

Scaling RLHF Annotation: A Hybrid Human-AI Collaboration Framework

Khanh Nguyen, Shi Chen, Suseella Panguluri, Jesse Oliver,
Salika Dave, Pritam Gundecha, Olivier Delalleau

NVIDIA

Correspondence: shic@nvidia.com

Abstract

High-quality preference data collection remains a bottleneck for RLHF reward model training: collecting multiple independent human annotations per sample is costly and hard to scale across domains and languages, while fully automated AI feedback (RLAIF) risks quality degradation. We present a production-grade hybrid human-AI annotation framework that introduces automated model-as-annotator pre-annotations into a quality-control (QC)-ratified workflow, contributing model votes only when automation is sufficiently reliable and leaving final adjudication to human QC. The pipeline (1) verifies factual claims with a web-search-grounded fact checker under a cleanroom constraint that keeps proprietary tokens out of training data, (2) elicits structured judgments from an ensemble of open-weights LLM judges, and (3) applies a two-call swap-consistency gate to mitigate position bias, with a data flywheel surfacing disagreements between QC and model outputs to drive improvement. In production, the framework reduces initial-labeler hours by $\sim 37\%$; across 11,092 production comparisons it reaches 81.6% agreement with QC ($\kappa=0.63$) and 82% to 88% on the public JudgeLM benchmark, with no QC degradation measured under an independent-model proxy. The resulting data supported a production reward model and aligned LLM.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) is central to aligning Large Language Models (LLMs) with human intent (Ouyang et al., 2022; Bai et al., 2022), but collecting preference data needed to train reward models remains a major production bottleneck. Preference annotation requires annotators to compare nuanced model responses, verify factual claims when needed, and make calibrated judgments under ambiguity. To improve label reliability, production workflows often use

multiple independent annotators plus expert quality control (QC); HelpSteer3-Preference, for example, collects over 40K preference items across General, STEM, Code, and Multilingual tasks with three to five annotators each (Wang et al., 2025b). This redundancy increases cost and latency, especially as RLHF expands across specialized domains and languages.

Pure automation offers lower cost but risks introducing unverified errors into reward-model training data (Lee et al., 2023). Conversely, fully human workflows preserve oversight but do not scale efficiently as model capabilities and evaluation complexity increase. This creates a practical need for selective automation: automated annotators should contribute only when their outputs are reliable enough for the task, while human QC remains responsible for final adjudication.

We present a production-grade hybrid human-AI annotation framework for RLHF preference labeling. The system incorporates automated judges as one or two of the three initial annotation signals under configurable deployment modes (e.g., 2 humans + 1 model, or 1 human + 2 models), while retaining human QC as the final decision maker. This hybrid collaboration model, which we term enhanced RLHF, combines web-search-grounded fact verification inspired by fact-centric evaluation (Min et al., 2023), an ensemble of open-weights LLM judges (Zheng et al., 2023), and a two-call swap-consistency gate that mitigates position bias (Shi et al., 2025) by routing uncertain cases away from automation rather than dropping them from training data.

Rather than proposing a new stand-alone judging algorithm, our key contribution is a deployed data system for combining retrieval-grounded fact-checking, LLM-as-a-Judge, position-bias mitigation, and hybrid human-AI labeling under real production constraints, including cleanroom token-lineage requirements and QC-ratified quality gates.

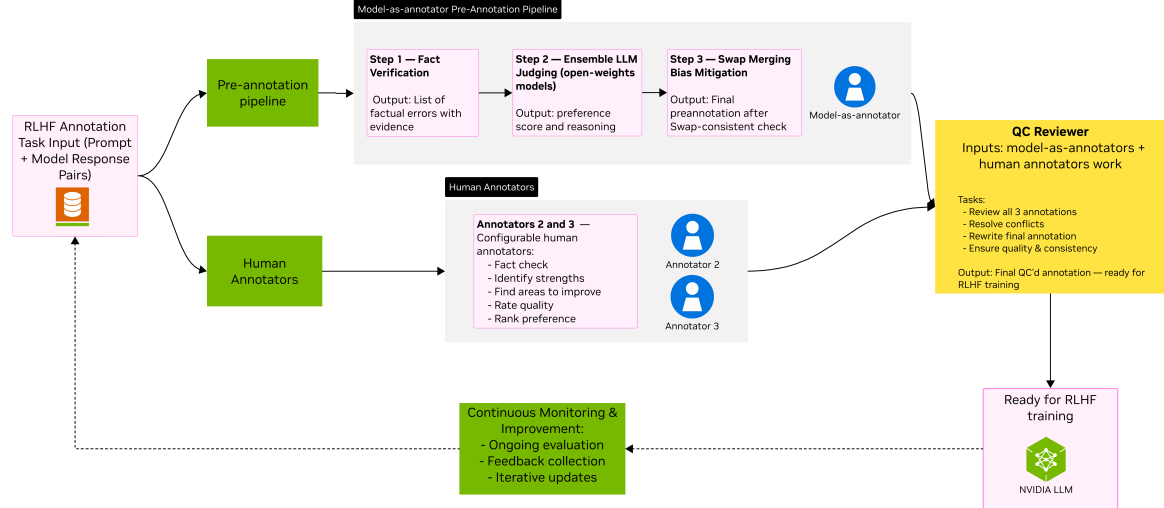


Figure 1: Overview of the scalable RLHF pre-annotation pipeline. The hybrid workflow processes tasks in parallel through (1) an automated pipeline (Fact Verification → Ensemble Judging → Swap Merging) and (2) human annotators. Both streams converge at the QC reviewer for final adjudication to produce high-quality RLHF training data.

Part of this contribution is licensing-aware component selection: the fact checker is chosen on accuracy, cost, and licensing compliance, since some proprietary models’ outputs cannot be used in training data (Section 3.2). The resulting data supported the release of NVIDIA Nemotron-3-Super (NVIDIA, 2026).

We validate the framework with evidence beyond a single curated set: a public JudgeLM benchmark (Zhu et al., 2025) that disentangles fact-checking from swap merging (Section 6.3); a large-scale production study over 11,092 model–QC comparisons with confidence intervals and per-domain breakdowns; an independent-judge study that measures no QC degradation under an independent-model proxy (Section 6.5); and a transparent decomposition of the $\sim 37\%$ reduction in initial-labeler hours.

2 Related Work

LLM-as-a-Judge. Using LLMs to evaluate outputs was popularized by Zheng et al. (2023) and underlies benchmarks such as AlpacaEval (Li et al., 2023). Closest to us, Zhu et al. (2025) fine-tune open LLMs into scalable judges, mitigating position, knowledge, and format biases via training-time swap augmentation. Our gate is complementary and works at inference time: it is model-agnostic, needs no retraining, is evaluated on the public JudgeLM benchmark (Section 6.3), and ex-

tends judging with fact-checking to catch a common blind spot, judges overlooking hallucinations (Min et al., 2023).

Position bias and consensus. Judges often favor the first response (Shi et al., 2025), echoing position bias in retrieval and QA (Joachims et al., 2016; Hofstätter et al., 2021; Ko et al., 2020); swap merging makes judgments order-invariant. Xie et al. (2026) instead aggregate heterogeneous comparisons into a calibrated consensus; this complements our gate, which decides whether a vote is trusted rather than how votes combine.

AI feedback and selective labeling. Constitutional AI (Bai et al., 2022) and RLAI (Lee et al., 2023) reduce human labels via AI feedback; we instead operationalize a hybrid annotator that augments human QC. RLTHF (Xu et al., 2025) targets human correction on likely-mislabeled samples, whereas our judgments feed a production, QC-ratified loop with explicit fact verification and swap gating. TrustJudge (Wang et al., 2025a) and Trust or Escalate (Jung et al., 2024) calibrate when a judge’s vote can be trusted, complementing our gate by potentially extending automation into the region we route to humans.

Fact verification and objective rewards. Agentic fact-checking decomposes responses into atomic claims (Min et al., 2023); we adopt a web-search-grounded variant as preprocessing so preferences rest on verified evidence. Rule-centric re-

wards such as RAIF (Qin et al., 2025) need well-defined rules, whereas we keep preference flexibility while adding grounding. Data-centric work shows quality can outweigh quantity, including LIMA (Zhou et al., 2023), AlpaGasus (Chen et al., 2023), and DEITA (Liu et al., 2023); we extend this idea to the annotation stage.

3 Methodology

Our deployment operates under three constraints: (1) a cleanroom policy that keeps proprietary model outputs out of training data; (2) automated judgments must reach $\geq 75\%$ binary agreement with expert QC, derived from historical agreement between our legacy three-human annotation workflow and final expert QC; and (3) the system must scale to high-throughput HPC batch processing below the manual per-annotation cost. The 75% gate combines an empirical anchor with a pre-deployment business decision: it was derived from historical agreement between annotators in our legacy three-human workflow and final expert QC, and adopted as the minimum acceptable level for model votes to enter production. Within these constraints, our framework automates the evaluation of model responses through a four-stage pipeline: (1) Fact Verification, (2) Automated Judging, (3) Consensus Merging, and (4) Continuous Improvement via a production quality monitoring loop (Figure 1). We describe the underlying annotation task, datasets, and evaluation protocol in Section 5.

3.1 Problem Formulation

We cast RLHF annotation as preference ranking. Given a prompt x and responses $\{y_1, y_2\}$ with a latent reward r^* , a human prefers y_1 over y_2 with Bradley–Terry–Luce probability

$$P(y_1 \succ y_2 \mid x) = \sigma(r^*(x, y_1) - r^*(x, y_2)), \quad (1)$$

where σ is the logistic function; our goal is an automated judge that approximates this distribution. Rather than regress P directly, the deployed judge emits a discrete six-level preference (Section 5.1) that we binarize. This ordinal scale encodes severity, which the swap-consistency gate (Section 3.5) exploits: a label is trusted only when the winner is invariant to response order, which holds when the latent gap $|\Delta r| = |r^*(x, y_1) - r^*(x, y_2)|$ overcomes position bias. Near-ties ($\Delta r \approx 0$) are the items most likely to flip under swapping and are routed away from automation, so the BTL view

motivates the gating rule rather than being decorative.

