COGRAISE: FROM COGNITIVE-PROCESS DIAGNOSIS TO PROGRESSION-AWARE LLM TUTORING

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective tutoring requires diagnosing how learners process knowledge and determining how instruction should support their cognitive progression. Yet existing LLM tutors emphasize observable pedagogical behaviors, with limited explicit modeling of learner cognition. To address this gap, we introduce CogRaise, a 4B cognition-aware tutor that operationalizes fine-grained cognitive processes as explicit training and optimization targets. First, we construct a Theory-Driven Dataset grounded in the 19 cognitive processes of Anderson's revised Bloom taxonomy, comprising multi-turn diagnosis-intervention-response trajectories for SFT-based initialization and reward learning. Second, a multi-head reward model separately scores cognition-related tutoring criteria and general response quality. Third, GRPO incorporates a cognitive-advancement reward conditioned on the learner's current cognitive level to encourage interventions supporting cognitive advancement. Finally, we introduce CogEdu-Bench, an 849-instance multi-subject benchmark evaluating fine-grained cognitive diagnosis, pedagogical decisions, and tutoring response quality. CogRaise improves fine-grained diagnosis QWK by 37.6% relative to its SFT initialization and enhances scaffolding on MathTutorBench, while general reasoning performance remains within 0.3 percentage points of that initialization on each of GSM8K, MATH500, and MMLU-Pro. These results suggest that established educational theory can be operationalized into structured supervision and reward signals to align LLM tutors with cognitive progression.

1 INTRODUCTION

Learning is fundamentally a process of transforming how knowledge is processed, rather than merely accumulating more information (Council, 2000; Mayer, 2002). Teaching, correspondingly, requires diagnosing how a learner is currently processing knowledge and intervening so that this processing can progress (Wood et al., 1976). For instance, a learner who only recalls that photosynthesis converts light energy into chemical energy needs pedagogical support to explain why light is necessary. Once the learner can explain this mechanism, pedagogical support should shift toward applying this understanding to predict effects of changing light conditions or analyzing how light interacts with other factors. Therefore, effective tutoring requires dynamically adapting pedagogical strategies to changes in how learners process knowledge (Council, 2000).

Recent LLM-based tutors have made substantial progress beyond direct answer provision by shifting toward guided pedagogical strategies (Puech et al., 2025; Aleven et al., 2004), such as asking guiding questions (Liu et al., 2024) and providing targeted hints. However, these approaches primarily regulate how a tutor responds rather than explicitly modeling how the learner is currently processing the knowledge at hand. Consequently, a tutor may become increasingly Socratic while still lacking a principled mechanism for systematically guiding the development of that processing.

The cognitive-process dimension of Anderson's revision of Bloom's Taxonomy of Educational Objectives (hereafter, the Bloom Taxonomy) provides a principled framework for addressing this limitation (Anderson & Krathwohl, 2001). It organizes cognition into six broad categories representing cognitive objectives, with 19 fine-grained processes grouped under these categories to specify the cognitive operations associated with each objective (Appendix B.1). While the six categories characterize the progressive deepening of cognitive target, the 19 processes offer finer-grained operation

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

for cognitive progression. We use their descriptions to inform the generation of multi-turn tutoring dialogues reflecting corresponding cognitive levels, and arrange these processes into an ordered sequence to guide progression.This sequence is our operationalization for tutoring, rather than an order directly specified by the taxonomy (Sec. 3.1). However, the taxonomy alone does not supply the mechanisms needed to realize cognition-aware tutoring. Three technical challenges remain:

1. Obtaining training evidence for cognition-aware tutoring is challenging because the cognitive processes underlying learner utterances are latent rather than directly observable (Liu et al., 2022). Existing interaction records and simulated tutoring data remain insufficient as direct supervision for identifying these processes and determining which subsequent process pedagogical strategies should target (Macina et al., 2023; Pal Chowdhury et al., 2024; Liu et al., 2024). The challenge is therefore to externalize latent cognitive processes into learnable trajectories of diagnosis, intervention, and response.

2. 0btaining such training signals alone does not ensure that a tutor will promote appropriate cognitive progression. Existing tutor-alignment methods primarily optimize observable pedagogical qualities (Yao et al., 2024; Puech et al., 2025; Singh et al., 2026) or problem-solving success (Ruan et al., 2024). Such objectives assess tutor responses against these observable criteria and use the resulting rewards to guide policy optimization. Yet, a response can be helpful, fluent, Socratic, and while still mischaracterizing how learners process knowledge or supporting an inappropriate subsequent cognitive process.

3. Even a well-aligned tutor cannot be fully validated if existing benchmarks do not evaluate the capability. Current tutor benchmarks primarily assess observable teaching behaviors, such as mistake identification, guidance, actionability, and tutor tone (Maurya et al., 2025), or evaluate the overall quality of tutoring interactions on domain-specific learning tasks (Macina et al., 2025). Cognition-oriented evaluations, meanwhile, often use cognitive taxonomies to assess a model's understanding of instructional content (Jin et al., 2026), but ignore Whether tutors gain the cognition-aware tutoring capability.

To address these challenges, we introduce CogRaise, a cognition-aware tutor trained to connect cognitive diagnosis with pedagogical intervention using the fine-grained processes of the Bloom taxonomy. Given a learner's utterance, it diagnoses the current cognitive level, selects an appropriate target process, and generates a tutoring response to support cognitive advancement.

Our key contributions are as follows:

1. **Theory operationalization and cognition-aware training data.** We develop a data synthesis framework that uses LLMs to synthesize tutoring dialogues under cognitive-process constraints derived from the Bloom taxonomy. These dialogues are structured as explicit diagnosis-intervention-response trajectories, with our tutoring-specific ordering of the 19 fine-grained processes guiding cognitive progression. These trajectories provide demonstration data for policy initialization and theory-grounded scoring and preference signals for reward learning.

2. **Cognition-aware reward modeling and policy alignment.** We develop a multi-head reward model that converts cognitive diagnosis and intervention criteria into decomposed reward signals, retains general quality criteria, and uses controlled aggregation to limit compensation across key tutoring dimensions. We complement these assessments with an explicit cognitive-advancement reward, thereby encouraging tutors to precisely and steadily promote the advancement of learners' cognitive levels. The combined reward signals guide policy optimization using GRPO.

3. **CogEdu-Bench for cognition-centered tutor evaluation.** We introduce CogEdu-Bench, a multi-subject benchmark separately evaluating fine-grained cognitive diagnosis, pedagogical decision-making, and response quality in supporting cognitive progression. Our 4B CogRaise achieves strong fine-grained cognitive diagnosis while remaining competitive on broader tutoring capabilities and largely preserving general reasoning performance.

2 Related Work

2.1 LLM-based Tutoring and Cognitive Process Modeling

Interaction-Based Learner Modeling. Existing tutoring research draws on observable interaction signals, including solution attempts and error patterns. For example, MathDial grounds feedback in students' problem-solving attempts (Macina et al., 2023), while SocraticLM constructs multi-turn dialogues around simulated learners' responses (Liu et al., 2024). Related learner-modeling methods infer latent states from interaction histories. Knowledge-tracing models estimate evolving knowledge mastery (Abdelrahman et al., 2023), whereas cognitive-diagnosis models infer concept-level proficiency (Wang et al., 2020). Yet these representations primarily capture learners' observable behavior and knowledge proficiency, rather than how they cognitively process the knowledge at hand.

Theory-Grounded Cognitive Process Modeling. Modeling this latent dimension requires a theory-grounded representation of cognitive processes. BloomNet classifies course learning outcomes according to broad categories (Waheed et al., 2021), which have also been used to guide and evaluate educational question generation (Elkins et al., 2024; Scaria et al., 2024). PedagogyBench further evaluates multimodal instructional-video understanding across taxonomy-derived cognitive levels (Jin et al., 2026). Despite these advances, existing studies primarily apply the six broad categories to static, content-level tasks rather than dynamic tutoring interactions. The use of 19 levels to model learners' ongoing cognitive processing during tutoring remains underexplored.

2.2 Reward-Based Alignment of LLM Tutors

Pedagogical Alignment of LLM Tutors. Effective tutoring requires selecting pedagogical strategies appropriate to learners' cognitive levels (Sonkar et al., 2024). Existing approaches rely mainly on supervised fine-tuning (SFT) with pedagogically structured demonstrations (Dinucu-Jianu et al., 2025), preference learning over alternative tutor responses (Shani et al., 2024), and reinforcement learning (RL) with teaching-specific feedback (Ruan et al., 2024; Muslimani et al., 2023). SFT instills scaffolding and answer withholding (Puech et al., 2025), whereas preference learning and RL optimize pedagogical quality and guided problem solving (Dinucu-Jianu et al., 2025). Recent methods combine turn-level pedagogical quality with trajectory-level effectiveness in multi-turn interactions (Shani et al., 2024).

Reward Modeling for Tutoring Alignment. Conventional reward models compress heterogeneous preferences into a scalar score, whereas multi-attribute and multi-objective models represent distinct response attributes, through separate scores (Wang et al., 2024b;a;d). This decomposition supports interpretability and control over objective trade-offs. In tutoring, online pedagogical RL combines post-dialogue solve rate with LLM-judged helpfulness and answer-leakage control (Ruan et al., 2024), while multi-horizon preference optimization integrates turn-level clarity, conciseness, and non-revealing Socratic scaffolding with trajectory-level student success and dialogue efficiency (Shi et al., 2026). However, these decompositions primarily distinguish response attributes and tutoring outcomes. Less explored is how separate reward criteria can assess whether a tutor accurately diagnoses the learner's cognitive level, supports the target cognitive process, and follows the intended cognitive progression.

Reinforcement Learning for Cognitive Advancement. Whereas reward models assess response quality (Wang et al., 2024d), reinforcement-learning objectives determine how these evaluations shape policy behavior (Ouyang et al., 2022). Existing pedagogical RL objectives optimize student success and tutoring effectiveness (Dinucu-Jianu et al., 2025), yet offer limited guidance for selecting pedagogical strategies appropriate to the learner's diagnosed cognitive levels. Cognition-aware optimization therefore requires a cognitive level-conditioned advancement signal that rewards responses for adopting strategies directed toward an appropriate subsequent cognitive process.

2.3 Evaluating Educational and Tutoring Capabilities

Educational Task Evaluation. Educational benchmarks commonly assess subject knowledge (Zhong et al., 2024; Huang et al., 2023), mathematical problem solving (Cobbe et al., 2021b; Hendrycks et al., 2021), and exam-based reasoning (Zhong et al., 2024). Although these capabilities are foundational to educational applications, evaluating them alone cannot establish whether a model can provide pedagogically appropriate support.

Interactive Tutoring Evaluation. Recent benchmarks therefore shift from task performance to pedagogical support in interaction. MathTutorBench evaluates open-ended tutoring quality (Macina et al., 2025); MRBench assesses mistake identification and feedback actionability; and SafeTutors evaluates pedagogical effectiveness and safety in multi-turn interactions (Hazra et al., 2026). However, they do not test whether a tutor can use possesses cognition-aware tutoring capability.

3 Method

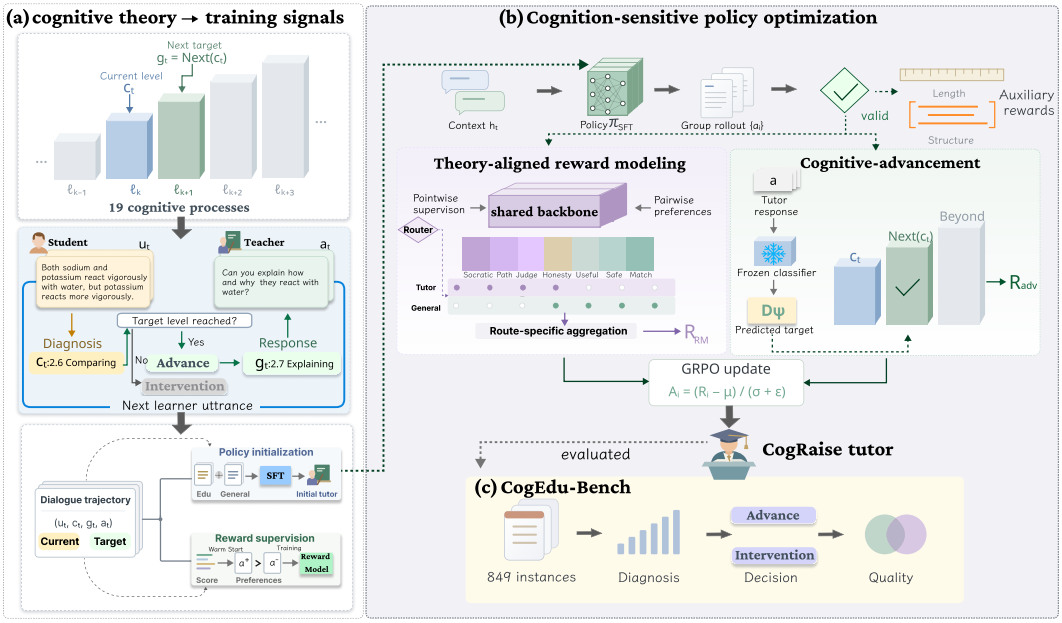


Figure 1: Overview of CogRaise: (a) theory-guided training signals, (b) cognition-sensitive policy optimization, and (c) CogEdu-Bench evaluation of diagnosis, decision, and tutoring quality.

