)\big)^{1/\tau_j} \tag{19}$$

where $\tau_j > 0$ is the calibration temperature of judge j ($\tau_j = 1$ calibrated, $\tau_j > 1$ under-confident, $\tau_j < 1$ over-confident). Under equal-weight tempered log pooling at temperature T, the fused distribution satisfies

$$q_k^T(x) \propto \big(p_k^*(x)\big)^{\gamma(T)}, \qquad \gamma(T) = \frac{1}{T\,\bar{\tau}_H}, \tag{20}$$

where $\bar{\tau}_H = m \big/ \sum_{j=1}^{m} \tau_j^{-1}$ is the harmonic mean of the judges' calibration temperatures. The fused distribution recovers the true posterior ($q^T = p^*$) if and only if

$$T^* = \frac{1}{\bar{\tau}_H} = \frac{1}{m} \sum_{j=1}^{m} \frac{1}{\tau_j}. \tag{21}$$

At this temperature the expected-score predictor $\hat{y}^{T^*}(x) = \sum_k s_k\, q_k^{T^*}(x)$ equals the Bayes-optimal point prediction $\mathbb{E}[y \mid x]$, and for any $T \neq T^*$ the conditional MSE increases by the squared bias:

$$\mathbb{E}\big[(\hat{y}^T - y)^2 \mid x\big] = \mathrm{Var}(y \mid x) + \big(\hat{y}^T(x) - \hat{y}^{T^*}(x)\big)^2. \tag{22}$$

Cell	Full-6 $\bar{\rho}$	Strong-3 $\bar{\rho}$
Average (10 cells)	0.7775	0.7071
SummEval 1–10	0.7611	0.5860
TopicalChat 1–5	0.8705	0.7875
TopicalChat 1–10	0.9467	0.9150
WMT MQM 1–10	0.5883	0.5696

Table 20: Mean off-diagonal residual correlation ρ among judge errors, computed from normalized soft scores relative to human labels. The large correlations on TopicalChat imply that most remaining error is shared across judges, which is consistent with Proposition 3 and helps explain why fixed covariance-aware pooling is already strong there.

Cell	Best test family	Selected	Correct
ROSCOE 1–10	sparse	sparse	yes
ROSCOE 1–5	fixed	fixed	yes
SummEval 1–10	fixed	fixed	yes
SummEval 1–5	sparse	fixed	no
TopicalChat 1–10	fixed	fixed	yes
TopicalChat 1–5	fixed	fixed	yes
WMT MQM 1–10	full	full	yes
WMT MQM 1–5	sparse	full	no

Table 21: Leave-one-cell-out diagnostic for the deployment heuristic. We perform a coarse threshold sweep over ρ, leave-one-out gain, and the fixed-vs-full margin on the eight direct-score cells with held-out validation and test splits. All eight folds recover the same selector: choose sparse when the leave-one-out gain is at least 0.14; otherwise choose fixed when $\rho \geq 0.5$ and the fixed pool is already at least as strong as full JudgeMoE on validation; otherwise choose full. This rule matches the best test family on 6/8 cells.

Proof sketch. The equal-weight tempered log pool gives

$$\log q_k^T(x) = \frac{1}{T} \cdot \frac{1}{m} \sum_{j=1}^{m} \log p_k^{(j)}(x) + \text{const.}$$

Substituting (19),

$$\log p_k^{(j)}(x) = \tau_j^{-1} \log p_k^*(x) - \log Z_j(x).$$

The normalization terms are constant across k and vanish into the pooled normalizer, so

$$\log q_k^T(x) = \underbrace{\frac{1}{mT} \sum_{j=1}^{m} \frac{1}{\tau_j}}_{\gamma(T)} \log p_k^*(x) + \text{const.}$$

Writing $\sum_j \tau_j^{-1} = m/\bar{\tau}_H$ yields (20); setting $\gamma(T) = 1$ gives $T^* = 1/\bar{\tau}_H$. The MSE identity then follows from Proposition 1 and the standard bias–variance decomposition around $\mathbb{E}[y \mid x]$.

Strategy	Avg. test Sp.	Weighted test Sp.
Best single (valid-selected)	0.2748	0.1860
Always fixed pool	0.3374	0.1319
Always full JudgeMoE	0.2975	0.1655
Always sparse variant	0.3497	0.1536
Best family by validation performance	0.3402	0.1650
LOO heuristic selector	0.3495	0.1694
Best family by test performance	0.3639	0.1701

Table 22: Validation-only strategy selection on the eight direct-score cells with held-out validation and test splits. Thresholds are selected on the remaining cells and then applied to the held-out test cell. The resulting selector improves over always using full JudgeMoE and over always using the strongest fixed pool on equal-cell average test Spearman, while sample-weighted averages remain dominated by the large WMT cells and therefore stay close to the best family by test performance.

