
CAN JEV JUDGE RADIOLOGY REPORTS? EVALUATING A SYSTEM ONE MODEL FOR CLINICAL FACTUALITY

A PREPRINT

Jiaju Huang¹, Hao Yang¹, Xinyu Ma¹, Xinglong Liang^{2,3},
Kunyan Cai¹, Junqiang Ma¹, Shaobin Chen¹, Yue Sun¹, Tao Tan^{1,*}

¹Intelligent Medical Computing Laboratory, Faculty of Applied Sciences,
Macao Polytechnic University

²Department of Radiology, Netherlands Cancer Institute,
Amsterdam, the Netherlands

³Department of Radiology and Nuclear Medicine,
Radboud University Medical Center, Nijmegen, the Netherlands

*Corresponding author: Tao Tan, taotanjs@gmail.com

September 23, 2026

ABSTRACT

An AI-generated radiology report can resemble a physician’s report while omitting an abnormality, adding an unsupported finding, or reversing its presence. Measuring these factual differences is essential for evaluating report generators. We study Jev, a System One decision model, as a simple, low-cost judge of agreement with physician-written reference reports. Our evaluator checks whether each statement is supported by the other report and combines these judgments in both directions to capture unsupported claims and omissions. A single-question configuration reaches Kendall correlations of 0.573 on RadEvalX and 0.398 on RadEvalExpert with expert error counts, outperforming an open natural language inference judge under matched decomposition and aggregation. One support question per statement retains similar expert agreement to seven while using 43–45% fewer judgment input tokens. Judgment API calls cost less than US\$0.03 per 100 report pairs, excluding local decomposition. In a separate controlled-error test, Jev detects false negation with an AUROC of 0.977. Local RadMatch achieves stronger agreement on clinically significant errors in both expert datasets and on total errors in the shared RadEvalExpert subset. Finding-count and error-scope analyses show that benchmark agreement reflects report size and error definitions as well as medical error detection. These results support Jev as a practical judgment component for measuring factual differences in generated radiology reports and identify where more elaborate evaluation remains valuable.

Keywords Radiology report evaluation · Jev · System One · Factuality

1 Introduction

A generated radiology report can read naturally and still change the medical story. It may add a finding that the physician’s report never mentions, leave out an abnormality, or describe it on the wrong side. Changing “a pleural effusion” to “no pleural effusion” preserves most words while reversing the finding. For researchers developing report generators, these are the differences an evaluator needs to capture: which assertions disagree with the reference, and which findings have been left out?

This task calls for many small judgments. For each statement, does the reference support it, contradict it, or leave it unaddressed? Checking in the reverse direction asks whether the generated report covers the reference’s findings. Jev offers a direct interface for making these decisions: its System One model returns typed answers with probabilities

that software can combine into a report score [16]. We ask whether Jev can measure factual differences in generated radiology reports with just one support question per statement.

We study Jev as a reference-based report evaluator. We extract statements from both reports, judge each against the other report, and aggregate their discrepancies (Figure 1). This design builds on report decomposition and bidirectional entailment, particularly RadFact [2]. We examine what Jev contributes within this framework and how much complexity is needed to use it effectively. Expert-annotated reports test whether the scores reflect overall error burden; controlled errors reveal which medical changes the judge detects. Component comparisons, token costs, and repeated calls test whether the evaluator is economical and stable enough for repeated report-generator comparisons.

The results support a simple configuration. One support question per statement produces expert agreement close to that of the seven-question design while reducing judgment API input cost by 43–45%. The resulting judgments cost under three cents per hundred report pairs, excluding local decomposition. Under matched decomposition and aggregation, this configuration outperforms an open natural language inference (NLI) judge. In a separate sentence-level test, Jev detects false negation—turning a present finding into an absent one—with an AUROC of 0.977. Its report rankings also repeat closely across five runs. Local RadMatch achieves higher agreement on clinically significant errors in both expert datasets and on total errors in the shared RadEvalExpert subset, showing where its more elaborate procedure remains useful.

A report that describes more findings offers more opportunities for errors. An evaluator can therefore correlate with expert error counts by capturing report size, even without identifying each error correctly. Jev exceeds a finding-count baseline in both expert datasets. We complement this comparison with candidate ordering within the same examination and with a separation of factual errors from spelling and repetition perturbations. The latter changes the relative ranking of Jev and NLI.

Our contributions are threefold. First, we evaluate Jev’s ability to measure factual discrepancies between generated and reference radiology reports, using expert error counts and controlled medical changes. Second, we identify a single-question configuration that combines competitive expert agreement with low judgment cost. Third, we characterize what the evaluator captures and how it behaves through analyses of report size, error scope, calibration, and repeatability.

2 Related Work

Radiology report metrics. Report evaluation spans lexical overlap, contextual similarity, extracted clinical structure, and learned error scores. BERTScore compares contextual token representations [22]; RadGraph measures agreement in clinical entities and relations [11]; RaTEScore uses radiology entities and their attributes [23]. RadCliQ combines metric signals to predict expert error counts [20]. The RadEval framework provides common implementations and an expert-annotated evaluation dataset [18]. These methods differ in their targets: a similarity ratio and a predicted number of errors need not rank reports identically.

Language models as report judges. GREEN identifies and describes clinically meaningful discrepancies between reports [14]. RadFact decomposes reports into single-finding phrases and verifies entailment in both directions, yielding logical precision and recall [2]. Its evidence extraction also supports grounded evaluation. RadMatch uses finding-level matching and attribute analysis to produce auditable discrepancy measures [6]. Our work adopts the established decomposition-and-entailment structure and examines Jev as a judgment engine. CRIMSON adds context-sensitive clinical significance and radiologist preference benchmarks [1]. AtomiMed separates disease and attribute facts and verifies them through bidirectional questions across imaging modalities [17].

Recent work also examines how to configure medical judges. VERT compares judge models, prompts, reasoning settings, ensembling, and fine-tuning on RadEval and RaTE-Eval [3]. Its analysis also discusses how finding counts affect normalized scores. RadSEM combines atomic findings, contradiction-constrained matching, and deterministic weighted scoring [19]. It evaluates sensitivity to graded semantic corruption and synonym–antonym rewrites. We focus on Jev’s typed probability outputs, its medical error detection, and the cost of a minimal question set. Our count baseline and error-scope comparisons examine what expert-agreement scores reward.

Evaluating the evaluators. General LLM-judge studies document position and verbosity biases alongside agreement with human preferences [24]. Length-controlled AlpacaEval uses regression adjustments to reduce preference for longer responses [8]. In summarization, SummEval includes output length among its simple comparison features [9]. Radiology benchmarks add a related measurement issue: reports with more findings offer more opportunities for counted errors. RaTE-Eval already normalizes sentence errors by the number of potential errors [23]. We measure the strength of a finding-count baseline on expert report benchmarks and contrast pooled error burden with candidate ordering within the same examination.

Catch what is wrong. Find what is missing.

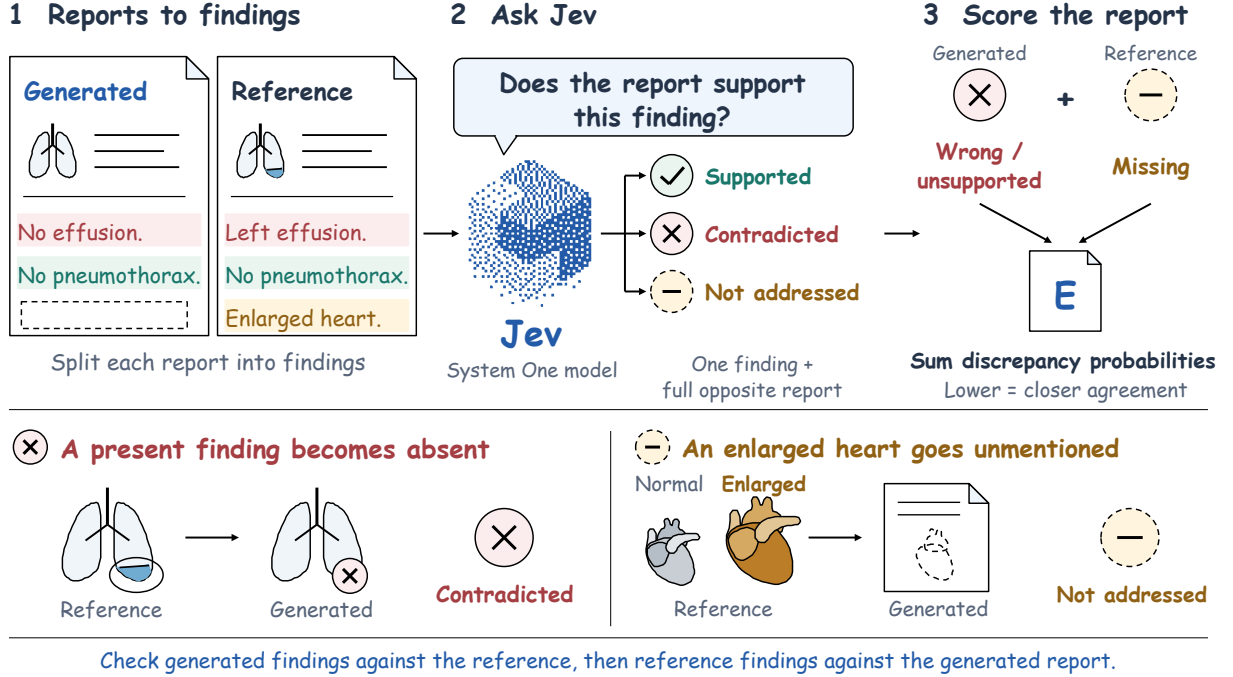


Figure 1: Jev checks what a report gets wrong and what it leaves out. OneQ compares each finding with the complete opposite report in both directions. The constructed example visualizes a denied effusion and an omitted enlarged heart; a normal-sized heart is shown for visual comparison. Jev returns support, contradiction, and noncoverage probabilities. Scores sum candidate-side contradiction/noncoverage with reference-side noncoverage (Eq. 2); lower is better. Model illustration adapted from [TypeSafe AI](#).