As shown in Figure 1, CogRaise formalizes cognitive progression (Sec. 3.1) and constructs tutoring trajectories for supervised initialization (Sec. 3.2). A gated multi-head reward model (Sec. 3.3) complements a cognitive-advancement reward for policy optimization (Sec. 3.4.2). CogEdu-Bench evaluates cognitive diagnosis, pedagogical decision making, and tutoring quality (Sec. 3.5).

3.1 Cognition-Aware Tutoring: Problem Formulation

Cognitive progression modeling. We use the 19 cognitive processes in the Bloom taxonomy to represent learners' cognitive levels. Guided by Meaningful Learning (Mayer, 2002) and Pedagogical Scaffolding (Wood et al., 1976), we define a fixed progression order across these levels: $\mathcal{C} = (\ell_1, \ldots, \ell_{19}), \quad \ell_1 \prec \ell_2 \prec \cdots \prec \ell_{19}$.Here, each ℓ_k corresponds to one cognitive process, and $\prec$ denotes the progression order adopted in our framework. This ordering provides a consistent basis for selecting the next pedagogical target in Sec. 3.2 and computing the cognitive advancement reward in Sec. 3.4. It is a modeling assumption for training, rather than a strict developmental sequence prescribed by the taxonomy, and adjacent levels are not assumed to be equally spaced.

Problem formulation. Given a tutoring problem q, a tutoring episode is represented as a multi-turn trajectory:$\tau = \big(q, (u_t, c_t, g_t, a_t)_{t=1}^{T}\big)$, where u_t and a_t denote the learner utterance and tutor

response at turn t. The label $c_t \in \mathcal{C}$ represents the learner's current cognitive level demonstrated in the dialogue, while $g_t \in \mathcal{C}$ denotes the pedagogical target for that turn. The target g_t specifies the level that the tutor aims to address; it does not represent a level already reached by the learner. Let $h_t = (q, a_1, \ldots, a_{t-1}, u_t)$ denote the dialogue context available before the tutor produces its response. The tutor policy generates $a_t \sim \pi_\theta(\cdot \mid h_t)$. Neither c_t nor g_t is provided to the policy as a separate structured input; they are used to construct training and reward signals. Our objective is to generate responses that provide high-quality support and address an appropriate cognitive target relative to the learner's current level.

3.2 Theory-Driven Data Construction and Policy Initialization

Cognitive trajectory construction. Following the progression order in Sec. 3.1, we define $\mathrm{Next}(\ell_k) = \ell_{k+1}$ for $k < 19$ and $\mathrm{Next}(\ell_{19}) = \ell_{19}$. At turn t, we update the learner's current cognitive level based on u_t: if the learner demonstrates the previous pedagogical target g_{t-1}, we set $c_t = g_{t-1}$ and select *Advance*; otherwise, we retain $c_t = c_{t-1}$ and select *Intervention* to provide further support. The pedagogical target is then set to $g_t = \mathrm{Next}(c_t)$.

Theory-driven policy initialization. For each tutoring turn, the pair (c_t, g_t) is serialized as a level annotation together with a tutor response that provides support appropriate to both the learner's current cognitive level and the pedagogical target. The resulting trajectories form the target-task corpus $\mathcal{D}_{\mathrm{tar}}$. To retain general instruction-following ability, we combine it with a general corpus $\mathcal{D}_{\mathrm{gen}}$, yielding a mixed SFT corpus $\mathcal{D} = \mathcal{D}_{\mathrm{tar}} \cup \mathcal{D}_{\mathrm{gen}}$ containing 6,196 dialogue instances. Illustrative examples are provided in Appendix B.3. We initialize the tutor policy π_{SFT} for subsequent reward-guided optimization by minimizing the following supervised fine-tuning objective:

$$\mathcal{L}_{\mathrm{SFT}} = -\mathbb{E}_{(h_t, a_t)\sim\mathcal{D}} \sum_{j=1}^{|a_t|} \log \pi_\theta(a_{t,j} \mid h_t, a_{t,<j}). \tag{1}$$

3.3 Theory-Aligned Gated Multi-Head Reward Modeling

Structured reward decomposition. Formative Assessment (Black & Wiliam, 2009), Pedagogical Scaffolding (Wood et al., 1976), and Socratic Questioning (Paul & Elder, 2008) inform our assessment of diagnosis, intervention, and response quality, respectively. Following Meaningful Learning (Mayer, 2002), we evaluate these aspects relative to the learner's current cognitive level and the target cognitive process. Given dialogue context h_t and tutor response a_t, a gated multi-head reward model produces seven scores, $q = (q_{\mathrm{soc}}, q_{\mathrm{path}}, q_{\mathrm{judge}}, q_{\mathrm{hon}}, q_{\mathrm{use}}, q_{\mathrm{safe}}, q_{\mathrm{match}})$. The tutoring-specific heads assess Socratic teaching strategy, path stability, and judgment accuracy. Honesty/authenticity provides a shared reliability criterion across tutoring and general samples, while usefulness, harmlessness, and matching degree assess general response quality. Detailed criteria are provided in Appendix C.2.

Task-dependent routing. Tutoring samples activate $\mathcal{H}_{\mathrm{tar}} = \{\mathrm{soc}, \mathrm{path}, \mathrm{judge}, \mathrm{hon}\}$, whereas general samples activate $\mathcal{H}_{\mathrm{gen}} = \{\mathrm{hon}, \mathrm{use}, \mathrm{safe}, \mathrm{match}\}$. A binary router g_ϕ is jointly trained with the reward heads: task labels determine the supervised branch during training, and the learned router selects the branch at inference time.

Reward Learning and Aggregation. Training proceeds in two stages. Pointwise supervision first calibrates the active dimension-level scores and router; Bradley–Terry pairwise supervision then improves within-context ranking while retaining the pointwise objective. An illustrative preference pair is provided in Appendix C.1.

For tutoring samples, the three core dimensions are combined using a weighted geometric mean and then balanced with the shared honesty score:

$$R_{\mathrm{tar}} = \lambda_{\mathrm{tar}} \exp\left(\sum_{h \in \mathcal{H}_{\mathrm{core}}} w_h \log\big(q^h + \epsilon\big) \right) + (1 - \lambda_{\mathrm{tar}})\, q^{\mathrm{hon}}. \tag{2}$$

$$\mathcal{H}_{\mathrm{core}} = \{\mathrm{soc}, \mathrm{path}, \mathrm{judge}\}, \qquad \sum_h w_h = 1. \tag{3}$$

The geometric mean limits compensation across core tutoring dimensions, so a severe weakness cannot be masked by high scores elsewhere. General scores are aggregated linearly as $R_{\text{gen}} = \sum_{h \in \mathcal{H}_{\text{gen}}} \bar{v}_h q^h$, where $\sum_h \bar{v}_h = 1$. The router produces the final score $R_{\text{RM}} = (1 - g_\phi) R_{\text{tar}} + g_\phi R_{\text{gen}}$. Exact losses, aggregation weights, and coefficients are provided in Appendix C.4. The reward model measures response quality but does not directly enforce ordered cognitive progression. Sec. 3.4.addresses this limitation through cognition-sensitive reinforcement learning.

3.4 Cognition-Sensitive Reinforcement Learning

To address this control problem, we introduce a cognitive advancement reward that complements R_{RM} during policy optimization. While R_{RM} rewards response quality, the additional signal measures the response's target level relative to the learner's current cognitive level. It favors the immediate next level in the adopted progression order and assigns lower rewards to deviations, including skipped or backward targets. Together, these signals encourage high-quality responses that provide stable, stepwise cognitive guidance.

3.4.1 Cognitive Advancement Reward

Cognitive advancement signal. During reinforcement learning, the policy is conditioned only on the dialogue context, while the query-level annotation c_t is used solely for reward computation, avoiding additional online diagnosis during rollout. After removing reasoning traces and explicit level annotations, a frozen classifier predicts the level addressed by the cleaned tutor response a_t and its confidence margin: $(g_t, \gamma_t) = \mathcal{D}_\psi(a_t)$. Here, g_t denotes the response's target level rather than the learner's post-response state. With $\text{idx}(\ell_k) = k$, we define $\Delta_t = \text{idx}(g_t) - \text{idx}(c_t)$ and $R_{\text{adv}} = \mathcal{F}(\Delta_t)$. The reward is highest for the immediate next level, assigns partial credit to nearby levels, and penalizes larger deviations. It is applied only to tutoring samples whose classifier margin exceeds the confidence threshold; otherwise, $R_{\text{adv}} = 0$. For general samples, R_{adv} is always set to zero. The full reward schedule and confidence threshold are provided in Appendix D.2, D.3.

3.4.2 Cognition-Sensitive Policy Optimization

We combine R_{RM} and R_{adv} with auxiliary rewards for response length and output structure. Before combining the reward components, a deterministic validity gate $G(a_t)$ filters empty visible responses, explicit level or internal-label leakage, and predefined direct-answer patterns. Candidates that fail the gate receive a fixed penalty. For valid candidates, the reward is

$$R_t = G(a_t)\big[\lambda_{\text{RM}} R_{\text{RM}} + \lambda_{\text{adv}} R_{\text{adv}} + \lambda_{\text{len}} R_{\text{len}} + \lambda_{\text{struct}} R_{\text{struct}}\big] + (1 - G(a_t)) R_{\text{invalid}}. \quad (4)$$

Here, R_{len} and R_{struct} provide auxiliary signals for response length and output structure. Exact reward coefficients and auxiliary reward definitions are provided in Appendix D.4. We optimize π_{SFT} with GRPO using this reward; the optimization procedure is detailed in Appendix D.4.

3.5 CogEdu-Bench: Construction and Evaluation

We introduce CogEdu-Bench to assess whether a tutor can diagnose the learner's current cognitive level, make an appropriate teaching decision, and generate an effective teaching response. Rather than collapsing these capabilities into a single score, the benchmark separates them into two tasks: (1) cognitive-state diagnosis and pedagogical decision making, and (2) teaching-response quality evaluation.

3.5.1 Benchmark Construction

Data construction and quality control. We construct CogEdu-Bench from C-Eval questions (Huang et al., 2023) across eight subjects, clustered and deduplicated by topic and knowledge content. Selected questions support multi-turn tutoring dialogues spanning cognitive levels and two conditions: learners ready to advance after achieving the current target, or requiring further intervention. We check achieved levels, pedagogical decisions, and target levels against dialogue evidence and exclude instances with answer leakage or explicit cognitive-level labels. The benchmark contains 849 instances, approximately balanced across STEM and humanities, junior and senior secondary levels, and Advance and Intervention decisions.

Instance structure and evaluation protocol. Each instance contains instructional metadata, a dialogue history ending with the learner's latest utterance, and gold annotations of the achieved cognitive level, pedagogical decision, target level, and ideal response. These annotations serve as evaluation references and are withheld from the model. Given only the metadata and dialogue history, the model produces a structured output containing the achieved level, an Advance or Intervention decision, and the subsequent teaching response.

3.5.2 Evaluation Tasks and Metrics

Task 1: Cognitive-level Diagnosis and Pedagogical Decision Making. We evaluate diagnosis of the learner's demonstrated cognitive level using Major QWK over six Bloom categories and conditional Minor QWK over 19 fine-grained processes. Minor QWK is computed within each category using only major-correct predictions and aggregated by retained-sample counts. Pedagogical decisions are either *Advance* for learners ready to progress beyond the current target or *Intervention* for those needing further explanation or scaffolding. Decision Accuracy covers all benchmark instances, counting missing, malformed, or invalid outputs as incorrect. Figure 2 compares models' cognitive diagnosis and pedagogical decision-making capabilities(See Appendix E.3).

Task 2: Tutoring Quality. We evaluate factual reliability along four dimensions—relevance, accuracy, answer-leakage control, and substantive content—and scaffolding quality along six dimensions—target-level matching, reasoning guidance, instructional scaffolding, affective support, decision consistency, and information completeness. Each dimension is scored from 0 to 4 and normalized to $[0, 1]$. For instance i, the two axis scores and tutoring-quality score are $S_{\mathrm{FC},i} = \frac{1}{4}\sum_{j=1}^{4} \mathrm{FC}_{j,i}$, $S_{\mathrm{SC},i} = \frac{1}{6}\sum_{j=1}^{6} \mathrm{SC}_{j,i}$, and $S_{\mathrm{T2},i} = \sqrt{S_{\mathrm{FC},i} S_{\mathrm{SC},i}}$. Tutoring Quality is the mean of $S_{\mathrm{T2},i}$ across completed judgments. The geometric mean penalizes responses that perform poorly on either axis(See Appendix E.3).

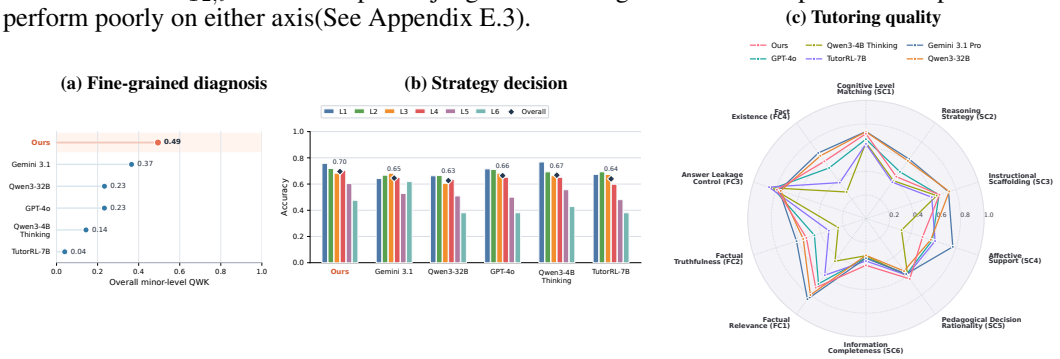


Figure 2: CogEdu-Bench results. (a) Fine-grained cognitive-process diagnosis. (b) Pedagogical-decision accuracy . (c) Tutoring quality across FC and SC.

