CORRELATION-AWARE RANKING AND FILTERING FOR HETEROGENEOUS SMALL-MODEL JUDGE ENSEMBLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Shared errors limit the benefits of combining small language model judges. Judge CIA (Correlation-Informed Agents) constructs heterogeneous same-backbone LoRA pools, ranks adapters by validation accuracy, and filters pairwise error correlation before majority voting. Our analysis separates individual competence from effective jury size in a majority-error upper bound, identifying conditions under which a smaller, less correlated jury has a tighter bound. Across 12 RAG-Bench backbone–task settings, CIA improves accuracy over the validation-selected Top-1 adapter by 1.99 percentage points on average (paired-bootstrap 95% interval $[0.57, 3.41]$, conditional on fixed pools and selections), with gains up to 7.00 points. In four illustrative matched-size comparisons, CIA shows point-estimate test-accuracy gains of 2.81–4.74 points over accuracy-only selection, accompanied by lower test error correlation. For high-stakes judging, a slice-risk decomposition motivates weighted ranking. In a constructed compliance study, weighting reduces mean critical-example error from 2.26% to 1.65% and overall error from 2.60% to 2.48%. Public class weighting reduces target-class error by 8.17 points, with increased non-target error.

1 INTRODUCTION

Adapters specialize small language models for labeled judging tasks. One can deploy the adapter with highest validation accuracy or combine the strongest adapters by majority vote. Individual accuracy, however, does not capture whether their errors overlap. Constructing a useful jury therefore requires identifying judges whose votes complement one another.

Shared backbones and training sources can produce common failures. Varying prompts, adapted modules, and training examples supplies different candidates; labeled error patterns identify complementary members for ordinary majority voting.

High-stakes judging adds a second selection question: which errors should receive priority? In compliance auditing, average accuracy can hide mistakes in operationally important categories. Shared errors across judges and errors concentrated in particular slices are complementary concerns. This motivates studying correlation-aware selection together with explicitly weighted validation objectives (Elkan, 2001).

Judge CIA varies prompt instructions, LoRA rank/scaling, adapted modules, and sampling seeds, then ranks adapters by validation accuracy and greedily filters redundant error patterns. Each accepted adapter must satisfy a pairwise error-correlation threshold against every earlier member. Members vote with equal weight. Validation determines label orientations and membership, which are frozen before test evaluation.

Our empirical contribution has three parts. First, the heterogeneous pool has lower mean validation error correlation than a seed-varying fixed-axis control in the Qwen3-1.7B/TAT-QA study (0.230 versus 0.330). Second, CIA achieves higher test accuracy than accuracy-only selection in four illustrative matched-size comparisons, with observed gains of 2.81–4.74 points and lower test error correlation. Third, in the constructed compliance study, slice-weighted ranking lowers mean critical-example error from 2.26% to 1.65%; public class weighting lowers target-class error by 8.17 points. Our analysis separates competence and redundancy in a majority-error bound, making explicit when fewer, less correlated judges yield a tighter bound.

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

Across four backbones and three RAGBench tasks (Friel et al., 2025), CIA gains 1.99 percentage points over Top-1, with eight wins, two ties, and two losses. Gemma-3-1B gains 5.60 points on TAT-QA and 7.00 on PubMedQA. The positive macro confidence interval supports an average improvement for the evaluated pools and selections.

Construction, size, threshold, and prefix analyses examine selection behavior; weighting studies report both target and non-target outcomes.

2 Related work

Judge CIA builds on classical ensemble learning, where gains depend on both individual competence and non-redundant errors (Dietterich, 2000; Geman et al., 1992). Prior work studies diversity measures such as pairwise correlation, disagreement, and double-fault statistics (Kuncheva & Whitaker, 2003), as well as pruning large ensembles into compact high-performing subsets (Margineantu & Dietterich, 1997). Judge CIA specializes these ensemble principles to trained small-model judge pools, combining competence-based feasibility filtering with labeled-error correlation screening.

LLM-based evaluators are widely used for scalable evaluation of generated text and assistant behavior (Liu et al., 2023; Zheng et al., 2023), and RAGBench provides explainable retrieval-augmented generation evaluation data (Friel et al., 2025). We focus on small, fine-tuned judge adapters rather than a single large judge. For high-stakes deployment, we also draw on cost-sensitive and slice-aware evaluation, where average accuracy may not match operational loss (Elkan, 2001).

Recent work directly addresses correlated judge panels. CARE models shared confounders and also studies validation-guided selection of generated programmatic judges (Zhao et al., 2026). JudgeBlender ensembles prompt/model relevance judgments (Rahmani et al., 2025), while RoPoLL uses robust geometric-median aggregation (Acharya et al., 2026). Judge CIA focuses on constructing heterogeneous, same-backbone LoRA judge pools and selecting competent, less redundant members using held-out labeled errors before ordinary majority voting. Our experiments examine this construction and selection procedure through within-pool comparisons of accuracy-only and correlation-aware juries.

3 Judge CIA: construction, selection, and voting

Judge CIA trains a heterogeneous adapter pool, selects members using validation errors, and combines their test predictions by equal-weight voting.

3.1 Heterogeneous adapter pool

For each backbone and task, we train $N = 252$ LoRA adapters (Hu et al., 2022). The pool crosses two prompt instructions, 18 rank/scaling pairs, and seven target-module sets. Adapter i uses sampling seed $500 + i$. Thus prompt wording, adapted projections, adapter capacity, and sampled training examples vary within a shared architecture. Appendix J.1 gives the exact instructions, module sets, and index mapping.

3.2 Validation statistics and label orientation

Let $V = \{(x^{(t)}, y^{(t)})\}_{t=1}^{m}$ be the validation set. The output parser returns `yes`, `no`, or `unknown`. Before selection, validation accuracy chooses between each adapter's original binary labels and a global yes/no inversion; ties retain the original orientation. Unknown outputs remain unknown. This orientation is frozen for test evaluation and is shared by Top-1, accuracy-only, random, and CIA comparisons.

For the oriented adapter A_i, define

$$z_i^{(t)} = \mathbf{1}\{A_i(x^{(t)}) \neq y^{(t)}\}, \qquad \hat{\varepsilon}_i = \frac{1}{m}\sum_{t=1}^{m} z_i^{(t)}, \qquad \hat{\rho}_{ij} = \mathrm{Corr}_t(z_i^{(t)}, z_j^{(t)}).$$

The last statistic is Pearson correlation of binary errors, or phi correlation (Kuncheva & Whitaker, 2003). Unknown outputs count as individual errors. A zero-variance pair is assigned correlation zero

for computation; no validation vector in the 12 public pools has zero variance. Selection correlations are rounded to six decimal places.

3.3 CORRELATION-AWARE SELECTION RULE

The candidate pool is $P_\tau = \{i : 1 - \hat{\varepsilon}_i \geq \tau\}$. Traverse this pool in increasing validation error, breaking ties by adapter index. Starting from $\widehat{\mathcal{K}} = \varnothing$, accept i exactly when

$$\hat{\rho}_{ij} \leq \rho \quad \text{for every } j \in \widehat{\mathcal{K}}.$$

The empty-set condition accepts the first candidate. Main results use the full feasible set, removing the last accepted adapter if necessary to make its size odd. Fixed-size diagnostics use odd prefixes of the accepted order. Algorithm 1 in Appendix G.1 gives full pseudocode.

The screen constrains every pair, not the mean correlation reported in the experiments. Traversal order matters because an early member may exclude later candidates; the greedy rule does not search all feasible subsets. Removing the last member preserves the remaining pairwise constraints.

We use $\tau = 0.5$ and $\rho = 0.4$ from the recorded protocol. The two statistics play distinct roles: the accuracy ordering prioritizes strong individuals, while the correlation threshold determines which can coexist. After label orientation, the 0.5 accuracy threshold is a weak screen: fully recognized binary outputs always reach at least 0.5 in the better validation orientation. The strict population competence margin in Section 4 therefore remains an analytical assumption.

3.4 VOTING AND SLICE-WEIGHTED RANKING

Each selected adapter receives one vote. The label with more recognized votes wins; unknown outputs abstain, and tied recognized counts are scored incorrect. Odd jury size does not exclude ties when abstentions occur.

High-stakes motivation. For a decision rule $\hat{y}$, partition a high-stakes subset into disjoint slices $\mathcal{H}_s$. The chain rule gives

$$\mathrm{CatRisk}(\hat{y}) = \sum_s \mathbb{P}(x \in \mathcal{H}_s)\, \kappa_s(\hat{y})\, \mathrm{Err}_s(\hat{y}), \tag{1}$$

where Err_s is slice-conditional error and κ_s is the probability that an error within that slice produces a specified severe consequence. Appendix B defines these quantities formally. Slice prevalence, error, and consequence probability all matter; weighted error alone need not track this risk.

We encode chosen priorities through validation weights $w_t \geq 0$ and rank by $r_i = m^{-1} \sum_t w_t z_i^{(t)}$. These weights specify error costs; they do not estimate κ_s. Correlation filtering addresses shared errors, while weighting addresses where errors occur. The exploratory studies in Section 5.7 test restricted variants: compliance uses plain top-k without correlation filtering, and public weighting re-ranks only within a correlation-feasible jury. Both retain equal voting weights.

4 HOW COMPETENCE AND REDUNDANCY GOVERN MAJORITY VOTING

For an idealized binary jury, let $Z_i = \mathbf{1}\{A_i(x) \neq f(x)\}$ and $S_\mathcal{K} = \sum_{i \in \mathcal{K}} Z_i$, with probabilities over $x \sim \mathcal{D}$ and any agent randomness. Majority error implies $S_\mathcal{K} \geq k/2$. Shared architecture and data motivate a dependence-aware variance diagnostic. Appendices A and C give the classical independence and ranking-stability results.

Assumption 4.1 (Bounded competence and bounded redundancy)**.** *For a selected jury $\mathcal{K}$ with $|\mathcal{K}| = k$, suppose each selected judge satisfies $\mathbb{P}(Z_i = 1) \leq \varepsilon_0 < 1/2 \quad (i \in \mathcal{K})$, and each selected pair satisfies $\mathrm{Corr}(Z_i, Z_j) \leq \phi \quad (i \neq j,\ i, j \in \mathcal{K})$, for some $0 \leq \phi \leq 1$.*

Proposition 4.2 (Diagnostic majority bound under bounded redundancy). *Let* $S_{\mathcal{K}} := \sum_{i \in \mathcal{K}} Z_i$ *and* $\Delta_0 := 1/2 - \varepsilon_0$. *Under Assumption 4.1,*

$$\mathrm{Var}(S_{\mathcal{K}}) \le k\varepsilon_0(1-\varepsilon_0)\big(1 + (k-1)\phi\big),$$

and Cantelli's inequality gives the diagnostic bound

$$\mathbb{P}(S_{\mathcal{K}} \ge k/2) \le \frac{\varepsilon_0(1-\varepsilon_0)\big(1+(k-1)\phi\big)}{\varepsilon_0(1-\varepsilon_0)\big(1+(k-1)\phi\big) + k\Delta_0^2} = \frac{\varepsilon_0(1-\varepsilon_0)}{\varepsilon_0(1-\varepsilon_0) + k_{\mathrm{eff}}(\phi)\Delta_0^2}$$

with $k_{\mathrm{eff}}(\phi) := \frac{k}{1+(k-1)\phi}$.

Why shared errors weaken voting. The proof starts from

$$\mathrm{Var}(S_{\mathcal{K}}) = \sum_{i \in \mathcal{K}} \mathrm{Var}(Z_i) + 2 \sum_{i<j,\ i,j \in \mathcal{K}} \mathrm{Cov}(Z_i, Z_j).$$

Each variance is at most $\varepsilon_0(1-\varepsilon_0)$ and each covariance at most $\phi\varepsilon_0(1-\varepsilon_0)$. The variance bound therefore has k individual terms but $k(k-1)$ correlation terms: individually accurate judges can still make redundant errors. Since $\mathbb{E}[S_{\mathcal{K}}] \le k\varepsilon_0$, majority failure requires an upward deviation of at least $k\Delta_0$. Cantelli's inequality bounds this event using the variance above. Appendix E gives the full proof.

Why jury size alone is insufficient. Dividing the variance bound by k^2 separates the contribution reduced by additional votes from the contribution due to shared errors:

$$\mathrm{Var}(S_{\mathcal{K}}/k) \le \varepsilon_0(1-\varepsilon_0) \underbrace{\left(\phi + \frac{1-\phi}{k}\right)}_{1/k_{\mathrm{eff}}(\phi)}.$$

When $\phi = 0$, $k_{\mathrm{eff}} = k$ and this variance bound decays as $1/k$. For fixed $\phi > 0$, the redundancy term persists and $k_{\mathrm{eff}} \to 1/\phi$. Adding judges therefore cannot remove the shared-error term from this bound. Reducing correlation directly lowers that term. This is a property of the upper bound, not a lower bound on actual error; the equicorrelated case is in Appendix D.

