

MetaCLASS: Metacognitive Coaching for Learning with Adaptive Self-regulation Support

Naiming Liu
Rice University
nl35@rice.edu

Richard Baraniuk
Rice University
richb@rice.edu

Shashank Sonkar
University of Central Florida
shashank.sonkar@ucf.edu

Abstract

Large language models can generate fluent explanations, but effective tutoring requires supporting the learner’s *thought process*, not just delivering content. Metacognitive tutoring targets this gap by prompting *planning*, *monitoring*, *debugging*, and *evaluation*, and, crucially, deciding when to be active versus minimally present, based on learner signals and trajectory. We introduce **MetaCLASS**, a learning-science grounded framework that formulates metacognitive tutoring as *move selection* over 11 interpretable actions aligned to self-regulated learning (SRL) processes. MetaCLASS uses a two-phase procedure that first plans a pedagogical trajectory conditioned on learner profiles (calibration, help-seeking) and then generates natural dialogue consistent with that plan. This process yields 1,015 conversations spanning 7,711 turns, each annotated with turn-level metacognitive labels and supported by comprehensive validation analyses. We benchmark nine open-source LLMs on predicting the next coach move given a question and dialogue context. The best model achieves only 43.2% accuracy in coach move prediction, and models exhibit *compulsive intervention bias*: in turns where effective metacognitive tutoring requires silence (41.7% of cases), models predict “no intervention” only 6.6% of the time, while severely over-predicting high-intervention moves. These results show that traditional content-based tutoring ability does not translate to metacognitive tutoring competence, positioning MetaCLASS as a testbed for developing tutors that promote SRL¹.

1 Introduction

Modern AI tutoring systems can generate accurate explanations across domains, but often fail at a core tutoring skill: supporting the learner’s *thought process*. The failure is not conceptual, but pedagogical.

Good tutoring is not always the same as delivering correct content. It is the art of steering a learner’s *process*: clarifying goals, choosing strategies, monitoring understanding, recovering from confusion, and reflecting to improve. Importantly, tutoring is not always action. Some of the most effective moments in tutoring are moments of *restraint*: letting a learner struggle productively rather than interrupting with a hint (Kapur, 2008). Human tutors do this naturally by prompting students to articulate what they know, when they are not sure, and how they would verify a step (Chi et al., 2001). A content tutor says “use $F = mg$ ”, while a metacognitive tutor asks: “What are you trying to find?” or “What would convince you this step is correct?”

These teachable process skills fall under **metacognition** and **self-regulated learning (SRL)** (Flavell, 1979; Zimmerman, 2002). In educational psychology, metacognitive regulation comprises four key components: **Planning** (setting goals and selecting strategies before acting), **Monitoring** (tracking comprehension and progress during work and surfacing uncertainty), **Debugging** (repairing detected problems through alternative strategies or resources), and **Evaluation** (reflecting on progress and supporting transfer to new problems).

Most LLM-based tutoring systems implicitly optimize for *helpfulness-as-output*: generating the next plausible hint or response under instructions (scaffolding, socratic questioning, mistake correction), or selecting among high-level pedagogical behaviors (Macina et al., 2025; Sonkar et al., 2023, 2024). This can obscure the core tutoring decision: *what should a tutor do right now, given the learner’s signals and trajectory*. Progress is bottlenecked by the lack of (1) a shared, interpretable *action space* for metacognitive tutoring; (2) scalable supervision with validity checks for those actions; (3) a benchmark that directly tests metacognitive tutoring as *decision-making under pedagogy*.

MetaCLASS reframes “AI tutoring” from con-

¹Dataset and code is available [here](#).

