REC-CBM: Rubric-Aware Error-Correction Concept Bottleneck Models for Trustworthy Open-Ended Grading

Chengshuai Zhao [1 *] **Fan Zhang** [1 *] **Kumar Satvik Chaudhary** [1] **Yiwen Li** [2] **Lo Pang-Yun Ting** [3] **Ying-Chih Chen** [2] **Huan Liu** [1]

Code: https://github.com/scott-f-zhang/REC-CBM Data: https://huggingface.co/datasets/scott-f-zhang/REC-CBM

Abstract

Open-ended grading is important for personalized education, but manual grading is costly to scale. Although neural and large language model based graders achieve strong performance, their black-box nature limits educators' ability to verify scoring rationales. Concept bottleneck models offer transparency through human-interpretable concepts, yet standard CBMs do not capture rubric structure, ordinal scoring semantics, or noise in concept annotations. We propose REC-CBM, a rubric-aware error-correction concept bottleneck model for trustworthy open-ended grading. REC-CBM introduces a rubric-aware concept encoder that learns concept-specific representations over responses and an ordinal pairwise calibration objective that preserves ranking structure among rubric dimensions. It further incorporates a latent concept error-correction module that denoises concept predictions before final grade prediction while preserving interpretability. Experiments on public datasets show that REC-CBM improves grading accuracy and produces more faithful concept-level reasoning than strong black-box and transparent baselines.

1. Introduction

Open-ended grading evaluates students' written responses to open questions, which plays a fundamental role in education across subjects and levels. At the same time, evaluating free-text responses at scale is labor-intensive and expensive, especially in settings that require timely feedback across large numbers of students (Clauser et al., 2024). This tension has made automated grading an important topic for educational data mining and AI for education communities (Attali & Burstein, 2006).

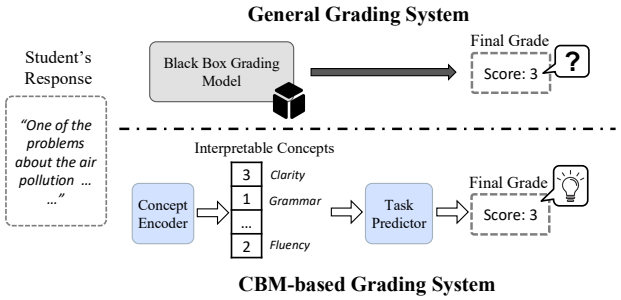

Figure 1. Comparison of open-ended grading systems.

Recent neural and large language model based grading systems have substantially improved predictive performance (Wang et al., 2022; Golchin et al., 2025), making automated scoring increasingly viable in practice. However, educational assessment is a high-stakes domain where raw accuracy alone is insufficient. When a system assigns a score to a student's response, instructors need to understand why that score was produced, whether it aligns with the intended rubric, and how to intervene when the model makes a questionable judgment. The black-box nature of these models makes it difficult for educators to verify the reasoning behind a grade, contest, or audit specific decisions, and provide feedback that is pedagogically useful (Wang et al., 2024; Chu et al., 2025; Zhao et al., 2025b). This lack of transparency can undermine trust and limit the potential of automated grading systems in reliable educational contexts (Winkelmes, 2023).

Concept bottleneck models (CBMs) offer a promising middle ground between predictive strength and user-centric transparency (Koh et al., 2020). By constraining predictions to pass through human-interpretable intermediate concepts, CBMs can expose task-specific evidence behind a

[*]Equal contribution [1]School of Computing and Augmented Intelligence, Arizona State University, USA [2]Mary Lou Fulton Teachers College, Arizona State University, USA [3]Department of Computer Science, National Yang Ming Chiao Tung University, TW. Correspondence to: Chengshuai Zhao <czhao93@asu.edu>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

final grade rather than leaving the decision process hidden inside an end-to-end encoder, which makes them particularly appealing for educational assessment.

Despite this promise, existing CBMs are not yet well-suited to open-ended grading. Firstly, standard CBMs assume all concepts reside in a shared latent space, which is suboptimal for fine-grained grading settings where concepts capture different rubric dimensions and attend to different aspects of responses. In addition, they treat concept scores as evenly spaced categories, which discards the natural ranking information in the rubric annotations. Furthermore, rubric concept labels in real world are inherently noisy and subjective due to annotator disagreement, overlapping rubric dimensions, and ambiguity in open-ended responses (Zhao et al., 2025c). Existing CBMs generally assume concept labels are reliable, which can degrade grading performance and undermine trust in the concept-level explanations.

To address these limitations, we propose REC-CBM, a rubric-aware error-correction concept bottleneck model for trustworthy open-ended grading. REC-CBM first performs rubric-aware concept extraction so that each grading dimension is grounded in evidence relevant to that aspect of the response. It then applies ordinal concept calibration to preserve ranking structure among rubric levels (Bradley & Terry, 1952), thus capturing the score semantics. Finally, it incorporates latent concept error correction to denoise intermediate concept predictions while preserving the interpretable bottleneck.

Through comprehensive experiments, REC-CBM demonstrates superior grading performance with faithful concept-level reasoning than both state-of-the-art black-box models and transparent baselines. Further analyses validate the contribution of each component and demonstrate the applicability in realistic educational settings. Our contributions are fourfold:

- **Problem formulation:** We approach trustworthy open-ended grading through the lens of transparent decision-making, grounding the grading process on human-interpretable concepts, enabling educators to inspect, verify, and intervene on reasoning.
- **Data resources:** We curate and annotate public open-ended grading benchmarks with rubric-aligned concept labels, creating a valuable resource for future research on interpretable educational assessment.
- **Methodological innovation:** We identify key limitations of standard CBMs in the grading context and propose REC-CBM, a novel framework that integrates rubric-aware concept encoding, ordinal pairwise calibration, and latent concept error correction to address these limitations.
- **Empirical validation:** We conduct extensive evaluations and experiments, offering insights into the performance, interpretability, and practical utility of REC-CBM in real-world grading scenarios.

2. Related Work

2.1. Automated Grading Systems

Automated grading has evolved from feature-engineered systems such as e-rater (Attali & Burstein, 2006) to neural scoring models (Clauser et al., 2024). Pre-trained language models further improved accuracy (Wang et al., 2022), and recent LLM-based approaches incorporate rubric descriptions through prompting (Golchin et al., 2025), yet their internal reasoning remains opaque. A growing line of work seeks to address this: Wang et al. (Wang et al., 2024) and Chu et al. (Chu et al., 2025) align scoring with human rationales, while Schaller et al. (Schaller et al., 2024) audit fairness in essay scoring. However, these approaches surface post-hoc explanations without mechanistically guaranteeing that the explanation determines the assigned grade.

2.2. Concept Bottleneck Models

Concept Bottleneck Models (Koh et al., 2020) improve transparency by routing predictions through human-interpretable concepts. Extensions address prediction uncertainty via probabilistic (Kim et al., 2023), stochastic (Vandenhirtz et al., 2024), and post-hoc (Yuksekgonul et al., 2023) concept bottlenecks, though none model measurement error from a psychometric perspective or account for ordinal concept structure. CBMs have also been adapted beyond vision: Tan et al. (Tan et al., 2024) interpret pretrained language models via concept bottlenecks, Ismail et al. (Ismail et al., 2025) target protein design, and Espinosa Zarlenga et al. (Zarlenga et al., 2023) extend CBMs to tabular data. These works demonstrate the versatility of the CBM approaches in different domains, but the application to open-ended grading remains underexplored.

2.3. Trustworthy AI in Education

Trustworthy AI in education, spanning fairness, transparency, accountability, and reliability, has become a growing concern as AI systems are deployed at scale (Holmes et al., 2022; Miao et al., 2021). Research has documented algorithmic bias across demographic groups in student modeling and admissions (Baker & Hawn, 2022), while explainable AI methods have been adapted for intelligent tutoring and learning analytics (Khosravi et al., 2022). Complementary efforts improve the reliability of AI-powered educational tools, such as ontology-aware retrieval for cybersecurity education (Zhao et al., 2025a), and regulatory bodies have begun codifying principles for AI in high-stakes assessment (Williamson et al., 2026). We extend this line of research by achieving trustworthy open-ended grading through concept bottleneck models that enable educators to

inspect, verify, and intervene in the grading process.

3. Preliminaries

3.1. Open-Ended Grading Task Formulation

We cast open-ended grading in education as an *ordinal* multi-class classification problem. Each instance consists of a question or prompt q, a student's free-text response r, and optional auxiliary grading context a, such as a reference answer or rubric description. We represent a grading instance as

$$\mathbf{x} = (q,\ r,\ a), \tag{1}$$

and let $\mathcal{X}$ denote the space of grading instances. The grading target is an ordinal score $y \in \mathcal{Y} = \{0, 1, \ldots, S\}$, where S denotes the maximum attainable score. Given a labeled dataset $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, the goal is to learn a grading function

$$f\colon \mathcal{X} \to \mathcal{Y}, \qquad \hat{y} = f(\mathbf{x}), \tag{2}$$

that minimizes the empirical risk

$$\hat{f} \in \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{n=1}^{N} \ell\Big(f(\mathbf{x}^{(n)}), y^{(n)}\Big), \tag{3}$$

for a hypothesis class $\mathcal{F}$ and loss ℓ.

3.2. Concept Bottleneck Models

CBMs (Koh et al., 2020) improve model transparency by introducing an intermediate layer of human-interpretable *concepts* $\mathbf{c}$ between the raw input $\mathbf{x}$ and the final prediction $\hat{y}$.

A standard CBM decomposes prediction into a concept encoder ϕ and a task predictor ψ. Given an input $\mathbf{x}$, the concept encoder first estimates:

$$\phi\colon \mathcal{X} \to \mathbb{R}^K, \qquad \hat{\mathbf{c}} = \phi(\mathbf{x}), \tag{4}$$

and the task predictor then maps the predicted concepts to the final score:

$$\psi\colon \mathbb{R}^K \to \mathcal{Y}, \qquad \hat{y} = \psi(\hat{\mathbf{c}}). \tag{5}$$

The objective of training a CBM is to minimize a joint loss that includes both the concept prediction loss ℓ_{con} and the task prediction loss ℓ_{task}:

$$\hat{f} \in \arg\min_f \tfrac{1}{N} \textstyle\sum_{n=1}^{N} \Big[\lambda_c \ell_{\text{con}}\big(\hat{\mathbf{c}}^{(n)}, \mathbf{c}^{(n)}\big) + \lambda_t \ell_{\text{task}}\big(\hat{y}^{(n)}, y^{(n)}\big)\Big], \tag{6}$$

where $\mathbf{c}^{(n)}$ is the human-annotated concept for instance n, and $\lambda_c, \lambda_t \geq 0$ are hyperparameters that balance the two losses. Equivalently, the overall model factors as $f = \psi \circ \phi$, so all task-relevant information must pass through the concept layer before a final prediction is produced, which provides a mechanistic interpretability.

The CBMs offer several advantages in educational settings. First, it provides *transparency*: the model's final grade can be decomposed into rubric-level concept predictions that are easier for instructors to inspect. Second, it supports *intervention and auditing*: educators can examine or modify concept scores and observe how the overall prediction changes. Third, it promotes *pedagogical usefulness*: concept-level outputs align naturally with feedback categories used in teaching and assessment. These properties make CBMs a promising framework for trustworthy open-ended grading.

4. The Proposed REC-CBM Framework

4.1. Rubric-Aware Concept Encoder

Vanilla CBMs assume a shared representation space for all concepts, limiting fine-grained grading where rubric dimensions focus on different response aspects. We address this with a rubric-aware concept encoder.

Given an input $\mathbf{x} = (q, r, a)$, we employ a text encoder $\mathcal{E}$ (e.g., a pretrained language model) to obtain contextualized token representations:

$$\mathbf{H} = \mathcal{E}(\mathbf{x}) \in \mathbb{R}^{T \times d}, \tag{7}$$

where T is the sequence length and d is the hidden dimension. Then, we introduce a *concept query bank* to formulate the concept prototypes:

$$\mathcal{Q} = \{\mathbf{q}_k\}_{k=1}^{K} \subset \mathbb{R}^d, \tag{8}$$

where $\mathbf{q}_k$ is a learnable vector specializing in the textual evidence relevant to rubric dimension k, initialized via QR orthogonal decomposition of a random Gaussian matrix, promoting diverse patterns across concepts.

For each concept k, we compute soft attention weights and aggregate the representation via:

$$\alpha_{k,t} = \frac{\exp\big(\mathbf{q}_k^\top \mathbf{H}_t \ / \ \tau\big)}{\sum_{t'=1}^{T} \exp\big(\mathbf{q}_k^\top \mathbf{H}_{t'} \ / \ \tau\big)}, \tag{9}$$

$$\mathbf{h}_k = \sum_{t=1}^{T} \alpha_{k,t}\, \mathbf{H}_t, \tag{10}$$

where $\tau > 0$ is a temperature parameter. The mechanism selects rubric-specific token spans; each $\mathbf{q}_k$ acts as a dimension-specific retrieval query, rendering representations directly interpretable.

