EVALUATING RUBRIC GENERATION WITH INTERVENTIONAL TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Instance-specific rubrics are common in AI benchmarks where reliable evaluation requires specific expert knowledge. This approach is difficult to scale, prompting research into the generation of rubrics with large language models (LLMs). However, even when expert rubrics are available as references, it is unclear how to productively evaluate the quality of generated rubrics at scale. In this paper, we introduce a method for the evaluation of rubric generation, which we call Interventional Transfer (IT), based on the idea that two rubrics are similar if they move together when a response is perturbed to pass/fail one of them. In contrast to existing approaches for evaluation of rubric generation, we argue that different forms of interventional transfer can be used to evaluate the utility of generated rubrics for different tasks. For instance, we apply this approach in a case study on HealthBench, where we demonstrate an asymmetry in rubrics generated by Qwen3.8-27B, Deepseek-V4-Flash, and Opus-5, used to evaluate responses from GPT-5.6-Terra. Perturbations that degrade responses according to the generated/expert rubric *transfer* into lower scores on the corresponding expert/generated rubric, but perturbations that improve on one rubric do not reliably transfer into higher scores on the other. We argue that this finding has implications for the usage of LLM-generated rubrics for performance monitoring and hill-climbing. We contrast our approach with existing approaches for rubric evaluation, which do not surface the same asymmetry that we observe.

1 INTRODUCTION

Instance-specific rubrics are a promising approach to automated evaluation using large language models (LLMs) as judges. They aim to reduce subjectivity by instructing the LLM judge to attend to specific qualities of a response, unlike unstructured prompts or general task-specific rubrics. Instance-specific rubrics are typically a collection of instance-specific criteria that a response either passes or fails. For instance, an instance-specific criterion might read "include guidance on acetaminophen dosage for the user's weight of 50kg" while a more general-purpose rubric might include a criterion like "include all relevant clinical facts". The latter requires the LLM to know all clinical facts, but the former specifies exactly what to do. Instance-specific rubrics are used in benchmarks across healthcare (Arora et al., 2025), law (Shi et al., 2026), and research (Sharma et al., 2026), but are hard to scale because the expert effort involved scales with the number of examples.

The difficulty with scaling instance-specific rubrics has prompted research into the task of *rubric generation* using LLMs (Dhole & Agichtein, 2026; Chen et al., 2026b; Cook et al., 2024; Gunjal et al., 2026; Shen et al., 2026). As with any task, an important aspect of rubric generation is determining how to *evaluate* rubric generators. In this paper, we focus on the setting where expert-written rubrics (*expert rubrics*) are available on a set of examples, and can be used to evaluate the *generated rubrics* produced by LLMs. Evaluation of rubric generators in this context requires a method for evaluating the quality of generated rubrics.

Two existing families of approaches to rubric evaluation are (a) directly comparing rubric criteria for semantic overlap (Zhou et al., 2026; Siro et al., 2026; Dhole & Agichtein, 2026; Hashemi et al., 2024) and (b) comparing the scores assigned by expert vs. generated rubrics on a population of responses (Chen et al., 2026b; Hong et al., 2026). However, generated criteria are rarely exactly equivalent to expert criteria, either in lexical terminology or exact semantic meaning. Moreover, even criteria that measure the same dimension of quality may differ in their specificity or specific

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

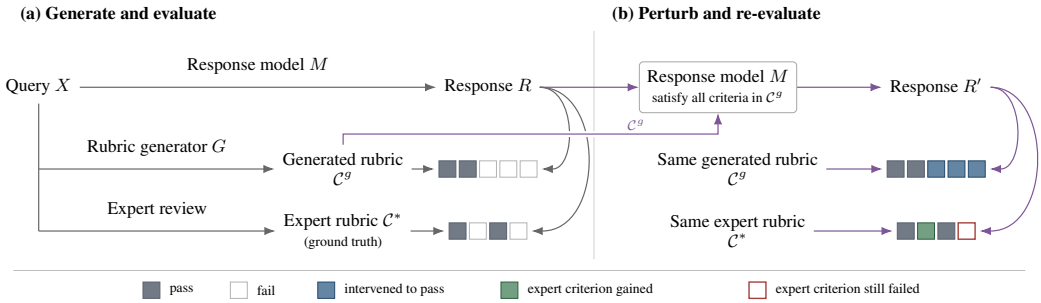


Figure 1: Overall setting and response perturbation. **(a)** Model M produces a response R to query X, rubric-generator G generates a rubric $\mathcal{C}^g$ from X alone, and humans generate the expert rubric $\mathcal{C}^*$. Both rubrics evaluate the same response. The boxes / cells represent criteria, and are filled (in grey) when passed and empty when failed. **(b)** The response model M edits R to satisfy all criteria in $\mathcal{C}^g$, and the resulting R' is evaluated on $\mathcal{C}^*$. Blue cells show generated criteria that are passed by design after intervention; green and red show an expert criterion satisfied post-intervention and one still failed. Outcomes are illustrative.

details, and the relevance of these differences is dependent on the distribution of responses being evaluated via rubric-based approaches. Meanwhile, methods based on comparing scores depend intrinsically on the population of responses being scored, and the exact comparison being performed (e.g., whether examining correlation of expert/generated scores across examples for a single response model, across response models at the example level, etc.).[1]

In this paper, we introduce an alternative framework for evaluating rubric generation, a framework which we refer to as *interventional transfer*, illustrated in Figure 1. Mechanically, our approach is to take a set of task-relevant responses (e.g., from an existing response model), perturb (or "intervene on") those responses to pass/fail selected criteria in a target (e.g., generated) rubric, and evaluate the impact on scores assigned by the non-target rubric (e.g., the expert rubric). We illustrate several variants of this approach, varying in the direction of intervention (pass/fail) and the target (e.g., generated or expert) criteria. Conceptually, we argue that interventional transfer is a useful proxy for performance on distinct downstream uses of rubrics, such as optimizing against a generated rubric, or using a generated rubric to monitor for degradation in response quality. We further argue that our approach provides a more intuitive notion of the *absolute quality* of generated rubrics in the context of different downstream use cases, in contrast to existing approaches described above.

We demonstrate these points in the context of applying our approach to the evaluation of rubric generation for HealthBench (Arora et al., 2025), a popular benchmark for health question-answering that includes expert-written rubrics. We consider the evaluation of rubrics generated by three models (Qwen3.8-27B, Deepseek-V4-Flash, and Opus-5) against expert rubrics, using a population of existing responses generated by GPT-5.6-Terra. We first illustrate the application of existing evaluation approaches to this dataset, showing that all rubrics exhibit moderate levels of semantic overlap with expert rubrics, and exhibit moderate levels of correlation with expert scores, though the precise values can be sensitive to implementation details.

We then demonstrate that our approach provides a perspective on task-dependent rubric utility. For instance, despite Qwen3.8-27B rubrics achieving moderate semantic overlap and correlation with expert scores, we observe **zero improvement** in expert rubric scores when **intervening on responses to pass Qwen3.8-27B generated rubrics**. In contrast, intervening to pass Opus-5 rubrics leads to a small, but statistically significant, improvement in expert rubric scores. We further demonstrate an asymmetry in the quality of rubrics generated by all three models: Interventions that **degrade responses** with respect to generated/expert rubrics tend to have substantial transfer to expert/generated rubric scores, while interventions to improve responses exhibit little transfer. These results suggest that, on this particular task, and with respect to this particular response model, the utility of generated rubrics lies mainly in their ability to detect degradations in response quality, rather than serving as a benchmark for improvement.

[1] We provide a compact summary of existing approaches in Tables A.1 and A.2.

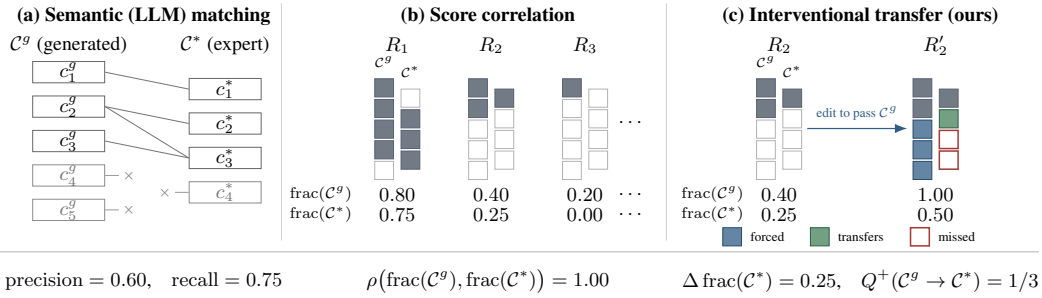


Figure 2: Illustration of three approaches for evaluating rubric generation. $\mathcal{C}^g$ is a 5-criterion generated rubric and $\mathcal{C}^*$ is a 4-criterion expert rubric. Scores are aggregated with $\text{frac}(\mathcal{C}; R)$, the fraction of passed criteria over total criteria. **(a)** Semantic matching of criteria, which enables the calculation of precision and recall metrics. **(b)** Correlation on a population of responses. **(c)** An example of an Interventional Transfer metric (ours), which measures the fraction of previously missed expert criteria that are passed after we intervene to pass all generated criteria.

In sum, our contributions are as follows: (1) We introduce the framework of *interventional transfer* (Section 3), along with a specific implementation for measuring the quality of rubrics for different tasks (e.g., rubrics as optimization targets, versus rubrics for degradation monitoring). (2) We demonstrate the application of our approach on HealthBench, and illustrate that our approach surfaces qualitatively different findings from existing metrics, including a distinction between the quality of generated rubrics as targets for improvement vs. as tools for degradation detection. (3) We illustrate that existing approaches do not surface these distinctions, and that existing metrics can vary substantially based on implementation details (e.g., definitions of what constitutes "similar" criteria).

2 Notation and Background

We first introduce some notation to describe our setting, before giving an overview of existing approaches to the evaluation of rubric generation. In our setting, we consider the problem of evaluating LLM responses to user queries using an instance-specific rubric, as illustrated in Figure 1(a). An **example** is a user query X. Our goal is to evaluate a **response** R to this query, generated by a **response-model** LLM M. For each example, we assume access to an **expert rubric**, denoted $\mathcal{C}^*$, and one or more **generated rubrics** $\mathcal{C}^g$. Each rubric consists of a set of binary **criteria** c that either pass (1) or fail (0) on a given response, and we use $c(R) \in \{0, 1\}$ to denote whether a response passes or fails a criterion. The **score vector** of a rubric $\mathcal{C}$ on response R is the binary vector indicating which criteria the response passes or fails, which we denote $\mathcal{C}(R) \in \{0, 1\}^{|\mathcal{C}|}$.[2]

Existing approaches to rubric evaluation tend to fall into two broad categories, which we term **semantic matching** and **score correlation**. These are shown in Figure 2 alongside our proposed approach. We use the term semantic matching to refer broadly to approaches that do not rely on having a population of responses, but which instead directly compare criteria for semantic similarity. For instance, Dhole & Agichtein (2026) prompt an LLM to assign a "semantic similarity" score 0–9 and Hong et al. (2026) threshold embedding cosine similarity. We use the term "score correlation" to refer broadly to approaches that examine the relationship (typically the rank correlation) between the scores of responses on the generated and expert rubrics. For instance, Cook et al. (2024) computes the Pearson correlation between fractions of passed criteria on generated and expert rubrics. We provide a more detailed overview of approaches and metrics in Table A.1 and Table A.2. Figure A.3 illustrates the distinction between these approaches and our own: These approaches typically examine correlation among a pre-existing population of responses, while we consider targeted perturbations to responses and measure the resulting change in non-targeted rubrics.