Reliability validation. To validate the reliability of CogEdu-Bench, we conduct expert-based audits for both evaluation tasks, examining inter-annotator agreement for the Task-1 reference annotations and agreement between human experts and the automatic evaluator for Task 2. Detailed annotation procedures, sampling protocols, and reliability statistics are provided in Appendix E.5.

4 Experiments

4.1 Experimental Setup

Models and baselines. We compare CogRaise with three groups of reference models. The Qwen3-4B-Thinking base model and its SFT separate the effects of supervised initialization and subsequent cognition-sensitive reinforcement learning. TutorRL (Ruan et al., 2024) and SocraticLM (Liu et al., 2024) serve as specialized tutoring baselines, while larger general-purpose LLMs provide capability references for comparing CogRaise with higher-capacity foundation models.

Evaluation benchmarks. CogEdu-Bench (Sec. 3.5) directly evaluates the capabilities targeted by CogRaise: cognitive-level diagnosis, pedagogical decision making, and tutoring quality. MathTutorBench (Macina et al., 2025) assesses broader tutoring capabilities, while GSM8K (Cobbe et al.,

2021a), MATH500 (Hendrycks et al., 2021), and MMLU-Pro (Wang et al., 2024c) measure general-reasoning retention. Together, these benchmarks distinguish gains in cognition-aware tutoring from broader tutoring generalization and general-capability retention.

Evaluation metrics. For CogEdu-Bench, we report four primary metrics. Major QWK evaluates diagnosis across the six broad-level Bloom categories, while Minor QWK evaluates diagnosis across the 19 fine-grained cognitive levels. Decision Accuracy measures pedagogical decision making, while Tutoring Quality evaluates tutoring response quality by combining factual correctness and scaffolding quality. For MathTutorBench, we report four representative capabilities: problem solving, mistake correction, scaffolding generation, and pedagogy following. We use accuracy for problem solving and mistake correction, and win rate for scaffolding generation and pedagogy following. Complete task-level results are provided in Appendix G.3.

Implementation details. CogRaise is initialized from the theory-driven SFT policy described in Sec. 3.2 and optimized using the gated multi-head reward model and cognition-sensitive GRPO objective introduced in Secs. 3.3 and 3.4.2. The reward model undergoes pointwise supervision followed by pairwise refinement, and its score is combined with the cognitive-advancement reward during GRPO. We use the same final RL checkpoint for all evaluations. Training configurations, including LoRA settings, optimization parameters, reward coefficients, generation settings, and hardware, are provided in Appendix A.

4.2 Main Results

Table 1: Main results on CogEdu-Bench.

Model	Major QWK ↑	Minor QWK ↑	Decision Acc. ↑	Tutoring Quality ↑
Our Models				
Qwen3-4B-Thinking (base)	0.2356	0.1435	0.6678	0.4241
Qwen3-4B-Thinking-SFT	0.2924	0.3590	<u>0.6855</u>	0.5687
CogRaise (full)	0.2875	**0.4940**	**0.6961**	0.5686
Baselines—Specialized Tutoring Models				
TutorRL-7B-think	0.0900	0.0400	0.6396	0.4938
SocraticLM	0.1081	0.0400	0.5735	0.3280
Open-Weight Models				
Qwen3-32B	0.2615	0.2334	0.6266	0.6029
GLM-4.7	<u>0.6301</u>	<u>0.3875</u>	0.6259	<u>0.6380</u>
Closed-Source Models				
GPT-4o	0.4880	0.2332	0.6643	0.5452
Gemini-3.1-Pro-Preview	**0.8155**	0.3654	0.6455	**0.6485**

Note. Higher values are better. Minor QWK is conditional on a correct major-category prediction. Best and second-best results are shown in bold and underlined, respectively.

Cognition-aware tutoring on CogEdu-Bench. Table 1 shows that theory-driven SFT substantially improves all four metrics, particularly Minor QWK and Tutoring Quality, providing a stronger initialization. Subsequent cognition-sensitive RL increases Minor QWK from 0.3590 to 0.4940 and improves Decision Accuracy, while Major QWK remains stable and response quality is preserved. Despite its 4B backbone, CogRaise achieves the highest Minor QWK and Decision Accuracy among all evaluated models. Several larger models perform better on Major QWK or Tutoring Quality but remain weaker on fine-grained diagnosis and pedagogical decisions. These results suggest that model scale alone is insufficient for the targeted cognition-aware capabilities, while explicit cognition-aware supervision and reward-guided optimization provide complementary benefits.

Retention of broader tutoring capabilities. We next evaluate CogRaise on MathTutorBench (Macina et al., 2025). We report representative results for problem solving, mistake correction, scaffolding generation, and pedagogical instruction following, as Table 2. For scaffolding generation and pedagogical instruction following, we report MathTutorBench's win rate against the reference teacher responses. Compared with the SFT initialization, CogRaise preserves problem-solving accuracy, while improving mistake-correction accuracy from 0.7774 to 0.8104 and scaffolding win rate from 0.4113 to 0.4261. Pedagogical instruction-following performance remains close to the SFT level. These results indicate that cognition-sensitive optimization strengthens the targeted

capabilities without substantially compromising the broader tutoring profile established during SFT.

Table 2: Representative results on MathTutorBench.

Model	MC Acc. ↑	PF Win ↑	SG Match ↑	SG Win ↑	SG Hard ↑	PS Acc. ↑
Our Models						
Qwen3-4B-Thinking (base)	0.8752	0.5791	0.4557	0.2557	-0.6590	0.9583
Qwen3-4B-Thinking-SFT	0.7774	0.5513	0.4670	0.4113	2.0030	0.9416
CogRaise	0.8104	0.5383	0.4835	0.4261	2.1680	0.9409
Baselines — Specialized Tutoring Models						
TutorRL-7B-think	0.6796	0.6183	0.4496	0.4174	3.0862	0.8582
SocraticLM	0.2265	0.3965	0.8643	0.3391	-0.2920	0.8400
Open-Weight Models						
Qwen3-32B	0.8653	0.5913	0.5165	0.2374	0.0560	**0.9651**
Closed-Source Models						
GPT-4o	0.7655	**0.7957**	0.5504	0.4322	2.5840	0.8764
Gemini-3.1-Pro-Preview	**0.9521**	0.7609	**0.8765**	**0.7565**	**7.1198**	0.8362

Preservation of general reasoning. As Table 3, CogRaise improves GSM8K and MATH500 over SFT, with MMLU-Pro nearly unchanged. It also outperforms specialized tutoring baselines on all three benchmarks. These results suggest that cognition-sensitive optimization adds targeted cognition-aware capabilities with negligible additional cost to the general reasoning performance established during SFT.

Table 3: Performance on general reasoning benchmarks.

Model	GSM8K ↑	MATH500 ↑	MMLU-Pro ↑
Our Models			
base	0.6277	0.7920	**0.7378**
SFT	0.8476	0.8920	0.6951
CogRaise	**0.8506**	**0.8940**	0.6934
Baselines			
TutorRL-7B-think	0.8196	0.7460	0.5501
SocraticLM	0.7900	0.8100	0.3479

4.3 Ablation Study

Table 4: Reward-component ablations on CogEdu-Bench. Higher values are better.

Model	RM	R_{adv}	Aux.	Major QWK ↑	Minor QWK ↑	Decision Acc ↑	Tutoring Quality. ↑
GRPO + RM	✓	–	–	0.2733	0.3658	0.6914	0.5638
GRPO + R_{adv}	–	✓	–	0.2806	0.3198	0.6855	0.5680
GRPO + Aux.	–	–	✓	0.2527	0.4851	0.6784	0.5601
CogRaise	✓	✓	✓	**0.2875**	**0.4940**	**0.6961**	**0.5686**

Table 4 compares CogRaise with variants using only reward-model, advancement, or auxiliary rewards. Auxiliary-only training achieves a conditional Minor QWK close to CogRaise but has the lowest Major QWK, Decision Accuracy, and Tutoring Quality among these variants. Because Minor QWK is evaluated on model-dependent major-correct subsets, its high value alone does not establish stronger overall diagnosis. CogRaise achieves the highest point estimates across all four metrics, supporting the combined reward configuration for the joint tutoring objectives. Further details are provided in Appendix H.3.

4.4 Analysis

Analysis of reward measurement. On 156 held-out preference pairs evaluated using the RL reward-scoring procedure, the final hybrid model achieves 91.67% ranking accuracy, compared with 82.69% for the pointwise checkpoint, supporting its role in measuring tutoring-response quality.

Analysis of cognitive advancement control. Using the frozen classifier, we analyze cognitive offsets on 96 queries confidently classified across SFT, GRPO + RM, and CogRaise. Compared with GRPO + RM, CogRaise increases immediate-next-level responses ($\Delta = 1$) from 17.71% to 21.88% and reduces backward offsets and large forward deviations ($\Delta \geq 3$). These modest shifts support the intended role of R_{adv} in guiding locally appropriate cognitive advancement.

5 Conclusion

We introduced CogRaise, a 4B cognition-aware tutor combining theory-guided supervision, structured rewards, and targeted evaluation. It improves fine-grained diagnosis within correctly predicted cognitive categories and alignment with instructional targets. However, its impact on learners' cognitive development remains unverified. Future work will assess learning outcomes through longitudinal studies with real learners.

AI USE STATEMENT

Beyond serving as components of our experimental pipeline (i.e., LLMs used for synthetic data generation, candidate-response generation, and evaluation), we also used general-purpose LLMs as research assistants. Specifically, they were employed to (i) assist with drafting and debugging code for data processing, model training, and evaluation, and (ii) polish grammar and wording and suggest LaTeX and formatting adjustments for the manuscript. The LLMs did not autonomously determine the research questions, methodological design, experimental conclusions, or scientific claims; all technical decisions, data curation, analyses, and conclusions are by the authors. All LLM-generated code, text, and data outputs used in the final work were reviewed and validated by the authors.

REFERENCES

Ghodai Abdelrahman, Qing Wang, and Bernardo Nunes. Knowledge tracing: A survey. *ACM Computing Surveys*, 55(11):224, 2023.

Vincent Aleven, Bruce McLaren, Ido Roll, and Kenneth Koedinger. Toward tutoring help seeking. In James C. Lester, Rosa Maria Viccari, and Fábio Paraguaçu (eds.), *Intelligent Tutoring Systems*, pp. 227–239, Berlin, Heidelberg, 2004. Springer.

Lorin W. Anderson and David R. Krathwohl. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Longman, New York, NY, 2001.

Paul Black and Dylan Wiliam. Developing the theory of formative assessment. *Educational Assessment, Evaluation and Accountability*, 21(1):5–31, 2009. doi: 10.1007/s11092-008-9068-5.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021a. URL https://arxiv.org/abs/2110.14168.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021b. URL https://arxiv.org/abs/2110.14168.

National Research Council. *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. The National Academies Press, Washington, DC, 2000.

David Dinucu-Jianu, Jakub Macina, Nico Daheim, Ido Hakimi, Iryna Gurevych, and Mrinmaya Sachan. From problem-solving to teaching problem-solving: Aligning LLMs with pedagogy using reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 272–292, Suzhou, China, 2025. Association for Computational Linguistics.

Scott Elkins, Ekaterina Kochmar, Jacky C. K. Cheung, and Iulian Serban. How teachers can use large language models and Bloom's taxonomy to create educational quizzes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(21):23084–23091, 2024.

Rima Hazra, Bikram Ghuku, Ilona Marchenko, Yaroslava Tokarieva, Sayan Layek, Somnath Banerjee, Julia Stoyanovich, and Mykola Pechenizkiy. Safetutors: Benchmarking pedagogical safety in ai tutoring systems, 2026. URL https://arxiv.org/abs/2603.17373.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the NeurIPS Track on Datasets and Benchmarks (NeurIPS 21 Datasets and Benchmarks)*, volume 1. Curran Associates, 2021.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. C-Eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems (NeurIPS 23)*, volume 36, pp. 62991–63010. Curran Associates, 2023.

Xiaokang Jin, Jia Zhu, Jingjiang Liu, Yabing Shi, Jueqi Guan, Hao Chen, and Pasquale De Meo. PedagogyBench: A cognitive-driven benchmark for multimodal instructional video understanding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 12621–12647, San Diego, California, 2026. Association for Computational Linguistics.

Haikun Liu, Weijie Peng, and Chao Qi. Knowledge tracing with multi-feature fusion and question difficulty. In *2022 IEEE 21st International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)*, pp. 231–238, 2022.

Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. SocraticLM: Exploring Socratic personalized teaching with large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS 24)*, pp. 2721, Vancouver, BC, Canada, 2024. Curran Associates.

Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5602–5621, Singapore, December 2023. Association for Computational Linguistics.

Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. MathTutorBench: A benchmark for measuring open-ended pedagogical capabilities of LLM tutors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 204–221, Suzhou, China, 2025. Association for Computational Linguistics.

Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina 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 Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), Volume 1: Long Papers*, pp. 1234–1251, Albuquerque, New Mexico, 2025. Association for Computational Linguistics.

Richard E. Mayer. Rote versus meaningful learning. *Theory Into Practice*, 41(4):226–232, 2002. doi: 10.1207/s15430421tip4104_4.