Let M denote the maximum ordinal level of each rubric concept, so that $c_k \in \{0, \ldots, M\}$. We employ a linear classifier $\mathbf{V}_k \in \mathbb{R}^{(M+1) \times d}$ to predict the per-concept ordinal

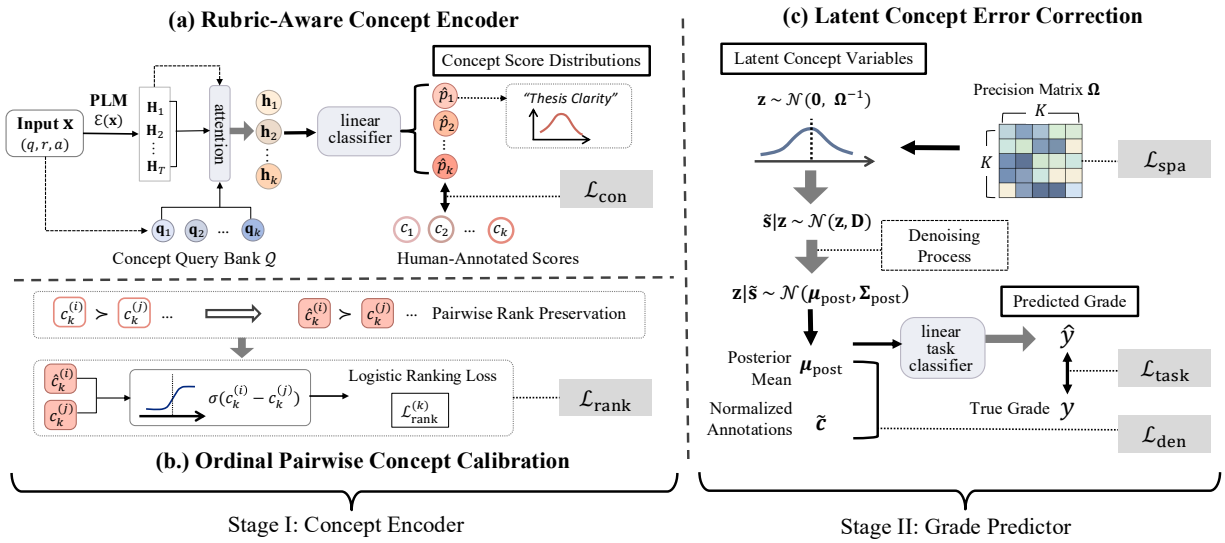

Figure 2. The proposed REC-CBM framework.

distribution $\hat{\mathbf{p}}_k$:

$$\hat{\mathbf{p}}_k = \mathrm{Softmax}(\mathbf{V}_k \mathbf{h}_k) \in \mathbb{R}^{M+1}, \tag{11}$$

We define the concept prediction loss as the sum of cross-entropy over all concepts.

$$\mathcal{L}_{\mathrm{con}} = \sum_{k=1}^{K} \mathrm{CrossEntropy}(\hat{\mathbf{p}}_k, c_k)\,, \tag{12}$$

4.2. Ordinal Pairwise Concept Calibration

The concept prediction loss from §4.1 treats the ordinal levels of each concept as unordered categories, which discards the natural ranking information in the rubric annotations. To preserve this ordinal structure, we introduce an ordinal pairwise concept calibration.

First, we compute the expected ordinal concept score $\hat{c}_k$ from the predicted distribution $\hat{\mathbf{p}}_k$:

$$\hat{c}_k = \mathbb{E}_{m\sim\hat{\mathbf{p}}_k}[m] = \sum_{m=0}^{M} m\,\hat{p}_{k,m} \in [0, M], \tag{13}$$

so that $\hat{c}_k$ lives on the same ordinal scale c_k. For each concept, we collect the set of within-batch B response pairs that admit a valid comparison under the rubric:

$$\mathcal{P}_k = \big\{(i,j) : c_k^{(i)} > c_k^{(j)},\ i,j \in [B]\big\}. \tag{14}$$

Inspired by the Bradley-Terry model (Bradley & Terry, 1952), we minimize a logistic ranking loss over these pairs, which encourages the predicted scores to respect the observed ordering:

$$\mathcal{L}_{\mathrm{rank}}^{(k)} = -\frac{1}{|\mathcal{P}_k|} \sum_{(i,j)\in\mathcal{P}_k} \log\sigma\Big(\hat{c}_k^{(i)} - \hat{c}_k^{(j)}\Big)\,. \tag{15}$$

where $\sigma(z) = 1/(1+e^{-z})$ is the logistic sigmoid function. This loss encourages the model to assign higher predicted scores to responses that are annotated with higher concept levels, thus providing the ordinal relationship calibration in the rubric-aligned concepts.

The overall ranking loss averages over all concepts that admit at least one valid comparison in the batch:

$$\mathcal{L}_{\mathrm{rank}} = \frac{1}{K'} \sum_{\substack{k=1 \\ |\mathcal{P}_k|>0}}^{K} \mathcal{L}_{\mathrm{rank}}^{(k)}, \tag{16}$$

where $K' = \big|\{k : |\mathcal{P}_k| > 0\}\big|$.

4.3. Latent Concept Error Correction

Even after ordinal calibration, the predicted scores $\{\hat{c}_k\}$ from §4.2 still inherit annotator disagreement, overlap across rubric dimensions, and ambiguity in open-ended assessment. Passing these scores directly to the grading head would force the final predictor to rely on noisy concept estimates. We therefore introduce a latent correction layer that treats each calibrated concept score as an error-corrupted observation of an underlying latent concept state and then computes a closed-form posterior correction before grade prediction.

Let $\mathbf{z} \in \mathbb{R}^K$ denote the latent concept vector. We place a multivariate Gaussian prior on $\mathbf{z}$ with learnable precision matrix $\mathbf{\Omega} \in \mathbb{R}^{K\times K}$:

$$\mathbf{z} \sim \mathcal{N}\big(\mathbf{0},\ \mathbf{\Omega}^{-1}\big)\,. \tag{17}$$

we normalize each predicted concept to the unit interval,

$$\tilde{s}_k = \hat{c}_k / M \in [0, 1],$$

and model it as a noisy observation of the corresponding latent component:

$$\tilde{s}_k \mid z_k \sim \mathcal{N}(z_k, \sigma_k^2), \quad k \in [K], \tag{18}$$

where $\sigma_k^2 > 0$ is the measurement-noise variance of concept k. Stacking the K concepts yields $\tilde{\mathbf{s}} \mid \mathbf{z} \sim \mathcal{N}(\mathbf{z}, \mathbf{D})$ with $\mathbf{D} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_K^2)$.

Under this measurement-error model, the posterior $\mathbf{z} \mid \tilde{\mathbf{s}}$ remains Gaussian with

$$\mathbf{\Sigma}_{\mathrm{post}} = \left(\mathbf{\Omega} + \mathbf{D}^{-1}\right)^{-1}, \tag{19}$$

$$\boldsymbol{\mu}_{\mathrm{post}} = \mathbf{\Sigma}_{\mathrm{post}}\, \mathbf{D}^{-1}\, \tilde{\mathbf{s}} = \mathbf{A}\, \tilde{\mathbf{s}}, \tag{20}$$

where $\mathbf{A} \triangleq \mathbf{\Sigma}_{\mathrm{post}}\, \mathbf{D}^{-1}$ is the denoising matrix. The posterior mean $\boldsymbol{\mu}_{\mathrm{post}}$ is the corrected concept score used by the task head. Intuitively, concepts with larger estimated noise are shrunk more strongly, while off-diagonal structure in $\mathbf{\Omega}$ allows related rubric dimensions to share statistical strength.

Proposition 1. *Under the Gaussian measurement model in Eqs. (17)–(18), the posterior mean $\boldsymbol{\mu}_{\mathrm{post}} = \mathbf{A}\tilde{\mathbf{s}}$ in Eq. (20) is the unique minimum mean-squared error* (MMSE) *estimator of $\mathbf{z}$ given $\tilde{\mathbf{s}}$:*

$$\mathbf{A}\tilde{\mathbf{s}} = \arg\min_{g}\ \mathbb{E}\left[\|\mathbf{z} - g(\tilde{\mathbf{s}})\|^2\right],$$

with the minimum over all measurable functions g.

The proof is deferred to Appendix B.1.

To enforce a positive-definite $\mathbf{\Omega}$, we parameterize it through a lower-triangular Cholesky factor $\mathbf{L} \in \mathbb{R}^{K \times K}$:

$$\mathbf{\Omega} = \mathbf{L}\mathbf{L}^{\top} + \varepsilon \mathbf{I}, \tag{21}$$

where $\varepsilon > 0$ is a small regularization constant. Each noise variance is parameterized as $\sigma_k^2 = \exp(\eta_k)$ with $\eta_k \in \mathbb{R}$ learnable, guaranteeing strict positivity.

The corrected concept vector is then passed to a linear task classifier $\mathbf{W} \in \mathbb{R}^{(S+1) \times K}$, producing logits over the $S+1$ grade levels:

$$\hat{y} = \mathbf{W}\, \boldsymbol{\mu}_{\mathrm{post}}. \tag{22}$$

This design preserves the interpretable bottleneck while correcting concept-level noise before grade prediction.

The latent head is trained with three objectives. The task loss supervises the final grade prediction:

$$\mathcal{L}_{\mathrm{task}} = \mathrm{CrossEntropy}(\hat{y}, y). \tag{23}$$

The denoising alignment loss keeps the corrected concepts close to the normalized rubric annotations $\tilde{\mathbf{c}} = \mathbf{c}/M \in [0,1]^K$:

$$\mathcal{L}_{\mathrm{den}} = \frac{1}{K} \left\|\boldsymbol{\mu}_{\mathrm{post}} - \tilde{\mathbf{c}}\right\|^2. \tag{24}$$

The sparsity penalty regularizes cross-concept dependencies by shrinking the off-diagonal entries of the Cholesky factor:

$$\mathcal{L}_{\mathrm{spa}} = \sum_{i>j} |L_{ij}|. \tag{25}$$

This encourages the correction layer to stay near-diagonal unless the data support stronger latent interactions among rubric dimensions.

4.4. Learning Objective and Training Paradigm

The full REC-CBM training objective combines losses from all components:

$$\mathcal{L} = \lambda_c \mathcal{L}_{\mathrm{con}} + \lambda_r \mathcal{L}_{\mathrm{rank}} + \lambda_t \mathcal{L}_{\mathrm{task}} + \lambda_d \mathcal{L}_{\mathrm{den}} + \lambda_s \mathcal{L}_{\mathrm{spa}}, \tag{26}$$

with non-negative hyperparameters $\lambda_c, \lambda_r, \lambda_t, \lambda_d, \lambda_s$.

Optimizing all parameters jointly may entangle the latent head's denoising role with the encoder's representation learning and collapse the interpretable bottleneck. We therefore use two-stage training.

Stage I jointly trains the text encoder $\mathcal{E}$, concept query bank $\mathcal{Q}$, and per-concept classifiers $\{\mathbf{V}_k\}$ under concept supervision and ordinal calibration objectives:

$$\min_{\mathcal{E},\, \mathcal{Q},\, \{\mathbf{V}_k\}}\ \lambda_c \mathcal{L}_{\mathrm{con}} + \lambda_r \mathcal{L}_{\mathrm{rank}}. \tag{27}$$

It thus learns rubric-aligned concept predictors that respect token-level evidence and ordinal structure, after which these components are frozen.

Stage II fits the latent-correction parameters $(\mathbf{L}, \{\eta_k\}, \mathbf{W})$ on top of the fixed concept outputs:

$$\min_{\mathbf{L},\, \{\eta_k\},\, \mathbf{W}}\ \lambda_t \mathcal{L}_{\mathrm{task}} + \lambda_d \mathcal{L}_{\mathrm{den}} + \lambda_s \mathcal{L}_{\mathrm{spa}}. \tag{28}$$

This decomposition ensures that Stage I yields interpretable, ordinally calibrated concept estimates, while Stage II learns to correct measurement error and map the corrected concepts to the final grade without distorting the bottleneck. The complete optimization procedure is summarized in Algorithm 1 in Appendix B.2.

Table 1. Dataset statistics.

Dataset	Samples	Concepts	Concept Levels	Grade Levels
Mohler	2,273	8	3	6
ASAP 2.0	17,292	8	5	6
MOCHA	31,069	7	3	5

5. Experiments

5.1. Datasets, Annotation, and Evaluation Protocol

Datasets. We evaluate REC-CBM on three open-ended grading benchmarks spanning short-answer and essay scoring settings: Mohler short-answer grading (Mohler et al.,

Table 2. Main results on the grading benchmarks. **Bold** marks the best result within each model block. Concept metrics are unavailable for black-box baselines, and "–" denotes unavailable values.

