

A Psychometric Framework for Context-Conditional Evaluation of LLM Tutoring in Programming Education

Jinnie Shin
University of Florida
jinnie.shin@ufl.edu

Zhe Li
University of Florida
zheli@ufl.edu

Pauline Aguinalde
University of Florida
aguinalde@ufl.edu

Laura M. Cruz Castro
University of Florida
cruzcastrol@ufl.edu

ABSTRACT

LLM-based chatbots are increasingly used as on-demand tutors in computer science education, yet evaluating pedagogical effectiveness remains difficult because rubric scoring is costly and raw aggregates confound response quality with rater severity, criterion difficulty, and student-query context. We propose a Bayesian explanatory many-facet Rasch model (MFRM) that places rubric-criterion endorsements on a common logit scale and decomposes ratings into latent response quality, criterion difficulty, and rater severity, while incorporating query-level covariates as criterion-specific effects. Applied to 249 CodeHelp exchanges from a large undergraduate C++ course, double-coded by two experts on mistake identification, mistake location, guidance, accuracy of guidance, and correctness of next steps, the model indicates minimal systematic rater differences and pronounced criterion-difficulty gaps. Diagnosis criteria are the most difficult to satisfy, whereas accuracy and next-step criteria are the easiest. Input sufficiency yields modest, criterion-dependent shifts in endorsement probability, and topic patterns are explored descriptively, underscoring that observed tutoring quality depends on what students provide and the type of help they request.

Keywords

educational chatbots; large language models; pedagogical effectiveness; Many-facet Rasch Model

1. INTRODUCTION

LLM-based chatbots, also referred to as conversational agents, are increasingly deployed as on-demand tutors across K–12 and higher education in a wide range of domains [28, 33]. In computer science education in particular, a growing body of work reports benefits for programming-focused learning tasks, including answering students’ questions [36], supporting learning of core programming concepts [32], and assisting with assessment or performance feedback [12, 49]. Unlike earlier chatbots that relied on scripted rules or narrow task-specific models, LLM-based chatbots are *generative*, Jinnie Shin, Zhe Li, Pauline Aguinalde, and Laura Cruz Castro. A Psychometric Framework for Context-Conditional Evaluation of LLM Tutoring in Programming Education. In Anthony Botelho, Maria Mercedes T. Rodrigo, Adish Singla, Hiroaki Ogata, Hyojeong So, and Young Hoan Cho (eds.) Proceedings of the 19th International Conference on Educational Data Mining, Seoul, Republic of Korea, June, 2026, pp. 29–43. International Educational Data Mining Society (2026).

© 2026 Copyright is held by the author(s). This work is distributed under the Creative Commons Attribution NonCommercial NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license.

<https://doi.org/10.5281/zenodo.21039780>

with an enhanced capacity to produce open-ended, context-sensitive explanations and guidance specific to a learner’s prompt. Recent studies have therefore examined a range of LLM-powered tools, including general-purpose models (e.g., GPTs) and code-specialized assistants (e.g., Codex- and Copilot style systems), with emerging evidence that these tools can reshape both pedagogical interactions (e.g., debugging support, conceptual explanation) and assessment-related practices (e.g., feedback generation and evaluation) [55, 19].

At the same time, rapid adoption has intensified the need for rigorous evaluation frameworks that can determine not only whether these systems work, but also how, for whom, and under what conditions they support learning and meaningful educational interactions [21, 1, 28]. To date, evaluation evidence is often assembled by triangulating quantitative outcomes, ranging from proximal indicators such as student completion or engagement [41, 17] to more distal outcomes such as learning gains, together with qualitative analyses including error coding, interaction reviews, and feedback from students and instructors [62, 8, 53]. Complementing these outcome-focused approaches, studies increasingly emphasize pedagogically grounded indicators such as alignment with curriculum standards [3], adaptability to individual learner needs [27], and support for higher-order thinking [14, 57, 8]. Collectively, this literature underscores that chatbot effectiveness is multi-dimensional and should not be reduced to a single score or outcome [31].

However, translating this multi-dimensional view into scalable measurement at the level where tutoring actually occurs, namely, the response and interaction level, remains challenging. In practice, pedagogical effectiveness is still most often judged through expert human annotation, typically via rubric-based scoring of chatbot responses or full interaction transcripts [61, 20, 37]. To improve scalability, recent computational work has explored automated proxies, including reference-based overlap measures (e.g., BLEU, METEOR, ROUGE) and reference-free confidence and similarity scores [10]. However, there is growing concern that these metrics provide limited pedagogical insight [37]. Although they can capture surface properties such as fluency or lexical overlap, they do not directly assess whether a chatbot adapts instruction to a learner’s needs, detects misconceptions, or supports conceptual understanding. This creates a practical tension: expert evaluation remains necessary for pedagogically consequential judgments, yet it is costly and difficult to interpret at scale.

Recent work has also begun to apply Rasch-family models to structure LLM evaluation. Some studies treat chatbots as “examinees” and use Rasch analysis to benchmark model performance relative to item difficulty in test-like formats [7]. Other work uses many-facet Rasch models in human rating systems, often to quantify severity, consistency, or bias when LLMs act as automated scorers or raters [59, 23, 26, 34]. While valuable, these applications do not directly address the most common educational tutoring evaluation setting, in which experts apply pedagogically grounded rubrics to open-ended responses and scores must be interpreted under rater variability, uneven criterion difficulty, and systematic differences in query context.

Against this backdrop, even when pedagogically grounded rubrics are applied by experts, results are often summarized using coarse aggregates such as mean rubric scores, percent endorsement by criterion, or composite ratings [37, 46]. These summaries implicitly treat observed ratings as direct measurements of pedagogical effectiveness, which is problematic for at least three reasons. First, human judgments can vary systematically with rater severity or leniency, producing shifts in scores that are unrelated to the chatbot’s underlying instructional quality [4, 54]. Second, rubric criteria are rarely equally difficult to satisfy, particularly for tutoring behaviors that require diagnosis and targeted remediation, so aggregation can mask which instructional capabilities are genuinely challenging for the system [31, 40, 38]. Third, many evaluations use cost-constrained annotation designs in which different raters score different subsets of responses, creating partially crossed designs where rater severity can become confounded with the specific responses and query contexts being scored [42]. Together, these limitations motivate a psychometric evaluation layer that can separate response quality from rater effects and criterion difficulty, and support context-conditional interpretation of rubric-based tutoring performance.

These limitations are especially consequential in programming support settings. Pedagogical effectiveness depends not only on producing a fluent response, but on accurately diagnosing student errors (e.g., identifying and localizing mistakes), providing conceptually sound guidance, and offering actionable next steps while respecting constraints that discourage directly revealing solutions. At the same time, the learner’s query context strongly conditions what the chatbot can infer and how its response is judged. Students may provide complete code and error traces, partial snippets, or only a high-level description of confusion. Likewise, the requested assistance may range from debugging and error localization to conceptual explanations or strategy-level guidance. Without modeling query sufficiency and question type, it is difficult to determine whether observed rubric scores reflect the chatbot’s underlying instructional capability or systematic variation in the interaction context that shapes both response generation and human judgment.

To address these gaps, we propose a Bayesian explanatory Many-Facet Rasch Model (MFRM) as a psychometric evaluation layer for rubric-based assessment of LLM tutoring responses in authentic programming help interactions. Our approach places rubric-criterion endorsements on a common logit scale and separates observed ratings into (i) latent

response-level pedagogical quality, (ii) criterion difficulty, and (iii) rater severity, enabling uncertainty-aware inference even under human-annotation designs where rater severity and criterion difficulty would otherwise be confounded in simple aggregates. We further incorporate interaction-context information, most notably whether the student provided sufficient inputs (e.g., code and error traces), as criterion-specific explanatory effects, so that tutoring quality can be interpreted conditional on the help-seeking context rather than as a single global average.

This framework yields criterion-level diagnostics that are not available from raw endorsement rates or standard automated metrics and supports more actionable conclusions about where LLM tutors succeed and where they break down. In this study, we focus on mistake-remediation tutoring behaviors in a C++ debugging task and use a rubric that explicitly distinguishes *mistake location* from mistake identification and isolates *accuracy of guidance* to capture potentially harmful hallucinated or misapplied recommendations. Guided by this evaluation framework, we address the following research questions:

- RQ1.** Which pedagogical rubric criteria are easiest versus most difficult for an LLM-based programming tutor to satisfy, after accounting for rater severity?
- RQ2.** How does student-query sufficiency shift the probability that the chatbot satisfies each pedagogical criterion in a programming task?
- RQ3.** How does student question content relate to criterion-level pedagogical effectiveness in a programming task? (*exploratory*)

We make three contributions. First, we operationalize mistake-remediation tutoring quality for programming support with a rubric and annotation protocol that separates diagnosis (identifying and locating mistakes) from downstream guidance and correctness. Second, we introduce a Bayesian explanatory MFRM that converts rubric-based human judgments into facet-adjusted, uncertainty-aware estimates of pedagogical effectiveness and quantifies how evaluation shifts with query sufficiency. Third, we provide empirical evidence that diagnosis is the limiting capability for the studied tutor and that commonly used reference-free automated metrics capture only a weak proxy for correctness-focused outcomes.

2. RELATED WORK

2.1 LLM Chatbots as Instructional Agents

LLM-based chatbots have emerged in recent years, advancing beyond traditional rule-based algorithms, which have been limited in language understanding. Additionally, responses from rule-based algorithms are planned and predefined [48], which pose challenges regarding unexpected inputs by learners. In contrast, LLM-based chatbots directly address this constraint by training on large-scale datasets, allowing it to capture more contextual and semantic understanding [61], forgoing the need for rigid pre-planning for every potential input. Furthermore, its generative ability to produce natural language supports student learning experiences. Interactions with LLM-based instructional agents are

adaptive, provide timely feedback, and serve as an effective resource for on-demand information [2, 45, 43, 44].

