MITIGATING RUBRIC INTERFERENCE IN LLM JUDGES VIA ON-POLICY SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM judges increasingly evaluate responses against fine-grained rubric checklists. Evaluating all rubrics in a single pass is efficient, but we find it introduces *rubric interference*: the verdict on one rubric shifts depending on which other rubrics are co-evaluated. In a preliminary study, only one-third of samples receive fully consistent verdicts across rubric sets of varying composition. We formalize this problem and develop a measurement framework that probes interference through three controlled operations: rubric set expansion, reordering, and noise injection. To mitigate interference without external supervision, we propose **Self-Anchored Rubric Alignment (SARA)**. SARA uses a model's own single-rubric judgments as stable anchors and aligns multi-rubric outputs to these anchors through on-policy self-distillation. We validate SARA on three datasets (HealthBench, FLASK, ResearchQA) across Qwen3 and Llama-3.1 model families. SARA improves rubric-level consistency by up to 16% and triples sample-level exact match, while preserving agreement with both the base model and GPT-4.1. We further show that judge consistency directly affects downstream reinforcement learning: a SARA-aligned reward model reduces inter-run reward variance by 5$\times$ under rubric-order perturbation, confirming that consistent rubric evaluation is a practical requirement for stable policy optimization. Our code is available at https://anonymous.4open.science/r/SARA-22E2/

1 INTRODUCTION

Large language models now serve as automated evaluators for open-ended text generation, replacing or supplementing human annotation in benchmarking and reward modeling (Zheng et al., 2023; Kim et al., 2024). To move beyond holistic scoring and pairwise comparison, recent work decomposes evaluation into rubric-based checklists that assess specific, verifiable quality dimensions (Lee et al., 2025; Kim et al., 2024; Pathak et al., 2025).

Many tasks require evaluating a single response against multiple rubrics simultaneously. HealthBench (Arora et al., 2025) defines up to 32 rubrics per medical conversation; ResearchQA (Li et al., 2025b) applies variable rubric sets to scientific answers. *Isolation evaluation* processes each rubric in a separate call, producing independent verdicts. *Joint evaluation* assesses all rubrics in a single pass. Joint evaluation is attractive for two reasons: it reduces inference cost proportional to rubric count, and it produces a coherent multi-dimensional report in one call. The second property is especially valuable when the judge operates as an online reward model for reinforcement learning (Gunjal et al., 2026), where per-step latency directly constrains training throughput. Current practice defaults to isolation mode (Lee et al., 2025), and the reliability of joint evaluation remains largely unexplored.

For joint evaluation to reliably replace isolation calls, each rubric's verdict must depend only on the conversation and that rubric's definition. We find that this invariance does not hold. Joint evaluation introduces *rubric interference*: the verdict on a target rubric shifts depending on the composition and ordering of co-present rubrics (Figure 1). On HealthBench, comparing isolation and joint verdicts, even Qwen3-32B achieves only 36% sample-level exact match. This reflects systematic distortion rather than inference noise: all models achieve $\geq$0.98 self-agreement across repeated evaluations with identical inputs (temperature 0.7, 5 seeds).

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.

A key observation enables our solution. While joint evaluation is unreliable, the same model evaluating each rubric in isolation produces stable judgments without any external annotation. The gap between isolation and joint verdicts reveals exactly where interference distorts reasoning, providing a self-contained training signal.

We propose **Self-Anchored Rubric Alignment (SARA)**, which aligns joint-mode reasoning with the model's own isolation judgments through on-policy self-distillation. SARA generates a multi-rubric evaluation, parses it into per-rubric segments, and distills from a slowly updated teacher that evaluates each segment in isolation context. A divergence loss corrects segments where interference distorts reasoning; a preservation loss maintains overall verdict structure.

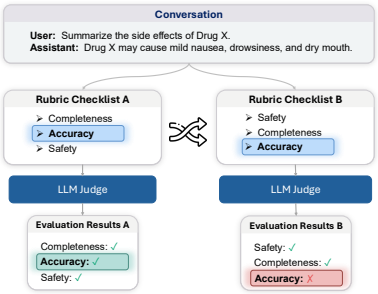


Figure 1: Rubric interference in joint evaluation. The same rubric receives different verdicts depending on the composition of co-present rubrics.

Our contributions:

1. We identify rubric interference as a systematic failure mode in multi-rubric LLM evaluation and develop a measurement framework with three controlled operations to quantify it.
2. We propose SARA, a self-distillation method that mitigates interference by anchoring joint-mode reasoning to the model's own isolation judgments, requiring no external supervision.
3. We validate SARA on three datasets and two model families. SARA improves consistency while preserving evaluation quality relative to both base models and the reference model.
4. We show that SARA's consistency gains propagate to downstream RL: a SARA-aligned reward model reduces inter-run reward variance by 5× under rubric-order perturbation, demonstrating the practical value of consistent multi-rubric evaluation.

2 Problem Definition

2.1 Joint Rubric Evaluation

A judge model $\mathcal{M}$ evaluates a conversation c against a rubric list $R = (r_1, \ldots, r_n)$, assigning each rubric a verdict v_i from a discrete scale $\mathcal{V}$ (binary $\{0, 1\}$ or scored e.g. $\{1, \ldots, 5\}$).

Isolation evaluation processes each rubric in a separate call: $v_i^{\text{iso}} = \mathcal{M}(c, (r_i))$. **Joint evaluation** assesses all rubrics in one pass: $v_i^{\text{jnt}} = \mathcal{M}(c, R)[i]$. Joint evaluation reduces inference cost by a factor of $|R|$ but exposes each verdict to the influence of co-present rubrics.

Definition 2.1 (Rubric Interference)**.** A judge $\mathcal{M}$ exhibits **rubric interference** on conversation c and rubric r if there exist two rubric lists R, R' with $r = R[i] = R'[j]$ such that $\mathcal{M}(c, R)[i] \neq \mathcal{M}(c, R')[j]$.

An interference-free judge satisfies:

$$\mathcal{M}(c, R)[i] = \mathcal{M}(c, R')[j] \quad \forall\, R, R' \text{ s.t. } R[i] = R'[j] \tag{1}$$

This requires that each verdict depends only on c and the target rubric, not on the composition, ordering, or size of R. We decompose this requirement into testable conditions below.

Operation	Manipulation	Property tested	Comparison
Expansion	$\lvert R\rvert$: $1 \to n$	Set-size invariance	v_i^{iso} vs. v_i^{jnt}
Reordering	Permute R	Order invariance	v_i across $\pi(R)$
Noise	$R \to R \cup R'_{\text{irr}}$	Irrelevance robustness	v_i before vs. after

Table 1: Three operations for testing rubric interference. Each targets a specific violation of the invariance property (Eq. 1).

2.2 Testing for Rubric Interference

We probe Equation 1 through three controlled operations. Each modifies the rubric set while keeping the conversation and target rubric fixed, isolating a specific violation of invariance.

Expansion is our primary test. We compare each rubric's isolation verdict against its joint verdict and report *overall consistency* (full set) and *consistency-at-K* (random subsets of size K) to trace how interference scales with rubric count, including intermediate subset sizes. **Reordering** evaluates the same set under multiple permutations, isolating positional effects from content-based interference. **Noise injection** appends rubrics from unrelated conversations, testing whether the model can ignore irrelevant evaluation context.

For all operations, we report rubric-level agreement (Agr), Cohen's κ (weighted for scored formats), and sample-level exact match (EM). Metric definitions are in Appendix A.

3 Self-Anchored Rubric Alignment

3.1 Overview

SARA trains a judge to resist rubric interference through on-policy self-distillation. The core idea is simple: the same model that struggles with joint evaluation judges each rubric reliably in isolation. These isolation judgments require no external annotation, making them natural training anchors.