Strategy	Avg. test Sp.	Weighted test Sp.
Best single (valid-selected)	0.2748	0.1860
Full fixed pool	0.3374	0.1319
Full JudgeMoE	0.3154	0.1665
Valid-selected subset fixed	0.3133	0.2113
Valid-selected family	0.3732	0.2213
Best family by test performance	0.3789	0.2215

Table 23: Validation-only subset selection on the eight held-out direct-score cells. The best fixed subset uses 3.38 judges on average and is smaller than the full six-judge pool in all eight cells. A validation-selected family over best single, full fixed pooling, full JudgeMoE, and subset-fixed pooling nearly matches the best family selected using test performance.

Interpretation. Proposition 6 turns the pooling temperature into a calibration diagnostic as well as an empirical hyperparameter. The empirical preference for T=0.7 in Table 12 is therefore consistent with a collectively under-confident judge pool.

Scope. The temperature-scaled model above is a first-order approximation. Real judges exhibit instance-dependent miscalibration, so Proposition 6 characterizes the pool-average calibration condition without predicting one exact optimal T for every example. We use it only as a directional diagnostic for the temperature sweep in Table 12. Its practical use is to explain why the best temperature often lies below one and to motivate initialization of the temperature sweep from a small calibration set.

Proposition 7 (Distribution–score information gap). Let $\mathbf{P} = (p^{(1)}, \dots, p^{(m)}) \in (\Delta^{K-1})^m$ be the distributional evidence from m judges over $K \geq 3$ ordered scores, and let $\boldsymbol{\mu} = (\mu_1, \dots, \mu_m)$ with $\mu_j = \sum_k s_k \, p_k^{(j)}$ be the corresponding score vector of expected scores.

Validation signal	Recommended choice
High residual correlation; covariance-aware pooling already strong	Fixed covariance-aware pooling
Large leave-one-out gain or large subset/full-pool gap	Subset selection + retraining
Remaining gain from example-specific weighting	Full JudgeMoE

Table 24: Validation signals and corresponding deployment choices. Appendix Table 21 reports the leave-one-cell-out evaluation.

Here $\sigma(\mathbf{P})$ denotes the information contained in the full collection of judge distributions, while $\sigma(\boldsymbol{\mu})$ denotes the coarser information retained after each distribution is reduced to its expected score. Thus $\sigma(\boldsymbol{\mu}) \subseteq \sigma(\mathbf{P})$ simply formalizes the fact that expected scores are deterministic summaries of the full distributions.

(i) The Bayes-optimal predictors $f^*(\boldsymbol{\mu}) = \mathbb{E}[y \mid \boldsymbol{\mu}]$ and $g^*(\mathbf{P}) = \mathbb{E}[y \mid \mathbf{P}]$ satisfy

$$\begin{aligned} &\underbrace{\mathbb{E}\big[(y - f^*(\boldsymbol{\mu}))^2\big]}_{\text{score risk}} - \underbrace{\mathbb{E}\big[(y - g^*(\mathbf{P}))^2\big]}_{\text{distribution risk}} \\ &\quad = \mathbb{E}\Big[\mathrm{Var}\big(g^*(\mathbf{P}) \mid \boldsymbol{\mu}\big)\Big] \geq 0. \end{aligned} \tag{23}$$

(ii) For any auxiliary feature $\phi(\mathbf{P})$ with information beyond $\boldsymbol{\mu}$, the gap admits the lower bound

$$\Delta_{\text{info}} \geq \mathbb{E}_{\boldsymbol{\mu}}\left[\frac{\big(\mathrm{Cov}(y, \phi(\mathbf{P}) \mid \boldsymbol{\mu})\big)^2}{\mathrm{Var}(\phi(\mathbf{P}) \mid \boldsymbol{\mu})}\right]. \tag{24}$$

(iii) When $K = 2$, $\sigma(\boldsymbol{\mu}) = \sigma(\mathbf{P})$ and the gap is exactly zero. When $K \geq 3$ and judge distributions are generically non-degenerate, $\sigma(\boldsymbol{\mu}) \subsetneq \sigma(\mathbf{P})$ and the gap is generically strict.