LLM-based chatbots are rapidly reshaping programming education by providing on-demand, natural-language tutoring that can explain concepts, diagnose errors, and support debugging workflows in ways that earlier pre-defined rules or narrowly trained course chatbots struggled to achieve [56]. In authentic course settings, students use LLM tutors for a wide range of needs, but the dominant pattern is highly task-proximal help on assignments (e.g., fixing code, interpreting errors), with comparatively fewer interactions aimed at deeper conceptual inquiry, which raises both opportunity and risk for learning [47]. Recent systems for computing education have introduced instructional guardrails that avoid revealing complete solutions while still offering actionable, context-aware guidance, demonstrating that LLM chatbots can be intentionally shaped toward pedagogically meaningful behaviors [29, 25].

At the same time, evidence from programming courses suggests that students’ use of general-purpose chatbots (e.g., ChatGPT) often concentrates on error checking and debugging and may be perceived as helpful for understanding and productivity, yet it can also introduce concerns about code quality, learning, and collaboration practices such as pair programming [18]. Alongside these classroom deployments, research prototypes increasingly combine LLM generation with program-analysis constraints to deliver finer-grained “next-step” hints, aiming to preserve learner agency while improving technical correctness and usefulness [5]. Collectively, this literature motivates treating LLM-based programming tutors not as monolithic tools but as designable instructional systems whose pedagogical value depends on their guardrails, the kinds of help students seek, and the learning context in which they are embedded [25, 47, 18].

2.2 Chatbot Pedagogical Efficacy Evaluation

Approaches to evaluating chatbots vary widely depending on the evaluation goal and the construct of interest. Prior reviews highlight that chatbot evaluation spans multiple perspectives, including user experience, information retrieval, linguistic quality, technical performance, and business outcomes, each emphasizing different attributes and metrics [39]. For instance, linguistic evaluations emphasize response correctness and coherence, whereas user-experience evaluations focus on usability and task completion. Within education, commonly reported objectives include acceptance and adoption, learning outcomes, motivation and affect, usability, technical correctness, and other psychological factors [21]. In this study, we focus specifically on *pedagogical efficacy*, which is defined as the extent to which a chatbot supports learning through instructionally appropriate behaviors (e.g., promoting active learning, adapting support to learner needs, managing cognitive load) [24, 31].

Given these objectives, chatbot evaluations are typically conducted through Wizard-of-Oz studies, controlled laboratory experiments, field deployments, or technical validation studies [39]. Measurement within these designs most often relies on human judgments or user-reported instruments (e.g., rubrics, rating scales, surveys), sometimes complemented by behavioral indicators such as task success or

learning gains. For example, to operationalize pedagogical ability, Maurya et al. propose eight rubric dimensions rated by human annotators on a 3-point scale [31]. More recently, automated evaluation has also been explored, particularly through lexical-overlap or classification metrics (e.g., BLEU, accuracy, F1, precision/recall) and related reference-based similarity measures (e.g., METEOR, ROUGE, CIDEr), which assume the availability of ground-truth target responses [13, 10, 35]. However, because pedagogical efficacy depends on context-sensitive instructional moves rather than surface similarity to a single reference, these automated metrics are often insufficient on their own, which helps explain why human judgment remains the dominant evidence source for pedagogical evaluation.

Consequently, the empirical literature most often reports effectiveness through aggregated human ratings or rankings. This spans the broader scope of the effectiveness of conversational agents, including educational chatbots. For example, the BEA 2023 shared task evaluated candidate *AI-teacher* responses using 298 crowd raters from Prolific [50], where each interaction was judged by multiple raters and systems were primarily compared via average ranks (1=best to 4=worst) and aggregate leaderboard scores. Similar aggregation based reporting is also common in the broader conversational-systems literature, where small panels of annotators or crowd workers rate responses on a handful of quality dimensions and results are reported as mean or composite scores [22, 16]. For instance, Jiang and Ahuja [22] collected three assessors’ judgments per response on informativeness, fluency, and human-likeness and used the mean rating as the primary quality measure. Gao and Jiang [16] similarly recruited crowd workers to rate responses on multiple attributes (e.g., usefulness, accuracy, fluency, human-likeness) and reported performance via aggregated scores across conditions.

More education-focused LLM-tutor evaluations also follow this pattern, but with limited crossing and sparse overlap for reliability estimation. For instance, Daheim et al. [11] annotated a random set of 40 conversations and computed inter-rater reliability using only a small overlapping subset (9 conversations annotated by at least two raters), while still presenting results as criterion-level or overall aggregates. Even when evaluation is framed as comparative judgment, the reporting often remains rank- or mean-based. A few studies proposed to use pairwise comparative judgments and a probabilistic ranking model to infer pedagogical ability, yielding system-level scores and rankings rather than rater- and criterion-adjusted measures [51]. Collectively, this literature provides clear evidence that (a) evaluation commonly depends on human labels collected under partial crossing (different raters score different subsets), (b) outcomes are typically communicated as means, ranks, or composites, and (c) rater disagreement is acknowledged via inter-rater reliability but rarely incorporated into the final scoring model, thus, motivating measurement-model-based approaches that explicitly separate latent pedagogical quality from rater severity and criterion difficulty.

2.3 Bayesian explanatory MFRM

The LLM-based chatbot response to a student’s single query is identified as a pair of *Chatbot response* and *student query*

that are rated by multiple raters. A Bayesian explanatory many-facet Rasch model (MFRM) is a Rasch-family model that decomposes the score or rating into multiple additive facets on a common logit scale. These typically involve criterion or item difficulty, response quality, and rater severity, while also allowing other covariates to explain systematic variation in ratings. This study showcases a novel application of MFRM by allowing the quality of chatbot’s response as an evaluation target, and introducing other critical covariates that can systematically influence pedagogical quality.

Specifically, let $i \in \{1, \dots, I\}$ index student–chatbot exchanges (responses), $j \in \{1, \dots, J\}$ index rubric criteria, and $r \in \{1, \dots, R\}$ index human raters. Let $y_{ijr} \in \{0, 1\}$ denote a binary endorsement indicating whether rater r judged response i as satisfying criterion j (1=endorsed, 0=not endorsed). We analyze the data in long format and index each observed rating by $n \in \{1, \dots, N\}$ with mappings $i(n)$, $j(n)$, and $r(n)$. The model decomposes endorsements into (i) response-level latent pedagogical quality θ_i (higher values imply higher endorsement probability across criteria), (ii) criterion difficulty b_j (larger values indicate harder-to-satisfy criteria), and (iii) rater severity s_r (larger values indicate stricter raters, reducing endorsement probability), while allowing criterion-specific effects of query context covariates.

$$y_n | p_n \sim \text{Bernoulli}(p_n), \quad \text{logit}(p_n) = \eta_n,$$

$$\eta_n = \theta_{i(n)} - b_{j(n)} - s_{r(n)} + \gamma_{j(n)} v_n + \sum_{k=1}^K \beta_{j(n)k} x_{nk}. \quad (1)$$

To characterize how the help-seeking context conditions rubric endorsement, we included two sets of query-level covariates. First, *input sufficiency* was encoded as a binary indicator $v_n \in \{0, 1\}$, where $v_n = 1$ denotes that the student query provided sufficient information (e.g., code, error traces, and/or descriptive context) for a meaningful diagnosis and response.¹ Second, we represented the *CS topic content* of each student query using K multi-label binary indicators $\{x_{nk}\}_{k=1}^K$, where $x_{nk} = 1$ if topic k is present in the query and $x_{nk} = 0$ otherwise (multiple topics may be active for the same query). Under this specification, γ_j captures the criterion-specific shift in endorsement associated with sufficient inputs, and β_{jk} captures the criterion-specific association between topic k and endorsement.

Because the logistic Rasch family is location-invariant, we imposed sum-to-zero constraints on additive facets via centering. Operationally, we sampled unconstrained raw parameters and applied mean-centering transformations: $\theta_i = \theta_i^{\text{raw}} - \bar{\theta}^{\text{raw}}$ and $s_r = s_r^{\text{raw}} - \bar{s}^{\text{raw}}$. For the topic coefficients, we imposed a within-criterion sum-to-zero constraint across the K topic effects, $\sum_{k=1}^K \beta_{jk} = 0$ for each criterion j , implemented as $\beta_{jk} = \beta_{jk}^{\text{raw}} - \bar{\beta}_{j\cdot}^{\text{raw}}$, where $\bar{\beta}_{j\cdot}^{\text{raw}} = \frac{1}{K} \sum_{k=1}^K \beta_{jk}^{\text{raw}}$. This constraint is compatible with multi-label topic coding because it centers the criterion-specific topic coefficients, not the observations.

¹If input sufficiency was coded separately by each rater, then v_n is indexed by the rater in observation n ; otherwise, v_n is treated as an exchange-level label shared across ratings for the exchange.

We used weakly informative Normal priors: $\theta_i^{\text{raw}}, b_j, s_r^{\text{raw}} \sim \mathcal{N}(0, 1)$ and $\gamma_j, \beta_{jk}^{\text{raw}} \sim \mathcal{N}(0, 0.5^2)$. Posterior inference targeted $p(\Theta, \mathbf{b}, \mathbf{s}, \gamma, \mathbf{B} | \mathbf{y})$, where $\Theta = \{\theta_i\}_{i=1}^I$, $\mathbf{b} = \{b_j\}_{j=1}^J$, $\mathbf{s} = \{s_r\}_{r=1}^R$, $\gamma = \{\gamma_j\}_{j=1}^J$, and $\mathbf{B} = \{\beta_{jk}\}_{j=1, k=1}^{J, K}$.

$$p(\Theta, \mathbf{b}, \mathbf{s}, \gamma, \mathbf{B} | \mathbf{y}) \propto \prod_{n=1}^N \text{Bernoulli}(y_n | \text{logit}^{-1}(\eta_n)) \quad (2)$$

$$\times p(\Theta) p(\mathbf{b}) p(\mathbf{s}) p(\gamma) p(\mathbf{B}).$$

We used Hamiltonian Monte Carlo with the No-U-Turn Sampler (NUTS), running 4 chains with 1,000 warmup iterations and 1,000 retained posterior draws per chain. Convergence was assessed using \hat{R} and effective sample size.