At each step, the student generates a joint evaluation and we parse it into per-rubric segments. A slowly updated teacher evaluates each segment under isolation context, producing anchor logits free of interference. A divergence loss aligns the student's joint-mode logits with these anchors on analysis tokens; a preservation loss maintains structural tokens (verdicts, formatting). Figure 2 shows the pipeline.

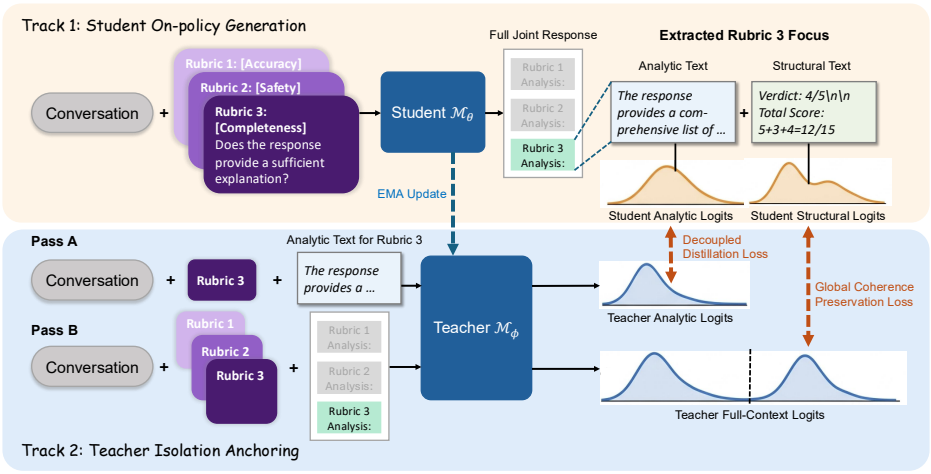


Figure 2: SARA training pipeline. The student generates a joint evaluation (top), parsed into per-rubric segments. The teacher evaluates each segment under isolation context (bottom), producing interference-free anchor logits. The distillation loss aligns student and teacher on analysis tokens; the preservation loss maintains structural tokens.

3.2 On-Policy Generation and Segment Extraction

At each step, the student $\mathcal{M}_\theta$ generates a joint evaluation for conversation c and rubric list $R = (r_1, \ldots, r_n)$:

$$y \sim \mathcal{M}_\theta(\cdot \mid c, R) \tag{2}$$

The output y follows a repeating structure $y = [a_1, s_1, \ldots, a_n, s_n]$, where a_i is the free-form analysis for rubric r_i and s_i contains the verdict and formatting tokens. We parse y into two complementary token masks:

$$\mathbf{m}_t^{\text{ana}} = \begin{cases} 1 & t \in \bigcup_i \mathcal{A}_i \\ 0 & \text{otherwise} \end{cases}, \qquad \mathbf{m}^{\text{str}} = \mathbf{1} - \mathbf{m}^{\text{ana}} \tag{3}$$

where $\mathcal{A}_i$ indexes token positions in analysis segment a_i. The analysis mask marks tokens susceptible to interference; the structure mask covers verdicts and formatting that coordinate across rubrics. Because y is sampled from the current policy (on-policy), the training distribution tracks the model's evolving behavior.

3.3 Isolation-Anchored Distillation

Teacher. The teacher $\mathcal{M}_\phi$ is an exponential moving average (EMA) of the student, updated after each step:

$$\phi \leftarrow \beta_{\text{ema}}\, \phi + (1 - \beta_{\text{ema}})\, \theta \tag{4}$$

A high decay rate ($\beta_{\text{ema}} = 0.999$) provides temporal smoothing and prevents the training signal from oscillating with each update. Using a separate EMA copy rather than the student's own isolation logits also breaks a potential degenerate loop: if the student anchored on itself, systematic bias would reinforce without correction.

Distillation loss. For each rubric r_i, we obtain two distributions over the same analysis tokens a_i under different contexts:

$$\begin{aligned} \mathbf{p}_t &= \mathcal{M}_\theta(y_t \mid y_{<t},\ c,\ R) && \text{(student, full context)} \\ \hat{\mathbf{q}}_t^{(i)} &= \mathcal{M}_\phi(a_{i,t} \mid a_{i,<t},\ c,\ \{r_i\}) && \text{(teacher, isolation context)} \end{aligned} \tag{5}$$

The student conditions on all rubrics R and may exhibit interference. The teacher sees only the target rubric r_i and produces interference-free logits. We align them with a symmetric Jensen-Shannon divergence across all analysis tokens:

$$\mathcal{L}_{\text{distill}} = \frac{1}{|\mathcal{A}|} \sum_{i=1}^{n} \sum_{t \in \mathcal{A}_i} \text{JSD}_{\beta_d}\left(\mathbf{p}_t \parallel \hat{\mathbf{q}}_t^{(i)}\right) \tag{6}$$

where $\mathcal{A} = \bigcup_i \mathcal{A}_i$. We use the β-skewed Jensen-Shannon divergence:

$$\text{JSD}_\beta(p\|q) = \beta\, D_{\text{KL}}(p\|m) + (1-\beta)\, D_{\text{KL}}(q\|m)\ , \quad m = \beta\, p + (1-\beta)\, q \tag{7}$$

Preservation loss. Structural tokens (verdicts, formatting, transitions) coordinate across rubrics and have no meaningful isolation equivalent. We preserve them by aligning student and teacher logits under the same full context:

$$\mathcal{L}_{\text{preserve}} = \frac{1}{|\mathcal{S}|} \sum_{t \in \mathcal{S}} \text{JSD}_{\beta_p}(\mathbf{p}_t \parallel \mathbf{q}_t) \tag{8}$$

where $\mathbf{q}_t = \mathcal{M}_\phi(y_t \mid y_{<t}, c, R)$ is the teacher's full-context logit and $\mathcal{S}$ is the token set marked by $\mathbf{m}^{\text{str}}$.

Total loss.

$$\mathcal{L} = \mathcal{L}_{\text{distill}} + \alpha\, \mathcal{L}_{\text{preserve}} \tag{9}$$

Structural tokens are far fewer than analysis tokens, so each structural token already receives higher effective weight at equal α. We set $\alpha = 0.4$ across all experiments.

3.4 Rubric Context Augmentation

To expose the model to diverse interference patterns during training, we apply two augmentations to each rubric list R at every step:

Rubric shuffling. We randomly permute the rubric order, preventing the model from memorizing position-specific patterns and directly targeting order invariance (Eq. 1).

Noise injection. With probability $p_{\text{noise}} = 0.5$, we append rubrics sampled from unrelated conversations, training the model to ignore irrelevant criteria.

These augmentations target the interference sources identified in Table 1: shuffling addresses positional sensitivity; noise injection addresses content-based distraction.

4 EXPERIMENTS

4.1 SETUP

Datasets. We evaluate on three datasets spanning distinct domains and scoring formats. **HealthBench** (HB) contains medical conversations evaluated against per-sample rubric checklists with binary met/unmet verdicts; rubric counts range from 2 to 32. **FLASK** evaluates instruction-following across 3 fixed skill dimensions on a 1–5 scale. **ResearchQA** (RQA) evaluates scientific question answering against per-sample rubrics; the original dataset defines a 0–4 scale, which we shift to 1–5 for consistency with FLASK (1000 sampled instances). We split each dataset 8:1:1 for training, validation, and test.

Models and baselines. We apply SARA to Qwen3-8B, Qwen3-14B, Qwen3-32B, and Llama-3.1-8B-Instruct. We compare against $\mathbf{SFT_{iso}}$: supervised fine-tuning on isolation outputs concatenated into joint format, the most direct way to teach joint-mode behavior from isolation references.