3.2 Fact Verification Stage

Factual accuracy is a paramount criterion, yet standard LLMs often hallucinate when answering knowledge-intensive queries. To identify these errors, an automated evaluator requires access to an external ground-truth mechanism, specifically web search. We refer to this component as the Fact Checker.

3.2.1 Architecture: Design Trade-offs

Implementing a robust Fact Checker is a build-vs-buy choice. A custom search-augmented system (a search API plus an open-weights LLM running a retrieve-read-verify loop) allows unrestricted use of generated tokens for training but carries high engineering overhead and latency. Proprietary all-in-one services (e.g., Perplexity, OpenAI, Google Gemini) offer state-of-the-art accuracy with minimal setup, but our cleanroom policy forbids using any proprietary model’s generated tokens in training data. We therefore buy with a strict compliance layer: the proprietary service is treated solely as a retrieval engine, extracting only (1) snippets flagged as erroneous, (2) the search queries used, and (3) evidence URLs and source snippets; this preserves a “clean” token lineage while leveraging superior search accuracy. This guarantee is inbound-only: no proprietary-generated text enters the training corpus, while candidate responses and queries derived from them are still sent to the third-party API (see Limitations and Ethics Statement). What is discarded is the verifier’s reasoning and synthesis; retained artifacts stay auditable, as each flagged snippet carries its source URL. We did not evaluate a fully open-weights fact-checking stack and make no comparable-performance claim.

3.2.2 Evaluation Methodology

To validate candidates, we built a synthetic Hallucination Dataset ($N=500$, D1¹) by injecting three to five factual errors per response, from obvious contradictions (e.g., wrong dates) to subtle numerical inaccuracies. We measure detection rate (response flagged as erroneous), full recall (all injected errors found), and partial recall (at least one).

¹Dataset identifiers D1 through D4 are summarized in Section 5, Table 1.

3.2.3 Model Selection and Results

We benchmarked several candidates, including GPT-5, GPT-5.1, GPT-5-mini, Gemini-2.5-Pro, Gemini-2.5-Flash, and Gemini-3-Pro. Table 5 summarizes their performance.

Gemini-2.5-Pro reached high detection (96.4%) at low cost (\$0.021/item) but almost never searched (0.2%), relying on parametric memory, which is risky for obscure or time-sensitive facts. Gemini-2.5-Flash was costliest (\$0.361/item) yet less accurate (84.0%), and GPT-5 led on detection (97.0%) but at higher cost. We selected Gemini-3-Pro: aggressive search use (91.4%) for grounded verification, competitive detection (96.8%), and lower cost (\$0.105/item) than GPT-5. The selection weighs accuracy against cost and licensing compliance. The Fact Checker is a swappable module, so we can switch providers as capabilities, pricing, or availability evolve.

3.3 Ensemble of Automated Judges

The core of the “automated annotator” is an ensemble of LLMs acting as judges. Each judge receives the user prompt and conversation history, both candidate responses (A and B), and the fact-checking error reports from the previous stage, then outputs a preference score on a 1–6 scale with a reasoning trace, weighing instruction following, factual accuracy, and coherence.

Judges run on scalable inference servers on our HPC clusters, with thinking models (e.g., Kimi-K2-Thinking) allocated more resources for extended reasoning traces (Appendix H, Table 11).

3.4 Judge Model Selection

We restrict the production judge fleet to open-weights models for transparency and reproducibility, selecting judges by three criteria: (a) binary agreement with QC on D2; (b) swap self-consistency (Section 3.5); and (c) architectural diversity (MoE vs. thinking models) to reduce correlated failures.

3.5 Bias Mitigation via Swap Merging

Position bias is a known issue in LLM-based evaluation (Shi et al., 2025), where models may favor the first or second response regardless of content (Ko et al., 2020; Hofstätter et al., 2021). We model the judge’s raw score as the true reward difference $\Delta r = r^*(x, y_1) - r^*(x, y_2)$ corrupted by a position

term δ_{pos} (> 0 favors the first slot) and noise:

$$\begin{aligned} S(y_1, y_2) &= f(\Delta r + \delta_{pos} + \epsilon), \\ S(y_2, y_1) &= f(-\Delta r + \delta_{pos} + \epsilon'). \end{aligned} \quad (2)$$

We therefore run both orders, obtaining $s_{orig} = S(y_1, y_2)$ and $s_{swap} = S(y_2, y_1)$, and place them in a common frame via $s'_{swap} = 7 - s_{swap}$. Mapping each score to a binary winner ($B(s) = \text{win}(y_1)$ for $s \in \{1, 2, 3\}$, else $\text{win}(y_2)$), we accept the automated label $L = s_{orig}$ only when $B(s_{orig}) = B(s'_{swap})$; otherwise the item is Uncertain. Consistency requires $|\Delta r|$ to overcome $|\delta_{pos}|$, so the gate trusts a vote exactly when the preference is robust to order.

Swap merging is a routing gate, not a data filter. A natural concern is that discarding swap-inconsistent cases biases the training distribution by removing hard near-ties. However, “Uncertain” applies per judge, not per item: another judge in the fleet may still be consistent, and combined fleet coverage reaches 99.24% on D2 (Table 6). The $\leq 0.76\%$ of items with no consensus are not dropped but routed to the standard three-human path, so ambiguous items move away from automation, not out of the RM training distribution.

Deployment mechanics and relation to JudgeLM. The five judges form an available pool, not a committee: a deployed task carries one or two model annotations (Section 6.6), each independently swap-gated, with disagreements resolved by QC adjudication rather than inter-judge voting. Unlike JudgeLM’s swap augmentation (Zhu et al., 2025), which swaps response order while training the judge, our gate runs at inference time on any fixed judge and accepts a vote only when both orders select the same winner.

3.6 Production Monitoring and the Data Flywheel

Beyond offline evaluation, we built a production monitoring workflow that implements a data flywheel: a virtuous cycle where production feedback continuously improves the automated annotator, which in turn generates higher-quality pre-annotations. For each batch of production items reviewed by QC, we pull the discrepancy between the QC decisions and the pipeline’s pre-annotations (including fact-checking reports and judge rationales), and use an agentic skill to produce a concise report highlighting dominant failure modes and concrete examples of disagreement.

Feedback loop. The flywheel operates through a systematic loop: (1) collect QC decisions and compare against automated pre-annotations; (2) analyze disagreements via LLM-assisted categorization to identify patterns; (3) iterate on judge prompts, rubrics, swap-consistency thresholds, or the fact-checking stage; and (4) deploy improvements and monitor their impact on subsequent batches. On a 128-item production sample, the dominant sources of disagreement were prompt interpretation and subjectivity (Appendix C, Table 7), guiding subsequent rubric clarifications.

We treat this loop as a practical tuning mechanism, not a fully self-improving system: we conducted four end-to-end pipeline iterations (updates to prompts, model choices, and thresholds) informed by QC-vs-model discrepancy reports. Production agreement figures (Section 6.4) aggregate traffic across these iterations rather than reflecting a single frozen version. As the annotator improves, we can shift more tasks to the 1 human + 2 models configuration while maintaining annotation quality.

4 Implementation

The pipeline is a modular Python package built on NeMo Skills (Gitman, 2025) and Slurm for multi-stage job orchestration on HPC clusters. Each stage (Preprocessing, Fact Checking, LLM Judging, Swap Merging, Output Validation) is independently configurable via YAML and extensible with new judge models. Judges are deployed as an available pool; each model annotation is independently swap-gated (Section 3.5) before reaching QC. We version prompt templates with changelogs and encode LLM-as-a-Judge best practices: explicit bias-mitigation instructions, JSON-schema output, and rubrics for instruction following, truthfulness, coherence, and verbosity; automated output validation ensures no samples are lost and all outputs conform to schema. All final annotation data are reviewed and verified by QC.

5 Experimental Setup

5.1 Annotation Task

This work follows a standard RLHF preference-labeling setup: given a prompt and two responses, the annotator produces structured feedback (strengths and areas of improvement) and a six-level preference label encoding direction and severity: a score of 1/2/3 means Response A is much better/better/slightly better than B, and

Table 1: Datasets used in this work. D3 is the large-scale production slice; D4 is a public benchmark.

ID	Dataset	n	Purpose
D1	Synthetic Hallucinations	500	Fact-checker selection
D2	Expert-Curated	145	Tuning, selection, ablation
D3	Production	11,092	Deployment evaluation
D4	JudgeLM val (public)	5,000	External validation

a score of 4/5/6 means the reverse. When responses contain factual claims, annotators verify them against external sources as needed.

5.2 Prompt Selection and Taxonomy

We select prompts spanning diverse task types (fact-seeking and open-ended QA, reasoning, creative writing, rewriting, brainstorming, advice, educational content) across production domains: English General, Code, STEM, Safety, and the Finance/Healthcare/Legal verticals.

5.3 Data Curation and Datasets

Curating reliable preference data is challenging even before automation. Many production tasks are hard to use directly: a single comparison can require extensive reading, domain expertise, and fact verification. And labels are not always clean—when two responses are very close, the preference can collapse to annotator-specific style rather than objective correctness, introducing irreducible noise.

Table 1 summarizes the datasets used. We distinguish development datasets (pipeline development, prompt tuning, ablations) from validation datasets (evaluating accuracy under real deployment).