Dataset		Mohler				ASAP 2.0				MOCHA			
Architecture	**Method**	**C-Acc.** ↑	**C-F1** ↑	**T-Acc.** ↑	**T-F1** ↑	**C-Acc.** ↑	**C-F1** ↑	**T-Acc.** ↑	**T-F1** ↑	**C-Acc.** ↑	**C-F1** ↑	**T-Acc.** ↑	**T-F1** ↑
Pre-trained Language Models (Black-box)													
BERT	PLM	–	–	0.706	0.529	–	–	0.591	0.485	–	–	0.578	0.463
BART	PLM	–	–	0.675	0.507	–	–	0.583	0.459	–	–	0.601	0.472
GPT-2	PLM	–	–	0.627	0.416	–	–	0.612	0.488	–	–	0.564	0.419
RoBERTa	PLM	–	–	0.715	0.563	–	–	**0.620**	**0.526**	–	–	**0.624**	**0.514**
T5-Base	PLM	–	–	**0.741**	**0.564**	–	–	0.590	0.508	–	–	0.598	0.478
Large Language Models (Black-box)													
Llama-3-8B	Zero Shot	–	–	0.158	0.207	–	–	0.334	0.175	–	–	0.305	0.219
Qwen2.5-14B	Zero Shot	–	–	0.070	0.104	–	–	**0.413**	0.187	–	–	0.370	0.273
Mistral-7B	Zero Shot	–	–	0.329	**0.376**	–	–	0.383	**0.225**	–	–	0.388	0.354
Mistral-7B	3-Shot	–	–	**0.399**	0.352	–	–	0.277	0.221	–	–	**0.561**	**0.370**
Concept Bottleneck Models (White-box)													
BERT	Vanilla CBM	0.692	0.622	0.706	0.510	0.555	0.532	0.589	0.458	**0.734**	0.565	0.586	0.473
	C^3M	0.708	0.571	0.689	0.466	0.566	0.522	0.606	0.445	0.723	0.575	0.593	0.473
	CT-CBM	0.638	0.522	0.583	0.237	0.492	0.478	0.598	0.449	0.669	0.525	0.576	0.446
	CB-LLM	–	–	0.298	0.077	–	–	0.276	0.094	–	–	0.470	0.131
	REC-CBM	**0.709**	**0.648**	**0.715**	**0.580**	**0.573**	**0.563**	**0.609**	**0.466**	0.731	**0.591**	**0.604**	**0.486**
BART	Vanilla CBM	0.678	0.606	0.671	0.494	0.570	0.534	0.592	0.492	0.735	0.560	0.615	0.473
	C^3M	0.628	0.568	0.592	0.421	0.566	0.515	0.584	0.482	0.732	0.558	0.597	0.479
	CT-CBM	0.557	0.495	0.557	0.203	**0.601**	**0.573**	0.609	0.467	0.688	0.504	0.596	0.472
	CB-LLM	–	–	0.298	0.077	–	–	0.276	0.094	–	–	0.470	0.131
	REC-CBM	**0.707**	**0.624**	**0.732**	**0.541**	0.578	0.513	**0.611**	**0.520**	**0.738**	**0.576**	**0.632**	**0.497**
GPT-2	Vanilla CBM	0.667	0.600	0.596	0.298	0.591	0.535	0.584	0.466	0.705	0.505	0.578	0.437
	C^3M	**0.677**	0.603	0.640	0.344	0.581	0.523	0.584	0.464	0.703	0.515	0.581	0.429
	CT-CBM	0.488	0.429	0.496	0.110	0.346	0.280	0.363	0.089	0.399	0.278	0.475	0.129
	CB-LLM	–	–	0.298	0.077	–	–	0.276	0.094	–	–	0.470	0.131
	REC-CBM	**0.677**	**0.623**	**0.724**	**0.615**	**0.595**	**0.557**	**0.618**	**0.507**	**0.732**	**0.584**	**0.617**	**0.486**
RoBERTa	Vanilla CBM	0.715	0.651	0.737	0.558	0.583	0.561	0.620	0.526	**0.763**	0.603	0.633	0.513
	C^3M	0.700	0.633	0.706	0.530	0.564	0.546	0.606	0.505	0.759	0.597	0.634	0.512
	CT-CBM	0.545	0.482	0.579	0.277	0.619	0.576	0.622	0.511	0.676	0.518	0.603	0.475
	CB-LLM	–	–	0.298	0.077	–	–	0.276	0.094	–	–	0.470	0.131
	REC-CBM	**0.735**	**0.686**	**0.746**	**0.570**	**0.638**	**0.607**	**0.642**	**0.545**	**0.763**	**0.617**	**0.642**	**0.530**
T5-Base	Vanilla CBM	**0.702**	0.622	0.711	0.484	0.593	0.555	0.571	0.473	0.729	0.558	0.608	0.481
	C^3M	0.688	0.600	0.702	0.532	0.603	0.555	0.584	0.443	0.728	0.556	0.600	0.477
	CT-CBM	0.589	0.518	0.610	0.267	0.610	0.576	0.595	0.463	0.734	0.570	0.618	0.483
	CB-LLM	–	–	0.298	0.077	–	–	0.276	0.094	–	–	0.470	0.131
	REC-CBM	0.695	**0.623**	**0.724**	**0.547**	**0.656**	**0.616**	**0.602**	**0.486**	**0.774**	**0.647**	**0.632**	**0.500**

2011), ASAP 2.0 essay scoring (Crossley et al., 2025), and MOCHA reading-comprehension answer grading (Chen et al., 2020).

Annotation. None of the three benchmarks include concept-level annotations, so we curate and annotate each dataset with a human-in-the-loop (HITL) annotation pipeline: three domain experts first propose the concepts and corresponding rubrics. Then we prompt GPT-4o (Hurst et al., 2024) and Gemini-2.5-pro (Comanici et al., 2025) to annotate the dataset with proposed rubrics, after which the experts independently review the annotations, identify and resolve inconsistent cases through discussion and majority voting. Dataset-specific descriptions, rubric, and annotation details are deferred to Appendix C, and the rubric design rationale, concept groupings, and the rubrics' role in the pipeline are detailed in §C.3. Table 1 summarizes the corpus scale and label-space statistics.

Evaluation Protocol. All datasets use a 7:2:1 train/dev/test partition. We report task-level accuracy and macro-F1 for grading quality, together with concept-level accuracy and macro-F1 for the intermediate concept predictions. Implementation details such as backbone-specific settings, model selection by validation macro-F1, and stage-specific hyperparameters are deferred to Appendix D. Without specific mention, all results are averaged over five random seeds.

5.2. Main Results

RQ1: Can REC-CBM *improve open-ended grading performance while preserving interpretable concept predictions relative to black-box graders and prior CBM baselines?*

We benchmark REC-CBM against three baseline classes: (I) fine-tuned PLMs, (II) zero-/few-shot prompted LLMs, and (III) SOTA CBMs adapted to grading; model-specific details appear in Appendix D. Table 2 reports the results. Across three datasets and five backbones, REC-CBM attains the best T-Acc and T-F1 within every backbone block, and the best C-F1 in 14 of 15 scenarios, improving the bottleneck alongside the final grade. The three baseline classes diverge sharply. Among PLMs, RoBERTa is the strongest black-box grader, yet same-backbone REC-CBM matches or exceeds its T-F1 on Mohler/ASAP 2.0/MOCHA

(0.570/0.545/0.530 vs. 0.563/0.526/0.514) while exposing rubric-level evidence that the PLM cannot. Prompted LLMs lag both PLMs and CBMs—T-Acc on Mohler falls to 0.070 (Qwen2.5-14B)—showing that generic instruction-following does not recover rubric-specific grading without task training. Among prior CBMs, Vanilla CBM and C^{3}M are the strongest priors and REC-CBM improves on both axes across all datasets, whereas CT-CBM and CB-LLM fail on at least one dataset (CB-LLM's T-F1 is constant at 0.077/0.094/0.131 across backbones, indicating majority-class collapse). Critically, accuracy and interpretability move together: on RoBERTa, REC-CBM lifts C-F1 by $+3.5/+4.6/+1.4$ points over Vanilla CBM on the three datasets while delivering matching task-side gains. Per-backbone gains and failure-mode diagnostics are in Appendix E.1. Overall, REC-CBM offers a promising white-box grading approach with superior performance and rubric-aligned concept interpretability.

5.3. Ablation Studies

RQ2: Does each component of REC-CBM *contribute a measurable gain to grading accuracy and concept reliability?*

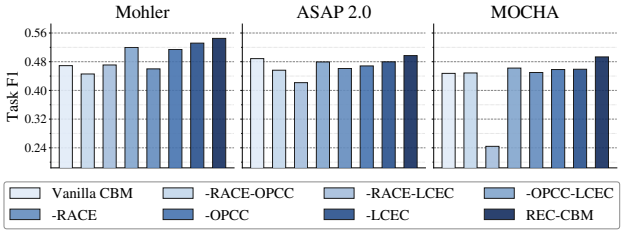

Figure 3. Ablation over components of REC-CBM: RACE (rubric-aware encoder), OPCC (ordinal pairwise calibration), and LCEC (latent error correction).

Fig. 3 compares the full REC-CBM against seven ablated variants that remove some components: RACE (§4.1), OPCC (§4.2), and LCEC (§4.3). REC-CBM attains the best T-ACC on every dataset, and removing any single component degrades T-ACC on at least two of the three datasets, indicating that the three modules contribute complementary rather than redundant signal. Among single removals, dropping RACE is the most damaging, consistent with the claim in §4.1 that per-rubric concept queries carry most of the discriminative evidence; removing OPCC or LCEC yields smaller but consistent declines. Double-removal variants sit between the single-removal variants and Vanilla CBM on every dataset, producing a monotone "more components, better grading" trend. The sharpest failure is −RACE−OPCC on MOCHA, where jointly ablating the rubric-aware encoder and the ordinal calibrator nearly halves T-ACC, leaving LCEC to correct a near-uninformative signal on MOCHA's coarse three-level rubric; this identifies RACE together with OPCC as the minimally sufficient upstream for LCEC to be useful. Per-dataset bars and component-wise gain decomposition are reported in Appendix E.4.

5.4. Parameter Analysis

RQ3: How sensitive is REC-CBM *to its core hyperparameters?*

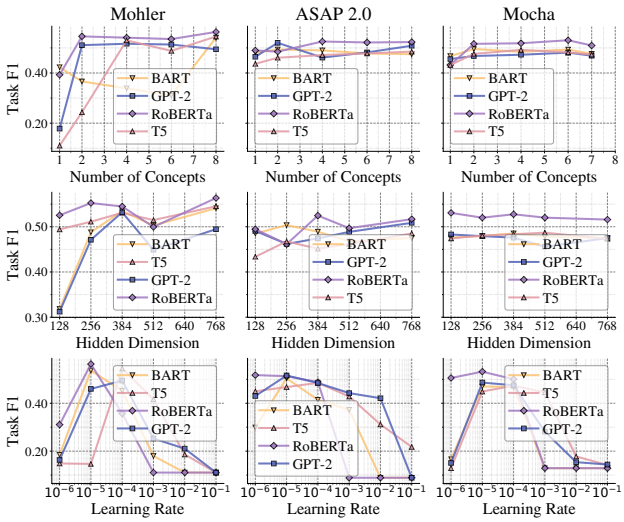

Figure 4. Parameter analysis of REC-CBM across various backbones on three grading datasets.

We sweep the three core hyperparameters of REC-CBM across four backbones on all three datasets and report task F1 in Fig. 4. Two of the three knobs are essentially flat: task F1 saturates by $K{\approx}4$ on every dataset and is insensitive to d once $d \geq 384$, so neither the rubric cardinality nor the encoder width requires backbone-specific tuning. The learning rate is the only stability-critical knob, forming a sharp inverted-U across backbones with a peak in $[10^{-5}, 10^{-4}]$ and a uniform collapse to near-zero F1 once LR$\geq 10^{-3}$. This pattern is consistent with the design in §4.1–§4.4: rubric-aware concept queries absorb additional capacity gracefully, while Stage I jointly updates the encoder, queries, and concept classifiers and is therefore the component most exposed to LR misconfiguration. The narrow tuned LR range used in Table 8 reflects this sensitivity. Detailed per-backbone analysis and dataset-specific failure modes are reported in Appendix E.2.

5.5. Human Intervention

RQ4: Can educators meaningfully steer REC-CBM *by intervening on its rubric-level concept predictions?*

To test whether the bottleneck is actionable, we simulate educator interventions on the top-k highest-confidence predicted concepts and re-evaluate the frozen Stage II head (Fig. 5). Wrong and Random degrade monotonically, losing 30–45 accuracy points at the largest k on every dataset,

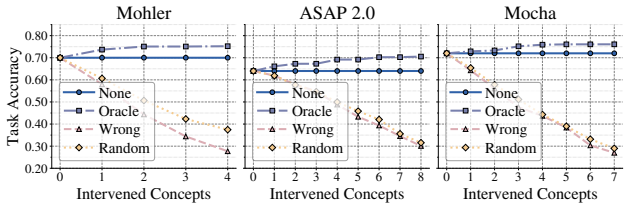

Figure 5. Human intervention on rubric concepts. For each dataset, we replace the top-k predicted concepts with the oracle rubric label (Oracle), an adversarial wrong label (Wrong), or a uniformly random label (Random), and no intervention (None).

while Oracle matches or slightly exceeds None. This asymmetry is the key faithfulness signal: corrupting rubric evidence flips the predicted grade, whereas correcting it preserves or improves the grade. The small Oracle–None gap reflects the measurement-error framing of §4.3—the Stage II head consumes Stage I concept outputs, so one-hot oracle substitutions are mildly off-distribution. Practically, educators can audit or override individual rubric dimensions and observe the grade respond accordingly, realizing the intervention property that motivates REC-CBM. Per-dataset analysis is deferred to Appendix E.3.

5.6. Latent Denoising Analysis

RQ5: Does the latent correction module learn a sparse, rubric-meaningful concept dependency structure rather than echoing raw label correlations?

