E-star 12B: Reliable Rubric-Following and Domain-Adaptive SLM Evaluator for Korean Industrial Settings

Yonghoon Kwon[2*†] **Heondeuk Lee**[1*] **Barom Kang**[1]
[1]DATUMO INC. [2]Yonsei University
{heondeuk.lee, rkdqkfha123}@selectstar.ai
yh1013@yonsei.ac.kr

Abstract

Automatic evaluation in industrial settings requires models to apply natural language rubrics reliably under language and domain shift, often without reference answers or access to proprietary models. We present E-Star-12B, a 12B-parameter evaluator for Korean industrial environments that combines structured outputs—feedback, highlight, and decision—with high-confidence training data constructed through multi-stage consensus filtering. We introduce two benchmarks: Ko Feedback Bench, which measures rubric following under Korean language transfer, and RAG Quality Bench, which evaluates domain-specific judgment in financial and legal settings. E-Star-12B achieves the strongest rubric alignment among small language models on Ko Feedback Bench, improving Pearson correlation by +0.173 over its base model. After lightweight domain adaptation, it narrows the gap to reference frontier models on RAG Quality Bench. Across the five SLMs we test, we observe a trend in which stronger rubric following and evaluator-structured outputs are associated with more stable domain adaptation.

1 Introduction

The evaluation of natural language generation systems still largely depends on manual judgments by domain experts. Although such approaches offer high fidelity, they are expensive and inherently unscalable(Li et al., 2023), requiring repeated calibration whenever evaluation criteria evolve. To address this, reference-based metrics (e.g., BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007)) and model-based similarity measures (e.g., BERTScore (Zhang et al., 2020), BARTScore (Yuan et al., 2021)) have been widely adopted. However, these methods are fundamentally limited in scenarios where evaluation requires interpreting and applying natural language rubrics. In practical industrial settings, evaluation targets are typically open-ended, making it infeasible to construct ground-truth references for each instance. As a result, the LLM-as-a-Judge paradigm, which uses language models as evaluators, has recently gained significant traction (Zheng et al., 2023).

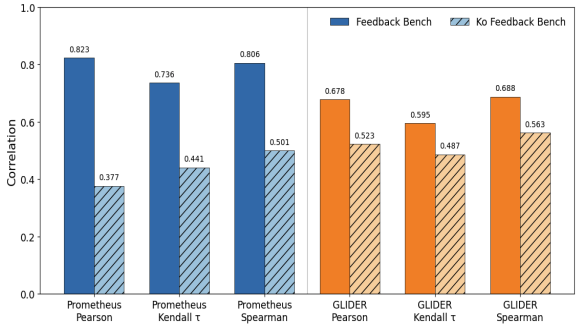
Prometheus (Kim et al., 2024a,b) introduced open-source rubric-based evaluators, and GLIDER (Deshpande et al., 2024) added span highlighting for interpretability with a compact evaluator model. However, such variations remain largely tied to predefined templates and training-time benchmark distributions, leaving unclear how well these evaluators generalize under novel rubric settings or domain shift. We empirically examine this generalization gap along two dimensions (Figure 1). First, Figure 1(a) demonstrates that existing evaluators are vulnerable to language shift. When Feedback Bench (Kim et al., 2024a) instances are translated into Korean (Ko Feedback Bench), both Prometheus and GLIDER show consistent drops across Pearson, Kendall τ, and Spearman correlations, despite identical rubric semantics and evaluation categories. One possible explanation is that current evaluators may overfit surface-level patterns rather than capture rubric semantics in a generalizable way.

Second, Figure 1(b) evaluates transfer to domain-specific settings. Even after additional fine-tuning on a 1K domain-specific instruction set, substantial performance gaps remain. Compared to their Feedback Bench performance (dashed lines), both models exhibit notable degradation on real-world retrieval-augmented generation (RAG) evaluation, particularly in Faithfulness and Context Relevancy.

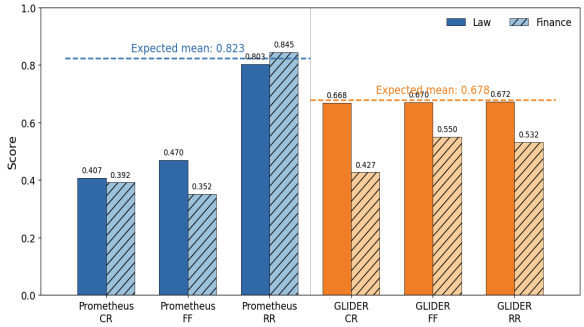
Together, these findings indicate that practical evaluators require two conceptually distinct but empirically related capabilities: (1) reliable interpretation and application of diverse natural language rubrics (*rubric following*), and (2) adaptation to

*Equal contribution.
†Work performed by the author while at DATUMO INC.

Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM), pages 456–471



(a) Performance drops from Feedback Bench to Ko Feedback Bench across all correlation metrics for both Prometheus and GLIDER.



(b) Performance on RAG Quality across Law and Finance domains. Dashed lines indicate Feedback Bench performance.

Figure 1: **Evaluator performance under distribution shift.** (a) Comparison between Feedback Bench and Ko Feedback Bench. (b) Performance degradation on real-world RAG evaluation.

domain-specific evaluation contexts (*domain adaptation*). In many industrial settings, strict data governance policies prevent the use of external API-based LLMs, creating demand for small language model (SLM) based evaluators despite their limited capacity. This raises a key research question: *how can these capabilities be learned together, and does rubric-focused training provide a stable base for later domain adaptation under limited model capacity?*

We propose an SLM-based evaluator tailored to Korean industrial environments, combining high-confidence training data constructed via multi-stage consensus-based filtering with a structured evaluation format. We further introduce Ko Feedback Bench for rubric-following evaluation and RAG Quality Bench for domain-specific evaluation in financial and legal settings.

Our contributions are threefold:

- **Rubric Transfer Gap.** We identify and quantify *the rubric transfer gap*—a failure to maintain evaluation consistency under language and rubric shifts—and demonstrate it in Korean industrial settings. To address this gap, we propose an SLM-based evaluator that mitigates this gap and narrows the distance to reference frontier baselines at the 12B scale.
- **Benchmark Suite.** We introduce a dual-benchmark suite that separates general rubric-following (Ko Feedback Bench) from domain-specific evaluation (RAG Quality Bench), enabling fine-grained analysis of evaluator capabilities.
- **Domain Adaptation Analysis.** We study domain-specific fine-tuning across the five SLMs we test and observe a trend: models with stronger rubric following or evaluator-structured behavior tend to adapt more stably to domain-specific RAG evaluation.

2 Related Work

Our work draws on three lines of research: LLM-as-a-Judge, RAG evaluation, and multi-agent debate.