Calarina Muslimani, Alex Lewandowski, Dale Schuurmans, Matthew E. Taylor, and Jun Luo. Reinforcement Teaching. *Transactions on Machine Learning Research*, 2023. URL `https://mlanthology.org/tmlr/2023/muslimani2023tmlr-reinforcement/`.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS 22)*, pp. 2011, New Orleans, LA, USA, 2022. Curran Associates.

Sankalan Pal Chowdhury, Vilém Zouhar, and Mrinmaya Sachan. AutoTutor meets large language models: A language model tutor with rich pedagogy and guardrails. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale (L@S '24)*, pp. 5–15, Atlanta, GA, 2024. Association for Computing Machinery.

Richard Paul and Linda Elder. Critical thinking: The art of Socratic questioning, part III. *Journal of Developmental Education*, 31, 2008.

Romain Puech, Jakub Macina, Julia Chatain, Mrinmaya Sachan, and Manu Kapur. Towards the pedagogical steering of large language models for tutoring: A case study with modeling productive failure. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 26291–26311, Vienna, Austria, 2025. Association for Computational Linguistics.

Sherry Ruan, Allen Nie, William Steenbergen, Jiayu He, J. Q. Zhang, Meng Guo, Yao Liu, Kyle Dang Nguyen, Catherine Y. Wang, Rui Ying, James A. Landay, and Emma Brunskill. Reinforcement learning tutor better supported lower performers in a math task. *Machine Learning*, 113(5): 3023–3048, 2024. doi: 10.1007/s10994-023-06423-9.

Nicy Scaria, Suma Dharani Chenna, and Deepak Subramani. Automated educational question generation at different Bloom's skill levels using large language models: Strategies and evaluation. In Andrew M. Olney, Ioanna A. Chounta, Zita Liu, Olga C. Santos, and Ig I. Bittencourt (eds.), *Artificial Intelligence in Education. AIED 2024*, volume 14830 of *Lecture Notes in Computer Science*, Cham, 2024. Springer.

Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, Idan Szpektor, Avinatan Hassidim, Yossi Matias, and Rémi Munos. Multi-turn reinforcement learning with preference human feedback. In *Advances in Neural Information Processing Systems (NeurIPS 24)*, volume 37, pp. 118953–118993. Curran Associates, 2024.

Xin Shi, Chao Zhang, Yifan Zhu, Xueqiao Zhang, and Yawei Luo. Beyond pedagogical principles: Multi-horizon preference optimization for efficient socratic tutoring. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pp. 11289–11306, San Diego, California, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.518. URL https://aclanthology.org/2026.acl-long.518/.

Raghvendra Singh, Pavankumar Mulgund, Yafang Li, and Sai Dheeraj Vodela. Enhancing compliance training with LLM-powered conversational agents: A design science evaluation of FERPA education. In *Proceedings of the 63rd ACM Conference on Computers and People Research (SIGMIS-CPR '26)*, pp. 84–90, USA, 2026. Association for Computing Machinery.

Shashank Sonkar, Kangqi Ni, Sapana Chaudhary, and Richard Baraniuk. Pedagogical alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13641–13650, Miami, Florida, 2024. Association for Computational Linguistics.

Abdul Waheed, Muskan Goyal, Nimisha Mittal, Deepak Gupta, Ashish Khanna, and Moolchand Sharma. BloomNet: A robust transformer-based model for Bloom's learning outcome classification. In *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pp. 209–218, Trento, Italy, 2021. Association for Computational Linguistics.

Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. Neural cognitive diagnosis for intelligent education systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4):6153–6161, 2020.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. Arithmetic control of LLMs for diverse user preferences: Directional preference alignment with multi-objective rewards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pp. 8642–8655, Bangkok, Thailand, 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.468. URL https://aclanthology.org/2024.acl-long.468/.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10582–10592, Miami, Florida, 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.620. URL https://aclanthology.org/2024.findings-emnlp.620/.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS 24)*, volume 37, pp. 95266–95290. Curran Associates, 2024c.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. HelpSteer 2: Open-source dataset for training top-performing reward models. In *Advances in Neural Information Processing Systems (NeurIPS 24)*, volume 37, pp. 1474–1501. Curran Associates, 2024d.

David Wood, Jerome S. Bruner, and Gail Ross. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2):89–100, 1976. doi: 10.1111/j.1469-7610.1976.tb00381.x.

Wenjin Yao, Yidong Wang, Zhuohao Yu, Rui Xie, Shikun Zhang, and Wei Ye. PURE: Aligning LLM via pluggable query reformulation for enhanced helpfulness. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8721–8744, Miami, Florida, 2024. Association for Computational Linguistics.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2299–2314, Mexico City, Mexico, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.149. URL https://aclanthology.org/2024.findings-naacl.149/.

A Implementation Details

A.1 Supervised Fine-Tuning

Supervised Fine-Tuning. We initialize CogRaise from Qwen3-4B-Thinking-2507 and perform supervised fine-tuning using LoRA. The LoRA adapters are applied to the query, key, value, output, gate, up-projection, and down-projection layers, with rank $r = 16$, scaling factor $\alpha = 16$, and dropout 0.1. We train for three epochs using a learning rate of 5×10^{-5}, cosine learning-rate scheduling, a warm-up ratio of 0.05, and weight decay of 0.01. Training uses bfloat16 precision, gradient checkpointing, a maximum sequence length of 4,096 tokens, and a random seed of 42. The per-device batch size is 1 with 16 gradient-accumulation steps. Five percent of the SFT corpus is randomly held out for validation, and the checkpoint with the lowest validation loss is retained.

A.2 Reward Model Training

Reward Model Training. The reward model is initialized from the merged SFT checkpoint and equipped with seven scalar reward heads and a binary routing gate. The backbone is adapted with LoRA using rank $r = 16$, $\alpha = 32$, dropout 0.05, and the query and value projection layers as target modules.

We train the reward model in two stages. The pointwise stage directly fits the dimension-wise reward scores of the active heads while jointly supervising the routing gate. It uses separate learning rates of 1×10^{-5} for the backbone and 1×10^{-4} for the reward heads and gate. Training runs for three epochs with a per-device batch size of 2 and eight gradient-accumulation steps.

The second stage warm-starts from the pointwise checkpoint and introduces pairwise Bradley–Terry supervision while retaining pointwise score regression. The hybrid objective weights the pointwise MSE loss by 2.0, the pairwise loss by 0.5, and the gate classification loss by 0.5. Hybrid training runs for three epochs with a per-device batch size of 1, 16 gradient-accumulation steps, a backbone learning rate of 1×10^{-5}, and a head/gate learning rate of 1×10^{-4}. The final epoch-3 checkpoint is used as the frozen reward model for reinforcement learning. It attains the best pairwise validation accuracy (0.7624) and Bradley–Terry loss (0.5388) among the evaluated checkpoints.

A.3 Reinforcement Learning

Reinforcement Learning. CogRaise is initialized from the merged SFT policy and optimized with GRPO using LoRA adapters with rank $r = 16$, $\alpha = 32$, dropout 0.05, and all linear layers as adaptation targets. Policy optimization uses a per-device batch size of 8, eight candidate generations per query, four gradient-accumulation steps, a learning rate of 1×10^{-6}, and a KL coefficient $\beta = 0.02$. The maximum completion length is 1,024 tokens, and training proceeds for one epoch in bfloat16 precision with gradient checkpointing.

Training and generation are separated across three GPUs: one hosts the policy optimizer, one hosts the frozen reward-model server, and one hosts the vLLM generation server. Because generation and

training use separate policy engines, we apply token-level truncated importance-sampling correction during GRPO optimization.

A.4 Hardware and Compute

Hardware and Compute. All training experiments were conducted on a server equipped with three NVIDIA RTX 4090 GPUs. Reward-model training uses multi-GPU distributed training, while reinforcement learning assigns the three GPUs to separate components: the policy optimizer, the frozen reward-model server, and the vLLM generation server, respectively. This separation avoids competition between policy optimization, reward inference, and rollout generation during GRPO training

B Theory-Driven Data Construction

B.1 Cognitive-Level Taxonomy

Cognitive-Level Taxonomy. CogRaise operationalizes the 19 fine-grained cognitive processes in Anderson's revised Bloom taxonomy as a cognitive-level space for tutoring. The six broad categories retain the hierarchical organization of the original taxonomy, while the fine-grained processes provide the instructional resolution required for modeling local cognitive advancement. For our tutoring framework, we impose an ordinal sequence on the 19 processes and denote the resulting level space by

$$\mathcal{C} = (\ell_1, \ell_2, \ldots, \ell_{19}) .$$

The ordering is an operationalization for tutoring rather than an order directly specified by the original taxonomy. It provides a tractable representation of cognitive progression that can be used to construct supervised trajectories and define local instructional targets. The original cognitive-process dimension is reproduced **(see Figures 3 and 4)**.

5.1 THE COGNITIVE PROCESS DIMENSION

CATEGORIES & COGNITIVE PROCESSES	ALTERNATIVE NAMES	DEFINITIONS AND EXAMPLES
1. REMEMBER—Retrieve relevant knowledge from long-term memory		
1.1 RECOGNIZING	Identifying	Locating knowledge in long-term memory that is consistent with presented material (e.g., Recognize the dates of important events in U.S. history)
1.2 RECALLING	Retrieving	Retrieving relevant knowledge from long-term memory (e.g., Recall the dates of important events in U.S. history)
2. UNDERSTAND—Construct meaning from instructional messages, including oral, written, and graphic communication		
2.1 INTERPRETING	Clarifying, paraphrasing, representing, translating	Changing from one form of representation (e.g., numerical) to another (e.g., verbal) (e.g., Paraphrase important speeches and documents)
2.2 EXEMPLIFYING	Illustrating, instantiating	Finding a specific example or illustration of a concept or principle (e.g., Give examples of various artistic painting styles)
2.3 CLASSIFYING	Categorizing, subsuming	Determining that something belongs to a category (e.g., Classify observed or described cases of mental disorders)
2.4 SUMMARIZING	Abstracting, generalizing	Abstracting a general theme or major point(s) (e.g. Write a short summary of the event portrayed on a videotape)
2.5 INFERRING	Concluding, extrapolating, interpolating, predicting	Drawing a logical conclusion from presented information (e.g., In learning a foreign language, infer grammatical principles from examples)
2.6 COMPARING	Contrasting, mapping, matching	Detecting correspondences between two ideas, objects, and the like (e.g., Compare historical events to contemporary situations)
2.7 EXPLAINING	Constructing models	Constructing a cause-and-effect model of a system(e.g., explain the causes of important 18th Century events in France)
3. APPLY—Carry out or use a procedure in a given situation		
3.1 EXECUTING	Carrying out	Applying a procedure to a familiar task (e.g., Divide one whole number by another whole number, both with multiple digits)
3.2 IMPLEMENTING	Using	Applying a procedure to an unfamiliar task (e.g., Use Newton's Second Law in situations in which it is appropriate)

Figure 3: The cognitive-process dimension of the revised Bloom taxonomy: Remember, Understand, and Apply. Source: Anderson and Krathwohl (2001).

5.1 THE COGNITIVE PROCESS DIMENSION (CONTINUED)

CATEGORIES & COGNITIVE PROCESSES	ALTERNATIVE NAMES	DEFINITIONS AND EXAMPLES
4. ANALYZE—Break material into its constituent parts and determine how the parts relate to one another and to an overall structure or purpose		
4.1 DIFFERENTIATING	Discriminating, distinguishing, focusing, selecting	Distinguishing relevant from irrelevant parts or important from unimportant parts of presented material (e.g., Distinguish between relevant and irrelevant numbers in a mathematical word problem)
4.2 ORGANIZING	Finding coherence, intergrating, outlining, parsing, structuring	Determining how elements fit or function within a structure (e.g., Structure evidence in a historical description into evidence for and against a particular historical explanation)
4.3 ATTRIBUTING	Deconstructing	Determine a point of view, bias, values, or intent underlying presented material (e.g., Determine the point of view of the author of an essay in terms of his or her political perspective)
5. EVALUATE—Make judgments based on criteria and standards		
5.1 CHECKING	Coordinating, detecting, monitoring, testing	Detecting inconsistencies or fallacies within a process or product; determining whether a process or product has internal consistency; detecting the effectiveness of a procedure as it is being implemented (e.g., Determine if a scientist's conclusions follow from observed data)
5.2 CRITIQUING	Judging	Detecting inconsistencies between a product and external criteria, determining whether a product has external consistency; detecting the appropriateness of a procedure for a given problem (e.g., Judge which of two methods is the best way to solve a given problem)
6. CREATE—Put elements together to form a coherent or functional whole; reorganize elements into a new pattern or structure		
6.1 GENERATING	Hypothesizing	Coming up with alternative hypotheses based on criteria (e.g., Generate hypotheses to account for an observed phenomenon)
6.2 PLANNING	Designing	Devising a procedure for accomplishing some task (e.g., Plan a research paper on a given historical topic)
6.3 PRODUCING	Constructing	Inventing a product (e.g., Build habitats for a specific purpose)

Figure 4: The cognitive-process dimension of the revised Bloom taxonomy: Analyze, Evaluate, and Create. Source: Anderson and Krathwohl (2001).

B.2 Cognitive-Level Update and Teaching Decision

Cognitive-Level Update. Given the ordered cognitive-level space, we construct tutoring trajectories using a local successor rule. For a current level ℓ_k, the immediate successor is defined as

$$\mathrm{Next}(\ell_k) = \begin{cases} \ell_{k+1}, & k < 19, \\ \ell_{19}, & k = 19. \end{cases}$$

At tutoring turn t, c_t denotes the cognitive level demonstrated by the learner, while g_t denotes the level that the tutor aims to address. The target therefore represents the intended instructional level rather than a level already attained by the learner.