[2] In this paper, we focus on rubrics that take the form of a checklist, rather than rubrics that assign point values to individual criteria. In part, this focus reflects our desire to evaluate whether or not generated rubrics cover the same concepts as expert rubrics, rather than additionally assessing whether or not they assign the same importance to different concepts.

	intervene on	
intervene to	$\mathcal{C}^g$ (Generated Rubric)	$\mathcal{C}^*$ (Expert Rubric)
pass all rubric items	Fraction of *expert* criteria that shift from *failing* to *passing* post-intervention.	Fraction of *generated* criteria that shift from *failing* to *passing* post-intervention.
fail all rubric items	Fraction of *expert* criteria that shift from *passing* to *failing* post-intervention.	Fraction of *generated* criteria that shift from *passing* to *failing* post-intervention.

Figure 3: The Interventional Transfer metrics described in Section 3, each of which examines the extent to which *intervention* on a response to pass/fail a target rubric *transfers* to a corresponding improvement/degradation in the measurement rubric. Each cell has an interpretation in the context of a particular application of generated rubrics. Measuring with $\mathcal{C}^*$ the intervention to pass $\mathcal{C}^g$ corresponds to the quality of $\mathcal{C}^g$ as a training target; measuring the intervention to fail corresponds to the trustworthiness of a degradation flag from $\mathcal{C}^g$. Measuring with $\mathcal{C}^g$ the intervention to pass/fail $\mathcal{C}^*$ corresponds to the ability of $\mathcal{C}^g$ to detect good/bad responses.

Algorithm 1 Interventional transfer, for one generator G and one response model M

Require: examples $\{(X_i, \mathcal{C}_i^*)\}_{i=1}^n$, generator G, response model M; target rubric $\mathcal{C}^t \in \{\mathcal{C}^g, \mathcal{C}^*\}$ and measurement rubric $\mathcal{C}^m$ (the other one); direction $\delta \in \{+, -\}$ with end state b (1 for $+$, 0 for $-$)

for $i = 1, \ldots, n$ **do**
  $R_i \leftarrow M(X_i)$; $\mathcal{C}_i^g \leftarrow G(X_i)$
  $h_i \leftarrow \sum_{c \in \mathcal{C}_i^m} \mathbb{1}[c(R_i) \neq b]$ ▷ measurement criteria that could move to b
  $R_i' \leftarrow R_{i,\,\delta\mathcal{C}_i^t}$ ▷ M minimally edits R_i so every criterion of $\mathcal{C}_i^t$ ends at b
  $g_i \leftarrow \sum_{c \in \mathcal{C}_i^m} \left(\mathbb{1}[c(R_i') = b] - \mathbb{1}[c(R_i) = b]\right)$ ▷ net measurement criteria moved to b
return $Q \leftarrow \sum_i g_i / \sum_i h_i$

3 Our Framework: Interventional Transfer

We use the term interventional transfer to refer to a broader framework, illustrated in Figure 1(b), that relies on perturbations to responses that cause them to pass or fail selected criteria. These perturbations are implemented via prompting the response model M to modify a response R with instructions to keep to the minimal necessary changes to cause a criterion (or set of criteria) to pass or fail. We first describe the different types of interventions we consider and the measurements of interventional transfer that we use.

We consider both **whole-rubric** interventions, as well as **single criterion** interventions, each of which we describe next. For single-criterion interventions, we write R_{+c} for the response obtained by prompting M to minimally edit a response R with $c(R) = 0$ so that $c(R_{+c}) = 1$, and R_{-c} for the analogous edit of a response with $c(R) = 1$ so that $c(R_{-c}) = 0$. To describe whole-rubric interventions, we introduce the notion of a *pass/fail set*: The pass set of a rubric $\mathcal{C}$, with X implicit, is the set of R that pass every criterion, denoted $\mathcal{R}(\mathcal{C}) := \{R : \mathcal{C}(R) = \mathbf{1}\}$. The fail set is $\mathcal{R}(\neg\mathcal{C}) := \{R : \mathcal{C}(R) = \mathbf{0}\}$, the set of responses that fail every criterion. For whole-rubric interventions, we write $R_{+\mathcal{C}}$ for the response obtained by editing R into the pass set $\mathcal{R}(\mathcal{C})$, and $R_{-\mathcal{C}}$ for the response obtained by editing R into the fail set $\mathcal{R}(\neg\mathcal{C})$.

When we intervene on a response to make it pass or fail a particular rubric or criterion, we refer to that rubric or criterion as the **target** rubric/criterion. The metrics we define largely consider whether or not intervening on a target rubric (e.g., the generated rubric) leads to a change in the non-target rubric (e.g., the expert rubric). In Figure 3 we show the four possible combinations of target rubric and direction of intervention, each of which is associated with a metric we define below.

Each metric Q (for "quality") captures the number of criteria in the measurement rubric that moved *due to the intervention on the target rubric*, as a fraction of measurement criteria that could have

moved.

$$Q^{\pm}(\mathcal{C}^t \to \mathcal{C}^m) = \frac{|\{c \in \mathcal{C}^m : c(R_{\pm\mathcal{C}^t}) = b\}| - |\{c \in \mathcal{C}^m : c(R) = b\}|}{|\{c \in \mathcal{C}^m : c(R) = a\}|}, \qquad (a,b) = \begin{cases} (0,1) & \text{for } + \\ (1,0) & \text{for } - \end{cases} \tag{1}$$

For example, for $Q^{+}(\mathcal{C}^g \to \mathcal{C}^*)$, with target $\mathcal{C}^g$ and measurement $\mathcal{C}^*$, the numerator is the number of criteria in $\mathcal{C}^*$ that went from 0 to 1 and the denominator is the number of criteria in $\mathcal{C}^*$ that were 0 under R.

The four combinations in Figure 3, and their corresponding quality metrics Q, can be seen as proxies for evaluating the utility of rubrics for different tasks: For instance, the quality of $\mathcal{C}^g$ as an **optimization target** is the extent to which passing generated criteria causes a response to pass expert criteria, and is measured by $Q^{+}(\mathcal{C}^g \to \mathcal{C}^*)$. We illustrate these perspectives, and their application to real data, in the experiments below.

4 Experiments

4.1 Data and Setup

All experiments use the same $n = 105$ queries and expert rubrics from HealthBench (Arora et al., 2025), which we obtained via stratified random sampling, choosing 15 examples from each of the 7 themes while excluding examples that are part of the HealthBench Hard subset. For our primary analysis, we obtain a response R for each query using GPT-5.6-Terra as our response model M. We evaluate three rubric generator models, using a representative example of a small open-weight model (Qwen3.8-27B), a larger open-weight model (Deepseek-V4-Flash), and a closed frontier model (Opus-5). We chose them because they have similar release dates, so Figure 5 shows a realistic comparison that could have been done around the time of their release. Scores are given by Gemini 3.8 Flash (medium effort) using majority vote over 3 samples using the prompt in Appendix E. Reasoning behind every model choice in the pipeline is in Table A.3. Intervals are 95% bootstrap intervals, see Algorithm 4.

4.2 Implementation of Existing Approaches

Existing rubric-evaluation methods compare generated to expert rubrics in broadly two ways: semantic similarity between criteria (using n-grams, embeddings, LLMs) and computing a relationship between the scores the two rubrics assign to a population of responses. We consider four such metrics from two recent rubric-generation papers, RubricRAG and GenRubric (Dhole & Agichtein, 2026; Chen et al., 2026b). Together these baselines represent the many slight variations on holistic evaluation of rubric-generators found in the literature, surveyed in Appendix B. Additional detail of our implementation of each baseline approach is given in Algorithms 2 and 3.

- **(RubricRAG) LLM-matching**: This approach prompts an LLM to compute a similarity score $\mathrm{sim}(c^g, c^*) \in \{0, 1, \ldots, 9\}$ between each pair of individual criteria, and defines $R := \frac{1}{|\mathcal{C}^*|}\sum_{c^*} \max_{c^g} \mathrm{sim}(c^g, c^*)/9$ as a notion of recall, and $P := \frac{1}{|\mathcal{C}^g|}\sum_{c^g} \max_{c^*} \mathrm{sim}(c^g, c^*)/9$ as a notion of precision. We follow Dhole & Agichtein (2026) in reporting the corresponding harmonic mean $F_1 := \frac{2PR}{P+R}$.
- **(GenRubric) Score-Correlation**: This approach generates responses $R_{k=1,\ldots,8}$ for each example with 8 different models (following Chen et al. (2026b)) and computes, for each example, the Spearman correlation $\rho := \mathrm{corr}\big(\mathrm{rank}(s^g), \mathrm{rank}(s^*)\big)$ between the scores $s^g_k := \frac{\sum_{c \in \mathcal{C}^g} c(R_k)}{|\mathcal{C}^g|}$ and $s^*_k := \frac{\sum_{c \in \mathcal{C}^*} c(R_k)}{|\mathcal{C}^*|}$ over k, and gives the average over examples.
- **(RubricRAG) Score-Correlation on physician responses**: Dhole & Agichtein (2026) consider a different population of responses for measuring correlation, using the physician-written responses in HealthBench and examining correlation in scores on those responses across examples. We compute the Spearman correlation over the examples that have a physician response ($92/105$ examples in our context).
- **(RubricRAG) Discrimination Between High/Low Quality Responses**: Dhole & Agichtein (2026) further investigate whether or not generated rubrics can distinguish between physician responses (presumed to be high-quality) and responses designed to be poor, using Qwen3-30B-A3B instructed to satisfy a random (unrelated) expert rubric from

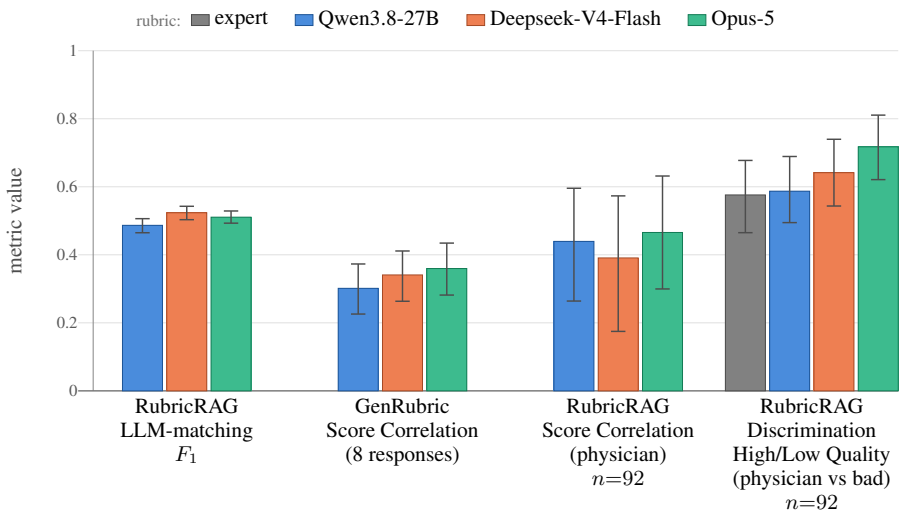


Figure 4: Published approaches to rubric evaluation semantically match criteria (left), or relate generated scores to expert scores with rank-correlation on a panel of responses (left-mid), rank-correlation across physician responses (right-mid), or by discrimination between high/low quality responses on physician responses versus artificially degraded responses (right).

another example. Letting R^{phy} be a physician response and R^{bad} be a "bad" response constructed as described, we follow their approach in giving for each of $\mathcal{C} \in \{\mathcal{C}^g, \mathcal{C}^*\}$

$$\mathbb{E}\left[\mathbb{1}\left[\frac{\sum_{c\in\mathcal{C}} c(R^{\text{phy}})}{|\mathcal{C}|} > \frac{\sum_{c\in\mathcal{C}} c(R^{\text{bad}})}{|\mathcal{C}|}\right]\right]$$

4.3 Case Study: Application of Existing Approaches

The results of applying existing methods to our data are given in Figure 4. Before showing the results of applying our framework, we make the following observations.