Probability quality and benchmark interpretation. Probabilistic outputs allow soft scores and task-specific calibration, but probability values alone do not establish calibration [10]. We compare Jev’s raw probabilities with those of a public DeBERTa NLI checkpoint [12], and evaluate post-hoc calibration on held-out studies. Expert datasets such as ReXVal [21], RadEvalX [5], and RadEvalExpert [18] measure report-level agreement, whereas ReXErr [15] provides controlled local errors. We use both forms of evidence and test how simple count signals and error definitions affect their conclusions.

3 Jev as a Radiology Report Judge

3.1 Task and statement units

Given a generated report C and a physician-written reference report R , we assess which findings disagree and which are left unaddressed. Let $c_1, \dots, c_n$ and $r_1, \dots, r_m$ denote the statements extracted from the two reports.

We implement two granularities. L0 applies sentence segmentation and removes pure section headings. L1 uses a local Qwen3.5-9B model to split each sentence into atomic findings while preserving negation, uncertainty, laterality, severity, measurements, and comparison phrases. Invalid JSON falls back to the original sentence. The reported OneQ configuration uses L1. L0 is a separate ablation.

3.2 One question, three outcomes

For a statement x and an opposing report Y , Jev returns

$$(p_s(x, Y), p_c(x, Y), p_u(x, Y)), \quad p_s + p_c + p_u \approx 1, \quad (1)$$

for *supported*, *contradicted*, and *not addressed*. We use the returned probabilities at API precision without renormalization. The prompt asks whether the report supports the claim about an imaging finding. Its criteria distinguish compatible descriptions, incompatible presence or attributes, and findings absent from the report text. For example, “left pleural

effusion” is contradicted by an explicit denial of left pleural effusion; a report that never addresses effusion supplies no support for the claim. Here, *not addressed* denotes missing textual evidence.

The same question is applied to every candidate statement against R and every reference statement against C . OneQ therefore means one question per statement–report input; a report pair requires multiple such judgments.

3.3 Bidirectional aggregation

Our primary score sums candidate-side discrepancies and reference-side noncoverage:

$$E(C, R) = \sum_{j=1}^n [p_c(c_j, R) + p_u(c_j, R)] + \sum_{i=1}^m p_u(r_i, C). \quad (2)$$

Lower values indicate greater agreement. The asymmetry avoids adding a second contradiction term from the reference direction. This is a soft discrepancy count over extracted statements; its numerical agreement with expert error counts is evaluated separately from its ranking ability.

We also report soft precision, recall, and F1:

$$P = \frac{1}{n} \sum_j p_s(c_j, R), \quad R_s = \frac{1}{m} \sum_i p_s(r_i, C), \quad F_s = \frac{2PR_s}{P + R_s}. \quad (3)$$

The normalized count is $E/(n + m)$. Empty-side means and undefined harmonic means are set to zero in the implementation. We report counts against count labels and use the normalized and F1 variants to examine agreement as a proportion of report content.

3.4 Configurations and open judge

OneQ uses only the support question. **Full** adds four negation/uncertainty questions, a discrepancy-type question, and a clinical significance question. Full also applies a negation-consistency correction: if the claim and report negation probabilities differ by more than 0.5 while $p_s > 0.5$, half of p_s is moved to p_c . OneQ has no negation outputs, so this correction is inactive. Significance and error-type outputs enter their own variants and do not otherwise alter Eq. 2.

We test sentence-only splitting, shorter criteria, relevance filtering, alternative discrepancy weights, and significance weighting. Relevance filtering asks whether each report sentence concerns the claim’s anatomical entity and retains probabilities above 0.3. An empty selection yields an empty opposing report. We evaluate each component separately against Full.

The open judge uses DeBERTa-v3-large-mnli-fever-anli-ling-wanli [12]. The opposing report is the premise and the statement is the hypothesis. Entailment, contradiction, and neutral probabilities map to p_s, p_c, p_u . It shares L1 statements and Eq. 2 with OneQ; its tokenizer uses a maximum sequence length of 512 tokens. This comparison isolates the judge within the evaluated configuration, including its input handling.

4 Experimental Design

4.1 Datasets and targets

Expert agreement. RadEvalX contains 100 IU-Xray reference reports, each paired with one M2Tr-generated report [5, 7]. Two radiologists reached consensus on significant and insignificant error counts across eight categories. The sample contains 80 abnormal and 20 normal reports, selected after stratification by RadCliQ scores. We evaluate total and clinically significant error counts over all eight categories; a six-category sensitivity analysis excludes the two uncertainty categories (Appendix A). RadEvalExpert contains 624 pairs from 208 studies, with three model-generated candidates per reference [18]. It covers findings and impression sections and supplies total and significant error labels. We keep each study together during uncertainty estimation.

Controlled errors. The ReXErr sentence test set contains 19,859 examples [15]. Excluding 346 neutral rewrites leaves 19,513 examples: 8,723 injected-error sentences and 10,790 unchanged sentences. Each candidate sentence is judged against its complete original report. This experiment uses the raw support probabilities from the Full question set, without atomic decomposition or negation correction. It tests sentence-level detection and calibration separately from the report-level OneQ pipeline. We retain all 11 positive error types and also analyze eight factual-error types and three language-perturbation types (typos, homophones, and repetition), grouped by the dataset’s intervention labels.

We use 2,000 sampled ReXErr report pairs for cost and report-level diagnostics. RaTE-Eval supplies supplementary sentence, paragraph, and synthetic comparison tests [23]. Their targets and processing are described in Appendix A.

4.2 Comparators and implementation

The main comparators include GREEN, RadCliQ, RadGraph F1, BERTScore, Temporal F1, open NLI, and the finding-count baseline $n + m$. RadEval supplies the standard metric implementations [18]. We also run RadFact and RadMatch on all 100 RadEvalX pairs and a fixed random subset of 200 RadEvalExpert pairs. All subset comparisons use exactly the same valid pairs.

Both generative baselines use a local Qwen3.8-27B-FP8 backend on two 32 GB RTX 5090 D GPUs. This replaces the original papers’ judge choices. On the RadEvalExpert subset, RadMatch retried 52 pairs and used validation fallback on 11; its default backend configuration sampled at temperature 1.0. RadFact used temperature zero. On RadEvalX, RadMatch used validation fallback on 8 of 100 pairs; all pairs received a final score. Both baselines disabled thinking. RadFact used an 8,192-token context, and RadMatch used 16,384 tokens to accommodate the complete few-shot prompt, with a 4,096-token output cap. GREEN’s default decoding is greedy.

Our implementation, S1Fact, uses Jev version `jev-1.13.0`. L1 uses temperature-zero local inference. Repeated-call experiments use fresh judgments with decomposition held fixed. Additional prompt and implementation details appear in Appendix B.

Repeatability. On a fixed random subset of 50 RadEvalX pairs, we repeat both Full and OneQ five times, holding decomposition fixed and bypassing the judgment cache. A matched experiment uses a fixed selection of 50 pairs from the 200-pair RadEvalExpert subset, covering 48 studies. We run Jev-Full and Jev-OneQ five times each on the same 616 extracted findings, bypassing the judgment cache. Five runs per baseline setting evaluate GREEN with greedy decoding, RadFact at temperature zero, and RadMatch at temperatures zero and one. The latter is the local backend’s default. Jev holds L1 decomposition fixed; the baselines repeat their local evaluation procedures. We measure exact score agreement, between-run ranking consistency, and variation in correlation with expert total-error counts.

4.3 Statistics and cost

Primary agreement is Kendall’s τ_b , oriented in advance so higher is better: error scores retain their sign against error labels, and similarity scores have their sign reversed. We do not take absolute correlations. Confidence intervals use percentile bootstrap over studies: 2,000 resamples for RadEvalX and 1,000 resamples for RadEvalExpert. Pairwise comparisons first select rows where both methods and the target are finite, then resample the same studies for both methods. Intervals describe individual comparisons and are not adjusted for multiple testing.

On RadEvalExpert, which supplies multiple candidates per study, we separately compute within-study ordering,

$$W = \frac{N_{\text{concordant}} - N_{\text{discordant}}}{N_{\text{all within-study candidate pairs}}}. \quad (4)$$

Ties contribute zero to the numerator and remain in the denominator. This statistic is not tie-corrected τ_b . It holds the reference report fixed within each comparison, while candidate lengths may still differ. Bootstrap copies of a study contribute independent copies of its pair counts.

Controlled-error analyses report AUROC, recall at 0.5, ten-bin expected calibration error (ECE), and Brier score. Error-scope intervals use 400 study resamples. Post-hoc logistic and isotonic calibration fit on one half of the studies and are evaluated on the other half. Raw and calibrated probabilities are compared on the same holdout.

We calculate API cost from input-token usage at \$0.042 per million tokens; output tokens are free [16]. Configuration comparisons count every judgment at that price, including judgments served from the local cache. This measures the cost of the complete judgment workload at a common price. Local decomposition cost is separate. We report request latency, per-report judgment time, and batch completion time separately.

5 Results

We first test whether report scores reflect expert-assessed factual errors and which medical changes Jev detects. We then examine the complexity and cost needed for these judgments, followed by analyses of score interpretation, calibration, and repeatability.

Table 1: Agreement with expert error counts. Signs are oriented so higher is better; brackets are 95% study-bootstrap intervals. Panel A uses all pairs in each dataset for the first nine rows; local RadFact and RadMatch also cover all 100 RadEvalX pairs. Panel B restricts every method to the same 200 RadEvalExpert pairs. RadFact and RadMatch use local Qwen3.8-27B-FP8 at temperatures 0 and 1, respectively (Section 4.2).