The learner's cognitive level is updated from the observed response u_t. If the response demonstrates the instructional target from the previous turn, that target becomes the learner's new current level. Otherwise, the learner remains at the previous level:

$$c_t = \begin{cases} g_{t-1}, & \text{if } u_t \text{ demonstrates the previous target,} \\ c_{t-1}, & \text{otherwise.} \end{cases} \qquad g_t = \mathrm{Next}(c_t).$$

This update produces two pedagogical decisions. *Advance* is assigned when the learner demonstrates the previous target level and is therefore ready to move to its successor. *Intervention* is assigned when the learner has not yet demonstrated the previous target and requires additional support before advancement. Thus, the same state-update mechanism jointly determines the learner's current level, the next instructional target, and the corresponding teaching decision.

B.3 SFT Data Construction and Statistics

SFT Data Construction. We construct Bloom-conditioned tutoring dialogues by specifying the subject knowledge, the learner's demonstrated cognitive process, the intended next process, and the instructional decision for each dialogue. We use Qwen-Max, DeepSeek, GPT-5.1, Gemini-3-Flash, Grok-4.1-Fast, and MoonShot as generators to produce varied expressions of learner responses and tutor guidance across cognitive levels. Using multiple generators reduces dependence on a single model's phrasing and progression patterns.

Table 5: Illustrative SFT dialogues with one, two, and four learner–tutor turns. The excerpts are translated from Chinese and lightly abridged; internal reasoning traces are omitted.

Illustrative SFT Training Examples

A. One-turn SFT dialogue: semiconservative DNA replication. Learner. *That is straightforward. Because every round of replication carries one old strand, both daughter molecules contain ^{14}N after the first round. All four molecules after the second round and all eight after the third round must therefore contain an original strand. Thus, eight molecules contain ^{14}N, and none consists entirely of ^{15}N.* **Tutor (3.1 Execute → 3.2 Implement).** *How many ^{14}N-labeled old strands were present in the original DNA molecule? Are these old strands created or destroyed during replication? If their number remains unchanged, how could they appear simultaneously in all eight daughter molecules?*

B. Two-turn SFT dialogue: the Industrial Revolution. Turn 1, Learner. *The Industrial Revolution created the working class and shattered the old aristocratic order.* **Turn 1, Tutor (2.4 Summarize → 2.5 Infer).** *Your summary captures two central effects of the Industrial Revolution. Based on these changes, can you infer how industrialization helped prepare the way for modern democratic institutions? Trace the causal sequence from the emergence of the working class to the redistribution of political power.* **Turn 2, Learner.** *Factory workers gradually recognized that they should have a political voice. They organized unions and strikes, which pressured governments to introduce electoral reforms and extend voting rights beyond the aristocracy. Political power therefore began to move from a small elite toward a broader public.* **Turn 2, Tutor (2.5 Infer → 2.6 Compare).** *Now compare the Industrial Revolution with the French Revolution. Which more thoroughly displaced aristocratic power? How did economic pressure and violent political change provide different routes toward modern democracy?*

C. Four-turn SFT dialogue: interpretations of the Meiji Restoration. Turn 1, Learner. *The author appears to support the Meiji Restoration because the article emphasizes modernization and praises the introduction of Western technology and institutions.* **Turn 1, Tutor (**4.3 **Attribute** $\rightarrow$ 5.1 **Check).** *You have identified the author's position. Is this favorable view of modernization maintained consistently throughout the article, or do any passages create an internal tension?* **Turn 2, Learner.** *The article introduces conservative opposition near the beginning but never explains how that opposition was addressed. This makes the argument appear incomplete.* **Turn 2, Tutor (**5.1 **Check** $\rightarrow$ 5.2 **Critique).** *Compare this account with the generally accepted historical record. Is the author's treatment of conservative opposition consistent with what is known about the period?* **Turn 3, Learner.** *The Meiji government used measures such as stronger central control to suppress conservative resistance, but the article does not discuss these measures.* **Turn 3, Tutor (**5.2 **Critique** $\rightarrow$ 6.1 **Generate).** *If the article included those measures, how might they change the reader's understanding of the Restoration's success? What new interpretation could be developed from that additional evidence?* **Turn 4, Learner.** *The additional information would give readers a more complete understanding of the period, although I am unsure how it would change the article's central claim.* **Turn 4, Tutor (**5.2 **Critique** $\rightarrow$ 6.1 **Generate).** *Consider a counterfactual case: if the Meiji government had responded to conservative opposition through negotiation rather than coercion, how might the outcome of the Restoration have differed? Use this assumption to formulate a new historical interpretation.*

Expert Review of Cognitive-Level Alignment. Three subject-matter experts independently reviewed a sample of level-labeled tutoring turns. For each turn, they assessed whether the annotated learner cognitive level and target level were consistent with the dialogue evidence and subject knowledge, and whether the tutor response appropriately supported the intended cognitive advancement. The three experts showed good agreement in their judgments.

Data Screening and Corpus Composition. We generate 10,323 candidate dialogues. Following structural and control-field consistency checks, removal of malformed records, and reduction of the general-dialogue portion, 6,196 dialogues are retained. These comprise 2,924 Bloom-conditioned tutoring dialogues, 1,849 education-oriented dialogue expansions, and 1,423 general dialogues. These comprise 2,924 Bloom-conditioned tutoring dialogues, 1,849 education-oriented dialogue expansions, and 1,423 general dialogues **(see Table 6)**.

Table 6: Composition of the 6,196 retained SFT dialogues by generator and data type.

Generator	Bloom-conditioned	Education expansion	General	Total
Qwen-Max	1,412	1,849	0	3,261
DeepSeek	96	0	1,423	1,519
GPT-5.1	547	0	0	547
Gemini-3-Flash	436	0	0	436
Grok-4.1-Fast	363	0	0	363
MoonShot	70	0	0	70
Total	**2,924**	**1,849**	**1,423**	**6,196**

Training–Validation Split. We randomly split the retained dialogues at the dialogue level with a fixed seed of 42, yielding 5,886 training dialogues and 310 validation dialogues. The validation set is excluded from gradient updates and is used to select the checkpoint with the lowest validation loss.

C Reward Model Details

C.1 Preference Data Construction

Preference Data Construction. We assemble tutoring and general-instruction dialogue contexts and generate two candidate responses independently for each shared conversation history. Candidate generation varies the response profile and generator model. The resulting data include responses from the SFT checkpoint, Qwen-Max, Gemini-3.1-Pro-Preview, DeepSeek-V4-Flash, and GLM-5.1. The scored export contains 15,746 candidate responses forming 7,873 pairs, comprising 4,743 tutoring pairs and 3,130 general-instruction pairs.

Dimension-wise Scoring. Qwen-Max scores each candidate separately against task-specific rubrics without seeing the other candidate. Tutoring responses are scored on Socratic teaching strategy, cognitive-path stability, judgment accuracy, and honesty/authenticity. General-instruction responses are scored on honesty/authenticity, usefulness, harmlessness, and matching degree.

Each active dimension receives a score from 0 to 1 in increments of 0.05, together with a textual rationale. Length and format compliance are checked separately and are excluded from the aggregate preference score. For a sample of task type τ, we average the active dimensions:

$$y(x) = \frac{1}{|\mathcal{H}_\tau|} \sum_{h \in \mathcal{H}_\tau} y^h(x),$$

where $\mathcal{H}_\tau$ denotes the set of active dimensions for task type τ.

The higher-scoring candidate is assigned as x^+ and the other as x^-. We retain a pair only if its score margin satisfies

$$m = |y(x_1) - y(x_2)| > 0.10.$$

This threshold removes weak or ambiguous preferences. The retained pairs preserve both the chosen/rejected relation and the original dimension-wise scores for hybrid reward-model training.

Qualitative Expert Review. Three subject-matter experts reviewed a subset of response pairs and assessed whether the model-preferred response better satisfied the task-specific scoring criteria. Their judgments showed good agreement. This review provides a qualitative check of the automatically assigned preferences.

Examples from the SFT dialogue corpus and the reward-model preference data are provided **(see Table 7)**.

Table 7: Reward-model preference pair on a photosynthesis misconception. The excerpt is translated from Chinese and lightly abridged.

Reward-Model Training Example

D. Reward-model preference pair: photosynthesis misconception. Learner. *ATP and NADPH from the light reactions are used to fix carbon dioxide in the dark reactions, but I think these substances are unnecessary because plants can obtain energy in other ways.* **Chosen** (4.2 **Organize** $\rightarrow$ 4.3 **Attribute).** *Your organization of photosynthesis is clear, but the causal relationship between the light and dark reactions remains unclear. How are ATP and NADPH used in the dark reactions, and what effects do they have on photosynthesis?* **Rejected** (4.3 **Attribute** $\rightarrow$ 6.1 **Generate).** *If the products of the light reactions are not essential, design an entirely new plant metabolic pathway that fixes carbon dioxide without ATP or NADPH from the light reactions, and predict how the plant would survive in extreme environments.* **Scores.** Chosen: Socratic 0.85, path stability 0.90, judgment 0.90, honesty 0.85, mean 0.875. Rejected: 0.65, 0.20, 0.35, 0.30, mean 0.375. Preference margin $= 0.500 > 0.10$.

C.2 Gated Multi-Head Reward Model

Multi-Head Reward Dimensions. The reward model predicts seven response-level scores:

$$\mathbf{q} = \left(q^{\text{soc}}, q^{\text{path}}, q^{\text{judge}}, q^{\text{hon}}, q^{\text{use}}, q^{\text{safe}}, q^{\text{match}}\right),$$

corresponding to Socratic teaching strategy, path stability, judgment accuracy, honesty/authenticity, usefulness, harmlessness, and matching degree.

Tutoring and general-instruction samples activate different subsets of reward heads:

$$\mathcal{H}_{\text{tutor}} = \{\text{soc}, \text{path}, \text{judge}, \text{hon}\},$$

$$\mathcal{H}_{\text{general}} = \{\text{hon}, \text{use}, \text{safe}, \text{match}\}.$$

A binary router is jointly trained with the reward heads. During training, task labels determine the active heads and supervise the router. We use hard Gumbel–Softmax routing with temperature 1.0. At inference time, the route is selected by

$$g_\phi = \arg\max_k z_k^{\text{gate}}.$$

The predicted route determines which reward branch is used for the input response.

Tutoring-Specific Reward Criteria. **Judgment accuracy** assesses whether the tutor's diagnosis of the learner's current cognitive level agrees with dialogue evidence. **Path stability** assesses stable and correct pedagogical decisions and scaffolding appropriate to that level. **Socratic teaching strategy** assesses whether questions and hints encourage further reasoning without prematurely revealing the answer. The shared honesty/authenticity head evaluates factual reliability and unsupported claims across tutoring and general samples.

C.3 Two-Stage Reward Learning

Pointwise Learning. We first train the active reward heads with pointwise regression while jointly learning the routing gate. Given the logit z_i^h of reward head h, its normalized prediction is

$$q_i^h = \sigma(z_i^h).$$

The pointwise objective is

$$\mathcal{L}_{\text{pw}} = \mathcal{L}_{\text{MSE}} + 0.5\mathcal{L}_{\text{gate}}.$$

Pairwise Refinement. The second stage is initialized from the pointwise model and introduces a margin-aware Bradley–Terry-style ranking objective while retaining pointwise score regression.

For pairwise refinement, we define the aggregate ranking score as

$$s(x) = \frac{1}{|\mathcal{H}_\tau|} \sum_{h \in \mathcal{H}_\tau} z^h(x).$$

The pairwise ranking loss is

$$\mathcal{L}_{\text{BT}} = -\log \sigma\left(s(x^+) - s(x^-) - m\right).$$

The complete hybrid objective is

$$\mathcal{L}_{\text{hyb}} = 2.0\mathcal{L}_{\text{MSE}} + 0.5\mathcal{L}_{\text{BT}} + 0.5\mathcal{L}_{\text{gate}}.$$

The scalar $s(x)$ is used only for pairwise training and is distinct from the final task-specific reward aggregation described below.

C.4 Task-Specific Reward Aggregation

At inference time, each reward-head logit is converted to a normalized score:

$$q^h = \sigma(z^h).$$

For numerical stability, we define

$$\widetilde{q}^h = \max\left(q^h, 10^{-8}\right).$$

For tutoring responses, judgment accuracy, Socratic teaching strategy, and path stability are combined using a weighted geometric mean:

$$R_{\text{tri}} = \left(\widetilde{q}^{\text{judge}}\right)^{0.40} \left(\widetilde{q}^{\text{soc}}\right)^{0.20} \left(\widetilde{q}^{\text{path}}\right)^{0.40}.$$

The tutoring reward is

$$R_{\text{tutor}} = 0.80 R_{\text{tri}} + 0.20 q^{\text{hon}}.$$

The geometric form limits compensation across the three core tutoring dimensions, so a low score on one dimension cannot be fully offset by high scores on the others.

For general-instruction responses, we use a weighted linear aggregation:

$$R_{\text{general}} = 0.25 q^{\text{hon}} + 0.35 q^{\text{use}} + 0.20 q^{\text{safe}} + 0.20 q^{\text{match}}.$$

Finally, the learned route determines the scalar reward:

$$R_{\text{RM}} = (1 - g_\phi) R_{\text{tutor}} + g_\phi R_{\text{general}}.$$

D Cognition-Sensitive Reinforcement Learning Details

D.1 RL Query Set

RL Query Set. We construct an RL query set containing 1,600 dialogue contexts, including 1,000 tutoring queries and 600 general-instruction queries. For tutoring samples, the learner's current cognitive level is precomputed before policy optimization and stored with the query. The policy itself receives the dialogue context, whereas the stored cognitive-level annotation is used only by the reward computation.