Proof sketch. Part (i) is the law of total variance applied to $\sigma(\boldsymbol{\mu}) \subseteq \sigma(\mathbf{P})$:

$$\begin{aligned} \mathrm{Var}(y \mid \boldsymbol{\mu}) = \mathbb{E}\big[\mathrm{Var}(y \mid \mathbf{P}) \mid \boldsymbol{\mu}\big] \\ + \mathrm{Var}\big(\mathbb{E}[y \mid \mathbf{P}] \mid \boldsymbol{\mu}\big). \end{aligned}$$

Taking expectations yields (23). For part (ii), conditional Cauchy–Schwarz gives

$$\mathrm{Var}\big(g^*(\mathbf{P}) \mid \boldsymbol{\mu}\big) \geq \frac{\big(\mathrm{Cov}(g^*(\mathbf{P}), \phi(\mathbf{P}) \mid \boldsymbol{\mu})\big)^2}{\mathrm{Var}(\phi(\mathbf{P}) \mid \boldsymbol{\mu})},$$

and the tower property identifies the covariance term with $\mathrm{Cov}(y, \phi(\mathbf{P}) \mid \boldsymbol{\mu})$. Part (iii) follows

Stage	Validation signal	Selection
Protocol	Scale / extraction differences	Select protocol
Family	Fixed / full / single performance	Select aggregation family
Subset	Subset performance	Select judge subset
Subset retraining	Subset / full-pool difference	Retrain on selected subset

Table 25: Validation procedure for protocol, aggregation-family, and judge-subset selection. Supporting results are reported in Appendix Tables 15, 21, 22, and 23, and in Table 53.

Subset	Weighted Sp.	Δ vs. single	Δ vs. cov-log
All 10 cells	0.3052	0.0413	0.0549
Exclude ROSCOE	0.3052	0.0413	0.0548
Exclude WMT	0.5393	0.1346	0.0961
Non-HelpSteer	0.1549	-0.0241	0.0236

Table 26: Sample-weighted summaries for the mainline 6-judge JudgeMoE system. On the full 10-cell suite, the weighted bootstrap 95% interval is $[-0.0630, 0.1399]$ against the best single judge and $[0.0109, 0.1014]$ against covariance-aware pooling. Excluding ROSCOE leaves the weighted means essentially unchanged.

because for $K = 2$ the simplex is one-dimensional, so the map from distribution to expected score is affine and invertible, whereas for $K \geq 3$ it has a non-trivial kernel inside Δ^{K-1}.

Interpretation. Proposition 7 formalizes the paper's central formulation choice. Distributional aggregation has a genuine information advantage whenever judge distributions vary in ways that are unrecoverable from their expected scores alone. This is the setting in which entropy, confidence, and near-tie structure become useful.

Example. For scores $\{1, 2, 3\}$, the distributions $(0.5, 0, 0.5)$ and $(0, 1, 0)$ have the same expected score, 2, while encoding very different uncertainty: the first is a polarized near-tie between low and high scores, while the second is concentrated on the middle score. A score-only aggregator treats them the same; a distributional aggregator can distinguish them.

Concrete instantiation. Taking $\phi(\mathbf{P})$ to be mean judge entropy, confidence, or maximum-bucket probability yields a computable lower bound on the score-to-distribution gap. Appendix Table 18 provides the corresponding empirical signal in our setting: moving from distributions alone to the full scalar summary bundle raises average Spearman from 0.3746 to 0.3832, suggesting that these extra distributional statistics carry useful information beyond the expected scores.

Scope. The lower bound in (24) is intentionally general. Its role is diagnostic: it says what kind of auxiliary distributional statistic can certify a non-trivial information gap, while making no claim that any single statistic captures the full gain.

Proposition 8 (Sparse improvement condition for expected-score MSE). Under the calibration model (19) with equal weights and pooling temperature T, define judge precision $r_j = \tau_j^{-1}$, mean pool precision $\bar{r} = \frac{1}{m}\sum_{j=1}^{m} r_j$, pool precision excess $\Delta = m(\bar{r} - T)$, and judge precision deviation $\delta_0 = r_0 - T$ for the candidate judge j_0 to be dropped. The effective exponent of the full pool is $\gamma_{\text{full}} = \bar{r}/T$, and dropping j_0 yields $\gamma_{-0} = (m\bar{r} - r_0)/((m-1)T)$.

Dropping judge j_0 moves the effective exponent strictly closer to the optimum $\gamma^* = 1$ (equivalently, $|\gamma_{-0} - 1| < |\gamma_{\text{full}} - 1|$) if and only if

$$\begin{aligned} \delta_0 &\in \left(\tfrac{\Delta}{m}, \tfrac{(2m-1)\Delta}{m}\right) \quad \text{when } \Delta > 0, \\ \delta_0 &\in \left(\tfrac{(2m-1)\Delta}{m}, \tfrac{\Delta}{m}\right) \quad \text{when } \Delta < 0. \end{aligned} \tag{25}$$

Equivalently, δ_0 lies strictly between $\bar{r} - T$ and $(2m-1)(\bar{r} - T)$. When $\Delta = 0$ (the pool is already optimally sharpened), dropping any judge can only hurt or leave the exponent unchanged.