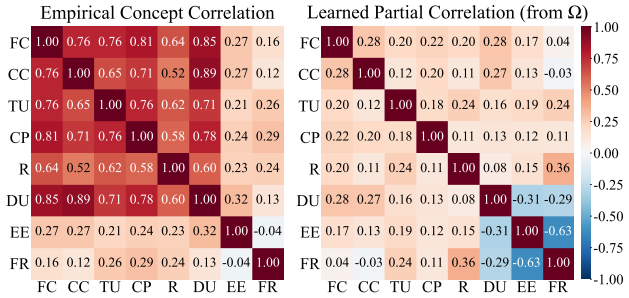

Figure 6. Latent denoising analysis: empirical rubric-label correlation (left) versus learned partial correlation from the latent precision $\mathbf{\Omega}$ (right).

Fig. 6 contrasts the empirical rubric-label correlation (left) with the partial correlations induced by the learned precision $\mathbf{\Omega}$ in Eq. (17) (right). The empirical side shows dense co-correlation across content and reasoning dimensions, confirming the annotator overlap and rubric redundancy that motivate §4.3. The learned $\mathbf{\Omega}$ is markedly sparser, consistent with the near-diagonal target enforced by $\mathcal{L}_{\text{spa}}$ in Eq. (25), while preserves most interpretable dependencies. Consequently, the denoising matrix $\mathbf{A} = \mathbf{\Sigma}_{\text{post}}\mathbf{D}^{-1}$ propagates information along a sparse rubric-level graph that decouples correlated concepts and amplifies the most discriminative dimensions. Detailed heatmap analysis and the named residual links are deferred to Appendix E.6.

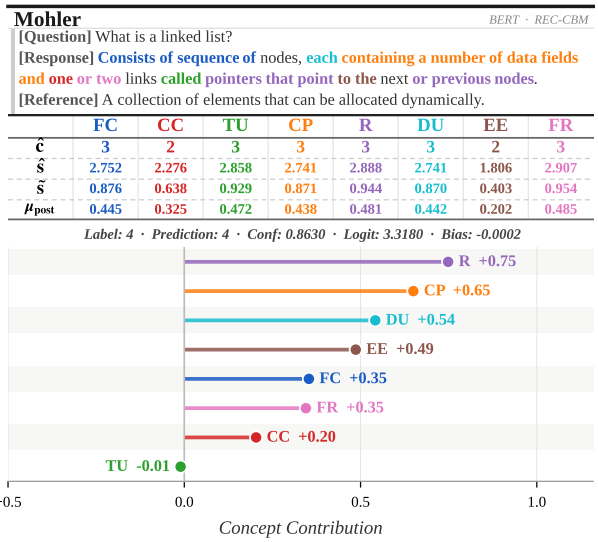

Figure 7. End-to-end decision trace for a Mohler short-answer response under BERT-REC-CBM. From top to bottom: rubric-aware token attention, per-concept ordinal predictions ($\hat{\mathbf{c}}$, $\hat{\mathbf{s}}$) with their LCEC input ($\tilde{\mathbf{s}}$) and posterior ($\boldsymbol{\mu}_{\text{post}}$), and each concept's contribution to the predicted grade.

5.7. Case Study

RQ6: Does REC-CBM *produce a human-readable decision trace that an educator can audit end-to-end on responses?*

Fig. 7 traces a single Mohler short-answer grading decision by a BERT-backbone REC-CBM, whose predicted grade matches the human label with high confidence. The panel exposes three strata of the bottleneck: rubric-aware token attention from §4.1 (colored response spans), per-concept ordinal scores calibrated by §4.2 together with their LCEC-corrected posteriors from §4.3 (table rows), and the additive per-concept contributions that compose the final grade logit (bar chart). Every quantity in the trace is tied to a named rubric dimension, so the grade is reconstructible from the bottleneck and any dimension can be inspected or overridden, realizing the inspect/verify/intervene workflow motivated in §1. A component-by-component walkthrough of this example is deferred to Appendix E.5.

6. Conclusion

In this paper, we present REC-CBM, a concept bottleneck framework for trustworthy open-ended grading that integrates a rubric-aware concept encoder, an ordinal pairwise concept calibration, and a latent concept error correction module. Extensive experiments and analyses demonstrate that REC-CBM attains stronger grading performance while delivering more faithful, robust, and actionable rubric-level reasoning than existing black-box and transparent approaches. Future work will extend REC-CBM to multilingual and domain-specific assessment and couple it with educator-in-the-loop rubric refinement.

References

Attali, Y. and Burstein, J. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3), 2006.

Baker, R. S. and Hawn, A. Algorithmic bias in education. *International journal of artificial intelligence in education*, 32(4):1052–1092, 2022.

Bhan, M., Choho, Y., Vittaut, J.-N., Chesneau, N., Moreau, P., and Lesot, M.-J. Towards achieving concept completeness for textual concept bottleneck models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 2007–2024. Association for Computational Linguistics, 2025.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Chen, A., Stanovsky, G., Singh, S., and Gardner, M. Mocha: A dataset for training and evaluating generative reading comprehension metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6521–6532, 2020.

Chu, S., Kim, J. W., Wong, B., and Yi, M. Y. Rationale behind essay scores: Enhancing s-llm's multi-trait essay scoring with rationale generated by llms. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5796–5814, 2025.

Clauser, B. E., Yaneva, V., Baldwin, P., An Ha, L., and Mee, J. Automated scoring of short-answer questions: A progress report. *Applied Measurement in Education*, 37 (3):209–224, 2024.

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Crossley, S. A., Baffour, P., Burleigh, L., and King, J. A large-scale corpus for assessing source-based writing quality: Asap 2.0. *Assessing Writing*, 65:100954, 2025.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Golchin, S., Garuda, N., Impey, C., and Wenger, M. Grading massive open online courses using large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 3899–3912, 2025.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Holmes, W., Porayska-Pomsta, K., Holstein, K., Sutherland, E., Baker, T., Shum, S. B., Santos, O. C., Rodrigo, M. T., Cukurova, M., Bittencourt, I. I., et al. Ethics of ai in education: Towards a community-wide framework. *International Journal of Artificial Intelligence in Education*, 32(3):504–526, 2022.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Ismail, A. A., Oikarinen, T., Wang, A., Adebayo, J., Stanton, S. D., Bravo, H. C., Cho, K., and Frey, N. C. Concept bottleneck language models for protein design. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Yt9CFhOOFe.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Khosravi, H., Shum, S. B., Chen, G., Conati, C., Tsai, Y.-S., Kay, J., Knight, S., Martinez-Maldonado, R., Sadiq, S., and Gašević, D. Explainable artificial intelligence in education. *Computers and education: artificial intelligence*, 3:100074, 2022.

Kim, E., Jung, D., Park, S., Kim, S., and Yoon, S. Probabilistic concept bottleneck models. In *International Conference on Machine Learning*, pp. 16521–16540. PMLR, 2023.

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. Concept bottleneck models. In *International conference on machine learning*, pp. 5338–5348. PMLR, 2020.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 7871–7880, 2020.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Miao, F., Holmes, W., et al. *AI and education: A guidance for policymakers*. Unesco Publishing, 2021.

Mohler, M., Bunescu, R., and Mihalcea, R. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 752–762, 2011.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Schaller, N.-J., Ding, Y., Horbach, A., Meyer, J., and Jansen, T. Fairness in automated essay scoring: A comparative analysis of algorithms on german learner essays from secondary education. In *Proceedings of the 19th workshop on innovative use of nlp for building educational applications (bea 2024)*, pp. 210–221, 2024.

Spearman, C. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

Sun, C.-E., Oikarinen, T., Ustun, B., and Weng, T.-W. Concept bottleneck large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=RC5FPYVQaH.

Tan, Z., Cheng, L., Wang, S., Yuan, B., Li, J., and Liu, H. Interpreting pretrained language models via concept bottlenecks. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 56–74. Springer, 2024.

Team, Q. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Vandenhirtz, M., Laguna, S., Marcinkevičs, R., and Vogt, J. E. Stochastic concept bottleneck models. *Advances in Neural Information Processing Systems*, 37:51787–51810, 2024.

Wang, Y., Wang, C., Li, R., and Lin, H. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3416–3425, 2022.

Wang, Y., Hu, R., and Zhao, Z. Beyond agreement: Diagnosing the rationale alignment of automated essay scoring methods based on linguistically-informed counterfactuals. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8906–8925, 2024.

Williamson, J., Bramley, T., Handford, J., Leahy, F., Stratton, T., and Wilson, F. Principles of ai use in marking. Technical report, Working paper, Ofqual. GOV. UK, 2026.

Winkelmes, M.-A. Introduction to transparency in learning and teaching. *Perspectives in Learning*, 20(1):2, 2023.

Yuksekgonul, M., Wang, M., and Zou, J. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=nA5AZ8CEyow.

Zarlenga, M. E., Shams, Z., Nelson, M. E., Kim, B., and Jamnik, M. Tabcbm: Concept-based interpretable neural networks for tabular data. *Transactions on Machine Learning Research*, 2023.

Zhao, C., Agrawal, G., Zhang, F., Kumarage, T., Tan, Z., Deng, Y., Chen, Y.-C., and Liu, H. Ontology-aware rag for improved question-answering in cybersecurity education. In *2025 IEEE International Conference on Big Data (BigData)*, pp. 3161–3170. IEEE, 2025a.

Zhao, C., Tan, Z., Ma, P., Li, D., Jiang, B., Wang, Y., Yang, Y., and huan liu. Is chain-of-thought reasoning of LLMs a mirage? a data distribution lens. In *First Workshop on Foundations of Reasoning in Language Models*, 2025b. URL https://openreview.net/forum?id=o2AoLPIjle.

Zhao, C., Tan, Z., Wong, C.-W., Zhao, X., Chen, T., and Liu, H. Scale: Towards collaborative content analysis in social science with large language model agents and human intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8473–8503, 2025c.

A. Notation

Table 3 summarizes the symbols used in §4 and Appendix B, grouped by role: inputs and labels, rubric-aware encoder, ordinal calibration, latent error correction, and training objective.

Table 3. Symbol list for REC-CBM.

Symbol	Meaning	Type	Range / Space
	Inputs and labels		
$\mathbf{x} = (q, r, a)$	Grading instance (question, response, auxiliary context)	Tuple	$\mathcal{X}$
$y,\ \hat{y}$	Grade label / predicted grade logits	Scalar / vector	$\mathcal{Y} = \{0, \dots, S\}$ / $\mathbb{R}^{S+1}$
K	Number of rubric concepts	Scalar	$\mathbb{N}$
M	Maximum ordinal level per concept	Scalar	$\mathbb{N}$
$c_k,\ \mathbf{c}$	Concept-level label (scalar entry / stacked vector)	Ordinal / vector	$\{0, \dots, M\}$ / $\{0, \dots, M\}^K$
$\tilde{c}_k,\ \tilde{\mathbf{c}}$	Normalized concept label $\tilde{c}_k = c_k/M$	Scalar / vector	$[0, 1]$ / $[0, 1]^K$
	Rubric-aware encoder (§4.1)		
$\mathcal{E}$	Text encoder (pretrained backbone)	Function	$\mathcal{X} \to \mathbb{R}^{T \times d}$
$T,\ d$	Sequence length / hidden dimension	Scalar	$\mathbb{N}$
$\mathbf{H},\ \mathbf{H}_t$	Token representations / t-th token state	Matrix / vector	$\mathbb{R}^{T \times d}$ / $\mathbb{R}^d$
$\mathcal{Q} = \{\mathbf{q}_k\}$	Concept query bank	Set of vectors	$\mathbb{R}^d$
$\alpha_{k,t}$	Soft attention weight of concept k on token t	Scalar	$[0, 1]$
$\mathbf{h}_k$	Concept-k aggregated representation	Vector	$\mathbb{R}^d$
τ	Attention temperature	Scalar	$\mathbb{R}_{>0}$
$\mathbf{V}_k$	Per-concept classifier	Matrix	$\mathbb{R}^{(M+1) \times d}$
$\hat{\mathbf{p}}_k$	Predicted ordinal distribution for concept k	Distribution	$\mathbb{R}^{M+1}$
$\hat{p}_{k,m}$	m-th entry of $\hat{\mathbf{p}}_k$	Scalar	$[0, 1]$
	Ordinal calibration (§4.2)		
$\hat{c}_k,\ \hat{\mathbf{c}}$	Predicted (expected) concept score	Scalar / vector	$[0, M]$ / $[0, M]^K$
B	Mini-batch size	Scalar	$\mathbb{N}$
$\mathcal{P}_k$	Within-batch valid comparison pairs	Set	$\subseteq [B] \times [B]$
K'	# concepts with non-empty $\mathcal{P}_k$	Scalar	$\{0, \dots, K\}$
$\sigma(\cdot)$	Logistic sigmoid	Function	$\mathbb{R} \to (0, 1)$
	Latent error correction (§4.3)		
$\tilde{s}_k,\ \tilde{\mathbf{s}}$	Normalized concept score $\tilde{s}_k = \hat{c}_k/M$	Scalar / vector	$[0, 1]$ / $[0, 1]^K$
$\mathbf{z}$	Latent concept vector	Vector	$\mathbb{R}^K$
$\mathbf{\Omega}$	Latent precision matrix	Matrix (PD)	$\mathbb{R}^{K \times K}$
$\mathbf{L}$	Cholesky factor of $\mathbf{\Omega} - \varepsilon\mathbf{I}$	Lower-triangular matrix	$\mathbb{R}^{K \times K}$
ε	Precision regularizer	Scalar	$\mathbb{R}_{>0}$
η_k	Log-variance parameter	Scalar	$\mathbb{R}$
$\sigma_k^2 = \exp(\eta_k)$	Measurement-noise variance of concept k	Scalar	$\mathbb{R}_{>0}$
$\mathbf{D}$	Diagonal measurement covariance	Diagonal matrix	$\mathbb{R}^{K \times K}$
$\mathbf{\Sigma}_{\text{post}}$	Posterior covariance of $\mathbf{z} \mid \tilde{\mathbf{s}}$	Matrix (PD)	$\mathbb{R}^{K \times K}$
$\boldsymbol{\mu}_{\text{post}}$	Posterior mean (corrected concept vector)	Vector	$\mathbb{R}^K$
$\mathbf{A} = \mathbf{\Sigma}_{\text{post}}\mathbf{D}^{-1}$	Denoising matrix	Matrix	$\mathbb{R}^{K \times K}$
$\mathbf{W}$	Task classifier (grade head)	Matrix	$\mathbb{R}^{(S+1) \times K}$
	Training objective (§4.4)		
$\mathcal{L}_{\text{con}},\ \mathcal{L}_{\text{rank}}$	Concept / ordinal ranking losses	Scalar	$\mathbb{R}_{\geq 0}$
$\mathcal{L}_{\text{task}},\ \mathcal{L}_{\text{den}},\ \mathcal{L}_{\text{spa}}$	Task / denoising / sparsity losses	Scalar	$\mathbb{R}_{\geq 0}$
$\lambda_c, \lambda_r, \lambda_t, \lambda_d, \lambda_s$	Loss weights	Scalar	$\mathbb{R}_{\geq 0}$
$E_{\text{I}},\ E_{\text{II}}$	Stage-I / Stage-II epoch budgets	Scalar	$\mathbb{N}$

B. Method Details

This appendix collects the supporting derivation and the full training algorithm referenced in §4.