Corollary 4.3 (Comparing competence, redundancy, and size). *For* $0 < \varepsilon_0 < 1/2$, *define*

$$Q(\varepsilon_0, \phi, k) := \frac{(1/2-\varepsilon_0)^2}{\varepsilon_0(1-\varepsilon_0)} \frac{k}{1+(k-1)\phi}.$$

The majority-error upper bound in Proposition 4.2 is $B = (1+Q)^{-1}$. *Among juries satisfying its assumptions, a larger* Q *gives a smaller bound. At a common competence bound, a smaller jury has a tighter bound exactly when its effective size is larger.*

This follows by rewriting the proposition's denominator and using that $(1+Q)^{-1}$ decreases in Q. The first factor measures the competence margin relative to individual variance; the second discounts the number of votes for redundancy. Thus correlation reduction can offset fewer members or a weaker competence margin, provided the product increases.

A smaller jury can have a tighter bound. As an analytical illustration with a common $\varepsilon_0 = 0.25$, nine judges with correlation ceiling 0.10 have $k_{\mathrm{eff}} = 5$, whereas 25 judges with ceiling 0.30 have $k_{\mathrm{eff}} \simeq 3.05$. Their error upper bounds are 0.375 and approximately 0.496, respectively. These are hypothetical population parameters, not estimates from our experiments. The comparison shows why additional votes need not compensate for redundant errors.

Connection to CIA and the experiments. CIA's accuracy ordering prioritizes the competence factor in Q, while its pairwise screen targets redundancy. A stricter screen can lower correlation while reducing jury size, explaining why the two must be assessed jointly. Table 2 compares membership at fixed size; Table 3 varies both size and composition. The jury-size curve is retained in Appendix G.3. CIA does not directly optimize the population quantity Q; these experiments assess the design motivated by its two factors.

These are population assumptions. Empirical screens do not certify them: $\tau = 0.5$ after inversion does not establish $\varepsilon_0 < 1/2$, and mean observed correlation is not a ceiling on every pair. The bound is a conservative diagnostic, not a calibrated prediction or a guarantee of improvement over Top-1.

Table 1: Main performance across SLLMs and RAGBench datasets. Δ is Vote minus Top-1 in percentage points; positive values favor voting. The final column averages Δ across the three tasks. Vanilla is the unfine-tuned backbone under the recorded output protocol. Per-cell confidence intervals and the paired-bootstrap protocol appear in Appendices G.2 and I.5.

	FinQA				TAT-QA				PubMedQA				
Model	Vanilla	Top-1	Vote	Δ	Vanilla	Top-1	Vote	Δ	Vanilla	Top-1	Vote	Δ	Avg. Δ
Qwen3-0.6B	49.49	56.38	61.22	4.85	43.97	61.21	58.62	-2.59	16.00	61.20	61.20	0.00	0.75
Qwen3-1.7B	48.72	65.31	67.09	1.79	45.69	66.81	68.10	1.29	35.60	59.60	62.20	2.60	1.89
Llama-3.2-1B	54.08	57.40	57.40	0.00	53.02	63.36	62.50	-0.86	50.80	57.20	60.40	3.20	0.78
Gemma-3-1B	50.00	52.81	53.83	1.02	45.26	59.05	64.66	5.60	50.40	52.60	59.60	7.00	4.54
Average	50.57	57.97	59.89	1.91	46.99	62.61	63.47	0.86	38.20	57.65	60.85	3.20	1.99

5 EXPERIMENTS

We evaluate Qwen3-0.6B and Qwen3-1.7B (Yang et al., 2025), Llama-3.2-1B (Meta Llama, 2024), and Gemma-3-1B (Gemma Team et al., 2025): 12 settings against Top-1, four matched-size comparisons, and one construction study. Weighting experiments assess critical compliance examples and public adherence classes.

5.1 TASK, DATA, AND EVALUATION PROTOCOL

The public task is binary answer-adherence judgment on the RAGBench versions of FinQA, TAT-QA, and PubMedQA (Friel et al., 2025). Each judge receives retrieved documents, a question, and a candidate answer. The target is `yes` for a non-adherent answer and `no` for an adherent answer. Accuracy measures these adherence judgments.

We use the provided training, validation, and test splits. Each adapter trains for one epoch on at most 1000 sampled training examples. Validation and test labels are balanced before the final 500-item cap. The resulting validation/test counts are 246/392 for FinQA, 254/232 for TAT-QA, and 500/500 for PubMedQA; the final cap can leave residual class imbalance. All backbones use the same evaluation items within a task. Appendix G and Appendix J.1 specify training settings and exact prompts.

Validation fixes orientations, rankings, and membership before test scoring. Test labels also define explicitly post hoc diagnostics. Greedy generation uses at most four new tokens with a substring yes/no parser and the abstention rule of Section 3. Appendix H gives external single-backbone results under this protocol.

Uncertainty uses 10,000 paired item-bootstrap draws. Within each task, all four backbones and compared rules share resampled item indices; tasks are resampled independently. The macro statistic equally weights the 12 cells. Intervals condition on trained pools, orientations, and selections, excluding retraining and validation-selection variability. Per-cell intervals appear in Appendix G.2.

Output calibration. All comparators share validation-selected orientations. Undoing inversion for the fixed Gemma/TAT-QA jury changes accuracy from 64.66% to 53.02%, showing calibration dependence. Appendix J reports inversion, unknown-output, and tie counts. Caches retain parsed labels but no decoded text, leaving exact-label parser robustness unmeasured.

5.2 MAIN PERFORMANCE

Table 1 shows that CIA improves macro accuracy over Top-1 by 1.99 percentage points (paired-bootstrap 95% interval $[0.57, 3.41]$), supporting an average gain for the fixed pools and selections. CIA wins in eight settings, ties in two, and loses in two. Gains reach 7.00 points on Gemma-3-1B/PubMedQA and 5.60 on Gemma-3-1B/TAT-QA; Qwen3-0.6B gains 4.85 points on FinQA. Matched-size comparisons below assess selection at fixed jury size.

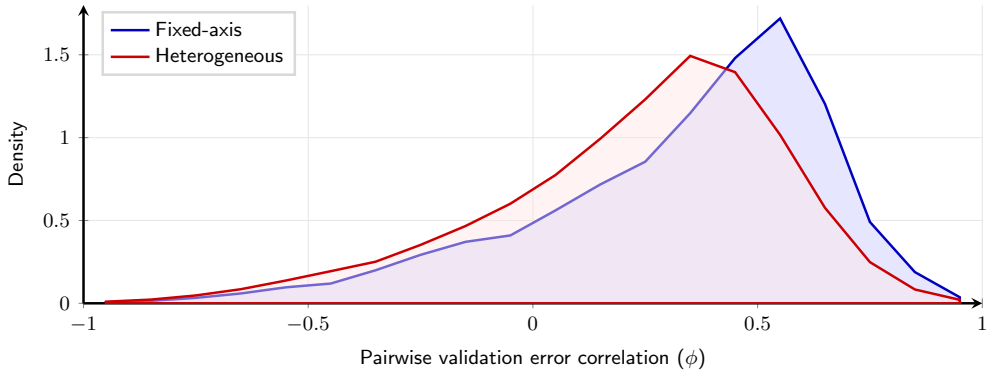


Figure 1: Pairwise validation error correlation on Qwen3-1.7B/TAT-QA. Fixed-axis varies the seed with prompt and LoRA settings fixed; heterogeneous varies all these choices. Mean phi is $0.330 \rightarrow 0.230$ and median phi is $0.409 \rightarrow 0.289$. Curves use the original histogram bins.

Table 2: Matched-size accuracy gains in four illustrative comparisons. Random is mean accuracy (SD); Acc. is accuracy-only top-k; Δ =CIA−Acc. Val./Test ϕ show mean error correlation, Acc.→CIA. Accuracy is in %; Δ is in percentage points.

Backbone / task	k	Random (SD)	Acc.	CIA	Δ	Val. ϕ	Test ϕ
Qwen3-0.6B / FinQA	9	58.33 (1.40)	57.40	61.22	+3.83	0.573→0.213	0.580→0.223
Qwen3-1.7B / FinQA	17	61.60 (1.86)	64.29	67.09	+2.81	0.438→0.066	0.455→0.080
Llama-3.2-1B / FinQA	3	55.36 (2.43)	54.59	57.40	+2.81	0.675→0.145	0.675→0.125
Gemma-3-1B / TAT-QA	23	57.84 (1.80)	59.91	64.66	+4.74	0.311→0.025	0.330→0.036

When voting loses to Top-1. On the 232 TAT-QA test items, the seven-member Qwen3-0.6B jury corrects 13 Top-1 errors but overturns 19 correct decisions. The nine-member Llama jury corrects 24 errors and overturns 26 correct decisions. The net six and two additional errors reproduce the -2.59 and -0.86 point differences. Both juries recover some errors, but incorrect overrides outweigh those gains: the validation correlation screen does not ensure an improvement over Top-1. Appendix I.2 gives the item-transition audit.

5.3 Heterogeneous construction and error correlation

Figure 1 compares heterogeneous and fixed-axis Qwen3-1.7B/TAT-QA pools; the control retains seed variation. The heterogeneous pool has lower mean validation phi (0.230 versus 0.330) and median phi (0.289 versus 0.409). The distributions still overlap broadly, motivating selection within the less redundant heterogeneous pool. Construction supplies varied candidates, and the pairwise screen chooses jury members. The comparison is descriptive because pairs share adapters and examples.

5.4 Accuracy gains at matched jury size

CIA achieves test-accuracy gains of 2.81–4.74 points over accuracy-only top-k in four illustrative matched-size comparisons: Gemma-3-1B/TAT-QA and three backbones on FinQA (Table 2). CIA also exceeds the random-subset mean in each case. Jury size is fixed, so gains occur without additional votes. Lower validation and test phi accompany these gains, consistent with the benefit of less redundant membership.

Here k is the full CIA jury size obtained on validation. Accuracy-only takes the first k candidates in validation-accuracy order; Random samples 50 subsets of that size from the same accuracy-screened pool. All methods share frozen orientations and equal voting weights. Random's SD measures subset variability. Appendix I.1 reports paired uncertainty estimates and their interpretation for these fixed cases.

Table 3: Correlation-threshold sensitivity, equally averaged over the 12 backbone–task cells. The accuracy contrast is CIA minus Top-1; the main protocol uses $\rho = 0.4$. These are fixed-pool test summaries of the existing sweep.

ρ	CIA−Top-1 (points)	Mean jury size
0.3	+0.66	6.2
0.4	+1.99	15.2
0.5	+1.19	31.8
0.6	+0.74	63.3

The gains span jury sizes from three to 23. At these extremes, Llama/FinQA reduces mean test phi from 0.675 to 0.125 and Gemma/TAT-QA from 0.330 to 0.036, showing reduced redundancy beyond the validation examples used by the filter.

Llama-3.2-1B/FinQA illustrates the distinction between comparators: CIA ties Top-1 at 57.40% but improves on accuracy-only voting at 54.59%. On Gemma/TAT-QA, random juries have lower mean validation phi than accuracy-only juries (0.087 versus 0.311), yet lower accuracy (57.84% versus 59.91%). Correlation alone is therefore insufficient; competence and joint errors matter together.

These cases evaluate the selection rule as a whole: adapter identities and correlations change together, precluding an isolated causal correlation effect.

5.5 Correlation threshold and jury-size effects

Table 3 sweeps $\rho \in \{0.3, 0.4, 0.5, 0.6\}$ at $\tau = 0.5$, equally averaging CIA-minus-Top-1 accuracy over the 12 cells. Main thresholds predate the cached-output reanalysis; the sweep describes sensitivity without selecting a new configuration from test performance. Unlike matched-size comparisons, it changes both membership and size.

The strictest threshold yields mean size 6.2 and a 0.66-point macro gain; at $\rho = 0.4$, these rise to 15.2 and 1.99 points. Larger juries at $\rho = 0.5$ and 0.6 yield smaller gains of 1.19 and 0.74 points. More votes therefore do not monotonically improve accuracy: the threshold controls both jury size and which error patterns coexist.

The Gemma/TAT-QA jury-size curve and its interpretation appear in Appendix G.3 (Figure 3).

