# RUBRIC OPTIMIZATION FOR LLM-BASED AUTOMATED GRADING WITH DATA SYNTHESIS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large language models (LLMs) have recently emerged as a promising approach for automatic grading (AG) due to their strong instruction-following and reasoning abilities. Recent work has explored automatic rubric optimization using labeled student responses to address rubric quality issues. However, its effectiveness is often limited by the scarcity of annotations. In this work, we propose a new optimization paradigm, RODS, which leverages the generative capabilities of LLMs to synthesize responses as supplementary data, enabling existing rubric optimization methods to perform robustly under limited-label settings. To mitigate noise in synthetic labels, we design a consistency-based optimization mechanism. We further introduce a generative adversarial-based quality control (GAQC) module to improve the fidelity of synthetic samples, reducing distribution shift and enhancing alignment with real student responses. In addition, we integrate both synthetic and scarce human-labeled data within an iterative optimization framework, ensuring that the optimization process remains grounded in improving grading performance on real responses. Experiments on multiple AG datasets demonstrate that RODS consistently improves rubric quality and grading accuracy, highlighting its effectiveness and practical potential for real-world LLM-based AG scenarios. Code is available at `https://anonymous.4open.science/r/RODS_ICLR-CD9B/`.

## 1 INTRODUCTION

Automatic grading (AG) has long been recognized as a critical research problem in applying artificial intelligence (AI) to educational settings (Wang et al., 2026). By reducing the substantial time and effort required for manual grading, AG methods enable educators to focus on designing more challenging and open-ended questions that promote creativity and deeper learning within curriculum design (Lu et al., 2025). Moreover, the objectivity and efficiency of AG methods facilitate timely and consistent feedback, which is essential for improving student learning outcomes (Bayerlein, 2014). To implement AG, prior work has extensively explored machine learning and deep learning approaches (Ramalingam et al., 2018; Bonthu et al., 2021). These methods typically follow a supervised learning paradigm, where models are trained on datasets of student responses paired with human-assigned scores, and the trained models are then used to grade new responses (Galhardi & Brancher, 2018). While these approaches have demonstrated strong performance on question-specific tasks, they suffer from a fundamental limitation: the indirect and inefficient communication between educators and models. Specifically, grading criteria are learned implicitly from labeled data, preventing teachers from directly specifying grading instructions. In addition, these methods require substantial labeled data for each new question to achieve satisfactory performance, which limits their scalability in real-world educational settings (Hernández-Blanco et al., 2019).

The recent emergence of LLMs has introduced a promising new paradigm for automatic grading. Leveraging their extensive prior knowledge and strong instruction-following capabilities, LLM-based approaches can perform grading by directly interpreting textual rubrics (Schneider et al., 2023). With appropriate rubric design, these methods can achieve performance on par with human graders without task-specific training (Chu et al., 2024). Moreover, the use of human-readable rubrics allows educators to explicitly define and modify grading criteria, enabling more transparent and controllable grading behaviors (Chu et al., 2026b). Despite these advantages, the effectiveness of LLM-based AG heavily relies on the quality of the grading rubric (Xie et al., 2025b). Designing effective rubrics remains labor-intensive, as teachers must repeatedly analyze misaligned model outputs and manually



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.

revise criteria to correct unexpected errors. This challenge is further exacerbated by the sensitivity of LLMs to instruction phrasing and structure, which can lead to discrepancies between rubric intent and model interpretation (Chu et al., 2024). To address this issue, recent work has explored automatic rubric optimization methods that leverage LLM reasoning and self-reflection capabilities (Pryzant et al., 2023). These approaches iteratively refine rubrics by identifying misgraded samples in labeled datasets and using them as feedback signals. Combined with self-reflection mechanisms that diagnose failure modes, LLMs are prompted to generate improved rubrics. While promising, these methods assume access to sufficiently large labeled datasets (e.g., over 100 samples per question in (Chu et al., 2024)). This requirement limits their practicality in real-world educational settings, where collecting such annotations for each question remains costly and labor-intensive (Ferreira Mello et al., 2025).

To address the above challenges, we leverage the generative capabilities of LLMs. By conditioning on the grading rubric and a target score level, the model can be prompted to generate responses that align with the specified score. These conditioning scores are then used as labels for the generated responses, allowing them to serve as substitutes for human-labeled samples in existing rubric optimization methods. This approach reduces reliance on human-scored data in the optimization process. However, replacing real data with synthetic data introduces several challenges. First, synthetic samples may contain label noise, and directly applying existing optimization methods to such data could lead to performance degradation. Second, synthetic responses may exhibit distribution shift, as they often fail to fully capture the linguistic characteristics of real student responses, thereby harming rubric generalization. To overcome these challenges, we propose a novel rubric optimization framework, RODS, which introduces a consistency-based supervision signal that does not depend on the correctness of synthetic labels. We further develop a generative adversarial-based quality control (GAQC) mechanism to filter and refine synthetic samples, mitigating potential distribution shift and improving robustness. We also design an iterative optimization framework that integrates supervision signals from both synthetic data and limited human-labeled data, ensuring alignment with the goal of improving grading performance on real student responses. Finally, we extend RODS to a semi-supervised setting, enabling it to utilize large amounts of unlabeled student data that are underutilized by prior work.

We evaluate RODS on more than 1,700 real student responses drawn from three AG datasets, covering 12 questions across diverse topics and educational levels. RODS consistently outperforms the strongest baselines, achieving improvements of 2% to 7% across multiple evaluation metrics with two backbone models. These results demonstrate the effectiveness of synthetic-data-driven rubric optimization in overcoming the limitations of existing RO methods under limited labeled-data settings. Comprehensive ablation studies further validate the contribution of each key component and illustrate how these components work together to improve rubric quality and generalizability.

## 2 PRELIMINARY

In general, most existing LLM-based AG methods can be formulated as follows. Given a student response $x_i$ to a question $q$, the AG method concatenates the question, the response, and the grading rubric $\phi$, along with a grading instruction prompt $\pi_g$ that specifies the task and output format. This combined input is then fed into a large language model $f_\theta$, parameterized by $\theta$, to produce a grading response. By parsing the output, the predicted score $\hat{y}_i$ is extracted and assigned to $x_i$. The overall grading process can be expressed as:

$$\hat{y}_i = f_\theta(\pi_g, \phi, q, x_i) \tag{1}$$

In most cases, score levels are discrete, and AG is therefore formulated as a classification problem, where labels belong to a finite set $y_i \in \mathcal{C} = \{c_k \mid k = 1, \ldots, K\}$, with $K$ denoting the number of score levels. The objective is to maximize agreement with human annotations, typically measured by metrics such as Accuracy and Cohen's Kappa (McHugh, 2012). Notably, unlike traditional machine learning or deep learning approaches, optimization in LLM-based AG focuses not on the model parameters $\theta$, but on the grading rubric $\phi$. This design choice is motivated by two main considerations. First, labeled data in AG tasks is often limited, and directly fine-tuning $\theta$ can lead to overfitting and catastrophic forgetting (Luo et al., 2025), thereby degrading generalization. Second, tuning $\theta$ incurs substantial computational cost and lacks transparency, as optimization signals are learned indirectly from labels, preventing direct human control over grading criteria. In contrast, optimizing the rubric $\phi$ has become a widely adopted approach in LLM-based AG (Chu et al., 2025b). The rubric can be viewed as a text-based parameter of the grading method, where refinement is both

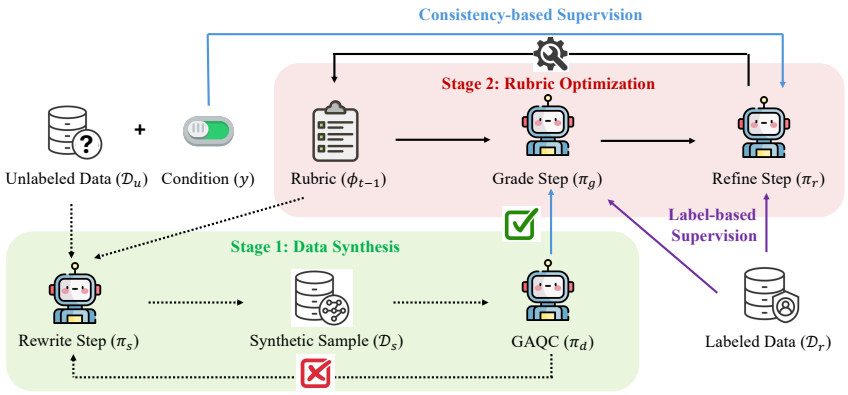



Figure 1: An Overview of the RODS Framework.

human-interpretable and significantly lower-risk in terms of computational cost and performance stability. The rubric optimization process is formulated as:

$$\phi^* = \arg\max_{\Phi} \ \mathrm{Eval}(\hat{\mathcal{Y}}, \mathcal{Y}), \quad \Phi = g(\phi_0) \tag{2}$$

where $\phi_0$ is the initial human-designed rubric, and $\mathrm{Eval}(\cdot)$ measures performance over $N$ responses using the predicted score set $\hat{\mathcal{Y}} = \{\hat{y}_i\}_{i=1}^N$ and ground-truth scores $\mathcal{Y} = \{y_i\}_{i=1}^N$. The candidate rubric space $\Phi$ is generated by an optimization strategy $g(\cdot)$, which may involve heuristic refinement by domain experts or data-driven automatic methods (Chu et al., 2026b). However, Eq. 2 reveals a key limitation: rubric optimization relies on labeled data $(x_i, y_i) \in \mathcal{D}_r$. In practice, such labeled datasets are often scarce (e.g., fewer than 20 samples per question), making the evaluation signal noisy and prone to overfitting, which can lead to suboptimal performance. To address this limitation, we propose an alternative supervision signal that does not rely on ground-truth labels. By reducing dependence on labeled data, our approach enables more effective rubric optimization in limited labeled settings, improving the practicality of LLM-based AG in real-world applications.

## 3 METHOD

A straightforward way to reduce reliance on labeled data is to leverage LLMs to generate synthetic responses conditioned on target scores, which then serve as labels for optimization. This approach enables efficient construction of large datasets for existing rubric optimization methods without additional human annotations, reducing deployment costs. However, it introduces two challenges. First, synthetic labels are inherited from generation conditions rather than human judgments, so responses may not align with their assigned scores, introducing optimization noise (Guo & Chen, 2024). Second, synthetic responses may differ from real student responses, causing distribution shift that degrades performance and generalization (Shumailov et al., 2023). Our ablation study in § 4.3 provides empirical evidence of both issues.

To address these challenges, we propose RODS, a framework comprising data synthesis and rubric optimization, as illustrated in Fig. 1. During data synthesis, RODS uses label-conditioned rewriting to adapt real student responses to target score levels rather than generating responses from scratch. This preserves the linguistic characteristics of authentic responses and mitigates distribution shift. We further introduce a generative adversarial quality control (GAQC) mechanism to assess and filter synthetic samples, retaining those judged indistinguishable from real student responses for downstream optimization. During rubric optimization, RODS replaces exclusive reliance on conditioning labels with bidirectional supervision that promotes consistency between these labels and model predictions. Besides aligning predictions with conditioning labels, this mechanism allows predicted scores to provide alternative supervision, improving robustness to synthetic label noise. Thus, optimization draws on both the generation conditions and the model's interpretation of the resulting responses, rather than assuming that every synthetic label is correct. Finally, a small set of human-labeled responses validates and guides optimization toward better grading performance on real student data. These stages jointly address synthetic-data quality and supervision reliability while limiting annotation requirements. The following subsections describe each stage in detail.

### 3.1 DATA SYNTHESIS

**Label-conditioned Generation.** To construct the synthetic dataset $\mathcal{D}_s$, we leverage the grading rubric and target labels to guide the generation process. Specifically, given a desired score $y$, we prompt the model to generate a response $\hat{x}_y$ that aligns with the corresponding scoring criteria. We adopt a rewrite-based generation strategy: Instead of synthesizing responses entirely from scratch, we condition the model on real student responses $x$ and instruct it to rewrite them so that they match the target label $y$. This approach preserves key characteristics of authentic student responses and helps reduce distribution shift. Formally, the construction of the synthetic dataset $\mathcal{D}_s$ is defined as:

$$\mathcal{D}_s = \{(\hat{x}_y, y) \mid \hat{x}_y = f_\theta(\pi_s, \phi, x, y),\ x \in \mathcal{D}_r,\ y \in \mathcal{C}\}, \tag{3}$$

where $\mathcal{D}_r$ denotes the set of real student responses, and $\mathcal{C}$ represents the label space. $\pi_s$ is the synthesis prompt, which incorporates the grading rubric $\phi$, the target label $y$, and the original response $x$ into a unified input that is fed into the LLM $f_\theta$ for generation.

**Generative Adversarial-based Quality Control (GAQC).** Although response rewriting mitigates distribution shift, synthetic responses may still differ from real student data. We therefore introduce a generative adversarial quality control mechanism with a discriminator $d_\omega(x) = P(\text{real} \mid x)$. Unlike conventional adversarial learning (Gui et al., 2021), which relies on parameter updates, we adopt an LLM-as-a-judge approach without fine-tuning. The LLM first compares samples from $\mathcal{D}_s$ and $\mathcal{D}_r$ to identify distinguishing characteristics $\omega$, then uses these characteristics to classify responses as real or synthetic. Formally:

$$d_\omega(x_i) \approx \frac{1}{M} \sum_{m=1}^{M} \mathbf{1}\left(f_\theta(\pi_d, \omega, x_i) = \text{real}\right), \quad \omega = f_\theta(\pi_c, \mathcal{D}_s, \mathcal{D}_r) \tag{4}$$

where $\mathbf{1}(\cdot)$ is the indicator function, $\pi_c$ and $\pi_d$ are the comparison and discrimination prompts, respectively, and $f\theta^{(m)}$ denotes the $m$-th stochastic evaluation. Repeating the evaluation $M$ times reduces variability in the estimated score. A higher $d_\omega(x_i)$ indicates that a synthetic response is more frequently judged real and is therefore used as a proxy for response fidelity. RODS retains responses with high scores for downstream optimization and refines those that remain distinguishable from real student responses.

Inspired by adversarial learning (Goodfellow et al., 2020), we use discriminator feedback to refine the synthesis prompt rather than update model parameters. Specifically, we incorporate $\omega$ into $\pi_s$ as constraints that instruct the LLM to avoid the identified synthetic artifacts during rewriting:

$$\pi_s' = \pi_s \oplus \neg\omega, \quad \tilde{x} = f_\theta(\pi_s', \phi, x, y), \tag{5}$$

where $\oplus\neg\omega$ denotes augmenting the synthesis prompt with instructions to avoid the distinguishing patterns. This feedback loop encourages synthetic responses to more closely resemble real student responses while preserving rubric- and score-conditioned generation. The overall synthesis procedure is summarized in Alg. 1 in Appx. C.1.