B.1. Proof of Proposition 1.

We now justify the MMSE claim in Proposition 1. Recall from §4.3 that the latent correction module assumes a Gaussian prior over the latent concept vector and a Gaussian measurement model for the normalized concept scores. Under these assumptions, the posterior mean admits a closed form and coincides with the Bayes estimator under squared error.

For reference, we first record the equivalent conditional-mean form

$$\boldsymbol{\mu}_{\text{post}} = \boldsymbol{\Omega}^{-1}\big(\boldsymbol{\Omega}^{-1} + \mathbf{D}\big)^{-1}\tilde{\mathbf{s}} = \big(\boldsymbol{\Omega} + \mathbf{D}^{-1}\big)^{-1}\mathbf{D}^{-1}\tilde{\mathbf{s}},$$

which makes explicit how the corrected concept vector balances latent precision against measurement noise.

Proof. By Eqs. (17)–(18), we may write the observation model in vector form as

$$\tilde{\mathbf{s}} = \mathbf{z} + \boldsymbol{\varepsilon}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Omega}^{-1}), \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{D}),$$

where $\boldsymbol{\varepsilon}$ is independent of $\mathbf{z}$. It follows that $(\mathbf{z}, \tilde{\mathbf{s}})$ is jointly Gaussian with mean zero and block covariance

$$\operatorname{Cov}\begin{pmatrix}\mathbf{z}\\ \tilde{\mathbf{s}}\end{pmatrix} = \begin{pmatrix}\boldsymbol{\Omega}^{-1} & \boldsymbol{\Omega}^{-1}\\ \boldsymbol{\Omega}^{-1} & \boldsymbol{\Omega}^{-1} + \mathbf{D}\end{pmatrix}.$$

Indeed, $\operatorname{Cov}(\mathbf{z}, \tilde{\mathbf{s}}) = \operatorname{Cov}(\mathbf{z}, \mathbf{z} + \boldsymbol{\varepsilon}) = \boldsymbol{\Omega}^{-1}$ because the measurement noise is independent of $\mathbf{z}$, and $\operatorname{Cov}(\tilde{\mathbf{s}}, \tilde{\mathbf{s}}) = \boldsymbol{\Omega}^{-1} + \mathbf{D}$.

For any square-integrable random vector, the estimator minimizing $\mathbb{E}[\|\mathbf{z} - g(\tilde{\mathbf{s}})\|^2]$ over all measurable functions g is the conditional expectation $\mathbb{E}[\mathbf{z} \mid \tilde{\mathbf{s}}]$. Since the pair $(\mathbf{z}, \tilde{\mathbf{s}})$ is jointly Gaussian, this conditional expectation is affine, and because the mean is zero it is in fact linear:

$$\mathbb{E}[\mathbf{z} \mid \tilde{\mathbf{s}}] = \operatorname{Cov}(\mathbf{z}, \tilde{\mathbf{s}})\operatorname{Cov}(\tilde{\mathbf{s}}, \tilde{\mathbf{s}})^{-1}\tilde{\mathbf{s}} = \boldsymbol{\Omega}^{-1}\left(\boldsymbol{\Omega}^{-1} + \mathbf{D}\right)^{-1}\tilde{\mathbf{s}}.$$

It remains to express this linear estimator in the posterior form used in the main text. Using the identity $P(P+R)^{-1} = (I + RP^{-1})^{-1} = (P^{-1} + R^{-1})^{-1}R^{-1}$ with $P = \boldsymbol{\Omega}^{-1}$ and $R = \mathbf{D}$, we obtain

$$\mathbb{E}[\mathbf{z} \mid \tilde{\mathbf{s}}] = \big(\boldsymbol{\Omega} + \mathbf{D}^{-1}\big)^{-1}\mathbf{D}^{-1}\tilde{\mathbf{s}} = \boldsymbol{\Sigma}_{\text{post}}\,\mathbf{D}^{-1}\tilde{\mathbf{s}} = \mathbf{A}\tilde{\mathbf{s}} = \boldsymbol{\mu}_{\text{post}}.$$

Therefore, $\boldsymbol{\mu}_{\text{post}}$ is the MMSE estimator of $\mathbf{z}$ given $\tilde{\mathbf{s}}$. Uniqueness holds up to almost-sure equality because conditional expectation is unique in L^2. □

Remark 1. *In the diagonal case $\boldsymbol{\Omega} = \operatorname{diag}(\sigma_{z_1}^{-2}, \ldots, \sigma_{z_K}^{-2})$, the denoising matrix becomes diagonal and the k-th entry simplifies to*

$$A_{kk} = \frac{\sigma_{z_k}^2}{\sigma_{z_k}^2 + \sigma_k^2} = \rho_k,$$

recovering Spearman's reliability coefficient and the scalar Wiener filter for concept k (Spearman, 1904). In the correlated case, the off-diagonal entries of $\boldsymbol{\Omega}$ propagate information from reliably measured concepts to noisily measured ones, implementing a multivariate generalization of reliability-weighted shrinkage that is optimal in the MMSE sense.

B.2. Two-Stage Training Algorithm

Algorithm 1 expands the two-stage optimization procedure described in §4.4. The key design choice is to decouple *concept learning* from *error-aware grade prediction*. In Stage I, the model learns rubric-aligned concept predictors directly from the response text, while the ordinal ranking objective encourages concept estimates to preserve the relative ordering induced by rubric labels. In Stage II, these concept predictors are frozen and treated as fixed measurements, allowing the latent correction module to focus on denoising and aggregating concept evidence for final grade prediction without altering the interpretable bottleneck.

This stage-wise decomposition serves two purposes. First, it stabilizes optimization by preventing the latent task head from backpropagating through the concept encoder and distorting the rubric-specific representations. Second, it preserves the semantics of the concept layer: the concept scores remain grounded in textual evidence learned in Stage I, while Stage II

Algorithm 1 Two-stage training procedure for REC-CBM

Require: Training set $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)}, \mathbf{c}^{(n)})\}_{n=1}^{N}$
Require: Loss weights $\lambda_c, \lambda_r, \lambda_t, \lambda_d, \lambda_s$ and epoch budgets $E_{\mathrm{I}}, E_{\mathrm{II}}$
Ensure: Trained parameters $\Theta = (\mathcal{E}, \mathcal{Q}, \{\mathbf{V}_k\}, \mathbf{L}, \{\eta_k\}, \mathbf{W})$
Initialize encoder $\mathcal{E}$ and concept-query bank $\mathcal{Q} = \{\mathbf{q}_k\}_{k=1}^{K}$
Initialize concept classifiers $\{\mathbf{V}_k\}_{k=1}^{K}$, latent parameters $(\mathbf{L}, \{\eta_k\})$, and task head $\mathbf{W}$
Stage I: Learn rubric-aligned ordinal concept predictors
for $e \leftarrow 1$ **to** E_{I} **do**
  for all minibatches $\mathcal{B} \subset \mathcal{D}$ **do**
    Encode each response in $\mathcal{B}$ with $\mathcal{E}$ to obtain token states $\mathbf{H}$
    Apply concept queries $\mathcal{Q}$ to compute rubric-specific representations $\{\mathbf{h}_k\}_{k=1}^{K}$
    Predict concept distributions $\{\hat{\mathbf{p}}_k\}_{k=1}^{K}$ with $\{\mathbf{V}_k\}_{k=1}^{K}$
    Compute expected concept scores $\{\hat{c}_k\}_{k=1}^{K}$ from $\{\hat{\mathbf{p}}_k\}_{k=1}^{K}$
    Construct within-batch comparison sets $\{\mathcal{P}_k\}_{k=1}^{K}$ from concept labels
    Evaluate concept loss $\mathcal{L}_{\mathrm{con}}$ and ranking loss $\mathcal{L}_{\mathrm{rank}}$
    Update $(\mathcal{E}, \mathcal{Q}, \{\mathbf{V}_k\})$ using $\lambda_c \mathcal{L}_{\mathrm{con}} + \lambda_r \mathcal{L}_{\mathrm{rank}}$
  end for
end for
Freeze $\mathcal{E}$, $\mathcal{Q}$, and $\{\mathbf{V}_k\}$
Stage II: Learn latent correction and grade prediction
for $e \leftarrow 1$ **to** E_{II} **do**
  for all minibatches $\mathcal{B} \subset \mathcal{D}$ **do**
    Recompute frozen concept predictions $\{(\hat{\mathbf{p}}_k, \hat{c}_k)\}_{k=1}^{K}$ for $\mathcal{B}$ without gradients
    Normalize concept scores: $\tilde{s}_k \leftarrow \hat{c}_k / M$ for all $k \in [K]$
    Form $\mathbf{D} = \mathrm{diag}(\exp(\eta_1), \ldots, \exp(\eta_K))$ and $\mathbf{\Omega} = \mathbf{L}\mathbf{L}^{\top} + \varepsilon \mathbf{I}$
    Compute posterior mean $\boldsymbol{\mu}_{\mathrm{post}}$ via Eq. (20)
    Predict grade logits $\hat{y} \leftarrow \mathbf{W}\, \boldsymbol{\mu}_{\mathrm{post}}$
    Evaluate $\mathcal{L}_{\mathrm{task}}$, $\mathcal{L}_{\mathrm{den}}$, and $\mathcal{L}_{\mathrm{spa}}$
    Update $(\mathbf{L}, \{\eta_k\}, \mathbf{W})$ using $\lambda_t \mathcal{L}_{\mathrm{task}} + \lambda_d \mathcal{L}_{\mathrm{den}} + \lambda_s \mathcal{L}_{\mathrm{spa}}$
  end for
end for
return $\Theta = (\mathcal{E}, \mathcal{Q}, \{\mathbf{V}_k\}, \mathbf{L}, \{\eta_k\}, \mathbf{W})$

only adjusts how these noisy concept measurements are combined downstream. The resulting procedure therefore maintains reviewer-facing interpretability while improving robustness to annotation noise and cross-concept dependency.

Compared with a one-stage end-to-end update, Algorithm 1 makes the information flow explicit. Stage I determines *what* rubric evidence is extracted from text, and Stage II determines *how* that evidence should be reliability-weighted and aggregated for grading. This separation is especially useful in open-ended assessment, where concept annotations are informative but imperfect, because it lets the latent head correct noisy concept measurements without overwriting the rubric-aligned concept definitions learned from the response text.

C. Datasets and Annotation

This appendix details the benchmarks and the rubric-aligned human-in-the-loop annotation pipeline used throughout §5.1.

C.1. Benchmark Details

We use three public grading benchmarks covering complementary open-ended assessment settings. Mohler (Mohler et al., 2011) is a short-answer grading benchmark in computer science, where each instance contains a question, a reference answer, and a student response. ASAP 2.0 (Crossley et al., 2025) is an essay-scoring benchmark containing longer student compositions graded on an ordinal holistic scale. MOCHA (Chen et al., 2020) is a reading-comprehension answer-grading benchmark in which each response is evaluated with respect to a question, supporting context, and reference answer.

C.2. Annotation Pipeline

None of the three benchmarks includes concept-level supervision, so we curate rubric-aligned concept annotations for all datasets. We use a human-in-the-loop workflow involving three domain experts. The experts first propose the concept inventory and corresponding rubrics for each benchmark, specifying the ordinal interpretation and anchored descriptors of every concept level. Given these rubric definitions, GPT-4o (Hurst et al., 2024) and Gemini-2.5-pro (Comanici et al., 2025) produce initial concept annotations for each response.

The same three experts then validate the LLM-generated annotations. They independently review the proposed labels, identify inconsistent or ambiguous cases, and resolve disagreements through discussion and majority voting. This pipeline combines scalable initial labeling with expert oversight, while preserving a transparent decision rule for the final concept labels.