Panel A: complete expert datasets				
Method	RadEvalX ($n = 100$)		RadEvalExpert ($n = 624$)	
	Total	Significant	Total	Significant
Jev–OneQ	0.573 [0.470, 0.661]	0.346 [0.204, 0.475]	0.398 [0.339, 0.455]	0.352 [0.287, 0.410]
Jev–Full	0.582 [0.480, 0.675]	0.343 [0.202, 0.470]	0.397 [0.337, 0.453]	0.350 [0.286, 0.408]
Open NLI	0.337 [0.205, 0.457]	0.162 [-0.005, 0.310]	0.323 [0.260, 0.381]	0.276 [0.207, 0.339]
Finding count	0.288 [0.145, 0.411]	0.123 [-0.032, 0.275]	0.261 [0.189, 0.324]	0.209 [0.137, 0.274]
GREEN	0.448 [0.327, 0.562]	0.347 [0.194, 0.487]	0.180 [0.101, 0.254]	0.193 [0.116, 0.267]
RadCliQ	0.375 [0.244, 0.491]	0.277 [0.125, 0.420]	0.143 [0.076, 0.208]	0.134 [0.070, 0.199]
Temporal F1	0.410 [0.275, 0.531]	0.234 [0.060, 0.386]	0.257 [0.187, 0.325]	0.283 [0.212, 0.351]
RadGraph F1	0.227 [0.103, 0.344]	0.129 [-0.024, 0.263]	0.014 [-0.062, 0.081]	0.026 [-0.053, 0.098]
BERTScore	0.107 [-0.042, 0.257]	0.000 [-0.162, 0.146]	0.034 [-0.043, 0.112]	0.039 [-0.036, 0.115]
RadFact (local 27B)	0.477 [0.355, 0.581]	0.398 [0.265, 0.526]	See Panel B	
RadMatch (local 27B)	0.580 [0.473, 0.679]	0.491 [0.365, 0.613]	See Panel B	

Panel B: common RadEvalExpert subset ($n = 200$)		
Method	Total	Significant
Jev–OneQ	0.322 [0.219, 0.415]	0.259 [0.152, 0.354]
Jev–Full	0.320 [0.216, 0.417]	0.258 [0.153, 0.356]
Open NLI	0.294 [0.198, 0.393]	0.230 [0.126, 0.336]
RadFact (local 27B)	0.151 [0.023, 0.281]	0.186 [0.058, 0.309]
RadMatch (local 27B)	0.453 [0.348, 0.549]	0.485 [0.385, 0.580]

5.1 Report scores track expert-assessed factual differences

The report-level test asks whether greater disagreement with the reference corresponds to more errors identified by radiologists. Jev–OneQ reaches $\tau_b = 0.573$ on RadEvalX and 0.398 on RadEvalExpert (Table 1). Full reaches 0.582 and 0.397, respectively. With the same statements and count aggregation, OneQ improves over open NLI by 0.236 [0.131, 0.345] and 0.075 [0.043, 0.106] on total errors. Its advantage over NLI also holds for significant-error labels. OneQ exceeds RadCliQ on total errors in both datasets. On RadEvalX, its difference from GREEN is 0.125 [−0.006, 0.256], while significant-error correlation is nearly equal (0.346 versus 0.347). On RadEvalExpert, OneQ exceeds both GREEN and RadCliQ on both targets. Appendix C reports the complete standard metric suite.

RadMatch’s advantage depends on the error target. On RadEvalX, its total-error correlation is 0.580, close to OneQ’s 0.573 and Full’s 0.582. The paired OneQ–RadMatch difference is −0.007 [−0.137, 0.114]. For significant errors, RadMatch reaches 0.491 versus OneQ’s 0.346; the difference is −0.145 [−0.294, −0.004]. Full also trails RadMatch on this target. Local RadFact reaches 0.477 on total errors and 0.398 on significant errors; neither paired difference from OneQ excludes zero.

Table 2: Error scope changes the ordering of Jev and open NLI on ReXErr. Each row uses the same 10,790 unchanged negative sentences. Positive counts differ by scope. Differences are Jev minus NLI; intervals use study-block bootstrap.

Positive error scope	n_+	Jev AUROC	NLI AUROC	Difference [95% CI]
All error types	8,723	0.8896	0.8969	-0.0073 [-0.011, -0.003]
Factual error types	7,085	0.9680	0.9614	+0.0066 [+0.004, +0.009]
Language perturbations	1,638	0.5505	0.6179	-0.0674 [-0.083, -0.052]

RadMatch leads on both error targets in the shared RadEvalExpert subset. Across these 200 pairs, its total-error correlation is 0.453, compared with 0.322 for OneQ and 0.320 for Full. OneQ minus RadMatch is -0.131 $[-0.226, -0.044]$ for total errors and -0.226 $[-0.320, -0.137]$ for significant errors. Both intervals exclude zero. On the same subset, OneQ exceeds local RadFact for total errors by 0.171 $[0.052, 0.294]$. Its significant-error difference is 0.074 $[-0.048, 0.195]$.

5.2 Detecting changes in medical assertions

The sentence-level test asks whether Jev recognizes a specific change in medical meaning. Using Full’s raw support outputs in the ReXErr sentence experiment, Jev detects false negation with an AUROC of 0.977 $[0.972, 0.982]$ and recall of 91.9% at threshold 0.5. It also detects changed locations (0.984), changed severity (0.975), and added medical devices (0.989). These are concrete factual differences: turning “a left pleural effusion” into “no left pleural effusion” reverses presence, while changing “left” to “right” changes location. Appendix F reports all error types and a separate language stress test.

The ranking changes when the test includes spelling and repetition (Table 2). Over all injected errors, NLI has higher AUROC than Jev, 0.8969 versus 0.8896. Restricting positives to factual-error types reverses that ordering: Jev reaches 0.9680 and NLI reaches 0.9614, with a paired difference of 0.0066 $[0.004, 0.009]$. For language perturbations, both scores fall, and NLI leads by 0.0674. These 1,638 language perturbations constitute 18.8% of all positive examples.

Jev’s AUROC for added typos is 0.562, and its recall for repetition is 0.003. Low sensitivity to meaning-preserving edits suits factuality evaluation. Some language perturbations can also change medical meaning, so we interpret this grouping by error type. Assessing overall writing quality requires broader error coverage.

5.3 One question retains agreement at lower cost

OneQ retains similar expert agreement with fewer input tokens (Figure 2). Relative to Full, its total-error correlation changes by -0.009 $[-0.026, 0.008]$ on RadEvalX and $+0.001$ $[-0.002, 0.004]$ on RadEvalExpert.

The token savings are consistent across datasets. On RadEvalX, Full uses 9,695.17 input tokens per pair and OneQ uses 5,374.55, a 44.56% reduction. On RadEvalExpert, the corresponding means are 10,797.20 and 6,151.13, a 43.03% reduction. OneQ’s judgment API cost is approximately \$0.000226 and \$0.000258 per pair, excluding local decomposition. In practical terms, the judgment API costs about 2.3–2.6 cents per hundred report pairs. Appendix E gives the calculation and latency measurements.

Other additions bring no consistent improvement. Replacing atomic findings with sentences changes correlation by -0.014 on RadEvalX and -0.015 on RadEvalExpert; both intervals include zero. Removing negation correction or shortening the support criteria also yields small changes. On RadEvalX, filtering retains 32.2% of report sentences and leaves 11.2% of statement inputs with an empty report. Correlation changes by -0.023 $[-0.057, 0.011]$, while total judgment-and-filtering input cost rises by 56.1%. On RadEvalExpert, the correlation change is about -0.009 . This filtering rule offers no observed scoring advantage.

Significance weighting reduces total-error correlation on RadEvalX by 0.129 $[0.050, 0.208]$. Its significant-error correlation rises from 0.343 to 0.396, but the paired interval for that gain includes zero. On RadEvalX, jointly halving candidate and reference noncoverage weights raises correlation by 0.053 $[0.005, 0.104]$. This is an exploratory choice from the nine-setting grid evaluated on the same labels; the main comparisons retain the default weights. On RadEvalExpert, halving candidate noncoverage weight raises correlation by 0.013 $[-0.006, 0.035]$. Appendix D reports the complete grid. The practical outcome is a simple configuration: L1 statements, one support question, and default count aggregation. The sentence-only and OneQ variants were tested separately; their combined effect remains an open comparison.

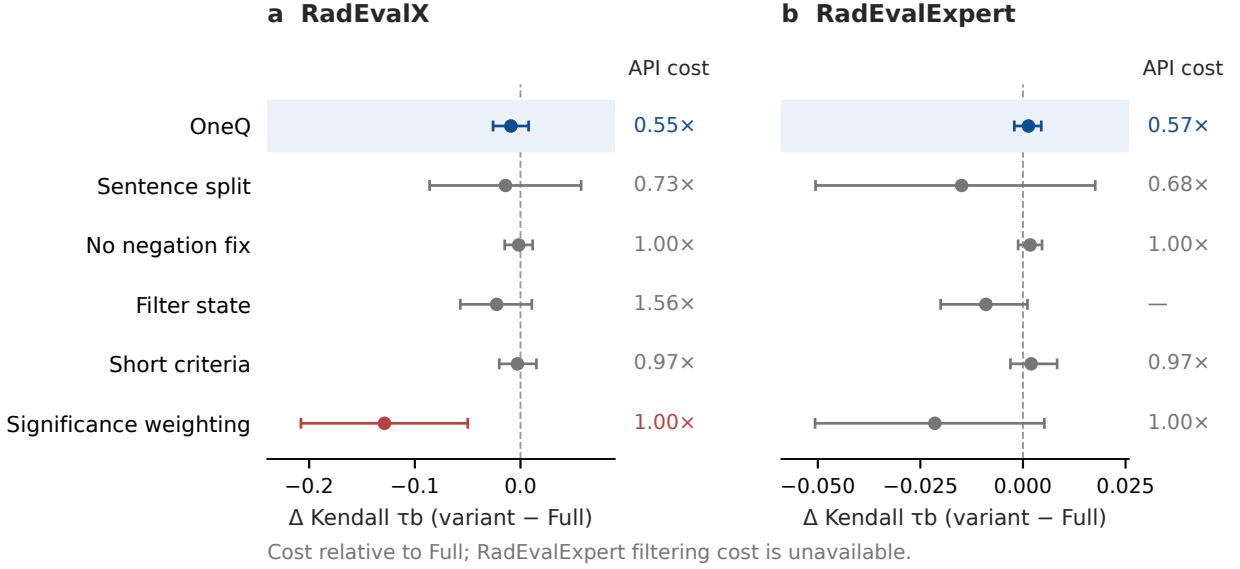


Figure 2: Component changes relative to Full, evaluated independently. Points show changes in total-error τ_b ; bars are paired 95% study-bootstrap intervals. API input cost uses the same price for all judgments, including cached ones. RadEvalX filtering cost includes sentence-selection and judgment calls; RadEvalExpert filtering cost is incomplete and omitted. Removing negation correction or changing aggregation reuses the Full judgments and leaves their API cost unchanged.