Evaluation. All evaluations use greedy decoding. We average shuffle tests over 5 permutations. We report rubric-level agreement (Agr), weighted Cohen's κ, and sample-level exact match (EM). EM requires all rubrics in a sample to match simultaneously. Additional metrics (MAD, RMSE, Pearson ρ, aggregate total-score metrics) are defined in Appendix A and reported in Appendix E. Training details and infrastructure are in Appendix B.

4.2 MAIN RESULTS

Table 2 reports isolation–joint consistency across all settings. SARA improves every model on every dataset. The largest gains appear on ResearchQA: Qwen3-8B's EM nearly triples (.22→.59) and κ jumps from .665 to .868. ResearchQA combines moderate rubric counts with per-sample rubric definitions, creating ample room for cross-rubric contamination.

SARA benefits weaker models most (Llama-3.1-8B: +14% Agr on ResearchQA), consistent with the hypothesis that larger models learn partial invariance through scale. Yet even Qwen3-32B improves substantially on scored formats (ResearchQA κ: .651→.851; FLASK EM: .41→.59), confirming that scale alone does not resolve interference.

	HealthBench			**ResearchQA**			**FLASK**		
	Agr↑	κ↑	EM↑	Agr↑	κ↑	EM↑	Agr↑	κ↑	EM↑
Qwen3-8B	.819	.630	.22	.774	.665	.22	.720	.663	.41
+*SARA*	**.874**	**.736**	**.34**	**.914**	**.868**	**.59**	**.810**	**.801**	**.52**
Qwen3-14B	.874	.747	.34	.768	.658	.23	.722	.665	.41
+*SARA*	**.903**	**.806**	**.38**	**.900**	**.879**	**.52**	**.803**	**.802**	**.55**
Qwen3-32B	.887	.775	.36	.720	.651	.25	.701	.652	.41
+*SARA*	.881	.762	.36	**.880**	**.851**	**.42**	**.836**	**.822**	**.59**
Llama-3.1-8B	.733	.467	.08	.711	.608	.12	.609	.637	.24
+*SARA*	**.830**	**.652**	**.18**	**.853**	**.800**	**.34**	**.722**	**.760**	**.41**

Table 2: Isolation–joint consistency across three datasets and four models. Agr = rubric-level agreement, κ = weighted Cohen's kappa, EM = sample-level exact match. Bold marks improvement over the base model.

The cross-dataset variation reflects rubric design. HealthBench rubrics are binary and test specific feature presence (e.g., "mentions drug interaction"), leaving little room for interference to shift

verdicts. ResearchQA rubrics describe broader quality dimensions on a 5-point scale, giving the model more room to rationalize different scores under different contexts. FLASK's 3 fixed but abstract trait definitions (e.g., "logical robustness") sit in between. EM gains consistently outpace Agr gains across all settings (e.g., Qwen3-8B on ResearchQA: Agr +.14, EM +.37), revealing that baseline interference scatters across many rubrics rather than concentrating on a few hard ones.

4.3 Consistency Without Quality Degradation

Consistency gains are only useful if they preserve judgment quality. We verify this from two angles.

SARA vs. prompting and naive SFT. A natural first attempt is to instruct the model to evaluate each rubric independently. We prepend an explicit independence instruction to the system prompt ("Evaluate each rubric COMPLETELY INDEPENDENTLY. Do NOT let your analysis or verdict on one rubric influence any other rubric."). Table 3 shows this provides no benefit on Qwen3-14B (ResearchQA): EM drops from .23 to .20 and Agr decreases from .768 to .755, confirming that interference is a behavioral pattern that prompting cannot override. We restrict all baselines to single-inference methods to match SARA's deployment cost in online RL reward modeling.[1]

A stronger baseline, SFT_{iso}, fine-tunes on isolation outputs concatenated into joint format (5 random-seed samples per rubric as synthetic targets). Beyond consistency, we track behavioral drift through rubric-level agreement with the untrained base model in each mode. SFT_{iso} achieves marginally higher consistency than SARA (Agr .909 vs .900, EM .54 vs .52) but drifts further from the base model in both modes (Iso: .892 vs .925; Joint: .777 vs .806). SFT forces the model to reproduce fixed targets, shifting its judgment distribution. SARA teaches the model to reason independently per rubric, preserving original behavior while reducing interference.

	Iso–Joint Consistency			**Iso Agr w/**	**Joint Agr w/**
	Agr↑	κ↑	EM↑	Base↑	Base↑
Qwen3-14B	.768	.658	.23	–	–
+Indep. prompt	.755	.672	.20	.936	.876
+SFT_{iso}	**.909**	.872	**.54**	.892	.777
+SARA	.900	**.879**	.52	.925	.806

Table 3: Baseline comparison on ResearchQA (Qwen3-14B). Independence prompting shifts model behavior (Agr w/ Base <1) but does not reduce interference. SFT achieves slightly higher consistency but drifts further from the base model. SARA balances consistency and behavioral preservation.

	Isolation				**Joint**			
	Agr w/ Base		Agr w/ GPT-4.1		Agr w/ Base		Agr w/ GPT-4.1	
	Base	+SARA	Base	+SARA	Base	+SARA	Base	+SARA
Qwen3-8B	–	.921	.729	.706	–	.874	.773	.731
Qwen3-14B	–	.912	.801	.806	–	.896	.795	.806
Qwen3-32B	–	.896	.830	.823	–	.894	.815	.819
Llama-3.1-8B	–	.839	.702	.700	–	.755	.726	.755

Table 4: Evaluation quality on HealthBench (rubric-level agreement). Both base and SARA variants report GPT-4.1 agreement to contextualize absolute levels.

Agreement with external judges. Table 4 reports evaluation quality on HealthBench. We measure rubric-level agreement with the untrained base model and with GPT-4.1 as an external reference, reporting both the base model and its SARA-aligned variant in both evaluation modes.

In isolation mode, SARA maintains agreement with the untrained base above .84 across all models. Joint-mode base agreement is inherently lower, particularly for weaker models (Llama-3.1-

[1]Multi-inference strategies (e.g., majority voting) can improve reliability but multiply cost per evaluation, making them impractical for reward models called at every RL policy step.

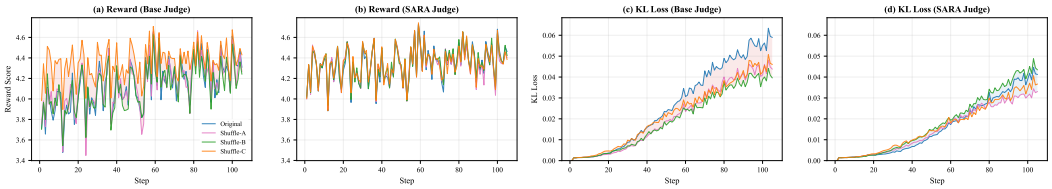


Figure 3: RL training trajectories under rubric-order perturbation (4 runs per condition, shaded region = min–max envelope). SARA-judge runs cluster tightly in both reward **(a–b)** and KL loss **(c–d)**.

	Base	**SARA**	**Ratio**
Training trajectory dispersion ($\times 10^{-3}$)			
Per-step σ (reward)	107.1	21.3	5.0×
Per-step σ (KL loss)	2.9	1.5	1.9×
Area between extremes	278.4	54.7	5.1×
Downstream eval (Qwen3.8-27B, step 60, 100 samples)			
Mean score	3.694	3.687	—
Cross-variant range ($\times 10^{-3}$)	63.1	13.1	4.8×

Table 5: RL stability under rubric-order perturbation (4 shuffle variants per condition). SARA reduces reward variance by 5× while maintaining equivalent downstream scores.