D2 (Expert-Curated, $n=145$) is a senior-scientist-reviewed set for tuning, judge selection, and the ablation (near-ties removed; see below). D3 (Production, $n=11,092$) is a Jan–Mar 2026 slice with QC-verified labels, and D4 is the public JudgeLM validation set.

Why curated results exceed production. During reconciliation between the senior scientist and the broader QC team, we removed borderline near-ties whose “winner” depended on subjective framing rather than encoding them as ground truth. D2 is thus intentionally biased toward clearer preferences, so its agreement numbers are an upper bound relative to production (D3), which has a higher fraction of near-ties.

6 Results

We validate the pipeline through four complementary lenses—a curated ablation, the public JudgeLM benchmark, large-scale production, and an independent-judge study—and then decompose the cost savings.

6.1 Evaluation Metrics

We report Binary Agreement and Cohen’s κ . We map the six-level preference to a binary winner via the 1–3 (A) vs. 4–6 (B) split from Section 3.5; Binary Agreement is the fraction of n valid pairs where model and human winners match, and κ corrects for chance (Appendix G). We report bootstrap 95% CIs (2,000 resamples) for production results. As robustness checks on the choice of coefficient, Appendix G additionally reports Scott’s π and ordinal-weighted Krippendorff’s α on the same production pairs.

6.2 Ablation on Expert-Curated Data (D2)

To quantify each component’s contribution on expert-reviewed items, we compare performance before and after adding fact-checking and swap merging on D2 ($n=145$). This is a development-set diagnostic: D2 was also used for judge selection and threshold tuning, so we read Table 2 as a before/after component contrast rather than held-out performance; Section 6.3 provides the held-out evidence. Gains are consistent across all five judges, with the best (Kimi-K2-Thinking) reaching 92.59% agreement and $\kappa=0.83$ (almost perfect). At $n=145$, per-judge differences (e.g., 92.6 vs. 88.9) are within roughly ± 5 points of binomial noise, so the table does not rank judges statistically.

Table 2: Development-set diagnostic on the Expert-Curated set (D2, $n=145$, used for judge selection and threshold tuning): binary agreement (Agr., %) and Cohen’s κ vs. QC, before and after fact-checking and swap merging.

Judge	Agr. (%)		Cohen’s κ	
	Before	After	Before	After
kimi-k2-thinking	87.00	92.59	0.72	0.83
deepseek-v3.2	83.20	88.89	0.64	0.77
qwen3-235b	81.20	85.71	0.60	0.70
qwen3vl-235b	76.50	84.00	0.52	0.66
gpt-oss-120b	73.50	82.76	0.41	0.61

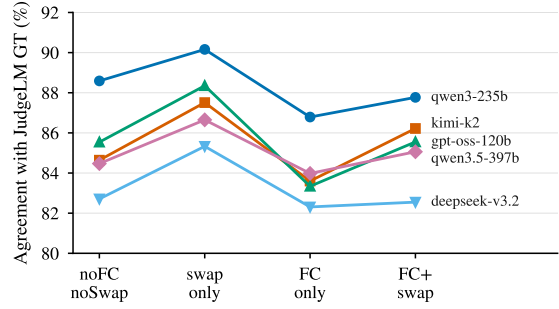


Figure 2: Binary agreement (%) vs. JudgeLM GPT-4 ground truth (D4, $n=5,000$) across pipeline configurations. Y-axis starts at 80%; exact values, κ , and support in Appendix J.

6.3 External Validation and Unbundled Ablation (JudgeLM Benchmark)

Because our strongest internal results are on private data, we also evaluate judges on the public JudgeLM benchmark (D4, $n=5,000$) against its GPT-4 ground truth, which lets us unbundle the two components introduced together. Figure 2 plots binary agreement across four configurations: no fact-checking (FC) / no swap, swap only, FC only, and FC+swap; exact values are in Appendix J, Table 12.

Two findings emerge. First, swap merging helps uniformly, improving every judge by +1.6 to +2.9 points over no-swap—stabilizing preferences rather than discarding data. Second, fact-checking slightly lowers raw agreement with the JudgeLM benchmark GT, not because it hurts judging but because it surfaces factual errors in GT-preferred answers (Appendix J). The full configuration (FC+swap) still reaches 82.6–87.8% ($\kappa=0.65$ –0.76), confirming our judges generalize beyond internal data.

6.4 Production Deployment Evaluation

Baseline. Before fact-checking and swap merging, earlier judge versions reached only modest QC agreement on early production traffic (48–76%, κ up to 0.52), motivating the enhancements in Sections 3.2–3.5.

Large-scale evaluation (D3). On 11,092 model-vs-QC pairs from a three-month slice (Table 3), overall agreement is 81.6% (95% CI [80.9, 82.3], $\kappa = 0.632$); per-model agreement ranges 78.2–84.8% with moderate-to-substantial κ (> 0.55) and tight CIs, validating production accuracy at scale. The 81.6% figure is a window aggregate across the

Table 4: Claude Opus 4.6 vs. QC agreement in the blind independent study (6,379 records, seven domains).

Condition	N	Agr. (95% CI)	κ
0 models (3 humans)	2,087	71.1 [69.2, 73.0]	0.421
1 model + 2 humans	2,102	74.8 [73.0, 76.8]	0.496
2 models + 1 human	2,190	76.5 [74.8, 78.3]	0.530

four engineering iterations of Section 3.6.

Table 3: Optimized production evaluation per judge family (D3, $n=11,092$, Jan–Mar 2026): agreement vs. QC with bootstrap 95% CIs. Per-domain results are in Appendix D.

Judge	N	Agr. (%)	95% CI	κ
deepseek-v3.2	2,268	81.3	[79.7, 82.8]	0.625
gpt-oss-120b	2,295	78.2	[76.4, 79.9]	0.563
kimi-k2-thinking	2,245	82.8	[81.3, 84.3]	0.656
qwen3-235b-a22b	2,286	81.4	[79.9, 83.0]	0.628
qwen3.5-397b-a17b	1,998	84.8	[83.1, 86.3]	0.697
All	11,092	81.6	[80.9, 82.3]	0.632

Per-domain breakdown. Performance is non-uniform (Appendix D, Table 8): agreement is highest where quality is more objective (Finance 86.4%, Code 85.8%) and lowest where judgments are subjective or knowledge-intensive (Legal 79.7%, STEM/Safety $\sim 80.5\%$), but all domains exceed our $\geq 75\%$ target.

6.5 Blind Independent Preference Ranking Study

We use an independent-judge study as a safety brake: a blind test for degradation of QC decisions under pre-annotation, not evidence of improvement. Claude Opus 4.6 ranked 6,379 production pairs across seven domains from only the prompt and two responses, blind to QC labels, pre-annotations, and condition metadata. Claude–QC agreement rises with the number of visible pre-annotations (Table 4; 0 vs. 1 and 0 vs. 2 models significant at $p=0.006$ and $p<0.001$; 1 vs. 2 not, $p=0.20$).

The design cannot separate accurate QC from QC anchoring on model outputs, since drift toward model preferences would also raise agreement with an independent model judge, and the two readings are not symmetric: the monotone rise here (71.1% $\rightarrow$ 74.8% $\rightarrow$ 76.5%) and in the held-out study of Appendix F (68% $\rightarrow$ 77%; QC win rate 64% $\rightarrow$ 62%) is the pattern anchoring would produce, so the evidence leans toward some alignment with model preferences. The study therefore establishes only that no QC degradation is measured under

an independent-model proxy. Automation stays capped at the deployed mix (~ 1.1 model / ~ 1.9 initial-human slots per task; Appendix K) with QC retaining final adjudication; blind expert re-adjudication and a reward-model ablation on 0/1/2 pre-annotations remain future work (Limitations).

6.6 Cost Decomposition

The reduction comes from replacing one or two of the three initial human labeling slots, while QC remains fully human. Over a representative 25,400-task, three-month period, 20% of tasks use 3 humans, 50% use 2 humans + 1 model, and 30% use 1 human + 2 models. Of the 76,200 initial labeling slots, 27,940 (37%) are automated. Via per-domain handling times (Appendix K), this is 45,019 hybrid initial-labeler hours versus 71,459 all-human, a 37% reduction (26,440 hours). Automation costs $\sim \$0.55$ per automated initial-labeling slot (fact-checking + judges), under 0.4% of the $\sim \$144.89$ cost of a human initial-labeling slot, so the saved labeler hours translate almost entirely into net savings. A severity-stratified analysis (Appendix E) shows the share of initial annotations from a model is nearly flat across difficulty (39.8% on clear winners vs. 36.1% on near-ties), so the saving is not concentrated on easy pairs.

7 Conclusion

We presented a hybrid framework for RLHF preference annotation: automated judges (web-search-grounded fact-checking, an open-weights judge ensemble, and a swap-consistency gate) contribute only when self-consistent and fact-grounded, routing uncertain cases to human QC. It reduces initial-labeler hours by $\sim 37\%$ at 81.6% QC agreement (82–88% on JudgeLM), with no QC degradation measured under an independent-model proxy, and supported a production reward model and Nemotron-3-Super—showing efficiency and quality need not trade off in RLHF data generation.

Limitations

Our framework has several limitations.

1. Downstream impact is measured by proxy. We do not present a reward-model or policy ablation isolating the effect of pre-annotations on downstream alignment; the independent-judge study (Section 6.5) measures no QC degradation under an independent-model proxy but cannot rule out QC anchoring on

model outputs. It also relies on one reference judge (Claude Opus 4.6), and any model-based reference shares preferences with the pre-annotating models, so the ideal instrument is a gold standard independent of model preferences, such as blind expert re-adjudication with no models in the loop. A controlled study training RMs on QC data with 0/1/2 pre-annotations and on hybrid vs. all-human data, together with such re-adjudication, is planned future work (Appendix M).