5.4 Judgment cost in a concrete example

Figure 3 shows what a judgment call buys: a decision about whether a report supports a finding. On three constructed examples covering contradiction, support, and an unmentioned finding, Jev and GPT-5.6 Luna return the same classifications in all five runs. Jev supplies the three class probabilities directly; Luna generates a JSON label with reasoning disabled. For the three findings together, mean API cost is \$0.00005044 for Jev and \$0.00018520 for Luna. Median completion time is 1.129 s and 3.569 s, respectively, using three concurrent requests per method. Appendix E.1 gives the protocol and pricing calculation.

5.5 Interpreting agreement beyond finding counts

We compare Jev with a finding-count baseline to assess the contribution of its support judgments beyond report size. Counting extracted findings alone reaches $\tau_b = 0.288$ on RadEvalX and 0.261 on RadEvalExpert. This baseline never checks whether a statement is correct; it counts the opportunities for disagreement. OneQ improves over it by 0.285 [0.157, 0.416] and 0.137 [0.080, 0.197], respectively. Jev’s judgments improve agreement beyond the finding-count baseline.

On RadEvalExpert, OneQ also orders candidates describing the same examination better than finding count in point estimate: 0.205 versus 0.101 (Figure 4). This view asks which candidate is better for a particular examination, complementing the pooled ranking of error burden across different examinations.

The common RadEvalExpert subset contains 69 within-study candidate comparisons. Here, both Jev configurations score 0.246 for total errors, compared with 0.043 for RadMatch. The paired difference is 0.203 [−0.011, 0.406]. For significant errors, the difference is 0.043 [−0.169, 0.250]. Both intervals include zero, leaving the within-study difference between Jev and RadMatch uncertain.

Aggregation is another part of the explanation. On this same subset, Full’s count score reaches 0.320, whereas its soft F1 reaches 0.111. The open NLI judge similarly changes from 0.294 with counts to 0.082 with F1. Summation retains report-size information; F1 normalizes support within each report. Both aggregation and judge choice influence agreement with error-count labels. The size of the aggregation effect also depends on the dataset: on RadEvalX, Full reaches 0.582 with counts and 0.561 with soft F1, a much smaller difference whose paired interval includes zero.

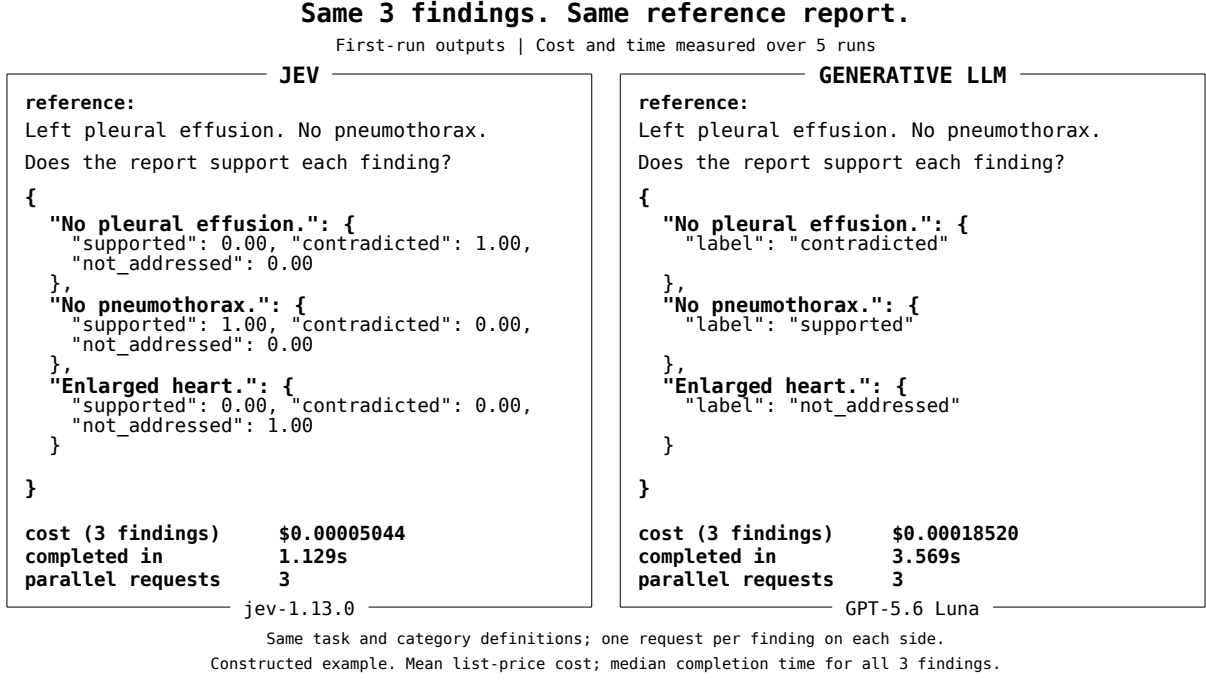


Figure 3: Same findings, two ways to judge. Both methods receive the same reference report, findings, and three-way support definitions. Outputs are from the first run on this constructed example. Cost is the five-run mean of token usage priced at official standard rates; time is the median wall time to finish all three requests.

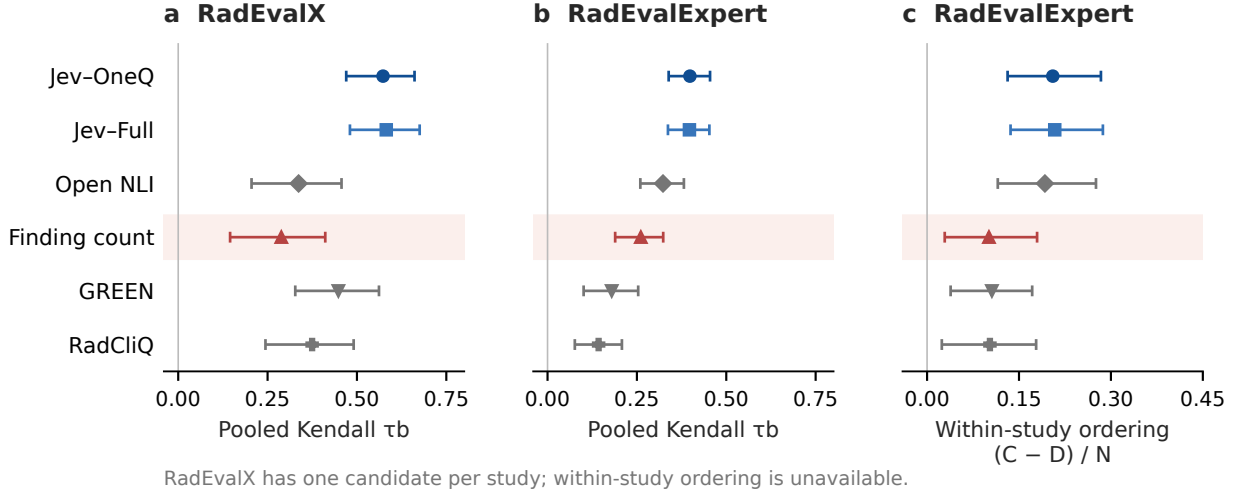


Figure 4: Finding counts and the scope of comparison. Left and center: pooled correlation with expert total-error counts on the two datasets. Right: ordering of RadEvalExpert candidates for the same study, using Eq. 4. All methods use the complete respective dataset. Intervals resample studies. RadEvalX has only one candidate per study and cannot support within-study ranking. The two statistics answer different questions; within-study comparisons fix the reference report while candidate lengths can still vary.

5.6 Calibration makes probabilities useful for thresholding

Jev’s raw sentence-level probabilities have ECE 0.1192 on the full test set, compared with 0.1045 for NLI. On the separate calibration holdout, Jev’s ECE falls from 0.1235 to 0.0077 after isotonic calibration. NLI improves from 0.1072 to 0.0063 under the same procedure. These results support task-specific calibration before using probability thresholds. On RadEvalX, Full’s highest score bin averages 8.53, while the corresponding expert count averages 5.30. Full tracks error burden, but its raw score overestimates expert error counts at the upper end (Appendix G).

Table 3: Repeatability on the same 50 RadEvalExpert pairs, with five runs per setting. Identical is the fraction of pairs with exactly the same score in all five runs. Rank consistency is the median Kendall τ_b across the ten pairs of runs. Correlation SD measures variation in τ_b against expert total-error counts. Jev repeats uncached judgments with L1 decomposition fixed; baseline pipelines are rerun. RadFact uses logical F1; RadMatch uses actionable-error counts.

Evaluator	Setting	Identical	Rank consistency	Correlation SD
Jev-OneQ	Fixed L1	0%	0.990	0.00381
Jev-Full	Fixed L1	0%	0.986	0.00126
GREEN	Greedy	100%	1.000	0.00000
RadFact	Temperature 0	88%	0.928	0.02182
RadMatch	Temperature 0	82%	0.965	0.00975
RadMatch	Temperature 1	10%	0.662	0.03582

5.7 Rankings remain stable across repeated judgments

Jev’s report rankings repeat closely in both configurations. On the same 50 RadEvalExpert pairs used for the baselines, five uncached runs give median between-run rank agreement of 0.990 for OneQ and 0.986 for Full (Table 3). The corresponding standard deviations in expert correlation are 0.00381 and 0.00126. Every report pair has some score variation across the five runs; the resulting rankings remain close. Median score SD is 0.042 for OneQ and 0.047 for Full. On 50 independently selected RadEvalX pairs, five runs each give correlation SDs of 0.00745 for Full and 0.00737 for OneQ. Appendix G.1 details the protocols and score variation.