8B: .755). GPT-4.1 agreement holds steady or improves for most settings (Qwen3-14B joint: .795→.806; Llama-3.1-8B joint: .726→.755). Comparing Base and +SARA columns confirms that absolute agreement levels reflect each base model's own accuracy rather than degradation from SARA.

4.4 Downstream Impact: Reward Model Stability

Rubric-based judges increasingly serve as reward models for RL training, where inconsistent reward signals inject noise into policy optimization and reduce reproducibility. We test whether SARA's consistency gains produce more stable RL training.

Setup. We train Qwen3-4B with GRPO (Shao et al., 2024) using either a base Qwen3-14B judge or its SARA-aligned variant as the reward model (ResearchQA, 800 training samples, 2 epochs). For each judge we run four training variants that differ only in rubric presentation order, isolating rubric-order sensitivity in the reward signal.

Results. Figure 3 overlays four runs per condition. Base-judge runs diverge visibly in both reward and KL trajectories; SARA-judge runs remain tightly clustered. Table 5 quantifies the gap: SARA reduces per-step reward σ by 5.0× and the area between extreme trajectories by 5.1×.

We evaluate step-60 checkpoints on 100 held-out samples using an independent Qwen3.8-27B judge. Both conditions achieve comparable mean scores (3.694 vs. 3.687), but base-judge training produces 4.8× wider cross-variant range. The base judge's higher variance means individual run outcomes depend on rubric ordering rather than policy quality. SARA's tight clustering ensures that measured gains reflect genuine improvement, and equivalent mean scores rule out the concern that SARA suppresses useful information.

4.5 Transfer Across Datasets

To confirm that SARA teaches a general capability rather than fitting dataset-specific patterns, we evaluate Qwen3-14B trained on ResearchQA (SARA_{rqa}) directly on HealthBench and FLASK without further fine-tuning.

Table 6 shows that SARA_{rqa} improves over the base model on both target datasets. On FLASK it even outperforms the in-domain SARA model in Agr (.820 vs .803). This confirms that SARA

	HealthBench			FLASK		
	Agr↑	κ↑	EM↑	Agr↑	κ↑	EM↑
Qwen3-14B	.874	.747	.34	.722	.665	.41
+SARA (in-domain)	**.903**	**.806**	**.38**	.803	**.802**	**.55**
+SARA$_{\text{rqa}}$ (transfer)	.890	.779	.34	**.820**	.800	**.55**

Table 6: Cross-dataset transfer. SARA$_{\text{rqa}}$ trains on ResearchQA and evaluates zero-shot on HealthBench and FLASK. It improves over the base model on both targets and exceeds the in-domain SARA model on FLASK Agr (.820 vs .803).

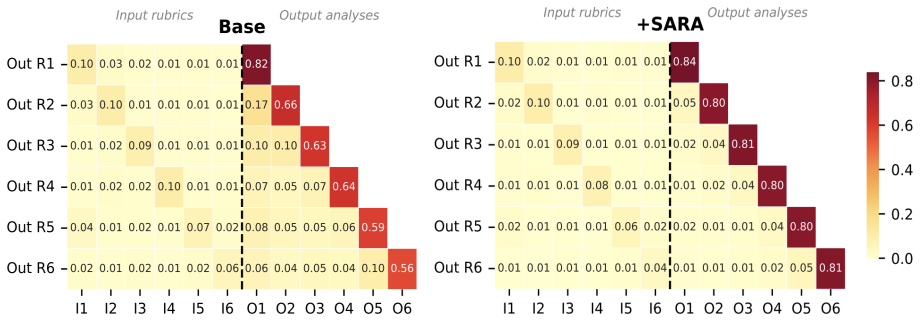


Figure 4: Normalized attention from output analysis segments to input rubric spans (I1–I6) and output analysis spans (O1–O6), averaged over 10 ResearchQA samples. Each row sums to one.

learns a domain-agnostic ability to decouple per-rubric reasoning from contextual interference rather than memorizing dataset-specific rubric patterns.

4.6 Analysis

Attention patterns. Figure 4 compares attention during joint evaluation (Qwen3-14B, ResearchQA, 10 samples). We extract attention probabilities from the final transformer layer. Since conversation context absorbs over 90% of raw attention mass, we normalize over input rubric spans and output analysis spans only to isolate cross-rubric interactions. Each row in the heatmap sums to one under this normalization.

Two effects stand out. SARA concentrates self-attention on the diagonal: average self-attention rises from .623 to .809 (+30%), meaning each rubric's analysis attends primarily to its own reasoning. SARA also suppresses cross-rubric leakage: off-diagonal attention in the output half drops from .039 to .014 (–64%). The base model's lower triangle shows visible leakage where later rubrics attend heavily to earlier analyses (Out R2 allocates .168 to Out R1); SARA reduces this to .048. Attention to other input rubric definitions also decreases (.018→.011, –38%). These patterns confirm that SARA achieves interference reduction through a concrete mechanistic change: redirecting attention from cross-rubric spans to the target rubric's own analysis.

Case study. Appendix H presents a FLASK example (Qwen3-14B) where interference distorts verdicts bidirectionally. In isolation, both models assign R2 (factuality) 3/5 with balanced language acknowledging partial accuracy. In joint mode, the base model adopts absolutist phrasing ("does not provide *any* factual evidence"), deflating R2 to 2/5, while simultaneously inflating R1 (comprehension) from 3 to 4 with softer language. SARA's joint output matches isolation on all three rubrics, preserving both the critical tone and the balanced assessment.

Controlled perturbations. We apply the perturbation framework from Table 1 to probe specific interference sources. As rubric count K grows from 2 to 8 on ResearchQA, base model MAD nearly doubles; SARA holds MAD nearly constant (Qwen3-8B: .080 at all K). Under rubric shuffling

and noise injection, SARA roughly doubles EM across all models. Full results and figures are in Appendix D.

Additional observations. Qwen3-32B plateaus on HealthBench (Agr .887→.881, κ .775→.762), consistent with the limited headroom on binary checklists discussed in Section 4.2. SARA-trained models produce longer outputs with more detailed per-rubric analyses (e.g., Qwen3-8B on HealthBench: 405→611 tokens); however, SFT_{iso} produces comparable lengths yet drifts further (Table 3), ruling out length as the primary driver. Hyperparameter sensitivity, preservation loss ablation, and token statistics are in Appendices F and G.

5 Related Work

LLM-as-a-Judge and rubric-based evaluation. LLM judges have become standard evaluation infrastructure (Zheng et al., 2023; Li et al., 2024). Chain-of-thought prompting improves correlation with human ratings (Liu et al., 2023), and task-specific checklists from real users have moved evaluation toward structured per-sample criteria (Lin et al., 2025). Rubric-based evaluation further decomposes quality into fine-grained, verifiable items with improved reproducibility (Lee et al., 2025; Kim et al., 2024; Pathak et al., 2025). We identify rubric interference as a reliability concern specific to joint multi-rubric assessment.

Biases, reliability, and prompt sensitivity. LLM judges exhibit systematic biases including position bias, verbosity bias, and self-enhancement bias (Wang et al., 2024; Wu and Aji, 2025; Panickssery et al., 2024). Sensitivity to prompt complexity and general leniency further compromise reliability (Thakur et al., 2025), and even GPT-4o performs near chance on challenging pairs (Tan et al., 2025). Calibration strategies partially address these issues (Li et al., 2025a; 2026). More broadly, LLMs show strong prompt sensitivity: formatting changes alone cause up to 76-point accuracy swings (Sclar et al., 2024), motivating multi-prompt evaluation (Mizrahi et al., 2024). Paraphrase-based consistency frameworks measure whether equivalent queries elicit the same factual answer (Elazar et al., 2021). All these findings concern how a judge compares different *responses* or how an LLM responds to different *prompts*. Rubric interference is orthogonal: the judgment on a fixed response shifts depending on the co-evaluated rubric set.