5.6 Validation-prefix ranking diagnostic

We examine how ranking changes with the number of validation examples within the fixed seven-member Qwen3-0.6B/TAT-QA jury. For each of 200 shuffled orders of the 254 validation items, we rank these members on each growing prefix and retain the top three. The reference is a post hoc proxy consisting of the three members with highest individual test accuracy. We measure exact set agreement, partial overlap, and the test voting-accuracy gap to this proxy.

This diagnostic conditions on membership, label orientations, and tie order selected using all validation items. Test labels define only the analysis proxy, which is not a known population optimum or the best test-voting subset. The curves measure conditional ranking behavior rather than the label requirements of the full selector. Appendix G.4 retains the original-order and boundary-gap diagnostics.

Figure 2 separates membership agreement from predictive consequences. At 64 examples, exact agreement is 39.5%, mean overlap is 77.8%, and the mean voting gap is 1.10 points. At 128 examples, the corresponding values are 69.0%, 89.5%, and 0.69 points; at 192 they reach 92.5%, 97.5%, and 0.22 points.

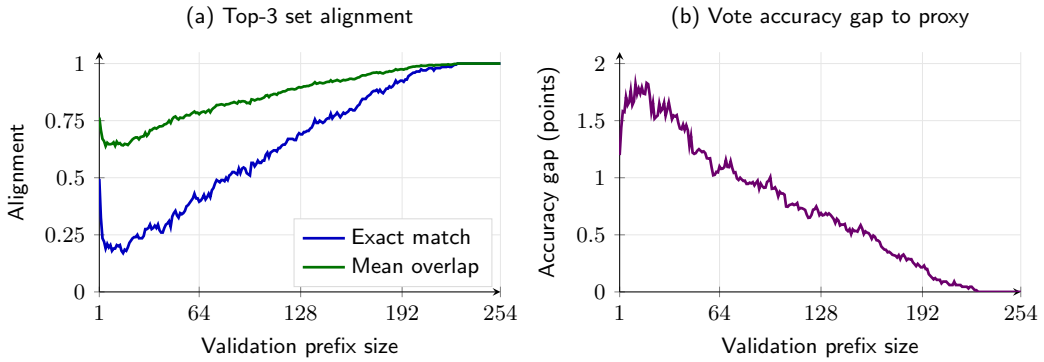


Figure 2: Ranking within a fixed Qwen3-0.6B/TAT-QA jury, averaged over 200 validation orders. (a) Agreement with the post hoc test top-3 proxy; overlap is the fraction of three members shared. (b) Proxy vote accuracy minus prefix-selected vote accuracy, in percentage points. Membership, orientations, and tie order use full validation data.

Exact set agreement is therefore a stricter criterion than either partial overlap or a small voting gap. Early behavior is non-monotonic, and all three criteria agree at the full 254-item endpoint.

5.7 Improving judgments on high-stakes slices

Motivated by Equation 1, we evaluate weighted ranking on critical compliance examples and public adherence classes. The compliance study has an explicit criticality annotation; the public study tests class prioritization without severity labels.

Compliance auditing. On constructed securities-domain consultation data, weighting lowers mean critical-example error from 2.26% to 1.65% and full-set error from 2.60% to 2.48% (Table 4). Results average five runs targeting different failure categories with the same Qwen3-1.7B pool, $k = 9$, and target weight 1.5. Of 500 test examples, 133 are critical. This flag is separate from the target categories, which include noncritical examples (Appendix F).

Mean target-category error also falls from 2.59% to 2.38%, and other-critical error from 2.26% to 1.70%. The observed gains thus extend beyond the target category. Target error counts change $1 \to 1$, $5 \to 4$, $3 \to 2$, $2 \to 3$, and $2 \to 2$; one category worsens. Both rules use plain top-k without correlation filtering, isolating weighted ranking. Appendix G.5 gives the size/weight sweep.

Table 4: Exploratory compliance weighting, averaged over five target-slice runs. All, Target, Other crit., and Critical are test error rates (%); Target mass is the target contribution to total error. The same Qwen3-1.7B pool supplies both plain top-k rules.

	Setting		Test error rate (%)				
Rule	k	w_{target}	All	Target	Other crit.	Critical	Target mass
Unweighted top-k	9	1	2.60	2.59	2.26	2.26	0.52
Slice-weighted top-k	9	1.5	2.48	2.38	1.70	1.65	0.48

Public adherence tasks. Weighted ranking reduces target-class error by 8.17 points (95% CI $[7.07, 9.29]$), improving seven public cells and tying five (Table 5). We re-rank each CIA jury with validation weight 1.5 on non-adherent examples and 1 otherwise, retaining at most nine members. The comparator is the same-sized accuracy-ordered prefix; mean size is 8.33. Non-target and overall errors increase by 9.57 and 0.58 points, respectively: the benefit is specific to detecting non-adherent answers.

Table 5: Public class weighting ($k_{\max} = 9$, $\lambda = 1.5$). Reductions are unweighted minus weighted errors (percentage points), averaged equally over 12 cells; negative values indicate deterioration.

Metric	Reduction	95% paired CI
Target-class error	+8.17	[7.07, 9.29]
Non-target error	-9.57	[-10.83, -8.31]
Overall error	-0.58	[-1.57, 0.42]
Target error contribution	+4.14	[3.53, 4.76]

Weight and size sensitivity. All eight public settings in Table 6 reduce mean target error, with no cell-level target deterioration. Reductions range from 10.60 to 17.30 points at cap three and 8.17 to 10.89 at cap nine. At cap three and weights 1.5 or 2, target gains accompany overall improvements of 0.21 and 0.19 points; the remaining settings increase overall error. The sweep characterizes prioritization strength without selecting a new primary configuration.

Table 6: Public class-weighting sweep. Reductions follow Table 5; W/T/L counts target-error improvements/ties/deteriorations across 12 cells.

$k_{\max}$	λ	Target reduction	Non-target reduction	Overall reduction	W/T/L
3	1.5	+10.60	-10.45	+0.21	8/4/0
3	2	+11.92	-11.81	+0.19	9/3/0
3	3	+15.93	-19.99	-1.81	11/1/0
3	4	+17.30	-21.21	-1.74	11/1/0
9	1.5	+8.17	-9.57	-0.58	7/5/0
9	2	+9.94	-11.20	-0.49	8/4/0
9	3	+10.36	-12.04	-0.69	8/4/0
9	4	+10.89	-12.97	-0.87	8/4/0

Compliance gains involve few errors, and neither weighting grid has a separate held-out assessment of configuration selection. These studies measure classification errors rather than realized harm. Appendix I.4 gives the per-cell public results.

6 Discussion

The four matched-size comparisons provide evidence that complementary membership can improve accuracy without adding votes. Weighted ranking also yields lower critical-example and overall errors in the constructed compliance study. Together with the public target-class gains, these findings support considering competence, shared errors, and error priorities in jury selection.

Limitations. Direct ensemble-accuracy optimizers are not compared. Bootstrap intervals condition on fixed pools and selections; alternative parsing, retraining, and distribution shift remain unmeasured. Empirical screens do not certify the diagnostic bound's population assumptions. Training 252 adapters and running multiple judges adds computation; latency and energy advantages remain unestablished. Slice results measure classification errors, not realized harm.

7 Conclusion

Judge CIA combines heterogeneous adapters with accuracy- and correlation-aware selection. Across the 12 evaluated backbone–task settings, CIA gains 1.99 percentage points over Top-1 on average, with gains up to 7.00 points. Four illustrative matched-size comparisons show gains of 2.81–4.74 points over accuracy-only voting, accompanied by lower error correlation. In the constructed compliance study, slice-weighted ranking lowers mean critical-example error from 2.26% to 1.65% alongside lower overall error. These findings support complementary membership and explicit error priorities in small-model judge design.

Reproducibility Statement

Appendices G and J.1 specify selection, the exact public-task prompts, adapter module sets, seed mapping, and hyperparameters. The anonymous supplementary archive provides cached correctness vectors and parsed test predictions, orientation metadata, selection/voting/bootstrap code, frozen configurations, and tests. It reproduces the main accuracy contrast, four matched-size endpoints and paired intervals, item transitions in the two Top-1 losses, threshold sweep, public class weighting, Gemma jury-size curve, heterogeneity and prefix statistics, and the output-protocol audit in Appendix J. It does not rerun training, inference, or external comparison backbones. Decoded text before parsing is unavailable, so alternative parsers cannot be evaluated from these caches. Validation correctness relies on the original item order. Private compliance examples and model checkpoints are excluded.

Ethics Statement

The public experiments assess answer adherence on existing benchmark data. They do not establish suitability for medical or financial decisions. The internal compliance study uses constructed audit scripts and is not presented as a measurement of harm in deployed consultations. The class-weighted results explicitly report increases in non-target errors so that a target-class improvement is not mistaken for overall safety. Public supplementary data exclude private compliance examples and original benchmark text. Any use of the methods with sensitive data or in consequential decisions requires application-specific evaluation and human accountability.

AI Use Statement

Generative AI tools were used for code development, literature search and comparison, manuscript drafting and editing, exposition and algebraic checking of theoretical results, and the analysis, interpretation, and presentation of cached experimental results. AI tools were not used to design the underlying methodology. The authors reviewed and verified the AI-assisted text, code, and analyses and take responsibility for the final manuscript.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sebastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, et al. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905.

Anish Acharya, Kris W. Pan, and Brian Verkhovsky. RoPoLL: Robust panel of LLM judges, 2026. URL https://arxiv.org/abs/2606.30931.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R. Routledge, and William Yang Wang. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711, 2021.

Thomas G. Dietterich. Ensemble Methods in Machine Learning. In Gerhard Goos, Juris Hartmanis, and Jan Van Leeuwen (eds.), *Multiple Classifier Systems*, volume 1857, pp. 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. ISBN 978-3-540-67704-8 978-3-540-45014-6. doi: 10.1007/3-540-45014-9_1.

Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pp. 973–978, 2001.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. Ragbench: Explainable benchmark for retrieval-augmented generation systems, 2025. URL https://arxiv.org/abs/2407.11005.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, et al. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2567–2577, 2019. doi: 10.18653/v1/D19-1259.

Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003. doi: 10.1023/A:1022859003006.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, 2023. doi: 10.18653/v1/2023.emnlp-main.153.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Dragos D. Margineantu and Thomas G. Dietterich. Pruning adaptive boosting. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 211–218, 1997.

Meta Llama. Llama 3.2 model card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md, 2024. Accessed: 2026-05-07.

Mistral AI. Introducing mistral 3. https://mistral.ai/news/mistral-3, 2025. Accessed: 2026-05-07.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

Hossein A. Rahmani, Emine Yilmaz, Nick Craswell, and Bhaskar Mitra. JudgeBlender: Ensembling automatic relevance judgments. In *Companion Proceedings of the ACM on Web Conference 2025*, 2025. doi: 10.1145/3701716.3715536. URL https://doi.org/10.1145/3701716.3715536.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

H. Peyton Young. Condorcet's theory of voting. *American Political science review*, 82(4):1231–1244, 1988.

Jitian Zhao, Changho Shin, Tzu-Heng Huang, Satya Sai Srinath Namburi GNVV, and Frederic Sala. CARE: Confounder-aware aggregation for reliable LLM evaluation, 2026. URL https://arxiv.org/abs/2603.00039. Version 2.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021.

A CLASSICAL INDEPENDENT-JURY BASELINE

We first present an idealized independence baseline. The theorem below is a standard concentration result, included for context rather than as a new theoretical contribution. We begin with a tail bound for abstract binary random variables.

Theorem A.1 (Hoeffding–Chernoff tail bound)**.** *Let $Z_1, \dots, Z_n \in \{0,1\}$ be independent random variables, and define*

$$S := \sum_{i=1}^{n} Z_i, \qquad \bar{\varepsilon} := \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[Z_i], \qquad \Delta := \tfrac{1}{2} - \bar{\varepsilon}.$$

If $\bar{\varepsilon} < 1/2$, then

$$\mathbb{P}(S \geq n/2) \leq \exp\bigl(-2n\Delta^2\bigr).$$

The proof is given in Appendix E.

Definition A.2 (Judge CIA jury)**.** *Let $\mathcal{X}$ be an input space, $\mathcal{D}$ a data distribution on $\mathcal{X}$, and $f : \mathcal{X} \to \{0,1\}$ the ground-truth oracle. An agent $A_i : \mathcal{X} \to \{0,1\}$ is a randomized decision rule.*

Given an odd number n of agents $\{A_1, \dots, A_n\}$ producing decisions $\hat{y}_i = A_i(x) \in \{0,1\}$, the Judge CIA collective decision is the majority vote

$$\hat{y}(x) = \mathbf{1}\Bigl\{\sum_{i=1}^{n}(2\hat{y}_i - 1) > 0\Bigr\}.$$

The odd-cardinality convention avoids ties in this idealized binary-output definition. In the implemented evaluation, unrecognized outputs abstain and tied recognized counts are scored incorrect; odd size alone then does not rule out a tie.