The baseline runs show how much inference settings can change this behavior (Table 3). With the local backend’s default sampling, RadMatch gives identical scores across five runs for only 10% of report pairs. Setting temperature to zero raises that proportion to 82% and between-run rank consistency from 0.662 to 0.965. The standard deviation of expert correlation falls from 0.03582 to 0.00975. On this subset, mean expert correlation also rises from 0.400 to 0.468.

GREEN gives exactly the same scores in all five runs. RadFact’s default temperature-zero setup gives identical F1 scores for 88% of pairs. Repeatability varies across these pipelines and their decoding settings.

6 Discussion

Jev combines support judgments in both directions to score unsupported candidate statements and missing reference findings. The resulting report scores track expert-assessed error burden. This gives researchers a low-cost way to score collections of generated reports, with individual support judgments available for inspection.

OneQ retains expert agreement close to Full while reducing judgment API input cost by 43–45%. Additional questions and filtering offer no consistent gain, and the matched NLI comparison shows that the choice of judge matters within this simple configuration. The result is a practical starting point for report-generator evaluation: extract statements, ask one support question, and aggregate discrepancies in both directions.

The choice also depends on the evaluation target. On RadEvalX, Jev and local RadMatch have similar total-error correlations, while RadMatch agrees more closely with clinically significant errors. On the shared RadEvalExpert subset, RadMatch leads on both targets. Researchers prioritizing significant errors therefore have a reason to use the more elaborate evaluator. Jev’s contribution is an inexpensive judgment component that detects medically meaningful discrepancies through a simple support question.

Discrepancy counts retain information about report size; normalized scores describe the proportion of content in agreement. An evaluator that tolerates a harmless spelling change may suit factuality assessment while missing an error in a writing-quality test. Finding-count baselines, comparisons on shared samples, and breakdowns by error scope help connect a metric’s benchmark score to its intended use. Where multiple candidates describe the same examination, within-study ordering adds a direct view of choosing between those candidates.

Jev’s aggregate rankings repeat closely across runs, while GREEN returns identical scores. RadMatch’s rank consistency rises from 0.662 to 0.965 when sampling is disabled on the same inputs. Decoding settings affect repeatability, while calibration improves probability estimates for selecting statements for further inspection.

6.1 Limitations and future work

Our evaluation measures agreement with reference text. Reference reports can contain errors or omit relevant findings; establishing correctness against the image and validating clinical use require separate studies. Report-level category

scores do not yet reliably distinguish location, severity, and false-finding errors. Probability thresholds benefit from calibration on the intended data, and language-quality assessment requires coverage beyond factual support.

The study covers one Jev version, small expert benchmarks, and controlled injected errors. RadEvalX uses one generator and RadCliQ-stratified sampling, with 80 abnormal and 20 normal reports. Broader generator and case distributions would test generality. Sentence-level detection and calibration use Full’s support outputs; direct evaluation of OneQ’s local probabilities is a further test. Repeating decomposition would extend Jev’s fixed-decomposition repeatability study to the full pipeline. Finally, matched end-to-end cost and timing measurements would complement the measured judgment costs and three-finding API demonstration when comparing deployment options.

7 Conclusion

Evaluating a generated radiology report requires identifying what it changes, adds, and leaves out relative to the reference. Jev scores these discrepancies through simple statement-level support judgments combined in both directions. Its single-question configuration achieves competitive agreement with expert error counts at a judgment API cost below three cents per hundred report pairs, excluding local decomposition. Controlled-error tests demonstrate sensitivity to several medically meaningful changes, and repeated calls produce closely agreeing report rankings. One support question per statement yields expert agreement close to the seven-question design at 43–45% lower judgment cost. This simple Jev configuration offers a low-cost way to evaluate factual differences in generated reports.

Data and Code Availability

RadEvalX and ReXErr-v1 (both version 1.0.0) are available through PhysioNet [5, 15]; RadEvalExpert and RaTE-Eval are available from their respective dataset distributions [18, 23], subject to each dataset’s terms of use. Exact questions and implementation details appear in Appendix B. The analysis scripts and aggregate results are not yet publicly available; we are preparing them for release.

References

- [1] Mohammed Baharoon, Thibault Heintz, Siavash Raissi, Mahmoud Alabbad, Mona Alhammad, Hassan AlOmaish, Sung Eun Kim, Oishi Banerjee, and Pranav Rajpurkar. CRIMSON: A Clinically-Grounded LLM-Based Metric for Generative Radiology Report Evaluation. *arXiv preprint arXiv:2603.06183*, 2026. URL <https://arxiv.org/abs/2603.06183>.
- [2] Shruthi Bannur, Kenza Bouzid, Daniel C. Castro, Anton Schwaighofer, Anja Thieme, Sam Bond-Taylor, Maximilian Ilse, Fernando Perez-Garcia, Valentina Salvatelli, Harshita Sharma, Felix Meissen, Mercy Ranjit, Shaury Srivastav, Julia Gong, Noel C. F. Codella, Fabian Falck, Ozan Oktay, Matthew P. Lungren, Maria Teodora Wetscherek, Javier Alvarez-Valle, and Stephanie L. Hyland. MAIRA-2: Grounded Radiology Report Generation. *arXiv preprint arXiv:2406.04449*, 2024. URL <https://arxiv.org/abs/2406.04449>.
- [3] Federica Bologna, Jean-Philippe Corbeil, Matthew Wilkens, and Asma Ben Abacha. VERT: Reliable LLM Judges for Radiology Report Evaluation. *arXiv preprint arXiv:2604.03376*, 2026. URL <https://arxiv.org/abs/2604.03376>.
- [4] Amos Calamida, Farhad Nooralahzadeh, Morteza Rohanian, Koji Fujimoto, Mizuho Nishio, and Michael Krauthammer. Radiology-Aware Model-Based Evaluation Metric for Report Generation. *arXiv preprint arXiv:2311.16764*, 2023. URL <https://arxiv.org/abs/2311.16764>.
- [5] Amos Rubin Calamida, Farhad Nooralahzadeh, Morteza Rohanian, Mizuho Nishio, Koji Fujimoto, and Michael Krauthammer. Radiology Report Generation Models Evaluation Dataset For Chest X-rays (RadEvalX), version 1.0.0, 2024. URL <https://physionet.org/content/rad-eval-x/1.0.0/>.
- [6] Charles Corbiere, Leo Machado, Aubin Charley, Baptiste Callard, Pierre Manceron, and Corentin Dancette. RadMatch: Auditable Radiology Report Evaluation via Finding-Level Matching. *arXiv preprint arXiv:2609.01470*, 2026. URL <https://arxiv.org/abs/2609.01470>.
- [7] Dina Demner-Fushman, Marc D. Kohli, Marc B. Rosenman, Sonya E. Shooshan, Laritza Rodriguez, Sameer Antani, George R. Thoma, and Clement J. McDonald. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 23(2):304–310, 2016. doi: 10.1093/jamia/ocv080.
- [8] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators. *arXiv preprint arXiv:2404.04475*, 2024. URL <https://arxiv.org/abs/2404.04475>.
- [9] Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. SummEval: Re-evaluating Summarization Evaluation. *arXiv preprint arXiv:2007.12626*, 2020. URL <https://arxiv.org/abs/2007.12626>.
- [10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. *arXiv preprint arXiv:1706.04599*, 2017. URL <https://arxiv.org/abs/1706.04599>.
- [11] Saahil Jain, Ashwin Agrawal, Adriel Saporta, Steven QH Truong, Du Nguyen Duong, Tan Bui, Pierre Chambon, Yuhao Zhang, Matthew P. Lungren, Andrew Y. Ng, Curtis P. Langlotz, and Pranav Rajpurkar. RadGraph: Extracting Clinical Entities and Relations from Radiology Reports. *arXiv preprint arXiv:2106.14463*, 2021. URL <https://arxiv.org/abs/2106.14463>.

- [12] Moritz Laurer. DeBERTa-v3-large-mnli-fever-anli-ling-wanli: Model card, n.d. URL <https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli>. Checkpoint documentation; accessed September 22, 2026.
- [13] OpenAI. API Pricing, 2026. URL <https://developers.openai.com/api/docs/pricing>. Accessed September 22, 2026. GPT-5.6 Luna standard short-context pricing.
- [14] Sophie Ostmeier, Justin Xu, Zhihong Chen, Maya Varma, Louis Blankemeier, Christian Bluethgen, Arne Edward Michalson, Michael Moseley, Curtis Langlotz, Akshay S Chaudhari, and Jean-Benoit Delbrouck. GREEN: Generative Radiology Report Evaluation and Error Notation. *arXiv preprint arXiv:2405.03595*, 2024. doi: 10.18653/v1/2024.findings-emnlp.21. URL <https://arxiv.org/abs/2405.03595>.
- [15] Vishwanatha Rao, Serena Zhang, Julian Acosta, Subathra Adithan, and Pranav Rajpurkar. ReXErr-v1: Clinically Meaningful Chest X-Ray Report Errors Derived from MIMIC-CXR. PhysioNet, version 1.0.0, 2025. URL <https://physionet.org/content/rexerr-v1/1.0.0/>.
- [16] TypeSafe AI. Jev 1.13: Models and System One documentation, 2026. URL <https://docs.typesafe.ai/models>. Model jev-1.13.0; accessed September 22, 2026.
- [17] Yuan Wang, Wanxing Chang, Songtao Jiang, Shujian Gao, Xiaotian Zhang, Ruifeng Yuan, Weiwei Cao, Bowen Shi, Ling Zhang, Zuozhu Liu, and Jianpeng Zhang. AtomiMed: Hierarchical Atomic Fact-Checking for Universal Clinical-Aware Medical Report Evaluation. *arXiv preprint arXiv:2606.31292*, 2026. URL <https://arxiv.org/abs/2606.31292>.
- [18] Justin Xu, Xi Zhang, Javid Abderezaei, Julie Bauml, Roger Boodoo, Fatemeh Haghighi, Ali Ganjizadeh, Eric Brattain, Dave Van Veen, Zaiqiao Meng, David Eyre, and Jean-Benoit Delbrouck. RadEval: A framework for radiology text evaluation. *arXiv preprint arXiv:2509.18030*, 2025. URL <https://arxiv.org/abs/2509.18030>.
- [19] Zhenhong Yang, Zhuoyun Liu, Jintao Fei, Wen Tang, Shichao Quan, Jun Zhao, and Jun Xu. RadSEM: A Finding-by-Finding Metric for Clinical Consistency in Radiology Reports. *arXiv preprint arXiv:2606.17062*, 2026. URL <https://arxiv.org/abs/2606.17062>.
- [20] Feiyang Yu, Mark Endo, Rayan Krishnan, Ian Pan, Andy Tsai, Eduardo Pontes Reis, et al. Evaluating Progress in Automatic Chest X-Ray Radiology Report Generation. *medRxiv*, 2022. doi: 10.1101/2022.08.30.22279318. Reference linked by the ReXVal dataset documentation.
- [21] Feiyang Yu, Mark Endo, Rayan Krishnan, Ian Pan, Andy Tsai, Eduardo Pontes Reis, Eduardo Kaiser Ururahy Nunes Fonseca, Henrique Lee, Zahra Shakeri, Andrew Ng, Curtis Langlotz, Vasantha Kumar Venugopal, and Pranav Rajpurkar. Radiology Report Expert Evaluation (ReXVal) Dataset. PhysioNet, version 1.0.0, 2023. URL <https://physionet.org/content/rexval-dataset/1.0.0/>.
- [22] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. *arXiv preprint arXiv:1904.09675*, 2019. URL <https://arxiv.org/abs/1904.09675>.
- [23] Weike Zhao, Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. RaTEScore: A Metric for Radiology Report Generation. *arXiv preprint arXiv:2406.16845*, 2024. URL <https://arxiv.org/abs/2406.16845>.
- [24] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*, 2023. URL <https://arxiv.org/abs/2306.05685>.