On-policy and self-distillation. On-policy distillation trains a student on self-sampled trajectories with per-token teacher guidance, mitigating distribution shift (Agarwal et al., 2024; Gu et al., 2024). Self-distillation removes the separate teacher entirely and can improve generalization (Furlanello et al., 2018). In the LLM setting, models have been trained to distinguish their own outputs from references (Chen et al., 2024), to serve as their own reward function (Yuan et al., 2024), and to teach themselves by conditioning on privileged information (Zhao et al., 2026). SARA shares this on-policy self-distillation principle but targets a different goal: aligning joint-mode with isolation-mode evaluation. The privileged information is not a stronger model but the absence of interfering rubrics.

6 Conclusion

We identify rubric interference as a systematic failure mode in multi-rubric LLM evaluation. Our method, SARA, aligns joint-mode reasoning with the model's own isolation judgments through self-distillation. It improves consistency across three datasets, two model families, and both binary and scored formats. Gains are largest on scored rubrics that describe broad quality dimensions (e.g., ResearchQA, FLASK), where definitions leave room for subjective interpretation. On binary rubrics that test specific feature presence (e.g., HealthBench), gains are moderate but consistent. Consistency also transfers across domains without retraining. Attention analysis confirms the mechanism: SARA suppresses cross-rubric information flow while preserving within-rubric coherence. Downstream, a SARA-aligned reward model cuts inter-run reward variance by 5× under rubric-order perturbation, with policy quality unchanged. The measurement framework itself serves as a diagnostic protocol for multi-rubric judges before deployment. Extending SARA to extended-thinking models, where reasoning lacks structured segment boundaries, is a natural next step.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2024. URL https://arxiv.org/abs/2306.13649.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. Healthbench: Evaluating large language models towards improved human health, 2025. URL https://arxiv.org/abs/2505.08775.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models, 2024. URL https://arxiv.org/abs/2401.01335.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021. doi: 10.1162/tacl_a_00410. URL https://aclanthology.org/2021.tacl-1.60/.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1607–1616. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/furlanello18a.html.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 32694–32717, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8ac015d409635f196f9e3e9dcfb9a94e-Paper-Conference.pdf.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean M. Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=c1bTcrDmt4.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.248. URL https://aclanthology.org/2024.emnlp-main.248/.

Yukyung Lee, Joonghoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. Checkeval: A reliable llm-as-a-judge framework for evaluating text generation using checklists, 2025. URL https://arxiv.org/abs/2403.18771.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. Llms-as-judges: A comprehensive survey on llm-based evaluation methods, 2024. URL https://arxiv.org/abs/2412.05579.

Haitao Li, Junjie Chen, Qingyao Ai, Zhumin Chu, Yujia Zhou, Qian Dong, and Yiqun Liu. CalibraEval: Calibrating prediction distribution to mitigate selection bias in LLMs-as-judges. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16537–16552, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.808. URL https://aclanthology.org/2025.acl-long.808/.

Qingquan Li, Shaoyu Dou, Kailai Shao, Chao Chen, and Haixiang Hu. Evaluating scoring bias in llm-as-a-judge, 2026. URL https://arxiv.org/abs/2506.22316.

S. Yifei Li, Allen Chang, Chaitanya Malaviya, and Mark Yatskar. Researchqa: Evaluating scholarly question answering at scale across 75 fields with survey-mined questions and rubrics, 2025b. URL https://arxiv.org/abs/2509.00496.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking LLMs with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=MKEHCx25xp.

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 Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024. doi: 10.1162/tacl_a_00681. URL https://aclanthology.org/2024.tacl-1.52/.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations, 2024. URL https://arxiv.org/abs/2404.13076.

Aditya Pathak, Rachit Gandhi, Vaibhav Uttam, Arnav Ramamoorthy, Pratyush Ghosh, Aaryan Raj Jindal, Shreyash Verma, Aditya Mittal, Aashna Ased, Chirag Khatri, Yashwanth Nakka, Devansh, Jagat Sesh Challa, and Dhruv Kumar. Rubric is all you need: Improving llm-based code evaluation with question-specific rubrics. In *Proceedings of the 2025 ACM Conference on International Computing Education Research V.1*, ICER '25, page 181–195. ACM, August 2025. doi: 10.1145/3702652.3744220. URL http://dx.doi.org/10.1145/3702652.3744220.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models'sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 25055–25083, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/6c0e99d736da621403018ca7b32b1a4d-Paper-Conference.pdf.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Yuan Tang, Alejandro Cuadron, Chenguang Wang, Raluca Popa, and Ion Stoica. Judgebench: A benchmark for evaluating LLM-based judges. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=G0dksFayVq.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges. In Ofir Arviv, Miruna Clinciu, Kaustubh Dhole, Rotem Dror, Sebastian Gehrmann, Eliya Habba, Itay Itzhak, Simon Mille, Yotam Perlitz, Enrico Santus, João Sedoc, Michal Shmueli Scheuer, Gabriel Stanovsky, and Oyvind Tafjord, editors, *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 404–430, Vienna, Austria and virtual meeting, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-261-9. URL https://aclanthology.org/2025.gem-1.33/.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting*

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9440–9450, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.511. URL https://aclanthology.org/2024.acl-long.511/.

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

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57905–57923. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yuan24d.html.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models, 2026. URL https://arxiv.org/abs/2601.18734.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

A Detailed Metric Definitions

All experiments compare verdicts on the same conversation–rubric pair (c, r_i) across two evaluation conditions A and B (e.g., isolation vs. joint, or two different rubric permutations). We use the same core metrics for both binary and scored formats.

A.1 Rubric-Level Metrics

Given N verdict pairs $\{(v_i^{(A)}, v_i^{(B)})\}_{i=1}^{N}$:

Agreement (Agr). Fraction of rubrics with identical verdicts:

$$\text{Agr} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[v_i^{(A)} = v_i^{(B)}] \tag{10}$$

For binary verdicts, this is equivalent to classification accuracy between two annotators. For scored verdicts, it measures exact score match (identical to what some prior work calls Exact Agreement Rate).

Cohen's κ. Agreement corrected for chance:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{11}$$

For unweighted κ, $p_o = \text{Agr}$. For quadratic-weighted κ, p_o and p_e are computed using quadratic disagreement weights $w_{ij} = 1 - (i-j)^2/(k-1)^2$ where k is the number of scale levels.

Mean Absolute Deviation (MAD). Average per-rubric deviation:

$$\text{MAD} = \frac{1}{N}\sum_{i=1}^{N}|v_i^{(A)} - v_i^{(B)}| \tag{12}$$

For binary verdicts, MAD $= 1 -$ Agr. For scored verdicts, MAD captures the typical magnitude of disagreement.

Root Mean Squared Error (RMSE).

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(v_i^{(A)} - v_i^{(B)})^2} \tag{13}$$

More sensitive to large deviations than MAD. Reported in appendix tables for scored formats.

Pearson Correlation (ρ). Linear correlation between $v^{(A)}$ and $v^{(B)}$.

A.2 Sample-Level Metric

Exact Match (EM). Fraction of samples where *all* rubrics agree:

$$\text{EM} = \frac{1}{|C|}\sum_{c\in C}\mathbb{1}\left[\forall\, r_i \in R_c:\ v_i^{(A)} = v_i^{(B)}\right] \tag{14}$$

EM is the strictest metric: a single rubric disagreement within a sample counts as a failure.