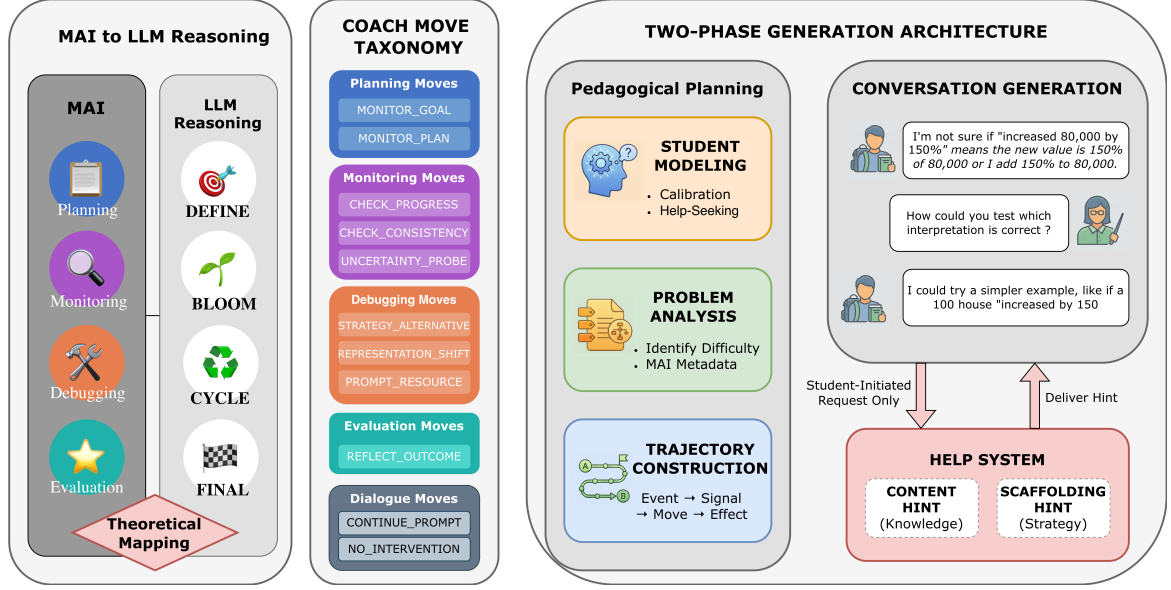


Figure 1: An overview of MetaCLASS Framework.

tent delivery to *coaching the learning process*. We operationalize these metacognitive processes as an interpretable *action space* of coach moves (Table 1), treating `no_intervention` as a first-class action to formalize restraint. Additionally, by turning metacognitive tutoring into a problem of *pedagogical action selection*, we make an explicit coaching decision of which metacognitive move to take next, given the learner’s signals and trajectory.

In this paper, we make **three key contributions**:

1. **MetaCLASS Framework.** We introduce the first learning-science grounded framework for LLM-based metacognitive tutoring that targets self-regulated learning. We derive **11 interpretable coach moves** grounded in the Metacognitive Awareness Inventory (MAI), and map them systematically to LLMs’ internal reasoning steps. The framework treats `no_intervention` as a first-class pedagogical action.
2. **MetaCLASS Dataset and Analysis.** We generate **1,015** metacognitive coaching conversations spanning **7,711** turns across GSM8K, MATH, and AIME datasets, annotated with turn-level coach move labels and learner metadata (calibration, help-seeking), alongside comprehensive dataset analyses.
3. **Benchmark and Findings.** We introduce the **Coach Move Prediction** task which predicts the next metacognitive coach move given a problem and dialogue context. We observe

a **compulsive intervention bias**, where models over-produce high-intervention moves and severely under-predict `no_intervention`.

2 MetaCLASS Framework

In this section, we first ground the framework by mapping MAI theory to LLM reasoning patterns, deriving 11 interpretable coach moves. We then detail the two-phase architecture used to generate contingent, metacognitive coaching dialogues (Framework in Figure 1, Example in Section D).

2.1 Connecting MAI to LLMs Reasoning

The generation pipeline takes an LLM reasoning chain as input and produces a tutoring conversation with ground-truth coach move labels. We first review MAI’s regulatory factors, then show how LLM reasoning chains map onto them, providing the principled basis for trajectory construction in the generation pipeline.

2.1.1 Metacognitive Awareness Inventory

MetaCLASS framework is grounded in the Metacognitive Awareness Inventory (Schraw and Dennison, 1994), which characterizes how learners control their cognitive processes during learning through four empirically validated factors.

- **Planning** includes pre-task activities, such as goal setting, strategy selection, and resource allocation (e.g., I set specific goals before I begin a task).