2.1 LLM-as-a-Judge and Evaluator Models

The LLM-as-a-Judge paradigm (Zheng et al., 2023) uses large language models as scalable proxies for human evaluation, and G-Eval (Liu et al., 2023) refined it via chain-of-thought form-filling. Surveys (Gu et al., 2024; Li et al., 2025) note persistent issues such as position bias and limited robustness under distribution shift (Ye et al., 2024; Park et al., 2024). Open evaluators have been developed to reduce reliance on proprietary models: Prometheus (Kim et al., 2024a,b) introduced rubric-based scoring, and GLIDER (Deshpande et al., 2024) added span highlighting for interpretability. Building on these models, we additionally study evaluator behavior under language and domain shift (Doddapaneni et al., 2024b) for non-English industrial settings. Prior work also shows that evaluation performance can degrade under cross-lingual transfer, especially in lower-resource or non-English settings (Hada et al., 2024; Doddapaneni et al., 2024a; Wu et al., 2024). While these studies focus on overall multilingual evaluation quality, we examine a more specific rubric-following transfer gap in Korean industrial evaluation.

2.2 RAG Evaluation

RAGAS (Es et al., 2023) introduced reference-free metrics for faithfulness, context relevancy, and answer relevancy via LLM-based decomposition, and ARES (Saad-Falcon et al., 2024) fine-tuned lightweight LM judges with prediction-powered inference. These provide automated scoring at inference time; our work complements them by *training* compact evaluators that apply domain-specific rubrics without frontier-model access, which is the focus of our RAG Quality Bench.

2.3 Multi-Agent Debate

Multi-agent debate improves factuality through iterative cross-examination (Du et al., 2024) and has been extended to evaluation with adaptive stability detection (Hu et al., 2025). Rather than as an inference-time component, we use debate as offline data curation, producing high-confidence training labels that complement majority-voting approaches (Verga et al., 2024).

3 Evaluator Design

We design an SLM-based evaluator for Korean rubric-based evaluation. Here, *rubric following* means criterion-conditioned evaluation: parsing a natural-language rubric, applying each criterion to a response, grounding the judgment in evidence, and producing the required score, which is narrower than general instruction following. Our proposed model is built on Gemma-3-12B-IT (Gemma Team, 2025) and trained via supervised fine-tuning (SFT), guided by two core design principles. First, we define the evaluator output using a structured evaluation format, so that both the judgment for each rubric criterion and its supporting evidence are made explicit. Second, we construct the training data through a multi-stage consensus-based filtering pipeline, encouraging the model to learn rubric-application patterns with high inter-model agreement rather than the idiosyncratic judgment patterns of any single model. In the following, we describe the evaluation output format in Section 3.1 and the training data construction process in Section 3.2.

3.1 Evaluation Output Structure

The proposed evaluator generates three components—*feedback*, *highlight*, and *decision*—for a given rubric and model output. This structure is designed to encourage the evaluator to follow an explicit process of criterion-level judgment, evidence identification, and score determination, rather than directly predicting a score. Compared with representative output formats used in prior evaluator models, Prometheus generates free-form feedback and a score, while GLIDER improves interpretability by additionally producing highlighted spans. Our output structure is inspired by GLIDER, but assigns more explicit roles to each component: feedback explains the criterion-based judgment, highlight marks the key spans supporting the score, and decision produces the final score. The full prompt template is provided in Appendix C.

Feedback The feedback component provides criterion-level judgments, indicating which rubric items are satisfied or violated. This itemized structure makes the evaluator's reasoning traceable and provides structural signals for learning diverse rubrics.

Highlight The highlight component marks evidence spans that support the evaluation judgment, anchoring scores to verifiable text. This is especially useful for Faithfulness-style settings where plausible but unsupported claims can otherwise receive inflated scores.

Decision The decision component aggregates the preceding feedback and highlight into the final score, making score prediction the endpoint of the structured evaluation process rather than an isolated label prediction.

3.2 Training Data Construction

To construct the SFT training data for the proposed evaluator, we use K2-Feedback (HAERAE-HUB, 2024) as the seed dataset and build a high-confidence training set through a three-stage filtering pipeline. During filtering, we employ multi-round debate among small frontier models as a mechanism for validating label reliability. We use *small frontier models* to refer to compact commercial frontier-family APIs, while *frontier models* in experiments denote stronger reference baselines. However, debate is used only as an auxiliary tool for training data refinement and benchmark reconstruction (Section 4), and is not applied during inference of the proposed evaluator.

We adopt *multi-round debate rather than majority voting* so that models can mutually challenge their judgments and uncover criteria missed by

independent aggregation (Hu et al., 2025), yielding high-confidence labels that reflect fine-grained rubric criteria (Appendix A).

The data size evolves through the pipeline as 99.7K (raw) → 26K (Stage 1) → 8K (Stage 2) → 6K (final). Appendix D shows that the final 6K subset substantially outperforms training on the raw 99K data. Detailed training and LoRA hyperparameters are reported in Appendix G.

Stage 1: Initial agreement across open-source models (99.7K → 26K) For the original 99.7K instances in K2-Feedback, we first generated evaluation outputs using two recently released open-source instruct models, Qwen3-30B-A3B-Instruct-2507 and Qwen3-Next-80B-A3B-Instruct (Qwen Team, 2025). We then selected only the samples for which the two models produced identical evaluation results, meaning the same parsed final score rather than verbatim rationales, resulting in an initial agreement subset of approximately 26K instances. The goal of this stage is to extract samples where rubric application patterns are relatively stable across models.

Stage 2: Balancing agreement and disagreement with the base model (26K → 8K) For the 26K candidate subset obtained in Stage 1, we further generated evaluation outputs using Gemma, the base model of our target evaluator. We then constructed the dataset to include both cases where Gemma's judgment agrees with the Qwen consensus and cases where it disagrees. This prevents bias toward easy samples while incorporating boundary cases where rubric application is ambiguous, consistent with prior observations that high-disagreement cases are particularly informative for model improvement (Baumler et al., 2023).

Stage 3: Reliability filtering with frontier models (8K → 6K) Finally, to improve label reliability, we use the single-pass evaluation output of GPT-5.2 (OpenAI, 2025a) as an additional filtering signal. Specifically, we retain only samples for which the GPT-5.2 judgment agrees with the debate outcome produced by small frontier models Claude Haiku 4.5 (Anthropic, 2025), GPT-4.1-mini (OpenAI, 2025c), and GPT-5-mini (Singh et al., 2025), as well as samples for which the GPT-5.2 judgment also agrees with the Qwen-family consensus. This process yields a final training set of approximately 6K instances. This preserves the agreement/disagreement diversity from Stage 2 while ensuring labels are supported by multiple strong reference groups; the final set retains 3,373 disagreement samples, confirming filtering preserves diversity. GPT-5.2 is used as a single-reference signal due to cost.