### 3.2 RUBRIC OPTIMIZATION

**Label-based Supervision.** In general, rubric optimization follows a unified three-step framework: grade, reflect, and refine. Given a dataset of student responses with human-annotated scores $\mathcal{D}_r$, each response is first assigned a predicted grade $\hat{y}_i$ using Eq. 1. By comparing predictions with ground-truth labels, we identify misaligned instances $\mathcal{E} = \{(x_i, y_i, \hat{y}_i) \mid (x_i, y_i) \in \mathcal{D}_r,\ y_i \neq \hat{y}_i\}$. Leveraging the self-reflection capability of LLMs, the model is then provided with either instance-level error cases or aggregated performance statistics (e.g., accuracy, confusion matrix) to produce a summary $\delta$. This summary diagnoses the shortcomings of the current rubric and proposes potential improvements. Finally, $\delta$ is combined with the current rubric $\phi$ and a refinement prompt $\pi_r$ to generate an updated rubric: $\phi' = f_\theta(\pi_r, \delta, \phi)$, $\delta = f_\theta(\pi_c, \mathcal{E}, \phi)$. In this work, we observe that the reflection output $\delta$ serves only as an intermediate signal for refinement. Given the strong step-by-step reasoning capabilities of modern LLMs, explicitly generating $\delta$ is unnecessary. To reduce inference cost, we instead internalize the reflection process within the model's reasoning and directly produce the refined rubric conditioned on the error set. The label-based optimization thus simplifies to:

$$\phi' = f_\theta(\pi_r', \mathcal{E}, \phi), \quad \pi_r' = \pi_r \otimes \pi_c \tag{6}$$

where $\otimes$ denotes prompt fusion, indicating that reflection is implicitly performed during generation.

**Consistency-based Supervision** To address the label quality issues of synthetic examples, we propose a consistency-based supervision paradigm built on the following self-consistency assumption: *If a rubric clearly defines the boundaries between score levels and is well understood by the model, then a generated sample conditioned on a label $y$ should be assigned the same label when graded using the same rubric.* Formally, given a rubric $\phi$, we generate synthetic samples $\hat{x}_y$ conditioned on label $y$, and then grade them using the same rubric. Ideally, the predicted label should match the conditioning label. Any misalignment indicates that the rubric fails to clearly distinguish between score levels and therefore requires refinement. The optimization objective can be expressed as:

$$\phi^* = \arg\max_{\Phi} \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\Big(y_t = f_\theta\big(\pi_g, \phi, q, f_\theta(\pi_s, \phi, x_t, y_t)\big)\Big),\ x_t \sim \mathcal{D}_r,\ y_t \sim \mathcal{C} \tag{7}$$

where $T$ denotes the number of synthetic samples, and $\mathbf{1}(\cdot)$ is the indicator function. Here, $x_t$ and $y_t$ represent the sampled student response and score label drawn from the student response dataset $\mathcal{D}_r$ and the score space $\mathcal{C}$, respectively. To identify the optimal self-consistent rubric $\phi^*$, we collect the inconsistent cases $\mathcal{E}' = \{(\hat{x}_y, y, \hat{y}) \mid \hat{x}_y \in \mathcal{D}_s,\ y \neq \hat{y}\}$, where the generated label $y$ and the graded label $\hat{y}$ do not match. These inconsistent instances are then exposed to the LLM for rubric refinement. However, allowing the LLM to fully decide how to revise the rubric without guidance may be suboptimal, as the rugged optimization landscape can easily cause the process to become trapped in local optima. To address this issue, RODS introduces two types of guided consistency optimization: prior-inclined guidance and posterior-inclined guidance. Prior guidance treats the generation condition label $y$ as the correct label and refines the rubric to resolve conflicts toward that label, while posterior guidance instead assumes the predicted label $\hat{y}$ is correct and revises the rubric accordingly. Given the inconsistent set $\mathcal{E}'$ and the guided refinement prompts $\pi_{c_0}$ and $\pi_{c_1}$, the consistency-based optimization can be formulated as:

$$\phi' = \arg\max\{\phi'_{c_0}, \phi'_{c_1}\} = \arg\max\{f_\theta(\pi_{c_0}, \mathcal{E}', \phi),\ f_\theta(\pi_{c_1}, \mathcal{E}', \phi)\}, \tag{8}$$

where $\phi^*_{c_0}$ and $\phi^*_{c_1}$ denote the refined rubrics generated under prior-guided and posterior-guided optimization, respectively.

**Iterative-based Optimization** As the optimization directions provided by the two supervision signals differ slightly, and we aim to balance their influence in a fine-grained manner, we propose an iterative optimization framework consisting of two steps: (1) the label-supervision step, which generates a new rubric $\phi'$ following Eq. 6, and (2) the consistency-supervision step, which further updates the rubric following Eq. 8. Meanwhile, since optimization is performed in a discrete sampled space, randomness introduced by sampling can significantly affect performance. To mitigate this issue, we incorporate the beam search strategy proposed in prior work (Pryzant et al., 2023) at each optimization iteration. Specifically, instead of generating only a single refined rubric $\phi'$, RODS produces a candidate rubric set $\Phi'$ by querying the LLM multiple times. To evaluate the quality of the refined rubrics, we measure both the grading accuracy on labeled samples and the self-consistency rate of each candidate rubric $\phi' \in \Phi'$. To balance the influence of these two supervision signals, we define the following weighted average of the two metrics as the ranking criterion:

$$\text{Index}(\phi) = \frac{\alpha}{|\mathcal{D}_r|} \cdot \sum_{i=1}^{|\mathcal{D}_r|} \mathbf{1}[y_i = f_\theta(\phi', x_i)] + \frac{1-\alpha}{|\mathcal{D}_s|} \cdot \sum_{j=1}^{|\mathcal{D}s|} \mathbf{1}[y_j = f_\theta(\phi', \hat{x}_{y_j})] \tag{9}$$

where $(x_i, y_i)$ and $(\hat{x}_{y_j}, y_j)$ are sampled from the labeled dataset $\mathcal{D}_r$ and the synthetic dataset $\mathcal{D}_s$, respectively, and $|\cdot|$ denotes the size of the corresponding dataset. The parameter $\alpha$ is the balancing factor that controls the relative influence of the two supervision signals during the optimization. Based on this ranking index, RODS retains the top-$k$ refined rubrics for the next optimization round, where $k$ denotes the beam size. The overall optimization procedure is summarized in Alg. 2 in Appx. C.2.

### 3.3 Semi-supervised Learning Paradigm

In RODS, consistency-based supervision requires only student responses $x$ and does not depend on ground-truth scores $y$. Therefore, when unscored student responses are available, RODS can

naturally extend consistency-based supervision to these unlabeled samples $\mathcal{D}_u$. By combining this signal with supervision from the existing score-labeled dataset $\mathcal{D}_r$, the refined rubric becomes more adaptive to the linguistic patterns and response characteristics of the target student population. This capability provides a significant advantage over prior rubric optimization methods, as it enables RODS to leverage the largely underexplored information contained in unlabeled responses.

# 4 EXPERIMENT

## 4.1 SETTINGS

**Dataset.** We evaluate RODS and the baselines on three AG datasets comprising more than 1,700 student responses to 12 questions in artificial intelligence ($\mathrm{D}_1$) (Aggarwal et al., 2024), earth science ($\mathrm{D}_2$) (Shin et al., 2026), and chemistry ($\mathrm{D}_3$) (Li et al., 2026). As discussed in § 1, existing LLM-based AG methods face challenges when labeled data are scarce. To evaluate RODS in this setting, we allocate 10 responses to training, 20 to validation, and 50 to testing for each question. For questions in $\mathrm{D}_1$, we instead use the first 10 responses for training, the next 10 for validation, and all remaining responses for testing. For the other two datasets, we apply stratified sampling by score level to account for highly imbalanced score distributions and maintain similar label distributions across the three splits. We repeat the data splitting procedure three times to account for variability in data partitioning and repeat the evaluation three times to account for variability in LLM generation. We report the mean performance across runs. Additional dataset statistics, question texts, and grading rubrics are provided in Appx. D.1.

**Baseline.** We compare RODS with six representative rubric optimization baselines: (1) Raw-rubric: The expert-provided rubric is directly combined with the grading prompt to score student responses. (2) Few-shot: All labeled training responses are included as in-context demonstrations, as they fit within the context window of each model. (3) Few-shot+: This variant augments Few-shot with LLM-generated synthetic demonstrations. (4) TextGrad (Yuksekgonul et al., 2025): A general-purpose prompt optimization (PO) framework adapted to optimize grading rubrics under the objective defined in Eq. 2. (5) GEPA (Agrawal et al., 2026): A PO framework that employs Pareto-based candidate selection to balance exploration and exploitation during optimization. (6) GradeOpt (Chu et al., 2024): A SOTA method specifically designed for grading rubric refinement. We evaluate all methods using *Qwen3-4B-Instruct* (Team, 2025) (Qwen) and *Gemini-3.1-Flash-Lite* (Team et al., 2023) (Gemini) to assess whether our findings generalize across backbone models with different capabilities.

**Implementation.** We implement all methods using Qwen, deployed locally on an H200 GPU, and Gemini, accessed through the Google AI Studio API[1]. Both models use their default generation parameters. The prompts used for all methods are provided in Appx. D.2. We tune the hyperparameters of RODS and the baselines over predefined discrete candidate values, with the final configurations selected based on validation performance. Additional details on the hyperparameter search space and selected configurations are provided in Appx. A.2. To account for the additional synthesis and discrimination steps introduced by RODS and ensure a fair comparison, we increase the optimization-step budget of all baseline methods by a factor of three. The average runtime is approximately two hours per question with Qwen, while the average Gemini API cost is approximately $10 per question. We evaluate grading performance using four metrics. Accuracy and Micro-F1 measure overall classification performance, while Cohen's Kappa (Kappa) and Quadratic Weighted Kappa (QWK) measure agreement between model predictions and human annotations beyond chance. Detailed definitions and interpretations of these metrics are provided in Appx. A.1.

## 4.2 MAIN RESULT

In Tab. 1, we compare the accuracy of RODS and the baselines across all questions. Results for the other three metrics are provided in Tab. 4 and Tab. 5 in Appx. B.1. We draw three main observations. (1) Comparing Few-shot with Few-shot+ shows that augmenting demonstrations with synthetic examples yields inconsistent benefits: accuracy improves in 10 question–backbone combinations but declines in the remaining 14. This finding reinforces the concerns about synthetic data quality discussed in § 1 and highlights the need for mechanisms to regulate its use. (2) TextGrad and GEPA improve average accuracy over Raw-rubric with both backbones but do not consistently outperform

[1] https://aistudio.google.com

Table 1: Accuracy (%) across twelve questions from three datasets using two backbones. Bold and underline mark the **best** and second-best results in each column per backbone.

| Method | $D_1$ | | | | $D_2$ | | | | $D_3$ | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | $Q_5$ | $Q_6$ | $Q_7$ | $Q_8$ | $Q_9$ | $Q_{10}$ | $Q_{11}$ | $Q_{12}$ | |
| *Qwen3-4B-Instruct* | | | | | | | | | | | | | |
| Raw Rubric | **19.1** | 22.2 | 60.6 | 50.0 | 29.3 | 30.0 | 41.3 | 30.7 | 55.3 | 51.3 | 34.7 | 46.0 | 39.2 |
| Few-shot | **19.1** | 15.6 | 63.6 | 54.8 | 34.7 | 43.3 | 58.0 | 52.0 | 54.0 | 52.7 | 30.0 | 41.3 | 43.3 |
| Few-shot+ | 19.0 | 17.8 | 57.6 | 52.4 | 38.0 | 47.3 | 54.0 | 58.7 | 56.0 | 47.3 | 38.7 | 46.7 | 44.5 |
| TextGrad | 14.3 | **31.1** | 54.5 | 54.8 | 48.0 | 32.0 | 26.0 | 53.3 | 64.7 | 48.0 | 46.0 | **48.7** | 43.4 |
| GEPA | 16.7 | 13.3 | 54.5 | 57.1 | 31.3 | 40.7 | 41.3 | 54.0 | 60.7 | 47.3 | **49.3** | 44.7 | 42.6 |
| GradeOPT | 16.7 | 26.7 | 66.7 | 64.3 | 41.3 | 43.3 | 60.7 | 47.3 | 62.7 | 50.7 | 30.0 | 42.7 | 46.1 |
| RODS | 16.7 | **31.1** | **75.8** | 73.8 | **49.3** | 54.0 | **62.0** | **60.7** | 62.7 | **54.0** | 36.0 | 44.7 | **51.7** |
| RODS(Semi) | 16.7 | 26.7 | 66.7 | **78.6** | 43.3 | **54.7** | 56.0 | 54.0 | **68.7** | 52.0 | 36.0 | 43.3 | 49.7 |
| *Gemini-3-Flash-Lite* | | | | | | | | | | | | | |
| Raw Rubric | 26.2 | 33.3 | 72.7 | 54.8 | 40.0 | 56.0 | 68.7 | 72.0 | 64.7 | 60.0 | 52.7 | 73.3 | 56.2 |
| Few-shot | 52.4 | 46.7 | 72.7 | 69.1 | 44.0 | 56.7 | 71.3 | 68.0 | 73.3 | **64.7** | **59.3** | 72.7 | 62.6 |
| Few-shot+ | 16.7 | 31.1 | 78.8 | 42.9 | 44.7 | **68.0** | 62.7 | 58.0 | 73.3 | 55.3 | 52.7 | 72.0 | 54.7 |
| TextGrad | 35.7 | 46.7 | 75.8 | 59.5 | 62.0 | 59.3 | 70.0 | **72.7** | 66.0 | 64.0 | 58.7 | 74.7 | 62.1 |
| GEPA | 38.1 | 46.7 | 75.8 | 69.0 | 57.3 | 62.0 | 66.0 | 68.0 | 71.3 | 58.7 | 51.3 | 72.7 | 61.4 |
| GradeOPT | 45.2 | 46.7 | 84.9 | 69.1 | **64.0** | 60.0 | 72.0 | 63.3 | 71.3 | 60.7 | 51.3 | 70.7 | 63.3 |
| RODS | 45.2 | **51.1** | 78.8 | 73.8 | 63.3 | 60.7 | **74.0** | **72.7** | **74.7** | **64.7** | 53.3 | **75.3** | 65.6 |
| RODS(Semi) | **54.8** | **51.1** | **90.9** | **76.2** | **64.0** | 62.7 | 72.7 | 71.3 | 72.7 | 62.7 | 52.0 | 72.7 | **67.0** |

Few-shot, suggesting that general-purpose prompt optimization offers limited advantages over in-context learning in this setting. GradeOpt achieves the second-highest average accuracy with both backbones, demonstrating the value of rubric-specific optimization. However, its modest gains over Few-shot underscore the challenges of optimizing rubrics with limited labeled data. (3) RODS achieves the highest average accuracy with both backbones, outperforming the strongest baselines by 4.4% and 2.3% with Qwen and Gemini, respectively. These results demonstrate the effectiveness of RODS and support its applicability to models with different grading capabilities.