Queries for which a reliable current-level annotation is unavailable remain in the training set but receive no cognitive-advancement reward. They can therefore continue contributing through the reward-model and auxiliary objectives.

D.2 Cognitive-Level Classifier

Cognitive-Level Classifier. The cognitive-advancement reward relies on a frozen 19-class classifier that predicts the cognitive level addressed by a generated tutoring response. Before classification, internal reasoning enclosed by `<think>` tags and explicit cognitive-level annotations are removed. This prevents the policy from obtaining advancement reward by directly emitting level labels instead of expressing the corresponding pedagogical behavior.

For a cleaned tutoring response a_t, the classifier outputs a predicted addressed level g_t and a confidence margin γ_t:

$$(g_t, \gamma_t) = D_\psi(a_t).$$

The confidence margin is computed from the two highest classifier decision scores. Predictions with

$$\gamma_t < 0.10$$

are treated as insufficiently confident and receive no cognitive-advancement reward. Here, g_t denotes the cognitive level addressed by the tutor response rather than the learner's cognitive level after receiving that response.

D.3 Cognitive Advancement Reward

Cognitive Advancement Reward. Let $\mathrm{idx}(\ell)$ denote the ordinal position of cognitive level ℓ. For an eligible tutoring response, we define its cognitive offset relative to the learner's current level as

$$\Delta_t = \mathrm{idx}(g_t) - \mathrm{idx}(c_t).$$

The reward assigns the highest value to responses addressing the immediate next cognitive level, partial credit to nearby levels, and negative reward to larger deviations **(see Figure 5)**:

$$R_{\mathrm{adv}}(\Delta) = \begin{cases} 1.0, & \Delta = 1, \\ 0.5, & \Delta \in \{0, 2\}, \\ 0.2, & \Delta = -1, \\ \max\{-2, -0.5(\Delta - 2)\}, & \Delta \geq 3, \\ \max\{-2, -0.5(|\Delta| - 1)\}, & \Delta \leq -2. \end{cases}$$

The advancement reward is applied only to tutoring samples with a valid current-level annotation and a classifier margin of at least 0.10. For all other samples, including general instruction-following samples, $R_{\mathrm{adv}} = 0$.

Figure 5 illustrates the piecewise relationship between the cognitive-level offset and the advancement reward.

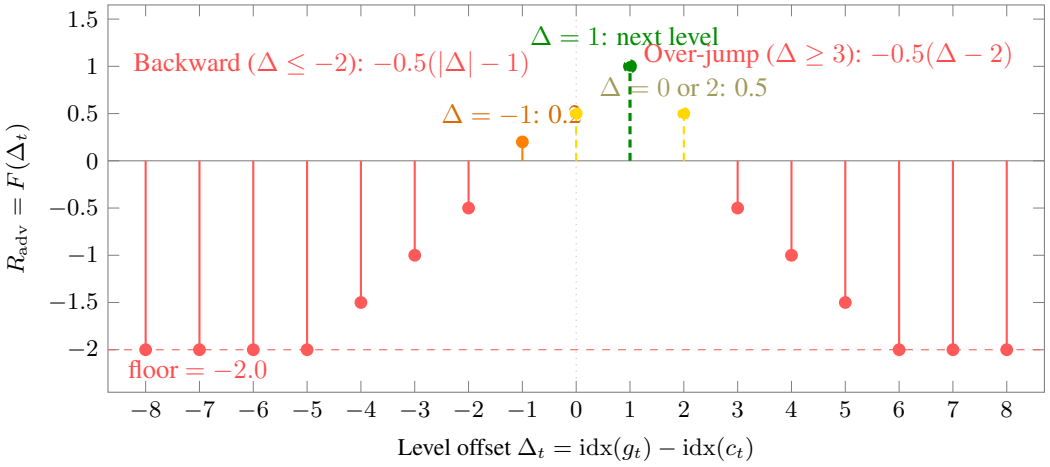


Figure 5: Piecewise cognitive-advancement reward as a function of the cognitive-level offset Δ_t.

D.4 Auxiliary Rewards, Hard Gate, and Total Reward

Auxiliary Rewards. We include lightweight auxiliary rewards for response length and structural compliance. The length reward uses prompt-dependent target ranges and decreases when a completion falls increasingly far outside its designated range. The structure reward checks whether the generated reasoning follows the required output organization.

Validity Gate. Before reward aggregation, a deterministic validity gate rejects responses with empty visible content, explicit internal-level leakage, or predefined direct-answer patterns. Let $G(a) \in \{0, 1\}$ denote whether response a passes this validity check. Invalid responses bypass all reward components and receive the fixed penalty

$$R_{\mathrm{invalid}} = -1.$$

For a valid response, the weighted reward is

$$R_{\text{valid}} = 0.65R_{\text{RM}} + 0.25R_{\text{adv}} + 0.05R_{\text{len}} + 0.05R_{\text{struct}}.$$

Including the validity gate, the complete reward is

$$R(a) = G(a)\,(0.65R_{\text{RM}} + 0.25R_{\text{adv}} + 0.05R_{\text{len}} + 0.05R_{\text{struct}}) + \big(1 - G(a)\big)(-1).$$

E CogEdu-Bench Details

E.1 Construction and Quality Control

Construction. CogEdu-Bench is constructed from middle- and high-school material in the 13,948-question C-Eval source dataset. After removing duplicate and near-duplicate source material, generating tutoring dialogues, and applying quality-control checks, we retain 849 evaluation instances in the final benchmark.

Quality Control. Dialogue generation applies deterministic checks for explicit cognitive-label leakage, direct answer leakage, malformed turn structure, duplicated student answers, unsupported third-person narratives, and the requirement that each released context ends with a student utterance. Stable instance identifiers support cross-run alignment.

In the frozen 849-instance file used for the reported evaluation, instance identifiers are unique, no two complete dialogue histories are identical, and all contexts end with a student turn.

CogEdu-Bench was constructed separately from the datasets used for supervised fine-tuning, reward-model training, and reinforcement learning. Its evaluation instances were not used in any of these training stages.

Final Composition. The benchmark contains 428 humanities and 421 STEM instances, 422 middle-school and 427 high-school instances, and 428 *Advance* versus 421 *Intervention* decisions **(see Figure 6)**. The dataset is therefore balanced at these coarse levels. It is not balanced across all 19 fine-grained cognitive processes: only 21 instances belong to the Create category**(see Figure 7)**.

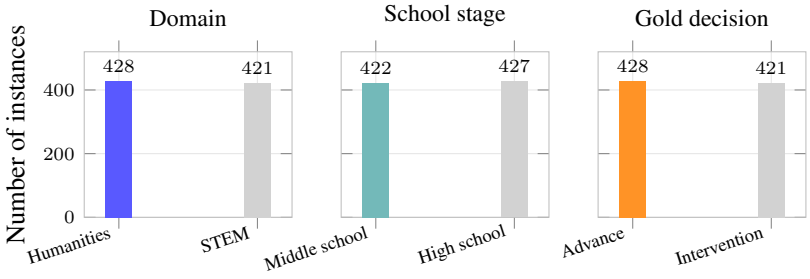


Figure 6: Coarse composition of CogEdu-Bench by subject domain, school stage, and gold pedagogical decision.

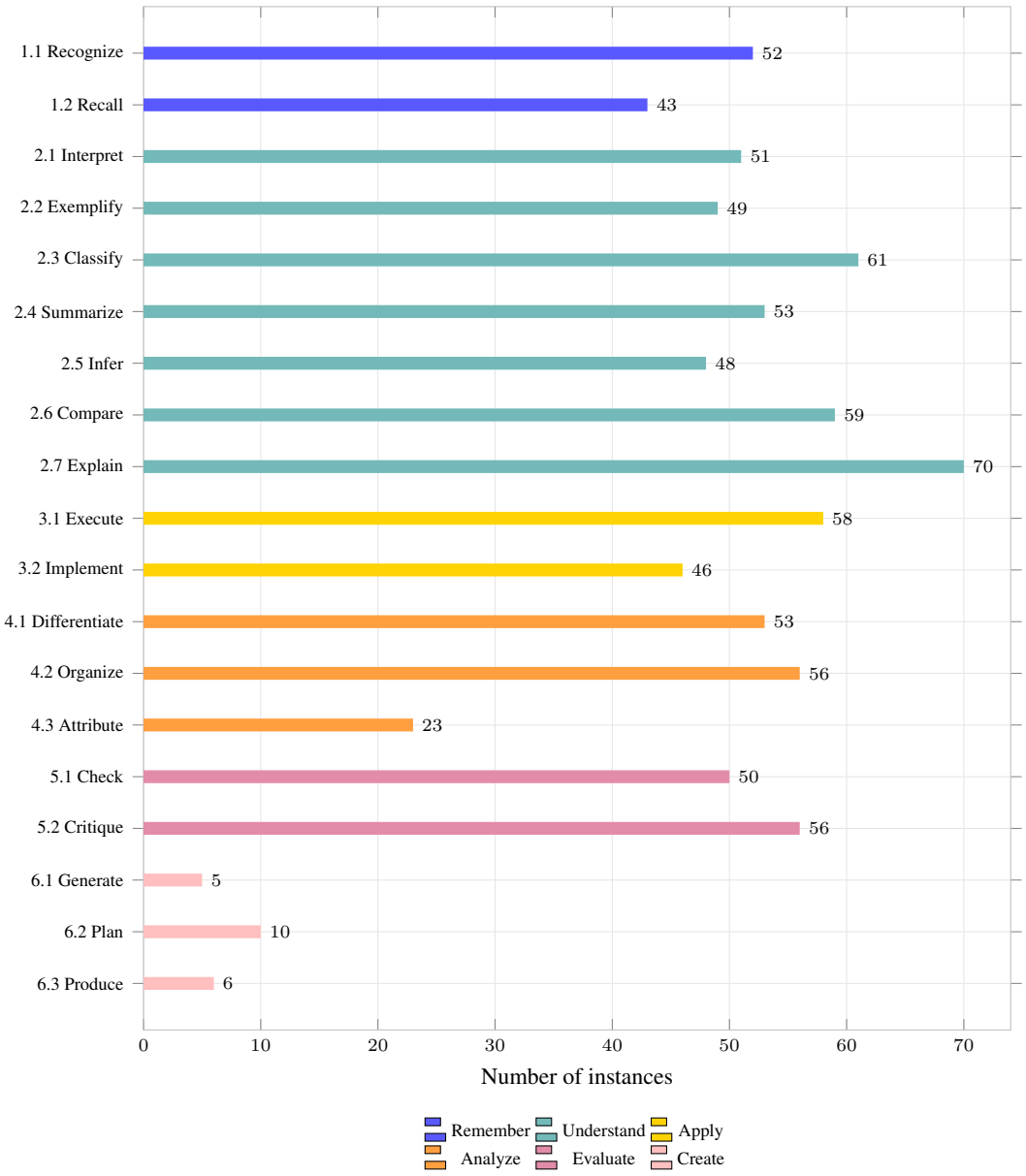


Figure 7: Distribution of the 849 CogEdu-Bench instances across the 19 fine-grained cognitive processes. Colors indicate the six major cognitive categories.

E.2 Benchmark Instance Format

Each benchmark instance separates the model-visible context from the evaluation-only metadata. The evaluated model receives the subject and school-stage metadata, the dialogue history with internal cognitive tags removed, and the learner's final utterance. It does not receive the reference knowledge, the achieved-level label, the target level, the gold action, the rationale, or the ideal response.

The model returns a single JSON object containing `achieved_level`, `action`, and `teacher_response`. This separation prevents direct access to the gold fields while retaining the multi-turn evidence required for cognitive diagnosis and pedagogical decision making. The visibility and evaluation role of each instance field are summarized **(see Table 8)**.

Table 8: CogEdu-Bench instance fields and information visibility.

Field	Visible to model	Used by evaluator	Evaluation role
`meta` (subject/stage)	Yes	Yes	Context
Dialogue history	Yes, tags removed	Yes	Learner evidence
`reference_knowledge`	No	Audit only	Source support
`achieved_level`	No	Yes	**Task 1: major category and minor process gold**
`target_level`	No	SC1 only	Task 2: intended next-process target
`gold_action`	No	Yes	**Task 1: decision gold (*Advance*/*Intervention*)**
`rationale`	No	Audit only	Explanation of the gold decision
`ideal_response`	No	SC6 only	Task 2: response-coverage reference

The reference knowledge and ideal response are evaluation aids rather than information available to the tested tutor. This distinction is especially important when interpreting SC6, because an ideal response may mention content that is absent from the model-visible dialogue.

E.3 Evaluation Tasks and Metrics

CogEdu-Bench evaluates cognitive-process diagnosis, pedagogical decision making, and tutoring-response quality on the same 849 instances.

Task 1: Cognitive-Process Diagnosis. The model predicts one of 19 fine-grained cognitive processes grouped into six major categories. Major-category quadratic-weighted kappa (Major QWK) evaluates prediction across the six broad categories.

Minor QWK evaluates the fine-grained process prediction only on instances for which the corresponding major category is predicted correctly. Because the eligible subset differs across models, Minor QWK is interpreted as a within-category diagnostic rather than a full-set measure of 19-class accuracy.