Proof sketch. Substitute the effective exponents into the inequality $|\gamma_{-0} - 1| < |\gamma_{\text{full}} - 1|$:

$$\gamma_{\text{full}} - 1 = \frac{\Delta}{mT}, \qquad \gamma_{-0} - 1 = \frac{\Delta - \delta_0}{(m-1)T}.$$

This reduces the problem to

$$\frac{|\Delta - \delta_0|}{m-1} < \frac{|\Delta|}{m},$$

which becomes a quadratic inequality in δ_0. Solving that inequality yields exactly the interval in (25).

Corollary 1 (Weak judges hurt under-sharpened pools). When the pool is under-sharpened ($\Delta <$

Cell	n	JudgeMoE	Best single	Δ	95% CI	p
helpsteer2_train@1-10	101620	0.5386	0.3849	0.1537†	[0.1502, 0.1572]	0.0100
helpsteer2_train@1-5	101620	0.5426	0.4087	0.1339†	[0.1303, 0.1373]	0.0100
roscoe_esnli@1-10	151	0.2041	0.0773	0.1268	[-0.0240, 0.2675]	0.1294
roscoe_esnli@1-5	151	0.2870	0.2641	0.0228	[-0.0063, 0.0465]	0.1791
summeval@1-10	6400	0.6101	0.5657	0.0444†	[0.0260, 0.0653]	0.0100
summeval@1-5	6400	0.5134	0.4687	0.0448†	[0.0319, 0.0591]	0.0100
topical_chat@1-10	2160	0.4270	0.4458	-0.0188	[-0.0479,0.0095]	0.1916
topical_chat@1-5	2160	0.4357	0.4683	-0.0325	[-0.0566,-0.0094]	0.0074
wmt_mqm@1-10	150347	0.2293	0.1797	0.0495†	[0.0454, 0.0524]	0.0100
wmt_mqm@1-5	150347	0.0376	0.1413	-0.1037	[-0.1091, -0.0991]	0.0100

Positive Δ in 7/10 cells; significant positive Δ in 5/10 cells. $^{\dagger}p < 0.05$ and CI excludes 0.

Table 27: Per-cell paired bootstrap significance for the main 10-cell aggregation comparison against the best single judge, using the exact per-example predictions of the reported mainline checkpoint. This supplements the aggregate 10-cell Wilcoxon test by showing which benchmark cells individually support the positive JudgeMoE trend.

System	ECE ↓	MAE ↓
JudgeMoE	**0.0588**	**0.1890**
Best single	0.2120	0.2782
Uniform log	0.1785	0.2718
Cov-log	0.2214	0.2976

Table 28: Continuous-score calibration check on the original 10 direct-score cells. ECE uses 10 equal-width bins over normalized predictions; within each bin we compute the absolute gap between mean prediction and mean human score, then average these gaps by bin frequency. Values are equal-cell averages, matching the main Spearman aggregation.

Dataset	JudgeMoE	Best single	Uniform log	Cov-log
SummEval	0.0661	0.3840	0.1621	0.2210
TopicalChat	0.0559	0.2640	0.1385	0.2118
ROSCOE	0.1053	0.2161	0.2693	0.3381
HelpSteer2	0.0133	0.0483	0.0859	0.0791
WMT MQM	0.0534	0.1476	0.2368	0.2568

Table 29: Per-dataset ECE breakdown, averaged over the two main scales for each dataset. The aggregate calibration gain spans the full benchmark set: JudgeMoE has the lowest ECE on all five dataset averages in this diagnostic.

0), a very weak judge has $r_0 \approx 0$ and therefore $\delta_0 \approx -T$. In this condition, dropping such a judge moves the pooled exponent closer to the optimum and improves expected-score aggregation. This matches the WMT MQM result in Table 53: removing Qwen2.5-7B raises the full-pool baseline from 0.0016 to 0.0389 at the 1–5 scale.

Corollary 2 (Weak judges can help oversharpened pools). When the pool is oversharpened ($\Delta > 0$), a weak judge may provide useful flattening; dropping it can fail to help be-

Comparison	Mean Δ	SD$_{\Delta}$	Power @10	Cells for 80%
vs. uniform log	+0.0794	0.0692	0.98	5
vs. cov-log	+0.0427	0.0729	0.58	19
vs. best single	+0.0419	0.0812	0.49	24

Table 30: First-order power note for the original 10-cell paired comparisons. We use the observed cell-level mean and standard deviation of the paired deltas with a one-sided α=0.05 normal approximation. The table explains why the 10-cell benchmark is sufficient for the uniform-log control but underpowered for the smaller gaps against covariance-aware pooling and the best single judge.

cause the pool already places too much concentrated mass on a small set of score buckets.

Interpretation. Proposition 8 reveals an asymmetry that is invisible in the mode-level veto analysis of Proposition 4. Weak judges are reliably harmful when the pool is already too flat, but can be harmless or even useful when the pool is too sharp. Sparse repair is therefore a calibration-aware intervention, with no universal preference for smaller pools.