For interpretable effect sizes, we translated selected effects into probability changes as the coefficients were estimated on the log-odds scale. For instance, for input sufficiency effect, for each rubric criteria j , we defined a baseline linear predictor at a reference point $\theta^*, s^*, \mathbf{x}^* = 0$ as follows:

$$\eta_{0j} = \theta^* - b_j - s^* \quad (3)$$

Then, the probability shift induced by input sufficiency is expressed as follows, where $\Delta P_j^{(suff)}$ was computed for each posterior draw to obtain a posterior distribution over probability-scale effects, summarized by posterior mean and 94% HDI.

$$\Delta P_j^{(suff)} = \sigma(\eta_{0j} + \gamma_j) - \sigma(\eta_{0j}), \quad \sigma(t) = \frac{1}{1 + e^{-t}} \quad (4)$$

3. METHODS

3.1 Participants and Data Collection

This study was conducted in a large-enrollment undergraduate computer science course at a public, research-intensive university in the southern United States (Spring 2024; $N = 669$)². The course emphasized object-oriented programming in C++, and students could voluntarily use CodeHelp [29], which is an LLM-based programming tutor designed to provide debugging-oriented guidance while discouraging solution revealing, during a course assignment. We analyzed 249 unique, single-turn student-chatbot exchanges (one per student), each pairing a student query (including any provided code/error context captured by the platform) with the corresponding chatbot response. We excluded attempted submissions that CodeHelp flagged as insufficient and for which no response was returned.

3.2 Programming Task and Interaction Inputs

Students completed a C++ programming task targeting an intermediate object-oriented programming concept commonly referred to as the “Big Three” implementation (Fig. 4). The task required implementing a destructor, copy constructor,

²This study was conducted in accordance with institutional ethical guidelines for research involving human participants. The research protocol was reviewed and approved by the Institutional Review Board (IRB) RB202400344. All data involving human participants were handled in compliance with relevant privacy and ethical standards.

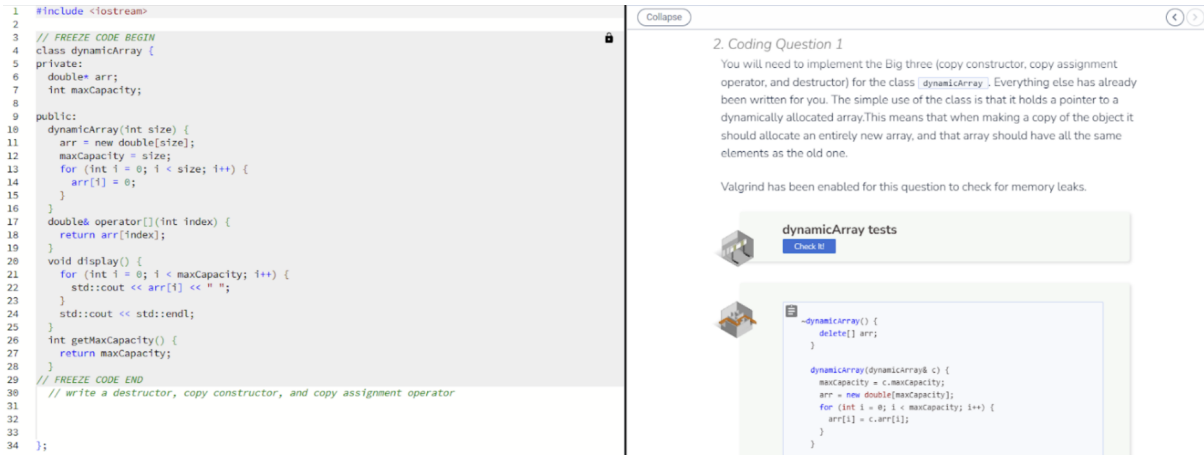


Figure 1: Programming Task Description

and copy assignment operator for a class that manages dynamically allocated memory and resources. The programming environment integrated Valgrind to detect runtime memory errors and memory leaks to support debugging. When interacting with CodeHelp, students could provide information via one or more fields: (1) *Code* (a snippet of their implementation), (2) *Error* (compiler output or Valgrind messages), and (3) *User Input* (a natural-language description of the issue). For each exchange, we logged the contents of these fields and the chatbot response.

3.3 Chatbot Evaluation

3.3.1 Rubric Development and Query Labels

To evaluate the pedagogical quality of chatbot responses in a mistake-remediation programming context, we adapted the unifying framework for mistake-remediation tasks [31, 24, 38]. The framework emphasizes core dimensions of tutoring quality (e.g., mistake remediation and guidance/actionability). Because our task required assessing whether the chatbot could pinpoint where an error occurred in student code, we introduced a dedicated *Mistake Location* dimension. To distinguish helpful guidance from potentially harmful or hallucinated recommendations, we additionally included an *Accuracy of Guidance* dimension. Table 1 summarizes all rubric dimensions. Detailed decision rules and benchmark examples used in annotator training are provided in Appendix A.

For modeling, we used a binary endorsement format. For dimensions i1–i4, raters initially assigned *yes/partial/no*, and we recoded *partial* as *no* (i.e., Yes= 1, Partial/No= 0). In this study, we treated *partial* as not-fully-met to target a conservative mastery-like endorsement. Dimension i5 was coded as Yes/No. In addition to rubric-based response-quality criteria, raters also coded two auxiliary query-level labels: *Input Sufficiency* (S; whether the student provided enough information for a meaningful response) and *CS topic categories* describing the content of the student query. Topic codes were multi-label (a single exchange could receive multiple topic categories). Table 2 summarizes the distribution of topic labels across exchanges.

3.3.2 Annotator Training and Coding Procedure

Table 1: Rubric Dimensions for Chatbot Evaluation

ID	Dimension	Scale	Description
Rubric dimensions (chatbot response quality)			
i1	Mistake Identification	yes partial no	Does the chatbot accurately recognize a mistake in the student’s input?
i2	Mistake Location	yes partial no	Does the chatbot correctly identify the location/aspect of the student’s mistake?
i3	Providing Guidance	yes partial no	Does the chatbot offer guidance relevant to the identified issue?
i4	Accuracy of Guidance	yes partial no	Is the guidance factually and conceptually correct?
i5	Correctness of Next Step	yes no	Are the suggested next steps and corrections accurate and appropriate?
Auxiliary query-level labels			
S	Input Sufficiency	yes no	Did the student provide enough information (e.g., code, error, context) for a meaningful response?

Table 2: Student Queries by Relevant CS Topics (multi-label)

Category	Description	Freq
Topic 1	Programming Fundamentals	146
Topic 2	Data Management and Processing	130
Topic 3	Object-Oriented Programming (OOP)	176
Topic 4	Code Construction	216
Topic 5	Language Unique Programming Concepts	239

Two subject matter experts independently coded all 249 student–chatbot exchanges. Before full coding, raters reviewed the task description, rubric definitions, and benchmark examples, and completed a structured training session to align decision rules. Raters then double-coded a random set of 20 exchanges and participated in a calibration meeting to resolve ambiguities and refine guidelines. A midpoint check-in was conducted to reinforce decision rules and ensure continued consistency. Disputed training cases were resolved by consensus and incorporated into an updated benchmark set used for ongoing reference during coding. Unless otherwise noted, analyses used both raters’ original codes to estimate and adjust for rater severity within the many-facet Rasch framework.

Table 3: Model comparison using PSIS-LOO

Model	elpd _{loo}	SE	Δelpd	p _{loo}	Wt.	Rank
$M1_{\text{suff}}$	-1419.00	22.23	2.45	135.55	1.00	0
$M0_{\text{baseline}}$	-1421.45	22.18	0.00	135.20	0.00	1
$M3_{\text{both}}$	-1428.32	22.72	-6.87	153.87	0.00	2
$M2_{\text{topics}}$	-1430.19	22.69	-8.74	153.42	0.00	3

Note. Higher elpd_{loo} indicates better expected out-of-sample predictive fit. Δelpd is relative to $M0_{\text{baseline}}$.

Table 4: Observed endorsement rates by raters

Rater	i1	i2	i3	i4	i5
Rater 1	0.530	0.542	0.406	0.892	0.827
Rater 2	0.510	0.482	0.394	0.880	0.867

4. RESULTS

All models were fit to a long-format dataset with $N = 2,490$ binary observations (249 exchanges \times 5 rubric criteria \times 2 raters). Submissions flagged by our LLM-based chatbot, CodeHelp, as insufficient were excluded prior to annotation because no response was returned in such cases; SMEs labeled 5% as insufficient on the Input Sufficiency label (S). For modeling, *partial* ratings were recoded as 0 (not endorsed) to yield binary endorsements..