To further assess the potential of RODS for semi-supervised rubric optimization, particularly in real-world educational settings where teachers may have access to large amounts of unlabeled student responses before rubric refinement, we incorporate unlabeled responses from the validation and test splits into the data synthesis process and report the resulting performance as RODS (semi). During synthesis, only the response text is used, while the corresponding labels remain inaccessible. The benefits of incorporating these unlabeled responses vary across backbones: Qwen performs best when synthesis relies only on training responses, whereas Gemini achieves its best performance when responses from the validation and test splits are also incorporated. Nevertheless, both the supervised and semi-supervised variants of RODS outperform the strongest competing baseline for their respective backbones. These results demonstrate that RODS can effectively leverage unlabeled student responses for rubric optimization, while also suggesting that the extent of the benefit depends on the underlying backbone model.

### 4.3 Ablation Study

**Data Synthesis Designs.** To assess response rewriting and GAQC, we evaluate two variants: (i) w/o Rewrite, which generates responses from scratch, and (ii) w/o GAQC, which uses synthetic responses without quality control. Fig. 2 reports average accuracy across 12 questions for both backbones, with gaps relative to RODS annotated above the markers; detailed results and additional metrics are provided in Appx. B.2. Removing either component reduces accuracy, although both variants still outperform Raw-rubric, highlighting the complementary benefits of rewriting and quality control. Removing GAQC has a greater impact on Gemini, whereas removing rewriting affects Qwen slightly more. Our analysis suggests that Qwen's less diverse generation benefits more from grounding in student responses, while Gemini's more varied responses may deviate from students' reasoning and therefore

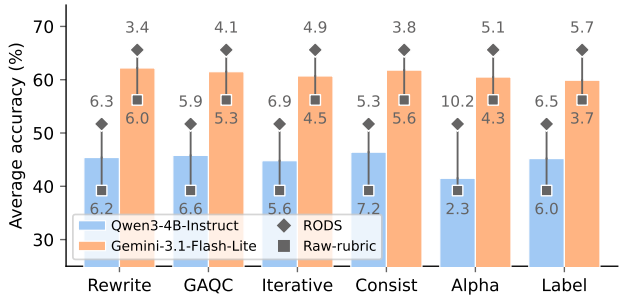
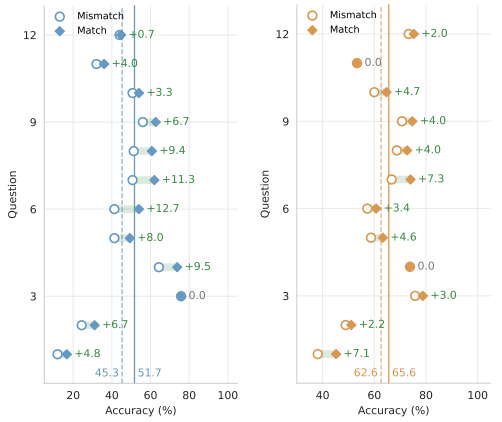



Figure 2: Accuracy of RODS ablation variants. Annotated gaps indicate differences relative to the full framework.

benefit more from quality control. Case studies of synthetic response quality and the distinguishing feature $\omega$ are provided in Appx. B.4.

**Rubric Optimization Designs.** To assess consistency-based supervision and iterative optimization, we evaluate two variants: (i) w/o Consistency, which 5 synthetic labels as ground truth throughout optimization, and (ii) w/o Iterative, which combines synthetic and human-labeled responses in a single refinement step. Fig. 2 reports average accuracy across 12 questions for both backbones; question-level results and the other three metrics are provided in Appx. B.2. Removing either component reduces accuracy, confirming their contributions to rubric optimization. Removing iterative optimization causes a larger decline for both models, highlighting the value of progressive rubric refinement over a single optimization step.

**Label Supervision Designs.** To assess the role of human-labeled responses in RODS, we evaluate two variants: (i) w/o Alpha, which sets $\alpha = 0$ in Eq. 9, so that candidate rubric selection relies solely on synthetic responses, and (ii) w/o Label, which additionally removes human-labeled responses from the reflection process, making optimization entirely dependent on synthetic data. Fig. 2 compares accuracy, while results for the other three metrics are provided in Appx. B.2. Both variants substantially reduce accuracy relative to RODS, highlighting the importance of human-labeled supervision. The relative impact differs by backbone: Gemini is more affected by removing labeled responses from reflection, whereas Qwen suffers more when candidate selection lacks human-labeled guidance. These differences suggest that the two models rely on human supervision differently during reflection and candidate selection.

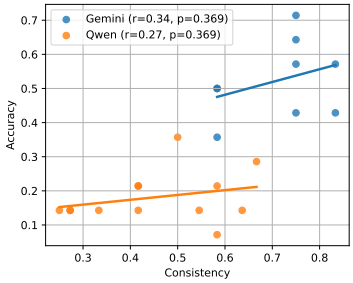



(a) Qwen3-4B-Instruct (b) Gemini-3.1-Flash-Lite

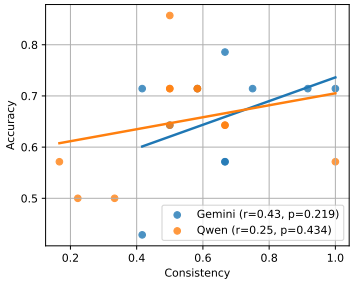
Figure 3: Accuracy across all question with two backbone models, comparing matched and mismatched LLMs for grading and data synthesis.

**Grade and Synthesis Model Match.** Beyond component ablations, we examine how matching the grading and synthesis models affects RODS. Fig. 3 presents question-level accuracy comparisons, with results for the other metrics provided in Appx. B.3. Using different models for grading and synthesis reduces average accuracy by 6.4% for Qwen and 3.0% for Gemini. These findings support the motivation in § 3.2: consistency-based supervision is more effective when synthesis and grading use the same model. A possible explanation is that responses generated by a different model provide less informative feedback about the grading model's interpretation of the rubric, making reflection less effective at identifying and correcting its misunderstandings.

**Correlation Between Self-Consistency and Accuracy.** To assess the assumption underlying RODS that self-consistency can serve as a surrogate supervision signal for rubric optimization (§ 3.2), we examine $Q_1$ and $Q_4$ and plot the accuracy and self-consistency of prompts generated during optimization in Fig. 4. The plots show a positive correlation between accuracy and self-consistency across prompts, supporting the use of self-consistency as a surrogate signal. This correlation is more pronounced for Gemini, the stronger backbone, than for Qwen. One possible explanation is that Gemini better interprets rubric details and identifies contradictory patterns, whereas Qwen may be less sensitive to these distinctions, resulting in a weaker association between self-consistency and accuracy.




(a) $Q_1$




(b) $Q_4$

Figure 4: Correlations between Accuracy and self-consistency.

## 5 Related Work

**LLM-based Automatic Grading.** The emergence of LLMs has significantly advanced AG methods (Schneider et al., 2023). Unlike earlier deep learning approaches that rely on parameter optimization, recent LLM-based methods focus on improving performance through input context design and prompt engineering. A key line of work explores automatic rubric optimization. For example, Chu et al. (2024) propose an iterative framework that refines grading rubrics by reflecting on errors observed in labeled samples. Building on this, Chu et al. (2025b) introduce a human-in-the-loop paradigm, incorporating expert feedback into the refinement process to improve both efficiency and transparency. Beyond rubric design, recent studies have investigated optimizing in-context learning (ICL) examples to enhance grading performance (Chu et al., 2026a). These methods leverage a small set of expert-provided grading rationales as seeds and use LLMs to generate synthetic rationales for ICL examples. Other directions include incorporating external knowledge and improving robustness. For instance, retrieval-augmented generation (RAG) has been applied to support grading tasks requiring domain-specific knowledge (Chu et al., 2025a). In addition, Shin et al. (2026) propose a multi-agent framework in which multiple LLMs collaboratively evaluate responses and aggregate consensus, improving stability while using disagreement as a signal of uncertainty.

**Learning under Limited-labeled Data.** Learning under limited labeled data has been extensively studied through data synthesis, self-supervised learning, and active learning. Back-translation (Sennrich et al., 2016) is a classical approach that leverages unlabeled data to generate synthetic supervision, and similar ideas have been extended to modern LLM-based data synthesis methods (Wang et al., 2023; Honovich et al., 2023). However, such approaches often suffer from noise and distributional mismatch in generated samples. Self-supervised learning (Gui et al., 2024) constructs training signals directly from unlabeled data via proxy objectives, enabling models to learn without manual annotations. In parallel, active learning (Ren et al., 2021) improves data efficiency by selecting informative samples for annotation, reducing labeling cost while maintaining performance. More recently, LLM-based methods combine generation and self-reflection to improve performance in limited-labeled settings (Wang et al., 2022). However, these approaches typically rely on either high-quality synthetic data or sufficient labeled supervision. In contrast, our method leverages a self-consistency objective as an intrinsic supervision signal, enabling effective optimization without requiring additional annotations and improving robustness to noisy synthetic data.

## 6 Conclusion

In this work, we propose RODS, a rubric optimization framework that leverages the generative capabilities of LLMs to improve automatic grading. RODS introduces a self-consistency objective that aligns label-conditioned generation with grading outcomes, enabling effective rubric optimization without additional scored samples. We further introduce GAQC to improve the fidelity of synthetic data and combine it with limited supervised signals in a unified optimization framework. Extensive experiments on 12 questions across three diverse datasets demonstrate that RODS consistently outperforms existing rubric optimization methods when labeled data are scarce. Ablation studies clarify the contributions of individual components and provide practical guidance for applying RODS across different backbone capabilities and levels of data availability. Our analysis also reveals a positive correlation between self-consistency and grading accuracy, supporting the use of self-consistency as a surrogate supervision signal when labels are limited. Several limitations remain. Although GAQC improves the fidelity of synthetic responses, their quality still depends on the capabilities of the underlying LLM, and distributional mismatches between synthetic and authentic responses may persist. In addition, the framework relies on automated optimization signals, without human verification during synthesis and refinement. Incorporating human feedback into these stages may further improve the reliability and interpretability of rubric optimization. We leave these directions for future work.

## AI use statement

In this work, we only used generative AI tools for polishing written content for grammar and spelling. We have not used generative AI tools for any experiment design, execution or content creation. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES


Dishank Aggarwal, Pritam Sil, Bhaskaran Raman, and Pushpak Bhattacharyya. " i understand why i got this grade": Automatic short answer grading with feedback. *arXiv preprint arXiv:2407.12818*, 2024.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 8479–8565, 2026.

Leopold Bayerlein. Students' feedback preferences: how do students react to timely and automatically generated assessment feedback? *Assessment & Evaluation in Higher Education*, 39(8):916–931, 2014.

Sridevi Bonthu, S Rama Sree, and MHM Krishna Prasad. Automated short answer grading using deep learning: A survey. In *International cross-domain conference for machine learning and knowledge extraction*, pp. 61–78. Springer, 2021.

Yucheng Chu, Hang Li, Kaiqi Yang, Harry Shomer, Hui Liu, Yasemin Copur-Gencturk, and Jiliang Tang. A llm-powered automatic grading framework with human-level guidelines optimization. *arXiv preprint arXiv:2410.02165*, 2024.

Yucheng Chu, Peng He, Hang Li, Haoyu Han, Kaiqi Yang, Yu Xue, Tingting Li, Joseph Krajcik, and Jiliang Tang. Enhancing llm-based short answer grading with retrieval-augmented generation. *arXiv preprint arXiv:2504.05276*, 2025a.

Yucheng Chu, Hang Li, Kaiqi Yang, Yasemin Copur-Gencturk, and Jiliang Tang. Llm-based automated grading with human-in-the-loop. In *2025 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pp. 1–8. IEEE, 2025b.

Yucheng Chu, Hang Li, Kaiqi Yang, Yasemin Copur-Gencturk, Kevin Haudek, Joseph Krajcik, and Jiliang Tang. Optimizing in-context demonstrations for llm-based automated grading. *arXiv preprint arXiv:2603.00465*, 2026a.

Yucheng Chu, Hang Li, Kaiqi Yang, Yasemin Copur-Gencturk, Joseph Krajcik, Namsoo Shin, and Jiliang Tang. Confusion-aware rubric optimization for llm-based automated grading. *arXiv preprint arXiv:2603.00451*, 2026b.

Rafael Ferreira Mello, Cleon Pereira Junior, Luiz Rodrigues, Filipe Dwan Pereira, Luciano Cabral, Newarney Costa, Geber Ramalho, and Dragan Gasevic. Automatic short answer grading in the llm era: Does gpt-4 with prompt engineering beat traditional models? In *Proceedings of the 15th international learning analytics and knowledge conference*, pp. 93–103, 2025.

Lucas Busatta Galhardi and Jacques Duílio Brancher. Machine learning approach for automatic short answer grading: A systematic review. In *Ibero-american conference on artificial intelligence*, pp. 380–391. Springer, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE transactions on knowledge and data engineering*, 35(4):3313–3332, 2021.

Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9052–9071, 2024.

Xu Guo and Yiqiang Chen. Generative ai for synthetic data generation: Methods, challenges and the future. *arXiv preprint arXiv:2403.04190*, 2024.

Antonio Hernández-Blanco, Boris Herrera-Flores, David Tomás, and Borja Navarro-Colorado. A systematic review of deep learning approaches to educational data mining. *Complexity*, 2019(1): 1306039, 2019.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14409–14428, 2023.

Hang Li, Kaiqi Yang, Xianxuan Long, Fedor Filippov, Yucheng Chu, Yasemin Copur-Gencturk, Peng He, Cory Miller, Namsoo Shin, Joseph Krajcik, et al. How uncertain is the grade? a benchmark of uncertainty metrics for llm-based automatic assessment. *arXiv preprint arXiv:2602.16039*, 2026.

Joan Lu, Bhavya Krishna Balasubramanian, Mike Joy, and Qiang Xu. Survey and analysis for the challenges in computer science to the automation of grading systems. *ACM Computing Surveys*, 58(1):1–37, 2025.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282, 2012.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7957–7968, 2023.