For the binary jury, write $Z_i = \mathbf{1}\{A_i(x) \neq f(x)\}$ and $S = \sum_i Z_i$. Probabilities are over $x \sim \mathcal{D}$ and any internal randomness of the agents.

Corollary A.3 (Average-error Condorcet-style bound for Judge CIA)**.** *If the agent error indicators $Z_1, \dots, Z_n$ are independent and their average marginal error satisfies*

$$\bar{\varepsilon} := \frac{1}{n}\sum_{i=1}^{n} \mathbb{P}(A_i(x) \neq f(x)) < \tfrac{1}{2},$$

then the Judge CIA majority vote satisfies

$$\mathbb{P}(\hat{y}(x) \neq f(x)) \leq \exp(-2n\Delta^2) \quad \textit{for the majority margin } \Delta := \tfrac{1}{2} - \bar{\varepsilon}.$$

In particular, $\mathbb{P}(\hat{y}(x) \neq f(x)) \to 0$ as $n \to \infty$ whenever $\Delta > 0$ is fixed.

The proof is given in Appendix E. This is a heterogeneous, average-error version of the classical *Condorcet Jury Theorem*: independence and an aggregate margin $\Delta > 0$ make majority error decay, even when the individual marginal errors are not identical (Young, 1988).

A.1 SHARED LATENT CONDITIONS

To explain one source of error correlation, let θ denote a common condition affecting all agents, such as a distribution shift, shared prompt artifact, or shared training-data bias. Agent i responds through its error propensity $p_i(\theta)$. If

$$Z_i \mid \theta \;\sim\; \mathrm{Bernoulli}(p_i(\theta)), \qquad Z_i \perp Z_j \mid \theta \;(i \neq j).$$

then conditional independence holds only after θ is fixed; marginally, agents may still be correlated because their propensities depend on the same θ. By the law of total covariance,

$$\mathrm{Cov}(Z_i, Z_j) = \mathrm{Cov}\bigl(p_i(\theta), p_j(\theta)\bigr) \quad (i \neq j),$$

so heterogeneity helps when it makes the propensity functions $p_i(\theta)$ and $p_j(\theta)$ less aligned. In our experiments, heterogeneity is induced by varying prompt template, LoRA rank/scaling settings,

target-module set, and sampling seed. These axes are intended to produce adapters whose error propensities respond differently to shared latent difficulty factors, thereby reducing alignment among the $p_i(\theta)$ functions. We use Pearson correlation on binary error-indicator vectors, equivalently the phi correlation coefficient, as the empirical redundancy statistic (Kuncheva & Whitaker, 2003). Empirically, we estimate pairwise error correlations $\widehat{\rho}_{ij}$ from adapter error vectors and report mean and maximum selected-jury pairwise error correlation as scalar redundancy diagnostics. Pairwise correlations involving zero-variance error vectors are handled using the convention described in Appendix G.1.

B A CONCEPTUAL SLICE-RISK DECOMPOSITION

Majority aggregation can reduce error, but high-stakes deployment also depends on which errors are operationally severe. The following chain-rule decomposition separates slice prevalence, error, and the probability of an adverse consequence. It is a conceptual identity, not an experimentally validated risk model. Throughout, a slice denotes a measurable subset of examples defined by task metadata, operational context, or a high-stakes attribute. In the catastrophic-risk decomposition below, $\mathcal{H} \subseteq \mathcal{X}$ is the high-stakes subset and $(\mathcal{H}_s)_{s\in\mathcal{S}}$ denotes a disjoint measurable partition of $\mathcal{H}$. If monitored metadata slices overlap in an application, the decomposition can be applied to a disjoint refinement of those slices. In experiments, compliance slices are defined by criterion criticality, policy flags, and violation or failure-type metadata; details are given in Appendix F.

Definition B.1 (Catastrophic event and risk)**.** *A* catastrophic event *is specified by a predicate* $\mathcal{C}(x, \hat{y}(x), f(x)) \in \{0, 1\}$*, where* $\mathcal{C} = 1$ *denotes that the realized decision induces a domain-specified severe adverse consequence. The predicate may depend on the input, the decision, and the ground truth. For* $x \sim \mathcal{D}$*, define the catastrophic risk of a decision rule* $\hat{y}$ *as*

$$\mathrm{CatRisk}(\hat{y}) := \mathbb{P}\big(\mathcal{C}(x, \hat{y}(x), f(x)) = 1 \ \wedge\ \hat{y}(x) \neq f(x) \ \wedge\ x \in \mathcal{H}\big).$$

Here, $\mathcal{C}$ identifies the adverse consequence itself, while the additional conditions $\hat{y}(x) \neq f(x)$ and $x \in \mathcal{H}$ restrict evaluation to high-stakes errors. Examples include falsely approving a non-compliant consultation or falsely flagging a compliant consultation as severe.

Definition B.2 (Slice-wise catastrophic amplification)**.** *Let* $(\mathcal{H}_s)_{s\in\mathcal{S}}$ *be the high-stakes slice partition above, with* $\mathbb{P}(x \in \mathcal{H}_s) > 0$*. Define the slice-wise misclassification rate*

$$\mathrm{Err}_s(\hat{y}) := \mathbb{P}\big(\hat{y}(x) \neq f(x) \mid x \in \mathcal{H}_s\big),$$

and, for a fixed decision rule $\hat{y}$*, the slice-wise* catastrophic amplification factor

$$\kappa_s(\hat{y}) := \mathbb{P}\big(\mathcal{C}(x, \hat{y}(x), f(x)) = 1 \mid x \in \mathcal{H}_s \ \wedge\ \hat{y}(x) \neq f(x)\big) \in [0, 1].$$

When the conditioning event has probability zero, we set $\kappa_s(\hat{y}) := 0$*. Thus* $\kappa_s(\hat{y})$ *is the decision-rule-dependent probability that an* error *becomes* catastrophic *on slice* $\mathcal{H}_s$*.*

Lemma B.3 (Slice-specific catastrophic risk decomposition)**.**

$$\mathrm{CatRisk}(\hat{y}) = \sum_{s\in\mathcal{S}} \mathbb{P}(x \in \mathcal{H}_s)\, \kappa_s(\hat{y})\, \mathrm{Err}_s(\hat{y}).$$

The decomposition distinguishes slice prevalence, slice-wise error, and the decision-rule-dependent factor $\kappa_s(\hat{y})$*. Reducing an unweighted or slice-weighted error proxy alone does not guarantee lower catastrophic risk, because the other factors need not remain fixed.*

The proof is given in Appendix E.

Remark B.4 (Why this is different from misclassification)**.** *Even when average error is small,* CatRisk *can be dominated by rare, high-impact slices. The decomposition separates slice-wise error control from the domain policy question of whether an error becomes catastrophic.*

C CLASSICAL RANKING-STABILITY RESULTS

Lemma B.3 motivates examining slice-wise error, while also showing why it alone does not determine catastrophic risk. Building on the slice terminology in Appendix B, we call $\mathcal{H}_s$ the target slice when

it is assigned larger weight during slice-weighted ranking. The target-slice weighting protocol is detailed in Appendix G.5. We record standard Hoeffding-plus-union-bound results for ordinary and weighted ranking. The candidate decision rules are treated as fixed before drawing the selection sample; these results do not certify the full pipeline that chooses orientations and jury membership on that same sample.

C.1 RANKING STABILITY

Definition C.1 (Selection sample and empirical error)**.** *Let $V := \{x^{(1)}, \dots, x^{(m)}\}$ be an i.i.d. selection sample set with $x^{(k)} \sim \mathcal{D}$. For agent A_i, define its true statistical error and empirical error by*

$$\varepsilon_i := \mathbb{P}(A_i(x) \neq f(x)), \qquad \hat{\varepsilon}_i := \frac{1}{m}\sum_{k=1}^{m} \mathbf{1}\{A_i(x^{(k)}) \neq f(x^{(k)})\}.$$

Equivalently, $\varepsilon_i = \mathbb{E}[Z_i]$ without assuming a common error rate across agents.

Definition C.2 (Top-k ranking rule)**.** *Fix an integer $k \in \{1, \dots, n\}$. Compute empirical errors $\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n$ on the selection sample set and keep the k agents with the smallest empirical error. Formally, let $\hat{\sigma}$ be a permutation such that*

$$\hat{\varepsilon}_{\hat{\sigma}(1)} \leq \hat{\varepsilon}_{\hat{\sigma}(2)} \leq \cdots \leq \hat{\varepsilon}_{\hat{\sigma}(n)}.$$

Define the kept set $\widehat{\mathcal{K}}_k := \{\hat{\sigma}(1), \dots, \hat{\sigma}(k)\}$.

Proposition C.3 (Ranking stability of top-k under a separation margin)**.** *Let σ be a permutation that sorts the* true *errors:*

$$\varepsilon_{\sigma(1)} \leq \varepsilon_{\sigma(2)} \leq \cdots \leq \varepsilon_{\sigma(n)}, \qquad \mathcal{K}_k := \{\sigma(1), \dots, \sigma(k)\}.$$

For $1 \leq k < n$, assume a separation margin at the boundary:

$$\varepsilon_{\sigma(k)} \leq \varepsilon_{\sigma(k+1)} - 2\gamma \quad \textit{for some } \gamma > 0.$$

Then the empirical top-k ranking recovers the true top-k set with probability at least

$$\mathbb{P}\big(\widehat{\mathcal{K}}_k = \mathcal{K}_k\big) \geq 1 - 2n\exp(-2m\gamma^2).$$

On the same strict concentration event, with probability at least $1 - 2n\exp(-2m\gamma^2)$,

$$\max_{i \in \mathcal{K}_k} \hat{\varepsilon}_i < \min_{j \notin \mathcal{K}_k} \hat{\varepsilon}_j.$$

The proof is given in Appendix E.

Risk note. This stability result controls only ranking error on the selection sample distribution. If V under-samples operationally important sub-slices (especially high-stakes or rare slices), then the recovered ranking may be optimal for $\mathcal{D}$ yet suboptimal for deployment. A stratified deployment evaluation may help examine the slices in Appendix B; changing the sampling distribution requires a corresponding change to the estimand or weights rather than a direct application of this i.i.d. statement.

C.2 SLICE-WEIGHTED RANKING

Definition C.4 (Slice-weighted empirical error)**.** *Let $(\mathcal{H}_s)_{s \in \mathcal{S}}$ be a measurable partition of a high-stakes subset $\mathcal{H} \subseteq \mathcal{X}$ (as in Appendix B). Let $w : \mathcal{X} \to [0, W]$ be a bounded weight function that upweights high-stakes slices; for example, one may use*

$$w(x) = 1 + \sum_{s \in \mathcal{S}} (w_s - 1)\mathbf{1}\{x \in \mathcal{H}_s\}, \qquad 1 \leq w_s \leq W.$$

Given an i.i.d. selection sample set $V = \{x^{(1)}, \dots, x^{(m)}\}$ with $x^{(k)} \sim \mathcal{D}$, define the slice-weighted empirical error of agent A_i as

$$\hat{\varepsilon}_i^{(w)} := \frac{\sum_{k=1}^{m} w(x^{(k)})\, \mathbf{1}\{A_i(x^{(k)}) \neq f(x^{(k)})\}}{\sum_{k=1}^{m} w(x^{(k)})}.$$

For ranking agents, we use the equivalent unnormalized weighted score

$$\widehat{L}_i^{(w)} := \frac{1}{m}\sum_{k=1}^{m} w(x^{(k)})\,\mathbf{1}\{A_i(x^{(k)}) \neq f(x^{(k)})\}.$$

This score induces the same ordering as $\hat{\varepsilon}_i^{(w)}$ whenever the common denominator is positive, and remains well-defined when the empirical denominator is zero. Define the corresponding weighted true *error*

$$\varepsilon_i^{(w)} := \frac{\mathbb{E}_{x\sim\mathcal{D}}[w(x)\,\mathbf{1}\{A_i(x) \neq f(x)\}]}{\mathbb{E}_{x\sim\mathcal{D}}[w(x)]}.$$

Definition C.5 (Top-k ranking with slice-weighted empirical error)**.** *Replace $\hat{\varepsilon}_i$ in the top-k ranking rule by $\widehat{L}_i^{(w)}$ and keep the k agents with the smallest weighted empirical scores. Denote the kept set by $\widehat{\mathcal{K}}_k^{(w)}$.*

Proposition C.6 (Ranking stability of weighted top-k under a separation margin)**.** *Let σ be a permutation that sorts the* weighted true *errors:*

$$\varepsilon_{\sigma(1)}^{(w)} \leq \varepsilon_{\sigma(2)}^{(w)} \leq \cdots \leq \varepsilon_{\sigma(n)}^{(w)}, \qquad \mathcal{K}_k^{(w)} := \{\sigma(1), \ldots, \sigma(k)\}.$$

For $1 \leq k < n$, assume a separation margin at the boundary:

$$\varepsilon_{\sigma(k)}^{(w)} \leq \varepsilon_{\sigma(k+1)}^{(w)} - 2\gamma \quad \textit{for some } \gamma > 0.$$

Assume further that the weight function is bounded as $w(x) \in [0, W]$ and that the mean weight *is uniformly bounded below: $\mu_w := \mathbb{E}_{x\sim\mathcal{D}}[w(x)] \geq \mu_{\min} > 0$. Then the weighted empirical top-k ranking recovers the true weighted top-k set with probability at least*

$$\mathbb{P}\big(\widehat{\mathcal{K}}_k^{(w)} = \mathcal{K}_k^{(w)}\big) \geq 1 - 2n\exp\Big(-\frac{2m\,\mu_{\min}^2\,\gamma^2}{W^2}\Big).$$

The proof is given in Appendix E.