Scope. The equal-weight analysis in Proposition 8 characterizes the fixed-pool baseline that JudgeMoE's learned gate improves upon. Its role is to identify which judges a fixed pool excludes; the learned gate can then adaptively reweight the surviving pool. Appendix Table 52 gives the matching empirical check: on WMT, the full gate already assigns negligible weight to the weak Qwen2.5 judge, but sparse retraining is still needed at 1–5 because coarse scores leave too little coherent ranking signal in the contaminated full pool.

Component	Selection setup	Reported use
Protocol study	Five score ranges with hard/soft extraction; validation-based protocol selection over 24 cells	Select scoring protocol
Main architecture	Five architecture/pooling variants, temperature sweep, and feature ablation on the original 10 cells	Select 6-judge DeepSets + log pooling
Compact 3-judge pool	Candidate triples from protocol-study screening	Efficiency setting
WMT subset selection	Top-k subsets and leave-one-out selection on WMT validation data	Retrain selected subsets
General subset selector	All 63 nonempty judge subsets over eight validation/test cells	Select judge subset and aggregation family
Heterogeneous experts	One task-aligned expert added at a time using task-specific train or validation data	Evaluate heterogeneous pools and standalone experts

Table 31: Selection procedures for the main experiment groups. The 6-judge architecture selected on the original 10 cells is reused for the HelpSteer2 validation, FLASK, WildBench, and heterogeneous evaluations.

Cell	JudgeMoE	Single	Cov-log	Δ single	Δ cov-log
HelpSteer2 val.@1–5	0.4394	0.3851	0.4212	+0.0543	+0.0182
HelpSteer2 val.@1–10	0.4562	0.3805	0.4358	+0.0756	+0.0204
12-cell avg.	0.3940	0.3483	0.3552	+0.0457	+0.0388

Table 32: Held-out HelpSteer2-validation extension of the selected mainline checkpoint. These two cells were added only after the original 10-cell architecture sweep, so they evaluate the frozen mainline checkpoint without rerunning architecture search. Unlike the FLASK support check, both new HelpSteer2-validation cells are positive against the best single judge and covariance-aware log pooling. The table reports the two new cells together with the resulting 12-cell averages; the corresponding one-sided paired Wilcoxon tests are p=0.0212 against the best single judge and p=0.0461 against covariance-aware log pooling.

Connecting the results. Propositions 6–8 form a coherent diagnostic chain. Proposition 6 diagnoses the pool-level calibration condition and interprets the best temperature. Proposition 7 justifies why JudgeMoE works with score distributions in addition to compressed scores, and identifies the $K \geq 3$ condition under which that advantage can exist. Proposition 8 then characterizes when removing a judge improves the pool, conditional on the calibration condition identified by Proposition 6. Together, they support the validation-guided deployment heuristic in Table 4: study calibration first, keep distributional evidence whenever the score range is fine enough to support it, and use sparse repair only when the pool diagnostics indicate contamination.

Scope. Propositions 1, 3, and 4 are standard background facts stated in compact form; Propositions 6–8 are specific to the JudgeMoE formulation and serve as structured diagnostic results with limited scope. None of them imply that several small judges must beat a larger single judge. Their role is more specific: to explain why soft scores help, why temperature tuning is interpretable, why distributional fusion can add information beyond score compression, and why sparse repair helps only in particular contamination conditions.

Cell	JudgeMoE	Single	Cov-log	Δ single	Δ cov-log
FLASK@1–5	0.5452	0.4897	0.5452	+0.0555	-0.0001
FLASK@1–10	0.4364	0.4148	0.4449	+0.0216	-0.0085
12-cell avg.	0.4011	0.3598	0.3663	+0.0413	+0.0349

Table 33: Secondary FLASK support check for the selected mainline checkpoint. These two cells were added only after the original 10-cell architecture sweep, so they evaluate the frozen mainline checkpoint without rerunning architecture search. FLASK@1–5 ties covariance-aware log pooling up to rounding, consistent with the mixed fixed-pool picture. The table reports the two new FLASK cells together with the resulting 12-cell averages; the corresponding one-sided paired Wilcoxon tests are p=0.0261 against the best single judge and p=0.1018 against covariance-aware log pooling. We use FLASK mainly as out-of-domain support for the best-single trend; HelpSteer2 validation remains the primary fixed-pool expansion check.