In absolute terms, rubric quality appears mediocre, without significant differences between generators While not all scores are directly comparable, each one is defined in a way such that 0 represents extremely poor quality, and 1 represents the best possible quality. We observe that for all rubric generation approaches, the scores lie broadly in the middle range (between .3–0.7 for all 4 metrics), which may give the impression that all rubric generators are of moderate but not exceptional quality. We will contrast with this impression in the next section, when we present our approach. Meanwhile, using pairwise hypothesis testing in Table A.4, we observe that none of the pairwise differences between generators are statistically significant, suggesting that they are all broadly equivalent in quality, with three exceptions: Qwen3.8-27B rubrics lose to Deepseek-V4-Flash and Opus-5 rubrics in F_1, and Qwen3.8-27B is worse than Opus-5 at discrimination between high/low quality responses.

Measures of absolute performance vary based on implementation details We further investigate the sensitivity of existing approaches to implementation details: For LLM-matching, these details include the prompting approach used to determine matches, and for score correlation measures, these details include the choice of response models in the panel. In Figure A.1 we observe that the absolute quality implied by LLM-matching depends on how semantic similarity is defined in the prompt, ranging from the F_1 scores of around 0.5 shown here, to as low as 0.1 when using a prompt that only considers a match to exist if one criterion logically entails the other. In Figure A.2, we observe that absolute measures of performance for score correlation are dependent on the choice of response models, even if the relative ranking of rubric generators is fairly stable.

Existing approaches to identifying or generating "good" and "bad" responses have limitations The two metrics on the far right of Figure 4, drawn from Dhole & Agichtein (2026), evaluate generated rubrics by their correlation with expert rubrics on physician responses, and on their ability to discern a physician response from an intentionally degraded response. We find that the expert

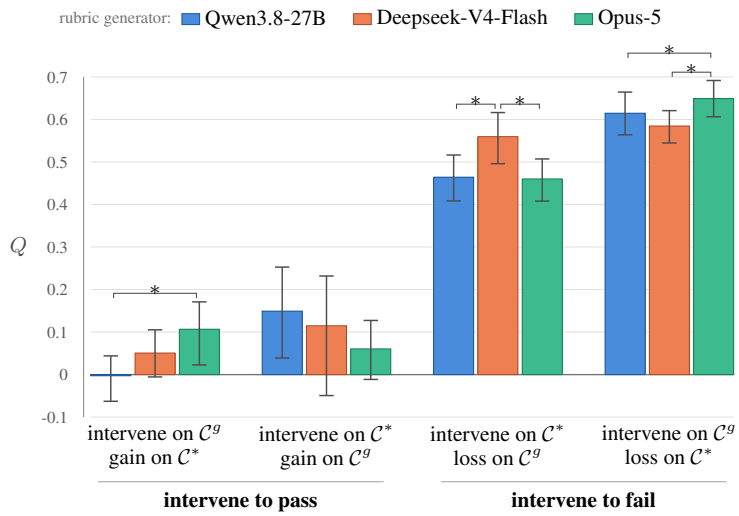


Figure 5: Application of our approach to HealthBench data. Each group represents a distinct whole-rubric intervention on responses from GPT-5.6-Terra to either pass or fail the generated rubric $\mathcal{C}^g$ or the expert rubric $\mathcal{C}^*$. The Y-axis is $Q^{\pm}(\mathcal{C}^t \rightarrow \mathcal{C}^m)$ from Equation (1): the net number of measurement-rubric criteria that moved to b after intervening to move the target rubric from a to b, divided by the number of measurement criteria initially at a. Differences where the pairwise confidence interval does not include zero are indicated by $*$. We observe that intervening to fail target criteria that currently pass (1 on the response) has a far greater effect on the measurement rubric than intervening to pass criteria that currently fail.

rubrics prefer physician responses only around 60% of the time when contrasting against the "bad" responses, which casts doubt on the utility of physician-written responses as a "gold standard" that is assumed to score well on the expert rubric. Furthermore, in Appendix Table A.5 we investigate directly the impact of their approach for constructing "bad" responses by measuring the change in expert score before and after the degradation. We find decreases capped at 37% of expert criteria, and a pattern that larger models degrade by about 10% compared to about 20% for smaller models (the 37% outlier is Grok 4.20). The implication is that this type of degradation is worse at generating bad responses for large models, and the responses it does generate could be worse.

4.4 Case Study: Application of Our Approach

We demonstrate our approach using both rubric-level interventions and criteria-level interventions, and demonstrate that our approach yields qualitatively different insights from prior approaches.

4.4.1 Rubric-Level Interventions

To start, we consider rubric-level interventions, with results reported in Figure 5. We demonstrate that rubric-level interventions reveal an asymmetry in rubric quality that is not revealed by the metrics in the prior section. In particular, all three rubric generators produce rubrics that mirror the expert rubrics to a greater extent on degradation than on improvements to responses. We also note that our approach reveals a qualitative difference in the quality of individual rubric generators when considered in the context of different interventions. For instance, intervening to pass Opus-5 rubrics yields statistically significant gains on the expert rubric, and Opus-5 rubrics are significantly better at yielding gains than Qwen3.8-27B rubrics.

The absolute quality is higher for all rubrics on tasks that intervene to fail than on tasks that intervene to pass. First, we observe that generated rubrics lose $40-65\%$ of passing criteria when a response is made to fail the expert rubric, and expert rubrics lose $55-70\%$ when a response is made to fail the generated rubric. However, when the response is made to pass all criteria in the generated rubric, the gain is significantly above 0 only for Opus-5 and caps at 20%. Similarly, a response made to pass the expert rubric gains significantly more than 0 only for Qwen3.8-27B, though that difference is itself not statistically significant when compared to other rubrics. In Figure A.4, we

find the same distinction between passing and failing criteria across different response models, with the exception of Sonnet 4.6 and Grok 4.20.

One possible interpretation of this observation is that initially passing criteria have more in common with each other than initially failing criteria. To investigate this further, we examine the similarity (as judged by the maximum similarity score, between 0–9, between a generated criterion and all the expert criteria) of generated criteria that are initially passing, and those that are initially failing. We give the results in Appendix Table A.6, where we find a significant difference in the scores assigned by LLM-matching to passing and failing criteria, around 4.5 and 3.5 respectively.

The differences between rubric-generators are task-dependent One way to view each of the four metrics is through the lens of distinct downstream tasks: For instance, intervening on the generated rubric to pass all criteria could be viewed as a proxy for what might occur when training a model using this rubric as a reward. For that purpose, our analysis provides evidence that Opus-5 is superior to Qwen3.8-27B, as it is the only rubric generator that yields significant gains on the expert rubric when used as the target rubric, and the difference along those lines to Qwen3.8-27B is also significant. Meanwhile, however, Deepseek-V4-Flash has a statistically significant advantage over the other rubric generators when it comes to detecting a large degradation in response quality, as measured by the expert rubric.

4.4.2 Criteria-Level Interventions and Measures

In the prior section, we considered substantial interventions that intervene to move all criteria simultaneously. In this section, we demonstrate that similar trends hold when considering more targeted interventions to pass or fail individual criteria one-at-a-time, and that this type of criteria-level analysis can shed further light on the conceptual overlap between the expert and generated rubrics. Recall that criteria-level interventions measure the effect on a rubric of changing a response to satisfy a single criterion. Each intervention starts from the original response and edits it for one criterion only, so the interventions are not stacked: every criterion gets its own rewrite of R, and no rewrite has the edits made for other criteria.

We use these interventions to categorize criteria into one of a few categories. We call a target criterion **irrelevant** to the measurement rubric if intervening to pass or fail the target criterion leads to *no change* in the set of measurement rubric criteria that are satisfied. We call a target criterion **matched** if the intervention leads to a *single* criterion moving in the measurement rubric, and we similarly call a target criterion **general** if the intervention leads to *multiple* criteria moving in the measurement rubric. Because we intervene on only a single criterion in a given rubric, we can also measure *within-rubric* effects of these interventions. In particular, we diagnose redundancy by measuring the effect of targeting c on the rubric from which c originates, reasoning that if editing for c moves a different criterion c' in its own rubric, then these criteria may be considered to measure similar things. We note that the LLM performing the edit for c does not see c' at the time of intervention. Figure 6 shows the fraction of these categories in each rubric, and we can make the following observations.

Similar trends between criteria-level and rubric-level interventions. In the first three rows, interventions on generated criteria measured by expert rubrics, we observe a higher rate of irrelevance in initially failing criteria of $82 - 87\%$ compared to $47 - 57\%$ for initially passing. We observe in the second three rows that rates of matched and general criteria, in particular, are higher in passing criteria in $\mathcal{C}^*$ than in failing criteria. Both of these observations are consistent with the observations in Figure 5. We attribute the slight differences to interactions between criteria, such as a general criterion flipping a criterion matched to a different criterion, which would not be double counted in the rubric-level intervention.

Passing criteria are more redundant and more general. Across all rubrics in the last four rows, we observe higher rates of redundancy in passing criteria than in failing criteria. In particular, the expert rubric is the most redundant over both passing and failing criteria, with respectively 53% and 28% of criteria exhibiting some form of redundancy (orange + purple). We note again the last column of the second three rows, the effect of editing for c^* on $\mathcal{C}^g$, where rates of generality and match were higher than on the first three rows, edits for c^g measured in $\mathcal{C}^*$.