To make this annotation stage reproducible, we use a shared prompt template that conditions on benchmark-specific input fields and rubric definitions, while keeping the output interface fixed across datasets. The template is used only to generate the initial concept annotations; all final labels still come from the expert-validation procedure described above.

Prompt for LLM Concept Annotation

```
You are an expert educational assessor for open-ended grading. Your task is to
assign one rubric-aligned ordinal label to each concept for a student response,
using only the provided benchmark inputs and rubric definitions.

Inputs:
Benchmark: <benchmark_name>
Question or prompt (q): <question_or_prompt>
Student response (r): <student_response>
Auxiliary grading context (a): <reference_answer>, <concept_list>,
<rubric_definitions>

Optional in-context examples:
Example 1 input: <example_1_question>, <example_1_reference>, <example_1_rubric>,
<example_1_response>
Example 1 output: <json_annotation_list_1>
Example 2 input: <example_2_question>, <example_2_reference>, <example_2_rubric>,
<example_2_response>
Example 2 output: <json_annotation_list_2>

Requirements:
1. Score each concept independently using only the rubric definitions and the
supplied benchmark context.
2. Use the reference answer, concept list, and rubric descriptors when evaluating
the student response.
3. Respect the ordered level meanings defined in the rubrics; do not interpolate
between levels or invent new labels.
4. For each concept, provide a brief evidence snippet grounded in the student
response or the supplied grading context.
5. If the evidence is ambiguous between two adjacent levels, choose the
lower-supported level.
6. Do not provide explanations outside the requested output structure.

Return only a structured list with one entry per concept in the format:
concept=<name>; label=<ordinal_level>; evidence=<brief_evidence>
```

The placeholders are instantiated with the benchmark-specific inputs used throughout the experiments. For Mohler, the prompt receives the question, student response, and auxiliary grading context consisting of the reference answer, the eight-concept inventory, and the corresponding three-level rubrics. For ASAP 2.0, it receives the essay prompt, student essay, and auxiliary grading context consisting of the concept inventory and the corresponding five-level rubrics. For MOCHA, it receives the reading context, question, candidate answer, and auxiliary grading context consisting of the reference answer, the seven-concept inventory, and the corresponding three-level rubrics. Grade labels follow the native benchmark scales summarized in Table 1.

C.3. Rubric Definitions

Tables 5–6 list the rubric dimensions for ASAP 2.0, Mohler, and MOCHA, including the textual definition, an illustrative excerpt with the assigned score, and the ordinal score scale. These rubrics are the inputs to the annotation pipeline in §C.2 and the supervision targets for the concept encoder in §4.1. ASAP 2.0 uses eight concepts on a five-level scale; Mohler uses eight concepts on a three-level scale; MOCHA uses seven concepts on a three-level scale.

Construction principles. The three domain experts described in §C.2 drafted each inventory under three criteria. First, the chosen concepts should *cover* the dominant rubric axes that determine the holistic grade for the benchmark, so that grading evidence is not pushed outside the bottleneck. Second, concepts should be *non-redundant*, with each dimension contributing distinct evidence rather than re-scoring an aspect already captured by a sibling concept. Third, every level of every concept should be *ordinally interpretable*, with an anchored descriptor that an expert can apply consistently. These criteria match how the rubrics are subsequently consumed: the concept encoder in §4.1 treats each rubric dimension as a separate supervision target, so coverage and non-redundancy directly determine whether the bottleneck is sufficient for the task.

Why the scale cardinalities differ. ASAP 2.0 uses a five-level ordinal scale per concept, while Mohler and MOCHA use three. The cardinality reflects the granularity that human raters can reliably assign given the response artifact. Multi-paragraph essays in ASAP 2.0 (§C.1) afford enough surface evidence per dimension to distinguish five anchored bands, whereas the short answers in Mohler and the short candidate spans in MOCHA support only a coarser correct/partial/incorrect distinction without forcing raters to interpolate. The same contrast is reflected by the per-benchmark concept-level counts in Table 1.

Concept groupings. Within each benchmark, the rubric dimensions in Tables 5–6 fall into a small number of clusters that aid scanning. For ASAP 2.0, Thesis Clarity, Use of Evidence, and Critical Thinking Depth target content and reasoning; Organization & Coherence and Sentence Variety target structural organization; Grammar & Mechanics, Vocabulary Appropriateness, and Fluency / Readability target surface form. For Mohler, Factual Correctness, Concept Coverage, and Relevance target answer correctness and coverage; Terminology Usage, Clarity / Precision, and Fluency / Readability target expression; Depth of Understanding and Example / Elaboration target reasoning depth. For MOCHA, Relevance to Question and Completeness target alignment with the question; Textual Grounding, Coreference Resolution, and Inference Accuracy target consistency with the supporting passage; Paraphrasing Quality and Conciseness & Clarity target expression of the candidate.

Role of the rubrics inside REC-CBM. Each table column maps to a specific component of the pipeline. The *definition* and *score scale* columns are inserted verbatim into the `<rubric_definitions>` and `<concept_list>` placeholders of the LLM annotation prompt in §C.2. The resulting expert-validated ordinal labels c_k become the per-concept supervision targets for the cross-entropy classifier in §4.1 and define the within-batch comparison set $\mathcal{P}_k$ that drives the ordinal pairwise calibration objective in §4.2. The illustrative excerpts in the *example* column are not consumed by the pipeline and serve only as anchors for the human reviewers in the HITL loop.

Table 4. Mohler rubric: eight concepts scored on a three-level ordinal scale.

Concept	Definition	Example	Score scale
Factual Correctness (FC)	The extent to which the answer contains accurate domain knowledge aligned with the desired answer.	*Student Answer:* "It simulates portions of the desired final product…" (FC= 3, correct) *Student Answer:* "A prototype program is used to collect data." (FC= 1, incorrect concept)	1 = Incorrect 2 = Partially correct 3 = Correct
Concept Coverage (CC)	The degree to which the answer includes all key ideas required by the reference answer.	*Student Answer:* "To simulate portions of the desired final product…" (CC= 3, complete) *Student Answer:* "To find problems before finalizing." (CC= 2, partial)	1 = Missing key ideas 2 = Partial coverage 3 = Complete coverage
Terminology Usage (TU)	The extent to which domain-specific terms are used accurately and appropriately.	*Student Answer:* "It helps test the program." (TU= 2, basic terminology) *Student Answer:* "It helps with things in coding." (TU= 1, lacks domain terms)	1 = Inaccurate or absent terminology 2 = Basic usage 3 = Appropriate usage

Continued on next page.

Table 4 continued from previous page.

Concept	Definition	Example	Score scale
Clarity / Precision (CP)	The extent to which the answer is clearly expressed and avoids ambiguity or vague wording.	*Student Answer:* "It simulates how parts of the system will behave." (CP= 3, clear) *Student Answer:* "It helps with things." (CP= 1, vague)	1 = Unclear / vague 2 = Somewhat clear 3 = Clear and precise
Relevance (R)	The extent to which the answer directly addresses the question without irrelevant content.	*Student Answer:* "It helps programmers improve code." (R= 2, partially relevant) *Student Answer:* "Programs are written using code." (R= 1, irrelevant)	1 = Irrelevant 2 = Partially relevant 3 = Fully relevant
Depth of Understanding (DU)	The extent to which the answer reflects conceptual understanding beyond surface-level recall.	*Student Answer:* "It simulates system behavior and helps identify issues before full development." (DU= 3, deeper understanding) *Student Answer:* "It finds problems before finalizing." (DU= 2, basic understanding)	1 = Surface-level 2 = Basic understanding 3 = Deeper understanding
Example / Elaboration (EE)	The extent to which the answer extends beyond a basic statement by providing explanation or elaboration.	*Student Answer:* "It simulates behavior so developers can test parts before building the full system." (EE= 3, elaborated) *Student Answer:* "It simulates behavior." (EE= 1, minimal)	1 = No elaboration 2 = Limited elaboration 3 = Elaborated explanation
Fluency / Readability (FR)	The overall clarity and flow of the answer, including grammatical correctness and readability.	*Student Answer:* "It help find problem in program." (FR= 2, some errors) *Student Answer:* "It help with thing." (FR= 1, difficult to read)	1 = Difficult to read 2 = Somewhat readable 3 = Fluent and readable

Table 5. ASAP 2.0 rubric: eight concepts scored on a five-level ordinal scale.

Concept	Definition	Example	Score scale
Thesis Clarity (TC)	The extent to which the essay presents a clear, specific, and coherent central argument or position.	*Excerpt:* "My position on driveless cars are bad because they can cause accidents." (TC= 4) *Excerpt:* "What if we could tell how all of the people use cars?" (TC= 1)	1 = No clear thesis 2 = Weak/unclear 3 = General thesis 4 = Clear 5 = Clear and specific
Use of Evidence (UE)	The quality and relevance of reasoning or examples used to support the argument.	*Excerpt:* "They can crash because computers can fail and people may get hurt." (UE= 3) *Excerpt:* "Driverless cars are bad." (UE= 1)	1 = No evidence 2 = Minimal support 3 = Some relevant evidence 4 = Adequate support 5 = Strong, well-developed evidence
Organization & Coherence (OC)	The extent to which ideas are logically structured and connected across the essay.	*Excerpt:* "First, they are unsafe. Next, they cost a lot. Finally, they may replace jobs." (OC= 4) *Excerpt:* "Driverless cars are bad. Technology is growing. People like cars." (OC= 1)	1 = Disorganized 2 = Weak structure 3 = Basic organization 4 = Clear structure 5 = Strong and coherent progression
Grammar & Mechanics (GM)	The degree of grammatical accuracy and correctness in sentence construction, spelling, and punctuation.	*Excerpt:* "Driverless cars are dangerous because they can fail." (GM= 4) *Excerpt:* "Driverless cars is dangerous because it fail." (GM= 2)	1 = Frequent errors 2 = Many errors 3 = Some errors 4 = Few errors 5 = Virtually error-free

Continued on next page.

Table 5 continued from previous page.

Concept	Definition	Example	Score scale
Vocabulary Appropriateness (VA)	The extent to which vocabulary is used accurately and appropriately for the task.	*Excerpt:* "Driverless cars may pose significant safety risks." (VA= 4) *Excerpt:* "Cars are very very bad and not good." (VA= 1)	1 = Very limited vocabulary 2 = Basic/repetitive 3 = Adequate 4 = Appropriate 5 = Precise and varied
Sentence Variety (SV)	The extent to which the essay demonstrates variation in sentence structures.	*Excerpt:* "Although driverless cars reduce human error, they introduce new technological risks." (SV= 4) *Excerpt:* "Cars are bad. Cars are unsafe. Cars are not good." (SV= 1)	1 = Very repetitive 2 = Limited variety 3 = Some variety 4 = Good variety 5 = Sophisticated variation
Critical Thinking Depth (CTD)	The extent to which the essay demonstrates reasoning, evaluation, or consideration of alternative perspectives.	*Excerpt:* "While they may reduce human error, system failures could create new risks." (CTD= 4) *Excerpt:* "They are bad." (CTD= 1)	1 = No analysis 2 = Minimal reasoning 3 = Basic reasoning 4 = Clear reasoning 5 = Insightful analysis
Fluency / Readability (FR)	The overall clarity, flow, and ease with which the essay can be read and understood.	*Excerpt:* "Driverless cars are risky because they rely on technology that may fail unexpectedly." (FR= 4) *Excerpt:* "What if we could tell how all of the people use cars…" (FR= 2)	1 = Very difficult to read 2 = Limited clarity 3 = Generally clear 4 = Fluent 5 = Highly fluent and natural

Table 6. MOCHA rubric: seven concepts scored on a three-level ordinal scale.

Concept	Definition	Example	Score scale
Relevance to Question (RQ)	The extent to which the candidate directly addresses the question being asked.	*Q:* Why might he be wearing a dressing gown? *Candidate:* "He is a fan of Vince McMahon." (does not address the question; RQ= 1) *Q:* Why did they call the fire dept? *Candidate:* "because they thought it would start to fire." (directly addresses the question; RQ= 3)	1 = Irrelevant 2 = Partially relevant 3 = Fully relevant
Inference Accuracy (IA)	The correctness of reasoning when the candidate requires inference beyond explicitly stated information.	*Q:* What might be true about Chinese divers? *Candidate:* "They are knowledgeable." (incorrect inference; IA= 1) *Q:* Why did they call the fire dept? *Candidate:* "because they thought it would start to fire." (reasonable inference; IA= 2)	1 = Incorrect inference 2 = Partially correct inference 3 = Correct inference
Textual Grounding (TG)	The degree to which the candidate is supported by or consistent with the passage.	*Q:* Why did they call the fire dept? *Candidate:* "because they thought it would start to fire." (supported by context; TG= 3) *Q:* Why might he be wearing a dressing gown? *Candidate:* "He is a fan of Vince McMahon." (not supported; TG= 1)	1 = Not grounded 2 = Partially grounded 3 = Fully grounded
Coreference Resolution (CR)	The accuracy with which the candidate correctly interprets and resolves references (e.g., pronouns, entities).	*Q:* How did we get fruit salad? *Candidate:* "We got it at our friend's place." (partially aligned but unclear attribution; CR= 2)	1 = Incorrect resolution 2 = Partially correct / unclear 3 = Correct resolution

Continued on next page.