4 Experimental Setup

We describe the benchmarks (Section 4.1), baselines (Section 4.2), and metrics (Section 4.3). For Ko Feedback Bench, all models are evaluated without additional fine-tuning. For RAG Quality Bench, open-weight baseline models and E-Star undergo LoRA fine-tuning (Hu et al., 2022) with a shared 1K domain instruction set, while frontier models are evaluated without additional training. This asymmetry reflects a practical deployment setting: API-based frontier models are often unavailable for local adaptation under data-governance constraints, whereas open-weight SLMs can be adapted in-house.

4.1 Benchmarks

In this work, we construct two complementary benchmarks to separately evaluate the two core capabilities of the proposed evaluator: rubric following and domain adaptation. Ko Feedback Bench measures the general ability to interpret and apply diverse rubric criteria, while RAG Quality Bench evaluates adaptation to domain-specific evaluation settings in financial and legal contexts.

4.1.1 Feedback Bench & Ko Feedback Bench

Ko Feedback Bench is a Korean benchmark designed to evaluate rubric-following capability. For fair comparison, we also reconstruct the original Feedback Bench under the same evaluation setting (i.e., *reference-free* and *debate-based relabeling*). Ko Feedback Bench is built by translating the 1,000 evaluation instances from Feedback Bench into Korean. The original rubric category structure is preserved, while ensuring that the instruction, rubric, and response are naturally expressed in Korean.

Translation and relabeling. The instruction, rubric, and model response of each instance are translated into Korean using Translation Agent (Ng et al., 2024), an agentic machine-translation workflow that first produces an initial translation, then reflects on potential improvements, and finally revises the translation based on the reflection. Instead of directly using the original English labels, we reassign labels to the translated instances using

the same debate-based consensus procedure described in Section 3.2. We treat translation quality as a limitation; consistent reference-frontier performance across both benchmarks (Table 1) and frontier-model participation in the relabeling debate provide indirect validity checks.

Reference-free setting. We exclude reference answers from the evaluation input, since references are often unavailable in industrial settings and their inclusion may cause evaluators to rely on reference matching rather than rubric interpretation. The impact of this choice is analyzed in Section 5.3.2.

4.1.2 RAG Quality Bench

RAG Quality Bench is a benchmark designed to evaluate domain-specific evaluation capability. It is constructed based on financial and legal machine reading comprehension datasets from AI Hub (AI Hub, 2022), and reflects realistic RAG evaluation scenarios in these domains.

Construction pipeline. The original dataset consists of domain documents paired with question–answer instances. To simulate realistic RAG evaluation scenarios, we generate model responses for each question using GPT-5-mini (Singh et al., 2025), forming a triplet structure of (document, question, response). We evaluate three criteria: *Faithfulness* measures whether the response is grounded in the document, *Context Relevancy* assesses whether the retrieved document is relevant to the question, and *Response Relevancy* captures whether the response appropriately addresses the question. Each domain–criterion split contains 600 examples. Label distributions are not artificially balanced (e.g., Finance Response Relevancy 4:1, Law Response Relevancy 1.4:1), reflecting the skew typical of deployed RAG evaluation logs. Detailed statistics and majority baselines are in Appendix E.

Labeling. Similar to Ko Feedback Bench, we assign labels to each instance using the debate-based consensus procedure described in Section 3.2. The labels in RAG Quality Bench are defined as a binary scale (pass/fail), with the rationale for this choice described in Section 4.3.

4.1.3 Benchmark Labeling Procedure

For both benchmarks, labels are determined through a debate-based consensus among multiple judge models, following the same procedure described in Section 3.2. This approach mitigates the individual model bias and high variance on ambiguous samples that are common in single-judge labeling, particularly in reference-free or domain-specific settings. For paired binary comparisons on RAG Quality Bench, we additionally use McNemar's test (McNemar, 1947); the main paired differences supporting our domain-adaptation claims are significant at $p < 0.0001$. Appendix D also reports a McNemar analysis showing that the GPT-4.1/GPT-5.2 difference remains significant under both original and debate-relabeled Feedback Bench labels; this scope does not extend to all SLM comparisons. Potential concerns regarding shared methodology between training and evaluation labels are discussed in the Limitations section.

4.2 Baseline Evaluators

To comprehensively evaluate the proposed model, we consider three groups of baselines: general instruct models, specialized evaluator models, and frontier models.

General instruct models. Gemma-3-12B-IT (the base model of our approach) and GPT-oss-20B (OpenAI, 2025b), included as LLM-as-a-Judge baselines.

Specialized evaluator models. Prometheus-8x7B-v2.0 (Kim et al., 2024b) (MoE, ∼47B total / ∼12B active) and GLIDER-3.8B (Deshpande et al., 2024). Scale differences should be considered when interpreting comparisons, particularly for GLIDER.

Frontier models. GPT-5.2 (OpenAI, 2025a) and Claude Sonnet 4.6 (Anthropic, 2026), included as reference frontier baselines for comparison. All models are evaluated under identical prompts and rubric definitions.

4.3 Evaluation Metrics

We use different evaluation metrics depending on the task-specific characteristics of each benchmark.

Ko Feedback Bench. Ko Feedback Bench follows a 5-point rubric-based scoring scheme, where the primary goal is to measure how well the evaluator's scores align with the reference labels. Accordingly, we report *Pearson correlation*, *Spearman correlation*, and *Kendall* τ.

RAG Quality Bench. RAG Quality Bench focuses on determining whether each criterion—Faithfulness, Context Relevancy, and Response

Type	Models	Feedback Bench			Ko Feedback Bench		
		P	Kτ	S	P	Kτ	S
Frontier	GPT-5.2	0.916	0.865	0.911	0.929	0.886	0.925
	Sonnet-4.6	0.840	0.776	0.847	0.820	0.758	0.833
Instruct SLM	Gemma-3-12B-IT	0.810	0.725	0.794	0.653	0.593	0.661
	oss-20b	0.844	0.762	0.839	0.778	0.704	0.779
Evaluator LM	Prometheus-8x7B-v2.0	0.823	0.736	0.806	0.377	0.441	0.501
	GLIDER 3.8B	0.678	0.595	0.688	0.523	0.487	0.563
Ours	**E-Star-12B-Base**	**0.856**	**0.778**	**0.847**	**0.826**	**0.754**	**0.819**

Table 1: **Rubric-following performance on Feedback Bench and Ko Feedback Bench.** Reported metrics are Pearson, Kendall's τ, and Spearman. Bold and underlining indicate the best SLM and frontier results, respectively.