VV Ramalingam, A Pandian, Prateek Chetry, and Himanshu Nigam. Automated essay grading using machine learning algorithm. In *Journal of Physics: Conference Series*, volume 1000, pp. 012030. IOP Publishing, 2018.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

Johannes Schneider, Bernd Schenk, and Christina Niklaus. Towards llm-based autograding for short textual answers. *arXiv preprint arXiv:2309.11508*, 2023.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 86–96, 2016.

Namsoo Shin, Xunlei Qian, Cory Susanne Miller, Hang Li, Yucheng Chu, Joseph Krajcik, Jiliang Tang, and Yue Xing. An artificial intelligence-enhanced assessment framework for analyzing middle school science students' written responses. *Journal of Educational Measurement*, 63(2): e70044, 2026.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. Large language models for education: A survey and outlook. *IEEE Signal Processing Magazine*, 42(6):51–63, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023.

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 from implicit weights to explicit rubrics for reward modeling. *arXiv preprint arXiv:2510.17314*, 2025a.

Wenjing Xie, Juxin Niu, Chun Jason Xue, and Nan Guan. Grade like a human: Rethinking automated assessment with large language models. In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, pp. 1–8, 2025b.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639(8055):609–616, 2025.

# A Additional Settings

## A.1 Evaluation Metrics

We adopt four evaluation metrics to comprehensively compare the performance of different methods. Specifically, the first two metrics, Accuracy and micro-F1 are standard classification metrics to measure overall classification performance. However, score-level distributions in student responses are often highly imbalanced, and the dominance of a single score category may lead to deceptively high accuracy. To mitigate this issue, we additionally adopt Cohen's Kappa (Kappa), an agreement-based metric that measures the degree to which model predictions exceed chance-level agreement with human annotations. Furthermore, score levels are ordinal rather than independent categories. For example, misclassifying a score of 0 as 3 is substantially more severe than misclassifying a score of 2 as 1. To better capture this ordinal structure, we also use Quadratic Weighted Kappa (QWK), which assigns larger penalties to more severe grading errors while being more tolerant of minor deviations. The definitions of these metrics are given below:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}(y_i = \hat{y}_i) \tag{10}$$

$$\text{micro-F}_1 = \frac{2\sum_{k=1}^{K}\text{TP}_k}{2\sum_{k=1}^{K}\text{TP}_k + \sum_{k=1}^{K}\text{FP}_k + \sum_{k=1}^{K}\text{FN}_k} \tag{11}$$

$$\text{Kappa} = \frac{p_o - p_e}{1 - p_e}, \quad p_o = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}(y_i = \hat{y}_i), \quad p_e = \sum_{k=1}^{K}\frac{n_k^{(y)}}{N}\cdot\frac{n_k^{(\hat{y})}}{N} \tag{12}$$

$$\text{QWK} = 1 - \frac{\sum_{i,j} w_{ij}O_{ij}}{\sum_{i,j} w_{ij}E_{ij}}, \quad w_{ij} = \frac{(i-j)^2}{(K-1)^2} \tag{13}$$

where $N$ denotes the total number of samples, $K$ is the number of score levels, and $\text{TP}_k$, $\text{FP}_k$, and $\text{FN}_k$ represent the true positives, false positives, and false negatives for class $k$, respectively. In addition, $n_k^{(y)}$ and $n_k^{(\hat{y})}$ denote the number of instances in class $k$ for the ground-truth and predicted labels. For the kappa metrics, $O_{ij}$ and $E_{ij}$ represent the observed and expected agreement counts for label pair $(i, j)$, respectively, and $w_{ij}$ defines the quadratic penalty that assigns larger weights to more severe disagreements between score levels.

## A.2 Hyper-parameter Tuning Settings

Table 2: Hyper-parameter candidate settings of GradeOpt and RODS.

| **Rubric Optimize** | | | **Data Synthesis** | | **GAQC** | |
|---|---|---|---|---|---|---|
| **Iteration** ($R$) | **Beam size** ($k$) | **Balance weight** ($\alpha$) | **Source** | **Size** ($T$) | **Iteration** ($S$) | **Threshold** ($\mu$) |
| 5 | 2 | {0.9,0.5} | {train,test[Semi]} | {4,16} | 2 | 0.5 |

The detailed hyper-parameter settings for our experiments are summarized in Tab. 2. The parameters are grouped into three categories corresponding to the key components of RODS. For the rubric optimization component, both RODS and GradeOpt use a fixed number of iterations $R = 5$. The beam size in the beam search strategy (Pryzant et al., 2023) is set to $k = 2$. For RODS, the balance factor $\alpha$, which controls the relative influence of labeled and synthetic data, is selected from two values: $0.9$, where labeled data dominates the optimization, and $0.5$, where label-based and consistency-based signals contribute equally. For the data synthesis component, we explore different sources of input samples for rewriting, including training, testing, and combined splits. When only the test split is used, RODS operates in a semi-supervised learning setting as described in § 3.3. We also vary the number of synthetic samples per score category, using values of 4 and 16, to evaluate its impact on performance. For the GAQC module, we fix two parameters: the number of refinement iterations is set to $S = 2$, and the threshold for indistinguishability is set to $\mu = 0.5$. Specifically, synthetic

Table 3: Parameter setting details of RODS over questions across the three datasets.

| Source | Backbone | Parameter | $D_1$ | | | | $D_2$ | | | | $D_3$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | $Q_5$ | $Q_6$ | $Q_7$ | $Q_8$ | $Q_9$ | $Q_{10}$ | $Q_{11}$ | $Q_{12}$ |
| Train | Qwen3-4B-Instruct | Balace weight | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.5 |
| | | Synthesis Size | 16 | 16 | 4 | 4 | 16 | 4 | 16 | 16 | 16 | 16 | 16 | 4 |
| | Gemini-3.1-Flash-Lite | Balace weight | 0.9 | 0.5 | 0.5 | 0.5 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.5 | 0.9 | 0.9 |
| | | Synthesis Size | 4 | 4 | 4 | 4 | 4 | 4 | 16 | 16 | 16 | 4 | 16 | 4 |
| Test[Semi] | Qwen3-4B-Instruct | Balace weight | 0.5 | 0.9 | 0.9 | 0.9 | 0.5 | 0.9 | 0.5 | 0.9 | 0.5 | 0.9 | 0.9 | 0.9 |
| | | Synthesis Size | 4 | 16 | 4 | 4 | 4 | 4 | 4 | 4 | 16 | 16 | 16 | 16 |
| | Gemini-3.1-Flash-Lite | Balace weight | 0.9 | 0.9 | 0.5 | 0.5 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.5 |
| | | Synthesis Size | 16 | 16 | 16 | 4 | 4 | 4 | 16 | 4 | 4 | 4 | 16 | 4 |

samples are retained for subsequent rubric optimization if the discriminator accuracy is below 0.5, indicating that they are difficult to distinguish from real student responses. We perform grid search over all parameter combinations, and the best configurations for each question are reported in Tab. 3.

# B Additional Results

## B.1 Main Result

The question-level results for RODS and the baselines using *Qwen3-4B-Instruct* and *Gemini-3.1-Flash-Lite* are presented in Tab. 4 and Tab. 5, respectively. From the tables, we make the following observations. First, RODS achieves the highest average performance on all four metrics with both backbones. Compared with GradeOPT, the strongest baseline in terms of average performance, RODS delivers absolute improvements of approximately 2% to 7%. These gains demonstrate improvements in both predictive performance and agreement with human grading. Second, RODS ranks first or second on most questions for each backbone and metric. Although the gains vary across questions, the results demonstrate its effectiveness across the question types and topics evaluated. This breadth is important for practical deployment, where automated grading systems must accommodate diverse assessment tasks. Third, Cohen's Kappa shows the largest absolute improvement over GradeOPT for both backbones, increasing by 7.3 for Qwen and 5.6 for Gemini. These gains indicate stronger chance-corrected agreement with human grading, an important consideration in educational assessment. Finally, RODS consistently delivers larger gains over GradeOPT for Qwen than for Gemini, which is consistent with prior findings (Xie et al., 2025a) and suggests that rubric optimization may act as a form of fine-grained instruction enhancement. More explicit grading guidance may be particularly beneficial for less capable models, whereas stronger models may derive smaller gains because of their existing reasoning and instruction-following capabilities.

## B.2 Component Ablation Result

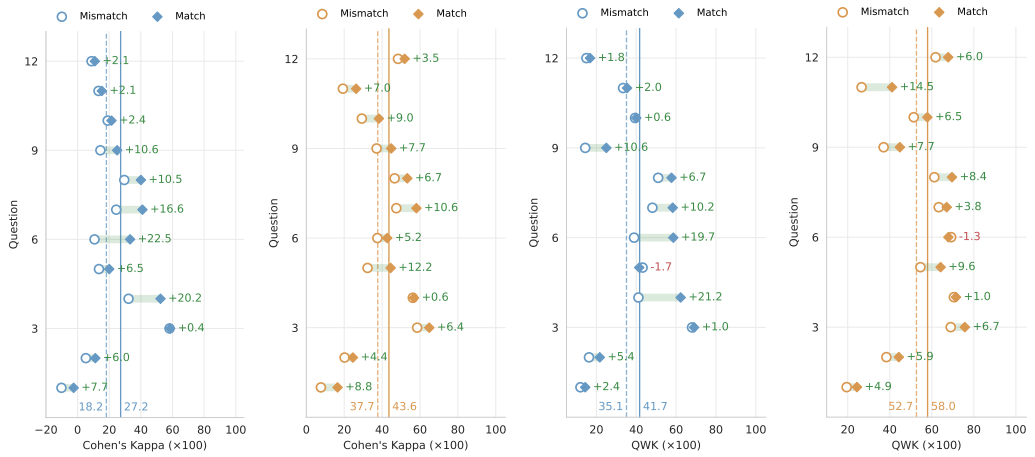
The question-level component ablation results across all four metrics are reported in Tab. 6 and Tab. 7 for *Qwen3-4B-Instruct* and *Gemini-3.1-Flash-Lite*, respectively. We make the following observations. First, consistent with the findings in § 4.3, removing any component of RODS reduces average performance across all four metrics. The absolute decreases range from 3.5 to 6.4 for Qwen and from 3.4 to 5.7 for Gemini. These results demonstrate that each component contributes to the overall effectiveness of RODS. Second, full RODS achieves the best performance, including ties, in 41 of the 48 question–metric combinations for Qwen and 37 of the 48 combinations for Gemini. Although individual ablated variants occasionally outperform full RODS, these exceptions do not translate into better average performance. The overall pattern supports the effectiveness of the proposed components across the evaluated questions and metrics. Third, removing components generally causes larger performance drops for Qwen than for Gemini, although this pattern is not uniform across all components and metrics. This suggests that stronger models such as Gemini may partially compensate for missing components through their existing capabilities, whereas less capable models may benefit more from a carefully designed optimization process.

Table 4: Results using *Qwen3-4B-Instruct* across twelve questions from three datasets. Values in parentheses indicate standard deviations across multiple runs. Bold and underlined values denote the **best** and second-best results, respectively, in each column for each metric