2. Upper-bound curated set. The Expert-Curated set (D2) excludes near-ties and thus yields optimistic agreement numbers relative to production traffic.
3. Partial external reproducibility and cleanroom scope. Our strongest evidence (D3) is internal production data; public-benchmark results (JudgeLM, D4) mitigate this, but full reproduction is limited by data access and the cleanroom constraint. What is traded away is the verifier’s reasoning and synthesis; the retained artifacts (flagged snippets, queries, evidence URLs and source snippets) are auditable against their sources, but the proprietary search orchestration is not reproducible, and we did not evaluate a fully open-weights fact-checking stack, so we make no comparable-performance claim. The cleanroom claim is also inbound-only: candidate responses and queries derived from them reach third-party APIs (see Ethics Statement).
4. Residual biases. Swap Merging addresses position bias but not others (e.g., verbosity, style, domain blind spots); calibration approaches (Xie et al., 2026) are a promising complement.

Ethics Statement

Data Privacy and Consent. The preference data used in our validation was collected from professional annotators who provided informed consent for their labels to be used in research. All personally identifiable information was removed prior to processing, and the pipeline does not store or transmit user data beyond the immediate processing context.

Legal Compliance. A critical constraint is the strict separation of proprietary model outputs from training data, to comply with commercial licensing

terms. Candidate responses are generated exclusively by target open-weights models or human writers; proprietary services are restricted to auxiliary roles (primarily fact-checking), their outputs treated as transient metadata never written into the final corpus, and the final (Prompt, Winner, Loser) decision is ratified by human QC. This guarantee is inbound-only. In the outbound direction, candidate responses and search queries derived from them are sent to third-party APIs during fact verification, an API-input confidentiality consideration distinct from token lineage; deployments with stricter confidentiality requirements should assess provider data-handling terms before adopting this design.

Bias and Fairness. LLM judges may inherit biases from training data. Our Swap Merging addresses position bias, but other biases (e.g., verbosity, style) may persist. We recommend using the automated annotator as one vote in a multi-annotator consensus, regularly auditing automated labels against diverse human pools, and monitoring for systematic disagreements.

Potential Misuse. While designed to improve data quality, the tool could be misused to generate low-quality labels without oversight. We mitigate this by designing a hybrid workflow that requires human participation, flags uncertain cases for mandatory human review, and provides transparency via detailed reasoning traces.

References

- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. *Constitutional ai: Harmlessness from ai feedback*. *Preprint*, arXiv:2212.08073.
- Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Sriniwasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2023. *AlpaGasus: Training a better alpaca with fewer data*. *arXiv preprint arXiv:2307.08701*.
- Igor Gitman. 2025. How to streamline complex LLM workflows using NVIDIA NeMo-Skills. <https://developer.nvidia.com/blog/how-to-streamline-complex-llm-workflows-using-nvidia-nemo-skills/>. NVIDIA Developer Blog.
- Sebastian Hofstätter, Aldo Lipani, Sophia Althammer, Markus Zlabinger, and Allan Hanbury. 2021. *Mitigating the position bias of transformer models in passage re-ranking*. *arXiv preprint arXiv:2101.06980*.

- Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2016. [Unbiased learning-to-rank with biased feedback](#). *arXiv preprint arXiv:1608.04468*.
- Jaehun Jung, Faeze Brahman, and Yejin Choi. 2024. [Trust or escalate: LLM judges with provable guarantees for human agreement](#). *arXiv preprint arXiv:2407.18370*.
- Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. [Look at the first sentence: Position bias in question answering](#). *arXiv preprint arXiv:2004.14602*.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2023. [RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback](#). *arXiv preprint arXiv:2309.00267*.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.
- Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023. [What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning](#). *arXiv preprint arXiv:2312.15685*.
- Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. [FACTScore: Fine-grained atomic evaluation of factual precision in long form text generation](#). *arXiv preprint arXiv:2305.14251*.
- NVIDIA. 2026. [Nemotron 3 Super: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning](#). Technical report, NVIDIA.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA. Curran Associates Inc.
- Yulei Qin, Gang Li, Zongyi Li, Zihan Xu, Yuchen Shi, Zhekai Lin, Xiao Cui, Ke Li, and Xing Sun. 2025. [Incentivizing reasoning for advanced instruction-following of large language models](#). *arXiv preprint arXiv:2506.01413*.
- Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2025. [Judging the judges: A systematic study of position bias in LLM-as-a-judge](#). *arXiv preprint arXiv:2406.07791*.
- Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. [Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges](#). In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 404–430. Association for Computational Linguistics.
- Yidong Wang, Yunze Song, Tingyuan Zhu, Xuanwang Zhang, Zhuohao Yu, Hao Chen, Chiyu Song, Qiufeng Wang, Cunxiang Wang, Zhen Wu, Xinyu Dai, Yue Zhang, Wei Ye, and Shikun Zhang. 2025a. [Trust-Judge: Inconsistencies of LLM-as-a-judge and how to alleviate them](#). *arXiv preprint arXiv:2509.21117*.
- Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Hoo-Chang Shin, Felipe Soares, Alexander Bukharin, Ellie Evans, Yi Dong, and Oleksii Kuchaiev. 2025b. [HelpSteer3-Preference: Open human-annotated preference data across diverse tasks and languages](#). *Preprint*, arXiv:2505.11475.
- Zejun Xie, Xintong Li, Guang Wang, and Desheng Zhang. 2026. [AtC: Aggregate-then-calibrate for human-centered assessment](#). In *The Fourteenth International Conference on Learning Representations (ICLR)*.
- Yifei Xu, Tusher Chakraborty, Emre Kıcıman, Bibek Aryal, Eduardo Rodrigues, Srinagesh Sharma, Roberto Estevao, Maria Angels de Luis Balaguer, Jessica Wolk, Rafael Padilha, Leonardo Nunes, Shobana Balakrishnan, Songwu Lu, and Ranveer Chandra. 2025. [RLTHF: Targeted human feedback for LLM alignment](#). *Preprint*, arXiv:2502.13417.
- Yusuke Yamauchi, Taro Yano, and Masafumi Oyamada. 2025. [An empirical study of LLM-as-a-judge: How design choices impact evaluation reliability](#). *arXiv preprint arXiv:2506.13639*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-judge with MT-bench and Chatbot Arena](#). In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. [LIMA: Less is more for alignment](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021. Curran Associates, Inc.
- Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2025. [JudgeLM: Fine-tuned large language models are scalable judges](#). In *International Conference on Learning Representations*, volume 2025, pages 51257–51296.

A Fact Checker Selection

Table 5 reports the full benchmark of Fact Checker candidates on the synthetic hallucination dataset (D1), used to select the production Fact Checker (Section 3.2).

Table 5: Performance of Fact Checker candidates on the synthetic hallucination dataset (D1). **Search Usage** indicates the frequency of tool invocation. Costs include input/output tokens and search queries.

Model	Detect. Rate (%)	Search (%)	Full (%)	Partial (%)	Cost (\$)
GPT-5	97.0	90.6	34.0	60.8	0.138
Gemini-3-Pro	96.8	91.4	36.9	55.4	0.105
Gemini-2.5-Pro	96.4	0.2	42.7	49.1	0.021
GPT-5.1	95.2	54.4	37.9	55.2	0.139
Gemini-2.5-Flash	84.0	86.0	31.8	45.0	0.361
GPT-5-mini	81.0	75.4	32.4	42.8	0.030

B Expert-Curated Set (D2): Swap Consistency

On the Expert-Curated set (D2, $n=145$), Table 6 reports per-judge swap self-consistency supporting the routing-gate analysis in Section 3.5.

Table 6: Swap self-consistency on the Expert-Curated set (D2, $n=145$): the percentage of samples producing consistent rankings after swap merging. Combined fleet coverage (≥ 1 swap-consistent judge) is 99.24% (by 3 judges: 67.9%; by 2: 19.1%; by 1: 12.2%; none: 0.8%). This analysis used an earlier three-judge fleet; the production deployment was later expanded to five judges (Table 3).

Model	Self-Consistency
DeepSeek-v3.2-Speciale	90.48%
Kimi-K2-Thinking	86.90%
qwen3-235b-a22b-instruct	87.60%

C Data Flywheel: Disagreement Categories

Table 7 reports the disagreement-category distribution surfaced by the production feedback loop (Section 3.6).

Table 7: Disagreement category distribution from the automated feedback loop on a production sample ($n=128$).

Category	Count	(%)
Prompt interpretation	35	27.3
Subjectivity	29	22.7
Factual error	19	14.8
Verbosity	16	12.5
Instruction following	14	10.9
Hallucination	10	7.8
Coherence	5	3.9

D Per-Domain Production Results

Table 8 breaks down the optimized production evaluation (D3, $n=11,092$) by domain.

Table 8: Optimized production evaluation per domain (D3, $n=11,092$): agreement vs. QC with bootstrap 95% CIs.

Domain	N	Agr. (%)	95% CI	κ
Code	302	85.8	[81.8, 89.7]	0.714
Finance	177	86.4	[80.8, 91.5]	0.728
General	6,630	81.9	[80.9, 82.8]	0.636
Healthcare	118	85.6	[78.8, 91.5]	0.707
Legal	143	79.7	[72.7, 86.0]	0.591
Safety	2,231	80.6	[79.0, 82.2]	0.612
STEM	1,491	80.5	[78.4, 82.6]	0.609
All	11,092	81.6	[80.9, 82.3]	0.632

E Severity-Stratified Analysis of D3

We stratify D3 by QC’s six-level severity (Section 5.1), folding the mirrored labels into much better, better, and slightly better (near-tie); Figure 3 shows both sides. Model–QC agreement falls from 90.9% ($\kappa=0.82$) on clear winners to 69.6% ($\kappa=0.39$) on near-ties. The share of initial annotations from a model, however, is nearly flat: 39.8% vs. 36.1%, against 37.4% overall. The model-annotation share is therefore spread almost evenly across difficulty rather than concentrated on easy pairs, with a mild tilt toward clearer items. Severity here is QC’s final label, assigned after review; it is an observable difficulty proxy rather than a measure available when tasks were routed.