MAI	Move	Function	Example Response
Planning	MONITOR_GOAL (MG)	Elicit goal awareness	“What are you trying to find?”
	MONITOR_PLAN (MP)	Elicit strategy awareness	“What’s your approach?”
Monitoring	CHECK_PROGRESS (CP)	Prompt progress tracking	“How’s it going?”
	CHECK_CONSISTENCY (CC)	Surface contradictions	“Does that fit with what you said?”
	UNCERTAINTY_PROBE (UP)	Localize confusion	“What’s making you hesitate?”
Debugging	STRATEGY_ALTERNATIVE (SA)	Invite different approach	“What else could you try?”
	REPRESENTATION_SHIFT (RS)	Suggest re-framing	“Would a diagram help?”
	PROMPT_RESOURCE (PR)	Prompt help-seeking awareness	“What would help you move forward?”
Evaluation	REFLECT_OUTCOME (RO)	Prompt retrospective reflection	“What worked here?”
Dialogue	CONTINUE_PROMPT (CT)	Neutral nudge	“Keep going.”
	NO_INTERVENTION (NI)	Preserve productive flow	(silence)

Table 1: MetaCLASS Coach Move Taxonomy: 11 interpretable coach moves mapped to MAI regulatory processes. The moves support metacognitive regulation rather than directly delivering content.

- **Monitoring** captures ongoing awareness of comprehension during learning, enabling detection of confusion, impasses, or inconsistencies (e.g., I find myself pausing regularly to check my comprehension).
- **Debugging** encompasses error correction strategies triggered when Monitoring detects problems (e.g., I re-evaluate my assumptions when I get confused).
- **Evaluation** involves post-task reflection on performance and strategy effectiveness, supporting transfer to future tasks (e.g., I ask myself how well I accomplished my goals once I’m finished).

These four factors form a validated taxonomy of metacognitive regulation, which MetaCLASS uses as a principled basis for designing coach moves.

2.1.2 From LLMs Reasoning Chains to Coach Moves

MetaCLASS uses LLM reasoning chains as think-aloud scaffolds that capture the structured problem-solving process the student works through. To assign coach moves at each point in a conversation, we need this process to have a consistent structure. Recent work (Marjanović et al., 2025) analyzes reasoning chains from DeepSeek-R1 (Guo et al., 2025) and identifies four consistent phases: **DEFINE** (reformulate the problem and identify goals), **BLOOM** (decompose and execute initial steps), **CYCLE** (reconsider assumptions and detect errors through self-monitoring), and **FINAL** (commit to an answer with confidence assessment).

These phases map systematically onto MAI’s regulatory factors, providing the mechanism by

which trajectories are constructed in the generation pipeline:

- **DEFINE** → **Planning**: Both set goals before execution.
- **BLOOM** → **Planning**: Both decompose the problem and select a strategy.
- **CYCLE** → **Monitoring + Debugging**: Both involve self-monitoring and adjusting strategies when stuck.
- **FINAL** → **Evaluation**: Both assess performance and reflect on the outcome.

This mapping has three key implications. First, it grounds LLM reasoning in validated learning science. Second, it enables principled coach design: each MAI factor prescribes interventions for the corresponding phase. Third, it positions MetaCLASS coach moves as external scaffolding of LLMs’ internal metacognitive processes.

2.2 Metacognitive Coach Moves

We operationalize MAI’s four regulatory factors into eleven coach moves (Table 1). Each move translates an internal metacognitive process into an external prompt. Critically, these moves prompt reflection on the problem-solving process itself, which should never provide domain knowledge, evaluate correctness, or reveal solutions. A special coach move is PROMPT_RESOURCE (PR), derived from help-seeking research (Roll et al., 2011), which could give hints, but only when students initiate requests. Additionally, two dialogue moves support natural flow: CONTINUE_PROMPT (CT) provides neutral nudges (“Keep going”), while

NO_INTERVENTION (NI) preserves productive struggle through silence.

2.3 Two-Phase Generation Architecture

MetaCLASS uses a two-phase architecture. Phase 1 instantiates a student model, analyzes the problem for learning obstacles, and integrates this information with the LLM reasoning chain and coach move taxonomy to construct a pedagogical trajectory. Phase 2 conditions dialogue generation on the student model and this trajectory, grounding every coach move in a planned pedagogical event.