Table 6 continued from previous page.

Concept	Definition	Example	Score scale
Paraphrasing Quality (PQ)	The extent to which the candidate preserves the meaning of the reference while using different wording.	*Reference:* "Because it requires a lot of time and energy." *Candidate:* "it takes a lot of time and energy." (accurate paraphrase; PQ= 3) *Candidate:* "it brought a heavy appetite." (distorted meaning; PQ= 1)	1 = Distorted meaning 2 = Partially accurate 3 = Accurate paraphrase
Completeness (C)	The degree to which the candidate fully addresses all aspects of the question.	*Q:* What may have caused the old man to wait? *Candidate:* "he was waiting for his relatives." (fully answers; C= 3) *Q:* Why did I have a fun time? *Candidate:* "I was in a good mood…" (does not fully capture intended reason; C= 2)	1 = Incomplete 2 = Partially complete 3 = Complete
Conciseness & Clarity (CC)	The extent to which the candidate is clearly expressed without ambiguity or unnecessary wording.	*Candidate:* "He was hungry." (clear; CC= 3) *Candidate:* "because it's no more, but no more." (unclear expression; CC= 1)	1 = Unclear / poorly formed 2 = Somewhat clear 3 = Clear and concise

Scope. The three rubrics are English-language and benchmark-specific—short computer-science answers for Mohler, argumentative essays for ASAP 2.0, and narrative reading-comprehension responses for MOCHA—so transferring any of them to a new language, subject domain, or response genre requires re-eliciting the inventory and the level descriptors through the same HITL pipeline described in §C.2.

D. Implementation Details

The main experiments use the five pretrained backbones shown in Table 2: BERT, BART, GPT-2, RoBERTa, and T5-Base. For REC-CBM, we follow the explicit two-stage training paradigm in §4.4. Stage I is implemented with the ordinally calibrated concept encoder, which optimizes concept supervision and ordinal calibration. After freezing the Stage I encoder, Stage II trains only the latent correction module on top of the fixed concept predictions. Unless otherwise noted, all results are averaged over five random seeds, model selection is based on validation task macro-F1, and we report the corresponding test-set scores.

Table 7 summarizes the shared training and architecture settings. In the implementation, the attention temperature corresponds to τ, the concept-supervision weight to λ_c, the ordinal-calibration weight to λ_r, the task-loss weight to λ_t, the denoising-alignment weight to λ_d, the sparsity weight to λ_s, and the Cholesky regularizer to ε. Across all five backbones, the encoder hidden size is $d = 768$, and the concept-query space matches the backbone hidden size because no separate concept projection dimension is used in the main training path. In Stage II, the latent task head consumes the corrected concept vector in $\mathbb{R}^K$. Table 8 then summarizes the key tuned parameters, their meanings, and the values considered in the search space; bold values indicate the final optima used in the main experiments.

Stage I tunes the backbone learning rate, attention temperature τ, and ordinal-calibration weight λ_r for the ordinally calibrated concept encoder. Stage II tunes the latent-head learning rate together with the denoising-alignment and sparsity weights λ_d and λ_s while keeping the Stage I encoder frozen.

D.1. Baseline Details for Main Results

PLM baselines. The PLM baselines fine-tune five pretrained encoders directly for grade prediction: BERT (Devlin et al., 2019), BART (Lewis et al., 2020), GPT-2 (Radford et al., 2019), RoBERTa (Liu et al., 2019), and T5-Base (Raffel et al., 2020). These models provide strong task-only comparisons without concept supervision or an interpretable bottleneck.

LLM baselines. The prompted LLM baselines evaluate Llama-3-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-14B-Instruct (Team, 2024), and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) as black-box graders under zero-shot and 3-shot prompting, testing whether general-purpose instruction following can recover benchmark-specific grading behavior without task-specific training.

Table 7. Shared implementation settings for REC-CBM.

Component	Description	Value
BERT	Pretrained encoder backbone used in the main experiments	`google-bert/bert-base-uncased`
BART	Pretrained encoder backbone used in the main experiments	`facebook/bart-base`
GPT-2	Pretrained encoder backbone used in the main experiments	`openai-community/gpt2`
RoBERTa	Pretrained encoder backbone used in the main experiments	`FacebookAI/roberta-base`
T5-Base	Pretrained encoder backbone used in the main experiments	`google-t5/t5-base`
Hidden size d	Hidden representation dimension used by the concept encoder and classifiers	768
`max_len`	Maximum token number of each input sequence	128/256/**512**
`batch_size`	Number of training instances per optimization step	8
`num_epochs`	Maximum number of training epochs	50
`early_stopping_patience`	Number of non-improving epochs before early stopping	3
`warmup_ratio`	Fraction of total optimization steps used for linear warmup	0.1
`query_init_method`	Initialization scheme for the concept query bank	Orthogonal
`num_heads`	Attention heads per concept query	1
`query_lr_multiplier`	Relative learning-rate multiplier for concept queries	1
λ_c	Weight of the concept-supervision loss in Stage I	1.0
λ_t	Weight of the task prediction loss in Stage II	1.0
ε	Diagonal regularizer used to keep the latent precision matrix positive definite	10^{-4}

Table 8. Key parameters in REC-CBM with their meanings and values. Bold values indicate the optima.

Parameter	Description	Value
`Stage I LR`	Backbone learning rate for the ordinally calibrated concept encoder	1e-6/2e-6/3e-6/5e-6/8e-6/**1e-5**/**2e-5** 3e-5/5e-5/7e-5/8e-5/**1e-4**/2e-4/5e-4/1e-3
τ	Attention temperature controlling the sharpness of rubric-aware token attention	**0.25**/0.5/**0.75**/**1.0**/**1.25**/1.5/1.75/2.0
λ_r	Weight of the ordinal pairwise calibration objective in Stage I	0.1/**0.2**/**0.3**/**0.4**/0.5/**0.6**/0.7/**0.8**/**1.0**
`Stage II LR`	Learning rate for the latent correction head on frozen concept predictions	0.001/0.002/**0.005**/0.01/**0.02**/**0.05**
λ_d	Weight of the denoising-alignment loss between corrected concepts and rubric labels	0.0/0.05/**0.1**/**0.2**/0.5
λ_s	Weight of the sparsity regularizer on latent cross-concept dependencies	0.0/**0.005**/0.01/0.05/0.1

CBM baselines. We compare against four prior transparent baselines. Vanilla CBM (Koh et al., 2020) is the standard concept-then-task bottleneck model with supervised rubric concepts. C^{3}M (Tan et al., 2024) augments textual CBMs with concept augmentation and concept-level mixup to improve robustness under limited or noisy concept annotations. CT-CBM (Bhan et al., 2025) expands the textual concept basis automatically rather than relying only on predefined human concepts. CB-LLM (Sun et al., 2025) adapts the bottleneck idea to large language models by introducing an interpretable concept layer inside the LLM pipeline. REC-CBM adds rubric-aware concept encoding, ordinal concept calibration, and latent error correction to improve both grading performance and concept reliability.

E. Extended Experimental Analysis

This appendix expands the main-text empirical sections (§5.2, §5.4, §5.5) with per-cell numerical detail and dataset-specific failure-mode discussion.

E.1. Extended Main-Result Analysis

This appendix expands Table 2 with per-class detail that supports the summary claims in §5.2.

Within-CBM dominance. Across the three datasets and five backbones, REC-CBM wins every task-F1 and task-accuracy cell against the other four CBM baselines (15/15 each), and wins concept F1 in 14 of the 15 cells; the lone exception is BART on ASAP 2.0, where CT-CBM reaches C-F1 0.573 vs. REC-CBM at 0.513 despite REC-CBM still leading in task F1 (0.520 vs. 0.467). Improvements over Vanilla CBM, the strongest concept-supervised prior, are consistent across backbones. Taking the median across the five backbones, REC-CBM gains $+6.3$ / $+1.9$ / $+1.9$ T-F1 points and $+2.3$ / $+3.1$ / $+2.6$ C-F1 points on Mohler / ASAP 2.0 / MOCHA. The largest single-cell gain is on GPT-2 / Mohler (T-F1 0.298$\rightarrow$0.615), where Vanilla CBM's shared encoder is a particularly poor match to the short-answer rubric and the rubric-aware encoder recovers most of the missing signal.

Black-box PLM head-to-head. Comparing each PLM directly to the same-backbone REC-CBM row, REC-CBM wins T-F1 on 11 of 15 same-backbone cells. The remaining four cells are BERT on ASAP 2.0 (PLM 0.485 vs. REC-CBM 0.466), T5-Base on Mohler (0.564 vs. 0.547), and T5-Base on ASAP 2.0 (0.508 vs. 0.486). Each of these is overturned by switching

to a different backbone: RoBERTa + REC-CBM beats every PLM row on every dataset, so there is no configuration in which black-box access is strictly required.

LLM failure modes. Mohler exhibits the most severe collapse: zero-shot Llama-3-8B and Qwen2.5-14B reach only 0.158 and 0.070 T-Acc, and Mohler T-F1 never exceeds 0.376 across any prompted configuration. The zero-shot/few-shot asymmetry is also dataset-dependent: 3-shot prompting helps Mistral-7B on Mohler (T-Acc 0.329→0.399) and MOCHA (0.388→0.561) but hurts on ASAP 2.0 (0.383→0.277), consistent with long-essay inputs saturating the instruction-following budget and the demonstrations displacing task-relevant context.

CB-LLM collapse. CB-LLM's T-Acc and T-F1 are identical across all five backbone rows within a dataset (0.298 / 0.276 / 0.470 T-Acc and 0.077 / 0.094 / 0.131 T-F1 on Mohler / ASAP 2.0 / MOCHA) because its bottleneck is attached to a frozen LLM and does not consume the tabulated backbone. The constant T-Acc corresponds to near-majority-class prediction on each dataset, and the low T-F1 is consistent with collapse to a single grade—a failure mode REC-CBM avoids by routing predictions through a supervised rubric bottleneck that must match the annotated concept distribution.

CT-CBM instability. CT-CBM's automatic concept-basis expansion couples predictive capacity to the backbone's lexical-prediction quality, and it degrades sharply on GPT-2, with Mohler T-F1 0.110, ASAP T-F1 0.089, and MOCHA T-F1 0.129—well below every other CBM row in the GPT-2 block. REC-CBM anchors each concept to a learnable query over the backbone's contextual states, which decouples concept quality from autoregressive backbone behavior and aligns with the motivation in §4.1.

E.2. Extended Parameter Analysis

This appendix expands Fig. 4 with per-backbone detail that supports the headline claim in §5.4: REC-CBM is robust to the rubric cardinality K and the encoder hidden dimension d, while the Stage I learning rate is the only hyperparameter whose misconfiguration meaningfully degrades grading.

Concept-count sweep. On all three datasets, task F1 rises steeply between $K{=}1$ and $K{\approx}4$ and then plateaus through the full rubric cardinality (8 / 8 / 7 on Mohler / ASAP 2.0 / MOCHA). The rise is largest where a single shared concept is most starved of evidence: on Mohler, GPT-2 climbs from $\sim$0.18 at $K{=}1$ to $\sim$0.51 at $K{=}2$, and T5 from $\sim$0.11 to $\sim$0.54 by $K{=}8$. RoBERTa dominates across K on every dataset, never falling below the other backbones in the saturated regime. The lone non-monotone trajectory is BART on Mohler, which peaks at $K{=}1$ and then partially recovers near $K{=}8$; this is consistent with BART's lower same-backbone score in Table 2 rather than a defect of the sweep itself. Using each benchmark's full rubric cardinality is therefore safe: it never underperforms a smaller K and matches the interpretability target of one concept per rubric dimension.

Hidden-dimension sweep. Curves are nearly flat for $d \geq 384$ on ASAP 2.0 and MOCHA across all four backbones, indicating that grading quality saturates well below the default $d{=}768$ used in Table 7. The only meaningful capacity effect appears on Mohler: BART and GPT-2 jump from $\sim$0.31 at $d{=}128$ to $\sim$0.53 at $d{=}384$, plausibly because short-answer responses contain little redundancy for under-parameterized concept heads to exploit. RoBERTa is essentially flat across d on every dataset, suggesting that its pretraining already provides representations rich enough for the rubric-aware queries. The default $d{=}768$ therefore sits comfortably in the saturated regime for every backbone–dataset pair, and there is no need to retune d when changing the encoder.

Stage I learning-rate sweep. Every backbone–dataset pair forms an inverted-U whose peak lies in $[10^{-5}, 10^{-4}]$. RoBERTa peaks slightly earlier (closer to 10^{-5}) than BART, GPT-2, and T5, whose optima sit nearer 10^{-4}. Once LR$\geq 10^{-3}$, all backbones collapse to near-zero F1 on Mohler and ASAP 2.0; T5 retains marginal signal on MOCHA at 10^{-3} before joining the collapse at 10^{-2}. This sensitivity is consistent with the two-stage design in §4.4: Stage I jointly updates the text encoder $\mathcal{E}$, the concept-query bank $\mathcal{Q}$, and the per-concept classifiers $\{\mathbf{V}_k\}$, so an oversized LR destabilizes rubric-aware token attention before the calibration loss in Eq. (16) can take effect; Stage II then has no usable concept signal to denoise. The narrow band of bold optima in Table 8 ($1e{-}5$, $2e{-}5$, $1e{-}4$) directly reflects this peak.