A Data Processing and Supplementary Tasks

RadEvalX uses the consensus of two radiologists over eight error categories. We sum significant and insignificant counts for the total-error target. Blank category cells encode zero: this convention exactly reproduces all 16 category totals for the 30 high-error reports in the original paper’s Appendix D [4]. The 100 annotated report IDs join one-to-one to the released 590-row metric file; the remaining rows have no expert labels and are excluded. Our main baseline scores are recomputed with the same implementations used for RadEvalExpert.

There are 100 distinct report IDs but 96 distinct reference texts and 78 candidate texts. We retain report IDs as the sampling unit; identical text alone does not identify a shared study. Each ID has only one candidate, so within-study candidate ranking is unavailable on RadEvalX. Table A1 removes the two uncertainty categories from the labels while keeping predictions fixed.

Table A1: RadEvalX label-scope sensitivity. The primary labels count all eight error categories; the six-category target excludes added and omitted uncertainty. Predictions stay fixed. Values are directed Kendall τ_b .

Method	Total (8)	Total (6)	Significant (8)	Significant (6)
Jev-Full	0.582	0.577	0.343	0.339
Jev-OneQ	0.573	0.568	0.346	0.343
Jev-Full soft F1	0.561	0.551	0.429	0.424
Open NLI	0.337	0.335	0.162	0.162
Finding count	0.288	0.295	0.123	0.126
GREEN	0.448	0.442	0.347	0.346
RadMatch (local 27B)	0.580	0.578	0.491	0.490

RadEvalExpert preserves its total and significant-error labels. The study identifier is the resampling unit, keeping candidates with their shared reference together. ReXErr sentence inputs contain one candidate sentence and its complete original report. We exclude the dataset’s neutral rewrites from the main binary-error analysis. Unchanged sentences form the negative class in every error-scope comparison.

The factual-error group contains false negation, false prediction, changed location, changed severity, changed measurement, added medical device, changed device position, and changed device name. The language group contains added typos, homophone substitutions, and repetition. This is an exploratory grouping by the dataset’s intervention labels. We have not manually adjudicated the medical meaning of every edit. The two groups share unchanged negatives but use disjoint positive examples.

RaTE-Eval contributes 440 sentence pairs, 370 paragraph pairs, and 847 synthetic synonym-opposite comparisons. Each synthetic comparison produces two reference-candidate pairs, giving 1,694 pairs in total. The distributed synthetic files contain rewrites and opposites without their original sentences. We reconstruct references from the sentence-task pool using text similarity and one-to-one assignment. Full gives the synonym a lower error score in 97.3% of these comparisons and a higher soft F1 in 99.9%. Excluding the seven comparisons below the preprocessing confidence threshold of 0.20 leaves 840 comparisons. The corresponding rates are 97.5% and 100.0%. These results describe the reconstructed task; the expert benchmarks and ReXErr analyses use their supplied references.

Table A2 reports the other RaTE-Eval tasks. The normalized Jev count correlates with sentence error ratios at 0.398 and paragraph quality ratings at 0.424. RadCliQ reaches 0.487 on paragraph ratings, showing that broader report quality remains a distinct evaluation target.

Table A2: Supplementary RaTE-Eval correlations. Sentence labels are error ratios (lower is better); paragraph labels are quality ratings (higher is better). Signs follow these targets. Brackets are study-bootstrap 95% intervals.

Method	Sentence ($n = 440$)	Paragraph ($n = 370$)
Jev-Full normalized count	0.398 [0.332, 0.461]	0.424 [0.353, 0.491]
Jev-Full soft F1	0.398 [0.331, 0.461]	0.369 [0.288, 0.437]
Open NLI normalized count	0.218 [0.154, 0.289]	0.328 [0.251, 0.401]
BERTScore	0.286 [0.218, 0.348]	0.474 [0.411, 0.540]
RadCliQ	0.324 [0.260, 0.382]	0.487 [0.425, 0.553]

B Implementation and Exact Questions

Report segmentation uses spaCy with repairs for abbreviations, measurements, and numbered lists. Template headings and empty units are removed. L1 passes each remaining sentence to the atomic-decomposition prompt below. The local decomposition model is Qwen3.5-9B, with temperature zero. Its invalid JSON outputs fall back to the input sentence.

Jev receives the claim and opposing report as structured state fields. Each statement–report comparison uses one request: Full includes all seven questions, while OneQ includes only the support question. Both configurations check statements in both directions. Repeated-call experiments obtain fresh judgments for the same inputs.

The default weights in Eq. 2 are one. The weight grid varies candidate-side not-addressed and reference-side not-addressed terms over $\{0.5, 1, 2\}^2$, leaving candidate contradiction weight fixed at one. Significance weighting multiplies each discrepancy term by the associated expected significance score divided by two. Category scores distribute candidate discrepancy mass according to the error-type probabilities; reference not-addressed mass supplies the omission category. These diagnostics use Full’s additional outputs.

B.1 Support question and Full extensions

OneQ sends the support entry below. Full sends all seven entries in one request. The state contains two text fields: `claim` and `report`.

support (choice). Does the report support the claim about the imaging finding?

supported The report states the same finding with compatible presence, location, and severity

contradicted The report states the opposite, or the same finding with incompatible presence, laterality, or severity

not_addressed The report does not mention this finding at all

claim_negated (noun). The claim states that the finding is ABSENT (for example ‘no’, ‘without’, ‘clear of’, ‘free of’)

claim_hedged (noun). The claim expresses uncertainty about the finding (for example ‘possible’, ‘cannot exclude’, ‘may represent’, ‘questionable’)

report_negated (noun). The report states that the same finding described in the claim is ABSENT

report_hedged (noun). The report expresses uncertainty about the same finding described in the claim

error_type (choice). If the claim disagrees with the report, what kind of discrepancy is it?

false_finding The claim asserts a finding that the report says is absent or never mentions

wrong_location Same finding, but a different anatomical location or laterality

wrong_severity Same finding, but a different severity, size, or extent

wrong_temporal Same finding, but a different comparison to prior imaging (new, unchanged, improved, worsened)

none The claim does not disagree with the report

significance (score). Clinical significance of the finding described in the claim, if the claim were wrong

1. Incidental or normal-variant statement with no effect on management
2. Would change follow-up imaging or outpatient management but not immediate care
3. Would change immediate management or is potentially life-threatening

B.2 Atomic decomposition prompt

You split one sentence from a radiology report into atomic findings.

Rules:

1. Each output item states exactly one finding about one anatomical entity.
2. Preserve negation and uncertainty words exactly as written ("no", "without", "cannot exclude", "possible", "unchanged").
3. Preserve laterality, severity, size, and comparison-to-prior phrases inside the item they belong to.
4. Do not add, infer, or normalize anything. Do not change tense.
5. If the sentence already contains one finding, return it unchanged.
6. Output a JSON list of strings and nothing else.

B.3 Filtering and shortened criteria

The filtering variant asks, “This sentence is about the same anatomical entity as the claim,” for every report sentence. It retains sentences whose probability exceeds 0.3 and passes their concatenation to Full. The shortened support criteria are “The report states the same finding,” “The report states the opposite,” and “The report does not mention it.” They replace the three support criteria in order; the other Full questions stay the same.

Table A3: All 17 standard metric outputs on the complete expert datasets. Columns report directed Kendall correlation with total (T) or significant (S) errors. Variants within each method family are listed separately.

Metric output	RadEvalX		RadEvalExpert	
	Total	Significant	Total	Significant
bleu	0.142	0.113	-0.021	-0.027
rouge1	0.232	0.172	0.006	0.005
rouge2	0.177	0.104	-0.013	-0.017
rougeL	0.262	0.173	0.040	0.032
bertscore	0.107	0.000	0.034	0.039
f1chexbert_sample_acc_5	0.177	0.277	0.172	0.143
f1chexbert_sample_acc_all	-0.031	-0.159	-0.167	-0.168
radgraph_simple	0.234	0.086	0.001	0.009
radgraph_partial	0.225	0.089	0.002	0.010
radgraph_complete	0.227	0.129	0.014	0.026
ratescore	0.290	0.177	0.073	0.068
srrbert_weighted_f1	0.332	0.261	0.139	0.150
srrbert_weighted_precision	0.332	0.261	0.137	0.148
srrbert_weighted_recall	0.332	0.261	0.137	0.148
temporal_f1	0.410	0.234	0.257	0.283
radcliq_v1	0.375	0.277	0.143	0.134
green	0.448	0.347	0.180	0.193

C Additional Expert Comparisons

Table A3 reports 17 standard metric outputs, including variants from the same method family. Similarity metrics are negated when correlated with error counts. A negative directed value is retained and indicates an association opposite to the expected direction. All subset correlations and paired differences are recomputed on the shared valid pairs before study resampling.