2.3.1 Phase 1: Pedagogical Planning

Student Modeling. Effective coaching requires adapting to individual differences. MetaCLASS models students along two orthogonal dimensions (Calibration, Help-Seeking).

Calibration refers to the accuracy of self-assessment: whether students' perceived understanding matches their actual understanding (Kruger and Dunning, 1999).

- **Over-Confident** students overestimate their understanding, using confident language ("obviously," "just") and dismissing challenges without detecting errors.
- **Under-Confident** students doubt correct reasoning, hedging constantly ("maybe," "I'm not sure") and hesitating to commit to answers.
- **Well-Calibrated** students accurately assess understanding, articulating specific confusions ("I understand X but I'm stuck on Y").

Help-Seeking captures patterns of resource utilization during learning (Aleven et al., 2006; Roll et al., 2011).

- **Avoidant** students resist asking for help even when stuck, and view help requests as failure and persist unproductively. They are primary targets for resource prompts.
- **Executive** students use resources strategically, attempting problems first and integrating help into understanding. They are the target pattern for self-regulated learning.
- **Dependent** students request help prematurely and broadly, seeking answers rather than understanding. They should not immediately receive resource prompts.

Crossing these dimensions yields nine theoretical profiles, of which **eight are valid** (overconfident + executive is contradictory). Each profile demands different coaching strategies.

Problem Analysis. Each problem is analyzed to identify potential learning obstacles. *Knowledge Gaps* (missing domain knowledge) are addressed through content hints; *Strategy Gaps* (knowledge present but cannot apply) are addressed through scaffolding hints; *Monitoring Gaps* (unnoticed errors) and *Execution Gaps* (careless mistakes) are addressed through prompts that highlight inconsistency or direct attention to verification without revealing answers. Both content and scaffolding hints are also pre-generated as problem-specific resources during planning but delivered only when students explicitly request them.

For each problem, we also generate metacognitive support structures aligned with the MAI framework (planning, monitoring, debugging, and evaluation support). This structured decomposition serves as the pedagogical foundation for trajectory construction.

Trajectory Construction. The core of planning is the trajectory, a sequence of pedagogical events that serves as the causal backbone of the conversation. Each event specifies a four-part chain:

- **Event:** What happens in the problem-solving process (e.g., student encounters an ambiguity, gets stuck on a concept)
- **Signal:** What observable behavior the student produces (e.g., hedging language, long pause)
- **Move:** What coach intervention is appropriate given the student's profile (e.g., uncertainty probe, consistency check)
- **Effect:** What change should result from the intervention (e.g., student articulates confusion, requests help)

2.3.2 Phase 2: Dialogue Generation

This phase transforms the established LLM reasoning and trajectory into a coherent student-coach dialogue. The generation process strictly enforces the pedagogical plan: students exhibit behaviors consistent with their assigned profile, encounter the anticipated learning obstacles, and progress through the trajectory's event sequence.

Dataset	# Conv	# Turn	Contingent	Trajectory
GSM8K	496	3,180	98.6%	99.4%
MATH	490	4,142	96.2%	100%
AIME	29	389	97.3%	100%
Total / Avg	1,015	7,711	97.3%	99.8%

Table 2: MetaCLASS dataset statistics and evaluation results for contingent scaffolding and trajectory following.

Help System Design The help system models student-initiated resource use. Unlike traditional tutoring systems where hints are offered by the tutor, MetaCLASS generates hints during planning but delivers them only when students explicitly request help.

Hint Types. *Content hints* provide domain knowledge (formulas, facts) for Knowledge Gaps, while *scaffolding hints* provide process guidance (strategies, approaches) for Strategy Gaps.

Help Quality. We evaluate the appropriateness of help requests: *appropriate* (asked when genuinely stuck, chose correct type), *premature* (asked without real attempt), *delayed* (asked too late after excessive struggle), or *mismatched* (asked for wrong hint type).