Relevancy—is satisfied. We formulate this as a *binary (pass/fail) classification* task and report *Accuracy*. We adopt a binary scale because (i) these criteria support pass/fail decisions in practice, and (ii) inter-judge agreement was consistently higher under binary than 5-point scoring in pilot experiments. To account for class skew, we compare model accuracy against majority-class baselines (e.g., Finance RR 0.800, Law RR 0.580) in Section 5.2.

5 Results

5.1 Rubric Following: Feedback Bench & Ko Feedback Bench

As shown in Table 1, **E-Star-12B-Base** achieves the best performance among SLMs on Ko Feedback Bench (Pearson 0.826), improving +0.173 over its base model Gemma-3-12B-IT. It also outperforms oss-20b by +0.048, despite the latter having 8B more parameters. Its Pearson score changes from 0.856 on Feedback Bench to 0.826 on Ko Feedback Bench (gap 0.030), whereas Prometheus drops from 0.823 to 0.377 (gap 0.446) and GLIDER from 0.678 to 0.523 (gap 0.155). E-Star-12B-Base therefore exhibits a smaller cross-lingual gap in this Korean transfer setting. Figure 2 shows the Pearson correlation across 10 rubric categories. E-Star-12B-Base achieves the highest performance in most categories, with particularly notable advantages in Global Cultural Context Understanding, Emotional Communication Ability, and Context-Adaptive Communication Ability.

While oss-20b shows slightly stronger results on Error Handling and Professional Language Proficiency, E-Star-12B-Base achieves the best overall average alignment, indicating balanced evaluation capability across diverse rubric types.

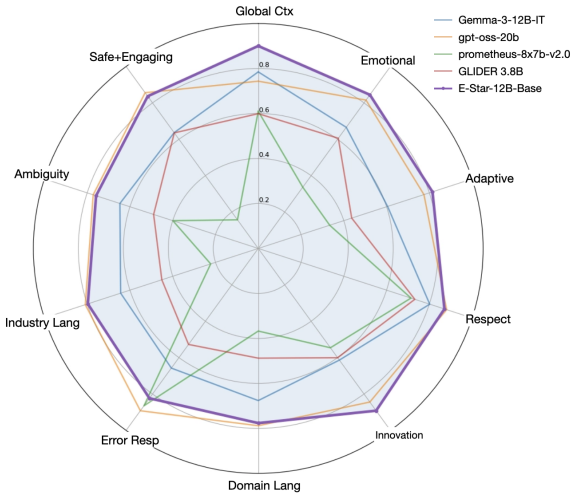


Figure 2: **Pearson correlation across 10 rubric categories on Ko Feedback Bench.** E-Star-12B-Base achieves balanced performance across categories.

5.2 Domain Adaptation: RAG Quality Bench

We now examine whether E-Star-12B-Base's rubric-following capability extends to **domain-specific evaluation** and how effective small-scale domain adaptation can be.

Prior to domain adaptation, E-Star-12B-Base already achieves 0.806 average accuracy, outperforming its base model Gemma-3-12B-IT (0.745) by +0.061 points, while existing evaluator models (Prometheus 0.512, GLIDER 0.567) show substantial degradation. This indicates that our SFT procedure with rubric-focused data can provide meaningful transfer to domain-specific settings.

Table 2 reports the results after additionally fine-tuning each SLM with domain-specific instruction data using LoRA (1K samples).

E-Star-12B-FT increases its overall average from 0.806 to 0.829 (+0.023 points). The gains are concentrated in Faithfulness (LAW +0.024, FINANCE +0.038) and Response Relevancy (LAW

Models	LAW			FINANCE			Avg.
	CR	FF	RR	CR	FF	RR	
Frontier models (no adaptation)							
GPT-5.2	0.846	0.785	0.941	0.882	0.740	0.970	0.861
Sonnet-4.6	0.910	0.786	0.872	0.932	0.845	0.925	0.878
SLMs (before / after LoRA adaptation)							
Gemma-3-12B-IT	0.620	0.658	0.742	0.830	0.713	0.821	0.731
Gemma-3-12B-IT-FT	0.622 (+0.002)	0.743 (+0.085)	0.838 (+0.096)	0.838 (+0.008)	0.737 (+0.024)	0.895 (+0.074)	0.778 (+0.048)
GPT-oss-20b	0.846	0.773	0.870	0.793	0.781	0.900	0.827
GPT-oss-20b-FT	0.838 (-0.008)	0.682 (-0.091)	0.855 (-0.015)	0.745 (-0.048)	0.770 (-0.011)	0.912 (+0.012)	0.800 (-0.027)
Prometheus-8x7B-v2.0	0.392	0.477	0.772	0.386	0.240	0.806	0.512
Prometheus-8x7B-v2.0-FT	0.407 (+0.015)	0.470 (-0.007)	0.803 (+0.031)	0.392 (+0.006)	0.352 (+0.112)	0.845 (+0.039)	0.545 (+0.033)
GLIDER 3.8B	0.657	0.670	0.680	0.432	0.415	0.548	0.567
GLIDER-3.8B-FT	0.668 (+0.011)	0.670 (0.000)	0.672 (-0.008)	0.427 (-0.005)	0.550 (+0.135)	0.532 (-0.016)	0.586 (+0.019)
E-Star-12B-Base	0.853	0.730	0.816	0.835	0.720	0.880	0.806
E-Star-12B-FT	**0.852** (-0.001)	**0.754** (+0.024)	**0.862** (+0.046)	**0.838** (+0.003)	**0.758** (+0.038)	**0.915** (+0.035)	**0.829** (+0.023)

Table 2: **RAG Quality Bench results.** Top: frontier models without domain adaptation. Bottom: SLMs before and after LoRA domain adaptation (1K samples). CR = Context Relevancy, FF = Faithfulness, RR = Response Relevancy. Bold = best among SLMs after adaptation.

+0.046, FINANCE +0.035), while Context Relevancy changes only marginally (LAW −0.001, FINANCE +0.003). This suggests that a small amount of domain data primarily contributes to factuality judgment and response appropriateness.

Gemma-3-12B-IT-FT improves its average from 0.731 to 0.778 (+0.048), but remains substantially below E-Star-12B-FT (0.829), indicating that domain adaptation alone is insufficient.