| Method | $D_1$ | | | | $D_2$ | | | | $D_3$ | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | $Q_5$ | $Q_6$ | $Q_7$ | $Q_8$ | $Q_9$ | $Q_{10}$ | $Q_{11}$ | $Q_{12}$ | |
| *Accuracy* | | | | | | | | | | | | | |
| Raw Rubric | **19.1**(0.0) | 22.2(0.0) | 60.6(0.0) | 50.0(0.0) | 29.3(0.9) | 30.0(0.6) | 41.3(0.2) | 30.7(0.0) | 55.3(0.0) | 51.3(0.2) | 34.7(0.2) | 46.0(0.2) | 39.2 |
| Few-shot | **19.1**(1.2) | 15.6(0.0) | 63.6(0.9) | 54.8(0.7) | 34.7(0.7) | 43.3(0.2) | 58.0(0.4) | 52.0(0.4) | 54.0(0.0) | 52.7(0.4) | 30.0(0.5) | 41.3(0.0) | 43.3 |
| Few-shot+ | 19.0(1.4) | 17.8(0.3) | 57.6(1.2) | 52.4(1.3) | 38.0(2.1) | 47.3(1.1) | 54.0(1.8) | 58.7(1.4) | 56.0(1.1) | 47.3(0.7) | 38.7(1.8) | 46.7(0.8) | 44.5 |
| TextGrad | 14.3(1.4) | **31.1**(5.9) | 54.5(0.9) | 54.8(1.8) | 48.0(4.0) | 32.0(1.0) | 26.0(6.4) | 53.3(1.7) | 64.7(0.7) | 48.0(0.4) | 46.0(2.3) | **48.7**(1.2) | 43.4 |
| GEPA | 16.7(1.8) | 13.3(0.0) | 54.5(0.9) | 57.1(0.7) | 31.3(1.7) | 40.7(0.9) | 41.3(2.7) | 54.0(0.2) | 60.7(1.4) | 47.3(0.2) | **49.3**(1.6) | 44.7(1.1) | 42.6 |
| GradeOPT | 16.7(1.8) | 26.7(1.3) | 66.7(1.5) | 64.3(1.8) | 41.3(1.4) | 43.3(0.2) | 60.7(3.4) | 47.3(2.9) | 62.7(0.8) | 50.7(0.4) | 30.0(4.0) | 42.7(0.5) | 46.1 |
| RODS | 16.7(3.7) | **31.1**(2.3) | **75.8**(3.6) | 73.8(3.2) | **49.3**(4.1) | 54.0(4.1) | **62.0**(3.4) | **60.7**(2.2) | 62.7(1.9) | **54.0**(1.8) | 36.0(2.1) | 44.7(0.7) | **51.7** |
| RODS(Semi) | 16.7(3.4) | 26.7(2.6) | 66.7(3.0) | **78.6**(3.2) | 43.3(4.2) | **54.7**(3.6) | 56.0(2.8) | 54.0(2.2) | **68.7**(2.2) | 52.0(1.6) | 36.0(2.4) | 43.3(0.8) | 49.7 |
| *Micro F1* | | | | | | | | | | | | | |
| Raw Rubric | 10.5(0.0) | 18.2(0.0) | 53.6(0.0) | 33.3(0.0) | 35.1(1.0) | 26.5(0.4) | 43.3(0.2) | 31.0(0.1) | 51.7(0.0) | 54.7(0.2) | 44.0(0.2) | 43.6(0.2) | 37.1 |
| Few-shot | 10.6(0.4) | 6.5(0.0) | 56.7(0.9) | 41.3(1.3) | 39.4(0.9) | 34.0(0.3) | 56.3(0.4) | 51.8(0.4) | 49.2(0.0) | 54.4(0.3) | 39.0(0.5) | 34.3(0.0) | 39.5 |
| Few-shot+ | 10.7(1.4) | 9.3(0.3) | 47.0(1.2) | 37.1(1.3) | 42.6(2.1) | 35.0(1.1) | 53.5(1.9) | 57.3(1.5) | 53.3(1.1) | 50.0(0.7) | 47.9(1.8) | 43.1(0.8) | 40.6 |
| TextGrad | 8.6(1.5) | 25.8(5.9) | 46.3(0.7) | 40.9(1.7) | 44.1(3.3) | 26.0(0.7) | 27.5(4.6) | 52.0(1.1) | 59.1(1.6) | 51.4(0.4) | 48.2(1.0) | **47.1**(2.3) | 39.8 |
| GEPA | 6.2(0.7) | 9.9(0.6) | 44.0(1.6) | 43.0(1.1) | 32.2(1.8) | 36.0(1.0) | 41.6(3.0) | 51.8(0.5) | 59.4(1.1) | 50.1(0.2) | **51.3**(1.1) | 43.9(0.6) | 39.1 |
| GradeOPT | 16.1(2.3) | 28.2(1.9) | 59.4(1.7) | 59.8(2.5) | 45.0(0.8) | 42.8(0.6) | 60.2(3.8) | 47.3(2.9) | 62.1(0.9) | 52.4(0.6) | 37.8(4.3) | 41.4(0.7) | 46.0 |
| RODS | 13.3(4.1) | **33.9**(2.7) | **72.3**(4.2) | 70.9(4.2) | **51.0**(3.8) | 51.9(3.4) | **62.1**(3.7) | **60.2**(1.9) | 61.3(1.9) | **55.1**(1.5) | 40.7(1.5) | 41.6(1.2) | **51.2** |
| RODS(Semi) | **18.8**(3.9) | 25.6(2.8) | 59.4(3.6) | **77.4**(4.0) | 47.1(3.6) | **52.1**(3.2) | 54.2(3.2) | 53.7(2.0) | **66.6**(2.2) | 54.7(1.2) | 44.6(1.6) | 41.8(1.4) | 49.7 |
| *Cohen's Kappa* | | | | | | | | | | | | | |
| Raw Rubric | -3.7(0.0) | 7.4(0.0) | 30.1(0.0) | 0.0(0.0) | 6.9(1.0) | 5.3(0.7) | 16.4(0.2) | 6.8(0.1) | 18.1(0.0) | **25.4**(0.3) | 15.6(0.2) | 11.5(0.3) | 11.7 |
| Few-shot | -3.3(1.3) | 2.4(0.0) | 35.7(1.6) | 10.5(1.6) | 10.3(0.9) | 10.1(0.3) | 33.5(0.6) | 28.3(0.3) | 16.4(0.0) | 23.7(0.7) | 12.3(0.6) | 9.3(0.0) | 15.8 |
| Few-shot+ | -5.3(1.4) | 4.7(0.8) | 30.8(1.8) | 6.1(1.6) | 14.0(2.1) | 13.3(1.1) | 28.7(2.3) | 33.0(1.5) | 18.6(1.3) | 22.8(0.7) | 19.2(1.9) | **15.4**(1.0) | 16.8 |
| TextGrad | -5.2(1.5) | 8.0(4.5) | 21.1(1.2) | 14.7(2.9) | 15.2(3.5) | 3.1(0.9) | -0.5(5.5) | 32.2(2.8) | 21.2(1.9) | 22.8(0.4) | 17.4(0.6) | 11.1(3.9) | 13.4 |
| GEPA | -4.7(1.6) | -3.2(0.6) | 18.7(1.6) | 17.9(1.8) | -2.2(1.3) | 11.3(1.4) | 15.0(3.6) | 23.4(2.2) | 23.0(2.3) | 22.3(0.2) | **20.1**(0.6) | 8.5(0.6) | 12.5 |
| GradeOPT | -4.0(1.9) | 9.0(1.2) | 41.2(2.8) | 33.1(3.9) | 11.0(2.3) | 17.0(0.7) | 38.1(4.1) | 26.4(1.9) | 28.0(1.9) | 21.2(0.4) | 12.6(2.1) | 5.8(0.9) | 19.9 |
| RODS | **-2.5**(2.7) | **11.2**(1.7) | **58.5**(6.2) | 52.4(6.4) | **20.0**(4.3) | **33.2**(4.9) | **41.0**(4.3) | **40.0**(2.4) | 25.0(2.7) | 21.4(1.6) | 15.3(0.7) | 10.9(0.7) | **27.2** |
| RODS(Semi) | -6.6(2.4) | 7.2(1.5) | 41.2(4.2) | **62.8**(5.4) | 16.4(4.2) | 29.5(4.2) | 33.3(4.1) | 31.2(2.2) | **32.3**(3.1) | 21.6(1.4) | 15.6(1.2) | 7.5(1.0) | 24.3 |
| *QWK* | | | | | | | | | | | | | |
| Raw Rubric | 8.4(0.0) | 18.2(0.0) | 39.9(0.0) | 0.0(0.0) | 28.1(0.3) | 25.7(0.7) | 29.6(0.1) | 41.2(0.2) | 18.1(0.0) | **44.2**(0.4) | **39.2**(0.3) | 17.6(0.3) | 25.8 |
| Few-shot | 10.3(2.1) | 6.6(0.0) | 45.6(1.7) | 12.9(2.0) | 39.1(0.3) | 32.4(0.5) | 50.8(0.4) | 54.7(0.1) | 16.4(0.0) | 41.1(0.4) | 35.7(0.4) | 17.7(0.0) | 30.3 |
| Few-shot+ | -0.6(2.0) | 10.2(0.9) | 42.2(2.1) | 9.6(1.7) | 40.7(1.7) | 31.2(1.4) | 45.0(1.7) | **58.1**(1.1) | 18.6(0.9) | 42.1(0.4) | 38.0(1.6) | **26.0**(1.3) | 30.1 |
| TextGrad | 9.8(0.5) | 10.3(3.4) | 35.3(1.8) | 25.2(1.9) | 40.0(2.6) | 29.2(2.2) | 32.8(2.1) | 52.3(2.7) | 21.2(1.9) | 41.8(0.8) | 30.7(2.2) | 23.2(5.2) | 29.3 |
| GEPA | 4.4(0.6) | 7.8(1.1) | 27.1(1.7) | 26.0(2.8) | 28.2(1.4) | 40.2(0.9) | 34.3(2.4) | 49.5(1.0) | 18.1(1.4) | 40.8(0.9) | 25.5(3.1) | 18.5(1.3) | 26.7 |
| GradeOPT | **21.1**(2.5) | **26.8**(2.6) | 50.4(2.5) | 43.5(4.7) | 37.0(0.7) | 48.6(1.0) | 54.6(0.9) | 53.0(1.6) | 28.0(1.9) | 34.5(2.5) | 32.7(2.5) | 14.1(0.3) | 37.0 |
| RODS | 14.4(2.2) | 21.7(1.5) | **69.0**(5.4) | 62.3(5.8) | 41.5(2.1) | **58.6**(1.9) | 58.3(2.0) | 57.7(0.6) | 25.0(2.7) | 40.0(1.8) | 35.4(0.7) | 16.8(1.3) | **41.7** |
| RODS(Semi) | 13.3(2.4) | 10.6(1.3) | 50.4(3.6) | **72.8**(4.3) | **44.8**(2.1) | 50.9(1.8) | **61.0**(1.8) | 54.8(1.2) | **32.3**(3.0) | 42.9(1.6) | 36.4(1.2) | 14.2(1.3) | 40.4 |

Table 5: Results using *Gemini-3-Flash-Lite* across twelve questions from three datasets. Values in parentheses indicate standard deviations across multiple runs. Bold and underlined values denote the **best** and second-best results, respectively, in each column for each metric

| Method | $D_1$ | | | | $D_2$ | | | | $D_3$ | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | $Q_5$ | $Q_6$ | $Q_7$ | $Q_8$ | $Q_9$ | $Q_{10}$ | $Q_{11}$ | $Q_{12}$ | |
| *Accuracy* | | | | | | | | | | | | | |
| Raw Rubric | 26.2(1.5) | 33.3(1.6) | 72.7(0.9) | 54.8(2.1) | 40.0(1.4) | 56.0(0.4) | 68.7(0.9) | 72.0(0.7) | 64.7(1.0) | 60.0(0.9) | 52.7(0.4) | 73.3(0.2) | 56.2 |
| Few-shot | 52.4(1.2) | 46.7(2.1) | 72.7(0.9) | 69.1(1.0) | 44.0(1.2) | 56.7(1.3) | 71.3(1.3) | 68.0(0.4) | 73.3(0.9) | **64.7**(0.6) | **59.3**(0.9) | 72.7(0.9) | 62.6 |
| Few-shot+ | 16.7(1.4) | 31.1(1.4) | 78.8(2.1) | 42.9(1.7) | 44.7(1.7) | **68.0**(1.3) | 62.7(1.1) | 58.0(0.9) | 73.3(0.8) | 55.3(0.9) | 52.7(0.9) | 72.0(1.1) | 54.7 |
| TextGrad | 35.7(1.2) | 46.7(0.7) | 75.8(4.7) | 59.5(1.2) | 62.0(2.4) | 59.3(2.3) | 70.0(1.0) | **72.7**(0.4) | 66.0(0.4) | 64.0(0.5) | 58.7(1.0) | 74.7(1.0) | 62.1 |
| GEPA | 38.1(2.0) | 46.7(1.1) | 75.8(2.8) | 69.0(3.0) | 57.3(3.2) | 62.0(1.2) | 66.0(1.0) | 68.0(0.7) | 71.3(0.5) | 58.7(0.9) | 51.3(0.4) | 72.7(0.6) | 61.4 |
| GradeOPT | 45.2(2.1) | 46.7(1.3) | 84.9(3.2) | 69.1(1.2) | **64.0**(1.9) | 60.0(1.6) | 72.0(0.4) | 63.3(2.2) | 71.3(1.0) | 60.7(0.6) | 51.3(0.2) | 70.7(1.5) | 63.3 |
| RODS | 45.2(1.8) | **51.1**(1.9) | 78.8(4.0) | 73.8(3.7) | 63.3(2.8) | 60.7(2.9) | **74.0**(1.4) | **72.7**(1.9) | **74.7**(0.8) | **64.7**(0.6) | 53.3(0.2) | **75.3**(1.0) | 65.6 |
| RODS(Semi) | **54.8**(2.1) | **51.1**(1.5) | **90.9**(3.2) | **76.2**(2.4) | **64.0**(1.8) | 62.7(2.5) | 72.7(1.4) | 71.3(1.6) | 72.7(0.8) | 62.7(0.5) | 52.0(0.4) | 72.7(0.9) | **67.0** |
| *Micro F1* | | | | | | | | | | | | | |
| Raw Rubric | 28.9(1.9) | 37.1(1.0) | 68.2(0.8) | 42.7(2.3) | 37.8(2.3) | 57.0(0.4) | 68.7(1.0) | 71.1(0.7) | 64.2(1.1) | 62.1(0.8) | 54.9(0.8) | 71.3(0.2) | 55.3 |
| Few-shot | 57.3(1.1) | 50.7(2.0) | 69.6(0.9) | 64.3(0.9) | 43.9(1.8) | 56.7(1.2) | 71.8(1.3) | 68.2(0.4) | 72.5(0.9) | 65.3(0.5) | **57.6**(0.9) | 70.6(1.2) | 62.4 |
| Few-shot+ | 10.9(1.4) | 29.0(1.4) | 78.9(2.1) | 32.8(1.7) | 45.5(2.1) | **67.8**(1.3) | 64.1(1.1) | 57.4(0.9) | 72.5(0.8) | 55.6(0.9) | 49.9(0.9) | 66.7(1.1) | 52.6 |
| TextGrad | 36.9(1.7) | 48.7(0.5) | 70.6(5.5) | 49.3(1.3) | **64.7**(3.1) | 58.0(2.9) | 68.9(0.6) | **72.0**(1.4) | 63.6(0.5) | 65.4(0.4) | 51.3(0.7) | 73.5(1.5) | 60.2 |
| GEPA | 41.0(2.4) | 50.0(1.3) | 72.9(3.9) | 62.7(3.5) | 59.1(3.4) | 62.6(1.4) | 67.6(1.2) | 67.5(0.8) | 70.1(0.2) | 60.9(0.8) | 53.6(0.6) | 71.0(0.6) | 61.6 |
| GradeOPT | 51.4(2.5) | 49.9(1.1) | 82.0(3.7) | 62.6(1.0) | 63.9(1.9) | 59.1(1.4) | 71.6(0.3) | 63.7(2.0) | 68.5(1.4) | 62.8(0.3) | 54.1(0.1) | 69.1(1.7) | 63.2 |
| RODS | 49.3(2.4) | **56.6**(1.9) | 78.2(4.5) | 73.9(4.0) | 64.5(3.2) | 59.3(2.5) | **73.9**(1.4) | **72.0**(1.6) | **72.9**(0.8) | **65.9**(0.6) | 54.0(0.4) | **74.0**(1.2) | 66.2 |
| RODS(Semi) | **59.4**(2.5) | 54.7(1.4) | **90.4**(3.9) | **75.4**(3.5) | **64.7**(3.2) | 61.2(2.8) | 72.7(0.8) | 71.5(1.4) | 69.3(0.9) | 63.9(0.4) | 53.3(0.6) | 70.4(1.1) | **67.2** |
| *Cohen's Kappa* | | | | | | | | | | | | | |
| Raw Rubric | 0.2(1.3) | 7.6(2.7) | 53.1(1.5) | 17.8(3.5) | 23.4(1.2) | 39.0(0.7) | 49.9(1.5) | 52.5(1.1) | 32.4(1.9) | **40.4**(1.4) | 23.9(0.5) | 48.3(0.4) | 32.4 |
| Few-shot | 20.8(1.5) | 17.8(3.0) | 53.5(1.4) | 45.0(1.8) | 26.2(1.2) | 39.0(1.7) | 54.2(2.1) | 47.5(0.6) | 43.3(1.8) | 38.8(0.8) | 23.7(2.4) | 46.5(1.9) | 38.0 |
| Few-shot+ | -2.0(1.7) | 11.1(2.5) | 65.7(3.5) | 7.0(2.7) | 25.0(3.2) | **53.1**(1.6) | 43.1(1.8) | 33.2(0.9) | **44.9**(1.9) | 32.2(1.2) | 16.0(1.7) | 40.9(1.3) | 30.9 |
| TextGrad | 9.8(2.6) | **24.6**(1.7) | 58.1(8.2) | 27.7(1.7) | 42.6(3.3) | 39.8(3.0) | 49.7(0.9) | 52.5(1.5) | 28.0(0.9) | 38.5(0.7) | 14.3(1.4) | 50.8(2.8) | 36.4 |
| GEPA | 11.0(1.7) | 22.3(1.3) | 58.0(4.7) | 44.3(5.3) | 34.3(4.2) | 45.9(1.8) | 47.3(1.5) | 48.1(1.2) | 40.3(0.3) | 30.8(1.2) | 23.0(0.6) | 46.8(1.1) | 37.7 |
| GradeOPT | 12.3(0.8) | 16.4(2.5) | 74.5(5.2) | 44.1(1.7) | 41.1(1.9) | 39.3(2.2) | 54.0(0.4) | 41.6(2.7) | 36.5(2.5) | 34.1(0.5) | 22.7(0.5) | 43.2(3.0) | 38.3 |
| RODS | 16.5(1.6) | **24.6**(1.6) | 64.9(6.9) | 56.8(5.6) | **44.5**(4.0) | 42.7(3.7) | **58.1**(2.1) | **53.3**(2.7) | 44.8(1.7) | 38.3(1.0) | **26.3**(1.1) | **51.9**(2.0) | 43.6 |
| RODS(Semi) | **22.8**(1.7) | 19.9(1.3) | **85.2**(5.2) | **60.7**(5.3) | 44.0(4.0) | 43.4(3.5) | 55.7(1.8) | 53.1(2.4) | 39.7(1.7) | 36.0(1.0) | 22.5(1.7) | 45.7(1.9) | **44.1** |
| *QWK* | | | | | | | | | | | | | |
| Raw Rubric | 15.3(0.8) | 31.6(0.3) | 64.6(0.9) | 34.9(2.5) | 52.6(0.7) | 68.2(0.8) | 60.4(1.4) | **72.1**(0.6) | 32.4(1.9) | 55.4(1.1) | 35.7(1.3) | **68.3**(0.4) | 49.3 |
| Few-shot | 26.9(1.1) | 38.1(2.2) | 65.8(1.0) | 58.3(1.5) | 54.2(0.2) | 67.2(0.6) | 63.0(1.3) | 68.8(0.3) | 43.3(1.8) | 49.9(1.0) | 26.2(2.3) | 66.5(0.9) | 52.4 |
| Few-shot+ | 5.3(1.9) | 27.9(1.7) | 77.9(2.7) | 38.4(2.6) | 54.5(3.0) | **74.8**(1.2) | 54.9(1.3) | 60.8(0.5) | **44.9**(1.8) | 41.4(0.9) | 22.4(1.9) | 54.5(1.2) | 46.5 |
| TextGrad | 14.8(3.3) | 33.6(0.9) | 68.0(7.4) | 44.5(1.3) | 58.4(2.5) | 67.6(0.8) | 62.3(0.7) | 69.9(1.6) | 28.0(0.9) | 52.7(0.4) | 15.5(2.3) | 67.4(3.3) | 48.6 |
| GEPA | 16.1(1.0) | 37.1(1.8) | 67.0(3.7) | 56.5(4.4) | 61.2(3.1) | 69.8(0.8) | 59.3(0.8) | 69.7(0.9) | 40.3(0.3) | 50.8(0.4) | 33.2(1.6) | 62.1(1.8) | 51.9 |
| GradeOPT | 28.4(2.2) | 34.1(1.9) | 81.9(3.6) | 55.8(1.0) | 62.8(2.0) | 66.5(0.6) | 65.5(0.8) | 63.2(1.6) | 36.5(2.5) | 55.6(1.3) | 37.5(0.6) | 60.2(2.7) | 54.0 |
| RODS | 24.4(1.2) | **44.3**(2.2) | 75.7(5.5) | 71.5(4.4) | **64.2**(2.3) | 67.9(2.6) | 67.1(1.5) | 69.6(0.6) | 44.8(1.7) | **57.9**(1.0) | **41.1**(2.5) | 67.8(1.8) | **58.0** |
| RODS(Semi) | **28.5**(1.2) | 36.4(1.9) | **90.4**(3.8) | **74.0**(4.2) | 60.6(2.5) | 67.3(2.4) | **67.8**(1.3) | 68.4(0.5) | 39.7(1.8) | 55.5(1.1) | 34.2(2.6) | 62.7(1.6) | 57.1 |