3 MetaCLASS Dataset

3.1 Dataset Statistics

Following MetaCLASS framework, we generate student-coach conversations with GPT-5.1 (Achiam et al., 2023) model, across three math datasets of varying difficulty: GSM8K (grade school math word problems) (Cobbe et al., 2021), MATH (high school competition) (Hendrycks et al., 2021), and AIME (advanced olympiad) (Mathematical Association of America, 2024). As shown in Table 2, the dataset comprises 1,015 conversations totaling 7,711 turns. Examples of generated trajectories and conversations can be found in Appendix D.

3.2 Dataset Validation

We validate the MetaCLASS dataset through nine complementary analyses grounded in learning science research, examining whether generated conversations exhibit the structural and pedagogical properties of effective tutoring. We present three core analyses below; six additional analyses (MAI factor coverage, move sequence structure, student voice authenticity, help-seeking trajectories, internal field coherence, and post-hint coaching se-

quences) appear in Appendix C, along with detailed methodology and per-dataset breakdowns.

Contingent Scaffolding. Contingent scaffolding assesses whether the coach adapts to the learner’s observed state rather than following a fixed script (Wood et al., 1976). We identify difficulty signals while excluding productive exploration, then test whether the coach’s next move is appropriate for the conversation phase (Appendix C.1). Across 1,890 signals in 828 conversations, coaches select appropriate moves 97.4% of the time (Table 2), with 92–100% rates across phases and datasets. Most remaining cases substitute one reasonable move for another rather than making pedagogically inappropriate choices.

Profile-Differentiated Coaching. A main claim of MetaCLASS is that the coach adapts to individual student profiles. We validate this by examining intervention timing, move selection, and student behavior across all three datasets.

Intervention Timing. Table 3 shows the Time-to-First-Intervention (TTFI), the number of student turns before the coach’s first active move. Across all datasets, avoidant students receive later intervention than dependent students. This differentiation respects Kapur’s productive struggle principle (Kapur, 2008): avoidant students benefit from more struggle time to build help-seeking confidence, while dependent students benefit from earlier scaffolding.

External Help Usage. Beyond coach behavior, student behavior also differs systematically by profile. Aggregating across datasets, Dependent students invoke external help (the SYSTEM speaker) most frequently (97.1% of conversations), Executive students at intermediate rates (80.0%), and Avoidant students least often (63.1%). The 34-point gap between Dependent and Avoidant students confirms that help-seeking profiles produce meaningfully different student behaviors, not just different coach responses.

Trajectory Alignment. We assess trajectory-execution fidelity by comparing planned and realized coach moves. As shown in Table 2, 99.8% of conversations follow the planned trajectory. Across datasets, trajectories average 4.3 planned moves spanning 4.1 unique move types, indicating alignment with nontrivial pedagogical plans. The three misaligned GSM8K cases reflect early termination with REFLECT_OUTCOME when students solved the

Help-Seeking	GSM8K	MATH	AIME
Avoidant	1.6–2.3	1.4–1.7	1.5–2.7
Executive	1.1–1.5	1.1	1.0
Dependent	1.1–1.3	1.1–1.2	1.0–1.2

Table 3: Time-to-First-Intervention (TTFI) across calibration sub-profiles, grouped by help-seeking type and dataset.

problem sooner than expected. As a complementary check, two independent domain experts annotated 30 randomly sampled conversations (10 per dataset) and both agreed that all sampled conversations align with their planned trajectories.

4 Coach Move Prediction Benchmark

4.1 Task Definition

We define **Coach Move Prediction** as inference with a prompted, fixed LLM: given a math problem and the tutoring dialogue up to the current student turn, the model must select the next metacognitive coaching action. Let $x = (P, H_t)$ denote the input, where P is the math problem and $H_t = \{(u_1, m_1, r_1), \dots, (u_{t-1}, m_{t-1}, r_{t-1}), u_t\}$ is the dialogue history up to the current student utterance u_t . Here u_i is the student utterance at turn i , m_i is the coach move label, and r_i is the coach response text.

The model outputs *structured text* containing (i) a discrete coach move m_t and (ii) an optional natural-language coach response r_t . We evaluate only the move decision: the predicted move $\hat{m}_t$ is extracted by parsing the model output and must belong to a finite action space $\mathcal{M}$ of 11 metacognitive coach moves (Table 1).