We evaluated out-of-sample predictive performance using PSIS-LOO computed from 4,000 posterior draws and the pointwise log-likelihood matrix. For the facets-only baseline model ($M0_{\text{baseline}}$), PSIS-LOO indicated stable importance sampling with all Pareto- k values ≤ 0.70 (100% “good”). We then compared four nested specifications (Table 3): a facets-only baseline ($M0_{\text{baseline}}$), a primary explanatory model adding criterion-specific input-sufficiency effects ($M1_{\text{suff}}$), and extended models that additionally included multi-label topic indicators ($M2_{\text{topics}}$) or both covariate sets ($M3_{\text{both}}$). Adding input sufficiency produced a small improvement in expected out-of-sample fit relative to the baseline ($\Delta\text{elpd}_{\text{loo}} = +2.45$), whereas including topic indicators increased effective model complexity ($p_{\text{loo}} \approx 153$) without improving PSIS-LOO, where $\Delta\text{elpd}_{\text{loo}} < 0$ for $M2_{\text{topics}}$ and $M3_{\text{both}}$. Accordingly, we use $M1_{\text{suff}}$ as the primary model for inference in RQ1–RQ2 (criterion difficulty and input sufficiency effects) and treat topic-based analyses (RQ3) as exploratory. We summarize posterior uncertainty using 94% highest-density intervals (HDIs). Across the 249 exchanges, the two raters showed broadly similar endorsement patterns across criteria (Table 4). Here, we use *endorsement* to denote cases in which a criterion was judged as met (i.e., rated **yes**; see rubric in Table 1). Both raters endorsed the chatbot most frequently on Accuracy of Guidance (i4) and correctness of next steps (i5), and least frequently on Providing Guidance (i3), with intermediate endorsement for mistake identification and localization (i1–i2). This suggests that the chatbot more often produced factually correct and actionable guidance once it engaged with the problem, but struggled more consistently with diagnosing and responding to the student’s specific error.

Table 5: Posterior mean rater severity estimates

Rater	\bar{s}_r
Rater 1	-0.036
Rater 2	+0.036

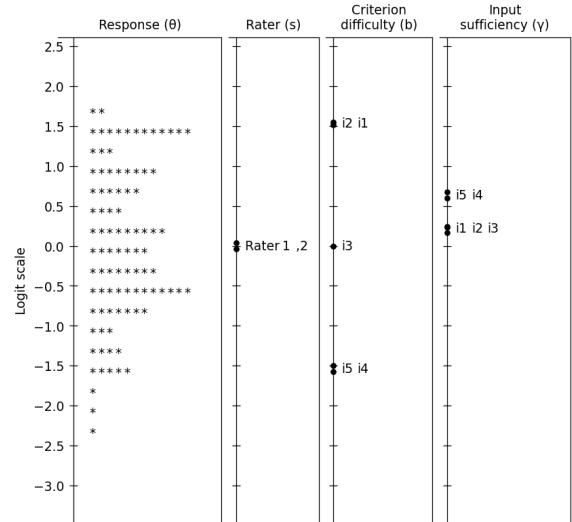


Figure 2: Posterior parameter map for the primary MFRM model ($M1_{\text{suff}}$).

We then fit a Bayesian explanatory Many-Facet Rasch Model (MFRM) to disentangle rater severity, criterion difficulty, and response-level pedagogical quality. Posterior mean rater severities were near zero and symmetric ($s_{\text{rater1}} = -0.036$, $s_{\text{rater2}} = +0.036$), indicating minimal systematic differences in endorsement tendency after accounting for criterion difficulty and response quality (Table 5). Response-level pedagogical quality estimates (θ_i ; $I = 249$) were centered near zero by model identification ($\theta \approx 0$, $SD \approx 1$) and spanned a broad range (min = -2.45, max = 1.60), indicating substantial heterogeneity in the chatbot’s pedagogical effectiveness across student queries and underscoring the need to model response quality beyond raw aggregate endorsement rates.

4.1 Rubric criterion difficulty (RQ1)

Using the primary model $M1_{\text{suff}}$ (facets + input sufficiency), criterion difficulty estimates (b_j) indicated substantial differences in how challenging each pedagogical criterion was for the chatbot to satisfy. Larger b_j corresponds to a lower probability of endorsement, holding response quality and rater severity constant. The mistake diagnosis criteria were most difficult: Mistake Identification (i1) and Mistake Location (i2) had the largest positive posterior mean difficulties ($\bar{b}_{i1} = 1.517$, $\bar{b}_{i2} = 1.553$). Providing Guidance (i3) was close to the reference point ($\bar{b}_{i3} \approx 0$). In contrast, the accuracy-oriented criteria were easiest: Accuracy of Guidance (i4) and Correctness of next steps (i5) had negative difficulty estimates ($\bar{b}_{i4} = -1.577$, $\bar{b}_{i5} = -1.493$), indicating these behaviors were endorsed more readily. For interpretability, we also report baseline endorsement probabilities at a common reference setting ($\theta = 0$, $s = 0$, and $v = 0$). Baseline endorsement was lowest for i1 (0.188) and i2 (0.183), moderate for i3 (0.500), and high for i4 (0.819) and i5 (0.807) (Table 6). Together, these results suggest that the primary difficulty for the chatbot lies in *diagnosing* and *localizing* student errors, rather than in producing generally accurate or actionable guidance once an issue is addressed.

Table 6: Rubric criterion difficulty and baseline endorsement probability at the reference point

ID	Criterion	\bar{b}_j	94% HDI	\bar{p}_{base}	94% HDI
i1	Mistake identification	1.517	[0.820, 2.451]	0.188	[0.079, 0.306]
i2	Mistake location	1.553	[0.835, 2.516]	0.183	[0.075, 0.303]
i3	Providing guidance	-0.000	[-0.773, 0.768]	0.500	[0.317, 0.684]
i4	Accuracy of guidance	-1.577	[-2.666, -0.812]	0.819	[0.693, 0.935]
i5	correctness of next steps	-1.493	[-2.481, -0.724]	0.807	[0.674, 0.923]

Table 7: Input sufficiency effects by criterion.

ID	$\bar{\gamma}_j$	94% HDI	\bar{p}_{base}	ΔP	94% HDI
i1	0.242	[-0.384, 0.908]	0.188	0.037	[-0.067, 0.143]
i2	0.237	[-0.403, 0.871]	0.183	0.035	[-0.069, 0.135]
i3	0.169	[-0.451, 0.867]	0.500	0.041	[-0.109, 0.209]
i4	0.596	[-0.107, 1.278]	0.819	0.072	[-0.015, 0.169]
i5	0.672	[-0.007, 1.371]	0.807	0.083	[-0.008, 0.188]

Note. ΔP summarizes the probability change induced by sufficiency at the reference point ($\theta = 0$, $s = 0$, covariates = 0).

4.2 Effects of input sufficiency (RQ2)

Using the primary model $M1_{\text{suff}}$ (facets + input sufficiency), we examined whether rater-judged input sufficiency (i.e., whether the student query contained enough information to diagnose and address the issue) was associated with higher endorsement probabilities via criterion-specific effects (γ_j). Posterior mean sufficiency effects were positive across all criteria (Table 7), with larger mean shifts for the accuracy-oriented criteria (i4, i5). On the probability scale, the estimated mean changes were modest for the diagnosis and guidance criteria ($\Delta P \approx 0.037$ – 0.041 for i1–i3) and larger for the accuracy-oriented criteria ($\Delta P \approx 0.072$ – 0.083 for i4–i5). However, the 94% HDIs indicate substantial uncertainty, consistent with the small proportion of exchanges labeled as insufficient (5%) in the current dataset.

Figure 2 visualizes posterior parameter estimates from $M1_{\text{suff}}$, with all panels plotted on a common logit scale for comparison. The response-quality distribution (θ , left) shows substantial heterogeneity across student–chatbot exchanges. Rater severity estimates (s , second) are centered near zero for both raters, indicating minimal systematic differences in endorsement tendency. Criterion difficulty (b , third) differentiates the rubric criteria, with mistake diagnosis (i1–i2) the most difficult and the accuracy-oriented criteria (i4–i5) the easiest. Finally, the input-sufficiency effects (γ , right) are generally positive but imprecisely estimated, suggesting that more sufficient student queries may increase endorsement probabilities, with the largest expected gains for the accuracy-related criteria.

4.3 Exploratory Analysis for Effects of student query topic (RQ3)

We conducted an exploratory analysis of whether student question content was associated with criterion-level rubric outcomes. Specifically, we used multi-label indicators for five CS topic categories (Table 2) to summarize topic-conditional differences in endorsement and to estimate topic-augmented MFRM specifications (Figure 3; Table 8). Because adding topic indicators did not improve PSIS-LOO relative to the

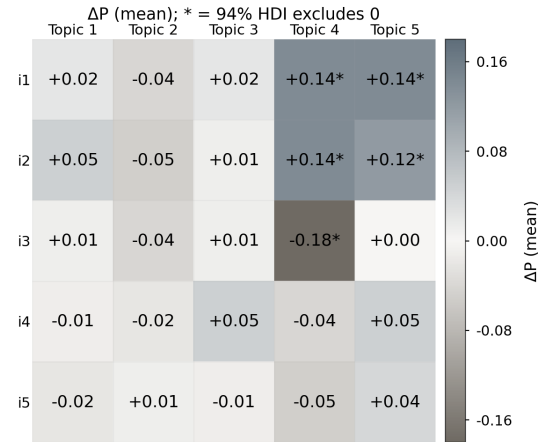


Figure 3: Topic effects by rubric criterion

primary model and increased effective complexity (Table 3), we interpret topic patterns as descriptive and hypothesis-generating rather than confirmatory effects.

Overall, topic-related variation was most apparent for the diagnosis-related criteria, Mistake Identification (i1) and Mistake Location (i2), and for Providing Guidance (i3). In contrast, topic associations for the accuracy-oriented criteria (i4–i5) were smaller and more uncertain, with 94% HDIs often spanning zero. In the topic-augmented model, topic effects are summarized as posterior mean probability shifts (ΔP) at a common reference point ($\theta = 0$, $s = 0$, input sufficiency $v = 0$, and other topic indicators set to 0), enabling criterion-by-topic comparisons on the probability scale.