Setting	Comparison	Left	Right	Δ	95% CI	p
SummEval 1–10 + BERTScore	Het. JudgeMoE vs. best single	0.6703	0.5172	+0.1531	[0.1104, 0.1944]	0.0007
SummEval 1–10 + BERTScore	Het. JudgeMoE vs. uniform log	0.6703	0.1930	+0.4773	[0.4219, 0.5320]	0.0007
SummEval 1–10 + BERTScore	Het. JudgeMoE vs. hom. JudgeMoE	0.6703	0.6460	+0.0242	[-0.0026, 0.0509]	0.0733
HelpSteer2 1–10 + reward	Het. JudgeMoE vs. best single	0.5454	0.3900	+0.1553	[0.1468, 0.1644]	0.0025
HelpSteer2 1–10 + reward	Het. JudgeMoE vs. uniform log	0.5454	0.3102	+0.2352	[0.2239, 0.2472]	0.0025
HelpSteer2 1–10 + reward	Het. JudgeMoE vs. hom. JudgeMoE	0.5454	0.5330	+0.0124	[0.0066, 0.0189]	0.0025

Table 34: Paired bootstrap significance for the strongest heterogeneous follow-ups. "Left" and "Right" are Spearman correlations; Δ is left minus right. Intervals and p-values come from paired bootstrap over matched examples.

Pool	Best single	Uniform log	Cov-log	JudgeMoE
3-judge LLM pool	0.1898	0.2031	0.1988	**0.2468**
3-judge LLM + COMET	**0.4326**	0.3926	0.3794	0.1620

Table 35: WMT MQM@1–10 follow-up with COMET as a task-aligned heterogeneous expert. COMET is a much stronger standalone evaluator than any constituent LLM judge, but appending it to the mixed pool degrades JudgeMoE sharply.

Setting	Best single	Full JudgeMoE	Δ vs. single	min frac > single	min frac ≥ 90% gain	min frac ≥ 95% gain
SummEval 1–10 + BERTScore (repaired)	0.5172	0.6682	0.1510	0.05	0.40	1.00
HelpSteer2 1–10 + reward	0.3900	0.5670	0.1769	0.05	0.70	0.70

Table 36: Sample-efficiency summary for the two strongest heterogeneous positives. Fractions refer to the portion of the training split used to fit JudgeMoE; the gain-recovery columns normalize against the full-data gain over the best single judge. Both settings beat the best single judge with only 5% of training data, but the reward-model append saturates more slowly than the BERTScore expert.

Setting	Expert	mean Δ vs. hom. abs.-err.	mean Δ vs. single abs.-err.	top-gate rate	Role
SummEval 1–10 + BERTScore	BERTScore	0.0744	0.3167	0.1203	frequent expert
HelpSteer2 1–10 + reward	Skywork reward	0.0031	0.0321	0.0000	soft calibration

Table 37: Compact qualitative summary for the strongest heterogeneous positives. The top-gate rate measures how often the added expert becomes the highest-weight judge on a test example. BERTScore acts like a genuine conditional expert on SummEval, whereas the reward model on HelpSteer2 behaves more like a soft calibration signal.

Setting	Strong single	Best single	Hom. JudgeMoE	Het. JudgeMoE	Δ Het.	Δ Hom.
SummEval@1–10	Qwen2.5-72B	0.5172	0.6460	0.6703	0.1779	0.1536
HelpSteer2@1–10	Qwen2.5-72B	0.3900	0.5330	0.5454	0.0716	0.0593
WildBench@1–10	Qwen2.5-72B	0.3520	0.1719	0.0854	0.0822	0.1688

Table 38: Aligned strong single-judge follow-ups using Qwen2.5-72B-Instruct on the same evaluation splits as the heterogeneous pilots. JudgeMoE remains stronger on the two positive follow-up cells, while WildBench remains difficult for all methods.

Setting	Groups	Avg. cand.	Best group score	Het.	Hom.	Single	Δ Het.
summeval best-of-N	100	9.3200	0.9979	0.9365	0.9446	0.9194	-0.0081

Table 39: Offline best-of-N SummEval case study, grouping candidate summaries by source article. The repaired heterogeneous pool still beats the best single judge, while the repaired homogeneous pool remains stronger on this reranking-style proxy.

Component	Setting
Direct-score target	Human scores normalized to $[0, 1]$
Direct loss	Huber loss on the fused score
Pairwise target	Binary preference converted to $\{-1, +1\}$
Pairwise loss	Margin-ranking loss with tuned auxiliary weight and fixed margin m=0.0
Large sweeps	JSONL-based resume support for interrupted runs
Subset comparisons	Exact example-ID overlap for sample-aligned evaluation

Table 40: Implementation details relevant to reproduction.

Setting	Mean Sp.	Std.	Δ vs. single
Mainline 6-judge	0.3826	0.0013	0.0413
Efficient 3-judge	0.3638	0.0028	0.0245

Table 41: Seed stability for the strongest 6-judge and 3-judge settings. The small standard deviations show that the two reported deployment settings are stable across random seeds.