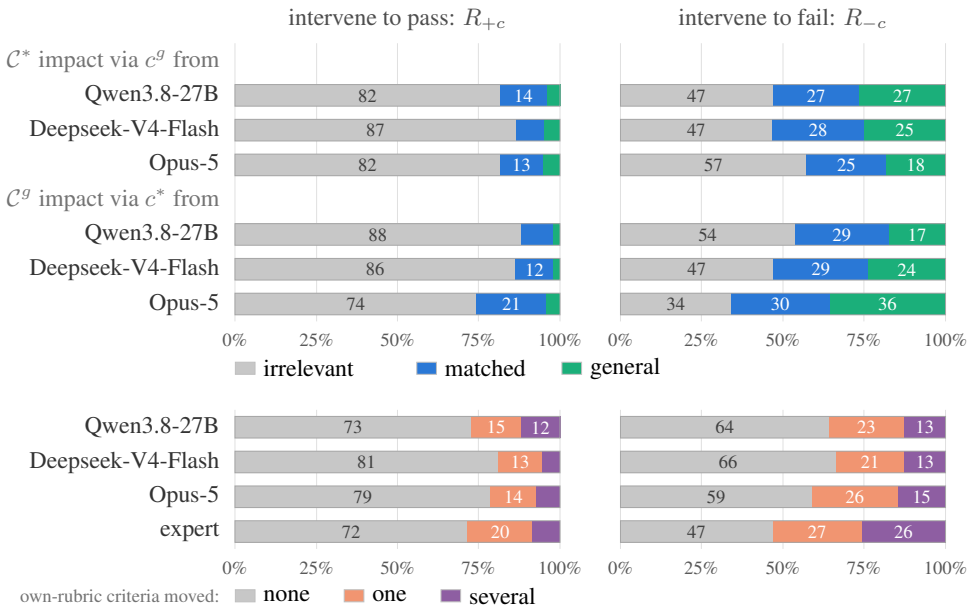


Figure 6: Criteria-level transfer between and within rubrics. Each bar is one rubric's criteria on responses from GPT-5.6-Terra, with columns designating whether the intervention is to pass (R_{+c}, left) or to fail (R_{-c}, right). Every criterion c gets its own intervention on the original response. In the upper chart, the first three rows intervene on criteria from each generated rubric $\mathcal{C}^g$ and measure the resulting change on the expert rubric $\mathcal{C}^*$. The next three rows intervene on expert criteria and measure changes in each generated rubric. In the top six rows, each criterion is categorized by how many criteria of the *other* rubric move with it. A target criterion is **irrelevant** if no criterion in the measurement rubric moves in the intervention direction, **matched** if one criterion moves, and **general** if several criteria move. The lower chart measures within-rubric effects. After one target criterion is made to pass or fail, it reports whether zero, one, or several *other* criteria (excluding itself) in that same rubric also move in the intervention direction. Numbers are percentages of the row's criteria. We observe that irrelevant criteria make up 82–87% of initially failing generated criteria but only 47–57% of initially passing ones, consistent with the pass/fail asymmetry of Figure 5, and that passing criteria are more redundant than failing criteria in every rubric.

5 Conclusion

In this paper, we introduce a framework for the evaluation of rubric generation approaches, based on the concept that a generated rubric is a suitable proxy for the expert rubric insofar as changes to the scores of responses as measured by one rubric are reflected in the other rubric. A core thesis of our approach is that evaluation of rubric generation should be tied to the downstream task for which a rubric is used. For instance, if the goal of rubric generation is to provide a target for downstream optimization, then it is worth understanding whether passing all generated criteria implies that scores will improve according to an expert rubric. In our case study, we show that this relationship often does not hold, even for generated rubrics whose scores are otherwise moderately correlated with expert rubric scores. On the other hand, if the goal of rubric generation is to provide a baseline for measuring degradation in response quality (e.g., due to unanticipated side effects of changes to prompts), then it is worthwhile to understand whether degradations in quality (as measured by the expert rubric) necessarily lead to degradation in quality as measured by the generated rubric. In our case study, for instance, we observe that on GPT-5.6-Terra responses, most rubric generators show some utility along these lines. More broadly, insofar as generated rubrics are intended to track *changes* in response quality (for a given response model like GPT-5.6-Terra), we argue that comparing how scores change under intervention is more natural than, e.g., assessing if a rubric can reliably distinguish between responses from a more capable model (like GPT-5.6-Terra) vs. a weaker model (like GPT-4o). Beyond the specific metrics we propose, we hope that the framework we present leads to new metrics for different tasks, and generally improves automatic rubric generation.

Reproducibility Statement

Numbers in the paper come from the same HealthBench conversations, which are publicly available and were selected with the rule given in Section 4. Appendix D gives each metric and each baseline as pseudocode, including the paired bootstrap, and Appendix E has every prompt. In Table A.3 we detail all the model choices for each part of the pipeline.

LLM Usage

We used LLMs, with significant human oversight, to write and debug experiment code, to find related work, to create figures and tables, and to check the main body for typos/grammar/phrasing. We also The research questions, the framework, the choice of metrics, and the claims drawn from the results are the authors' own.

REFERENCES

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint (2505.08775v1)*, 5 2025. URL http://arxiv.org/abs/2505.08775v1.

Hao Chen, Ziyu Han, Yukun Yan, Qingfu Zhu, Maosong Sun, and Wanxiang Che. From holistic evaluation to structured criteria: Rubrics across the evolving LLM landscape. 2026a. URL https://arxiv.org/abs/2606.08625.

Yifan Chen, Haitao Li, Qingyao Ai, Fengbin Zhu, Tat-Seng Chua, Min Zhang, and Yiqun Liu. GenRubric: Self-evolving rubric generation for scalable LLM evaluation. 2026b. URL https://arxiv.org/abs/2608.29856.

Yinzhu Chen, Abdine Maiga, Hossein A. Rahmani, and Emine Yilmaz. Retrieval-augmented agentic rubric generation for reliable medical response evaluation. 2026c. URL https://arxiv.org/pdf/2601.15161.

Jonathan Cook, Tim Rocktäschel, Jakob Foerster, Dennis Aumiller, and Alex Wang. Ticking all the boxes: Generated checklists improve llm evaluation and generation. *arXiv preprint (2410.03608v1)*, 10 2024. URL http://arxiv.org/abs/2410.03608v1.

Kaustubh D. Dhole and Eugene Agichtein. Rubricrag: Towards interpretable and reliable llm evaluation via domain knowledge retrieval for rubric generation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2026. URL http://arxiv.org/abs/2603.20882v1.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6556–6576, 2024. URL https://arxiv.org/abs/2302.04166.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean M. Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.17746.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. LLM-Rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13806–13834, 2024. doi: 10.18653/v1/2024.acl-long.745. URL https://arxiv.org/abs/2501.00274.

Hanhua Hong, Yizhi Li, Jiaoyan Chen, Luu Gia Huy, Sophia Ananiadou, Jung-jae Kim, and Chenghua Lin. Can LLMs write reliable rubrics? a meta-evaluation for experiment reproduction. 2026. URL https://arxiv.org/abs/2607.12835.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, Xijun Gu, Peiyi Tu, Jiaxin Liu, Wenyu Chen, Yuzhuo Fu, Zhiting Fan, Yanmei Gu, Yuanyuan Wang, Zhengkai Yang, Jianguo Li, and Junbo Zhao. Reinforcement learning with rubric anchors, 2025. URL https://arxiv.org/abs/2508.12790.

Wenhan Liu, Jiajie Jin, Zhaoheng Huang, Tongyu Wen, Guanting Dong, Ziliang Zhao, Yutao Zhu, Zhicheng Dou, and Ji-Rong Wen. The rules of the game: A survey of rubrics for large language models, 2026. URL https://openreview.net/pdf?id=FnSimngGYk.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, 2023. doi: 10.18653/v1/2023.emnlp-main.153. URL https://arxiv.org/abs/2303.16634.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. Hd-eval: Aligning large language model evaluators through hierarchical criteria decomposition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7641–7660, 2024. doi: 10.18653/v1/2024.acl-long.413. URL https://arxiv.org/abs/2402.15754.

Tianjun Pan, Xuan Lin, Wenyan Yang, Qianyu He, Shisong Chen, Licai Qi, Wanqing Xu, Hongwei Feng, Bo Xu, and Yanghua Xiao. RubricEval: A rubric-level meta-evaluation benchmark for LLM judges in instruction following. 2026. URL https://arxiv.org/abs/2603.25133.

Zhengyang Qi, Charles Dickens, Derek Pham, Amanda Dsouza, Armin Parchami, Frederic Sala, and Paroma Varma. RIFT: A RubrIc failure mode taxonomy and automated diagnostics. 2026. URL https://arxiv.org/abs/2604.01375.

Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. Branch-solve-merge improves large language model evaluation and generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, 2024. URL https://arxiv.org/abs/2310.15123.

Shreya Shankar, J. D. Zamfirescu-Pereira, Björn Hartmann, Aditya G. Parameswaran, and Ian Arawjo. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2024. doi: 10.1145/3654777.3676450. URL https://arxiv.org/abs/2404.12272.

Manasi Sharma, Chen Bo Calvin Zhang, Chaithanya Bandi, Clinton Wang, Ankit Aich, Huy Nghiem, Tahseen Rabbani, Ye Htet, Brian Jang, Sumana Basu, Aishwarya Balwani, Denis Peskoff, Marcos Ayestaran, Sean M. Hendryx, Brad Kenstler, and Bing Liu. Researchrubrics: A benchmark of prompts and rubrics for evaluating deep research agents. In *International Conference on Learning Representations*, 2026. URL http://arxiv.org/abs/2511.07685v1.

William F. Shen, Xinchi Qiu, Chenxi Whitehouse, Lisa Alazraki, Shashwat Goel, Francesco Barbieri, Timon Willi, Akhil Mathur, and Ilias Leontiadis. Rethinking rubric generation for improving LLM judge and reward modeling for open-ended tasks. 2026. URL https://arxiv.org/abs/2602.05125.

Yuzhen Shi, Huanghai Liu, Yiran Hu, Gaojie Song, Xinran Xu, Yubo Ma, Tianyi Tang, Li Zhang, Qingjing Chen, Di Feng, Wenbo Lv, Weiheng Wu, Kexin Yang, Sen Yang, Wei Wang, Rongyao Shi, Yuanyang Qiu, Yuemeng Qi, Jingwen Zhang, Xiaoyu Sui, Yifan Chen, Yi Zhang, An Yang, Bowen Yu, Dayiheng Liu, Junyang Lin, Weixing Shen, Bing Zhao, Charles L. A. Clarke, and Hu Wei. Plawbench: A rubric-based benchmark for evaluating llms in real-world legal practice. *arXiv preprint (2601.16669v2)*, 1 2026. URL http://arxiv.org/abs/2601.16669v2.

Clemencia Siro, Pourya Aliannejadi, and Mohammad Aliannejadi. Learning to judge: Llms designing and applying evaluation rubrics. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6371–6389, 2026. doi: 10.18653/v1/2026.findings-eacl.335. URL https://arxiv.org/abs/2602.08672.

Matt Turk. Counterfactual evaluation reveals hidden capability profiles in clinical LLMs and agents. 2026. URL https://arxiv.org/abs/2605.30590.

Vijay Viswanathan, Yanchao Sun, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. Checklists are better than reward models for aligning language models. In *Advances in Neural Information Processing Systems*, 2025.

Zijie Wang and Eduardo Blanco. Generating and refining dynamic evaluation rubrics for LLM-as-a-judge. 2026. URL https://arxiv.org/abs/2605.30568.

Lipeng Xie, Sen Huang, Zhuo Zhang, Anni Zou, Yunpeng Zhai, Dingchao Ren, Kezun Zhang, Haoyuan Hu, Boyin Liu, Haoran Chen, et al. Auto-rubric: Learning to extract generalizable criteria for reward modeling. 2025. URL https://arxiv.org/abs/2510.17314.

Zhichao Yang, Sepehr Janghorbani, Dongxu Zhang, Jun Han, Qian Qian, Andrew Ressler, Gregory D. Lyng, Sanjit Singh Batra, and Robert E. Tillman. Health-SCORE: Towards scalable rubrics for improving health-LLMs. 2026a. URL https://arxiv.org/abs/2601.18706.