Implications. The three sweeps jointly justify two implementation choices in the main experiments. First, we use each benchmark's full rubric cardinality and the default $d{=}768$ across backbones, since both lie inside the saturated regime and incur no degradation. Second, the Stage I learning-rate search in Table 8 is restricted to a narrow band around 10^{-5}–10^{-4}, which avoids the catastrophic regime visible in Fig. 4 while remaining wide enough to accommodate per-backbone preferences.

E.3. Extended Human-Intervention Analysis

This appendix expands Fig. 5 with per-dataset detail that supports the headline claim in §5.5: editing the rubric-level concept predictions moves the final grade in the direction expected by an educator, so the bottleneck is mechanistically load-bearing rather than decorative.

Adversarial decline. Wrong and Random both decline monotonically with the number of intervened concepts k. On Mohler ($k \in [0, 4]$, K=8 rubric concepts but only the top-k are intervened), accuracy under Wrong falls from 0.70 to 0.275 (a 42.5-point drop), while Random falls to 0.375 (a 32.5-point drop). On ASAP 2.0 ($k \in [0, 8]$), both Wrong and Random fall from 0.64 to ≈0.30, with the two trajectories nearly overlapping. On MOCHA ($k \in [0, 7]$), both fall from 0.72 to ≈0.28, again essentially overlapping. The Wrong–Random gap on Mohler reflects the small concept space and three-level ordinal scale, where adversarial flips deterministically choose the most damaging level while uniform random sometimes lands on the original or an adjacent level. The gap closes on ASAP 2.0 and MOCHA where larger K or more uniform marginals make random labels nearly as harmful as adversarial ones.

Oracle improvement. Oracle stays at or above the no-intervention baseline on every dataset and saturates well before k=K: Mohler 0.70→0.75 (+5 points) by k=2; ASAP 2.0 0.64→≈0.70 (+6 points) at k=8, with a steady climb consistent with eight independent rubric dimensions each contributing partial information; MOCHA 0.72→≈0.75 (+3 points) by k=4. The early saturation indicates that the top few concept corrections capture most of the residual measurement error; correcting additional dimensions yields diminishing returns once the dominant rubric signal is fixed.

Why Oracle does not match the maximum possible accuracy. The latent task head $\mathbf{W}$ in Eq. (22) was fit on the calibrated Stage I outputs $\boldsymbol{\mu}_{\text{post}}$, not on one-hot rubric labels; the denoising matrix $\mathbf{A} = \boldsymbol{\Sigma}_{\text{post}}\,\mathbf{D}^{-1}$ shrinks every observation toward the prior mean, so even oracle-substituted concepts are partially attenuated before reaching the task head. This is the intended behavior of the measurement-error model in §4.3: it trades a small amount of intervention head-room for robustness to noisy concept predictions in the no-intervention regime. The fact that Oracle still rises above None confirms that interventions flow through the bottleneck even after this attenuation.

Operational implication. Combining the two sides, a curve where Oracle climbs and Wrong/Random fall is the faithfulness contract behind a usable concept bottleneck: an educator who rewrites a single rubric judgment can move the predicted grade in a predictable direction, and an audit that flags incorrect rubric evidence can be verified by observing the corresponding grade change. Both behaviors are necessary for REC-CBM to support the inspect/intervene/audit workflow argued for in §3.2.

E.4. Extended Ablation Analysis

This appendix expands Fig. 3 with per-component role descriptions, a per-dataset walkthrough of the ablation bars, and a gain decomposition that supports the headline claim in §5.3: the three components of REC-CBM are complementary, so the stack is the only configuration that is uniformly best across the three benchmarks.

Component roles. RACE (§4.1) replaces the shared encoder of Vanilla CBM with a per-rubric concept-query bank, routing each concept head through its own soft-attention pool over the token sequence and thereby isolating dimension-specific textual evidence. OPCC (§4.2) converts the independently trained ordinal heads into a calibrated ranker via a Bradley-Terry-style pairwise log-sigmoid loss on the expected concept scores, preserving the rubric's ordering structure that a plain cross-entropy head discards. LCEC (§4.3) treats the calibrated concept scores as error-corrupted observations of a latent concept vector and applies a closed-form Gaussian posterior correction before the grade head, effectively denoising annotator disagreement without breaking the interpretable bottleneck. Each component targets a distinct Vanilla-CBM failure: evidence entanglement, ordinal information loss, and concept-label noise, respectively.

Per-dataset bars. Mohler exhibits the cleanest monotone ordering—Vanilla CBM ¡ single removals ¡ double removals ¡ full REC-CBM—and the largest cumulative lift, consistent with short-answer grading where every rubric dimension is narrowly observable and benefits from all three corrections stacking. ASAP 2.0 shows a flatter profile: Vanilla CBM is already competitive and the single-removal variants cluster tightly, indicating that long essays give the text encoder enough signal that the three modules sharpen rather than rescue performance. MOCHA is dominated by the −RACE−OPCC outlier where LCEC alone nearly halves T-F1, while the remaining variants occupy a narrow band; the coarse three-level rubric makes LCEC brittle unless RACE and OPCC supply clean, ordinally calibrated upstream scores.

Component-wise gain decomposition. Reading single-vs.-double removal contrasts as marginal contributions, RACE carries

the largest marginal on Mohler and ASAP 2.0, where rubric-aware attention recovers evidence that a shared encoder conflates; OPCC carries the largest marginal on MOCHA, because it is the component whose absence triggers the −RACE−OPCC collapse and it rescues LCEC from correcting an uninformative signal; LCEC's marginal is smaller in isolation but additive when RACE and OPCC are already present, adding a consistent final-step lift across all three datasets. This ordering matches the two-stage training paradigm of §4.4, where Stage I (RACE + OPCC) supplies the clean, ordered concept signal that Stage II (LCEC) denoises.

Operational implication. No single component can be dropped without losing uniform dominance across datasets: RACE is load-bearing on short-answer grading, OPCC is load-bearing when the rubric is coarse, and LCEC is load-bearing as the final denoising step. The full REC-CBM is therefore the recommended configuration for practitioners, with the ablation evidence supporting each component as a default rather than an optional feature.

E.5. Extended Case Study

This appendix expands Fig. 7 with a component-by-component walkthrough of the decision trace that supports the headline claim in §5.7: every stage of REC-CBM is individually inspectable on a single Mohler response. The instance is a short answer to "What is a linked list?" graded by a BERT-backbone REC-CBM, with predicted grade 4 matching the human label at confidence 0.863.

RACE token attention. The colored response spans in Fig. 7 visualize the eight rubric-aware attention heads of §4.1: "sequence of nodes" anchors Factual Correctness (FC) and Depth of Understanding (DU); "pointers that point to the next or previous nodes" anchors Concept Coverage (CC) and Example / Elaboration (EE); the remaining surface markers ("called", "or two") anchor Fluency / Readability (FR) and Clarity / Precision (CP). Each rubric dimension pools a dedicated, human-readable token span rather than sharing a single attention map, matching the interpretability goal of the concept-query bank $\mathcal{Q}$.

OPCC-calibrated ordinal reading. The predicted ordinal levels $\hat{\mathbf{c}}$ are consistent with Table 4: the response is factually correct (FC$=3$), uses appropriate domain terminology (TU$=3$), is fully relevant (R$=3$), clearly expressed (CP$=3$), fluent (FR$=3$), and demonstrates deeper understanding (DU$=3$), but is thin on explicit elaboration (EE$=2$) and partial on coverage (CC$=2$), since the reference answer's "dynamic allocation" aspect is not mentioned. The expected scores $\hat{\mathbf{s}}$ preserve this ordering with a graded margin, e.g., FR (2.907) > TU (2.858) > FC (2.752) > CC (2.276) > EE (1.806), which is exactly the ranking structure the pairwise logistic objective in Eq. (15) was trained to enforce.

LCEC shrinkage. Dividing by $M = 3$ gives the normalized observations $\tilde{\mathbf{s}}$, and the latent posterior mean $\boldsymbol{\mu}_{\text{post}}$ applies approximately uniform shrinkage of factor ~ 0.51 to every coordinate (FR $0.954 \to 0.485$, R $0.944 \to 0.481$, TU $0.929 \to 0.472$, EE $0.403 \to 0.202$). Under the measurement-error model of §4.3, this near-uniform contraction indicates that the learned precision $\boldsymbol{\Omega}$ and noise variances σ_k^2 are roughly isotropic on this instance, so no single rubric dimension is singled out as anomalously noisy; the posterior nevertheless pulls every observation toward the prior mean, preventing any one concept prediction from dominating the grade head.

Grade-head attribution. Because the task head $\mathbf{W}\,\boldsymbol{\mu}_{\text{post}}$ in Eq. (22) is linear, the bar chart at the bottom of Fig. 7 reads as an exact local decomposition of the grade-4 logit: the eight per-concept contributions $W_{4,k}\mu_{\text{post},k}$ sum to 3.32, which together with the near-zero bias reproduces the reported logit (3.318) and softmax confidence (0.863). The ranked contributions (R $+0.75$, CP $+0.65$, DU $+0.54$, EE $+0.49$, FC $+0.35$, FR $+0.35$, CC $+0.20$, TU -0.01) show that content and reasoning dimensions carry the prediction, while Terminology is effectively inert on this response; the grade is therefore not merely produced by the bottleneck but also explained by it, at the granularity of individual rubric dimensions.

E.6. Extended Latent Denoising Analysis

This appendix expands Fig. 6 with the numerical detail that supports the headline claim in §5.6: the learned latent precision $\boldsymbol{\Omega}$ reorganizes densely correlated rubric labels into a sparse, psychometrically interpretable dependency structure. Both panels are computed on Mohler with the BERT-backbone REC-CBM instance used in Fig. 7; the empirical matrix is the Pearson correlation of the expert-validated rubric labels on the training split, and the learned partial correlation matrix is read off from the Stage II precision $\boldsymbol{\Omega}$ via the standard identity $r_{\text{partial},ij} = -\Omega_{ij}/\sqrt{\Omega_{ii}\Omega_{jj}}$.

Dense empirical structure. Among the six content and reasoning dimensions (FC, CC, TU, CP, R, DU), every pairwise correlation falls in $[0.52, 0.89]$, with the strongest links concentrated around Depth of Understanding and the factual core: CC–DU=0.89, FC–DU=0.85, FC–CP=0.81, CP–DU=0.78, FC–CC=0.76, FC–TU=0.76. In contrast, the two surface

dimensions (EE, FR) are only weakly coupled to the content block ($|r| \leq 0.32$ with any content concept) and near-decoupled from each other (EE–FR=-0.04). This dense content-block, loose-surface pattern reflects how rubric evaluators co-assess reasoning and factuality: a response that is factually correct and covers the key ideas tends also to score high on depth, clarity, and terminology, while elaboration and fluency vary along largely orthogonal axes.

Sparsified learned structure. The learned partial correlation matrix is markedly sparser: a majority of off-diagonal entries satisfy $|r_{\text{partial}}| < 0.3$, and the dense content block collapses into a handful of residual links. Using $|r_{\text{partial}}| < 0.1$ as a near-zero threshold, roughly a third of off-diagonal entries are effectively zeroed out in the learned matrix, whereas no off-diagonal of the empirical matrix falls below that threshold. This contraction is the statistical signature of the sparsity penalty $\mathcal{L}_{\text{spa}}$ in Eq. (25), which shrinks off-diagonal entries of the Cholesky factor $\mathbf{L}$ unless the data support a stronger conditional dependency. Under the MMSE correction in Eq. (20), information sharing among concepts is therefore routed through a small number of directed channels rather than spread uniformly across the rubric.

Interpretable residuals. Three families of surviving entries remain. First, residual positive links inside the content block (FC–CC=0.28, FC–DU=0.28, CC–DU=0.27) indicate that factuality, coverage, and depth retain genuine conditional coupling once the remaining rubric dimensions are controlled for. Second, the R–FR partial of 0.36 links relevance to readability, capturing the observation that off-topic responses tend also to be less fluent. Third, a triad of negative suppressors emerges once shared variance is removed: DU–EE=-0.31, DU–FR=-0.29, and most strikingly EE–FR=-0.63. These signs are rubric-consistent: given a fixed level of content and reasoning, a response that adds more elaboration (EE) trades off against surface fluency (FR), and deeper reasoning (DU) similarly trades against polished-but-shallow expression. This depth-versus-surface tension is invisible in the raw correlation matrix but recovered by the latent precision.

Operational implication. A diagonal-$\boldsymbol{\Omega}$ baseline would treat each concept as independently noisy and apply per-concept shrinkage only, reducing to the Spearman reliability case highlighted in the remark after Proposition 1. The learned dense-but-sparse $\boldsymbol{\Omega}$ instead propagates evidence from reliably measured concepts to noisily measured ones along the residual graph above. Combined with the human-intervention behavior in §5.5, this explains why oracle substitutions on a few dominant Mohler content concepts already saturate the intervention curve: once the content block is corrected, the sparse partial-correlation graph transports the correction to the remaining dimensions through a small number of high-magnitude edges.

F. Use of Generative AI

To enhance clarity and readability, we utilized the GPT-5.4 model exclusively as a language polishing tool. Its role was confined to proofreading, grammatical correction, and stylistic refinement—functions analogous to those provided by traditional grammar checkers and dictionaries. This tool did not contribute to the generation of new scientific content or ideas, and its usage is consistent with standard practices for manuscript preparation.