Conceptually, prompting and decoding induce an implicit distribution over moves, and we decode a single prediction:

$$\hat{m}_t = \arg \max_{m \in \mathcal{M}} P_\theta(m | x), \quad (1)$$

where P_θ is the distribution of a fixed pretrained model under our prompt template and decoding procedure. We report accuracy of $\hat{m}_t$ against the ground-truth move m_t specified by the MetaCLASS pedagogical trajectory. If the model outputs an invalid or unparseable label (not in $\mathcal{M}$), we count the prediction as incorrect.

4.2 Models and Evaluations

Models. We evaluate nine open-source LLMs spanning four families. Large models include

Model	GSM8K		MATH		AIME	
	Full	Min	Full	Min	Full	Min
<i>Large Models ($\geq 70B$)</i>						
GPT-oss-120b	36.2	32.9	30.8	27.1	25.2	19.3
Qwen3-80b-Think	43.2	32.3	37.5	27.6	29.6	20.3
Qwen3-80b-Inst	36.2	29.7	30.4	26.4	22.1	24.7
LLaMA-70b	36.0	29.3	30.1	23.0	22.4	18.3
<i>Small Models ($\leq 20B$)</i>						
GPT-oss-20b	34.5	29.8	29.3	23.8	21.9	17.5
Phi4-14b	33.7	25.1	29.1	21.3	21.1	18.8
LLaMA-8b	28.5	20.8	25.9	17.9	15.7	14.4
Qwen3-4b	34.4	5.1	28.4	5.9	22.6	6.4
Phi4-4b	27.0	19.8	22.3	16.8	13.1	9.8

Table 4: Model performance on Coach Move Prediction across three mathematical reasoning datasets using full prompt versus minimal prompts. Best accuracy per column is in **bold**.

GPT-oss-120B (Agarwal et al., 2025), Qwen3-80B-Instruct, Think (Yang et al., 2025), and LLaMA-3.3-70B (Touvron et al., 2023). Small models include GPT-oss-20B, Qwen3-4B, Phi-4-14B (Abdin et al., 2024), Phi-4-4B, and LLaMA-3.1-8B. We focus on general-purpose LLMs because no existing model is purpose-built for metacognitive coaching with an explicit NO_INTERVENTION action. We use hyperparameters following model card recommendations (see Appendix B).

Prompting Strategy. We evaluate two prompt conditions (see Appendix E). The comparison tests a diagnostic hypothesis: do LLMs possess internalized metacognitive coaching ability, or do they require explicit instruction? *Full prompt* provides (1) definitions and examples for all 11 coach moves, (2) move selection guidelines specifying when each is appropriate, and (3) coaching rules emphasizing metacognitive support over content. *Minimal prompt* provides only move names without definitions, examples, or guidelines.

Evaluation Metrics. Our primary evaluation is *Accuracy*, reported both overall and per-move. We also report *NI Detection rate* for assessing whether LLMs can support students with appropriate silent restraints.

5 Results and Discussion

Our analysis reveals three fundamental failures distinguishing metacognitive coaching from regular tutoring: (1) models generally struggle with metacognitive move prediction; (2) models exhibit compulsive intervention bias that explicit prompt

Model	MG	MP	CP	CC	UP	SA	RS	PR	RO	CT	NI	Avg
<i>Large Models ($\geq 70B$)</i>												
gpt-oss-120b	23.0	60.5	33.1	31.7	69.6	23.8	33.0	9.0	98.6	15.4	4.5	32.7
Qwen3-80b-Think	10.1	54.4	5.4	33.6	67.2	12.6	7.2	8.7	98.1	0.0	29.0	39.5
Qwen3-80b-Inst	51.8	38.1	19.2	33.0	71.3	12.1	8.2	37.5	99.6	0.0	5.7	32.4
LLaMA-70b	11.3	63.4	19.7	16.7	71.5	34.4	7.2	23.3	99.7	7.7	6.9	32.1
<i>Small Models ($\leq 20B$)</i>												
gpt-oss-20b	14.7	59.6	17.6	23.6	80.8	19.1	6.2	11.8	98.0	0.0	5.0	31.0
Phi4-14b	32.8	55.3	15.3	46.5	36.4	38.8	1.0	24.3	99.2	7.7	1.4	30.6
LLaMA-8b	18.1	46.8	23.2	28.6	64.9	22.1	3.1	15.3	79.9	7.7	2.0	26.4
Qwen3-4b	66.0	25.2	4.0	45.1	72.5	25.9	15.5	35.8	98.4	0.0	0.3	30.6
Phi4-4b	67.5	22.9	18.8	30.6	30.9	1.2	0.0	6.2	93.6	0.0	0.1	23.8