λ	Direct Sp.	Δ pair acc.
0.1	0.3725	0.0360
0.5	**0.4192**	**0.0462**
1.0	0.4190	0.0431

Table 42: Joint training with pairwise auxiliary loss in the separate 3-judge pairwise-ablation setting (MT-Bench-human and Summarize-from-Feedback validation, each at 1–5 and 1–10). The direct Spearman values summarize this 4-row setting, distinct from the 10-cell 6-judge mainline benchmark. Moderate pairwise weight improves pairwise accuracy without hurting direct-score quality.

Scale	Dataset	Base acc.	Joint acc.
1–5	MT-Bench-human	0.6369	0.6823
1–5	Summarize-from-Feedback	0.5140	0.6054
1–10	MT-Bench-human	0.7200	0.7250
1–10	Summarize-from-Feedback	0.5994	0.6244

Table 43: Per-dataset pairwise gains for the best joint-training setting. The joint model improves on both pairwise datasets at both scales, which is why the paper treats pairwise supervision as a useful auxiliary signal for direct-score training.

Scale	Best single	Uniform log	Direct	Joint	Pair-only
1–5	0.5482	0.5385	0.5411	**0.5495**	0.5112
1–10	**0.5674**	0.5578	0.5601	0.5507	0.5026

Table 44: Supplementary OOD evaluation on 3,114 English non-tie Arena human preferences. The pattern is consistent with the main pairwise story: joint training helps at 1–5, pairwise-only remains weak at both scales, and at 1–10 the strongest single judge is already difficult to improve upon.

Scale	Best single	Uniform log	Direct	Joint	Pair-only
1–5	0.4870	0.4479	**0.4917**	0.4656	0.4562
1–10	**0.5201**	0.4870	0.5036	0.5012	0.5130

Table 45: Supplementary OOD evaluation on a more recent, length-controlled Arena Expert subset with 844 English A/B comparisons. The benchmark is filtered to contexts that fit the local three-judge pool without OOM failures (‘source $\leq$ 3000‘ chars; responses $\leq$ 6000 chars). The result is more conservative than Arena Human Preference: direct-only is the strongest learned setting at 1–5, while at 1–10 the best single judge remains strongest.

Scale	Best single	Uniform log	Direct	Joint	Pair-only
1–5	**0.5605**	0.5446	0.5361	0.5259	0.4952
1–10	**0.5821**	0.5685	**0.5821**	0.5657	0.4832

Table 46: Supplementary OOD evaluation on a recent length-controlled PPE-Human-Preference-V1 subset with 1,759 English A/B comparisons. The benchmark is filtered to ‘source $\leq$ 4000‘ chars and responses ‘$\leq$ 8000‘ chars. The result is more conservative than Arena Human Preference: at 1–5 the best single judge remains strongest, and at 1–10 direct-only only matches the best single judge while joint training no longer helps.

Scale	Best single	Uniform log	Cov-log	JudgeMoE	Hard
1–5	0.2285	0.2183	**0.2638**	0.1955	0.0804
1–10	0.2351	0.2650	0.2589	**0.2709**	-0.0157

Table 47: Supplementary direct-score evaluation on a modern open-ended WildBench v2-hard subset. We use the released train split, take the provided GPT-4 reference response as the candidate answer, and keep a length-controlled subset of 246 examples (‘prompt $\leq$ 6000‘ chars; candidate ‘$\leq$ 8000‘ chars). The pattern remains conditional: covariance-aware log pooling is strongest at 1–5, while the 6-judge JudgeMoE mainline becomes strongest at 1–10.

Standalone evaluator	Cells	Standalone	JudgeMoE	JMoE wins
DeepSeek-V3.2	4	0.5710	0.4966	1/4
Qwen2.5-72B	4	0.5362	0.4966	1/4

Table 48: Aligned comparisons with larger standalone evaluators on the same evaluation subsets. These evaluators are included as standalone deployment options in the validation analysis.

System	Active params / token	Relative compute
Local single judge	7–12B dense	0.19–0.32×
Local 3-judge efficient pool	28B dense across separate passes	0.76×
Local 6-judge mainline pool	53B dense across separate passes	1.43×
DeepSeek-V3.2-Exp	37B activated, 671B total	1.00×

Table 49: Hardware-independent first-order inference-compute proxy. For autoregressive inference, weight-matrix FLOPs scale approximately linearly with active parameters and processed tokens, so under the same evaluation workload the relative compute is well approximated by active-parameter counts per token. Under the 1K-example SummEval 1–5 workload used in our deployment note (roughly 474K processed tokens), this proxy corresponds to about 26.5 PFLOPs for the local 3-judge pool, 50.2 PFLOPs for the local 6-judge pool, and 35.1 PFLOPs for DeepSeek-V3.2-Exp. The six local judges are independent and can be run in parallel; in the same probe, single-judge jobs took 16.8–27.4 minutes, so ideal six-way parallel latency is bounded by the slowest job. The sequential sum was 134.4 minutes.