Conditional Minor QWK is computed separately within each correctly predicted major category and aggregated using sample-count weighting across groups with defined QWK values.

Instances with a missing `achieved_level` field or an unparseable cognitive-level prediction are excluded from Task 1 metric computation.

Predicted cognitive-level labels are parsed from the `achieved_level` field using regular expressions. The parser extracts the leading numeric identifier (e.g., `2.3`) without requiring an exact match of the accompanying process name. The extracted major-category and process identifiers are then used to compute Major QWK and conditional Minor QWK, respectively.

Pedagogical Decision. The model predicts either *Advance* or *Intervention*. Decision accuracy is computed over all 849 benchmark instances. Missing, malformed, or invalid action outputs are counted as incorrect.

Task 2: Tutoring Quality. Teacher responses are evaluated along a factual-content axis containing FC1–FC4 and a scaffolding axis containing SC1–SC6. Each dimension is scored from 0 to 4 and normalized to the interval $[0, 1]$.

For instance i, let F_i and S_i denote the mean normalized scores on the factual-content and scaffolding axes, respectively. The instance-level tutoring-quality score is

$$Q_i = \sqrt{F_i S_i}.$$

The final tutoring-quality score is the mean of Q_i across completed judgments. Judge failures and successful-judge coverage are reported separately.

E.4 LLM-AS-A-JUDGE PROTOCOL

Task 2 uses separate judge calls with restricted field visibility. Candidate responses and dialogue histories are treated as evaluation data, and model thinking traces are never provided to the judge.

Factual-Content Axis. The factual-content axis evaluates whether

- factual statements support the current question (FC1);
- factual statements are correct (FC2);
- the tutor reveals the answer that the learner is expected to provide (FC3);
- the response contains substantive factual content (FC4).

The factual-content judge does not receive `reference_knowledge`.

Scaffolding Axis. The scaffolding axis evaluates

- target-level alignment (SC1);
- reasoning guidance (SC2);
- use of an allowed scaffolding strategy (SC3);
- emotional safety and feedback (SC4);
- pedagogical decision consistency (SC5);
- information coverage relative to the ideal response (SC6).

SC5 Evaluation. SC5 is evaluated in two stages. The judge first infers the instructional intention expressed by the teacher response without seeing the gold action. A separate call then compares the inferred intention with the gold *Advance* or *Intervention* label. SC5 therefore evaluates the pedagogical decision expressed by the response rather than agreement between two explicitly supplied action fields.

SC6 Evaluation. For SC6, the judge first extracts the main concepts and instructional direction from `ideal_response` and then evaluates their coverage in the candidate response. Because an ideal response represents one acceptable instructional path, we additionally examine the sensitivity of aggregate results to removing SC6.

Failure Handling. A judgment is complete only when every required FC and SC stage is present and parseable. If a required stage remains missing after retry, the instance is recorded as a judge failure. Raw outputs, stage-level scores, failures, prompts, and candidate responses are retained for audit.

The reported evaluation uses DeepSeek-V3 (`deepseek-v3`), accessed through Alibaba Cloud Model Studio's DashScope OpenAI-compatible API during August 2026. We use a temperature of 0.0, a maximum output length of 3,200 tokens, and at most three retries.

E.5 ANNOTATION AND JUDGE RELIABILITY

Task-1 Annotation Reliability. We assess the reliability of the Task-1 annotations on 100 benchmark instances. Three subject-matter experts independently assign one of the 19 fine-grained cognitive levels and an *Advance* or *Intervention* decision based on the dialogue evidence, without access to the predefined benchmark labels.

Fleiss' κ is 0.806 for the cognitive-process labels and 0.817 for the pedagogical decisions, with unanimous agreement rates of 74% and 87%, respectively. At least two experts agree on the joint process–decision assignment for 95% of the instances. The experts' exact agreement with the benchmark labels ranges from 0.82 to 0.97 for cognitive levels and from 0.92 to 0.99 for pedagogical decisions. These results support the reliability of the Task-1 reference annotations.

Task-2 Judge Reliability. We audited 100 responses obtained from 25 benchmark contexts covering all 19 cognitive levels, with one response from each of four representative systems per context. Three subject-matter experts independently scored each response on the four factual-content dimensions and six scaffolding dimensions using the same 0–4 rubrics as the automatic evaluator. Model identities and automatic judge scores were hidden during annotation, and the median of the three expert ratings was used as the human consensus score.

Inter-expert reliability was good, with a macro-average ordinal Krippendorff's α of 0.8025 and dimension-level values ranging from 0.6913 to 0.8527. The automatic judge achieved a macro QWK of 0.8051 against the human consensus, with dimension-level QWK values ranging from 0.6812 to 0.8584. At the aggregate Task-2 score level, Spearman's ρ was 0.7229 ($p < 0.001$), with a mean absolute error of 0.0688. These results support the use of the automatic judge on the audited subset and should be interpreted within the scope of the 25 evaluated contexts.

F Evaluation Details

F.1 Models and Baselines

Models. We evaluate three stages of our model: the original Qwen3-4B-Thinking-2507 backbone, the theory-driven SFT initialization, and the final CogRaise model. The specialized tutoring baselines include TutorRL-7B-think and SocraticLM.

We additionally evaluate larger open-weight and closed-source general-purpose models as capability references, including Qwen3-32B, Qwen3-Next-80B-A3B-Thinking, GLM-4.7, GLM-5.2, GPT-4o, and Gemini-3.1-Pro-Preview.

All models are evaluated using the same CogEdu-Bench input information and output requirements. For our models, the same final checkpoint is used across all reported evaluations.

F.2 Prompt Templates

All models received the same frozen Chinese evaluation prompt and the same model-visible instance fields. The prompt asks the model to infer `achieved_level`, select *Advance* or *Intervention*, and generate `teacher_response` in a single JSON object. It specifies the "X.X process-name" format and includes three worked output examples; one original example is reproduced below(**see Table 9**). The prompt does not enumerate all 19 process labels, and no examples or label definitions were added after the reported evaluations. Raw model outputs were retained before parsing.

Table 9: Evaluation prompt: translated output example.

Field	Example value
`achieved_level`	2.5 Infer
`action`	Intervention
`teacher_response`	Can you compare the Truman Doctrine and the Marshall Plan more specifically in terms of their implementation steps and intended recipients?

F.3 Generation Settings

Generation used a temperature of 0.0 and a maximum output length of 2048 tokens. API-served models were queried through provider-specific OpenAI-compatible endpoints, whereas local checkpoints were served through an OpenAI-compatible vLLM endpoint. Requests used a timeout of 180

seconds, up to three attempts per instance, and a concurrency level of five. Reasoning mode was enabled only for models whose serving interfaces supported it. When available, reasoning traces were stored separately and excluded from evaluation. Outputs were parsed into `achieved_level`, `action`, and `teacher_response`, while unparseable responses were retained for debugging and treated as invalid predictions.

G Full Experimental Results

G.1 Full CogEdu-Bench Results

The complete CogEdu-Bench results compare our backbone, SFT checkpoint, and final CogRaise model with reward-component ablations, specialized tutoring models, and general-purpose models**(see Table 10)**.

Table 10: Full CogEdu-Bench results. Higher values are better. Minor QWK is evaluated on instances with a correct major-category prediction. Decision accuracy uses all 849 instances; missing and invalid actions count as incorrect.

Model	**Major QWK** ↑	**Minor QWK** ↑	**Decision Acc.** ↑	**Tutoring Quality** ↑
Our Models				
Qwen3-4B-Thinking-2507	0.2356	0.1435	0.6678	0.4241
Qwen3-4B-Thinking-2507-SFT	0.2924	0.3590	0.6855	0.5687
CogRaise	0.2875	**0.4940**	**0.6961**	0.5686
Reward-Component Ablations				
Advancement only	<u>0.2806</u>	0.3198	0.6855	<u>0.5680</u>
RM only	0.2733	0.3658	<u>0.6914</u>	0.5638
Auxiliary only	0.2527	<u>0.4851</u>	0.6784	0.5601
Specialized Tutoring Models				
TutorRL-7B-think	0.0900	0.0400	0.6396	0.4938
SocraticLM	0.1081	0.0400	0.5735	0.3280
Open-Weight General Models				
Qwen3-32B	0.2615	0.2334	0.6266	0.6029
Qwen3-Next-80B-A3B-Thinking	0.5246	0.2611	0.6302	0.4051
GLM-4.7	0.6301	0.3875	0.6259	0.6380
GLM-5.2	<u>0.6383</u>	0.3689	0.6419	<u>0.6379</u>
Closed-Source Models				
GPT-4o	0.4880	0.2332	0.6643	0.5452
Gemini-3.1-Pro-Preview	**0.8155**	0.3654	0.6455	**0.6485**

Gemini-3.1-Pro-Preview achieves the strongest major-category diagnosis result, with a Major QWK of 0.8155. CogRaise obtains a Major QWK of 0.2875 and a conditional Minor QWK of 0.4940 on 191 instances for which the major category is predicted correctly.

SocraticLM produces valid pedagogical-action outputs for 483 of the 849 instances and makes 277 correct decisions. Its decision accuracy over the full benchmark is therefore 0.3263, whereas its accuracy restricted to valid outputs is 0.5735. We use the full-set value in the main comparison because missing and invalid action outputs are counted as incorrect.

Compared with the SFT checkpoint, CogRaise increases decision accuracy from 0.6855 to 0.6961. Its conditional Minor QWK increases from 0.3590 to 0.4940, while Major QWK changes from 0.2924 to 0.2875. Tutoring quality remains essentially unchanged, changing from 0.5687 to 0.5686.

CogRaise improves fine-grained diagnosis QWK by 37.6% relative to its SFT initialization. The higher conditional Minor QWK should be interpreted as stronger fine-grained discrimination within correctly predicted major categories. Because this metric is evaluated only on the major-correct subset, it does not by itself establish uniformly improved diagnosis over the complete 19-class space.

G.2 Cognitive-Level Breakdown

Performance varies across the six major cognitive categories, with both 4B checkpoints showing their strongest performance on Apply. Within the 4B setting, CogRaise substantially improves fine-grained within-category diagnosis over the SFT checkpoint while preserving major-category recognition and tutoring-response quality. It also achieves higher decision accuracy, demonstrating that reward-guided training strengthens the model's ability to distinguish closely related cognitive processes and make appropriate pedagogical decisions**(see Figure 8)**.

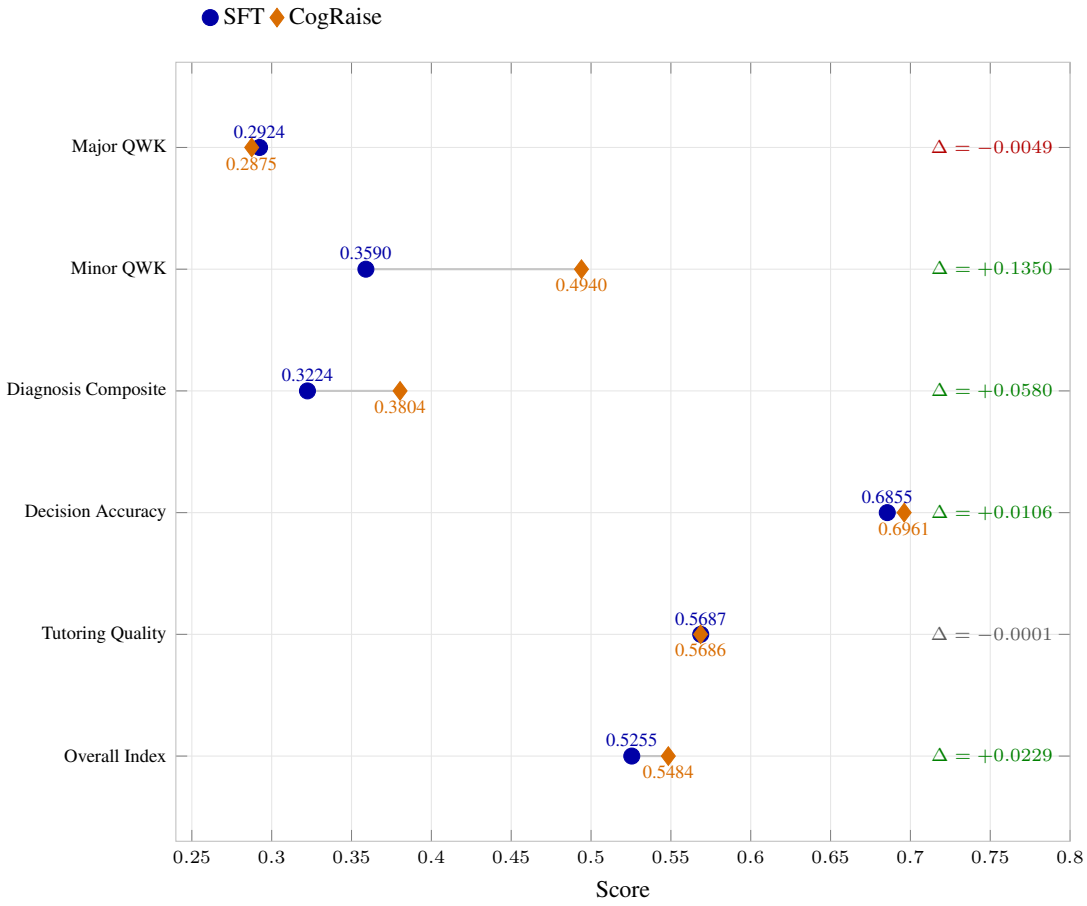


Figure 8: Changes in CogEdu-Bench metrics from the SFT checkpoint to CogRaise. Higher values are better. The diagnosis composite combines Major QWK and Minor QWK. The Overall Index summarizes cognitive diagnosis, decision accuracy, and tutoring-response quality.