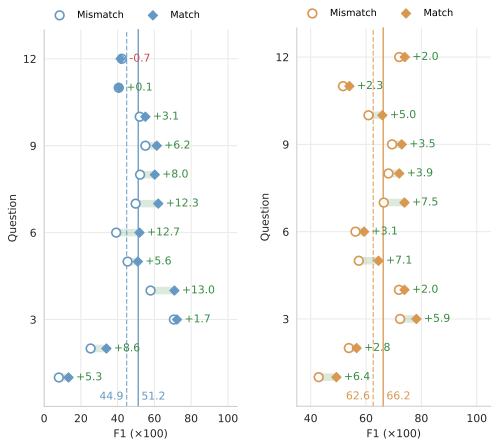

(a) Qwen3-4B-Instruct (b) Gemini-3.1-Flash-Lite (c) Qwen3-4B-Instruct (d) Gemini-3.1-Flash-Lite

Figure 5: Cohen's Kappa, QWK across all questions with two backbone models, comparing matched and mismatched LLMs for grading and data synthesis.

Moreover, the relative contributions of individual components vary across backbones and metrics. Removing Rewrite is less detrimental than removing GAQC on Gemini for Accuracy, Micro F1, and Cohen's Kappa, whereas the opposite pattern holds for Qwen on these metrics. Removing iterative optimization also causes a larger average performance drop than removing consistency control across both backbones and all four metrics. This finding supports the importance of iteratively incorporating supervision signals during optimization. Finally, the Alpha and Label components have particularly pronounced effects. For Qwen, **w/o Alpha** produces the largest average performance drop among all ablated variants on every metric: 10.2 in Accuracy, 4.8 in Micro F1, 7.4 in Cohen's Kappa, and 3.5 in QWK. For Gemini, **w/o Label** produces the largest drops in Accuracy, Micro F1, and QWK with the values 5.7, 6.0, and 5.8, respectively, and ties for the largest drop in Cohen's Kappa at 8.3. Removing either component reduces average performance across all four metrics for both backbones, while their differing impact indicates that the relative importance of Alpha and Label depends on the backbone.

### B.3 Grade and Synthesis Model Match Result

In Fig. 5 and Fig. 6, we present the comparison results under Micro-F1, Cohen's Kappa, and QWK, focusing on whether the grading and synthesis models use the same backbone. Across all questions and evaluation metrics, using the same backbone consistently improves the performance of RODS. This consistent improvement provides strong evidence that matching the grading and synthesis backbones is important for achieving the best performance, and further validates our assumption in § 3.2. Specifically, when the same model is used for both response generation and grading, discrepancies revealed between these two processes can expose weaknesses in the model's understanding of the rubric, thereby providing more informative signals for identifying and correcting deficiencies in the rubric. Moreover, although the mismatched-backbone setting generally yields lower performance than the matched setting, its performance remains comparable to the second-best-performing GradeOpt baseline and consistently outperforms the Few-shot method. This observation suggests that, even when the syn-




(a) Qwen3-4B-Instruct (b) Gemini-3.1-Flash-Lite

Figure 6: Micro-F1 across all questions with two backbone models, comparing matched and mismatched LLMs for grading and data synthesis.

Table 6: Ablation results across various components of RODS across twelve questions from three datasets using *Qwen3-4B-Instruct*. All scores are multiplied by 100. Bold and underline mark the **best** and second-best results in each column within each metric; ties receive the same marking.

| Metric | Method | $D_1$ | | | | $D_2$ | | | | $D_3$ | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | $Q_5$ | $Q_6$ | $Q_7$ | $Q_8$ | $Q_9$ | $Q_{10}$ | $Q_{11}$ | $Q_{12}$ | |
| Accuracy | RODS | **16.7** | **31.1** | **75.8** | **73.8** | **49.3** | **54.0** | **62.0** | **60.7** | 62.7 | **54.0** | 36.0 | 44.7 | **51.7** |
| | w/o Rewrite | 14.3 | 22.2 | 69.7 | 71.4 | 36.0 | 38.7 | 51.3 | 51.3 | 62.0 | **54.0** | 28.0 | **45.3** | 45.4 |
| | w/o GAQC | 14.3 | 26.7 | 60.6 | 71.4 | 41.3 | 48.7 | 52.0 | 41.3 | 58.0 | 50.7 | 40.7 | 43.3 | 45.8 |
| | w/o Iterative | 11.9 | **31.1** | 66.7 | 71.4 | 38.0 | 43.3 | 52.7 | 36.0 | 59.3 | 47.3 | 36.0 | 43.3 | 44.8 |
| | w/o Consist | 14.3 | 26.7 | 69.7 | 69.1 | 37.3 | 39.3 | 52.0 | 48.0 | **64.7** | 50.0 | **42.0** | 43.3 | 46.4 |
| | w/o Alpha | 9.5 | 20.0 | 54.5 | 57.1 | 37.3 | 45.3 | 47.3 | 39.3 | 60.0 | 48.7 | 35.3 | 44.0 | 41.5 |
| | w/o Label | 11.9 | 22.2 | 66.7 | 61.9 | 39.3 | 46.0 | 55.3 | 50.0 | 62.0 | 48.7 | 34.0 | 44.0 | 45.2 |
| Micro F1 | RODS | **13.3** | **33.9** | **72.3** | **70.9** | **51.0** | **51.9** | **62.1** | **60.2** | 61.3 | 55.1 | 40.7 | 41.6 | **51.2** |
| | w/o Rewrite | 9.1 | 21.2 | 63.4 | 68.6 | 38.3 | 37.9 | 51.1 | 52.2 | 58.4 | **55.2** | 37.2 | **43.7** | 44.7 |
| | w/o GAQC | 8.7 | 28.6 | 53.6 | 67.8 | 45.5 | 49.2 | 52.7 | 41.2 | 57.0 | 54.0 | 45.9 | 42.3 | 45.5 |
| | w/o Iterative | 7.9 | 33.2 | 60.8 | 68.6 | 41.8 | 42.1 | 52.2 | 32.0 | 58.5 | 50.8 | 46.1 | 41.8 | 44.6 |
| | w/o Consist | 8.9 | 29.9 | 63.7 | 64.3 | 41.5 | 38.8 | 52.4 | 48.5 | **63.6** | 51.6 | **50.7** | 42.9 | 46.4 |
| | w/o Alpha | 3.6 | 16.7 | 44.0 | 47.5 | 41.3 | 43.7 | 46.2 | 34.0 | 58.8 | 51.4 | 43.0 | 43.1 | 39.5 |
| | w/o Label | 8.1 | 22.0 | 59.6 | 58.4 | 42.2 | 44.4 | 54.8 | 50.2 | 60.1 | 51.5 | 41.2 | 42.7 | 44.6 |
| Cohen's Kappa | RODS | **-2.5** | **11.2** | **58.5** | **52.4** | **20.0** | **33.2** | **41.0** | **40.0** | 25.0 | **21.4** | 15.3 | **10.9** | **27.2** |
| | w/o Rewrite | -7.9 | 5.8 | 47.0 | 48.2 | 5.4 | 8.7 | 26.7 | 29.9 | 22.7 | 18.0 | 10.5 | 10.7 | 18.8 |
| | w/o GAQC | -4.3 | 5.7 | 30.1 | 49.6 | 13.3 | 24.4 | 28.4 | 24.6 | 18.6 | 19.3 | 18.4 | 6.6 | 19.6 |
| | w/o Iterative | -8.9 | 10.6 | 41.2 | 47.8 | 10.3 | 16.2 | 30.9 | 19.7 | 21.7 | 18.7 | 14.7 | 7.5 | 19.2 |
| | w/o Consist | -5.1 | 6.6 | 47.9 | 42.1 | 8.4 | 11.0 | 29.0 | 27.9 | **26.9** | 18.4 | **19.2** | 5.4 | 19.8 |
| | w/o Alpha | -9.0 | 3.6 | 18.7 | 17.9 | 10.3 | 18.1 | 22.2 | 22.7 | 23.2 | 17.5 | 14.5 | 7.4 | 13.9 |
| | w/o Label | -9.3 | 2.4 | 41.6 | 34.0 | 11.1 | 18.5 | 30.7 | 27.2 | 25.2 | 17.3 | 14.7 | 8.3 | 18.5 |
| QWK | RODS | 14.4 | **21.7** | **69.0** | 62.3 | **41.5** | **58.6** | **58.3** | **57.7** | 25.0 | 40.0 | 35.4 | 16.8 | **41.7** |
| | w/o Rewrite | 9.3 | 19.1 | 57.2 | 59.3 | 37.5 | 41.0 | 50.6 | 50.9 | 22.7 | 39.5 | 33.1 | **17.7** | 36.5 |
| | w/o GAQC | 15.9 | 17.5 | 39.9 | **62.6** | 38.2 | 53.5 | 51.1 | 51.9 | 18.6 | 40.4 | 35.4 | 15.0 | 36.7 |
| | w/o Iterative | 16.3 | 19.7 | 50.7 | 58.4 | 36.5 | 45.2 | 48.4 | 50.7 | 21.7 | 39.2 | 38.0 | 14.2 | 36.6 |
| | w/o Consist | **19.2** | 20.5 | 58.3 | 51.2 | 36.4 | 46.7 | 54.6 | 50.6 | **26.9** | 37.6 | **41.3** | 15.2 | 38.2 |
| | w/o Alpha | 9.3 | 16.5 | 27.1 | 26.9 | 37.4 | 50.3 | 53.3 | 50.0 | 23.2 | 38.5 | 34.3 | 15.9 | 31.9 |
| | w/o Label | 8.7 | 14.1 | 51.6 | 52.2 | 40.4 | 45.1 | 52.8 | 56.5 | 25.2 | **41.3** | 32.8 | 15.0 | 36.3 |

Table 7: Ablation results across various components of RODS across twelve questions from three datasets using *Gemini-3.1-Flash-Lite*. All scores are multiplied by 100. Bold and underline mark the **best** and second-best results in each column within each metric; ties receive the same marking.