Across the five adapted SLMs we test, Table 2 shows a trend rather than causal evidence. The two Gemma-family models, especially E-Star-12B-Base with the strongest Ko Feedback Bench performance among SLMs, incorporate domain data relatively stably. Evaluator-specialized models, Prometheus and GLIDER, improve by +0.033 and +0.019 points on average, respectively, but their absolute performance remains low, suggesting that small-scale domain adaptation alone is insufficient to overcome their domain transfer limitations. By contrast, oss-20b-FT is the only model whose overall average declines after domain adaptation, dropping from 0.827 to 0.800 (-0.027 points). Most metric-level changes are negative, with particularly large drops on FINANCE (CR) and LAW (FF). Since oss-20b has strong rubric-following on Ko Feedback Bench, rubric following alone appears insufficient. A likely additional factor is evaluator-specific structured training: E-Star-12B-Base learns to organize feedback, evidence, and decisions explicitly, whereas oss-20b is a general instruct model whose judgment patterns can be perturbed by small-scale LoRA updates. This suggests that domain adaptation may be more stable when rubric following is paired with an evaluator-oriented output structure; one possible mechanism is that strongly learned feedback/evidence/decision slots constrain LoRA updates and preserve judgment behavior. We treat this as an observed trend; controlled causal validation is left to future work.

Compared to reference frontier models (Table 2, top), E-Star-12B-FT (0.829) narrows but does not close the average gap to GPT-5.2 (0.861; gap 0.032) and Sonnet-4.6 (0.878; gap 0.049), while outperforming GPT-5.2 on LAW (CR) and FINANCE (FF). Because GPT-5.2 is also used as a filtering signal during data construction, these split-level gains are compatible with domain adaptation beyond direct copying, but do not remove the self-reference concern. Compared with majority-class baselines (Finance RR 0.800, Law RR 0.580), E-Star-12B-FT reaches 0.915 and 0.862 respectively, with margins that cannot be explained by class skew alone, while Prometheus-FT and GLIDER-FT often approach or fall below these baselines.

5.3 Ablation

5.3.1 Effect of Highlight

Table 3 compares the full model with a variant trained without the highlight field (E-Star-12B-Base-w/o H).

On Ko Feedback Bench (Table 3(a)), highlight yields modest but consistent gains (+0.007 Pearson, +0.011 Kendall τ, +0.010 Spearman). The effect is more pronounced on RAG Quality Bench (Table 3(b)), where average performance improves

Model	P	Kτ	S
E-Star-12B-Base-w/o H	0.819	0.743	0.809
E-Star-12B-Base	0.826	0.754	0.819

(a) Ko Feedback Bench.

Model	L-CR	L-FF	L-RR	F-CR	F-FF	F-RR	Avg.
E-Star w/o H	0.827	0.650	0.850	0.838	0.715	0.900	0.797
E-Star	0.853	0.730	0.816	0.835	0.720	0.880	0.806

(b) RAG Quality Bench.

Table 3: **Effect of the highlight field.** The highlight field provides its clearest benefit in Faithfulness metrics, with mixed smaller changes elsewhere.

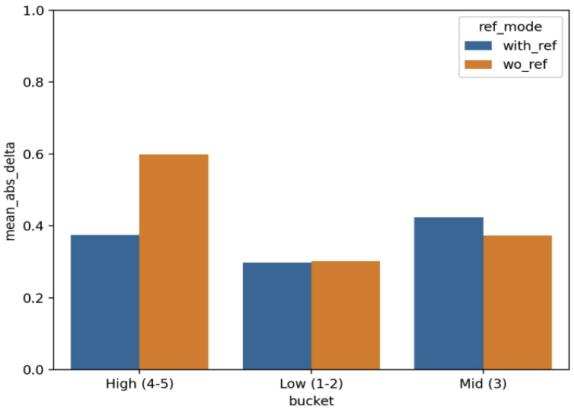


Figure 3: **Effect of the reference-free setting across score buckets.** Average absolute score difference between GPT-5.2 and GPT-4.1 across score buckets. In the with-reference setting, disagreement is concentrated in the mid-score range, whereas in the without-reference setting, the score difference increases most sharply in the high-score range, where it reaches 0.60.

from 0.797 to 0.806, with the largest gain on LAW (FF): +0.080. This suggests that highlight is particularly useful for Faithfulness evaluation, where anchoring judgments to explicit evidence spans may reduce overly high scores for plausible but unsupported responses. The benefit is not uniform across all metrics; small decreases appear on LAW (RR) and FIN (RR), indicating that highlight primarily strengthens evidence grounding rather than uniformly improving performance.

5.3.2 Effect of the Reference-Free Setting

We analyze how the reference-free setting (Section 4.1.1) affects evaluation stability by comparing the score difference Δ between GPT-5.2 and GPT-4.1. The average absolute difference $|\Delta|$ increases from 0.358 (with reference) to 0.459 (without reference), indicating that the reference-free setting is associated with larger inter-judge disagreement. Additional distribution and score-pair views are provided in Appendix F.

As shown in Figure 3, disagreement shifts from the mid-score range (with reference) to the high-score range (without reference), where the average absolute score difference reaches 0.60. This indicates that reference removal is associated with the strongest divergence on samples that would otherwise be rated as "good," shifting the location of inter-judge disagreement. Given that references are often unavailable in industrial settings, this supports the reference-free setup as a more deployment-aligned protocol for assessing practical evaluator performance.

6 Conclusion

We presented **E-Star-12B**, an SLM-based evaluator for rubric-based automatic evaluation in Korean industrial settings, combining a structured output format (*feedback*, *highlight*, and *decision*) with high-confidence training data via multi-stage consensus filtering. E-Star-12B-Base attains the strongest rubric alignment among compared SLMs on Ko Feedback Bench (+0.173 Pearson over its base model), and after lightweight LoRA adaptation, E-Star-12B-FT narrows the average gap to reference frontier baselines on financial and legal RAG evaluation (0.829 vs. 0.861/0.878). Across the five SLMs we tested, evaluators with stronger rubric following or evaluator-structured outputs tended to adapt more stably to domain-specific evaluation, while a general instruct model degraded under the same adaptation, suggesting that structured rubric following may serve as a scaffold for domain adaptation and motivating a rubric-first, domain-later strategy. Future work includes human evaluation by domain experts, retrieval-augmented evaluation for improving Faithfulness, and extension to multilingual settings.

Limitations

Single-language scope. Our experiments should be read as a case study of Korean industrial evaluation, where strict data-governance requirements, domain-specific RAG use cases, and reference-free evaluation are common. The proposed framework is language-agnostic in principle because it relies on structured evaluator outputs and consensus-based data construction rather than Korean-specific model components. Nevertheless, we do not test

typologically diverse languages, and multilingual validation remains necessary before claiming broad cross-lingual generality.