G.3 Full MathTutorBench Results

MathTutorBench is used to examine whether cognition-oriented training transfers to an external mathematical tutoring benchmark. CogRaise preserves the SFT checkpoint's problem-solving performance and improves the scaffolding score from 0.4113 to 0.4261. Its pedagogy-following score remains broadly comparable to that of the SFT checkpoint (0.5383 versus 0.5513), indicating that cognition-oriented training can enhance scaffolding without materially disrupting the model's external tutoring capabilities. These results indicate a modest improvement in the scaffolding dimension rather than a uniform improvement across all external tutoring metrics**(see Table 11)**.

Table 11: Complete results on MathTutorBench. ↑ indicates that higher values are better, while → 1 indicates that values closer to 1 are better. The best result in each column is shown in bold, and the second-best distinct result is underlined. Rankings are computed from the unrounded values.

	Math Expertise			Student Understanding							
	Problem Solving	Socratic Questioning		Solution Correctness				Mistake Location			Mistake Correction
Model	Acc. ↑	BLEU ↑	Avg. Q. → 1	Acc. ↑	F1 ↑	Prec. ↑	Recall ↑	Macro F1 ↑	Micro F1 ↑	Weighted F1 ↑	Acc. ↑
Qwen3-4B-Thinking-2507	0.9583	0.2491	**1.0000**	0.8453	0.8404	0.8681	0.8144	0.3894	0.7001	0.7128	0.8752
Qwen3-4B-Thinking-2507-SFT	0.9416	0.2576	1.0030	0.7989	0.7904	0.8252	0.7585	0.1211	0.4910	0.5273	0.7774
CogRaise (Full Model)	0.9409	0.2512	1.0053	0.7924	0.7831	0.8199	0.7495	0.1121	0.4845	0.5196	0.8104
Qwen3-32B	**0.9651**	0.2438	1.0182	0.8817	0.8766	0.9162	0.8403	0.4015	0.7440	0.7529	0.8653
TutorRL-7B-think	0.8582	0.2371	1.0023	0.6472	0.5809	0.7153	0.4890	0.0108	0.2011	0.2433	0.6796
SocraticLM	0.8400	0.2767	2.6687	0.5833	0.3783	0.7449	0.2535	0.0100	0.3802	0.3353	0.2265
GPT-4o	0.8764	0.2933	1.4578	0.7385	0.7171	0.7812	0.6627	0.3643	0.5978	0.6042	0.7655
Gemini-3.1-Pro-Preview	0.8362	**0.3498**	**1.0000**	0.8972	0.8939	0.9234	0.8663	**0.4990**	**0.8039**	**0.8048**	**0.9521**
GLM-5.2	0.4011	0.2982	**1.0000**	**0.9062**	**0.9039**	**0.9266**	**0.8822**	0.4876	0.7904	0.7990	0.9212

	Pedagogy Following							
	Standard				Hard			
Model	Match ↑	Win Rate ↑	Score ↑	Mean Margin ↑	Match ↑	Win Rate ↑	Score ↑	Mean Margin ↑
Qwen3-4B-Thinking-2507	0.9522	0.5791	5.6750	0.8860	0.8991	0.6024	5.2550	0.8320
Qwen3-4B-Thinking-2507-SFT	0.7235	0.5513	4.7560	-0.0340	0.6820	0.5199	3.7820	-0.6410
CogRaise (Full Model)	0.7191	0.5383	4.6910	-0.0990	0.6911	0.4740	3.8440	-0.5790
Qwen3-32B	0.8835	0.5913	5.5800	0.7910	0.8012	0.4465	3.1550	-1.2690
TutorRL-7B-think	0.9078	0.6183	5.7630	0.9740	0.8440	0.6300	5.7200	1.2970
SocraticLM	0.8739	0.3965	3.0870	-1.7030	0.8257	0.3517	0.5700	-3.8530
GPT-4o	0.9809	**0.7957**	**7.6370**	**2.8480**	0.9480	**0.7615**	**7.1580**	**2.7350**
Gemini-3.1-Pro-Preview	**0.9896**	0.7609	7.1125	2.3230	**0.9786**	0.7401	7.0427	2.6195
GLM-5.2	0.9626	0.7243	6.8079	2.0184	0.9633	0.7278	6.8785	2.4554

	Scaffolding Generation							
	Standard				Hard			
Model	Match ↑	Win Rate ↑	Score ↑	Mean Margin ↑	Match ↑	Win Rate ↑	Score ↑	Mean Margin ↑
Qwen3-4B-Thinking-2507	0.4557	0.2557	2.1330	-2.6560	0.4343	0.1651	-0.6590	-5.0820
Qwen3-4B-Thinking-2507-SFT	0.4670	0.4113	3.5250	-1.2650	0.4618	0.3823	2.0030	-2.4200
CogRaise (Full Model)	0.4835	0.4261	3.4830	-1.3060	0.4801	0.3517	2.1680	-2.2550
Qwen3-32B	0.5165	0.2374	2.1770	-2.6130	0.4679	0.1804	0.0560	-4.3670
TutorRL-7B-think	0.4496	0.4174	3.0420	-1.7470	0.5199	0.4709	3.0862	-1.3370
SocraticLM	0.8643	0.3391	2.1990	-2.5900	0.8349	0.3089	-0.2920	-4.7150
GPT-4o	0.5504	0.4322	3.9600	-0.8290	0.5474	0.3853	2.5840	-1.8390
Gemini-3.1-Pro-Preview	**0.8765**	**0.7565**	**7.2549**	**2.4654**	**0.9083**	**0.7462**	**7.1198**	**2.6967**
GLM-5.2	0.7626	0.6087	5.6259	0.8364	0.7920	0.5688	4.5525	0.1294

G.4 General Reasoning Results

We use general-reasoning evaluation to examine capability retention during specialization. On MMLU-Pro, the original Qwen3-4B-Thinking-2507 backbone scores 0.7378, the SFT checkpoint scores 0.6951, and CogRaise scores 0.6934**(see Table 12)**.

The difference between the SFT checkpoint and CogRaise is small, suggesting that the reinforcement-learning stage largely preserves the general-reasoning ability of the supervised checkpoint. However, both specialized checkpoints remain below the original backbone. The result should therefore not be interpreted as complete preservation across the entire specialization pipeline.

Table 12: General-reasoning results on MATH-500 ($N = 500$), GSM8K ($N = 1{,}319$), and MMLU-Pro ($N = 12{,}032$). Point estimates are reported for readability. Higher values are better.

	MATH-500	GSM8K		MMLU-Pro	
Model	Acc. ↑	Strict EM ↑	Flexible EM ↑	Acc. ↑	Macro Avg. ↑
Our Models					
Qwen3-4B-Thinking-2507	.792	**.870**	.628	**.7378**	**.7228**
Qwen3-4B-Thinking-2507-SFT	.892	.845	.848	.6951	.6825
CogRaise	**.894**	.848	**.851**	.6934	.6791
Specialized Tutoring Models					
TutorRL-7B-think	.746	.766	.820	.5501	.5604
SocraticLM	.810	.764	.790	.3479	.3327

H Additional Analyses

H.1 Reward Model Reliability

We evaluate the pointwise and hybrid reward models on a fully held-out set of 156 preference pairs satisfying the same 0.10 preference-margin criterion used for hybrid training. Under the reward-scoring procedure used during reinforcement learning, the hybrid model correctly ranks 143 of the 156 pairs, achieving an accuracy of 91.67% with a 95% confidence interval of $[0.863, 0.951]$. In comparison, the pointwise model correctly ranks 129 pairs, achieving an accuracy of 82.69% with a 95% confidence interval of $[0.760, 0.878]$. An exact McNemar test confirms that the difference is statistically significant ($p = 0.0013$). Routing accuracy is 1.000 for both checkpoints, indicating that the improvement is attributable to stronger reward discrimination rather than changes in routing behavior.

H.2 Cognitive Advancement Analysis

We compare the cognitive levels addressed by responses generated by GRPO + RM and the full CogRaise model. To obtain a matched comparison, we retain only queries for which the frozen cognitive-level classifier provides a confident prediction for both systems. On the resulting shared subset of 96 queries, CogRaise increases the proportion of responses targeting the immediate next cognitive level from 17.71% to 21.88%. It also reduces backward offsets from 37.50% to 36.46% and large forward deviations from 23.96% to 21.88%**(see Figure 9)**. These changes are consistent with the intended role of the cognitive-advancement reward: it encourages the model to select a more appropriate next cognitive target while reducing undesirable backward and excessive forward shifts.

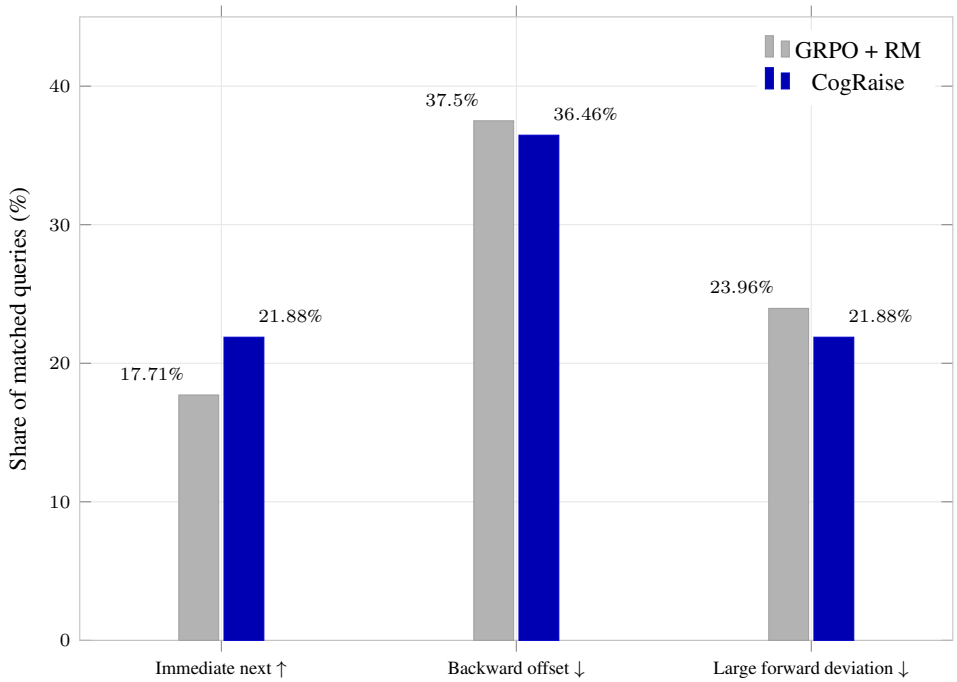


Figure 9: Cognitive-offset outcomes on the 96 matched queries for which the frozen classifier provides confident predictions for both GRPO + RM and CogRaise. Higher immediate-next rates are preferable, whereas lower backward-offset and large-forward-deviation rates are preferable.

H.3 Extended Ablations

Single-Component Ablations. We compare GRPO runs using only the auxiliary rewards, only the reward-model score, or only the cognitive-advancement reward with the full CogRaise objective at checkpoint 300**(see Table 13)**.

CogRaise achieves the highest point estimate among the four variants on all reported metrics: 0.2875 Major QWK, 0.4940 conditional Minor QWK, 0.6961 decision accuracy, and 0.5686 tutoring quality.

The auxiliary-only variant is particularly informative. Its conditional Minor QWK reaches 0.4851, close to the full model's 0.4940. However, it has the lowest point estimate among the four variants for Major QWK (0.2527), decision accuracy (0.6784), and tutoring quality (0.5601). Its strong conditional Minor QWK therefore does not imply equally strong performance across the benchmark's other objectives.

The principal advantage of the full reward configuration is that it retains strong fine-grained within-category diagnosis while performing better on the other three measures. Neither the reward-model-only nor the advancement-only variant matches this joint performance profile. The ablation therefore supports the full reward configuration for the combined objectives of cognitive diagnosis, pedagogical decision making, and tutoring-response quality.

The eligible subset for conditional Minor QWK differs across variants, ranging from 184 to 205 instances. Accordingly, the results support the complementarity of the reward components but do not by themselves establish statistical interaction effects. Multi-seed experiments would be required for that stronger claim.

Table 13: Single-component ablation results at checkpoint 300. Minor QWK is reported together with the number of eligible major-correct instances in parentheses. Higher values are better.

Model	Major QWK↑	Minor QWK↑	Decision Acc.↑	Tutoring Quality↑
GRPO+Aux.	.2527	.4851	.6784	.5601
GRPO+RM	.2733	.3658	.6914	.5638
GRPO+R_{adv}	.2806	.3198	.6855	.5680
CogRaise	**.2875**	**.4940**	**.6961**	**.5686**

I QUALITATIVE EXAMPLE

Correct Cognitive Transition. In instance BT-HUM-HIS-HS-T1-0055, the learner explains how color, texture, decoration, shape, and foot-ring characteristics can be used to distinguish several types of Chinese porcelain. The gold cognitive-process label is 3.1 Execute, with an Advance decision toward 3.2 Implement. CogRaise correctly identifies 3.1 Execute and asks the learner to apply the identified criteria to a mixed collection of real objects. The response preserves the original reasoning task while extending it from a familiar description to a less structured application setting. This case demonstrates CogRaise's ability to jointly perform cognitive diagnosis, select an appropriate pedagogical decision, and generate a response aligned with the next cognitive target.