Remark C.7 (A proxy for individual slice-weighted error)**.** *Choosing $w(x)$ concentrated on high-stakes slices makes top-k selection prioritize lower slice-weighted errors. In particular, if $w(x) = \sum_s w_s \mathbf{1}\{x \in \mathcal{H}_s\}$, then ranking by $\varepsilon_i^{(w)}$ emphasizes the slices with larger w_s. This ranks individual weighted errors. It does not directly optimize jury error or account for the decision-rule-dependent factors in* CatRisk*.*

D Equicorrelated Diagnostic Special Case

Assumption 4.1 bounds population error probabilities and pairwise error correlations. The validation screens are empirical analogues; passing them does not establish these population conditions. For intuition, the bounded-redundancy diagnostic reduces to the following common-error, equicorrelated special case. This appendix result is not an assumption enforced by Judge CIA.

Assumption D.1 (Equicorrelated diagnostic model)**.** *Assume a common marginal error $\mathbb{E}[Z_i] = \varepsilon$ and*

$$\mathrm{Corr}(Z_i, Z_j) = \rho \quad \textit{for all } i \neq j,$$

where ρ lies in the admissible range for the joint Bernoulli distribution and $1 + (n-1)\rho > 0$. The main regime of interest is $\rho \geq 0$, where redundant errors reduce the usable jury size.

Proposition D.2 (Equicorrelated effective jury size diagnostic)**.** *Under Assumption D.1, assume $\varepsilon < 1/2$ and let $\Delta_{\mathrm{diag}} := 1/2 - \varepsilon$. Define*

$$n_{\mathrm{eff}} := \frac{n}{1 + (n-1)\rho}.$$

Then $\mathrm{Var}(S) = n\varepsilon(1-\varepsilon)(1 + (n-1)\rho)$, and

$$\mathbb{P}(S \geq n/2) \leq \frac{\mathrm{Var}(S)}{\mathrm{Var}(S) + n^2\Delta_{\mathrm{diag}}^2} = \frac{\varepsilon(1-\varepsilon)}{\varepsilon(1-\varepsilon) + n_{\mathrm{eff}}\Delta_{\mathrm{diag}}^2}.$$

The same variance-equivalent reading applies to Proposition 4.2 in the main text. There, $k_{\mathrm{eff}}(\phi) = k/(1 + (k-1)\phi)$ is a diagnostic for how much pairwise redundancy inflates the variance of the jury error count. It should not be read as a calibrated prediction of test error or as an assumption that selected adapters have common error rates or exact equicorrelation.

E PROOFS

Proof of Theorem A.1. We have $\mathbb{E}[S] = n\bar{\varepsilon}$, so $\{S \geq n/2\} = \{S - \mathbb{E}[S] \geq n\Delta\}$. Since the $Z_i \in [0,1]$ are independent, Hoeffding's inequality gives

$$\mathbb{P}(S - \mathbb{E}[S] \geq n\Delta) \leq \exp(-2n\Delta^2).$$

□

Proof of Corollary A.3. A majority-vote error can occur only when at least half of the agents err:

$$\{\hat{y}(x) \neq f(x)\} \subseteq \{S \geq n/2\}.$$

The result follows from Theorem A.1. □

Proof of Proposition D.2. First,

$$\mathrm{Var}(S) = \sum_i \mathrm{Var}(Z_i) + 2\sum_{i<j} \mathrm{Cov}(Z_i, Z_j) = n\varepsilon(1-\varepsilon)(1 + (n-1)\rho).$$

Cantelli's inequality gives

$$\mathbb{P}(X - \mu \geq t) \leq \frac{\sigma^2}{\sigma^2 + t^2}.$$

Apply it with $X = S$, $\mu = \mathbb{E}[S] = n\varepsilon$, $t = n\Delta_{\mathrm{diag}}$, and $\sigma^2 = \mathrm{Var}(S)$. □

Proof of Proposition 4.2. For each $i \in \mathcal{K}$, $\mathbb{P}(Z_i = 1) \leq \varepsilon_0 < 1/2$, so

$$\mathrm{Var}(Z_i) \leq \varepsilon_0(1 - \varepsilon_0).$$

For $i \neq j$ in $\mathcal{K}$,

$$\mathrm{Cov}(Z_i, Z_j) \leq \phi\sqrt{\mathrm{Var}(Z_i)\mathrm{Var}(Z_j)} \leq \phi\,\varepsilon_0(1-\varepsilon_0).$$

Therefore

$$\mathrm{Var}(S_{\mathcal{K}}) \leq k\varepsilon_0(1-\varepsilon_0) + k(k-1)\phi\varepsilon_0(1-\varepsilon_0).$$

Also $\mathbb{E}[S_{\mathcal{K}}] \leq k\varepsilon_0$, hence

$$\{S_{\mathcal{K}} \geq k/2\} \subseteq \{S_{\mathcal{K}} - \mathbb{E}[S_{\mathcal{K}}] \geq k\Delta_0\}.$$

Applying Cantelli's inequality with the variance bound above gives the result. □

Proof of Lemma B.3. For readability, write $\mathcal{C} = 1$ for $\mathcal{C}(x, \hat{y}(x), f(x)) = 1$. Since $(\mathcal{H}_s)_{s\in\mathcal{S}}$ partitions $\mathcal{H}$,

$$\mathrm{CatRisk}(\hat{y}) = \sum_{s\in\mathcal{S}} \mathbb{P}(\mathcal{C} = 1 \wedge \hat{y} \neq f \wedge x \in \mathcal{H}_s).$$

For a fixed slice s,

$$\mathbb{P}(\mathcal{C} = 1 \wedge \hat{y} \neq f \wedge x \in \mathcal{H}_s) = \mathbb{P}(x \in \mathcal{H}_s)\mathbb{P}(\mathcal{C} = 1 \wedge \hat{y} \neq f \mid x \in \mathcal{H}_s).$$

If $\mathrm{Err}_s(\hat{y}) = 0$, this term is zero. Otherwise, the chain rule gives

$$\begin{aligned}\mathbb{P}(\mathcal{C} = 1 \wedge \hat{y} \neq f \mid x \in \mathcal{H}_s) &= \mathbb{P}(\mathcal{C} = 1 \mid x \in \mathcal{H}_s \wedge \hat{y} \neq f)\,\mathbb{P}(\hat{y} \neq f \mid x \in \mathcal{H}_s) \\ &= \kappa_s(\hat{y})\,\mathrm{Err}_s(\hat{y}).\end{aligned}$$

Summing over $s \in \mathcal{S}$ gives the equality. □

Proof of Proposition C.3. If $|\hat{\varepsilon}_i - \varepsilon_i| < \gamma$ holds for all $i \in \{1, \ldots, n\}$, then

$$\max_{i\in\mathcal{K}_k} \hat{\varepsilon}_i \leq \varepsilon_{\sigma(k)} + \gamma \leq \varepsilon_{\sigma(k+1)} - \gamma \leq \min_{j\notin\mathcal{K}_k} \hat{\varepsilon}_j.$$

Thus the empirical ordering cannot swap any element across the $k/(k+1)$ boundary, implying $\widehat{\mathcal{K}}_k = \mathcal{K}_k$. By the single-agent Hoeffding lemma and a union bound,

$$\mathbb{P}(\exists i : |\hat{\varepsilon}_i - \varepsilon_i| \geq \gamma) \leq 2n\exp(-2m\gamma^2),$$

which proves the claim. □

Proof of Proposition C.6. Recall the numerator score

$$L_i^{(w)} := \mathbb{E}_{x\sim\mathcal{D}}[w(x)\,\mathbf{1}\{A_i(x) \neq f(x)\}].$$

The denominator in $\hat{\varepsilon}_i^{(w)}$ is common to all agents, so the weighted empirical-error ordering is identical to the ordering by $\widehat{L}_i^{(w)}$ whenever the denominator is positive. Similarly,

$$L_i^{(w)} = \mu_w\, \varepsilon_i^{(w)},$$

where $\mu_w = \mathbb{E}_{x\sim\mathcal{D}}[w(x)]$ is common to all agents, so the true ordering by $\varepsilon_i^{(w)}$ is identical to the true ordering by $L_i^{(w)}$. Each summand in $\widehat{L}_i^{(w)}$ lies in $[0, W]$, and Hoeffding's inequality gives

$$\mathbb{P}\big(|\widehat{L}_i^{(w)} - L_i^{(w)}| \geq \mu_w\gamma\big) \leq 2\exp\Big(-\frac{2m\,\mu_w^2\gamma^2}{W^2}\Big) \leq 2\exp\Big(-\frac{2m\,\mu_{\min}^2\gamma^2}{W^2}\Big).$$

Union bound over $i \in \{1, \ldots, n\}$ gives that, with probability at least

$$1 - 2n\exp\Big(-\frac{2m\,\mu_{\min}^2\gamma^2}{W^2}\Big),$$

we have $|\widehat{L}_i^{(w)} - L_i^{(w)}| < \mu_w\gamma$ for all i. The separation condition implies

$$L_{\sigma(k)}^{(w)} = \mu_w\varepsilon_{\sigma(k)}^{(w)} \leq \mu_w\varepsilon_{\sigma(k+1)}^{(w)} - 2\mu_w\gamma = L_{\sigma(k+1)}^{(w)} - 2\mu_w\gamma.$$

Thus the same boundary-swap argument as in the unweighted case implies that no agent can cross the $k/(k+1)$ boundary, hence $\widehat{\mathcal{K}}_k^{(w)} = \mathcal{K}_k^{(w)}$. □

F Compliance Dataset Details

The high-stakes dataset is an internally constructed telephone-consultation audit dataset for securities-domain compliance. It is generated by a script-based data-construction pipeline from normalized criterion items, violation templates, policy flags, and deterministic audit metadata. The canonical sample unit is a single-unit script audit centered on one normalized criterion item, rather than a full dialogue as the primary training unit. Each example contains dialogue turns, evidence turns, item labels, and metadata describing the intended compliance criterion and any injected fault. The full constructed split contains 4720 training rows, 590 validation rows, and 590 test rows, with balanced binary labels in each split. Following the common evaluation protocol in Appendix G, the experiments cap validation and test evaluation at 500 examples each.

The data specification defines high-stakes examples through criterion-level criticality and policy flags. Examples of policy flags include explicit-confirmation requirements, numerical-precision requirements, prohibitions on leading the customer's answer, waiting-for-answer requirements, and mandatory explanation-on-request rules. Violations are organized by operational fault types corresponding to coercive inducement, deceptive bypass, topic evasion, vulnerable-group handling, and fact omission. These fields induce slices that correspond to concrete compliance failures, which is why the dataset is used for the slice-wise high-stakes error proxy experiment in Section 5.7. For the slice-weighted experiment, validation and test results are stored as a single bundle containing sample metadata and adapter-level correctness vectors. Each metadata row records the high-stakes flag and slice label, and each adapter row stores a binary correctness vector aligned to the metadata order. Thus a value of 1 means that the adapter is correct on the corresponding sample, while 0 means that it is incorrect.

G Experimental Details

Table 7 reports hyperparameters shared by the RAGBench and internally constructed compliance experiments. The model and dataset choices are described in Section 5; this appendix records implementation-level settings needed to reproduce the adapter pools and selection pipeline.

Table 7: Training, evaluation, and selection hyperparameters used for the main experiments.