For mistake diagnosis, the largest positive associations were observed for *Code Construction* (Topic 4) and *Language-Unique Programming Concepts* (Topic 5). For i1, Topic 4 and Topic 5 were associated with increases of approximately $\Delta P \approx +0.14$, with 94% HDIs excluding zero; a similar

Table 8: Exploratory topic associations with criterion endorsement (topic-augmented model).

	Topic	$\bar{\beta}$	94% HDI	ΔP	94% HDI	Topic	$\bar{\beta}$	94% HDI	ΔP	94% HDI	
i1	1	0.103	[-0.390, 0.666]	0.019	[-0.057, 0.119]	i1	2	-0.308	[-0.827, 0.226]	-0.039	[-0.111, 0.036]
i1	3	0.113	[-0.324, 0.555]	0.019	[-0.047, 0.098]	i1	4	0.774	[0.209, 1.308]	0.142	[0.028, 0.268]
i1	5	0.768	[0.068, 1.450]	0.139	[0.015, 0.275]	i2	1	0.320	[-0.238, 0.875]	0.055	[-0.034, 0.165]
i2	2	-0.390	[-0.958, 0.131]	-0.047	[-0.128, 0.017]	i2	3	0.062	[-0.365, 0.486]	0.011	[-0.056, 0.079]
i2	4	0.777	[0.236, 1.333]	0.140	[0.023, 0.257]	i2	5	0.713	[0.022, 1.401]	0.125	[-0.005, 0.247]
i3	1	0.049	[-0.539, 0.633]	0.012	[-0.129, 0.146]	i3	2	-0.170	[-0.780, 0.413]	-0.040	[-0.190, 0.095]
i3	3	0.050	[-0.395, 0.506]	0.012	[-0.097, 0.117]	i3	4	-0.807	[-1.396, -0.247]	-0.185	[-0.316, -0.059]
i3	5	0.014	[-0.682, 0.711]	0.003	[-0.165, 0.169]	i4	1	-0.035	[-0.652, 0.598]	-0.010	[-0.106, 0.087]
i4	2	-0.096	[-0.783, 0.539]	-0.019	[-0.126, 0.078]	i4	3	0.369	[-0.145, 0.914]	0.045	[-0.028, 0.118]
i4	4	-0.274	[-0.986, 0.401]	-0.045	[-0.165, 0.064]	i4	5	0.376	[-0.414, 1.189]	0.047	[-0.056, 0.157]
i5	1	-0.111	[-0.753, 0.503]	-0.020	[-0.128, 0.078]	i5	2	0.093	[-0.562, 0.717]	0.010	[-0.091, 0.099]
i5	3	-0.060	[-0.591, 0.489]	-0.011	[-0.106, 0.067]	i5	4	-0.307	[-0.983, 0.344]	-0.053	[-0.180, 0.055]
i5	5	0.292	[-0.427, 1.039]	0.041	[-0.066, 0.151]						

Note. Each row reports the criterion–topic effect $\beta_{j,k}$ and the corresponding probability shift ΔP at the reference point ($\theta = 0$, $s = 0$, and other topic indicators set to 0). 94% HDIs are computed from the posterior.

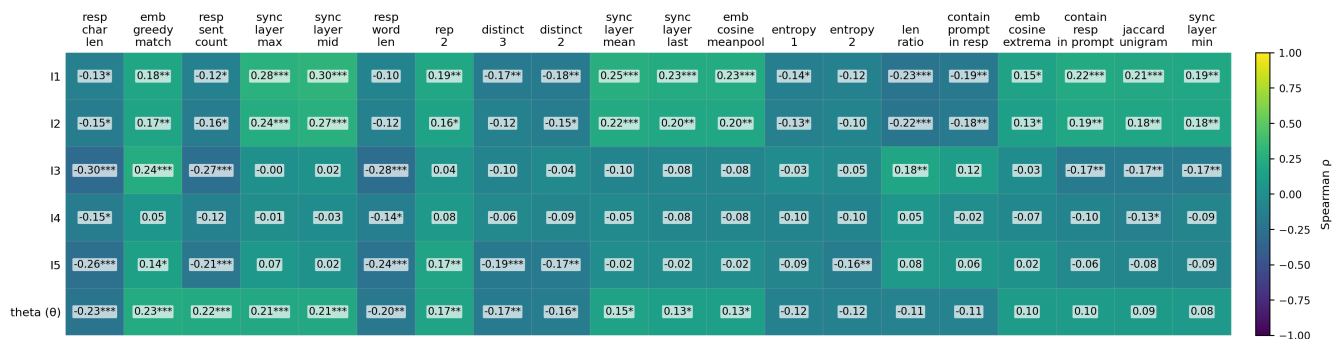


Figure 4: Spearman’s correlations between the evaluation metrics and target rubric

pattern emerged for i2 (Topic 4: $\Delta P \approx +0.14$; Topic 5: $\Delta P \approx +0.12$). By comparison, *Programming Fundamentals* (Topic 1), *Data Management and Processing* (Topic 2), and *Object-Oriented Programming* (Topic 3) showed smaller shifts for i1 and i2 (typically $|\Delta P| \leq 0.05$) with uncertainty intervals overlapping zero.

Providing Guidance (i3) exhibited an asymmetric pattern: most topics were near zero, whereas *Code Construction* (Topic 4) was associated with a notable decrease in endorsement ($\Delta P \approx -0.18$; 94% HDI excluding zero). Taken together, this pattern suggests that topic differences may be most consequential for upstream diagnosis and for translating diagnosis into actionable guidance. However, given the lack of PSIS-LOO improvement for topic-augmented models, these results should be interpreted cautiously and motivate future work with larger or more balanced samples to test topic moderation more decisively.

4.4 Comparisons to Automated Metrics

To contextualize our rubric-based evaluation and provide convergent and discriminant validity evidence for the pedagogical constructs targeted by i1–i5, we compared rubric outcomes against a set of automated, reference-free metrics computed from each student–chatbot exchange. These included (a) *surface length and structure* (response character length, word length, sentence count, and the response-to-

prompt length ratio), (b) *lexical diversity and redundancy* (type–token ratio; distinct- n for $n \in 1, 2, 3$; normalized n -gram entropy for $n \in 1, 2$; and repetition rates for unigram and bigram reuse), (c) *prompt–response overlap* (unigram and bigram Jaccard overlap; prompt containment in the response and vice versa), and (d) *readability* (Flesch Reading Ease and Flesch–Kincaid Grade). To approximate semantic relatedness without an external reference, we additionally computed *embedding-based similarity* between the prompt and response (mean-pooled cosine similarity, vector-extrema cosine similarity, and greedy token-level matching) and *layerwise synchrony* scores, defined as the cosine similarity between mean-pooled transformer representations across layers and summarized by the minimum, mean, maximum, mid-layer, and last-layer similarity. For each metric, we computed Spearman rank correlations (ρ) with (a) criterion-level endorsements (i1–i5) and (b) the response-level pedagogical quality estimate (θ) from the MFRM, and we summarize the correlation pattern in Figure 4. These correlations provide convergent evidence when rubric scores align with the signal captured by a specific class of metrics.

Overall, the correlation coefficients between automated metrics and rubric-based outcomes were generally modest (typically $|\rho| \leq .30$), indicating that commonly used reference-free metrics capture only partial information about pedagogical response quality. The MFRM latent quality estimate

θ aligned most consistently with semantic relatedness between the student query and the chatbot response, including embedding-based relevance (greedy matching; $\rho = .23^{***}$) and synchrony-based similarity (sync layer max and mid; $\rho = .21^{***}$), with smaller positive associations for overlap and reuse proxies (rep-2; $\rho = .17^{**}$). In contrast, θ was negatively associated with verbosity and lexical diversity indicators, including response length ($\rho \approx -.20$ to $-.23$) and distinctness measures (distinct-2/3; $\rho \approx -.16$ to $-.17$). Together, these findings suggest that higher-quality chatbot responses, as captured by θ , tend to be more directly aligned with the student prompt and more concise, rather than longer or more lexically diverse.

At the rubric-criterion level, the clearest alignment with automated metrics emerged for the chatbot’s *diagnosis* behaviors. Mistake Identification (i1) and Mistake Location (i2) were positively associated with prompt-response semantic relatedness, including synchrony-based similarity (sync layer max and mid; i1 $\rho = .28^{***}$ and $.30^{***}$; i2 $\rho = .24^{***}$ and $.27^{***}$) and lexical overlap (unigram Jaccard; i1 $\rho = .21^{***}$; i2 $\rho = .18^{**}$). Both criteria were also negatively related to indicators of disproportionate or overly expansive responses, including the response-to-prompt length ratio (len ratio; i1 $\rho = -.23^{***}$; i2 $\rho = -.22^{***}$) and prompt-containment features (contain prompt in response; i1 $\rho = -.19^{**}$; i2 $\rho = -.18^{**}$), suggesting that higher diagnostic ratings tended to occur when responses stayed targeted to the student’s query. Providing Guidance (i3) showed a different pattern that is related to a more concise and actionable help. Guidance endorsements were strongly negatively associated with verbosity (response character length $\rho = -.30^{***}$; sentence count $\rho = -.27^{***}$; word length $\rho = -.28^{***}$) while remaining positively related to embedding-based relevance (greedy matching; $\rho = .24^{***}$). This profile is consistent with raters rewarding guidance that is directly responsive and succinct rather than lengthy or diffuse.