A.3 Aggregate-Level Metrics

For scored formats, we also compute the total score per sample $T_c^{(X)} = \sum_{r_i\in R_c} v_i^{(X)}$ and report:

Total Exact Match (TotalEM). Fraction of samples with identical total scores:

$$\text{TotalEM} = \frac{1}{|C|}\sum_{c\in C}\mathbb{1}[T_c^{(A)} = T_c^{(B)}] \tag{15}$$

Total MAE. Mean absolute error of total scores:

$$\text{TotalMAE} = \frac{1}{|C|}\sum_{c\in C}|T_c^{(A)} - T_c^{(B)}| \tag{16}$$

Total Pearson (ρ_T). Pearson correlation of total score sequences. Measures whether sample-level ranking is preserved across conditions.

A.4 Metric Selection

B Training and Infrastructure Details

B.1 SARA Training

Optimization. We use AdamW with learning rate 1×10^{-6}, linear warmup over the first 10% of steps, and cosine decay to 1×10^{-7}. Batch size is 4 per GPU (effective batch size 32 on 8×H20). We train for 3 epochs on each dataset and select the checkpoint with the highest shuffle consistency (EM, 5 permutations) on the validation set.

On-policy generation. At each step we generate joint evaluations using vLLM with greedy decoding (temperature 0). We regenerate on-policy outputs every epoch to track the model's evolving behavior.

Metric	HB	FLASK	RQA
Main tables (all formats)			
Agr, κ, EM	✓	✓	✓
Appendix (scored only)			
MAD, RMSE, ρ		✓	✓
TotalEM, TotalMAE, ρ_T		✓	✓
Operations			
Overall	✓	✓	✓
Con-K	✓		✓
Shuffle	✓	✓	✓
Noise	✓	✓	✓

Table 7: Metric and operation applicability. FLASK's fixed 3-rubric format does not support Con-K or Subsetting.

Segment parsing. We parse per-rubric segments using a regex that matches the pattern `**Rubric \d+ Analysis**: ... **Verdict**: ...`. On a manual audit of 200 samples across all three datasets, parsing succeeds on 99.2% of rubric segments. The remaining 0.8% (malformed outputs) are excluded from loss computation for that step.

Noise injection. With probability $p_{\text{noise}} = 0.5$, we append 2 rubrics sampled from other conversations in the same batch. Noise rubrics are appended after the original rubric list and excluded from loss computation.

Hyperparameters. Table 8 lists all SARA hyperparameters.

Hyperparameter	Value
Learning rate	1×10^{-6}
LR schedule	cosine decay
Warmup ratio	10%
Batch size (per GPU)	4
GPUs	8× H20 (96 GB)
Epochs	3
EMA decay β_{ema}	0.999
JSD weight β_d (distill)	0.5
JSD weight β_p (preserve)	1.0
Preservation coefficient α	0.4
Noise injection probability	0.5
Noise rubric number	2
Max rubrics per sample	10
Decoding	greedy

Table 8: SARA hyperparameters (shared across all datasets and models unless noted).

B.2 SFT_{ISO} Baseline

We generate 5 isolation outputs per rubric using different random seeds (temperature 0.7) and concatenate them into joint format. Training uses the same optimizer and schedule as SARA. We select the checkpoint with the lowest validation loss.

B.3 RL Experiment (Section 4.4)

Policy model. Qwen3-4B, trained with GRPO (Shao et al., 2024) using the verl framework.

Reward model. Either base Qwen3-14B or SARA-aligned Qwen3-14B, served via vLLM. The reward model evaluates each policy response against the full rubric list in joint mode and returns the normalized total score as the reward signal.

Training. 800 training samples from ResearchQA (the training split), 2 epochs, batch size 16, learning rate 1×10^{-6}, KL coefficient 0.001. For each judge variant, we run 4 training variants that differ only in rubric presentation order (one original + three random shuffles). All other hyperparameters and random seeds are identical across variants.

Evaluation. We evaluate step-60 checkpoints (selected before significant KL divergence in any run) on 100 held-out ResearchQA samples. An independent Qwen3.8-27B judge scores each response using the same rubric set in joint mode. We report mean score and cross-variant range across the 4 checkpoints.

Cost estimate. SARA training on ResearchQA (1000 samples, Qwen3-14B) takes approximately 10 hours on 8×H20 GPUs. Each RL training run takes approximately 2 hours. The total compute for the RL stability experiment (8 runs) is approximately 16 H20-hours.

B.4 Infrastructure

All experiments run on 8× NVIDIA H20 GPUs (96 GB each). We use vLLM (v0.11.0) for batched inference during on-policy generation and evaluation. Training uses DeepSpeed ZeRO Stage 2 for memory efficiency on 32B models.

C Evaluation Prompts

We use the following task prompts for joint-mode evaluation. In isolation mode, the same prompt is used with a single-rubric list. The FLASK prompt follows the same format as ResearchQA with the scoring scale adapted to 1–5.

C.1 HealthBench (Binary)

Joint-Mode Evaluation Prompt (Binary)

```
# Task
You are a professional dialogue grader.  Your task is to evaluate
the **last response (completion)** from the Assistant in the provided
conversation against a list of scoring rubrics.  You must determine
which criteria are met, and output the indices of the met rubrics.
--------------------
# Input Data
**Conversation**:
{conversation}
--------------------
**Rubrics**:
{rubrics}
(Each rubric contains a "criterion" and a "point_value".  Points can
be positive or negative.)
--------------------
# Scoring Logic
1.  **Evaluation**:  For each item in the Rubrics, determine if the
Assistant's last response meets the specific criterion.
2.  **"Such as" Exception**:  If a criterion includes phrases like
"such as", "for example", or "including", the response meets the
criteria if it addresses the core requirement, even if it does not
mention every specific example listed.
3.  **Negative Criteria**:  If a criterion describes an undesirable
behavior (e.g., "is overly verbose" or "contains inaccuracies"), the
criteria is "met" only if the Assistant actually exhibited that bad
behavior.
# Output Format
For each rubric, output strictly in this format (one entry per rubric,
do NOT split a single rubric into multiple entries):
```

```
**Rubric 1 Analysis**: <analysis>. **Verdict**: Met/Not Met.
**Rubric 2 Analysis**: <analysis>. **Verdict**: Met/Not Met.
...
Met Rubric Indices: [list of index numbers, separated by commas]
```

C.2 RESEARCHQA / FLASK (SCORED)

Joint-Mode Evaluation Prompt (Scored)

```
# Task
You are a professional dialogue grader. Your task is to evaluate
the **last response (completion)** from the Assistant in the provided
conversation against a list of scoring rubrics. Each rubric describes
a specific requirement. You must judge the degree to which the
response fulfills each requirement on a 1-5 scale.
--------------------
# Input Data
**Conversation**:
{conversation}
--------------------
**Rubrics**:
{rubrics}
(Each rubric describes a specific requirement that the response should
fulfill.)
--------------------
# Scoring Logic
1. **Evaluation**: For each rubric, determine how well the
Assistant's last response fulfills the described requirement.
2. **Scoring Scale**:
- **5**: Fully fulfilled --- the response completely and accurately
addresses the requirement.
- **4**: Mostly fulfilled --- the response addresses the requirement
with minor omissions or imprecisions.
- **3**: Partially fulfilled --- the response addresses some aspects
but misses key elements.
- **2**: Minimally fulfilled --- the response only tangentially or
superficially relates to the requirement.
- **1**: Not fulfilled --- the response fails to address the
requirement or directly contradicts it.
3. **Holistic Judgment**: If the response falls between two levels,
choose the lower one.
# Output Format
For each rubric, output strictly in this format (one entry per rubric,
do NOT split a single rubric into multiple entries):
**Rubric 1 Analysis**: <analysis>. **Verdict**: <score>/5.
**Rubric 2 Analysis**: <analysis>. **Verdict**: <score>/5.
...
Total Score: <s1>+<s2>+...=<sum>
```

The FLASK prompt uses the identical format with the scoring scale adjusted to match its 1–5 skill-based rubrics. For isolation-mode evaluation, the same prompts are used with $|R| = 1$ (a single rubric in the rubric list). The output format reduces to a single analysis–verdict pair.