| Metric | Method | $D_1$ | | | | $D_2$ | | | | $D_3$ | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ | $Q_5$ | $Q_6$ | $Q_7$ | $Q_8$ | $Q_9$ | $Q_{10}$ | $Q_{11}$ | $Q_{12}$ | |
| Accuracy | RODS | **45.2** | **51.1** | 78.8 | **73.8** | **63.3** | 60.7 | **74.0** | **72.7** | **74.7** | 64.7 | 53.3 | **75.3** | **65.6** |
| | w/o Rewrite | **45.2** | 37.8 | **81.8** | 66.7 | 58.7 | 57.3 | 73.3 | 70.0 | 69.3 | 58.0 | 54.0 | 74.7 | 62.2 |
| | w/o GAQC | 33.3 | 40.0 | 75.8 | **73.8** | 59.3 | **62.0** | 67.3 | 68.0 | 73.3 | 61.3 | 52.7 | 70.7 | 61.5 |
| | w/o Iterative | 33.3 | 42.2 | 72.7 | 66.7 | 60.7 | 52.0 | 70.0 | 68.7 | 72.7 | 62.7 | **54.7** | 72.7 | 60.7 |
| | w/o Consist | 40.5 | 44.4 | 72.7 | 64.3 | 60.0 | 58.7 | 70.0 | 70.7 | 73.3 | **65.3** | 50.0 | 71.3 | 61.8 |
| | w/o Alpha | 31.0 | 44.4 | 78.8 | 57.1 | 55.3 | 55.3 | 72.7 | 70.0 | 70.0 | 63.3 | 53.3 | 74.7 | 60.5 |
| | w/o Label | 33.3 | 33.3 | 75.8 | 64.3 | 55.3 | 60.7 | 72.0 | 62.0 | 73.3 | 61.3 | 52.7 | 74.7 | 59.9 |
| Micro F1 | RODS | 49.3 | **56.6** | 78.2 | **73.9** | **64.5** | 59.3 | **73.9** | **72.0** | **72.9** | 65.9 | 54.0 | **74.0** | **66.2** |
| | w/o Rewrite | **50.8** | 42.8 | **79.1** | 62.3 | 59.7 | 56.9 | 73.1 | 69.3 | 67.5 | 59.5 | 54.0 | 72.6 | 62.3 |
| | w/o GAQC | 34.8 | 45.2 | 76.2 | 72.3 | 59.1 | **62.2** | 66.1 | 67.0 | 71.1 | 63.5 | 55.1 | 68.6 | 61.8 |
| | w/o Iterative | 39.4 | 49.0 | 71.7 | 62.0 | 61.1 | 49.6 | 69.4 | 67.9 | 70.8 | 63.3 | 54.3 | 71.1 | 60.8 |
| | w/o Consist | 45.0 | 51.2 | 69.6 | 60.7 | 60.1 | 58.9 | 69.9 | 69.1 | 71.0 | **66.1** | 53.2 | 69.7 | 62.0 |
| | w/o Alpha | 35.8 | 47.4 | **79.1** | 52.6 | 56.5 | 55.8 | 72.4 | 69.3 | 67.5 | 64.4 | **55.9** | 73.6 | 60.9 |
| | w/o Label | 36.3 | 41.7 | 71.8 | 60.0 | 55.2 | 61.9 | 71.6 | 61.4 | 71.6 | 63.0 | 54.6 | 73.5 | 60.2 |
| Cohen's Kappa | RODS | **16.5** | **24.6** | 64.9 | **56.8** | **44.5** | 42.7 | **58.1** | **53.3** | **44.8** | 38.3 | 26.3 | **51.9** | **43.6** |
| | w/o Rewrite | 12.7 | 9.8 | **69.4** | 41.8 | 35.0 | 37.2 | 56.8 | 50.6 | 33.7 | 33.6 | 23.9 | 49.7 | 37.9 |
| | w/o GAQC | 5.8 | 10.7 | 61.3 | 55.2 | 34.1 | **45.9** | 45.1 | 45.7 | 41.4 | 37.6 | 23.8 | 43.5 | 37.5 |
| | w/o Iterative | 2.8 | 13.4 | 55.9 | 42.0 | 38.5 | 28.7 | 51.2 | 47.1 | 40.4 | 34.6 | 23.9 | 46.6 | 35.4 |
| | w/o Consist | 12.8 | 13.8 | 53.1 | 41.4 | 36.8 | 40.8 | 51.6 | 49.8 | 41.4 | **41.8** | 21.7 | 43.9 | 37.4 |
| | w/o Alpha | 3.4 | 21.0 | 65.8 | 28.3 | 35.0 | 36.1 | 55.8 | 49.5 | 34.2 | 36.1 | **26.9** | 51.1 | 36.9 |
| | w/o Label | 5.7 | -0.4 | 59.2 | 38.7 | 29.8 | 45.4 | 55.0 | 38.7 | 41.9 | 33.5 | 25.2 | 51.4 | 35.3 |
| QWK | RODS | 24.4 | **44.3** | 75.7 | **71.5** | **64.2** | 67.9 | 67.1 | **69.6** | **44.8** | **57.9** | 41.1 | 67.8 | **58.0** |
| | w/o Rewrite | 24.1 | 29.3 | **78.8** | 57.7 | 56.6 | 65.9 | 65.4 | 66.6 | 33.7 | 51.2 | 35.3 | 65.7 | 52.5 |
| | w/o GAQC | 10.7 | 33.5 | 75.6 | 68.3 | 53.8 | **72.7** | 56.8 | 64.1 | 41.4 | 54.2 | 38.8 | 62.4 | 52.7 |
| | w/o Iterative | **25.4** | 33.0 | 70.8 | 58.0 | 58.0 | 60.9 | 63.2 | 64.9 | 40.4 | 55.3 | 34.3 | 62.8 | 52.2 |
| | w/o Consist | 22.0 | 41.3 | 62.4 | 61.3 | 59.1 | 63.4 | 64.9 | 68.5 | 41.4 | 54.1 | 36.2 | 63.1 | 53.1 |
| | w/o Alpha | 18.4 | 36.5 | 78.1 | 50.6 | 60.4 | 66.4 | 66.1 | 69.2 | 34.2 | 52.9 | **42.7** | 68.1 | 53.6 |
| | w/o Label | 17.4 | 21.7 | 70.0 | 56.4 | 54.8 | 71.1 | **68.1** | 63.5 | 41.9 | 55.3 | 37.4 | **69.0** | 52.2 |

thesis and grading models are not matched, the generated synthetic data can still provide useful supervision for rubric optimization, albeit less effectively than in the matched-backbone setting.

### B.4 CASE STUDY ON GAQC

In Fig. 7, we present a case study examining the influence of GAQC on synthetic response quality. The case study includes three groups of responses: (i) real student responses, (ii) synthetic responses generated through direct prompting, and (iii) synthetic responses refined using GAQC. In addition, Fig. 8 provides an example of the distinguishing feature $\omega$ identified by GAQC. From these two figures, we make the following observations. By comparing the linguistic characteristics across the three groups, we observe that real student responses are generally concise and focused on key scoring points, whereas synthetic responses generated without GAQC tend to be overly lengthy and detailed, exhibiting noticeable distributional differences from authentic student responses. As illustrated by the distinguishing feature $\omega$, the model is able to identify and articulate salient differences between real and synthetic responses. Such information provides explicit guidance to the synthesis model for refining subsequent generations toward responses that more closely resemble authentic student answers. Consistent with this guidance, responses refined through GAQC become substantially more similar to real student responses in both style and structure. This improved alignment may facilitate more effective consistency-based optimization and contribute to better grading performance on real student responses.

**Synthetic responses w/o GAQC:**

- Score=0 response: The student does not provide a response.

- Score=1 response: The photo shows a mountain and a river. The mountain has rocks and the river flows near it. The river water moves the rocks and breaks them down. This changes the land over time.

- Score=2 response: I see a river and a mountain in the photo. The river flows from the mountain and carries water down. Over time, the water wears away the mountain side, making it smaller. This is how the land changes.

- Score=3 response: In the photo, I see a river and a mountain. The river flows down from the mountain, carrying water that breaks down the rock over time. This process, called weathering, slowly wears away the mountain's sides. The water also moves sand and small rocks from the mountain to the riverbed, which is erosion. Over time, this causes the mountain to get smaller and the riverbed to build up. This shows how the river and mountain interact to change the land.

**Synthetic responses w/ GAQC:**

- Score=0 response: I saw mountain range and valleys.

- Score=1 response: The land and the mountain might go into the ground so it will be flat land.

- Score=2 response: weathering and erosion happen together and make the mountain smaller over time, like how rain and wind break down the rock.

- Score=3 response: Water and land change over time. Water hits the mountains and wears them down. The land gets smaller as water flows and breaks the rocks.

**True Student responses:**

- Score=0 response: it might be cold up in it but the only cold thing that I see is the peak of mount everest, it's 32 degrees F at the peak

- Score=1 response: There is a lot of bumpie parts and there is some snow on the back side of the mountains.

- Score=2 response: The dirt and clouds could work together. The clouds could rain on the dirt to make grass because dirt needs water and the sun to grow grass

- Score=3 response: Water can splash up and hit the grass so many time it will erod the land and rocks can hold the land.

Figure 7: Comparisons of responses generated w/ and w/o GAQC over $\mathrm{Q}_5$.

**True responses**

['mentions of water melting from snow or ice', 'references to mountains and valleys as interconnected parts', 'use of informal or fragmented sentence structures', "repetition of terms like 'water,' 'land,' 'mountains,' 'valley'", 'focus on physical processes like erosion and weathering without clear cause-effect chains']

**Synthetic responses**

['explicit explanation of erosion and deposition processes', 'clear identification of system components and their interactions', 'use of formal and scientific language', 'structured cause-effect relationships between parts of a system', 'specific reference to river carrying eroded rocks and shaping landscapes']

Figure 8: Example of distinguishing feature $\omega$ between true and synthetic responses over $\mathrm{Q}_5$.

# C Additional Method Details

## C.1 Data Synthesis Algorithm

The overall data synthesis procedure is summarized in Alg. 1. From lines 1 to 3, the algorithm first generates the initial synthetic samples and identifies the most distinguishing features between real and synthetic responses. Then, in line 6, each synthetic sample is classified to determine whether it is distinguishable from real student responses. Samples that are already indistinguishable are directly retained. For the remaining samples, lines 7 to 9 iteratively rewrite them while explicitly avoiding the identified distinguishing features. This refinement process continues until the sample becomes indistinguishable from real responses or the maximum number of iterations is reached.

**Algorithm 1:** Data Synthesis with GAQC

**Input:** Rubric $\phi$; question $q$; response dataset $\mathcal{D}_r$; LLM $f_\theta$; number of improving iterations $S$, quality filtering threshold $\mu$;
**Output:** Synthetic Dataset $\mathcal{D}_s^*$

1 Generate candidate synthetic dataset: $\mathcal{D}_s$ via Eq. 3;
2 Initialize qualified dataset: $\mathcal{D}_s^* \leftarrow \emptyset$;
3 Summarize the distinguishing feature: $\omega$ via Eq. 4;
4 **for** $s = 1$ **to** $S$ **do**
5  **foreach** *synthetic response* $\hat{x}_y \in \mathcal{D}_s$ **do**
6  Calculate classification accuracy: $d_\omega$ via Eq. 4;
7  **if** $d_\omega < \mu$ **then**
8  Update the qualified dataset: $\mathcal{D}_s^* \leftarrow \mathcal{D}_s^* \cup \{\hat{x}_y\}$;
9  **end**
10  **end**
11  Update the candidate synthetic dataset: $\mathcal{D}_s$ via Eq. 5;
12 **end**
13 **return** $\mathcal{D}_s^*$

## C.2 Rubric Optimization Algorithm

The overall optimization procedure is summarized in Alg. 2. From lines 4 to 7, the algorithm performs grading and reflection on the labeled samples. Based on the reflection results, a refined rubric is generated. Then, from lines 8 to 11, a set of synthetic samples is generated, and the refined rubrics are used to grade these synthetic examples. By jointly evaluating the consistency between synthetic labels and predicted scores, together with the grading accuracy on labeled samples, the algorithm computes a ranking score to determine which candidate rubrics in the beam are retained for the next stage. Next, from lines 13 to 16, the algorithm uses the updated rubrics to generate new synthetic samples and further refines the rubrics based on consistency-based supervision. The refined rubrics are then evaluated again on both synthetic and labeled samples, and the ranking scores are recomputed to determine which candidates are preserved for the next iteration. The optimization process continues until the maximum number of optimization iterations is reached.

**Algorithm 2:** RODS: Rubric Optimization via Data Synthesis

**Input:** Initial rubric $\phi_0$; question $q$; labeled dataset $\mathcal{D}_r$; LLM $f_\theta$; beam size $k$; rubric optimizing iterations $R$; synthetic improving iterations $S$; balance factor $\alpha$; filtering threshold $\mu$

**Output:** Optimized rubric $\phi^*$

Initialize beam set $\Phi_0 \leftarrow \{\phi_0\}$;
**for** $r = 1$ **to** $R$ **do**
  Initialize candidate rubric set $\Phi'_c \leftarrow \emptyset$;
  **foreach** *rubric* $\phi \in \Phi_{r-1}$ **do**
    Generate label-supervised refinement: $\Phi'$ via Eq. 6;
    Update candidate rubric set: $\Phi'_c \leftarrow \Phi'_c \cup \Phi'$;
  **end**
  **foreach** *rubric* $\phi \in \Phi'_c$ **do**
    Generate candidate synthetic dataset: $\mathcal{D}_s$ via Alg. 1;
    Calculate Index($\phi$) via Eq. 9;
  **end**
  Retain top-$k$ rubrics: $\Phi'_c \leftarrow \mathrm{TopK}(\Phi'_c, k)$;
  **foreach** *rubric* $\phi \in \Phi'_c$ **do**
    Generate candidate synthetic dataset: $\mathcal{D}_s$ via Alg. 1;
    Generate consistency-supervised refinement: $\Phi'$ via Eq. 8;
    Update candidate rubric set: $\Phi'_c \leftarrow \Phi'_c \cup \Phi'$;
  **end**
  **foreach** *rubric* $\phi \in \Phi'_c$ **do**
    Generate candidate synthetic dataset: $\mathcal{D}_s$ via Alg. 1;
    Calculate Index($\phi$) via Eq. 9;
  **end**
  Retain top-$k$ rubrics: $\Phi'_r \leftarrow \mathrm{TopK}(\Phi'_c, k)$;
**end**
Select final rubric: $\phi^* \leftarrow \arg\max_{\Phi_R} \mathrm{Index}(\phi)$;
**return** $\phi^*$

Table 8: Statistics of the three datasets across subjects and grade levels.

| Dataset ID | Subject | Grade Level | Question Count | Response Count |
|---|---|---|---|---|
| $D_1$ | Artificial Intelligence | College | 4 | {$Q_1$:34, $Q_2$:35, $Q_3$:31, $Q_4$:34} |
| $D_2$ | Earth Science | Primary School | 4 | {$Q_5$:199, $Q_6$:230, $Q_7$:249, $Q_8$:267} |
| $D_3$ | Chemistry | Middle School | 4 | {$Q_9$:156, $Q_{10}$:156, $Q_{11}$:177, $Q_{12}$:177} |

# D Dataset and Prompt Details

## D.1 Question and Rubric Details

Detailed dataset statistics are provided in Tab. 8. For each question, an initial scoring rubric ($\phi_0$) is designed by domain experts, with the number of score levels ranging from 2 to 5. Each rubric follows a structured format in which every score level contains a detailed description of the corresponding grading criteria. Examples of the questions and rubrics for $D_1$, $D_2$, and $D_3$ are shown in Fig. 9, Fig. 10, and Fig. 11, respectively. Due to page limitations, some contents are omitted, while complete details are available in the source files released with the code.