In contrast, the correctness-focused criteria exhibited the weakest convergence with the automated metrics. Accuracy of Guidance (i4) showed near-zero correlations across most measures (with only small negative relationships with response length), and Correctness of next steps (i5) was primarily associated with surface length and diversity proxies rather than stronger positive semantic alignment signals. Overall, these findings indicate that standard reference-free features predominantly capture relevance and alignment and surface form (verbosity and lexical diversity), but provide incomplete coverage of pedagogically consequential constructs, particularly factual or conceptual correctness and the appropriateness of next-step suggestions. This limitation motivates the use of expert rubric scoring and psychometric modeling for rigorous evaluation of educational chatbot responses.

5. DISCUSSION

This study introduces a Bayesian explanatory Many-Facet Rasch Model (MFRM) as a psychometric evaluation layer for rubric-based assessment of LLM tutoring responses in authentic programming help interactions. Instead of interpreting rubric endorsements as direct measures of pedagogical effectiveness, the model disentangles latent response quality from rubric-criterion difficulty and rater severity, and incor-

porates interaction context through query-level covariates. The resulting facet-adjusted estimates provide a more diagnostic and context-sensitive characterization of tutoring performance than raw means or endorsement rates, clarifying when our LLM-based chatbot performs well and where it breaks down across pedagogically meaningful dimensions.

Our work builds on a small but growing set of Rasch-family applications to LLM and chatbot evaluation, which largely follow two patterns. One line of work treats a chatbot as an “examinee” and applies Rasch analysis to place chatbots and test items on a common proficiency scale, often using domain testing formats such as medical or STEM multiple-choice benchmarks [6]. While useful for benchmarking performance relative to item difficulty, this approach typically focuses on correctness in fixed-response tasks and does not directly address human rater-mediated judgments of open-ended tutoring quality. A second line of work uses many-facet Rasch models in rating systems, particularly when an LLM functions as an automated scorer or rater, with the primary goal of quantifying severity, fit, and bias relative to human raters [60, 23, 26]. Related work extends MFRM to broader human-evaluation pipelines for AI-generated conversations by jointly modeling rater effects and contextual facets [34]. Our contribution is distinct from both benchmark scaling and rater auditing alone. We operationalize MFRM for rubric-scored, open-ended tutoring behaviors under authentic help-seeking conditions and provide a principled procedure for determining which contextual extensions add predictive value. In our data, PSIS-LOO comparisons favored a parsimonious explanatory specification that adds input sufficiency to the core facets, while topic-augmented models increased effective complexity without improving predictive fit (Table 3). This positions the MFRM not only as a scoring model, but as a validity-oriented evaluation layer that supports model checking, uncertainty quantification, and context-conditional interpretation.

Addressing RQ1, criterion difficulty results indicate that the chatbot’s pedagogical strengths and weaknesses were not uniformly distributed across tutoring behaviors. Mistake diagnosis criteria, including Mistake Identification (i1) and Mistake Location (i2), were substantially more difficult to satisfy than correctness-oriented criteria, Accuracy of Guidance (i4) and Correctness of next steps (i5). Pedagogically, this pattern suggests that the system more reliably produces technically plausible guidance once it commits to a direction, but is less consistent in upstream tutoring behaviors that require diagnosing what the student did wrong and where the error originates. This asymmetry matters in programming help contexts because effective support typically depends on diagnosis before explanation [58]. Without correctly recognizing and localizing the mistake, fluent and confident guidance can be misaligned with the student’s actual issue. From an evaluation standpoint, these findings also clarify why global aggregates can be misleading. If criteria differ substantially in difficulty, collapsing across rubric dimensions can hide critical weaknesses. For example, high endorsement for correctness-oriented criteria can coexist with systematic fragility in diagnosis. The Rasch framing makes this visible by placing criteria on a common logit scale and treating difficulty differences as interpretable measurement parameters [30, 52].

The estimated rater severity parameters were small and symmetric around zero, indicating minimal systematic rater bias after accounting for response quality and criterion difficulty. This is encouraging for rubric-based evaluation because it suggests that conclusions about criterion ordering and response-quality ranking are not driven by one rater being consistently more lenient or severe. At the same time, Cohen’s κ was low-to-moderate across dimensions while exact-match agreement was comparatively high. This divergence is consistent with the κ paradox under endorsement imbalance, where high observed agreement can coexist with depressed chance-corrected agreement [15, 9]. In this setting, the MFRM approach is particularly useful because it does not assume perfect agreement. Instead, it uses both raters’ codes to estimate and adjust for severity on the same latent scale and propagates uncertainty into criterion-level and response-level estimates.

Addressing RQ2, input sufficiency effects were positive across criteria, with larger posterior mean shifts for the accuracy-oriented criteria (i4–i5). Conceptually, this direction is expected. Richer student context, such as code, error traces, and clear problem statements, should increase the probability that the chatbot can respond appropriately and that raters can evaluate the response favorably. Although uncertainty around these effects remained nontrivial, likely because only a small fraction of retained exchanges were labeled insufficient (about 5%), the PSIS-LOO comparison provided modest evidence that adding sufficiency improves predictive adequacy relative to a facets-only baseline (Table 3). This yields a practical implication for evaluation design. To estimate sufficiency effects more precisely, future datasets may require higher prevalence of insufficient queries or sampling strategies that deliberately balance sufficient and insufficient contexts. More broadly, these results support a context-conditional view of tutoring effectiveness. Observed quality reflects both system capability and what students provide, and a psychometric evaluation layer that conditions on query quality helps separate these components and makes evaluation conclusions more transportable across settings with different help-seeking behavior distributions.

Addressing RQ3, we explored whether question content was associated with criterion-level endorsements using multi-label topic indicators. Topic-conditional patterns suggested that some categories were associated with higher endorsement for diagnosis behaviors, while at least one category was associated with lower endorsement for providing guidance. However, topic-augmented specifications did not improve PSIS-LOO and increased effective complexity, so these findings are best interpreted as descriptive and hypothesis-generating rather than as confirmatory moderation effects (Table 3). Even so, they point to a practical consideration for computing education and LLM tutoring. The same system can appear effective for some help-seeking contexts and less effective for others, and reporting criterion-level outcomes helps make this heterogeneity visible without collapsing everything into a single global score. Future work with larger samples or targeted designs can more decisively test topic moderation and evaluate whether structured interfaces can shift students toward contexts where the tutor behaves more reliably.

The comparison to automated metrics is best interpreted as construct coverage evidence rather than a competition for a single “best” score. We observed only modest alignment between common reference-free metrics and rubric targets, and the correlation structure differed by rubric dimension. In particular, automated metrics captured limited signal for correctness-focused criteria, suggesting that surface form, overlap, and prompt-response relatedness measures do not reliably recover pedagogically consequential judgments about factual correctness and appropriateness of next steps. Taken together, these findings motivate the continued role of expert rubric-based evaluation in educational settings and clarify where automated evaluation may complement, but not replace, pedagogical scoring.

Overall, our results support a context-sensitive interpretation of educational LLM-based chatbot effectiveness. Pedagogical quality depends on what the system can do, what students provide, and which instructional behaviors are being evaluated. By combining expert rubric judgments with a Bayesian explanatory MFRM and explicitly validating model extensions with PSIS-LOO, the proposed approach offers a principled path toward uncertainty-aware, diagnostically meaningful evaluation of LLM-based tutors in authentic learning settings.

6. CONCLUSION AND LIMITATIONS

Several limitations should be considered when interpreting the findings. First, the study examined a single course context, a single programming task (the “Big Three”), and a single tutoring system configuration; generalization to other topics, courses, or tutor designs requires additional evidence. Second, the analysis focused on single-turn exchanges. Multi-turn tutoring introduces additional facets, such as dialogue history, repair trajectories, and evolving student understanding, which may change both rubric constructs and model structure. Future evaluations may benefit from rubric refinements or sampling strategies that ensure adequate variability across criteria. Finally, input sufficiency was relatively rare within the retained sample (5%), which limits precision for estimating sufficiency effects and suggests the need for intentional sampling or larger datasets to more sharply quantify context effects.

This work suggests several directions for improving both evaluation practice and tutoring design. For evaluation, future studies can extend the framework to (i) ordinal scoring (rather than binary endorsement), (ii) additional facets such as rubric-dimension severity differences by rater, and (iii) multilevel structures that account for student-level or prompt-level clustering. For tutor design, the strongest need indicated by criterion difficulty is improved mistake diagnosis: systems may benefit from structured prompting that elicits minimal reproducible examples, explicit error traces, or targeted clarifying questions. Finally, because effectiveness varies by question type, future work should explore adaptive interfaces that guide students toward higher information queries and monitor when the tutor is operating outside its reliable regimes.