D CONTROLLED PERTURBATION RESULTS

This appendix reports the full results for the controlled perturbation experiments summarized in Section 4.6.

D.1 Consistency-at-K Full Results

Tables 9 and 10 report consistency as co-evaluated rubric count K grows. FLASK is omitted because its fixed 3-rubric format does not support variable-K evaluation.

	$K=2$			$K=4$			$K=8$		
	Agr↑	κ↑	EM↑	Agr↑	κ↑	EM↑	Agr↑	κ↑	EM↑
HealthBench									
Qwen3-8B	.871	.722	.754	.844	.672	.515	.843	.675	.266
+*SARA*	.914	.816	.832	.896	.779	.654	.885	.756	.403
Qwen3-14B	.901	.802	.806	.890	.779	.635	.874	.746	.355
+*SARA*	.953	.905	.912	.932	.864	.746	.912	.864	.468
Qwen3-32B	.895	.789	.796	.882	.763	.600	.884	.768	.350
+*SARA*	.913	.824	.836	.900	.798	.654	.898	.795	.415
Llama-3.1-8B	.797	.594	.627	.749	.498	.309	.733	.469	.094
+*SARA*	.861	.718	.741	.821	.636	.467	.824	.641	.240

Table 9: Consistency-at-K on HealthBench. Metrics: Agr = rubric-level agreement, κ = Cohen's kappa, EM = sample-level exact match.

	$K=2$			$K=4$			$K=8$		
	Agr↑	MAD↓	EM↑	Agr↑	MAD↓	EM↑	Agr↑	MAD↓	EM↑
ResearchQA									
Qwen3-8B	.867	.147	.761	.808	.229	.487	.787	.260	.216
+*SARA*	.928	.080	.866	.927	.080	.751	.922	.081	.568
Qwen3-14B	.867	.167	.751	.803	.278	.460	.787	.324	.243
+*SARA*	.926	.082	.862	.913	.096	.705	.909	.101	.460
Qwen3-32B	.837	.203	.718	.756	.335	.405	.760	.362	.216
+*SARA*	.892	.124	.797	.876	.144	.592	.910	.108	.378
Llama-3.1-8B	.788	.296	.628	.739	.374	.324	.740	.362	.027
+*SARA*	.889	.117	.809	.862	.171	.563	.872	.176	.324

Table 10: Consistency-at-K on ResearchQA. Agr here is exact score agreement. MAD captures deviation magnitude.

D.2 Shuffle Invariance and Noise Robustness

Figure 5 summarizes EM under shuffle and noise perturbations. SARA roughly doubles EM across all models and datasets. Full numerical results are in Tables 11 and 12.

E Supplementary Metrics for Main Experiment

Table 13 reports additional rubric-level metrics (MAD, RMSE, Pearson ρ) for the scored datasets in the main experiment (overall consistency: isolation vs. joint mode). Table 14 reports aggregate-level metrics computed from per-sample total scores.

F Hyperparameter Sensitivity

Hyperparameter sensitivity. Table 15 ablates the two key hyperparameters on Qwen3-14B (ResearchQA). Setting decay=1 freezes the teacher at its initialization (since $(1-\beta_{\text{ema}})=0$ zeroes out the student contribution), which drops EM from .52 to .48, confirming that the teacher must be decoupled from the student to provide a stable distillation target. Performance is robust across

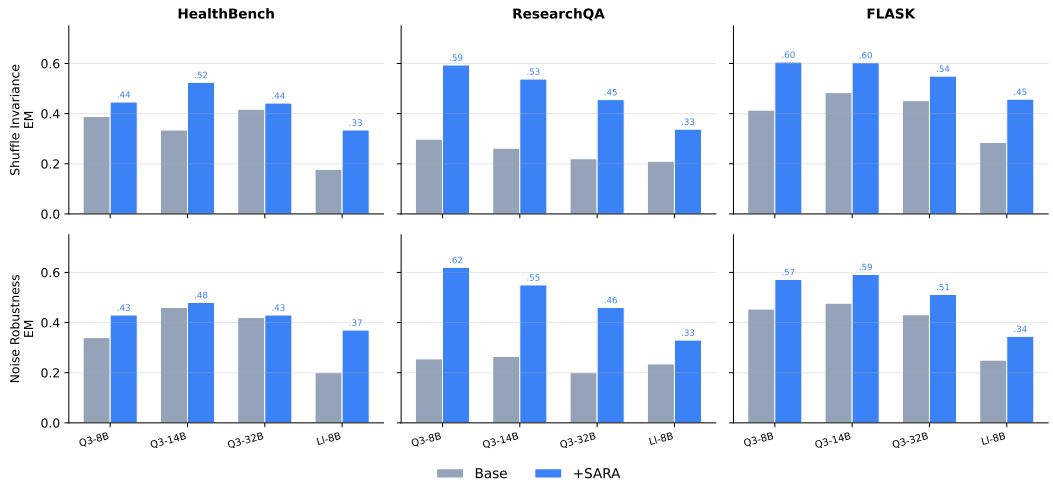


Figure 5: Sample-level exact match (EM) under shuffle invariance (top row) and noise robustness at 40% noise ratio (bottom row). Gray bars denote base models; blue bars denote SARA-trained models.

	HealthBench			**ResearchQA**			**FLASK**		
	Agr↑	κ↑	EM↑	Agr↑	κ↑	EM↑	Agr↑	κ↑	EM↑
Qwen3-8B	.886	.770	.388	.783	.695	.298	.709	.667	.414
+*SARA*	.911	.819	.446	.921	.879	.594	.839	.832	.606
Qwen3-14B	.888	.773	.335	.791	.746	.262	.732	.657	.484
+*SARA*	.934	.867	.525	.912	.886	.538	.831	.816	.603
Qwen3-32B	.900	.800	.417	.740	.733	.220	.703	.658	.452
+*SARA*	.908	.816	.442	.888	.868	.456	.862	.868	.651
Llama-3.1-8B	.783	.566	.178	.741	.626	.210	.604	.642	.285
+*SARA*	.887	.769	.335	.852	.780	.338	.772	.799	.458

Table 11: Shuffle invariance (5 permutations, averaged).

decay $\in [.99, .999]$. The JSD interpolation weight β_d shows similar robustness: EM varies by only .02 across $\beta_d \in [0.1, 0.7]$. However, β_d=0.9 causes EM to collapse to .38—when the loss is dominated by the teacher distribution, the student loses the capacity to produce coherent independent judgments.

We note that decay=0.99 achieves the highest EM (.59) in this sweep. However, faster teacher updates risk instability on longer training runs and larger models. We select decay=0.999 as the default for robustness across all datasets and model scales.

The preservation loss is necessary for structural integrity. Setting $\alpha = 0$ (no preservation loss) causes the model to lose the ability to produce structural outputs within tens of training steps; even $\alpha = 0.1$ makes a difference (Table 16). Unlike the EMA and β_d parameters (Table 15), the preservation coefficient is not a tunable dial but a binary structural requirement.