Stage	Setting	Value
Data	Training split	At most 1000 training examples per task
	Evaluation splits	At most 500 validation examples and 500 test examples per task
	Reported metric	Accuracy on RAGBench binary-label validation/test sets; balanced before the final cap, which can leave residual imbalance
Training	Adapter method	LoRA; dropout 0.05; bias disabled
	Optimization	1 epoch; learning rate 2×10^{-4}; per-device train batch size 4; gradient accumulation 4; bf16 enabled
	Runtime setup	Single NVIDIA RTX 5090 GPU; training sequence length 1024; greedy decoding with temperature 0.0 and maximum 4 new tokens
Adapter pool	Size	252 adapters per task and base model
	Heterogeneity axes	Prompt template, LoRA rank/scale settings, target-module set, and adapter-specific sampling seed
	Grid values	Two prompt variants; $r \in \{4, 8, 12, 16, 24, 32\}$; $\alpha/r \in \{1, 1.5, 2\}$; seven target-module sets; base data seed 500
Selection	Candidate filter	Validation accuracy threshold $\tau = 0.5$ and pairwise correlation threshold $\rho = 0.4$
	Ranking and voting	Sort candidates by the chosen validation ranking score, break ties by validation accuracy, and enforce odd cardinality for majority voting

G.1 Correlation-Aware Jury Construction

Pairwise correlations are computed on error vectors. If either error vector has zero variance, the implementation uses correlation 0.0 for filtering. This is a computational convention, not a population correlation guarantee or conservative rejection of all duplicates. None of the 252 validation correctness vectors in any of the 12 public backbone–dataset pools has zero variance, so this convention does not affect the reported public jury selections.

Algorithm 1 specifies the selection procedure.

G.2 Main Performance Confidence Intervals

Table 8 reports the same point estimates as Table 1, with 95% paired bootstrap confidence intervals over test examples.

Algorithm 1 Correlation-aware jury construction

Require: adapters $\{A_i\}_{i=1}^{N}$, selection sample set V, accuracy threshold τ, correlation threshold ρ, ranking scores r_i, optional jury size k
Ensure: selected jury $\widehat{K}$

Freeze each adapter's validation-selected label orientation; unknown outputs remain unknown.
Compute oriented accuracies $\widehat{\mathrm{Acc}}_i$, error vectors, and pairwise empirical phi correlations $\widehat{\rho}_{ij}$ on V.
Form $P_\tau \leftarrow \{i : \widehat{\mathrm{Acc}}_i \geq \tau\}$.
Sort P_τ by increasing r_i, breaking ties by accuracy and then adapter index.
Initialize $\widehat{K} \leftarrow \emptyset$.
for i in the sorted order **do**
  if $\widehat{\rho}_{ij} \leq \rho$ for all $j \in \widehat{K}$ **then**
    Add i to $\widehat{K}$.
  end if
  if optional fixed size k is specified and $|\widehat{K}| = k$ **then**
    break
  end if
end for
If odd cardinality is required and $|\widehat{K}|$ is even, remove the lowest-ranked adapter in $\widehat{K}$.

return $\widehat{K}$

Table 8: Main performance with 95% bootstrap confidence intervals over test examples. Δ denotes the signed accuracy difference $\mathrm{Vote} - \mathrm{Top\text{-}1}$; positive values favor Vote. Top-1, Vote, and difference intervals use the verified paired bootstrap; Vanilla intervals retain the original recorded estimates. The macro difference CI is $[0.57, 3.41]$.

		Accuracy (%)			Accuracy difference	
Model	Dataset	Vanilla	Top-1	Vote	Δ	Avg. Δ
Qwen3-0.6B	FinQA	49.49	56.38	61.22	4.85	0.75
		[44.39, 54.34]	[51.53, 61.22]	[56.38, 65.82]	[1.02, 8.93]	
	TAT-QA	43.97	61.21	58.62	-2.59	
		[37.50, 50.43]	[54.74, 67.67]	[52.16, 65.09]	[-7.33, 2.16]	
	PubMedQA	16.00	61.20	61.20	0.00	
		[13.00, 19.40]	[57.00, 65.60]	[56.80, 65.60]	[-3.80, 3.80]	
Qwen3-1.7B	FinQA	48.72	65.31	67.09	1.79	1.89
		[43.62, 53.83]	[60.71, 69.90]	[62.50, 71.68]	[-2.04, 5.61]	
	TAT-QA	45.69	66.81	68.10	1.29	
		[39.22, 52.16]	[60.78, 72.84]	[62.07, 74.14]	[-3.88, 6.47]	
	PubMedQA	35.60	59.60	62.20	2.60	
		[31.40, 39.80]	[55.20, 63.80]	[58.00, 66.40]	[-1.60, 7.00]	
Llama-3.2-1B	FinQA	54.08	57.40	57.40	0.00	0.78
		[49.23, 58.93]	[52.55, 62.24]	[52.55, 62.24]	[-0.77, 0.77]	
	TAT-QA	53.02	63.36	62.50	-0.86	
		[46.55, 59.48]	[56.90, 69.40]	[56.47, 68.97]	[-6.90, 5.17]	
	PubMedQA	50.80	57.20	60.40	3.20	
		[46.40, 55.20]	[52.80, 61.60]	[56.20, 64.60]	[-1.00, 7.40]	
Gemma-3-1B	FinQA	50.00	52.81	53.83	1.02	4.54
		[45.15, 54.85]	[47.96, 57.65]	[48.98, 58.67]	[-4.34, 6.38]	
	TAT-QA	45.26	59.05	64.66	5.60	
		[38.79, 51.72]	[52.59, 65.52]	[58.62, 70.69]	[-0.86, 12.50]	
	PubMedQA	50.40	52.60	59.60	7.00	
		[46.00, 54.80]	[48.20, 57.00]	[55.40, 63.80]	[2.60, 11.60]	

G.3 JURY-SIZE CURVE AND ENDPOINT STATISTICS

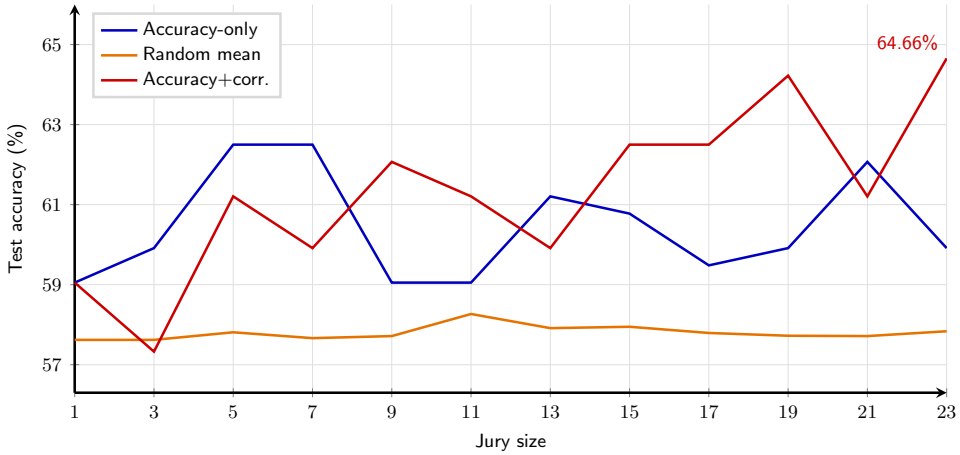


Figure 3: Gemma-3-1B/TAT-QA jury-size diagnostic using the same cached predictions as the main accuracy table. Each curve takes odd prefixes of a validation-ordered jury. Random averages 50 subsets sampled at the full matched size and ordered by validation accuracy. The relative benefit of filtering varies with jury size.

Figure 3 shows the Gemma/TAT-QA jury-size curve. At $k = 23$, CIA achieves 64.66%, compared with 59.91% for accuracy-only and 57.84% for the random mean. The gain is not monotonic in jury size, and CIA trails accuracy-only at several smaller sizes. The endpoint is the full validation-feasible jury, not a jury size chosen to maximize test accuracy. Table 9 gives the endpoint correlation statistics.

Table 9: Matched-size comparison of random, accuracy-only, and correlation-aware juries at $k = 23$. Accuracy differences are measured relative to Top-1 in percentage points. Mean and Max $\hat{\rho}$ summarize empirical phi correlation between selected adapters' validation error-indicator vectors. For Random, accuracy and within-jury mean/max correlations are averaged over 50 draws.

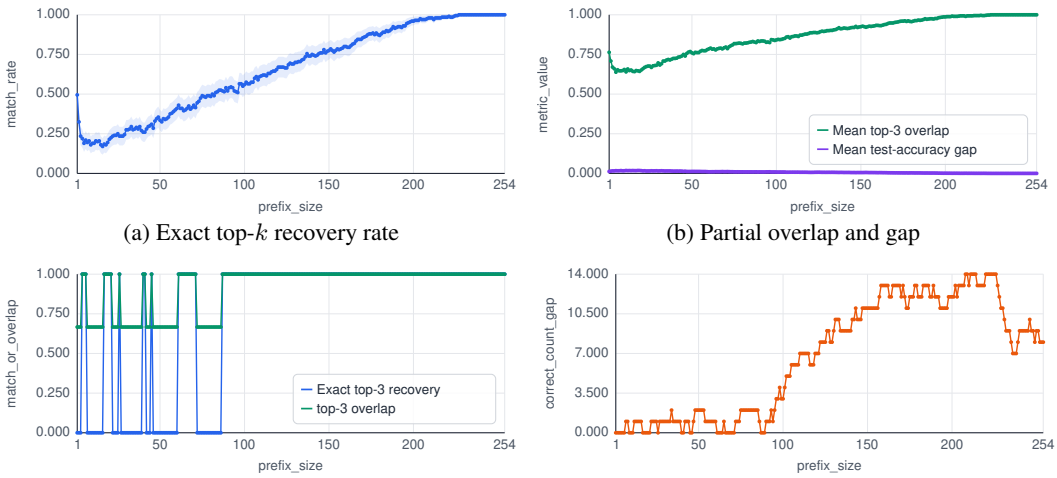
Selection rule	Jury acc. (%)	Jury size	Accuracy difference	Mean $\hat{\rho}$	Max $\hat{\rho}$
Random	57.84	23	-1.22	0.087	0.846
Accuracy-only	59.91	23	0.86	0.311	0.761
Accuracy+corr.	64.66	23	5.60	0.025	0.398

G.4 VALIDATION-PREFIX STABILITY DIAGNOSTIC

Figure 4 re-ranks the seven-member Qwen3-0.6B/TAT-QA jury on growing validation prefixes and compares its top-3 set with a post hoc full-test top-3 proxy. Test labels define only this analysis reference and never select the deployed jury. Membership, orientations, and tie order were fixed using the full validation set, so this is not a full-pipeline sample-efficiency experiment or empirical recovery of a known population-optimal set. The shuffled-prefix panels aggregate 200 orders of the same validation examples, rather than independently trained pools.

G.5 COMPLIANCE DATASET SLICE-WEIGHTED EXPERIMENTS

This exploratory compliance analysis compares plain top-k rankings without correlation filtering. The evaluated grid evaluates both validation-ranked rules on the same test set; no separate held-out assessment of choosing the jury size and target weight is supplied. We therefore treat the displayed configuration and sweep descriptively, without a confirmatory claim of risk reduction. All results in Tables 10 and 11 use the Qwen3-1.7B heterogeneous adapter pool. Specifically, we varied the jury size over $k \in \{3, 9\}$ and the target-slice weight over $w_{\text{target}} \in \{1.5, 2, 3, 4\}$, while keeping the



(a) Exact top-k recovery rate

(b) Partial overlap and gap

(c) Original-order alignment

(d) Boundary gap

Figure 4: Prefix-alignment diagnostics for validation-based ranking within a fixed jury selected using the full validation set. The reference set is the post hoc full-test top-k proxy. Panels (a) and (b) aggregate over shuffled validation-prefix orders: (a) reports the strict exact-recovery rate, while (b) reports partial top-k overlap and the test-accuracy gap to that proxy. Panel (c) shows the same recovery and overlap diagnostics in the original validation order; panel (d) reports the empirical boundary gap.

candidate pool fixed. For each target slice $\mathcal{H}_s$, slice-targeted selection used

$$w(x) = \begin{cases} w_{\text{target}}, & x \in \mathcal{H}_s, \\ 1, & x \notin \mathcal{H}_s. \end{cases}$$

This weighting increases the importance of validation errors on the targeted slice while leaving all other examples at unit weight. Each weighted ranking was compared against the unweighted top-k baseline on the same candidate pool, so the only difference was whether target-slice validation errors were upweighted during ranking.

For each (k, w_{target}) pair, we report five test metrics. They measure full-set error, target-slice error, pooled error on the other critical slices, error over the full critical subset, and the mass of target-slice mistakes in the overall test distribution. The last quantity is used as a simple proxy for slice-specific deployment risk.