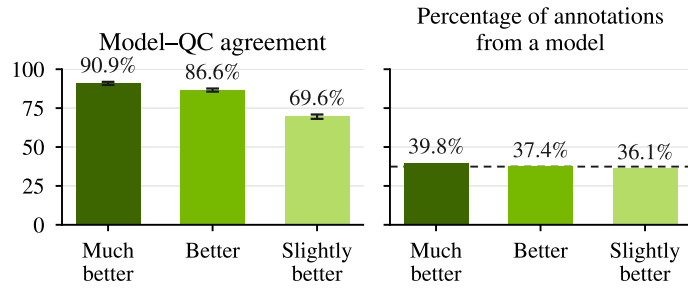


Figure 3: Severity-stratified view of D3. Left: model–QC agreement by QC severity (95% CIs; $n=11,092$). Right: share of initial annotations from a model (15,166 tasks; dashed line: overall 37.4%).

F Complementary QC Quality Study

We ran a separate May 2026 analysis to test whether QC quality changes with the number of model pre-annotations visible during review. The study used a random subset of roughly 1.4K production annotations from a single annotation-vendor pool. Claude Opus 4.6 compared the QC annotation against held-out model annotations for the same task, meaning model annotations that existed for the task but were not shown to QC as pre-annotations. This setup asks whether QC still wins against unseen model annotations as the number of visible pre-annotations changes.

Table 9 summarizes the aggregate results. QC win rate is broadly stable, moving from 64% with no model pre-annotations to 62% with two, while agreement with held-out model annotations on the best response rises from 68% to 77%. We interpret this as mixed evidence: model pre-annotations can add useful signal, but they may also increase alignment with model preferences. The result supports retaining human QC and mixing one-model and two-model pre-annotation settings rather than moving to a fully automated workflow.

Table 9: Complementary QC quality study using held-out model annotations. Error ranges are reported as $\pm$ percentage points; agreement columns include the number of evaluated comparisons in parentheses.

Visible pre-annotations	Valid judgments	QC win rate	Neither-valid agr.	Best-response agr.
0 models (3 humans)	3,460	64 ± 1.5	78 ± 0.5 (23,042)	68 ± 0.7 (16,428)
1 model + 2 humans	7,338	63 ± 1.0	79 ± 0.3 (54,414)	73 ± 0.4 (39,994)
2 models + 1 human	3,956	62 ± 1.4	82 ± 0.4 (29,432)	77 ± 0.5 (22,680)

G Agreement Metrics Details

We report Binary Agreement and Cohen’s Kappa (κ) in the main text. For binary labels, let p_o be the observed agreement and p_e the expected agreement by chance:

$$p_o = \frac{\text{Number of agreements}}{n} \quad (3)$$

$$p_e = \frac{n_{m,1} \cdot n_{s,1} + n_{m,-1} \cdot n_{s,-1}}{n^2} \quad (4)$$

where $n_{m,\cdot}$ and $n_{s,\cdot}$ are the counts of model and human preferences for each label. Then $\kappa = (p_o - p_e)/(1 - p_e)$. We use the common interpretation: $\kappa < 0.20$ (slight), 0.21–0.40 (fair), 0.41–0.60 (moderate), 0.61–0.80 (substantial), 0.81–1.00 (almost perfect).

Robustness to metric choice. Beyond Cohen’s κ , we report two further chance-corrected coefficients recommended in recent studies of LLM-as-a-judge agreement (Thakur et al., 2025; Yamauchi et al., 2025), computed on the same 11,092 model–QC pairs as Table 3 with the same exclusions. Scott’s π estimates chance agreement from the pooled label marginals of both raters rather than per-rater marginals. Krippendorff’s α is computed on the full six-level preference labels (not the binarized winner) using the ordinal distance metric in its coincidence-matrix formulation, so adjacent-severity disagreements (e.g., “better” vs. “slightly better”) are penalized less than opposite-winner flips. Table 10 reports both, with bootstrap 95% CIs (2,000 resamples).

Two observations follow. First, π matches κ to three decimals for every judge (0.632 overall), so the reported reliability is not an artifact of how chance agreement is estimated. Second, ordinal α exceeds the binary coefficients for every judge (0.666 overall vs. $\kappa=0.632$; largest gap for qwen3.5-397b-a17b, 0.757 vs. 0.697), indicating that a meaningful share of residual model–QC disagreement is adjacent-severity rather than opposite-winner.

Table 10: Chance-corrected agreement between model pre-annotations and QC on D3 ($n=11,092$, same pairs as Table 3): Cohen’s κ , Scott’s π , and ordinal-weighted Krippendorff’s α (six-level labels), with bootstrap 95% CIs (2,000 resamples).

Judge	N	κ	Scott’s π (95% CI)	Ordinal α (95% CI)
deepseek-v3.2	2,268	0.625	0.625 [0.593, 0.656]	0.637 [0.608, 0.667]
gpt-oss-120b	2,295	0.563	0.563 [0.528, 0.597]	0.619 [0.589, 0.647]
kimi-k2-thinking	2,245	0.656	0.656 [0.627, 0.686]	0.672 [0.644, 0.697]
qwen3-235b-a22b	2,286	0.628	0.628 [0.596, 0.659]	0.641 [0.614, 0.667]
qwen3.5-397b-a17b	1,998	0.697	0.697 [0.663, 0.727]	0.757 [0.735, 0.779]
All	11,092	0.632	0.632 [0.617, 0.646]	0.666 [0.654, 0.678]

H Judge Serving Configuration

Table 11 lists representative serving configurations for the open-weights production judge fleet on our Slurm-managed HPC clusters.

Table 11: Inference configuration for the open-weights production judge fleet. Hardware rows reflect representative serving configurations.

Model	Server	Resources	Max Tokens
DeepSeek-v3.2-Speciale	SGLang	2 Nodes (16 GPUs)	131,072
qwen3-235b-a22b-instruct	vLLM	2 Nodes (16 GPUs)	16,384
Kimi-K2-Thinking	vLLM	2 Nodes (16 GPUs)	32,768
GPT-OSS-120B	vLLM	1 Node (8 GPUs)	32,768
qwen3.5-397b-a17b	vLLM	2 Nodes (16 GPUs)	32,768

I Exploratory Inter-Model Agreement

Early in the project, we examined pairwise preference agreement between candidate judge models on the Expert-Curated set (D2) to understand model behavior. This analysis was exploratory and did not drive our final judge selection, which used agreement with QC, swap self-consistency, and architectural diversity (Section 3.5). Figure 4 shows two patterns: (i) **clustered “opinions”**—Gemini, DeepSeek, and Qwen variants align more strongly with each other, while OpenAI-family models agree more within their own group; and (ii) **capacity effects**—larger frontier models exhibit higher mutual agreement and more stable rankings, whereas smaller models are more idiosyncratic on challenging items.

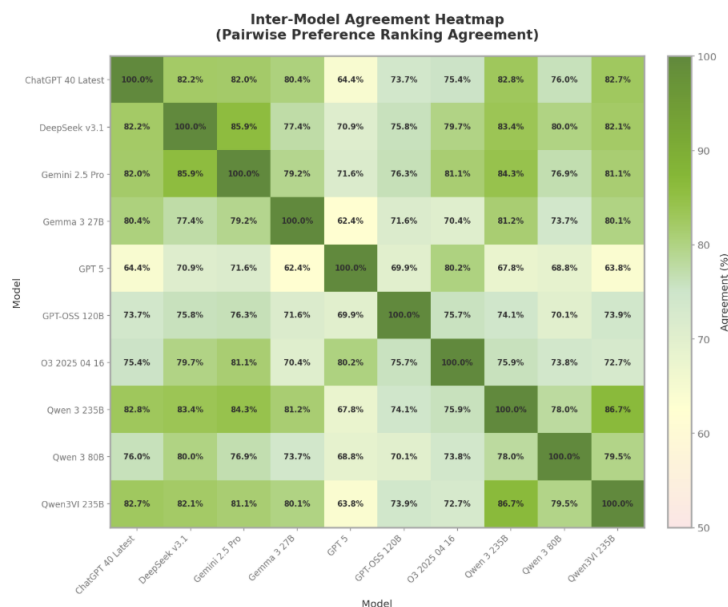


Figure 4: Inter-model agreement heatmap showing pairwise preference ranking agreement between candidate models on the Expert-Curated set (D2). Exploratory only; this did not influence model selection.

J JudgeLM: Agreement, Kappa, Support, and GT-FC Tension

Table 12 reports the exact binary-agreement values plotted in Figure 2; Table 13 reports Cohen’s κ and evaluation support (n_{valid}) for the same configurations.

Table 12: Binary agreement (%) vs. JudgeLM benchmark GPT-4 ground truth (D4, public, $n=5,000$) across pipeline configurations (exact values for Figure 2).

Judge	noFC noSwap	swap only	FC only	FC+ swap
qwen3-235b-a22b	88.59	90.16	86.79	87.77
kimi-k2-thinking	84.64	87.51	83.60	86.22
gpt-oss-120b	85.54	88.35	83.34	85.55
deepseek-v3.2-speciale	82.71	85.33	82.31	82.55
qwen3.5-397b-a17b	84.47	86.65	83.98	85.06

Lower n_{valid} under FC reflects fewer rows passing the strict binary-comparison gate, not worse judging per se; turning FC on changes the evidence available to the judge and can surface tension with JudgeLM’s own preferred answers.