G Output Token Statistics

Table 17 reports the average number of generated tokens per sample in joint mode. SARA-trained models produce longer outputs due to more detailed per-rubric analyses. The increase is proportional to the average rubric count: HealthBench (avg 11.5 rubrics) shows the largest absolute increase, while FLASK (3 rubrics) shows the smallest.

	HealthBench			ResearchQA			FLASK		
	Agr↑	κ↑	EM↑	Agr↑	κ↑	EM↑	Agr↑	κ↑	EM↑
Qwen3-8B	.876	.751	.340	.770	.663	.255	.740	.701	.454
+*SARA*	.908	.813	.430	.923	.877	.620	.826	.819	.572
Qwen3-14B	.896	.792	.460	.773	.712	.265	.737	.646	.477
+*SARA*	.924	.847	.480	.921	.880	.550	.819	.815	.592
Qwen3-32B	.892	.783	.420	.726	.714	.200	.694	.644	.431
+*SARA*	.905	.810	.430	.885	.861	.460	.847	.861	.615
Llama-3.1-8B	.755	.509	.200	.755	.621	.235	.589	.626	.250
+*SARA*	.873	.743	.370	.842	.772	.330	.713	.743	.345

Table 12: Noise robustness (40% noise ratio).

	ResearchQA			FLASK		
	MAD↓	RMSE↓	ρ↑	MAD↓	RMSE↓	ρ↑
Qwen3-8B	.273	.608	.816	.473	1.039	.728
+*SARA*	**.101**	**.361**	**.918**	**.289**	**.762**	**.862**
Qwen3-14B	.372	.844	.794	.441	.983	.743
+*SARA*	**.113**	**.385**	**.942**	**.247**	**.605**	**.889**
Qwen3-32B	.408	.832	.794	.471	.986	.737
+*SARA*	**.148**	**.472**	**.911**	**.243**	**.662**	**.822**
Llama-3.1-8B	.437	.919	.748	.542	.968	.763
+*SARA*	**.186**	**.535**	**.886**	**.364**	**.766**	**.854**

Table 13: Supplementary rubric-level metrics for overall consistency (scored datasets). MAD = mean absolute deviation, RMSE = root mean squared error, ρ = Pearson correlation.

H Case Study: Full Outputs

Table 18 shows the complete joint-mode outputs. The task asks to fill in the blank: "_____ is the direct attempt to formally or informally manage ethical issues or problems, through specific policies, practices and programmes." The assistant responds: "Let's think step by step. We refer to Wikipedia articles on business ethics for help. The direct attempt to manage ethical issues through specific policies, practices, and programs is business ethics management."

I Declaration of AI Assistant Usage

We explicitly disclose the nature and scope of AI assistance in this work:

Writing and Refinement: We used Claude (Anthropic) strictly as a writing assistant to verify grammar, polish writing style, and refine the text of the manuscript.

Coding Assistance: We employed Claude to draft routine boilerplate data processing scripts and infrastructure execution modules. However, the human authors entirely conceptualized and developed all core SARA algorithms, distillation losses, and custom pipeline code.

Human Ownership: The human authors maintain sole ownership of all core scientific ideas, mathematical formulations, experimental designs, and data interpretations. We manually created all primary data visualizations, plots, and tables without any AI generation. The use of GPT-4.1, as detailed in Section 4, serves strictly as an automated reference baseline/judge within our evaluation framework, rather than an interactive assistant in our research or writing process.

	ResearchQA			FLASK		
	TotalEM↑	TotalMAE↓	ρ_T↑	TotalEM↑	TotalMAE↓	ρ_T↑
Qwen3-8B	.250	1.620	.941	.431	1.316	.767
+*SARA*	**.640**	**0.470**	**.989**	**.557**	**0.695**	**.920**
Qwen3-14B	.250	2.100	.931	.437	1.206	.791
+*SARA*	**.610**	**0.550**	**.988**	**.609**	**0.592**	**.913**
Qwen3-32B	.310	2.340	.906	.425	1.264	.788
+*SARA*	**.530**	**0.720**	**.983**	**.615**	**0.626**	**.910**
Llama-3.1-8B	.210	2.100	.900	.305	1.316	.854
+*SARA*	**.390**	**0.950**	**.973**	**.386**	**0.454**	**.915**

Table 14: Aggregate-level metrics for overall consistency (scored datasets). TotalEM = fraction of samples with identical total scores, TotalMAE = mean absolute error of total scores, ρ_T = Pearson correlation of total scores.

Configuration	Agr↑	EM↑
EMA decay (β_d=0.5)		
decay = 1.0 (no EMA)	.890	.48
decay = 0.995	.911	.51
decay = 0.999 (default)	.900	.52
decay = 0.99	.918	.59
JSD interpolation β_d (decay=.999)		
β_d = 0.1	.905	.53
β_d = 0.3	.900	.54
β_d = 0.5 (default)	.900	.52
β_d = 0.7	.900	.52
β_d = 0.9	.867	.38

Table 15: Hyperparameter ablation (Qwen3-14B, ResearchQA, overall EM). Default: decay = .999, β_d=0.5.

α	Agr	EM
0	–	– (format collapse)
0.1	.885	.46
0.2	.893	.49
0.4 (default)	.900	.52
0.8	.895	.50

Table 16: Preservation loss weight ablation (Qwen3-14B, ResearchQA).

	HB		RQA		FLASK	
	Base	+S	Base	+S	Base	+S
Qwen3-8B	405	611	453	691	230	290
Qwen3-14B	425	659	480	764	253	319
Qwen3-32B	466	762	501	821	262	594
Llama-3.1-8B	273	465	405	520	200	209
Avg Δ	+220		+259		+117	

Table 17: Average output tokens per sample (joint mode). +S = +SARA.

Base Model (joint)	+SARA (joint)
R1 Analysis: The response attempts to address the instruction by identifying the concept as "business ethics management." However, it does not fully answer the question, which is asking for the term that fills in the blank. The response is somewhat relevant but misses the key requirement of directly providing the term that completes the sentence. **Verdict: 4/5.** (iso: 3)	**R1 Analysis**: The response attempts to address the question by identifying "business ethics management" as the concept being described. However, it does not fully or clearly define the term, nor does it provide a complete or precise answer to the question. The response is somewhat relevant but lacks the clarity and completeness expected. **Verdict: 3/5.** (iso: 3)
R2 Analysis: The response does not provide any factual evidence or citation to support the claim that the term is "business ethics management." While the concept is plausible, there is no reliable source or background knowledge explicitly referenced to confirm its accuracy. **Verdict: 2/5.** (iso: 3)	**R2 Analysis**: The assistant's response states that the direct attempt to manage ethical issues through specific policies, practices, and programs is "business ethics management." However, the response lacks any citation or reference to a reliable source such as Wikipedia or any other authoritative text. While the concept may be accurate, the support for the response is incomplete and not fully reliable. **Verdict: 3/5.** (iso: 3)
R3 Analysis: The response is short and straightforward, with minimal ambiguity. It is generally understandable, though it could be more clearly structured to directly answer the question by providing the term in the blank. **Verdict: 4/5.** (iso: 4)	**R3 Analysis**: The response is clear and well-structured, beginning with a statement that acknowledges the need to think step by step and referencing a source (Wikipedia articles on business ethics) for support. The answer directly addresses the question and provides a concise explanation. The organization is logical and easy to follow. **Verdict: 5/5.** (iso: 5)
Total: 4+2+4=10	**Total: 3+3+5=11**

Table 18: Full joint-mode outputs for the case study. The highlighted region in the base model's R2 shows absolutist phrasing ("does not provide *any*") absent from its own isolation output, where it acknowledges partial accuracy. Parenthetical annotations show each model's isolation verdict for comparison. SARA matches isolation on all three rubrics; the base model deviates on R1 (+1) and R2 (–1).