Zihao Yang, Mosh Levy, Yoav Goldberg, and Byron C. Wallace. Compared to what? baselines and metrics for counterfactual prompting. 2026b. URL https://arxiv.org/abs/2605.01048.

Zhiling Ye, Yun Yue, Haowen Wang, Xudong Han, Jiadi Jiang, Cheng Wei, Lei Fan, Jiaxin Liang, Shuowen Zhang, Ji Li, Chunxiao Guo, Jian Wang, Peng Wei, and Jinjie Gu. Self-rewarding rubric-based reinforcement learning for open-ended reasoning. 2025. URL https://arxiv.org/abs/2509.25534.

Junyi Zhou, Qiyuan Zhang, Yufei Wang, Fuyuan Lyu, Yidong Ming, Can Xu, Qingfeng Sun, Kai Zheng, Peng Kang, Xue Liu, and Chen Ma. RubricBench: Aligning model-generated rubrics with human standards. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31179–31200, 2026. doi: 10.18653/v1/2026.acl-long.1439. URL https://aclanthology.org/2026.acl-long.1439/.

A Related Work

A.1 Prior Work

Instance-specific rubrics. LLM judges are sometimes given a single rubric of general criteria that applies to every example in a dataset (Liu et al., 2023; Fu et al., 2024). In this paper, we focus on domains where judging an answer requires expert knowledge, and some benchmarks in these domains give each example its own expert-written criteria, which we call an instance-specific rubric (Arora et al., 2025; Shi et al., 2026; Sharma et al., 2026).

Generating rubrics. Many papers propose new methods for generating rubrics. RubricRAG (Dhole & Agichtein, 2026) and Chen et al. (2026c) retrieve domain knowledge to guide generation, and Chen et al. (2026b) propose a self-evolving rubric generator. Cook et al. (2024) generate a checklist of yes/no questions for each query, and use it both to evaluate responses and to improve them by revising a response to satisfy failed items. Saha et al. (2024) generate evaluation criteria from the question alone and validate the judgments those criteria produce. Wang & Blanco (2026) generate dataset-level and instance-level rubrics, train the generator on a meta-judge's pairwise preferences between rubrics, and evaluate the rubrics by their agreement with human judgments. Xie et al. (2025) extract criteria for reward modeling from human preference data without comparing them to expert rubrics. Shen et al. (2026) recursively decompose generated rubrics into finer criteria, then filter and weight them. Yang et al. (2026a) cluster the criteria of HealthBench's instance-specific rubrics into a shared set of 29 criteria, and evaluate this set by the quality of the responses it produces, when used as an RL reward. Many such papers define measures of rubric quality: coverage, the share of examples on which a rubric gives a discriminative signal; precision, how often its judgment agrees with the preference label; and contribution, the drop in accuracy when it is removed.

Using rubrics. Other work uses rubrics for a downstream task, most often as rewards for reinforcement learning. Work on rubrics as rewards trains on generated rubrics and reports gains on expert rubrics (Gunjal et al., 2026; Shen et al., 2026) or on standard benchmarks (Huang et al., 2025; Viswanathan et al., 2025). Huang et al. (2025) mix generated rubrics with human-written ones. Viswanathan et al. (2025) generate a checklist of requirements for each instruction and use the response's checklist score as the reward in place of a reward model. Ye et al. (2025) train on HealthBench's expert criteria rather than on generated ones.

Assessing other properties of rubrics. A few papers assess properties of rubrics other than their agreement with a reference. Qi et al. (2026) assess rubrics using a taxonomy of eight failure modes and a few automated signals, such as an LLM classifier prompted with the taxonomy and agreement between labelers. Pan et al. (2026) measure whether LLM judges apply individual criteria correctly by viewing them as instructions.

Evaluating generated rubrics. Most papers evaluate rubrics by comparing them to an expert rubric for the same example. Comparisons to an expert rubric fall into two groups: agreement between the criteria themselves, and agreement between the scores the two rubrics give to the same responses. For the first, RubricRAG (Dhole & Agichtein, 2026) uses n-gram overlap and LLM ratings of criterion similarity, and Hong et al. (2026) use lexical and embedding similarity. For the second, RubricRAG and Hong et al. (2026) correlate the scores of the two rubrics, and Chen et al. (2026b) compute the Spearman correlation between the two rubrics' scores over responses from models of different sizes, along with top-1 agreement and pairwise accuracy. Cook et al. (2024) use both kinds of comparison for generated checklists, and score correlation is also the standard way LLM judges are validated against human judgments (Hashemi et al., 2024; Liu et al., 2024; Shankar et al., 2024). Zhou et al. (2026) measure the recall and hallucination rate of generated criteria against human-written reference rubrics, and also check whether the preferences a rubric induces on response pairs match human preferences. Chen et al. (2026c) measure how much of a physician-written rubric a generated rubric covers, and also check whether a rubric scores a response above a copy of it with one injected clinical error. RubricRAG runs a similar check without a reference rubric, testing whether a rubric prefers a physician-written response over a deliberately degraded one. Evaluation is rarely the main subject of these papers.

In a recent survey of rubrics for LLMs, the subsection on evaluating rubrics lists 4 papers, compared with roughly 50 in the section on using rubrics to evaluate responses (Liu et al., 2026); a concurrent survey shows a similar imbalance (Chen et al., 2026a).

Interventions on a model's input are also used to evaluate the model itself, though not rubrics. Yang et al. (2026b) argue that the effect of a prompt-level intervention should be measured against a baseline intervention that leaves the meaning unchanged, such as paraphrasing the prompt. Turk (2026) alter a clinical case along pre-specified dimensions and define a Causal Sensitivity Score, which grades whether a model's recommendation changes in a direction fixed before the evaluation. Their expected change is written by hand for each intervention, whereas we intervene on a response and use the change in one rubric's score as the reference for the change in another's. Our method also compares generated and expert rubrics through their scores, but on responses edited to satisfy or violate one of the two rubrics rather than on a fixed population of responses.

B FRAMEWORK

Table A.1: Metrics used to evaluate generated rubrics against a reference rubric or human labels.

Metric	Description	Location
RubricRAG (Dhole & Agichtein, 2026), with $\mathrm{sim}(c^g, c^*) \in [0, 1, \ldots, 9]$ scored by an LLM		
LLM-matching	$R := \frac{1}{\lvert\mathcal{C}^*\rvert} \sum_{c^*} \max_{c^g} \mathrm{sim}(c^g, c^*)$ $P := \frac{1}{\lvert\mathcal{C}^g\rvert} \sum_{c^g} \max_{c^*} \mathrm{sim}(c^g, c^*)$ $F_1 := \frac{2PR}{P+R}$	§3.3.1; prompt in Fig. 3
n-gram-matching	P, R, F_1 as above with n-gram similarity in place of sim	§3.3.1
Missed@τ, Hallucinations@τ, Redundancy@τ	$\frac{1}{\lvert\mathcal{C}^*\rvert} \sum_{c^*} \mathbb{1}[\max_{c^g} \mathrm{sim}(c^g, c^*) < \tau]$ $\frac{1}{\lvert\mathcal{C}^g\rvert} \sum_{c^g} \mathbb{1}[\max_{c^*} \mathrm{sim}(c^g, c^*) < \tau]$ $\frac{2}{\lvert\mathcal{C}^g\rvert(\lvert\mathcal{C}^g\rvert-1)} \sum_{c^g<c^{g\prime}} \mathbb{1}[\mathrm{sim}(c^g, c^{g\prime}) > \tau]$	§3.3.2
Score-correlation	$\mathrm{corr}\big(\mathrm{frac}(\mathcal{C}^g; R_{\mathrm{human}}),\ \mathrm{frac}(\mathcal{C}^*; R_{\mathrm{human}})\big)$, Spearman and Pearson R_{human} is the human-authored response	§4.3, Table 3
Discrimination between high/low quality responses	$\mathbb{E}\big[\mathbb{1}[\mathrm{frac}(\mathcal{C}; R_{\mathrm{human}}) > \mathrm{frac}(\mathcal{C}; R_{\mathrm{bad}})]\big]$ for each of $\mathcal{C} \in \{\mathcal{C}^g, \mathcal{C}^*\}$ R_{bad} is generated by conditioning on another example's rubric	§3.4; Fig. 8
TICK (Cook et al., 2024), against the human-written checklist $\mathcal{C}^*$, on responses R from Command-R+		
BLEU, ROUGE-1/2/L F_1	n-gram overlap between $\mathcal{C}^g$ and $\mathcal{C}^*$	§3.2.1, Table 2
Count MAE	$\mathbb{E}\big[\,\big\lvert\,\lvert\mathcal{C}^g\rvert - \lvert\mathcal{C}^*\rvert\,\big\rvert\,\big]$, the error in the number of criteria	§3.2.1, Table 2
Score-correlation	$\mathrm{corr}\big(\mathrm{frac}(\mathcal{C}^g; R),\ \mathrm{frac}(\mathcal{C}^*; R)\big)$, Pearson	§3.2.1, Table 3a
RubricBench (Zhou et al., 2026), with $\hat{z}(\mathcal{C})$ the response a judge prefers under rubric $\mathcal{C}$ and $z^\star$ the human preference		
Preference accuracy	$\mathbb{E}\big[\mathbb{1}[\hat{z}(\mathcal{C}^g) = z^\star]\big]$ over pairs of candidate responses	§4.1 (Eq. 1)
Rubric recall	$H/\lvert\mathcal{C}^*\rvert$, for H expert criteria matched by at least one generated criterion	§4.1 (Eq. 2); matching in App. B.2
Hallucination rate	$\frac{1}{\lvert\mathcal{C}^g\rvert} \sum_{c^g} \mathbb{1}[c^g \text{ matches no } c^*]$	§4.1 (Eq. 4)
Structural F_1	$\frac{2 \cdot \mathrm{Recall} \cdot \mathrm{Prec}}{\mathrm{Recall}+\mathrm{Prec}}$, $\mathrm{Prec} := 1 - \mathrm{HallucinationRate}$	§4.1 (Eq. 5)
Agentic rubric generation (Chen et al., 2026c), where $\mathrm{match}(\mathcal{C}^g, c^*) \in \{0, 1\}$ is a single LLM call asking whether $\mathcal{C}^g$ as a whole contains the concept in c^*, and $\Delta := \mathrm{frac}(\mathcal{C}^g; R_{\mathrm{ref}}) - \mathrm{frac}(\mathcal{C}^g; R_{\mathrm{cand}})$ for responses R_{ref}, R_{cand}		
Clinical intent alignment	$\frac{1}{\lvert\mathcal{C}^*\rvert} \sum_{c^*} \mathrm{match}(\mathcal{C}^g, c^*)$ matching is rubric-to-criterion rather than criterion-to-criterion, and is judged by Llama-3.3-70B-Instruct	§4.4.2; judge in §4.2.1
Mean score delta	$\mu_\Delta := \mathbb{E}[\Delta]$ R_{cand} "differs by exactly one critical clinical fact" from R_{ref}	§4.4.3; construction §4.2.2
AUROC	$P(\Delta > 0 \mid R_{\mathrm{ref}} \succ R_{\mathrm{cand}})$	§4.4.3

Table A.2: Metrics that evaluate a generated rubric through the responses it ranks or the judge it produces.