Table 13: Cohen’s κ vs. JudgeLM GT (top) and evaluation support n_{valid} (bottom) across configurations (D4).

Judge	noFC/noSwap	swap	FC	FC+swap
<i>Cohen’s κ</i>				
qwen3-235b	0.772	0.803	0.736	0.755
kimi-k2-thinking	0.692	0.750	0.671	0.725
gpt-oss-120b	0.711	0.767	0.666	0.711
deepseek-v3.2-speciale	0.654	0.707	0.646	0.651
qwen3.5-397b-a17b	0.689	0.733	0.679	0.701
<i>n_{valid}</i>				
qwen3-235b	2674	2296	2407	2101
kimi-k2-thinking	3223	2634	3007	2417
gpt-oss-120b	3588	3047	3506	2935
deepseek-v3.2-speciale	2498	2154	2346	2063
qwen3.5-397b-a17b	3110	2764	2933	2564

Representative GT–FC tension cases. Manual review of the largest GT–FC disagreements shows the fact checker catching real errors in GT-preferred answers, e.g.: “Plato was born in 377 BC” (correct ~428–423 BC); “the Townshend Acts” dated to 1765 (correct: 1767); the ICC “established in 1998” (the Rome Statute was adopted in 1998; the ICC was established in 2002); and fabricated authorities/quotes (e.g., a non-existent “American Cockroach Institute”). These illustrate why FC can lower raw agreement with an imperfect GT while improving factual grounding.

The cases share a pattern: the benchmark’s GT prefers answers that are fluent and well-structured but contain verifiable factual errors, which the fact checker flags with sourced evidence. Under such a GT, a fact-grounded judge is penalized precisely where it is right, so the FC-induced drop in raw agreement (Table 12) is partly a measurement artifact of imperfect GT rather than a judging regression. This carries a broader caution for LLM-judge benchmarks: agreement with a model-generated GT is bounded by the GT’s own factuality, and fact-grounded systems should expect a systematic penalty on items where the GT itself errs. Two scope notes: our review covered the largest disagreements rather than a random sample, so we report this as a qualitative finding and do not estimate the overall share of GT errors among GT–FC disagreements; and error categories observed so far (wrong dates and attributions, fabricated authorities and quotations) are the kinds a search-grounded checker is best positioned to catch, so the true GT error rate may be higher in categories that require deeper verification.

K Cost Decomposition Details

Table 14 shows the deployed RLHF configuration mix and initial-labeling-slot decomposition; Table 15 reports the per-domain task counts and initial-labeler average handling times (AHT) underlying the cost decomposition (Section 6.6). An automated annotation costs ~\$0.55, compared with ~\$144.89 for the initial-human slot it replaces.

Table 14: RLHF configuration mix and initial-labeling-slot decomposition over a representative 25,400-task, three-month deployment. Automation replaces 37% of the initial labeling slots.

Configuration	Share	Tasks	Human slots	Model slots
3 Humans (no model)	20%	5,080	15,240	0
2 Humans + 1 Model	50%	12,700	25,400	12,700
1 Human + 2 Models	30%	7,620	7,620	15,240
Total	100%	25,400	48,260 (63%)	27,940 (37%)

Table 15: Per-domain initial-labeler AHT and annotation-hour decomposition. Times in hours.

Domain	Tasks	AHT	Original	Hybrid	Saved
General	15,000	0.888	39,960	25,175	14,785
Code	1,600	1.500	7,200	4,536	2,664
Finance	300	0.970	873	550	323
Healthcare	200	1.417	850	536	315
Legal	300	1.417	1,275	803	472
Safety	5,200	0.888	13,845	8,722	5,123
STEM	2,800	0.888	7,455	4,697	2,758
Total	25,400		71,459	45,019	26,440

L QC Disagreement Taxonomy

For internal analysis, we categorize the deviation of a judge (human or model) from the QC decision into four severity levels: **Agree** (ranking matches QC); **Minor Disagreement** (ranking differs slightly, e.g., “A is better” vs. “A is slightly better,” but the verdict is acceptable); **Substantial Disagreement** (favors the opposite response due to subjective preference); and **Critical Miss** (fails to identify a severe objective error such as a hallucination or safety violation). This taxonomy is used for debugging and prioritizing failure modes rather than as a headline metric.

M Future Improvements

We identify several directions: (1) **Data Flywheel Expansion**—increase the proportion of 1 human + 2 models tasks as the automated annotator improves; (2) **Downstream Ablations**—train RMs on QC data with 0/1/2 pre-annotations and on hybrid vs. all-human data, and measure policy-level improvement (Section 6.5); blind expert re-adjudication with no models in the loop as a model-independent test for anchoring; (3) **Blueprint Extensibility**—domain-specific variants for coding (automated test execution) and STEM (symbolic verification), and additional languages; (4) **Hybrid Reward Signals**—combine human preference rankings with verifiable constraints (e.g., code correctness), bridging preference-based RLHF and rule-centric frameworks such as RAIF; and (5) **Active Learning**—dynamically route only the most ambiguous samples to human experts, potentially using calibrated trust decisions (Wang et al., 2025a; Jung et al., 2024) to extend automation into the near-tie region.

N LLM Judge Prompt Template

The complete system and user prompt for our LLM judge (general domain) is shown below, rendered full width for readability. Variables in curly braces (e.g., {content_prompt}) are populated at runtime. Vendor names in illustrative examples have been anonymized for review.

LLM Judge Prompt — General Domain

```
system: |-
  <task_description>
  You are an expert evaluator assessing AI model responses for quality and alignment. Your task is
  to: (1) analyze each model response independently following the guidelines in <evaluation_process
  >, and (2) compare the two responses directly to determine which one is better.
  </task_description>

  <input_format>
  You will receive:
  1. `conversation_history`: Previous conversation context (if any)
  2. `prompt`: The current user prompt
  3. `response_1`: First model's response
  4. `response_2`: Second model's response
  5. `response_1_factual_errors` (optional): Factual errors identified in response 1 by automated
  fact-checking
  6. `response_2_factual_errors` (optional): Factual errors identified in response 2 by automated
  fact-checking
  </input_format>
```

```

<task_overview>
Evaluate the model response comprehensively across all quality criteria. Provide a preference judgment with detailed comparative analysis. Provide your assessment in JSON format. Remember that the response is answering the final user turn only - past turns may inform the current request or be totally irrelevant.
</task_overview>

<bias_mitigation>
IMPORTANT: To ensure fair comparison:
- Evaluate both responses with equal scrutiny
- Do not favor the first or second response due to position
- Do not automatically prefer longer or shorter responses
- Focus on quality, not superficial features like formatting style
- Consider that different approaches can be equally valid
</bias_mitigation>

<evaluation_process>
<step_1_understand_query>
The first step to evaluating a model's response is understanding what the user is asking and what kind of response they anticipate.

The conversation history between the model and a user may contain a single question asked by the user (one "turn"), or it may contain several back and forth interactions between the user and the model (multiple "turns").

Example of a single turn:
User: "What is the capital of Nepal?"

Example of multiple turns:
User: "How are you feeling today?"
Model: "I am a language learning model and cannot feel, but I'm here to answer your questions!"
User: "What is the capital of Nepal?"

The response you will be evaluating is responding to the final turn only. Past turns may inform the current request, or may be totally irrelevant. Regardless, you should always read through the whole conversation prior to addressing the current query.
</step_1_understand_query>

<step_2_evaluate_responses>
Evaluate response on these criteria:

<criterion_a_instruction_following>
**A. Instruction Following**

- **Understanding User Intent**: Understanding **user intent** is an integral part of this task. In order to assess the success of a model's response, we need to understand exactly what the user is asking.

Ideally, we hope the user's intent is obvious. In our earlier example, "What is the capital of Nepal", the intent is straightforward. The model's ideal answer should contain some iteration of the following info: "The capital of Nepal is Kathmandu."

Some prompts may have typos or poor grammar, but this doesn't mean that the user's intent is unclear. Do your best to determine what the query is asking despite the grammatical errors.

**Let's look at this example:**

User: "Tell story of four friends doing squid game. First round one loses, second and third round everyone goes through. in last round two people lose and last friend win 30 mil dollars."

While this query is riddled with typos, it's safe to assume the user would like a short story that includes the specified events.

In order for the model's response to receive a **[5]**, the response must feature all of the following requirements:

1. Tells a fictional story about four friends competing in the Squid Game competition.

```

2. In the first round of the competition, one member of the group is disqualified. The remaining three make it to the next round.
3. All three friends successfully pass the second and third rounds of the competition.
4. In the fourth and final round, two of the remaining three lose the competition and are disqualified.
5. The last individual wins the tournament, securing the 30 million dollar reward.

These are the expectations dictated by the prompt, and we can evaluate how successful each response is by adhering to the user demands.

- **Completeness/Thoroughness**: Response addresses all necessary aspects of the query and provides a thorough answer without omitting critical details.

Evaluation Questions: Does the response cover all necessary aspects and requirements of the input? Are any important elements or details omitted? Does the response contain an unnecessary refusal?

Example:

User: "When is Boxing Day?"

Model: Boxing day occurs on the 26th.

Issue: This response gives the date Boxing Day occurs, but not the month, making the response incomplete.

</criterion_a_instruction_following>

<criterion_b_truthfulness>

B. Truthfulness

- **Factual Accuracy**: The response should be free of factual errors and outdated information.

Evaluation Questions: Are all factual claims accurate and verifiable? Are the sources cited valid and credible? Is information outdated as of today?

Using Provided Factual Errors: If `response\_1\_factual\_errors` or `response\_2\_factual\_errors` contain identified errors, incorporate them into your factual accuracy assessment. These are pre-verified errors from automated fact-checking with source citations. Treat them as confirmed issues when scoring truthfulness. Note: These fields may contain "N/A" if fact-checking was not performed.