## D.2 Prompt Details for Baselines and RODS

To ensure fair comparisons, all methods share the same grading task instruction prompt $\pi_g$, shown as Fig. 12, and differ only in the rubric used for scoring. For both GradeOpt and RODS, we adopt the same rubric refinement prompt $\pi'_r$, shown as Fig. 13. For the synthetic data generation stage of RODS, we design three prompts: $\pi_s$, shown as Fig. 14, for sample rewriting, $\pi_c$, shown as Fig. 16, for distinguishing feature extraction, and $\pi_d$, shown as Fig. 15, for real-versus-synthetic response classification.

**$\mathbf{Q}_1$**

**Question stem ($q$):**
Assume you want to binary classify linear separable data point using a perceptron. Write down the learning algorithm in pseudo code!

**Raw-rubric ($\phi_0$):**
Review the student's response and compare it with the reference (gold) answer: Label the data with positive and negative (+/-) labels, initialize the weights randomly, apply (simplified) update rule: Dw = eta*x(n) if ¡w,x¿ ¡= 0, repeat on all epochs till the weights don't change much. The algorithm will converge as the data is linearly separable. Assign a score of 0, 1, or 2 based on the following criteria:
- Score 2: Fully correct answer.
- Score 1: Partially correct answer.
- Score 0: Incorrect answer.

**$\mathbf{Q}_2$**

**Question Stem ($q$):**
Define: a hypothesis $h \in H$ shatters a dataset $A \subseteq X \Leftrightarrow \ldots$

**Raw Rubric ($\phi_0$):**
Review the student's response and compare it with the reference (gold) answer: For all $2^N$ possible binary labeling of every data, if a hypothesis h splits the positive data from the negative data with no error, then it means that the hypothesis h shatters the dataset A. Assign a score of 0, 1, or 2 based on the following criteria:
- Score 2: Fully correct answer.
- Score 1: Partially correct answer.
- Score 0: Incorrect answer.

**$\mathbf{Q}_3$**

**Question Stem ($q$):**
How does a Reduced Boltzman Machine work (main idea)?

**Raw Rubric ($\phi_0$):**
Review the student's response and compare it with the reference (gold) answer: The reduced Boltzmann machine is a bi-parted graph which works by flipping the states of binary neurons based on a probability determined by the activation produced at the neuron. Neurons are arranged in a visible and a hidden layer in a recurrent fashion. There are two states involved called the clamped state in which the visible neuron is connected to the input and a free running state in which both layers run free. Assign a score of 0, 1, or 2 based on the following criteria:
- Score 2: Fully correct answer.
- Score 1: Partially correct answer.
- Score 0: Incorrect answer.

**$\mathbf{Q}_4$**

**Question Stem ($q$):**
What are three items to be learned for an RBF network? Difference to other NNs, Pros/cons?

**Raw Rubric ($\phi_0$):**
Review the student's response and compare it with the reference (gold) answer: Three items to be learned are centers, weights, and biases. RBFN consists of a single hidden layer and a linear output layer. NN can have multiple hidden layers and a linear or non-linear output layer. Pros: RBFN is a universal approximator and it is easy to add more centers. Con: The bias is not unique. Assign a score of 0, 1, or 2 based on the following criteria:
- Score 2: Fully correct answer.
- Score 1: Partially correct answer.
- Score 0: Incorrect answer.

Figure 9: Example question stem text $q$ and raw rubric $\phi_0$ of four questions in $\mathrm{D}_1$.

**$\mathbf{Q}_5$**

**Question stem ($q$):**
Look closely at the photo in a system. Think about parts or components that work together. Identify two or three parts of the photo in a system and explain how they work together to cause change over time.

**Raw-rubric ($\phi_0$):**
Review and apply the scoring criteria for grading student explain answer and give a score of 0, 1, 2, 3 or 4.
- Score 4: Complete correct claim and reasoning . . .
- Score 3: Correct claim and partial evidence and reasoning . . .
- Score 2: Correct claim; but no or incorrect reasoning . . .
- Score 1: no claim, or claim and reasoning are incorrect. . .
- Score 0: tudent does not provide a response . . .

**$\mathbf{Q}_6$**

**Question Stem ($q$):**
What evidence of energy do you see in the system shown in the photo? Explain your thinking.

**Raw Rubric ($\phi_0$):**
Review and apply the scoring criteria for grading student explain answer and give a score of 0, 1, 2, 3 or 4.
- Score 4: Correct evidence and reasoning . . .
- Score 3: Correct evidence, but reasoning is not clear including correct and incorrect ideas . . .
- Score 2: Correct evidence, but no or incorrect reasoning . . .
- Score 1: Incorrect evidence and no or incorrect reasoning . . .
- Score 0: Student does not provide a response . . .

**$\mathbf{Q}_7$**

**Question Stem ($q$):**
Consider what you learned in class about transferring energy. Use the diagram above to think about what change over time might be causing the decrease in the production of electricity. Explain your thinking. Describe what you think may have happened over time to cause less energy to be transferred from the moving water to the generator. Use evidence and reasoning.

**Raw Rubric ($\phi_0$):**
Review and apply the scoring criteria for grading student explain answer and give a score of 0, 1, 2, 3 or 4.
- Score 4: The student's response includes ALL three parts listed below: . . .
- Score 3: The student's response includes part 1 below and either part 2 or 3 to describe the. . .
- Score 2: The student's response includes any one of the following parts . . .
- Score 1: The student's response does not reflect an accurate change that causes less water to move through . . .
- Score 0: Student does not provide a response . . .

**$\mathbf{Q}_8$**

**Question Stem ($q$):**
Think about the diagram of the dam system and how it generates energy. What changes to the river and the existing dam system could increase the amount of energy available to power homes? Design a solution to increase the amount of energy available to power homes. Make sure you explain how the changes you suggest to the river and the existing dam system increase the transfer of energy from the moving water to the generator.

**Raw Rubric ($\phi_0$):**
Review and apply the scoring criteria for grading student explain answer and give a score of 0, 1, 2, 3 or 4.
- Score 4: The student's answer includes all 3 parts below. Part 1: The student suggests . . .
- Score 3: The student's answer includes 2 of the 3 parts below. Part 1: The student suggests . . .
- Score 2: The student's answer includes 1 part below. Part 1: The student suggests a partial solution . . .
- Score 1: The student does not suggest a solution or suggests an incorrect solution that . . .
- Score 0: Student does not provide a response . . .

Figure 10: Example question stem text $q$ and raw rubric $\phi_0$ of four questions in $\mathrm{D}_2$.

**$Q_9$**

**Question stem ($q$):**
Miranda found four different unlabeled bottles filled with liquids in the supply closet of her science classroom. She wondered if the liquids were the same or different. She measured the mass, volume, and boiling point of these liquids. She used mass and volume to calculate the density of each liquid. She arranged these data and the density calculations into a table (See the table below).
— Sample — Mass — Volume — Density — Boiling Point —
———————————————————————————
— 1 — 6.10g — 6.10 cm³ — 1.00 g/cm³ — 100°C —
— 2 — 5.43g — 6.10 cm³ — 0.890 g/cm³ — 211°C —
— 3 — 9.38g — 10.2 cm³ — 0.920 g/cm³ — 300°C —
— 4 — 9.08g — 10.2 cm³ — 0.890 g/cm³ — 211°C —
Write a scientific explanation about whether any of the liquids are the same substance? Make sure your explanation includes a claim, evidence, and reasoning.

**Raw-rubric ($\phi_0$):**
Review and apply the scoring criteria for grading student answer and give a level score of 0, 1.
- Level 1: In this task, we are looking for students who address if two samples have at least two properties . . .
- Level 0: If the student's answer does not meet the requirements of level 1, it will be classified as level 0 . . .

**$Q_{10}$**

**Question Stem ($q$):**
(Same with the question stem of $Q_9$)

**Raw Rubric ($\phi_0$):**
Review and apply the scoring criteria for grading student answer and give a level score of 0, 1, or 2.
- Level 2: Relational–correlational explanations should identify patterns within the data and establish . . .
- Level 1: The descriptive–coincidental explanation identifies patterns in the data and provides . . .
- Level 0: If the student's answer does not meet the requirements of level 1 or level 2 . . .

**$Q_{11}$**

**Question Stem ($q$):**
Danny wondered what would happen if he mixed coconut oil with lye and heated them together. Danny then decided to mix the coconut oil and lye together with the help of his science teacher to see if he could create soap. After heating, Danny set the mixture aside for a few days. Danny soon observed that a soap-like substance had formed as well as an oily product. He measured and calculated some properties of the coconut oil and lye before heating and the soap-like substance and oily material that formed after heating. He arranged his data into a table. Danny wondered if the process of making soap involved a chemical reaction.
Table 1: Data for substances before and after heating.
— State — Sample — Mass — Odor — Density — Solubility in Water — Melting Point —
———————————————————————————————————————
— Before — Coconut oil — 100g — Coconut — 0.93 g/cm³ — No — 27°C —
— Before — Lye — 20g — Odorless — 2.13 g/cm³ — Yes — 318°C —
— After — Soap — 115g — Coconut — 0.95 g/cm³ — Yes — 48°C —
— After — Glycerol — 5g — Odorless — 1.26 g/cm³ — No — 17.8°C —
Write a scientific explanation about whether a chemical reaction occurs when coconut oil is mixed with lye and heated together. Make sure your explanation includes a claim, evidence, and reasoning.

**Raw Rubric ($\phi_0$):**
Review and apply the scoring criteria for grading student answer and give a level score of 0, 1, or 2.
- Level 2: Substances react chemically in characteristic ways that the new substances have . . .
- Level 1: Under a given set of conditions, the state and some properties (e.g., density, odor, . . .
- Level 0: If the student's answer does not meet the requirements of level 1 or level 2 . . .

**$Q_{12}$**

**Question Stem ($q$):**
(Same with the question stem of $Q_{11}$)

**Raw Rubric ($\phi_0$):**
Review and apply the scoring criteria for grading student answer and give a level score of 0, 1, or 2.
- Level 2: The relational–correlational explanations could identify patterns within the data and . . .
- Level 1: The descriptive–coincidental explanation identifies patterns in the data and . . .
- Level 0: If the student's answer does not meet the requirements of level 1 or level 2 . . .

Figure 11: Example question stem text $q$ and raw rubric $\phi_0$ of four questions in $D_3$.

**Grading prompt $\pi_g$**

Given the question, student response, scoring rubric and example answers for each score level, evaluate the response and provide:
- score: a numerical score based on the rubric.
- explanation: a concise explanation describing the reasoning behind your score.
# Question
{question}
# Student Response
{response}
# Scoring Rubric
{rubric}
# Example Answers
{example}

Figure 12: Universal grading instruction prompt using for LLM-base AG method.

**Rubric refining prompt $\pi_r'$**

You are an expert Rubric Revision Model specialized in improving grading rubrics for automatic assessment tasks. You are given:
- The evaluation question
- The current grading rubric
- A band confusion matrix indicating where score levels collapse or diverge.
- Examples of misgraded responses (false positives and false negatives)
- Reference responses with correct scores for comparison
Your goal: Revise the rubric so that the misgraded examples would be correctly evaluated, while improving consistency, fairness, and realism.
—
# Question
{question}
# Current Rubric
{rubric}
# Confusion Matrix
{matrix}
# Misgraded Examples
{example}
# Reference Responses
{reference}
—
Now, please revise the rubric according to the information above. Make only the necessery and meaningful changes that improve clarity, fairness, and score boundary definitions. Avoid adding unnecessary complexity.

Figure 13: Universal rubric refining instruction prompt using for GradeOpt and RODS.

**Response rewriting prompt $\pi_s$**

Revise the student response to the question so that it would earn the specified target score according to the provided scoring rubric. Maintain the style and tone of the reference sample and make only minimal, necessary edits to align with the target score criteria. Return the updated response in the same JSON field as the input response.
# Question
{question}
# Scoring Rubric
{rubric}
# Target Score
{score}
# Student Response
{answer}

Figure 14: Response rewriting instruction prompt using for RODS.

**Distinguishing feature comparing instruction prompt $\pi_c$**

Given two groups of answers, analyze their linguistic characteristics and identify up to five features for each group that are common within the group but distinct from the other group. Do not reference the group names in the description. Present the results as two separate lists of features.
# Group 1 Answer
{g1_answer}
# Group 2 Answer
{g2_answer}

Figure 15: Distinguishing feature comparing instruction prompt using for GAQC of RODS.

**Synthetic response identifying instruction prompt $\pi_d$**

Given the linguistic features of two groups and a target text, analyze the target text and determine which group it more closely aligns with. Base your decision on the similarity between the target text's linguistic characteristics and the provided group features.
# Group 1 Feature
{g1_feature}
# Group 2 Feature
{g2_feature}
# Target Text
{target}

Figure 16: Synthetic response identifying instruction prompt using for GAQC of RODS.

**Distinguishing feature avoided response rewriting prompt $\pi'_s$**

Revise the student response to the question so that it would earn the specified target score according to the provided scoring rubric. Maintain the style and tone of the reference sample and make only minimal, necessary edits to align with the target score criteria. Additionally, align the revised response with the linguistic features of Group 1 and avoid the features of Group 2 whenever possible, provided that doing so does not conflict with the scoring requirements. If any group feature is None, ignore the corresponding group alignment or avoidance requirement. Return the revised response in the same JSON field as the original student response.
# Question
{question}
# Scoring Rubric
{rubric}
# Group 1 Features
{g1_feature}
# Group 2 Features
{g2_feature}
# Target Score
{score}
# Student Response
{response}

Figure 17: Distinguishing feature avoided response rewriting prompt using for GAQC of RODS.