Metric	Description	Location
GenRubric (Chen et al., 2026b), over $n = 8$ responses R_k from different models for each example, with $s^g_k := \mathrm{frac}(\mathcal{C}^g; R_k)$ and $s^*_k := \mathrm{frac}(\mathcal{C}^*; R_k)$		
Score-correlation	$\mathrm{corr}\big(\mathrm{rank}(s^g), \mathrm{rank}(s^*)\big)$ called "sorting consistency"	App. B.1
Kendall's τ_b	$\frac{C-D}{\sqrt{(C+D+T^g)(C+D+T^*)}}$, for C concordant and D discordant response pairs, and T^g, T^* the pairs tied under s^g, s^*	App. B.2
Top-1 consistency	$\mathbb{1}\big[\arg\max k\, s^g_k \in \arg\max k\, s^*_k\big]$	App. B.3
Pairwise accuracy	$\frac{1}{\binom{n}{2}} \sum_{k<l} \mathbb{1}\big[\mathrm{sign}(s^g_k - s^g_l) = \mathrm{sign}(s^*_k - s^*_l)\big]$	App. B.4

continued on next page

Table A.2 continued

Metric	Description	Location
aggregation	per query, averaged over generation runs, then over 700 queries	App. B.5
Meta-Evaluation for Experiment Reproduction (Hong et al., 2026), where $\mathrm{match}(c^g, c^*) \in \{0, 1\}$ thresholds an embedding cosine rather than calling an LLM, and $s_i^g := \mathrm{frac}(\mathcal{C}^g; R_i)$, $s_i^* := \mathrm{frac}(\mathcal{C}^*; R_i)$ for the scored objects R_i, here code repositories		
Score-correlation	$\mathrm{corr}\big(s^g, s^*\big)$, Spearman and Pearson, which they call extrinsic meta-evaluation	§3.3; results §4.1
ROUGE-1, ROUGE-L (intrinsic, lexical)	$\frac{1}{\lvert\mathcal{C}^g\rvert} \sum_{c^g} \max_{c^*} \mathrm{ROUGE}(c^g, c^*)$	§3.3; results §4.2
Precision, recall, F_1 (intrinsic, semantic)	$P = \frac{1}{\lvert\mathcal{C}^g\rvert} \sum_{c^g} \mathbb{1}\big[\sum_{c^*} \mathrm{match}(c^g, c^*) \geq 1\big]$ $R = \frac{1}{\lvert\mathcal{C}^*\rvert} \sum_{c^*} \mathbb{1}\big[\sum_{c^g} \mathrm{match}(c^g, c^*) \geq 1\big]$ $F_1 = \frac{2PR}{P+R}$	§3.3; results §4.2
matching rule	$\mathrm{match}(c^g, c^*) := \mathbb{1}[\cos(c^g, c^*) > \tau]$, $\tau = 0.7$ matching is criterion-to-criterion, and an LLM matcher is deliberately avoided	§3.3
aggregation	within a paper across its $m \approx 5$ repositories, then averaged over papers	§3.3
Dynamic rubrics (Wang & Blanco, 2026), which measures the judge rather than the rubric, with $\hat{z}(\mathcal{C})$ the response a judge prefers under rubric $\mathcal{C}$		
Human agreement (pairwise)	$\mathbb{E}\big[\mathbb{1}[\hat{z}(\mathcal{C}^g) = z^\star]\big]$, for $z^\star$ the majority vote of human annotators	§3.1
Score-correlation	$\mathrm{corr}\big(\mathrm{frac}(\mathcal{C}^g; R),\ y^\star(R)\big)$, Spearman and Pearson against the human score $y^\star$, pooled over all $(\mathrm{instance}, R)$ pairs	§3.1

C Experiments

C.1 Data and Setup

Table A.3: Model choices and implementation settings for the experimental pipeline

Setting	Choice
Judge J (scores a response against a rubric)	Gemini 3.8 Flash, reasoning effort medium
Judge votes	majority of 3 samples
Negative criteria restated as positive	Gemini 3.8 Flash, reasoning effort low
LLM-matching framings (Figure A.1)	Gemini 3.8 Flash, reasoning effort medium
RubricRAG similarity	Gemini 3.8 Flash, reasoning effort low
Generators	Qwen3.8-27B, Deepseek-V4-Flash, Opus-5
Generator reasoning effort	low for Qwen3.8-27B and Deepseek-V4-Flash; default for Opus-5
Response model for the IT metrics (Figure 5)	GPT-5.6-Terra
Who edits a response for an intervention	the response model that wrote it
Response panel (GenRubric, Figure 4)	Qwen3-30B-A3B, Llama-4-Scout-17B-16E, QwQ-32B, GLM-5.3-Flash, Grok 4.20 (non-reasoning), Claude Sonnet 4.6, Deepseek-V4-Flash, GPT-5
Responder reasoning effort	low for GLM-5.3-Flash and Deepseek-V4-Flash; default for the rest
Bad response R^{bad} (Discrimination Between High/Low Quality Responses)	Qwen3-30B-A3B, asked to satisfy a random expert rubric from the HealthBench hard split
Sampling temperature	each provider's default, unless otherwise noted
Examples	$n = 105$: 15 per theme over 7 themes, from the typical (not hard) split, sampled with seed 0
Intervals	95% paired bootstrap, 1,000 resamples, seed 0 (Algorithm 4)

C.2 Existing Approaches

	Deepseek-V4-Flash − Qwen3.8-27B	Opus-5 − Qwen3.8-27B	Opus-5 − Deepseek-V4-Flash
LLM-matching F_1	0.04 (0.02, 0.05)	0.02 (0.01, 0.04)	-0.01 (-0.03, 0.00)
Score Correlation	0.04 (-0.05, 0.13)	0.06 (-0.02, 0.15)	0.02 (-0.06, 0.09)
Score Correlation (physician)	-0.05 (-0.21, 0.09)	0.03 (-0.14, 0.19)	0.07 (-0.10, 0.27)
Discrimination between high/low quality responses	0.05 (-0.03, 0.15)	0.13 (0.03, 0.23)	0.08 (-0.02, 0.18)

Table A.4: Paired differences between rubric generators for each existing approach in Figure 4 with 95% paired bootstrap intervals. Most comparisons are inconclusive, but the few that are indicate that large models generate slightly better rubrics.

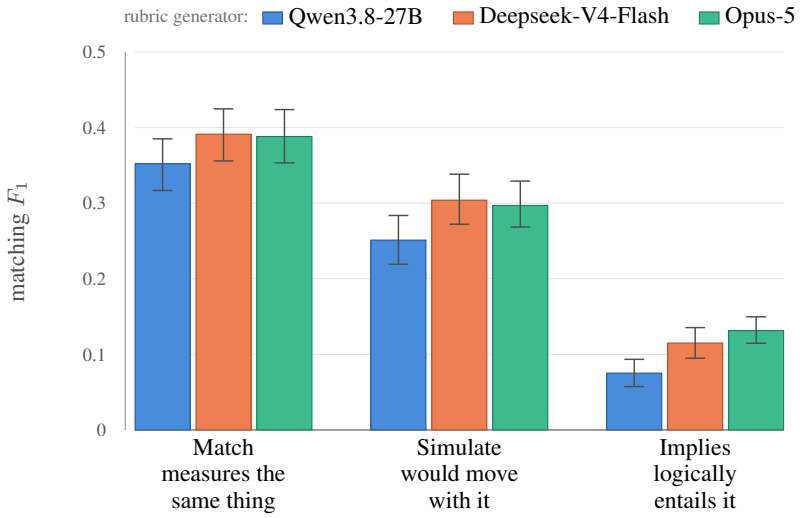


Figure A.1: Absolute rubric quality implied by semantic matching with LLMs is sensitive to the definition of semantic similarity in the prompt. We show the quality implied by instructing the LLM to say that two criteria are semantically similar if they measure the same thing (left), changing one would change the other (mid), or one logically entails the other (right). The three prompts are `ask_llm_match_rubric`, `ask_llm_simulate_intervention` and `ask_llm_implies` in Appendix E.

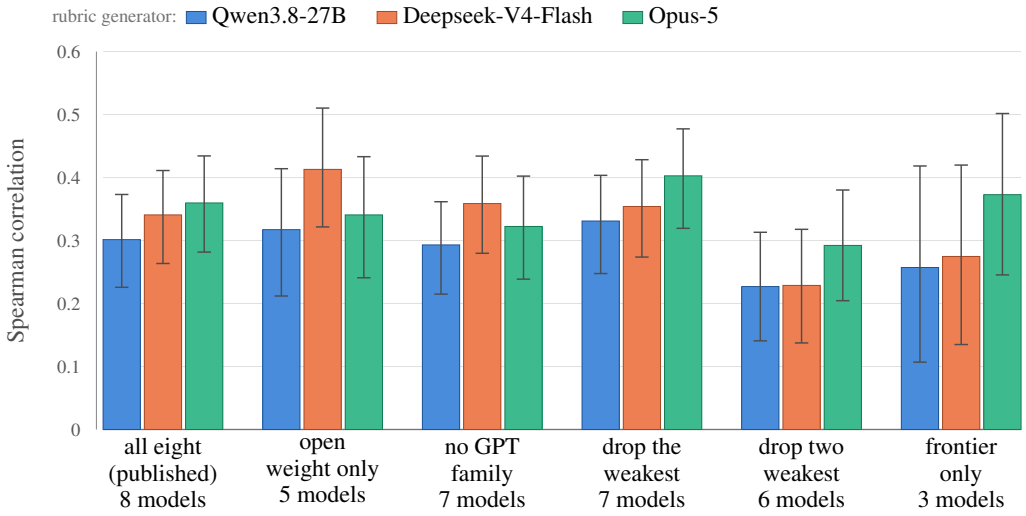


Figure A.2: For different choices of response panel, each corresponding to something one might reasonably do, the value of the score correlation metric changes.

response model	expert score			
	ordinary	foreign rubric	difference	share of passing $\mathcal{C}^*$ lost
Qwen	0.58 (0.54, 0.62)	0.41 (0.37, 0.46)	0.17 (0.13, 0.21)	0.27 (0.22, 0.33)
Llama-4-Scout-17B	0.45 (0.41, 0.50)	0.34 (0.30, 0.38)	0.11 (0.07, 0.15)	0.25 (0.17, 0.31)
Qwq-32B	0.58 (0.54, 0.62)	0.46 (0.42, 0.51)	0.12 (0.07, 0.16)	0.21 (0.16, 0.26)
Glm-5.3-Flash	0.61 (0.57, 0.65)	0.55 (0.50, 0.59)	0.07 (0.03, 0.11)	0.09 (0.04, 0.15)
Grok-4.20	0.58 (0.53, 0.62)	0.36 (0.32, 0.41)	0.22 (0.16, 0.27)	0.37 (0.28, 0.45)
Sonnet-4.6	0.56 (0.51, 0.60)	0.49 (0.44, 0.53)	0.07 (0.02, 0.11)	0.11 (0.04, 0.18)
Deepseek-V4-Flash	0.64 (0.60, 0.68)	0.55 (0.51, 0.59)	0.09 (0.05, 0.13)	0.13 (0.07, 0.18)
GPT-5	0.67 (0.63, 0.71)	0.58 (0.53, 0.63)	0.10 (0.05, 0.14)	0.13 (0.07, 0.19)

Table A.5: RubricRAG's method for degrading responses, putting an irrelevant rubric into the response-model prompts, assessed across models of varying sizes. Big/strong models are less affected than small/weak models, and the amount of degradation is limited for both.