Translation-quality validation. Ko Feedback Bench is constructed with Translation Agent, a reflection-based machine-translation workflow, rather than simple one-pass machine translation. This reduces but does not remove the possibility of translation artifacts. Frontier models' consistent performance across Feedback Bench and Ko Feedback Bench, together with frontier-model participation in debate-based relabeling, provides indirect support for the validity of relative comparisons. However, we do not conduct native-speaker evaluation or automatic translation-quality checks such as BLEU or COMET, so translation noise remains a limitation.

Shared labeling methodology. Both the training data (Section 3.2) and benchmark labels are constructed using the same debate-based consensus procedure. While this makes it difficult to fully separate genuine capability improvements from alignment to the labeling methodology, all baselines are evaluated under the same labels, preserving relative comparisons. GPT-5.2 is also used as a filtering signal and as a reported frontier baseline; E-Star-FT exceeding GPT-5.2 on some splits is compatible with domain-adaptation effects beyond direct distillation, but does not eliminate self-reference concerns. Future work should include human evaluation by domain experts to establish absolute evaluation quality.

Partial output-structure ablation. Section 5.3.1 evaluates the effect of the highlight field and shows that evidence grounding is particularly useful for Faithfulness evaluation. We do not ablate all alternative output formats, such as feedback-only, score-only, or decision-only variants. Therefore, the current ablation supports the usefulness of highlight but does not fully decompose the contribution of every component in the structured output.

Causal limits of the scaffold claim. The domain-adaptation results show a trend among the five SLMs we tested, including two Gemma-family variants: E-Star adapts stably, Gemma improves but remains lower, Prometheus and GLIDER improve from low absolute performance, and oss-20b declines despite strong general rubric-following scores. This suggests that evaluator-oriented rubric-following structure may function as a scaffold for domain adaptation. However, this evidence is observational and statistically limited. A stricter causal test would require controlled experiments over the same base model, isolating rubric-following SFT, output structure, and LoRA domain adaptation; Korean-native SLM baselines would also strengthen the comparison.

Practical cost. Our training-data construction and benchmark relabeling rely on commercial API models, and the structured feedback, highlight, and decision output can increase inference length and latency. These costs should be considered before deployment in low-latency or API-constrained environments.

Acknowledgments

This research used datasets from The Open AI Dataset Project (AI-Hub, S. Korea). All data information can be accessed through AI-Hub.[1]

References

AI Hub. 2022. Financial and legal document machine reading comprehension data. https://www.aihub.or.kr/aihubdata/data/view.do?dataSetSn=71610. AI Hub dataset page; updated 2023-11.

Anthropic. 2025. Claude haiku 4.5 system card. System card, Anthropic.

Anthropic. 2026. Introducing claude sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6. Official model announcement.

Connor Baumler, Anna Sotnikova, and Hal Daumé III. 2023. Which examples should be multiply annotated? active learning when annotators may disagree. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10352–10371, Toronto, Canada. Association for Computational Linguistics.

Darshan Deshpande, Selvan Sunitha Ravi, Sky CH-Wang Wang, Bartosz Mielczarek, Anand Kannappan, and Rebecca Qian. 2024. GLIDER: Grading LLM interactions and decisions using explainable ranking. *arXiv preprint arXiv:2412.14140*.

Sumanth Doddapaneni, Vaibhav Adlakha, Anoop Kunchukuttan, and Mitesh M. Khapra. 2024a. Cross-lingual auto evaluation for assessing multilingual LLMs. *arXiv preprint arXiv:2410.13394*.

[1] https://www.aihub.or.kr

Sumanth Doddapaneni, Negar Arabzadeh, Mitesh M. Khapra, and Anoop Kunchukuttan. 2024b. Finding blind spots in evaluator LLMs with interpretable checklists. *arXiv preprint arXiv:2406.13439*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11733–11763. PMLR.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. RAGAS: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Gemma Team. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2024. A survey on LLM-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Rishav Hada, Varun Gumma, Adrian de Wynter, Harshita Diddee, Mohamed Ahmed, Monojit Choudhury, Kalika Bali, and Sunayana Sitaram. 2024. Are large language model-based evaluators the solution to scaling up multilingual evaluation? In *Findings of the Association for Computational Linguistics: EACL 2024*. Association for Computational Linguistics.

HAERAE-HUB. 2024. K^2-Feedback. https://huggingface.co/datasets/HAERAE-HUB/K2-Feedback. Hugging Face dataset card.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Tianyu Hu, Zhen Tan, Song Wang, Huaizhi Qu, and Tianlong Chen. 2025. Multi-agent debate for LLM judges with adaptive stability detection. *arXiv preprint arXiv:2510.12697*.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. PROMETHEUS: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations*.

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

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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.

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

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Andrew Ng, Joaquin Dominguez, Nedelina Teneva, and John Santerre. 2024. Translation agent: Agentic translation using reflection workflow. https://github.com/andrewyng/translation-agent. GitHub repository.

OpenAI. 2025a. GPT-5.2 system card. System card, OpenAI.

OpenAI. 2025b. gpt-oss-120b & gpt-oss-20b model card. https://openai.com/index/gpt-oss-model-card/. Official model card.

OpenAI. 2025c. Introducing GPT-4.1 model family. https://openai.com/index/gpt-4-1/. Accessed: 2025-07-09.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Junsoo Park, Seungyeon Jwa, Ren Meiying, Daeyoung Kim, and Sanghyuk Choi. 2024. OffsetBias: Leveraging debiased data for tuning evaluators. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1043–1067, Miami, Florida, USA. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. ARES: An automated evaluation framework for retrieval-augmented generation systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 338–354, Mexico City, Mexico. Association for Computational Linguistics.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. OpenAI GPT-5 system card. *arXiv preprint arXiv:2601.03267*.

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

Zhaofeng Wu, Ananth Balashankar, Jonathan Berant, Dragomir Radev, and Adina Williams. 2024. Reuse your rewards: Reward model transfer for zero-shot cross-lingual alignment. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. 2024. Justice or prejudice? quantifying biases in LLM-as-a-judge. *arXiv preprint arXiv:2410.02736*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

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 *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623.

A Rubric Detail Capture via Multi-Round Debate

Figure 4 illustrates how multi-round debate helps uncover fine-grained rubric criteria that are missed in single-pass evaluation. In Round 1, the evaluator incorrectly assigns a passing score after overlooking that the second paragraph begins with uppercase "Curve" rather than the required lowercase "curve." By Round 3, iterative cross-examination identifies this violation and revises the score to 0.