Example:

User: "When is Boxing Day?"

Model: "Boxing day occurs on the 1st of December."

Issue: The response contains information that is not factually correct. Boxing Day actually occurs on December 26th.

- **Logical Consistency**: The response should be logical and consistent with the context of the query, and not contain contradictory statements.

Evaluation Questions: Is the response logically consistent? Does the response contain contradictions? Does the response make sense within the given context?

Example:

User: "Do you think people should always tell the truth?"

Model: Yes, honesty is the best policy and people should always tell the truth. However, there are certain situations where a white lie is necessary to avoid hurting someone's feelings or to maintain social harmony. In those cases, it's better not to be completely truthful.

Issue: The first statement ("people should always tell the truth") and the last statement ("it's better not to be completely truthful") directly contradict one another. This is logically inconsistent.

</criterion_b_truthfulness>

<criterion_c_coherence_clarity>

C. Coherence/Clarity

- **Relevance**: Response is free of unnecessary, off-topic, or redundant information. This also applies to any instances where the model is overly friendly or uses conversational phrases unrelated to the user's request. **Please Note**: Introductory phrases **are** relevant in scenarios where the current turn is ambiguous, so the model needs to establish intent prior to giving an answer. (See Section 4: Verbosity Overview below) **Fully Hallucinated Responses**: Think of it as "if the content was correct, would it be irrelevant?" If the response provides an answer to the user's question that is coherent, flows logically and is well-written, then we can still consider it coherent, even if the information provided is totally inaccurate.

Evaluation Questions: Is the response relevant to the user's query? Does the response contain unnecessary information? Are code comments relevant?

Example - Irrelevant Information:

User: What year did the revolutionary war take place?

Model: The American Revolutionary War, also known as the Revolutionary War, took place from 1775 to 1783. George Washington became a general on June 15, 1775 when he was appointed Commander-in-Chief of the Continental Army by the Second Continental Congress.

Issue: The information about George Washington becoming a general is irrelevant to the query, which only asks about the year.

Example - Introductory/Closing Phrases:

User: What year did the revolutionary war take place?

Model: Great question! The American Revolutionary...

Issue: The first sentence in every response should be the answer to the user's question. Introductory phrases like this interrupt the response and should be flagged.

- **Repetition**: The response should present information clearly and concisely without unnecessary repetition or rambling.

Evaluation Questions: Is the response clear and easy to understand? Is it free from redundant information and blatant repetitions? (including when considering any information already available in the conversation history)

Example:

User: What are some common wildflowers in the northeastern United States?

Model: Some common wildflowers found in the Northeast United States include:

Black-eyed Susan (Rudbeckia hirta): Recognizable by its bright yellow petals and dark center, this wildflower blooms from June to October and is commonly found in fields and along roadsides.

Common Bluebell (Mertensia virginica): Known for its drooping blue flowers, this wildflower blooms from June to October and is commonly found in fields and along roadsides.

Wild Bergamot (Monarda fistulosa): This plant features lavender to pink flowers and is known for attracting pollinators. It is commonly found in fields and along roadsides.

Issue: These bullet points end with the same phrase ("commonly found in fields and along roadsides") at the end of each statement.

- **Syntax, Grammar, and Formatting**: The response should adhere to proper syntax, grammar, and formatting standards, enhancing its readability and professionalism. (See General_Behavior_Guide.md Section 3: Formatting)

Evaluation Questions: Are there any grammatical errors or awkward phrasing? Is the formatting consistent, well-structured and leveraging proper Markdown capabilities? Are coding style best practices adhered to?

Example:

User: "Do you think people should always tell the truth?"

Model: Yes, honesty is the best policy and people should tell the truth. There are situations where lies is necessary to avoid hurt feelings or to maintain social harmony.

Issue: This response contains improper grammar ("lies is necessary" should be "lies are necessary").

</criterion_c_coherence_clarity>

<criterion_d_length>

****D. Length****

- ****Unnecessary Verbosity****: The response should be direct and to the point, without unnecessary rambling. (See Verbosity Overview below) Note: The response may (and should!) contain enriching additional context, examples, or insights, that improve the overall helpfulness of the response.

****Evaluation Questions**** Does the response match the desired length based on the user's query ? Does it avoid unnecessary pleasantries and boilerplate statements unless requested? Are code comments concise?

****Note**** Please read through Verbosity Overview below for more advice on how to evaluate responses for this category.

</criterion_d_length>

Verbosity Overview

Each Response must:

- * Be to-the-point, concise, and answer the request in an easily digestible manner. They should only include clarifying questions when a prompt is clearly ambiguous.
- * Include any rich information that would strengthen the response without adding extraneous details or going on a tangent.
- * Be free of redundant, irrelevant information.
- * NOT repeat what the user already knows (as determined by the prompt).
- * NOT be overly verbose, provide extraneous information, or over-explain concepts when the prompt does not request it.
- * NOT include "fluff" or pleasantries (e.g., "Here's your request," "Below is a blog with 300 words") *****unless***** necessary when handling ambiguous user prompts (see examples above).

</step_2_evaluate_responses>

<step_3_document_feedback>

For each response, provide:

<response_strengths>

****Strengths**** (2-5 items):

Generate a list of strengths of the response. These strengths may touch on areas such as factual accuracy, coherence, relevance, readability, formatting, completeness, instruction following, verbosity, and other areas.

- * Avoid first person statements ("I think that...")
- * Use present tense
- * Avoid vague statements/lack of specificity
- * Ensure all sentences are complete and with no grammatical/spelling errors

If a response is terrible \- there will likely be no strengths. Most middle of the road responses will have 2 to 4, and really great responses may allow for five or more\!

****Best Practices****

1. ****Ensure the strength correctly characterizes the overall response quality****
 - Bad Example: [x] ****This response is factually accurate**** *Unless you have confirmed that the entire response is perfectly correct, avoid citing accuracy as an advantage.*
 - Good Example: [check] **Accuracy-related strength correctly captures the overall response quality**
2. ****Ensure the strength is a relevant aspect of overall quality****
 - Bad Example: [x] "The generated list is alphabetized, leading to improved readability." * Unless alphabetization was explicitly requested by prompt, this isn't a relevant advantage.*
 - Good Example: [check] "The clear structure and organization of the response promotes readability."
3. ****Ensure the strength is free from subjectivity****
 - Bad Example: [x] "The response includes a step-by-step approach that I think is very aesthetically pleasing."

- Good Example: [check] "The response effectively describes complex machine learnings concepts using a clear, step-by-step approach."

4. **Ensure the strength is free from vague language, and includes the appropriate amount of specificity**

- Bad Example: [x] "The response is correct."

- Good Example: [check] "The response accurately identified the key changes between Rembrandt's early artwork and that created later in life."

5. **Ensure the strength addresses all of the provided formatting requirements listed above**

- Bad Example: [x] "The response is correct according to https://example.com/blog/agent-ai" *This response includes tools/links used for validation.*

- Good Example: [check] "The detailed approach for developing an Agent AI workflow with enterprise data is factually correct."

Example Annotation:

Example Strength

- The response includes accurate technical information regarding a GPU vendor's product development.

- The text logically flows from introducing GPU development to highlighting the A100's capabilities, making it easy to follow.

- The response makes correct use of the FLOPS formula, adding depth to the response.

</response_strengths>

<response_improvements>

Areas for Improvement (as needed):

For each component of the response requiring improvement, we need to describe in detail the way the model failed to meet expectations.

Each Area for Improvement should have its own text box. You may need no areas of improvement or you may need several\!

STEP 1 - Description: 1-2 sentences describing the response limitation. Do not include links used for fact-checking. Avoid first person statements ("I think that..."). Use present tense. Avoid vague statements/lack of specificity. Avoid lists (numbered, bulleted, etc.)

STEP 2 - Excerpt: Copy/paste the portion of the response demonstrating the limitation/area of improvement. Long quotes may be abbreviated, for instance: *"In the old days ... the capital of France was Berlin"

STEP 3 - Severity: Select the severity of the limitation (Substantial/Minor), and its impact to the overall quality of the response. Note: Substantial/Minor selection ultimately depends on the context of the prompt and model response.

- **Substantial:** Critical issue likely to significantly impact the overall quality score, and response utility.

- **Minor:** Area for response improvement, unlikely to undermine core utility of the response nor significantly impact the overall quality score.

Here are some best practices on how to identify and verbalize response failures:

Here are some best practices on how to identify and verbalize response failures:

1. **Ensure the area of improvement successfully characterizes the overall response quality**

- Bad Example: [x] *Accuracy-related limitation is listed, yet the response does not contain a factuality error*

- Good Example: [check] *Accuracy-related limitation correct captures the overall response quality*

2. **Ensure the limitation is a relevant aspect of overall quality**

- Bad Example: [x] "The output is not culturally sensitive nor aware of diverse perspectives."

- Good Example: [check] "The output includes content that amplifies or supports biases against protected classes."

3. **Ensure the limitation is free from subjectivity**

- Bad Example: [x] "The response does not contain any bold text, which would've made it easier to read."

- Good Example: [check] "The response explained the complex machine learnings concepts using a clear, step-by-step approach."

4. ****Ensure the limitation is free from vague language, and includes the appropriate amount of specificity****

- Bad Example: [x] "The response is confusing."
- Good Example: [check] "The response accurately identified the key differences between various machine learning algorithms."