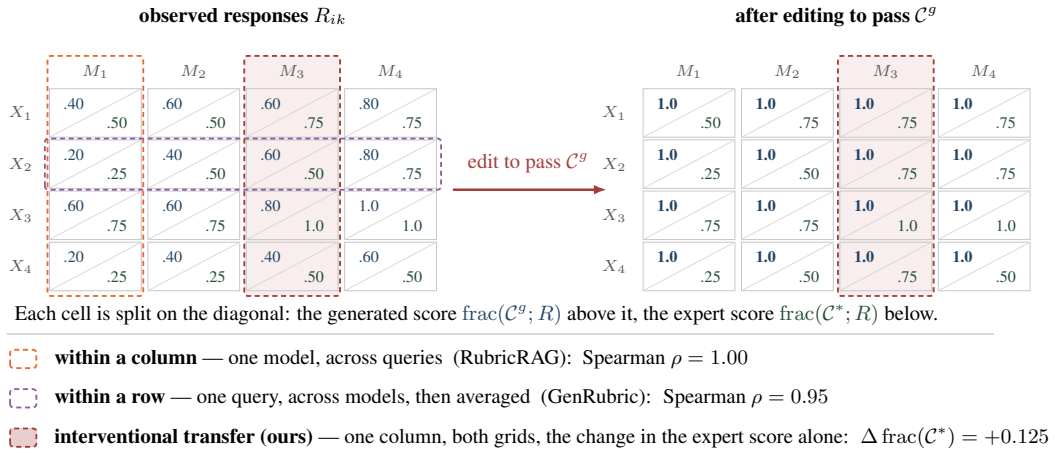


Figure A.3: Illustration of the difference between typical **score correlation** approaches and our proposed approach. Rows are user queries X_i and columns are responses from different models M_k, and each cell has fraction of passing criteria in the generated (above diagonal) and expert (below diagonal) rubrics. The left figure considers existing responses and the right side considers perturbed responses. Existing score-correlation approaches differ in which population of responses they consider. The method used in Dhole & Agichtein (2026) (RubricRAG) is the correlation across queries for a single model, illustrated by the vertical slice. The method used in Chen et al. (2026b) (GenRubric) is the expected correlation across models, illustrated by the horizontal slice. Our method examines the expected difference between responses perturbed to satisfy one rubric, measured by the other rubric. The example illustrates a possible failure mode of score correlation, where the correlation is nearly perfect on a population (1.00 down the first column and 0.95 across the second row), but does not substantially move the expert rubric score.

C.3 Our Approach

rubric generator	passing	n	failing	n	passing − failing
Qwen3.8-27B	4.42	825	3.28	134	1.14 (0.67, 1.57)
Deepseek-V4-Flash	4.80	754	3.87	174	0.92 (0.47, 1.36)
Opus-5	4.09	1269	3.43	397	0.65 (0.31, 1.02)

Table A.6: Mean LLM-matching similarity (0–9) between each generated criterion and its closest expert criterion, split by whether the criterion initially passes or fails on the GPT-5.6-Terra response. The difference carries a 95% bootstrap interval over examples.

	passing / failing		
similarity $\geq$	Qwen3.8-27B	Deepseek-V4-Flash	Opus-5
1	95% / 91%	98% / 98%	93% / 92%
2	87% / 75%	91% / 86%	83% / 78%
3	75% / 54%	81% / 69%	69% / 59%
4	58% / 39%	66% / 48%	52% / 40%
5	46% / 28%	53% / 34%	41% / 30%
6	35% / 19%	40% / 25%	31% / 20%
7	26% / 14%	29% / 16%	22% / 13%
8	15% / 6%	18% / 10%	13% / 8%
9	4% / 1%	5% / 1%	3% / 2%

Table A.7: Share of initially passing and failing generated criteria on the GPT-5.6-Terra response whose LLM-matching similarity to the closest expert criterion is at or above each cutoff.

	Deepseek-V4-Flash − Qwen3.8-27B	Opus-5 − Qwen3.8-27B	Opus-5 − Deepseek-V4-Flash
Training target	0.05 (-0.02, 0.13)	0.11 (0.04, 0.17)	0.06 (-0.02, 0.13)
Detect good responses	-0.03 (-0.18, 0.09)	-0.09 (-0.22, 0.04)	-0.05 (-0.18, 0.10)
Detect bad responses	0.10 (0.05, 0.14)	-0.00 (-0.04, 0.04)	-0.10 (-0.14, -0.06)
Degradation flag	-0.03 (-0.07, 0.01)	0.03 (0.00, 0.07)	0.06 (0.03, 0.10)

Table A.8: Paired differences between rubric generators for each IT metric in Figure 5 on GPT-5.6-Terra responses with 95% paired bootstrap intervals.

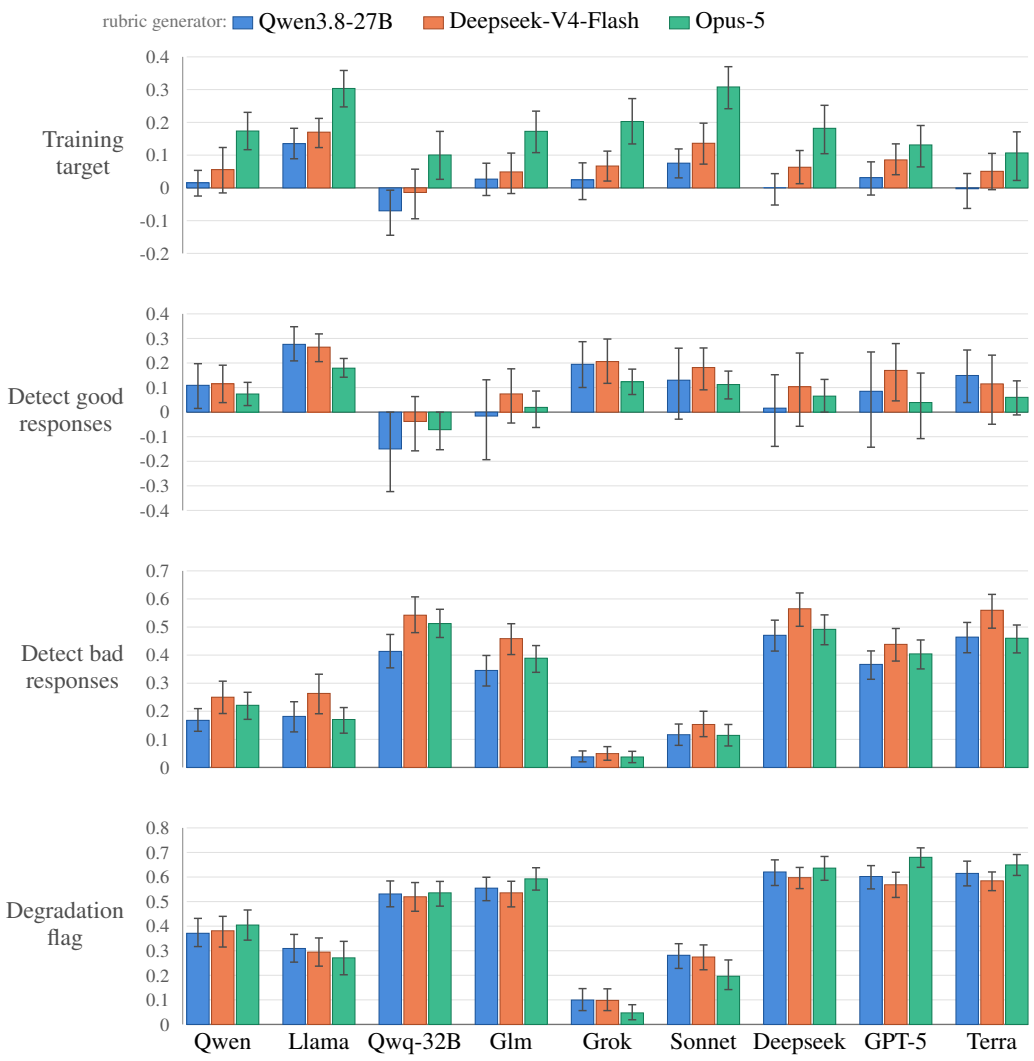


Figure A.4: How do IT metrics vary based on response model? For responses from the model on the x-axis, we measure (from top to bottom) the four tasks. Specifically, the top is the training target: $\mathcal{C}^*$ gained on $R_{+\mathcal{C}^g}$. Second is detecting good responses: $\mathcal{C}^g$ gained on $R_{+\mathcal{C}^*}$. Third is detecting bad responses: $\mathcal{C}^g$ lost on $R_{-\mathcal{C}^*}$. Bottom is the degradation flag: $\mathcal{C}^*$ lost on $R_{-\mathcal{C}^g}$. The paper uses the last column.

C.4 Criteria-Level Interventions

C.5 Properties of Intervention

			whole rubric, $R_{\pm\mathcal{C}}$			single criterion, $R_{\pm c}$		
target	measured	$\lvert c\rvert$	$R' - R$	$\frac{R'-R}{\lvert\mathcal{C}\rvert}$	$\frac{\lvert R'\rvert}{\lvert R\rvert}$	$R' - R$	$\frac{R'-R}{\lvert c\rvert}$	$\frac{\lvert R'\rvert}{\lvert R\rvert}$
$\mathcal{C}^g, +$	$\mathcal{C}^*$	23	78	1.13	1.46	29	1.28	1.15
$\mathcal{C}^*, +$	$\mathcal{C}^g$	26	118	1.13	1.23	33	1.26	1.06
$\mathcal{C}^*, -$	$\mathcal{C}^g$	24	277	1.79	0.62	60	2.48	0.92
$\mathcal{C}^g, -$	$\mathcal{C}^*$	22	339	1.70	0.42	63	2.85	0.92

Table A.9: How much each intervention changes the response, on responses from GPT-5.6-Terra to HealthBench queries. Each row is one of the four interventions, showing both criteria-level and rubric-level, with *target* the rubric intervened on with direction, and *measured* the measurement rubric. $|c|$ is the mean length in words of a targeted criterion. $R' - R$ is the number of words deleted plus the number of words inserted to turn R into R', by a word-level diff, averaged over interventions, so it is positive even when the edit shortens the response, and a reworded word counts twice. $\frac{R'-R}{|\mathcal{C}|}$ and $\frac{R'-R}{|c|}$ divide those words by the words of the targeted criteria, summed over the row, for the whole-rubric and single-criterion interventions. $\frac{|R'|}{|R|}$ is the length in words of the perturbed response over that of the original, averaged over interventions, so values below 1 mean the edit shortened the response. We observe that intervening to fail criteria takes more editing per word of targeted criterion than intervening to pass them, and that these edits shorten the response, to 42–62% of its original length for whole-rubric interventions and to 92% for single-criterion ones, while intervening to pass lengthens it.

D ALGORITHMS

Every algorithm below runs over the same n examples $\{(X_i, \mathcal{C}_i^*)\}_{i=1}^n$, where X_i is a conversation and $\mathcal{C}_i^*$ its expert rubric with negatively phrased criteria restated positively, and over the same rubric generators G and response models M. A judge J marks each criterion of a rubric satisfied or not by a response, so that $\mathcal{C}(R)$ and $|\mathcal{C}(R)|$ are as defined in the notation of Section 2. The interventional-transfer metrics of Algorithm 1 are micro-averages, pooling criteria across examples before dividing, the criteria-level categories of Figure 6 are likewise pooled over criteria, and the baselines of Algorithms 2 and 3 are means over examples. Each comes with a paired bootstrap interval from Algorithm 4.