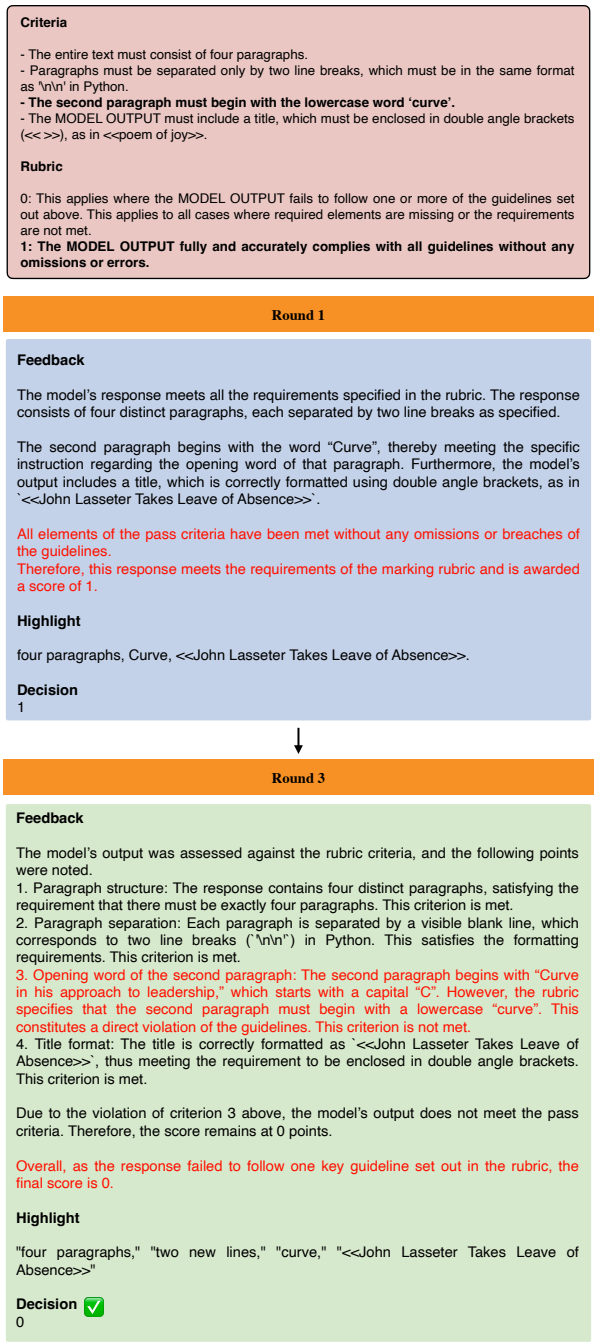


Figure 4: **Multi-round debate example for rubric detail capture.** In Round 1, the evaluator overlooks a fine-grained criterion (lowercase "curve") and assigns a passing score. By Round 3, iterative cross-examination identifies the violation, correcting the score to 0.

B Debate Evaluation Algorithm

The full pipeline is presented in Algorithm 1.

Algorithm 1 Multi-Agent Debate Evaluation

Require: Question q, response y, scoring rubric $\mathcal{R}$, evaluator agents $\mathcal{A} = \{a_1, \ldots, a_N\}$ each with persona p_i and model m_i, max debate rounds R, arbiter model m_{arb}
Ensure: Final evaluation $\hat{e}$

Phase 1: Initialization
$r \leftarrow 0;\ \mathcal{E} \leftarrow \{\}$ ▷ $\mathcal{E}[r, a_i]$: evaluation store keyed by round and agent

Phase 2: Iterative Debate
while $r < R$ **do**
  for each agent $a_i \in \mathcal{A}$ **in parallel do** ▷ fan-out: all agents evaluate concurrently
    if $r = 0$ **then** ▷ independent evaluation; no peer context
      $\pi_i \leftarrow \textsc{PromptR0}(p_i,\ q,\ y,\ \mathcal{R})$
    else ▷ peer-informed evaluation; agents may revise based on evidence
      $P \leftarrow \textsc{PeerEvals}(\mathcal{E},\ r{-}1)$ ▷ aggregate all evaluations from round $r{-}1$
      $\pi_i \leftarrow \textsc{PromptRN}(p_i,\ r,\ q,\ y,\ \mathcal{R},\ P)$
    end if
    $e_i^{(r)} \leftarrow \text{LLM}(\pi_i,\ m_i)$
    $\mathcal{E}[r,\ a_i] \leftarrow e_i^{(r)}$
  end for
  $r \leftarrow r + 1$
end while

Phase 3: Arbiter Judgment
$E \leftarrow \left\{\ \mathcal{E}[R{-}1,\ a_i] \ \mid\ a_i \in \mathcal{A}\ \right\}$ ▷ collect final-round evaluations
$\hat{e} \leftarrow \text{LLM}\big(\textsc{ArbiterPrompt}(q,\ y,\ \mathcal{R},\ E),\ m_{\text{arb}}\big)$ ▷ deterministic arbitration
return $\hat{e}$

Sub-procedures. PROMPTR0 constructs an evaluation prompt with the agent's persona, rubric, and question–response pair, instructing the agent to produce criterion-focused feedback grounded in direct quotes. PROMPTRN extends this by appending peer evaluations from round $r{-}1$ and requiring an evaluator trace that identifies disagreements and justifies its stance with textual evidence; agents retain their score unless peers provide stronger rubric-grounded evidence. PEEREVALS aggregates all agent evaluations from a given round into a structured block. ARBITERPROMPT presents the final-round evaluations to the arbiter with instructions to independently verify each claim against the rubric and response content.

Output format. All evaluations follow a structured format: ⟨`feedback`⟩ (criterion-focused analysis), ⟨`highlight`⟩ (key verbatim spans from the response), and ⟨`decision`⟩ (integer score or pairwise preference).

C Evaluator Input Format

Figure 5 shows the full prompt template. The system prompt defines a five-step evaluation procedure designed to enforce strict rubric adherence, prohibiting external standards and requiring verbatim evidence in feedback. The user prompt supplies the evaluation instance, and the evaluator is instructed to produce all outputs (<feedback>, <highlight>, <decision>) in the input language, enabling cross-lingual evaluation without prompt modification.