7. REFERENCES

- [1] B. Abeysinghe and R. Circi. The challenges of evaluating llm applications: An analysis of automated,

- human, and llm-based approaches. *arXiv preprint arXiv:2406.03339*, 2024.
- [2] Y. Albadarin, M. Saqr, N. Pope, and M. Tukiainen. A systematic literature review of empirical research on chatgpt in education. *Discover Education*, 3(1):60, 2024.
 - [3] B. Alsafari, E. Atwell, A. Walker, and M. Callaghan. Towards effective teaching assistants: From intent-based chatbots to llm-powered teaching assistants. *Natural Language Processing Journal*, 8:100101, 2024.
 - [4] Z. Ashktorab, M. Desmond, Q. Pan, J. M. Johnson, M. Santillan Cooper, E. M. Daly, R. Nair, T. Pedapati, S. Achintalwar, and W. Geyer. Aligning human and llm judgments: Insights from evalassist on task-specific evaluations and ai-assisted assessment strategy preferences, 2024.
 - [5] A. Birillo, E. Artser, A. Potriasaeva, I. Vlasov, K. Dziales, Y. Golubev, I. Gerasimov, H. Keuning, and T. Bryksin. One step at a time: Combining LLMs and static analysis to generate next-step hints for programming tasks. In *Proceedings of the 24th Koli Calling International Conference on Computing Education Research*, pages 9:1–9:12. ACM, 2024.
 - [6] Y. Chang et al. Evaluating chatbots in psychiatry: Rasch-based insights into clinical knowledge and reasoning. *PLOS ONE*, 2025.
 - [7] Y. Chang, S.-S. Huang, W.-Y. Hsu, and Y.-C. Liu. Evaluating chatbots in psychiatry: Rasch-based insights into clinical knowledge and reasoning. *PLOS ONE*, 2025.
 - [8] G. Cheng, W. Wong, L. Luo, and M. Yu. Integrating a scaffolding-based, llm-driven chatbot into programming education: A university case study. In *2025 International Symposium on Educational Technology (ISET)*, pages 1–5. IEEE, 2025.
 - [9] D. V. Cicchetti and A. R. Feinstein. High agreement but low kappa: II. resolving the paradoxes. *Journal of Clinical Epidemiology*, 43(6):551–558, 1990.
 - [10] R. Coppola and L. Ardito. Quality assessment methods for textual conversational interfaces: a multivocal literature review. *Information*, 12(11):437, 2021.
 - [11] N. Daheim, J. Macina, M. Kapur, I. Gurevych, and M. Sachan. Stepwise verification and remediation of student reasoning errors with large language model tutors. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8386–8411, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics.
 - [12] S. K. Dam, C. S. Hong, Y. Qiao, and C. Zhang. A complete survey on llm-based ai chatbots. *arXiv preprint arXiv:2406.16937*, 2024.
 - [13] T. Debets, S. K. Banihashem, D. Joosten-Ten Brinke, T. E. Vos, G. M. de Buy Wenniger, and G. Camp. Chatbots in education: A systematic review of objectives, underlying technology and theory, evaluation criteria, and impacts. *Computers & Education*, page 105323, 2025.
 - [14] L. Favero, J. A. Pérez-Ortiz, T. Käser, and N. Oliver. Enhancing critical thinking in education by means of a socratic chatbot. In *International workshop on AI in education and educational research*, pages 17–32. Springer, 2024.
 - [15] A. R. Feinstein and D. V. Cicchetti. High agreement but low kappa: I. the problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549, 1990.
 - [16] Z. Gao and J. Jiang. Evaluating human-AI hybrid conversational systems with chatbot message suggestions. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, 2021.
 - [17] S. Goren, O. Kalinsky, T. Stav, Y. Rapoport, Y. Fairstein, R. Yazdi, N. Cohen, A. Libov, and G. Kushilevitz. Chai-tea: A benchmark for evaluating autocompletion of interactions with llm-based chatbots. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 18–32, 2025.
 - [18] S. Groothuijsen, A. van den Beemt, J. C. Remmers, and L. W. van Meeuwen. Ai chatbots in programming education: Students’ use in a scientific computing course and consequences for learning. *Computers and Education: Artificial Intelligence*, 7:100290, 2024.
 - [19] S. Groothuijsen, A. van den Beemt, J. C. Remmers, and L. W. van Meeuwen. Ai chatbots in programming education: Students’ use in a scientific computing course and consequences for learning. *Computers and Education: Artificial Intelligence*, 7:100290, 2024.
 - [20] M. Hmoud, H. Swaity, E. Anjass, and E. M. Aguaded-Ramírez. Rubric development and validation for assessing tasks’ solving via ai chatbots. *Electronic Journal of e-Learning*, 22(6):1–17, 2024.
 - [21] S. Hobert. How are you, chatbot? evaluating chatbots in educational settings—results of a literature review. *DELFI 2019*, pages 259–270, 2019.
 - [22] J. Jiang and N. Ahuja. Response quality in human-chatbot collaborative systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1545–1548, 2020.
 - [23] H. Jiao, D. Song, and W.-C. Lee. Comparing human and AI rater effects using the many-facet rasch model, 2025.
 - [24] I. Jurenka, M. Kunesch, K. R. McKee, D. Gillick, S. Zhu, S. Wiltberger, S. M. Phal, K. Hermann, D. Kasenberg, A. Bhoopchand, et al. Towards responsible development of generative ai for education: An evaluation-driven approach. *arXiv preprint arXiv:2407.12687*, 2024.
 - [25] M. Kazemitabaar, R. Ye, X. Wang, A. Z. Henley, P. Denny, M. Craig, and T. Grossman. Codeaid: Evaluating a classroom deployment of an LLM-based programming assistant that balances student and educator needs. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI ’24)*. ACM, 2024.
 - [26] K. Kim, J. H. Lee, and D. Shin. The potential advantages of using an llm-based chatbot for automated writing evaluation for english teaching practitioners. *Language Learning & Technology*, 29(1):1–12, 2025.

- [27] C. Koyuturk, E. Theophilou, S. Patania, G. Donabauer, A. Martinenghi, C. Antico, A. Telari, A. Testa, S. Buršić, F. Garzotto, et al. Understanding learner-llm chatbot interactions and the impact of prompting guidelines. In *International Conference on Artificial Intelligence in Education*, pages 364–377. Springer, 2025.
- [28] V. Kumar, P. Srivastava, A. Dwivedi, I. Budhiraja, D. Ghosh, V. Goyal, and R. Arora. Large-language-models (llm)-based ai chatbots: Architecture, in-depth analysis and their performance evaluation. In *International Conference on Recent Trends in Image Processing and Pattern Recognition*, pages 237–249. Springer, 2023.
- [29] M. Liffiton, B. Sheese, J. Savelka, and P. Denny. Codehelp: Using large language models with guardrails for on-demand programming assistance. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*. ACM, 2023.
- [30] J. M. Linacre. *Many-Facet Rasch Measurement*. MESA Press, Chicago, IL, 1989.
- [31] K. K. Maurya, K. V. A. Srivatsa, K. Petukhova, and E. Kochmar. Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2025)*, 2025.
- [32] H. D. Nguyen, T.-V. Tran, X.-T. Pham, A. T. Huynh, V. T. Pham, and D. Nguyen. Design intelligent educational chatbot for information retrieval based on integrated knowledge bases. *IAENG International Journal of Computer Science*, 49(2), 2022.
- [33] S. A. Nikou, A. Guliya, S. Van Verma, and M. Chang. A generative artificial intelligence empowered chatbot: system usability and student teachers’ experience. In *International Conference on Intelligent Tutoring Systems*, pages 330–340. Springer, 2024.
- [34] C. Niu, K. Bradley, B. Ma, B. Waltman, L. Cossette, and R. Jin. Bias and reliability in AI safety assessment: Multi-facet rasch analysis of human moderators. In *Proceedings of AI in Measurement and Education Conference (AIMEcon)*, pages 393–397, 2025.
- [35] W. S. Nsaif, H. M. Salih, H. H. Saleh, and B. T. Al-Nuaimi. Chatbot development: Framework, platform, and assessment metrics. *The Eurasia Proceedings of Science Technology Engineering and Mathematics*, 27:50–62, 2024.
- [36] C. W. Okonkwo and A. Ade-Ibijola. Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence*, 2:100033, 2021.
- [37] H. Pauzi, S. Dodman, and M. Mavrikis. Automating pedagogical evaluation of llm-based conversational agents in educational settings. In *Proceedings of the 2nd Workshop on Automatic Evaluation of Learning and Assessment Content (EvalLAC 2025), co-located with AIED 2025*, volume 4006 of *CEUR Workshop Proceedings*, 2025.
- [38] H. Pauzi, S. Dodman, and M. Mavrikis. Automating pedagogical evaluation of llm-based conversational agents in educational settings. In *Proceedings of the 2nd Workshop on Automatic Evaluation of Learning and Assessment Content (EvalLAC 2025), co-located with AIED 2025*, volume 4006 of *CEUR Workshop Proceedings*, 2025.
- [39] D. Peras. Chatbot evaluation metrics. *Economic and Social Development: Book of Proceedings*, pages 89–97, 2018.
- [40] K. Petukhova and E. Kochmar. Intent matters: Enhancing AI tutoring with fine-grained pedagogical intent annotation. In E. Kochmar, B. Alhafni, M. Bexte, J. Burstein, A. Horbach, R. Laarmann-Quante, A. Tack, V. Yaneva, and Z. Yuan, editors, *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 860–872, Vienna, Austria, July 2025. Association for Computational Linguistics.
- [41] A. Piscitelli, G. Costagliola, M. De Rosa, and V. Fuccella. Influence of large language models on programming assignments—a user study. In *Proceedings of the 2024 the 16th International Conference on Education Technology and Computers*, pages 33–38, 2024.
- [42] R. Primi, P. J. Silvia, E. Jauk, and M. Benedek. Applying many-facet rasch modeling in the assessment of creativity. *Psychology of Aesthetics, Creativity, and the Arts*, 13:176–186, 2019.
- [43] M. M. Rahman and Y. Watanobe. Chatgpt for education and research: Opportunities, threats, and strategies. *Applied sciences*, 13(9):5783, 2023.
- [44] S. Rohini and T. Sethukarasi. Chatbots-based systematic review for student learning and engagement with llms. In *2025 3rd International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)*, pages 1978–1983. IEEE, 2025.
- [45] L. M. Sánchez-Ruiz, S. Moll-López, A. Nuñez-Pérez, J. A. Morano-Fernández, and E. Vega-Fleitas. Chatgpt challenges blended learning methodologies in engineering education: A case study in mathematics. *Applied Sciences*, 13(10):6039, 2023.
- [46] J. Sedoc, D. Ippolito, A. Kirubarajan, J. Thirani, L. Ungar, and C. Callison-Burch. ChatEval: A tool for chatbot evaluation. In W. Ammar, A. Louis, and N. Mostafazadeh, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 60–65, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [47] B. Sheese, M. Liffiton, J. Savelka, and P. Denny. Patterns of student help-seeking when using a large language model-powered programming assistant. In *Proceedings of the 26th Australasian Computing Education Conference (ACE ’24)*, pages 49–57. ACM, 2024.
- [48] D. Song, E. Y. Oh, and M. Rice. Interacting with a conversational agent system for educational purposes in online courses. In *2017 10th international conference on human system interactions (HSI)*, pages 78–82. IEEE, 2017.