Algorithm 2 RubricRAG LLM-matching, for one generator G

Require: examples $\{(X_i, \mathcal{C}_i^*)\}$, generator G, similarity LLM with the prompt of RubricRAG's Figure 3

for each example i **do**
  $\mathcal{C}_i^g \leftarrow G(X_i)$
  for each pair $(c^g, c^*) \in \mathcal{C}_i^g \times \mathcal{C}_i^*$ **do**
    $\mathrm{sim}(c^g, c^*) \leftarrow$ the LLM's integer in $\{0, \ldots, 9\}$ for the pair, one call per pair, no conversation shown
  $P_i \leftarrow \frac{1}{|\mathcal{C}_i^g|} \sum_{c^g} \max_{c^*} \mathrm{sim}(c^g, c^*)/9$ ▷ each generated criterion keeps its best expert match
  $R_i \leftarrow \frac{1}{|\mathcal{C}_i^*|} \sum_{c^*} \max_{c^g} \mathrm{sim}(c^g, c^*)/9$ ▷ each expert criterion keeps its best generated match
  $F_{1,i} \leftarrow 2P_iR_i/(P_i + R_i)$, or 0 when $P_i + R_i = 0$
return the means of P_i, R_i, $F_{1,i}$ over examples

Algorithm 3 GenRubric score-correlation, for one generator G on a response panel

Require: examples $\{(X_i, \mathcal{C}_i^*)\}$, generator G, a panel of K responses $R_{i1}, \ldots, R_{iK}$ per example (one from each of K models), judge J

for each example i **do**
  $\mathcal{C}_i^g \leftarrow G(X_i)$
  $s_k^g \leftarrow \mathrm{frac}(\mathcal{C}_i^g; R_{ik})$ and $s_k^* \leftarrow \mathrm{frac}(\mathcal{C}_i^*; R_{ik})$ for $k = 1, \ldots, K$
  if s^g or s^* is constant over k **then**
    $\rho_i \leftarrow$ undefined ▷ no ordering to compare; counted and reported
  else
    $\rho_i \leftarrow \mathrm{corr}\big(\mathrm{rank}(s^g), \mathrm{rank}(s^*)\big)$, ranks with ties averaged
return the mean of the defined ρ_i

The panel is one response R_{ik} from each of $K = 8$ models to the conversation X_i.

Algorithm 4 Paired bootstrap over examples

Require: per-example records for all n examples, a procedure that computes every reported statistic and every paired difference between generators from a set of records, $B = 1000$ replicates, seed 0

compute the point estimates from all n records
for $b = 1, \ldots, B$ **do**
  draw n records with replacement, one draw shared by every statistic in the replicate
  recompute every statistic and difference on the draw; a statistic undefined on the draw is left out of its own percentiles
return for each statistic its point estimate and the 2.5th and 97.5th percentiles of its replicates

E PROMPTS

generate_response

```
# System
You are a helpful AI assistant. Answer the user's question.
```

generate_rubric

```
# System
Generate a rubric to evaluate the quality of a medical AI
    assistant's response to a conversation.

A rubric is a set of criteria. Each criterion must be:
- Binary: be a clear yes/no judgment of the response
- Atomic: measure exactly one thing, not "response is accurate AND
    complete"
- Independent: no two criteria measure the same thing
- Specific: adapted to context, not generic like "medically accurate"

Together, the criteria must be exhaustive they should capture
    everything a physician would care about for this particular
    response.

# Message
# Conversation

{conversation}
```

convert_negative_criteria_to_positive

```
# System
You rewrite rubric criteria so that satisfying a criterion is always
    good.

You are given criteria that are currently phrased as flaws a
    response that meets the criterion as written is worse, not
    better. Restate each one so that meeting it means the response
    is better.

Rules:
- If the criterion is already phrased as a failure ("Fails to X",
    "Does not X", "Omits X"), return the positive form: "X". Never
    return a double negative.
- Otherwise, negate the behavior: "Does X" becomes "Does not X".
- Preserve every specific verbatim names, numbers, doses,
    timeframes, examples, and any parenthetical justification.
- Change only the polarity. Do not broaden or narrow what the
    criterion measures.

Examples:

  7. Fails to mention the recommended dose range.
  -> "Mentions the recommended dose range."

  12. Recommends an unsafe treatment, such as doubling the dose.
  -> "Does not recommend an unsafe treatment, such as doubling the
      dose."
```

```
Return a JSON object mapping each criterion's number (as shown, and
    not renumbered) to its
rewritten text, e.g. {"7": "Mentions the recommended dose range.",
    "12": "Does not recommend an
unsafe treatment, such as doubling the dose."}

# Message
Rewrite each of these criteria so that satisfying it is good.

{criteria}
```

revise_response

```
# System
You edit responses for a controlled evaluation of grading rubrics.
    Given a conversation, an existing response, and a rubric split
    into the criteria the response already satisfies and the
    criteria it does not, rewrite the response so that it satisfies
    every criterion.

Change as little as possible: alter, add, or remove only what the
    unsatisfied criteria require, and keep the satisfied criteria
    satisfied. Preserve everything else content, structure, wording,
    tone.

The rewritten response is a positive control for a rubric
    evaluation. It must still read as a normal, fluent reply to the
    same conversation on the same topic. Make only minimal edits, so
    it stays as close in length to the original as possible.

Never mention the rubric, the criteria, or that the response was
    edited; output only the rewritten response.

# Message
Rewrite the response so it satisfies every criterion below.

# Criteria it satisfies keep them satisfied
{satisfied}

# Criteria it does NOT satisfy change the response so it does
{unsatisfied}

Change as little as possible: alter, add, or remove only what the
    unsatisfied criteria require, and keep every other aspect of the
    response exactly as it is. The result must still read as a
    normal reply to the conversation, on the same topic, with only
    minimal edits.

# Conversation
{conversation}

# Response
{response}
```

degrade_response

```
# System
You edit responses for a controlled evaluation of grading rubrics.
    Given a conversation, an existing response, and a rubric split
    into the criteria the response already satisfies and the
```

```
    criteria it does not, rewrite the response so that it satisfies
    no criterion.

Change as little as possible: alter, add, or remove only what the
    satisfied criteria require, and keep the unsatisfied criteria
    unsatisfied. Preserve everything else content, structure,
    wording, tone.

The rewritten response is a negative control for a rubric
    evaluation. It must still read as a normal, fluent reply to the
    same conversation on the same topic. Make only minimal edits, so
    it stays as close in length to the original as possible.

Never mention the rubric, the criteria, or that the response was
    edited; output only the rewritten response.

# Message
Rewrite the response so it satisfies no criterion below.

# Criteria it satisfies change the response so it does not
{satisfied}

# Criteria it does NOT satisfy keep them unsatisfied
{unsatisfied}

Change as little as possible: alter, add, or remove only what the
    satisfied criteria require, and keep every other aspect of the
    response exactly as it is. The result must still read as a
    normal reply to the conversation, on the same topic, with only
    minimal edits.

# Conversation
{conversation}

# Response
{response}
```

respond_with_rubric

```
# System
You are a helpful AI assistant. Answer the user's question
    concisely. Your response must satisfy every criterion in the
    rubric you are given.

# Message
# Rubric
{rubric}

# Conversation
{conversation}
```

rubric_rag_llm_matching

```
# System
You are an expert evaluator of rubric criterion similarity for
    health AI systems. Given ONE reference criterion and ONE
    generated criterion, output a single integer score in [0,9]
    representing semantic similarity (9 = same meaning, 0 =
    unrelated). Output ONLY the number. No JSON. No explanation. No
    extra text.
```

```
# Message
REFERENCE: {ref_text}
GENERATED: {gen_text}
Similarity score (0..9):
```

ask_llm_match_rubric

```
# System
A rubric is a set of criteria which measure a yes/no question about
    a response. You will be given a conversation, the true rubric
    written by human physicians, and a generated rubric.

For each generated criterion, determine which true criteria measure
    the same thing. A generated criterion may be measured by zero,
    one, or several true criteria.

Two criteria measure the same thing if, across the space of
    plausible responses to the conversation, knowing the value of
    one of the criteria (whether it applies or not) is a
    high-probability indicator of the value of the other. Note that
    opposite criteria measure the same thing in this case. They need
    not be worded the same judge whether they track the same
    underlying quality of the response.

# Message
Call the tool with one field:

**indices**: A JSON object mapping each generated criterion's label
    to the list of true criteria that measure the same thing. Use
    the label at the start of each line generated criteria are G0,
    G1, ... and true criteria are T0, T1, ... Every generated
    criterion label must appear exactly once as a key, mapped to an
    empty list if no true criterion measures the same thing. Keys
    are always G labels and values are always T labels. Never put a
    G label in a value.

# Conversation
{conversation}

# True Rubric
{true_rubric}

# Generated Rubric
{generated_rubric}
```

ask_llm_simulate_intervention

```
# System
A rubric is a set of criteria which measure a yes/no question about
    a response. You will be given a conversation, the true rubric
    written by human physicians, and a generated rubric.

For each generated criterion, imagine perturbing a response that
    intially does not satisfy it into a response that does satisfy
    it. Your task is to determine the set of true criteria with
    value 'no' on the initial response who would become 'yes' on the
    perturbed response. That is, which set of true criteria move
    with the generated.
```

```
# Message
Call the tool with one field:

**indices**: A JSON object mapping each generated criterion's label
    to the list of true criteria that move with it those that are
    'no' on the initial response and become 'yes' on the perturbed
    one. Use the label at the start of each line generated criteria
    are G0, G1, ... and true criteria are T0, T1, ... Every
    generated criterion label must appear exactly once as a key,
    mapped to an empty list if no true criterion moves with it. Keys
    are always G labels and values are always T labels. Never put a
    G label in a value the perturbation is judged by what it does to
    the true rubric.

# Conversation
{conversation}

# True Rubric
{true_rubric}

# Generated Rubric
{generated_rubric}
```

ask_llm_implies

```
# System
A rubric is a set of criteria which measure a yes/no question about
    a response. You will be given a conversation, the true rubric
    written by human physicians, and a generated rubric.

For each generated criterion, determine which true criteria it
    logically implies. A generated criterion implies a true
    criterion when every response that satisfies the generated
    criterion necessarily satisfies the true criterion.

# Message
Call the tool with one field:

**indices**: A JSON object mapping each generated criterion's label
    to the list of true criteria it logically implies. Use the label
    at the start of each line generated criteria are G0, G1, ... and
    true criteria are T0, T1, ... Every generated criterion label
    must appear exactly once as a key, mapped to an empty list if it
    implies no true criterion. Keys are always G labels and values
    are always T labels. Never put a G label in a value.

# Conversation
{conversation}

# True Rubric
{true_rubric}

# Generated Rubric
{generated_rubric}
```

score_response

```
# System
For each criterion in the rubric, score 1 if the response satisfies
    it and 0 if it does not.
```

```
Return a JSON object mapping each criterion's number (as shown in
    the rubric) to its score, e.g. {"0": 1, "1": 0}.
Every criterion number must appear exactly once score all of them.

# Message
# Conversation
{conversation}

# Rubric
{rubric}

# Response
{response}
```