For within-study ordering, every unordered pair of candidates contributes +1, −1, or zero according to whether the method and expert orderings agree, disagree, or contain a tie. The denominator includes all candidate pairs. Full RadEvalExpert contributes 624 comparisons. The common 200-pair RadFact/RadMatch subset spans 138 studies and contributes 69 comparisons. Each bootstrap draw sums the pair counts of the sampled study blocks, with repeated studies contributing repeated counts. Within-study comparisons share a reference report; candidate lengths can still differ.

C.1 Source, section, and modality

Table A4 separates RadEvalExpert by data source and report section. Jev-Full has positive total-error correlation in each source, with wider uncertainty for the smallest source. Table A5 reports the RaTE-Eval modalities with at least 20 evaluated pairs. Performance varies with both modality and evaluation target; paragraph quality ratings often favor BERTScore or RadCliQ. These strata show how performance varies within the evaluated datasets.

Table A4: RadEvalExpert total-error correlation by source and report section. Each row uses the same pairs for the three methods. Source and section partitions overlap. Intervals resample studies within each stratum.

Stratum	n	Jev-Full	Finding count	GREEN
CheXpert Plus	96	0.323 [0.143, 0.483]	0.191 [-0.017, 0.351]	0.171 [-0.017, 0.365]
MIMIC-CXR	306	0.410 [0.335, 0.495]	0.221 [0.137, 0.338]	0.182 [0.079, 0.289]
ReXGradient	222	0.396 [0.300, 0.482]	0.341 [0.250, 0.440]	0.257 [0.137, 0.369]
Findings section	444	0.426 [0.353, 0.483]	0.242 [0.165, 0.318]	0.259 [0.159, 0.342]
Impression section	180	0.261 [0.126, 0.361]	0.185 [0.053, 0.314]	0.068 [-0.086, 0.179]

D Complete Component Comparisons

Table A6 gives the numerical comparisons shown in Figure 2. Table A7 covers all nine weight combinations. The default configuration is retained throughout the main comparisons. RadEvalX peaks at $w_h = w_o = 0.5$, with $\tau_b = 0.635$ and paired gain 0.053 [0.005, 0.104]. RadEvalExpert peaks at $w_h = 0.5$, $w_o = 1$, with $\tau_b = 0.410$. These exploratory settings were compared on the evaluation data, so the largest observed value is not an independently validated tuning gain.

Table A5: RaTE-Eval correlation by modality, for strata with at least 20 pairs. Jev uses Full normalized error count. Similarity/error signs follow each task's label direction. Brackets give 95% study-bootstrap intervals.

Task / modality	n	Jev-Full	BERTScore	RadCliQ
Sentence / CT	89	0.386 [0.237, 0.526]	0.235 [0.078, 0.381]	0.312 [0.162, 0.459]
Sentence / CTA	51	0.396 [0.219, 0.576]	0.381 [0.209, 0.543]	0.389 [0.227, 0.561]
Sentence / MRA	32	0.385 [0.111, 0.632]	0.241 [-0.057, 0.506]	0.348 [0.109, 0.560]
Sentence / MRI	88	0.390 [0.241, 0.525]	0.325 [0.183, 0.455]	0.277 [0.118, 0.412]
Sentence / MRV	26	0.502 [0.131, 0.758]	0.277 [-0.155, 0.545]	0.284 [-0.120, 0.556]
Sentence / US	40	0.397 [0.172, 0.619]	0.197 [-0.025, 0.416]	0.314 [0.128, 0.526]
Sentence / XR	91	0.341 [0.198, 0.473]	0.245 [0.083, 0.384]	0.289 [0.140, 0.418]
Paragraph / CT	62	0.465 [0.284, 0.625]	0.493 [0.351, 0.637]	0.462 [0.304, 0.624]
Paragraph / CTA	64	0.359 [0.187, 0.526]	0.457 [0.273, 0.613]	0.479 [0.315, 0.612]
Paragraph / MRA	38	0.363 [0.137, 0.539]	0.390 [0.158, 0.608]	0.433 [0.155, 0.645]
Paragraph / MRI	84	0.383 [0.238, 0.509]	0.422 [0.267, 0.563]	0.408 [0.243, 0.559]
Paragraph / US	76	0.429 [0.259, 0.580]	0.573 [0.458, 0.683]	0.557 [0.425, 0.680]
Paragraph / XR	36	0.548 [0.325, 0.735]	0.521 [0.273, 0.721]	0.595 [0.379, 0.755]

Table A6: Paired changes in total-error τ_b relative to Full. These are independent variants, not successive simplifications. Cost ratios appear in Figure 2.

Variant	RadEvalX	RadEvalExpert
Sentence split	-0.014 [-0.086, +0.057]	-0.015 [-0.051, +0.018]
OneQ	-0.009 [-0.026, +0.008]	+0.001 [-0.002, +0.004]
Filter state	-0.023 [-0.057, +0.011]	-0.009 [-0.020, +0.001]
Short criteria	-0.003 [-0.020, +0.015]	+0.002 [-0.003, +0.008]
No negation fix	-0.002 [-0.015, +0.012]	+0.002 [-0.001, +0.005]
Significance weighting	-0.129 [-0.208, -0.050]	-0.021 [-0.051, +0.005]

E Cost and Latency

Let $T_{v,i}$ denote the returned input-token count for configuration v on report pair i , summed over both directions. With N pairs, workload cost is

$$\overline{\text{cost}}_v = \frac{1}{N} \sum_i T_{v,i} \frac{0.042}{10^6}, \quad \text{saving} = 1 - \frac{\sum_i T_{\text{OneQ},i}}{\sum_i T_{\text{Full},i}}. \quad (5)$$

On RadEvalX, $1 - 5374.55/9695.17 = 44.56\%$. On RadEvalExpert, $1 - 6151.1346/10797.1971 = 43.03\%$. The resulting mean API costs are shown in Table A8. Full and OneQ use the same L1 decomposition, so its local cost is shared.

Full on RadEvalExpert has median request latency 433.6 ms and 95th-percentile latency 548.4 ms. Its median pair-level judgment time is 0.71 s with concurrent statement requests. Batch completion took 10.2 minutes, including scheduling and rate limits. Request-latency measurements exclude local cache hits, whereas the workload-cost comparison above includes them. No matched cross-method timing of the full report-evaluation pipeline is included.

Table A7: Exploratory weight grid for Full. Weights multiply candidate noncoverage (w_h) and reference noncoverage (w_o); contradiction weight stays one. Columns give total-error τ_b and paired change from (1, 1). Intervals are unadjusted 95% study-bootstrap intervals.

w_h	w_o	RadEvalX τ_b	Change [95% CI]	RadEvalExpert τ_b	Change [95% CI]
0.5	0.5	0.635	+0.053 [+0.005, +0.104]	0.407	+0.010 [-0.006, +0.025]
0.5	1	0.597	+0.015 [-0.026, +0.056]	0.410	+0.013 [-0.006, +0.035]
0.5	2	0.496	-0.086 [-0.159, -0.020]	0.381	-0.016 [-0.049, +0.017]
1	0.5	0.560	-0.022 [-0.077, +0.032]	0.371	-0.026 [-0.047, -0.007]
1	1	0.582	+0.000 [+0.000, +0.000]	0.397	+0.000 [+0.000, +0.000]
1	2	0.519	-0.062 [-0.112, -0.018]	0.393	-0.004 [-0.025, +0.019]
2	0.5	0.450	-0.132 [-0.205, -0.059]	0.318	-0.078 [-0.116, -0.044]
2	1	0.492	-0.090 [-0.136, -0.045]	0.351	-0.046 [-0.069, -0.025]
2	2	0.514	-0.068 [-0.101, -0.039]	0.381	-0.016 [-0.026, -0.006]

Table A8: API workload cost at \$0.042 per million input tokens. Cached judgments are priced at the same rate as fresh judgments. Local GPU costs are excluded.

Dataset	Configuration	Input tokens/pair	USD/pair
RadEvalX	Full	9,695.17	0.00040720
	OneQ	5,374.55	0.00022573
RadEvalExpert	Full	10,797.20	0.00045348
	OneQ	6,151.13	0.00025835

Table A9: Jev detection by injected error type. Each row includes the same 10,790 unchanged negative sentences. This sentence experiment uses the Full question set and raw support probabilities.

Injected error	n_+	AUROC	95% CI	Recall at 0.5
Add medical device	910	0.9888	[0.984, 0.993]	0.9451
Change position of device	438	0.9869	[0.979, 0.992]	0.9041
Change location	950	0.9841	[0.978, 0.989]	0.8947
False negation	1398	0.9773	[0.972, 0.982]	0.9185
Change severity	849	0.9753	[0.968, 0.981]	0.8492
Change measurement	67	0.9707	[0.937, 0.996]	0.8955
False prediction	2059	0.9471	[0.939, 0.955]	0.8436
Change name of device	414	0.9226	[0.905, 0.941]	0.6691
Change to homophone	127	0.6477	[0.601, 0.695]	0.0945
Add typo	1131	0.5617	[0.549, 0.576]	0.0292
Add repetition	380	0.4848	[0.469, 0.500]	0.0026

E.1 Three-finding API demonstration

Figure 3 uses three constructed claims fixed before execution. Each method receives the same complete reference and the support question and category definitions in Appendix B.1. We send one request per finding, with concurrency three, for five rounds in alternating method order. There is no local response cache or warmup. We show first-round outputs and summarize all five rounds; all 30 classifications match the expected labels. Timing covers the judgment requests and excludes decomposition.

Jev uses version `jev-1.13.0` with `OneQ`. Luna responses report `gpt-5.6-luna`, standard service, and zero reasoning tokens. Its requests set reasoning effort to none, temperature to zero, JSON output, and a 64-token output limit. Completion time includes communication and service overhead. The five-run ranges are 0.607–1.212 s for Jev and 2.869–4.301 s for Luna.