- [49] S. Stefanova and Y. Kalmukov. Analysis of the possibilities of using llm chatbots for solving course and exam tasks. *International Journal of Advanced Computer Science & Applications*, 16(8), 2025.
- [50] A. Tack, E. Kochmar, Z. Yuan, S. Bibauw, and C. Piech. The BEA 2023 shared task on generating AI teacher responses in educational dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 785–795, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [51] A. Tack and C. Piech. The AI teacher test: Measuring the pedagogical ability of blender and GPT-3 in educational dialogues. 2022.
- [52] M. Tavakol, G. Pinner, and R. Dennick. Using the many-facet rasch model to analyse and evaluate clinical assessment. *International Journal of Medical Education*, 10:126–131, 2019.
- [53] R. Tsang and S. Y. WOOD. Help or hype? exploring llm-based chatbots in self-regulated learning. In *2025 ASEE Annual Conference & Exposition*, 2025.
- [54] M. Uto, J. Tsuruta, K. Araki, and M. Ueno. Item response theory model highlighting rating scale of a rubric and rater–rubric interaction in objective structured clinical examination. *PLOS ONE*, 19(9):e0309887, 2024.
- [55] M. Verleger and J. Pembridge. A pilot study integrating an ai-driven chatbot in an introductory programming course. In *2018 IEEE frontiers in education conference (FIE)*, pages 1–4. IEEE, 2018.
- [56] M. Verleger and J. J. Pembridge. A pilot study integrating an AI-driven chatbot in an introductory programming course. In *Proceedings of the 2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–4, San Jose, CA, USA, 2018. IEEE.
- [57] P. Wang, K. Yin, M. Zhang, Y. Zheng, T. Zhang, Y. Kang, and X. Feng. The effect of incorporating large language models into the teaching on critical thinking disposition: An “ai+ constructivism learning theory” attempt. *Education and Information Technologies*, pages 1–23, 2025.
- [58] T.-T. Wu, H.-Y. Lee, P.-H. Chen, W.-S. Wang, and Y.-M. Huang. Integrating computational thinking diagnostic mechanism and reflective learning: An innovative approach to enhance learning outcomes in introductory programming. *Journal of Computer Assisted Learning*, 41(5):e70121, 2025.
- [59] T. Yamashita. An application of many-facet rasch measurement to evaluate automated essay scoring: A case of chatgpt-4.0. *Research Methods in Applied Linguistics*, 3(3):100133, 2024.
- [60] T. Yamashita. An application of many-facet rasch measurement to evaluate automated essay scoring: A case of ChatGPT-4.0. *Assessing Writing*, 2024. Article in press/online first on ScienceDirect.
- [61] D. Yigci, M. Eryilmaz, A. K. Yetisen, S. Tasoglu, and A. Ozcan. Large language model-based chatbots in higher education. *Advanced Intelligent Systems*, 7(3):2400429, 2025.
- [62] Y. Zhang, E. L. Ouh, A. Ho, S. L. Lo, K. W. Tan, and F. Lin. Prompttutor: Effects of an llm-based chatbot on learning outcomes and motivation in flipped classrooms. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 445–451, 2025.

APPENDIX

A. DETAILED EVALUATION RUBRIC

i1: Mistake Identification

Guiding Question: *Did the response correctly identify or recognize a mistake in the student's original response?*

This category assesses whether the chatbot response explicitly or implicitly acknowledges **all** errors in the student's answer. Recognition can be **direct** (e.g., pointing out what is incorrect) or **indirect** (e.g., offering a correction or prompting the student to revisit a specific part). When multiple errors are present, it is expected to recognize **each** of them.

- **Yes:** The response correctly identifies all mistakes present in the context, evaluating both the issue and the code.
- **Partially:** The response identifies some mistakes but misses others, or identifies them incompletely.
- **No:** The response fails to identify any mistakes, or incorrectly labels correct elements as mistakes.

Rater Tip: Focus on whether the response shows awareness of the student's mistake, not on how thorough the explanation is or whether the correction is accurate.

i2: Mistake Location

Guiding Question: *Did the response accurately point to a genuine mistake and its location?*

This category assesses whether the chatbot correctly indicates where each identified mistake occurs. For **each** error it recognizes, it should clearly and correctly indicate the relevant portion of the response. Line numbers are not required if the referenced location is unambiguous.

- **Yes:** The response accurately pinpoints the location of all identified mistakes.
- **Partially:** The response locates some, but not all, mistakes. If the response does not explicitly cite a line number but the intended location is clear (e.g., "in the copy assignment operator"), count this as partial.
- **No:** The response fails to locate the mistakes or points to incorrect or nonexistent locations.

Rater Tip: For each mistake identified, check whether the chatbot points to the corresponding location with enough specificity for a student to find it. If the response misses mistakes or misidentifies them, this category should generally be rated **Partially** or **No**.

i3: Providing Guidance

Guiding Question: *Does the chatbot offer guidance that is relevant and responsive to the student's issue?*

This category assesses whether the chatbot offers guidance, such as explanations, elaborations, tips, hints, and examples, that is both relevant and responsive to the issues provided and identified in the context. Fully appropriate guid-

ance clearly addresses the issue without introducing unrelated information, omitting the student's request, or becoming overly generic, tailored to the students' needs.

- **Yes:** The response directly addresses the student's question(s) in context and provides guidance that helps remediate the identified mistake(s).
- **Partially:** The response provides incomplete guidance or includes distracting, tangential information that adds unnecessary cognitive burden.
- **No:** The response provides no guidance or guidance that is largely irrelevant to the student's issue.

Rater Tip: If the response misses part of the question or provides an irrelevant answer, rate **Partially** or **No** depending on the extent of the mismatch.

i4: Accuracy of Guidance

Guiding Question: *Is the guidance provided by the chatbot factually and conceptually correct?*

This category evaluates whether the chatbot's guidance is free from factual inaccuracies and conceptual misunderstandings. A fully appropriate response demonstrates correct understanding of the concepts, procedures, or principles and presents them without errors or misleading statements.

- **Yes:** Guidance is factually and conceptually correct.
- **Partially:** Guidance is mostly correct but contains minor inaccuracies, oversimplifications, or omissions that may reduce correctness or lead to misunderstandings.
- **No:** The guidance contains major errors, is misleading, or reflects conceptual misunderstandings.

Rater Tip: Focus on correctness of claims and reasoning. Do not penalize for brevity if the guidance is accurate.

i5: Correctness of Next Steps

Guiding Question: *Are the suggested next steps or corrections accurate and appropriate to the student's problem?*

This category evaluates whether the chatbot's proposed corrections or next steps are factually accurate, conceptually sound, and well matched to the student's specific error or question. Suggestions should be correct in principle and appropriate to the given context so that following them would resolve the problem without introducing new errors.

- **Yes:** The suggested corrections or next steps are accurate and directly appropriate to the student's problem.
- **Partially:** Suggestions are generally on the right track but are incomplete, partially misapplied, or would require important missing steps to work reliably.
- **No:** Suggestions or next steps are inaccurate, infeasible, unsafe, or not aligned with the student's problem.

Rater Tip: Evaluate whether the proposed actions would solve the stated problem in the given context.

B. EXAMPLE LLM-CHATBOT RESPONSE PATTERN AND RUBRIC TRIGGERS

Table 9: Common chatbot response pattern observed in the dataset and the rubric criteria it typically triggers.

Typical response move	Primary rubric criteria triggered
Interpret the error signal (e.g., compiler/runtime message)	i1 Mistake Identification, i4 Accuracy of Guidance
Locate the source of the error in the student code	i2 Mistake Location
Explain why the error occurs and connect it to the student’s implementation	i3 Providing Guidance; i4 Accuracy of Guidance
Provide actionable remediation steps (what to change and how)	i5 Correctness of Next Steps
Address additional pitfalls or edge cases (e.g., self-assignment)	i1–i4 (as applicable); i5 if a fix is proposed
Close with general best-practice reminders (e.g., Rule of Three)	Typically supports i3; may be neutral if not tied to the specific code

Annotated example (representative response):
 In the dataset, the chatbot typically provides a consolidated explanation of the student’s error and offers remediation steps. Responses often follow a recurring structure, which allows the rubric to be applied consistently. Below is one representative example, with each paragraph annotated by the rubric criteria it primarily triggers.

Chatbot response (example):

[i1: Mistake Identification]
 The error message you’re seeing from Valgrind indicates that there’s an “Invalid write of size 8,” which means that your program is trying to write to a memory location that it shouldn’t. This is often due to writing to memory that has already been freed or is out of bounds of the allocated array.

[i2: Mistake Location; i3: Providing Guidance; i4: Accuracy of Guidance]
 In your copy constructor and copy assignment operator, you’re allocating a new array and then copying the elements from the `other` array. However, after repointing `arr` to the new array, you immediately delete the array that `arr` points to. This is a mistake because you’re deleting the same memory you’ve just allocated and copied data into. As a result, when you try to access or write to this memory later, you get the invalid write error because the memory has been freed.

[i5: Correctness of Suggestions; Clarity of Next Steps]
 To fix this, you should only delete the old array after you’ve done everything else, including repointing the `arr` pointer to the new array. Also, make sure that you’re not leaking the previous array that `arr` pointed to before you assign it to the new array. You need to save the old pointer, assign the new array to `arr`, and then delete the old array.

Figure 5. Annotated example illustrating a common chatbot response structure and the rubric criteria it triggers.