Type	Cell	Example	Human	JudgeMoE	Cov-log	Top gate	Judge scores	Gate weights
+	summeval@1-5	`S1499-relevance`	0.833	0.825	0.344	L3	[.11,1,.50,.32,.01,.43]	[.08,.25,.35,.15,.04,.13]
+	helpsteer2@1-10	`H10914-helpfulness`	1.000	0.901	0.524	G3	[.57,.90,.52,.50,.50,.55]	[0,.98,0,.01,0,0]
+	wmt-mqm@1-10	`W66502`	0.800	0.798	0.564	G2	[.77,.64,.54,.57,.52,.56]	[1,0,0,0,0,0]
+	roscoe-esnli@1-10	`R30`	1.000	0.822	0.580	G3	[.56,.90,.50,.54,.49,.60]	[.18,.41,0,.17,0,.23]
-	topical-chat@1-5	`T12-Engaging`	0.333	0.898	0.390	G3	[.39,1,.51,.39,.42,.34]	[.10,.31,.15,.03,.16,.24]

Table 50: Illustrative per-example cases where learned aggregation improves over (+) or fails against (-) fixed cov-log pooling. Judge-score and gate-weight vectors use the order G2/G3/L3/Min/Q2.5/Q3.5. Examples are selected automatically by the largest absolute-error change in the indicated direction. Across the 10 direct-score cells, the JudgeMoE prediction is closer to the human score than cov-log in 75.5% of examples on an equal-cell basis (84.6% example-weighted).

Scale	Cand.	ρ_s(valid,test)	Sel. valid	Sel. test	Test rank
1-5	120	0.973	0.143	0.153	1
1-10	120	0.974	0.213	0.232	3

Table 51: Validation–test stability for the WMT sparse fixed-pool search. Candidates include all single judges and all non-single subsets under uniform-log and cov-log pooling. The selected candidate is chosen by validation Spearman; its test rank is computed among the same candidates by test Spearman.

Scale	Judge	Judge Sp.	MAE	Mean gate	Top-gate	Diagnostic readout
1-5	G2	0.141	0.186	0.373	46.1%	best single; largest gate mass
1-5	G3	0.019	0.256	0.244	21.7%	secondary evidence
1-5	L3	-0.030	0.269	0.221	19.2%	secondary evidence
1-5	Min	0.086	0.299	0.140	12.7%	secondary evidence
1-5	Q2.5	-0.148	0.651	0.006	0.3%	known contaminant; gate almost removes it
1-5	Q3.5	0.014	0.400	0.017	0.0%	secondary evidence
1-10	G2	0.180	0.111	0.529	69.5%	best single; dominant route
1-10	G3	0.128	0.122	0.120	3.8%	secondary evidence
1-10	L3	-0.086	0.319	0.000	0.0%	negative single; gate suppresses
1-10	Min	0.122	0.163	0.117	12.4%	secondary evidence
1-10	Q2.5	0.037	0.292	0.054	0.5%	secondary evidence
1-10	Q3.5	0.139	0.195	0.179	13.8%	secondary evidence

Table 52: WMT per-judge gate behavior connecting sparse-repair theory to the learned gate. Judge Spearman/MAE provide an empirical precision proxy; mean gate and top-gate rates are computed from the learned full-pool gate. Rank correlation between per-judge Spearman and mean gate is 1-5: 0.71, 1-10: 1.00. These correlations use only six judges and are diagnostic. The gate identifies weak judges, but at 1–5 sparse retraining is still needed because coarse scores leave too little coherent signal in the contaminated full pool.

Scale / subset	System or diagnostic	Spearman
1–5 (full)	Best single judge	0.1413
1–5 (full)	Top-2 sparse log pool	0.1473
1–5 (full)	Sparse JudgeMoE (2 selected judges)	**0.1648**
1–5 (full)	Full-6 uniform log pool	0.0016
1–5 (full)	Mainline JudgeMoE	0.0376
1–10 (full)	Best single judge	0.1797
1–10 (full)	Top-4 sparse log pool	0.2161
1–10 (full)	Sparse JudgeMoE (4 selected judges)	0.2246
1–10 (full)	Full-6 uniform log pool	0.1504
1–10 (full)	Mainline JudgeMoE	**0.2293**

Table 53: Deeper WMT MQM failure-and-recovery analysis. At 1–5, naive all-judge fusion degrades, but held-out-validation-selected sparse composition recovers the signal and a sparse JudgeMoE trained on the selected subset performs best. At 1–10, better score resolution already makes the full 6-judge JudgeMoE competitive, while the sparse 4-judge JudgeMoE retains most of that performance with a cleaner pool.