At the September 22, 2026 standard API prices [13, 16], Jev’s 1,201 input tokens per round cost $1201 \times 0.042/10^6 = \0.000050442 . Luna uses 686 input tokens and 40 output tokens per round, with no cached input or cache-write tokens. Its cost is $(686 \times 0.20 + 40 \times 1.20)/10^6 = \0.00018520 .

F Error Types and Language Stress Tests

Table A9 gives the complete sentence-level error profile. The 131-item constructed stress set probes nine forms of negation, uncertainty, and clinical attributes. Jev assigns the expected three-way support label in 125 cases (95.4%) in the original run. After error review, we corrected three device-status labels and simplified one compound claim, obtaining 129 correct cases (98.5%). The corrections followed the support criteria: an unmentioned device is unaddressed, and each claim should state one finding. Both results are retained because these revisions followed inspection of model outputs. On the revised set, Jev answers all 15 laterality cases, 22 direct-negation cases, and 16 uncertainty cases correctly. These are small constructed sets, rather than estimates of clinical sensitivity for naturally occurring errors.

Two cases fail. First, a double-negation statement, “There is no evidence to suggest the absence of a left pleural effusion,” is labeled contradicted against a report stating a left effusion. Second, a claim about a femur fracture is labeled contradicted rather than unaddressed against a chest report containing “No acute osseous abnormality.” The latter illustrates how broad negative language can blur contradiction and noncoverage.

Report-level category attribution is weaker than binary error detection. Figure A1 shows each predicted category’s average share of total category score, grouped by injected error class. For location errors, the false-finding share is 0.468 and the location share is 0.339. For severity errors, the corresponding shares are 0.493 and 0.301. The broad false-finding category dominates even when a more specific category is expected. The matrix summarizes score allocation, not a finding-matched confusion matrix. These results do not support treating the category scores as reliable report-level error explanations.

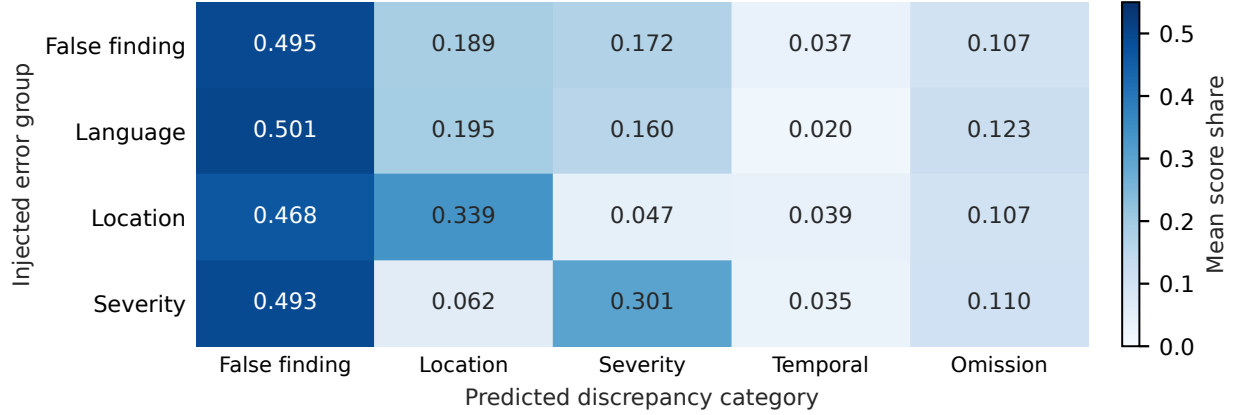


Figure A1: Report-level category allocation on 2,000 sampled ReXErr pairs. Rows group the injected errors; cells show mean proportions of predicted category score. The omission component uses the reverse direction. This diagnostic measures score allocation, not classification accuracy.

G Calibration and Repeated Calls

The raw ReXErr sentence experiment has 19,513 examples and an error prevalence of 44.7%. Jev’s mean error probability is 0.3278; NLI’s is 0.3755. Their Brier scores are 0.1235 and 0.1192, respectively. Figure A2 shows the raw reliability curves. Bin populations are uneven: Jev assigns 12,377 examples to the lowest bin and 5,761 to the highest. The middle bins contain 113–301 examples each. The observed error fraction in Jev’s lowest bin is 0.1672 despite a mean predicted probability of 0.0037.

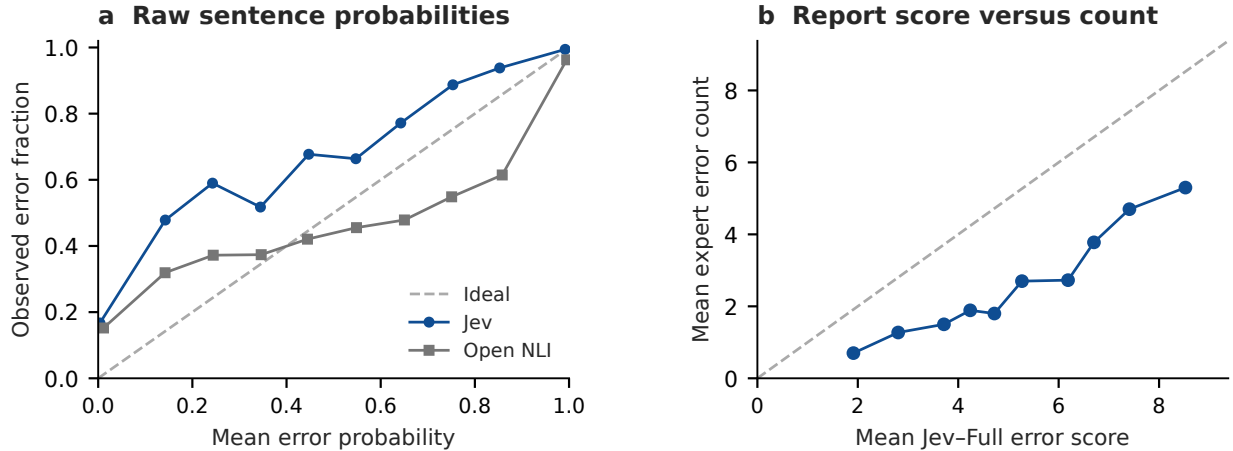


Figure A2: Calibration diagnostics. (a) Raw sentence error probabilities on all 19,513 ReXErr examples, using ten equal-width probability bins. (b) Jev-Full score bins versus mean expert errors on RadEvalX, using ten score quantiles. Dashed lines indicate perfect calibration or equality with expert counts. Neither plot uses the post-hoc calibrated holdout probabilities.

Post-hoc calibration uses a fixed, equal split of studies into development and evaluation sets. This gives 9,562 development examples and 9,951 holdout examples, without study overlap. Logistic calibration fits a sigmoid to raw error probabilities; isotonic regression fits a monotone map with out-of-range values clipped. Table A10 compares raw and calibrated outputs on the same holdout. AUROC remains nearly unchanged in this experiment; isotonic maps can introduce ties. These results concern the Full sentence configuration and this error mixture.

G.1 Repeatability

The RadEvalX experiment randomly selects 50 report pairs and holds 615 extracted findings fixed. Five uncached runs each of Full and OneQ produce 6,150 judgments, all using `jev=1.13.0`. The sample standard deviations in correlation with expert total-error counts are 0.00745 and 0.00737, respectively. Table A11 summarizes ranking and score variation on this independent dataset.

Table A10: Probability calibration on the same 9,951-example holdout. Fits use 9,562 examples from disjoint studies. Full-set raw ECE is reported separately in the main text.

Judge	Calibration	ECE	Brier	AUROC
Jev	Raw	0.1235	0.1269	0.8858
	Logistic	0.0129	0.1061	0.8858
	Isotonic	0.0077	0.1034	0.8858
Open NLI	Raw	0.1072	0.1224	0.8938
	Logistic	0.0185	0.1115	0.8938
	Isotonic	0.0063	0.1064	0.8937

Table A11: Repeated Jev judgments on 50 RadEvalX pairs, five uncached runs per configuration, holding 615 extracted findings fixed. Rank agreement is the median of ten between-run Kendall correlations; correlation SD is across the five correlations with total-error labels.

Configuration	Rank agreement	Correlation SD	Median score SD
Jev-Full	0.965	0.00745	0.054
Jev-OneQ	0.974	0.00737	0.042

The matched repeatability experiment evaluates 50 RadEvalExpert pairs from 48 studies, with five runs per setting. These are a fixed selection of 50 pairs from the 200-pair subset. GREEN uses greedy decoding; RadFact uses its default temperature zero. RadMatch uses the same local Qwen3.8-27B-FP8 backend at temperature zero and at its default temperature one.

Jev-Full and Jev-OneQ use the same 616 L1 statements: 297 candidate statements and 319 reference statements. Five rounds alternate Full and OneQ, with 616 uncached judgments per configuration and round, all using `jev-1.13.0`.

Table 3 counts a pair as identical when its score is exactly equal across all five runs. Each of the six settings evaluates the same 50 pairs five times. Between-run rank consistency is the median of all ten pairwise Kendall correlations. Correlation SD uses the sample standard deviation across five correlations with expert total-error counts. Mean expert correlations are 0.202 for OneQ, 0.211 for Full, 0.221 for GREEN, 0.221 for RadFact, 0.468 for RadMatch at temperature zero, and 0.400 for RadMatch at temperature one.

Jev’s matched experiment holds decomposition fixed and measures API judgment repeats; the baselines repeat their local evaluation pipelines. Every row in Table 3 uses the same pairs and repetition count. The RadEvalX experiment above provides an additional check on both configurations.

Table A12: RadMatch matching-stage validation across all five runs per setting. Each setting includes 250 report-pair evaluations on the same 50 pairs. Retried counts evaluations requiring at least one retry.

Setting	Validation fallback	Retried
Temperature 0	17/250 (6.8%)	61/250 (24.4%)
Temperature 1	9/250 (3.6%)	70/250 (28.0%)

Disabling sampling improves scoring stability while validation failures remain: temperature zero triggers fallback in 17 of 250 evaluations, compared with nine at temperature one (Table A12). Repeated scores and output validation capture different aspects of operational reliability.