System prompt:

```
You are a rubric evaluator.
Your task is to evaluate a response strictly and only according to the provided pass criteria and scoring rubric.
In your output, return the final evaluation (the three output tags: <feedback>, <highlight>, and <decision>).

# Evaluation Procedure (must follow all steps):
1. First, carefully read the Data to Evaluate, the pass criteria, and the scoring rubric to fully understand the requirements.
2. Evaluate the response only against the given criteria: do not introduce external standards, do not reward style unless the rubric explicitly allows it, and judge by absolute rubric definitions rather than relative comparisons.
3. Re-check fine-grained details in the response and the rubric, ensuring any tags (if present) are correctly mapped to the pass criteria and that small deviations are not overlooked.
4. Write criterion-focused feedback that explicitly references the rubric, quoting exact words or phrases from the response when they are decisive, and clearly stating which criteria are satisfied and which are violated.
5. Finally, extract the key verbatim spans that most influenced your judgment and assign the final score according to the scoring rubric.
```

User prompt:

```
# Data to Evaluate

### Question
{question}

### Model Response
{response}

# Rubric {rubric} You MUST write ALL output (<feedback>, <highlight>, <decision>) in the SAME language as the input question and response being evaluated. If the input is in Korean, your entire output MUST be in Korean.

# Output Format:
<feedback>
Write detailed feedback (reasons) that strictly evaluates the quality of the response using only the given scoring rubric. Do not explicitly state the score in a sentence (e.g., "Therefore, the score is …").
</feedback>
<highlight>
List of words or phrases that you believe are the most important in determining the score.
</highlight>
<decision>
Provide the final integer score assigned based on the scoring rubric.
</decision>
```

Figure 5: **Evaluator prompt template.** The system prompt defines the evaluation procedure, while the user prompt provides the question, model response, and rubric for each instance.

D Filtering and Relabeling Validation

Table 4 reports a McNemar analysis comparing GPT-4.1 and GPT-5.2 under two label conditions: the original Feedback Bench labels and the debate-relabeled Ko Feedback Bench labels. In both conditions, the paired correctness difference remains statistically significant ($p < 0.0001$), suggesting that this particular model difference is preserved under relabeling. This analysis does not directly test robustness for all SLM comparisons.

Correctness pattern	Original labels	Debate relabeling
Both correct (a)	552	445
GPT-4.1 only correct (b)	69	87
GPT-5.2 only correct (c)	210	174
Both incorrect (d)	169	294
p-value	< 0.0001	< 0.0001

Table 4: **McNemar analysis under original and debate-relabeled conditions.** The paired correctness patterns compare GPT-4.1 and GPT-5.2 on Feedback Bench-style evaluation labels.

Tables 5 and 6 compare training with the raw 99K K2-Feedback data and the final 6K high-confidence subset produced by the filtering pipeline. In this evaluator-training setting, the filtered 6K set yields substantially stronger performance on both Ko Feedback Bench and RAG Quality Bench than the raw 99K data, indicating the benefit of our reliability filtering rather than a general rule about data volume.

Data size	Pearson	Kendall τ	Spearman
99K	0.598	0.565	0.651
6K	0.826 (+0.228)	0.754 (+0.189)	0.819 (+0.168)

Table 5: **Effect of data filtering on Ko Feedback Bench.** The final 6K high-confidence subset outperforms training on the raw 99K data.

Data size	L-CR	L-FF	L-RR	F-CR	F-FF	F-RR	Avg.
99K	0.385	0.481	0.636	0.431	0.630	0.610	0.528
6K	0.853 (+0.468)	0.730 (+0.249)	0.816 (+0.180)	0.835 (+0.404)	0.720 (+0.090)	0.880 (+0.270)	0.806 (+0.277)

Table 6: **Effect of data filtering on RAG Quality Bench.** L = Law, F = Finance, CR = Context Relevancy, FF = Faithfulness, and RR = Response Relevancy.

E RAG Quality Evaluation Details

RAG Quality Bench consists of three evaluation criteria (Es et al., 2023), each requiring a distinct input structure:

Context Relevancy takes the question and retrieved document as input and evaluates how much relevant information the document contains with respect to the query.

Response Relevancy takes the question and model response as input and measures how faithfully the generated response addresses the requirements of the query.

Faithfulness takes the retrieved document and model response as input and evaluates whether the response is factually consistent with the document content, focusing on identifying unsupported information or unfounded reasoning.

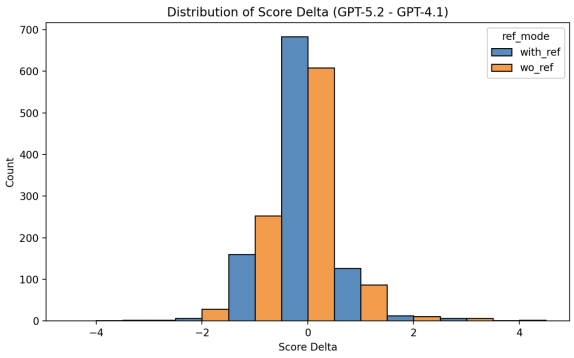
Rather than using the original RAGAS scoring procedure, we reformulate these criteria under a binary (pass/fail) evaluation setting tailored to financial and legal domains with long-context inputs. Each criterion uses a criterion-specific input format, and we construct 600 samples per domain per criterion. Notably, input lengths for Context Relevancy and Faithfulness evaluation average approximately 29K characters, requiring long-context comprehension and evidence tracking rather than simple keyword matching.

Domain / Criterion	Pass	Fail	Maj. base
Law / Response Relevancy	348	252	0.580
Finance / Response Relevancy	480	120	0.800

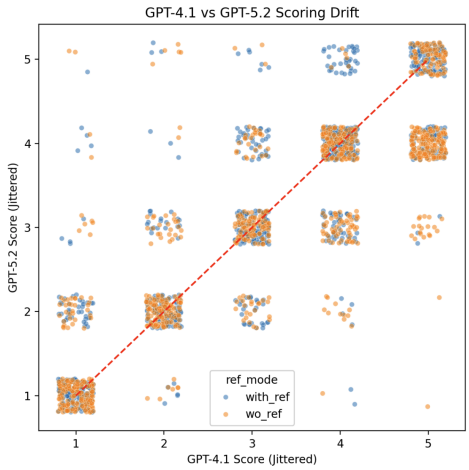
Table 7: **Response Relevancy label distribution in RAG Quality Bench.** Each domain contains 600 examples for the criterion.

F Reference-Free Scoring Drift

Figure 6 provides complementary views of the reference-free analysis in Section 5.3.2. The score-delta distribution remains centered near zero, but the reference-free setting is associated with more off-zero mass, and the jittered score plot shows that disagreement is not limited to random noise around the diagonal.



(a) Distribution of score differences.



(b) Jittered GPT-4.1/GPT-5.2 score pairs.

Figure 6: **Judge scoring drift under reference-free evaluation.** The distribution and jittered score views show how removing references changes the pattern of disagreement between GPT-4.1 and GPT-5.2.

G Training Details

Table 8 summarizes the hyperparameters for LoRA-based domain adaptation.

Hyperparameter	Value
LoRA rank (r) / alpha	4 / 16
LoRA target modules	all-linear
Adaptation samples	1,000
Learning rate	1.0×10^{-6}
Number of epochs	3
Effective batch size	16

Table 8: Hyperparameters for LoRA-based domain adaptation. Effective batch size $= 1 \times 4$ GPUs $\times$ 4 accum. steps.