We retain $k = 9$ and $w_{\text{target}} = 1.5$ as a descriptive moderate-weight setting, alongside the full grid. Table 4 summarizes its five runs. The target error counts in Table 11 are $1 \to 1$, $5 \to 4$, $3 \to 2$, $2 \to 3$, and $2 \to 2$ for coercive inducement, deceptive bypass, topic evasion, vulnerable group, and fact omission, respectively. Thus the 0.21-point macro target-error reduction rests on two one-error improvements and one one-error deterioration across separately targeted juries. The critical subset has 133 items, while target categories also include noncritical examples. These are small, heterogeneous changes, not evidence of stable catastrophic-risk reduction. The sweep in Table 10 includes gains, near-zero differences, and deteriorations, and does not establish uniform robustness across weights.

G.6 Existing Assets and Licenses

Our experiments use existing public datasets, pretrained model checkpoints, and standard open-source training libraries. For the public benchmark experiments, we use the RAGBench release of TAT-QA, FinQA, and PubMedQA (Friel et al., 2025). RAGBench is released under the CC BY 4.0 license. The original TAT-QA (Zhu et al., 2021) and FinQA (Chen et al., 2021) datasets are released under CC BY 4.0, and PubMedQA (Jin et al., 2019) is released under the MIT license. We cite RAGBench and the corresponding original datasets where applicable, and we do not redistribute the original datasets.

For pretrained backbones, we use public checkpoints from the Qwen3 (Yang et al., 2025), Llama 3.1/3.2 (Grattafiori et al., 2024; Meta Llama, 2024), Gemma 3 (Gemma Team et al., 2025), Phi-

Table 10: Summary sensitivity sweep for compliance-dataset slice-targeted filtering. Each row corresponds to one (k, w_{target}) configuration evaluated on the same fixed candidate pool, averaged over the five target-slice runs. Target error reduction is measured relative to the corresponding unweighted top-k baseline and uses the same percentage scale as the error-rate columns. The remaining columns report average test error rates; All is the full test set, Critical is the full critical subset, and Target mass is the target-slice contribution to total test error. The displayed moderate-weight setting is highlighted in bold.

Setting		Target error reduction	Average test error rate (%)		
k	w_{target}		All	Critical	Target mass
3	1.5	-0.20	2.84	2.11	0.60
3	2	0.64	2.84	2.26	0.44
3	3	0.45	2.84	2.11	0.48
3	4	0.24	2.88	2.41	0.52
9	**1.5**	**0.21**	**2.48**	**1.65**	**0.48**
9	2	0.00	2.52	1.80	0.52
9	3	0.02	2.44	2.11	0.52
9	4	-0.19	2.56	2.11	0.56

Table 11: Detailed slice-level results for the displayed compliance-dataset setting $k = 9$ and $w_{\text{target}} = 1.5$. Each target slice corresponds to one failure-type category across the test set, including both critical and noncritical examples. Target count is the number of examples for that slice in the 500-example test evaluation subset. Columns use the same shorthand as Table 4: All, Target, Other crit., Critical, and Target mass are test error rates in percent. The target-error reductions summarized in Table 10 are heterogeneous across slices; bold entries mark reductions relative to the corresponding unweighted top-k row.

			Setting		Test error rate (%)				
Target slice	Target count	Rule	k	w_{target}	All	Target	Other crit.	Critical	Target mass
Coercive inducement	100	Unweighted top-k	9	1	2.60	1.00	2.88	2.26	0.20
		Slice-weighted top-k	9	1.5	2.60	1.00	**1.92**	**1.50**	0.20
Deceptive bypass	105	Unweighted top-k	9	1	2.60	4.76	1.90	2.26	1.00
		Slice-weighted top-k	9	1.5	**2.40**	**3.81**	**0.95**	**1.50**	**0.80**
Topic evasion	95	Unweighted top-k	9	1	2.60	3.16	0.92	2.26	0.60
		Slice-weighted top-k	9	1.5	**2.20**	**2.11**	0.92	**1.50**	**0.40**
Vulnerable group	104	Unweighted top-k	9	1	2.60	1.92	2.86	2.26	0.40
		Slice-weighted top-k	9	1.5	2.60	2.88	2.86	2.26	0.60
Fact omission	96	Unweighted top-k	9	1	2.60	2.08	2.75	2.26	0.40
		Slice-weighted top-k	9	1.5	2.60	2.08	**1.83**	**1.50**	0.40

4 (Abdin et al., 2024), and Ministral 3 (Mistral AI, 2025) families. Qwen3 and Ministral 3 checkpoints are released under Apache 2.0, Phi-4 checkpoints under the MIT license, Llama 3.1/3.2 checkpoints under the corresponding Meta Llama Community Licenses and acceptable use policies, and Gemma 3 checkpoints under the Gemma Terms of Use. We use these checkpoints only for research fine-tuning and evaluation, and we do not redistribute the original model weights. Any derived adapter artifacts would be subject to the corresponding base-model terms.

The implementation uses standard open-source libraries, including PyTorch (Paszke et al., 2019), Hugging Face Transformers (Wolf et al., 2020), and Hugging Face PEFT (Mangrulkar et al., 2022). PyTorch is distributed under a BSD-style license, while Transformers and PEFT are distributed under Apache 2.0. We respect the licenses and terms of use of all assets used in the paper.

Table 12: Additional comparison-model results on RAGBench. Very low vanilla scores for some instruction-tuned backbones reflect failures under the fixed binary-answer extraction protocol, not a claim about the backbone's general capability.

	Accuracy (%)					
	FinQA		TAT-QA		PubMedQA	
Comparison model	Vanilla	Fine-tuned	Vanilla	Fine-tuned	Vanilla	Fine-tuned
Gemma-3-4B	50.00	57.91	50.43	65.95	51.80	49.20
Gemma-3-12B	50.00	76.28	50.00	75.43	51.80	58.80
Llama-3.1-8B	30.87	50.00	41.38	81.47	37.80	54.00
Llama-3.2-3B	50.00	68.62	50.00	75.43	51.80	49.20
Phi-4	1.02	49.49	1.29	50.43	0.00	52.00
Phi-4-mini	50.00	50.26	50.00	50.00	51.80	52.00
Ministral-3-3B	52.04	58.16	51.29	55.60	51.20	63.80
Ministral-3-8B	50.00	61.22	50.00	55.60	51.80	54.20
Ministral-3-14B	50.77	83.16	50.43	84.05	52.00	55.60
Qwen3-4B	7.91	53.06	4.74	49.14	11.80	58.60
Qwen3-8B	50.00	52.30	50.00	50.00	51.80	51.60
Qwen3-14B	11.48	85.97	12.50	83.19	9.60	67.00

H Additional Comparison Results

The main performance table compares Judge CIA majority voting with the strongest validation-selected adapter from the same heterogeneous small-model pool. For completeness, Table 12 reports single-backbone comparison models, each evaluated before and after LoRA fine-tuning under the recorded binary-answer extraction and validation label-orientation protocol. These rows describe individual models under the recorded generation and extraction protocol. Format failures can dominate the very low scores; the table does not rank the models' general capabilities. The audit in Appendix J covers the small-adapter pools rather than these external comparison backbones. For each comparison backbone, we use a LoRA configuration adapted to that model family, including the corresponding projection-module names and architecture-compatible target modules. Thus the comparison is intended to measure the effect of fine-tuning each backbone under a reasonable model-specific adapter setup, rather than forcing all models to share an identical target-module recipe.

H.1 High-Stakes Comparison Backbone Check

For completeness, Table 13 reports high-stakes compliance results for Qwen3-8B under the same validation-calibrated label-orientation protocol used for the comparison models. This check compares the vanilla and LoRA fine-tuned backbone, not a slice-weighted adapter-pool sweep.

I Additional Diagnostics and Public Class Weighting

This appendix provides details for the cross-backbone matched-size diagnostics, a threshold sweep, public class-weighted selection, and a reproducible cached-output uncertainty analysis.

I.1 Matched-size case comparisons and uncertainty

The Gemma-3-1B/TAT-QA case is complemented by Qwen3-0.6B, Qwen3-1.7B, and Llama-3.2-1B on FinQA. Each comparison uses the same realized jury size for random competent subsets, accuracy-only top-k ranking, and correlation-aware selection. Random means and standard deviations are over 50 subsets of the fixed competent pool (seed 42). These are illustrative comparisons of the four reported cases. They jointly change adapter identities and error correlation and do not isolate a causal correlation effect.

Table 2 in the main text summarizes these four matched-size comparisons.

Table 13: Detailed high-stakes slice results for the Qwen3-8B comparison backbone. Each target slice corresponds to one failure-type slice in the 500-example test evaluation subset, matching the target-count convention in Table 11. All, Target, Other crit., Critical, and Target mass are test error rates in percent; Other crit. and Critical use critical-subset masks. Average rows aggregate over the five target-slice rows.

Target slice	Target count	Model	All	Target	Other crit.	Critical	Target mass
Coercive inducement	100	Qwen3-8B vanilla	41.60	39.00	45.19	44.36	7.80
		Qwen3-8B fine-tuned	**1.80**	**1.00**	**2.88**	**3.01**	**0.20**
Deceptive bypass	105	Qwen3-8B vanilla	41.60	43.81	43.81	44.36	9.20
		Qwen3-8B fine-tuned	**1.80**	**3.81**	**2.86**	**3.01**	**0.80**
Topic evasion	95	Qwen3-8B vanilla	41.60	35.79	44.95	44.36	6.80
		Qwen3-8B fine-tuned	**1.80**	**2.11**	**2.75**	**3.01**	**0.40**
Vulnerable group	104	Qwen3-8B vanilla	41.60	40.38	43.81	44.36	8.40
		Qwen3-8B fine-tuned	**1.80**	**0.00**	**3.81**	**3.01**	**0.00**
Fact omission	96	Qwen3-8B vanilla	41.60	48.96	44.04	44.36	9.40
		Qwen3-8B fine-tuned	**1.80**	**2.08**	**2.75**	**3.01**	**0.40**
Average	–	Qwen3-8B vanilla	41.60	41.59	44.36	44.36	8.32
		Qwen3-8B fine-tuned	**1.80**	**1.80**	**3.01**	**3.01**	**0.36**

For each case, we form the per-item correctness difference between CIA and accuracy-only votes and use 10,000 paired item-bootstrap draws (seed 20260923) to obtain a percentile 95% interval. The three FinQA cases share resampled item indices; TAT-QA is resampled independently. The intervals condition on the reported cases, trained pools, orientations, and jury memberships. They do not account for case-selection or validation-selection uncertainty, retraining, or multiple comparisons, and are descriptive rather than confirmatory.

In Table 2 order, the CIA-minus-accuracy-only 95% intervals are $[-0.26, 7.91]$, $[-0.51, 6.12]$, $[-0.77, 6.38]$, and $[-2.16, 11.64]$ percentage points. All four include zero, so the observed case gains do not individually establish a population-level improvement. The counts of items correct only under CIA versus only under accuracy-only are 39/24, 29/18, 30/19, and 39/28. The supplementary script `reproduce_matched_ci.py` reproduces the intervals and validates their endpoints against the exact finite bootstrap distribution of these paired differences. It loads only the four reported cases.

Mean test phi also decreases in these four comparisons: 0.580 to 0.223, 0.455 to 0.080, 0.675 to 0.125, and 0.330 to 0.036, respectively. The CIA test maxima are 0.454, 0.507, 0.358, and 0.589, illustrating that the validation ceiling 0.4 is not a guaranteed test-time ceiling.

I.2 Item transitions in the two Top-1 losses

We compare the frozen CIA vote with the frozen validation-selected Top-1 on the same 232 TAT-QA test items. Qwen3-0.6B has 123 items correct under both rules, 13 corrected only by CIA, 19 correct only under Top-1, and 77 wrong under both. For Llama-3.2-1B the corresponding counts are 121, 24, 26, and 61. The selected juries have no unknown outputs or voting ties. The paired differences are $100(13 - 19)/232 \simeq -2.59$ and $100(24 - 26)/232 \simeq -0.86$ percentage points.

These transitions describe the two negative point estimates in Table 1, without establishing a population-level loss or a causal explanation for the overrides. The supplementary script `reproduce_vote_losses.py` checks the existing jury and Top-1 identities, independently verifies votes by counting correct members, and reproduces the original accuracies. Its output retains the four-way counts, class breakdowns, and affected item positions. It loads only these two caches and introduces no additional accuracy-only jury comparisons.