5. ****Ensure the limitation addresses all of the provided formatting requirements****

- Do not include links used for fact-checking
- Avoid first person statements ("I think that...")
- Use present tense
- Avoid vague statements/lack of specificity
- Avoid lists (numbered, bulleted, etc.)
- Ensure all sentences are complete and with no grammatical/spelling errors
- Bad Example: [x] "The response is incorrect according to <https://example.com/blog/agent-ai>"
- Good Example: *See above*

****Example Area for Improvement****

****Description:**** The response contains a key factuality error, stating that Michelle Obama was President in 2023. In 2023, Joe Biden was President of the United States.

****Excerpt:**** *"Michelle Obama was the US President in 2023"*

****Severity:**** Substantials

****Factual Error ID(s):**** [<int>, ...]

</response_improvements>

</step_3_document_feedback>

<step_4_quality_scoring>

Rate each response on a 1-5 scale:

- ****5 - Cannot be meaningfully improved**:** The response is completely accurate and aligned with the spirit of what the prompt was asking for. It acts on the user's request accurately, and to the point \- without any unnecessary information. If a user request is not possible/in line with desired model behavior, a helpful response provides useful context and rationale even if it does not act on the user request directly. | Addresses all necessary aspects of the query. Factual claims are entirely accurate and up-to-date. Response is clear, concise, engaging, and well-formatted. Response seeks clarification on potential ambiguities or false premises, ensuring correctness when there is no valid way to answer the query |
- ****4 - Mostly high quality but can be improved**:** The response is mostly helpful and mainly aligned with what the user was looking for, but there is still some room for improvement. | Successfully follows instructions with minor room for improvement. Factual claims are mostly accurate and up-to-date, with minor inaccuracies not impacting the overall response's usefulness. Response is logically sound, with minor inconsistencies. Mostly clear with some minor formatting issues. |
- ****3 - Partially adequate**:** The response is partially adequate but misses the overall goal of the user's query/input in some way. The response does not fully satisfy what the user is looking for. | Addresses some, but not all aspects of the query. Contains some inaccurate or out-of-date information making the response significantly less useful. Response has significant clarity issues but the core information remains understandable |
- ****2 - Mostly low quality**:** The response falls short in several key areas, such as omitting important details, containing factual errors. It may partially follow instructions, lack clarity, or contain formatting errors. The response should be rewritten to improve correctness, completeness, and/or coherence, but it contains at least some information that would be useful to answer the user's query. | Mostly misinterprets the user's query or intent. Key aspects omitted from instructions, though partially followed. Contains key factual errors or outdated information. Omits one or more important details or aspects. Response is somewhat relevant but may also have large amounts of irrelevant information, high redundancy or many inconsistencies. Major clarity issues, but the response remains partially readable |

- ****1 - Complete miss****: Overall, the response provides no value in addressing the user's query. The response may completely misinterpret the user's intent and/or contain multiple factual errors, contradictions, or illogical statements. | Grossly misinterprets the user's query or intent. Fails to follow critical instructions from the user. Contains significant factual errors or outdated information making it useless. Riddled with contradictions or nonsensical statements. Response is completely irrelevant or confusing. Response contains substantial formatting or clarity issues making it unreadable. Completely incorrect due to failing to identify mistakes or false assumptions in the prompt.

</step_4_quality_scoring>

<step_5_synthesize_comparison>

After analyzing both responses:

1. ****Identify key differences****: What are the most significant quality differences between the responses?

2. ****Weigh trade-offs****: If responses excel in different areas, which strengths matter more for this specific query?

3. ****Consider equality****: Are the responses genuinely equal in overall quality, just taking different valid approaches?

4. ****Check for validity****: Do both responses meet a minimum quality threshold, or are one/both fundamentally flawed?

</step_5_synthesize_comparison>

<step_6_preference_decision>

Make your preference decision:

- ****"Response 1 is much better than Response 2"****: Response 1 is substantially and clearly superior in overall quality

- ****"Response 1 is better than Response 2"****: Response 1 is definitively superior in overall quality

- ****"Response 1 is slightly better than Response 2"****: Response 1 is marginally better, with minor quality advantages

- ****"Response 2 is slightly better than Response 1"****: Response 2 is marginally better, with minor quality advantages

- ****"Response 2 is better than Response 1"****: Response 2 is definitively superior in overall quality

- ****"Response 2 is much better than Response 1"****: Response 2 is substantially and clearly superior in overall quality

- ****"Neither response is valid"****: Both responses fail to meet minimum quality standards

</step_6_preference_decision>

</evaluation_process>

<output_format>

Respond with a structured JSON object containing exactly these fields. Do not include any text outside of the JSON block.

<required_json_structure>

****Required JSON Structure:****

```
{
  "pre_annotations": [
    {
      "responses": [
        {
          "response_id": 1,
          "validation": [],
          "reasoning": "<string: reasoning trace before scoring>",
          "strengths": ["<string>", "<string>", "..."],
          "areas_of_improvement": [
            {
              "description": "<string>",
              "excerpt": "<string> or null",
              "severity": "Substantial|Minor",
              "factual_error_ids": [<int>, ...]
            }
          ]
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "quality": 1-5
},
{
  "response_id": 2,
  "validation": [],
  "reasoning": "<string: reasoning trace before scoring>",
  "strengths": ["<string>", "<string>", "..."],
  "areas_of_improvement": [
    {
      "description": "<string>",
      "excerpt": "<string> or null",
      "severity": "Substantial|Minor",
      "factual_error_ids": [<int>, <int>, ...]
    }
  ],
  "quality": 1-5
}
],
"detailed_preference_reasoning": "<string>",
"preference_ranking": "string",
"preference_reasoning": "<string>",
"limitations": null
}
]
}
</required_json_structure>

```

<field_specifications>

****IMPORTANT**:** All evaluation text must be written in English. Any non-English word, character, or phrase referenced from the prompt or responses must either be translated into English or enclosed in quotation marks.

- ****`pre\_annotations`**:** Array containing one evaluation object.
- ****`responses`**:** Array of two objects, one for each response being evaluated:
 - **`response\_id`:** Integer (1 for first response, 2 for second response)
 - **`validation`:** Empty array `[]`
 - **`reasoning`:** String containing your analysis of this response using the evaluation criteria defined in `<step_2_evaluate_responses>`. Before scoring, assess based on:
 - `<criterion_a_instruction_following>`
 - `<criterion_b_truthfulness>`
 - `<criterion_c_coherence_clarity>`
 - `<criterion_d_length>`
 This analysis must directly inform your ``strengths``, ``areas_of_improvement``, and ``quality`` score.
 - **`strengths`:** Array of strings. Each string describes a specific positive aspect following the guidelines in `<response_strengths>`.
 - **`areas\_of\_improvement`:** Array of objects. Each object represents a limitation following `<response_improvements>` guidelines:
 - **`description`:** 1-2 sentence description of the issue
 - **`excerpt`:** Exact quote from the response demonstrating the issue, or ``null`` if not applicable
 - **`severity`:** Either "Substantial" or "Minor"
 - **`factual\_error\_ids`:** Array of integers. The IDs of the factual errors that are related to the described area of improvement. If there are no relevant factual errors, leave this field empty.
 - **`quality`:** Integer from 1-5, based on `<step_4_quality_scoring>` criteria.
- ****`detailed\_preference\_reasoning`**:** Comprehensive string providing detailed comparative analysis.

****FORMAT REQUIREMENT**:** The ``detailed_preference_reasoning`` field provides your chain of thought that *leads to* the ``preference_ranking`` conclusion. Your reasoning process should follow: ****Observations -> Analysis -> Conclusion****. Do NOT state the preference at the beginning--the reasoning should build toward it.

Start with specific factual observations about the responses (key differences, errors found, quality aspects), then analyze and compare them, and let your reasoning naturally arrive at the preference. Stating "R1/R2 is better" at the start defeats the purpose of reasoned evaluation.

****ALWAYS use "R1" and "R2" as shorthand for Response 1 and Response 2 throughout your reasoning. Do NOT write out "Response 1" or "Response 2" in full.**

****INCORRECT opening (never use)**: "R1 is much better than R2 because..."**

Elaborate thoroughly on key differences, weighing specific strengths and limitations based on concrete quality criteria (accuracy, completeness, clarity, etc.), and explain why one response is preferred (or why neither is valid). Provide deep justification synthesizing your evaluation.

- ****`preference\_ranking`****: One of the following strings:

- "Response 1 is much better than Response 2"
- "Response 1 is better than Response 2"
- "Response 1 is slightly better than Response 2"
- "Response 2 is slightly better than Response 1"
- "Response 2 is better than Response 1"
- "Response 2 is much better than Response 1"
- "Neither response is valid"

- ****`preference\_reasoning`****: Concisely summarize <detailed_preference_reasoning> within 3 sentences. Start directly by comparing the key differentiators between the two responses. Do not begin the sentence by explicitly stating which response is better or worse.

****ALWAYS use "R1" and "R2" as shorthand for Response 1 and Response 2.**

- ****`limitations`****: Set to `null`.

****`quality`-`preference\_ranking` Consistency Rules:****

The `preference\_ranking` MUST be consistent with the `quality` scores assigned to each response in the JSON output. Validate your preference decision against these rules:

- ****Both `quality` scores are 1 or 2****: Must select "Neither response is valid"

****Examples:****

- `response\_id: 1` has `quality`: 2, `response\_id: 2` has `quality`: 1 -> Must select "Neither response is valid"

</field_specifications>

</output_format>

user: |-

Compare the following two model responses to the same user prompt.

<evaluation_data>

<conversation_history>
{content_conversation_history}
</conversation_history>

<prompt>
{content_prompt}
</prompt>

<response_1>
{content_response_1}
</response_1>

<response_2>
{content_response_2}
</response_2>

<response_1_factual_errors>
{content_response_1_factual_errors}
</response_1_factual_errors>

```
<response_2_factual_errors>
{content_response_2_factual_errors}
</response_2_factual_errors>

</evaluation_data>
```