I.3 Threshold sensitivity

We retain the main protocol's $\tau = 0.5$, $\rho = 0.4$ and report the existing sweep across $\rho \in \{0.3, 0.4, 0.5, 0.6\}$. These thresholds were fixed before the present reanalysis. We do not se-

lect another main configuration from this test sweep. The relationship between size and gain is non-monotonic. Table 3 in the main text reports this sweep.

I.4 PUBLIC CLASS-WEIGHTED SELECTION

For the public experiment, first construct the full correlation-feasible jury at $\tau = 0.5$, $\rho = 0.4$. Re-rank members of that jury by weighted validation error, with weight λ on non-adherent answers (gold label `yes`) and unit weight otherwise. Keep the first $\min(k_{\max}, |K|)$ members and apply unweighted voting. The comparator takes the same-sized prefix in the original accuracy order. This implements ranking within an already feasible jury; it does not rerun the full-pool greedy filter in weighted order. Test labels only define evaluation strata.

At $k_{\max} = 9$, $\lambda = 1.5$, mean realized jury size is 8.33. Target error decreases by 8.17 percentage points (95% CI [7.07, 9.29]), but non-target error increases by 9.57 points (CI [8.31, 10.83]). Overall error increases by 0.58 points (CI $[-0.42, 1.57]$). The target contribution to total error decreases by 4.14 points (CI [3.53, 4.76]). Target error improves in seven cells and ties in five. These results show a class-specific tradeoff, not a general reliability or catastrophic-risk improvement. The target is an adherence class, not an independently annotated severity slice.

Table 5 in the main text reports the paired intervals.

Table 6 in the main text reports the full public size/weight sweep.

Table 14: Per-cell public weighting results at $k_{\max} = 9$, $\lambda = 1.5$. Positive values indicate error reductions in percentage points.

Backbone	Task	Realized k	Target reduction	Overall reduction
Qwen3-0.6B	FinQA	9	+0.00	+0.00
Qwen3-1.7B	FinQA	9	+0.00	+0.00
Llama-3.2-1B	FinQA	3	+0.00	+0.00
Gemma-3-1B	FinQA	9	+29.59	-1.02
Qwen3-0.6B	TAT-QA	7	+0.00	+0.00
Qwen3-1.7B	TAT-QA	9	+4.31	+0.00
Llama-3.2-1B	TAT-QA	9	+0.00	+0.00
Gemma-3-1B	TAT-QA	9	+26.72	-4.31
Qwen3-0.6B	PubMedQA	9	+6.95	-0.20
Qwen3-1.7B	PubMedQA	9	+11.58	+0.60
Llama-3.2-1B	PubMedQA	9	+6.18	-0.20
Gemma-3-1B	PubMedQA	9	+12.74	-1.80

I.5 CACHED-OUTPUT REANALYSIS AND UNCERTAINTY

All 252 validation correctness vectors and test prediction vectors are read for each of the 12 backbone–task cells. Recomputed jury indices, Top-1 indices, and majority-vote predictions agree with the saved selection and voting outputs. Public test labels and ordering are reconstructed from the dataset snapshots and checked against each saved vote record. Cached validation correctness vectors rely on the common evaluation order and do not independently identify input content.

The main accuracy table, matched-size diagnostics, and jury-size curve in Figure 3 use the same cached predictions and voting rule. The macro gain over Top-1 is 1.99 percentage points, with eight wins, two ties, and two losses across the 12 public settings.

We use 10,000 paired item-bootstrap draws, seed 20260911. Within a dataset, all four backbones and compared decision rules share the same sampled item indices. Datasets are resampled independently, and the 12 cells receive equal weight in each macro average. Class-conditional denominators are recalculated per draw. The 95% percentile CI for the macro gain is [0.57, 3.41]; individual intervals are unadjusted for multiple comparisons. All pools, orientations, and jury selections remain fixed, so the intervals exclude retraining and validation-selection uncertainty.

Evaluation balances the two labels before the final cap of 500 items. Final validation/test counts are 246/392 for FinQA, 254/232 for TAT-QA, and 500/500 for PubMedQA; truncation after balancing can create a small residual class imbalance. The parser recognizes the substring `yes` first, then `no`, and otherwise returns `unknown`. Validation chooses whether to invert an adapter's binary outputs; the orientation is frozen on test. Unknown outputs count as individual errors but abstain in voting; tied recognized counts are incorrect. Pairwise phi uses 0 for a zero-variance pair and six-decimal precision in selection.

Scope of the analyses. The public supplementary archive reproduces the fixed-pool reanalysis, matched-size endpoints, jury-size curve, sensitivity and class-weighting results, and heterogeneity and prefix statistics. It does not rerun training or the external comparison backbones. The compliance primary setting is verified locally on the first 252 adapters in the available expanded bundle. Its failure-type targets include both critical and noncritical examples, whereas critical error uses the separate 133-item critical mask; the experiment uses ordinary top-k ranking without correlation filtering. These distinctions delimit the interpretation of the reported studies.

J Prompts, adapter mapping, and output-protocol audit

J.1 Exact public-task prompts and adapter mapping

All four public-task configurations use the following base prompt, substituting the retrieved documents, question, candidate response, and one instruction. Documents are joined with two newline characters. This is a plain completion prompt: the recorded adapter evaluation tokenizes this string directly, without adding a chat-template wrapper.

```
Knowledge:
{context}

Question:
{query}

Candidate Answer:
{candidate_text}

{instruction}
```

The two instruction strings are:

```
Task: Judge if the candidate answer is hallucinated.
Respond with one token only: yes or no.

Task: Determine whether the candidate answer is
hallucinated. Respond with one token only: yes or no.
```

The gold label is `no` when RAGBench's `adherence_score` is true and `yes` otherwise, after excluding rows with missing questions, responses, or adherence labels. These are answer-adherence judgments using the RAGBench annotations, not answers to the original QA tasks. Training appends a newline and the gold label to the prompt.

The seven ordered target-module sets are

1. `q_proj, v_proj`;
2. `q_proj, k_proj, v_proj`;
3. `q_proj, v_proj, o_proj`;
4. `gate_proj, up_proj`;
5. `q_proj, gate_proj`;
6. `v_proj, o_proj`;
7. `gate_proj, up_proj, down_proj`.

The rank/scaling grid uses $r \in \{4, 8, 12, 16, 24, 32\}$ in ascending order, and for each r, $\alpha/r \in \{1, 1.5, 2\}$ in that order. For zero-based adapter index i, prompt index is $i \bmod 2$, module-set index

Table 15: Output-protocol audit. Flip pool counts inverted adapters out of 252; Flip jury counts inverted members in the selected jury. Unknown rates (%) use all adapter–item outputs in the pool. The last columns report test accuracy (%) of the same selected jury after undoing inversion (Raw) and under its frozen validation-selected orientation (Cal.). Selection is held fixed: Raw is not a method reselected without calibration.

Backbone	Task	Flip pool	Flip jury	Unk. val.	Unk. test	Raw	Cal.
Qwen3-0.6B	FinQA	15	0/9	0.00	0.00	61.22	61.22
Qwen3-1.7B	FinQA	65	4/17	0.00	0.00	57.14	67.09
Llama-3.2-1B	FinQA	5	0/3	0.00	0.00	57.40	57.40
Gemma-3-1B	FinQA	92	6/15	3.37	1.92	48.98	53.83
Qwen3-0.6B	TAT-QA	5	0/7	0.00	0.00	58.62	58.62
Qwen3-1.7B	TAT-QA	3	1/15	0.00	0.00	65.52	68.10
Llama-3.2-1B	TAT-QA	19	2/9	0.00	0.00	58.62	62.50
Gemma-3-1B	TAT-QA	77	8/23	1.79	1.06	53.02	64.66
Qwen3-0.6B	PubMedQA	15	1/25	0.00	0.00	60.40	61.20
Qwen3-1.7B	PubMedQA	19	1/29	0.00	0.00	64.60	62.20
Llama-3.2-1B	PubMedQA	124	2/11	0.00	0.00	58.20	60.40
Gemma-3-1B	PubMedQA	31	2/19	3.48	2.43	58.40	59.60

is $\lfloor i/2 \rfloor \bmod 7$, and rank/scaling index is $\lfloor i/14 \rfloor \bmod 18$. Sampling seed is $500 + i$; it is not an additional Cartesian-product axis. Each adapter samples up to 1000 training examples with balanced labels when both classes permit it. The frozen YAML configurations in the supplement record the remaining settings.

J.2 Parsing, orientation, and what the caches retain

The recorded evaluation greedily generates at most four new tokens, decodes only the generated suffix with special tokens removed, lowercases and strips it, then checks for the substring `yes` before `no`. If neither is present, the output is `unknown`. This parser is permissive: it does not require an exact label, and a response containing both substrings is assigned `yes`. The 1024-token sequence setting in Table 7 is a training setting; the adapter evaluation path directly tokenizes the prompt without an explicit truncation argument.

The validation log stores raw accuracy, accuracy after binary inversion, the chosen orientation, and its accuracy for each adapter. Test caches store the parsed labels after the frozen orientation. The decoded text before parsing is absent from the supplied caches. Consequently, this audit checks orientation and abstention but cannot establish robustness to an exact-label parser, a longer generation budget, or constrained decoding. No such alternative decoding result is claimed.

J.3 Reproducible orientation and abstention audit

The supplementary script `audit_protocol.py` joins the 252 orientation records per pool with the cached validation correctness and test predictions. It verifies the recorded orientation decision, validation accuracy, test correctness, and original jury accuracy. Let a_i and b_i be an adapter's validation accuracies before and after inversion. With binary gold labels and unknown predictions counted wrong in both orientations, the unknown count is $m(1 - a_i - b_i)$; the script checks that this is a nonnegative integer. Test unknown counts are read directly from parsed predictions. Counts and denominators are retained in `protocol_audit.json`.

Table 15 reports the fraction of unknown adapter–item outputs and the number of inverted adapters. For the nine Qwen/Llama pools all validation and test outputs are recognized by the original parser. Recognition alone does not imply that the generated text was exactly a valid binary label. Gemma has nonzero unknown rates, so abstention is part of its measured voting protocol. Its selected juries contain 109, 50, and 205 unknown test outputs on FinQA, TAT-QA, and PubMedQA, respectively, over 15×392, 23×232, and 19×500 adapter–item decisions. Recognized-vote ties occur on two, zero, and two test items, respectively, and are counted wrong. The other nine juries have no test ties.

The table distinguishes inversion in the full pool from inversion in the selected jury. For example, Llama-3.2-1B/PubMedQA inverts 124 of 252 adapters but only two of its 11 jury members, whereas Gemma-3-1B/TAT-QA inverts eight of 23 selected members. The voting effect depends on the identities and joint predictions of the affected members, rather than on the pool-wide inversion count alone.

The fixed-jury counterfactual reverses each calibrated adapter's yes/no mapping on test while preserving its identity and all unknowns. It describes dependence on label orientation, not the performance of a newly selected uncalibrated ensemble. Applying the recorded orientation changes Qwen3-1.7B/FinQA accuracy from 57.14% to 67.09% and Gemma-3-1B/TAT-QA from 53.02% to 64.66%. The effect is not uniformly beneficial: Qwen3-1.7B/PubMedQA changes from 64.60% to 62.20%. The orientation is shared across the selection comparators, and its effect is separate from the matched-size comparison of filtering rules.

Table 16 separates pool-average adapter accuracy from jury accuracy. The validation gain is mechanical because validation chooses the better orientation; its test counterpart can be positive or negative. These pool means are not selected-model baselines.

Table 16: Mean individual-adapter accuracy (%) over all 252 members, before and after validation-selected label orientation. Raw → Cal. holds each adapter and its test predictions fixed.

Backbone	Task	Validation: Raw → Cal.	Test: Raw → Cal.
Qwen3-0.6B	FinQA	54.91→55.01	54.74→54.66
Qwen3-1.7B	FinQA	53.11→54.92	53.60→55.04
Llama-3.2-1B	FinQA	54.61→54.63	54.50→54.43
Gemma-3-1B	FinQA	49.33→50.48	49.10→49.22
Qwen3-0.6B	TAT-QA	56.21→56.26	54.64→54.65
Qwen3-1.7B	TAT-QA	61.21→61.26	59.85→59.80
Llama-3.2-1B	TAT-QA	52.31→52.42	54.63→54.09
Gemma-3-1B	TAT-QA	50.80→51.70	52.90→51.80
Qwen3-0.6B	PubMedQA	53.92→54.08	53.75→53.93
Qwen3-1.7B	PubMedQA	54.12→54.34	54.99→55.12
Llama-3.2-1B	PubMedQA	51.45→52.90	51.77→53.00
Gemma-3-1B	PubMedQA	50.37→50.65	51.53→51.47