Table 5: Per-move prediction accuracy (%) for the 11 coach moves (Full prompt and aggregate across all datasets.)

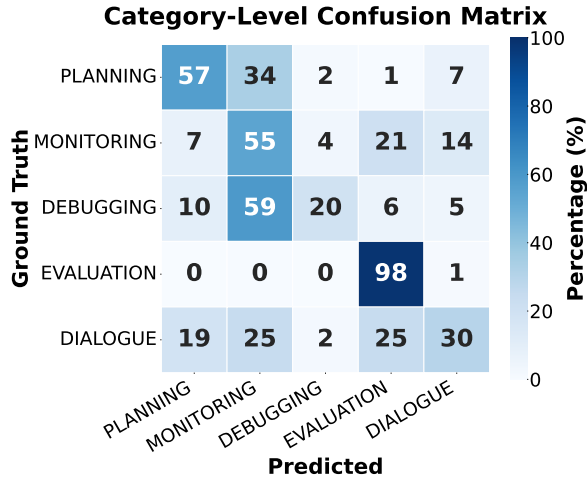


Figure 2: Category-level confusion matrix for Qwen3-80b-Think. DEBUGGING moves are predominantly misclassified as MONITORING (59%), suggesting the model struggles to distinguish exploratory probing from corrective intervention.

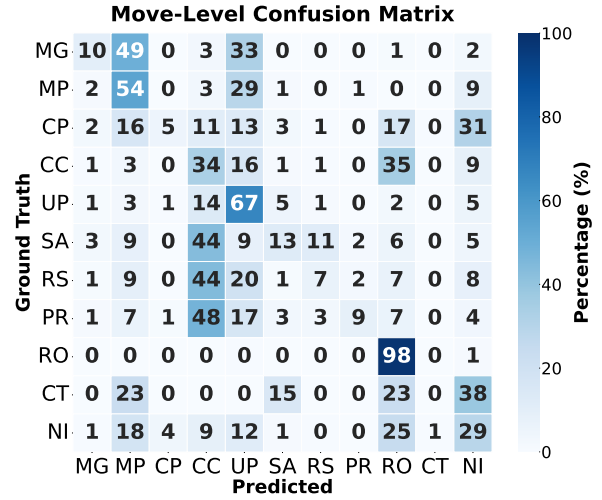


Figure 3: Move-level confusion matrix for Qwen3-80b-Think. NI reaches only 29% accuracy, despite comprising 42% of ground truth, revealing compulsive intervention bias.

instruction cannot override; and (3) models tend to favor consistency checks over strategic redirection.

5.1 Performance Across Models and Prompts

Table 4 shows overall accuracy for coach move prediction. Model accuracy plateaus at 43.2%, revealing that open-source LLMs still struggle to identify when and how to intervene “metacognitively”. This contrasts with their strong performance on content-based tutoring (Scarlato et al., 2025; Sonkar et al., 2023), showing that knowing how to explain does not translate to knowing when to intervene. While large models generally achieve higher accuracy, smaller models are inconsistent. For instance, Qwen3-4B occasionally matches larger models yet exhibits catastrophic drops under certain configurations (5.1% on GSM8K with Minimal prompt).

Impact of Context (Full vs Minimal). Full prompts with explicit MAI definitions consistently outperform Minimal prompts. Qwen3-4B shows this dependency most starkly, with the performance dropping from 34.4% to 5.1% without explicit instruction. This reveals LLMs lack internalized metacognitive tutoring ability and cannot autonomously apply coaching strategies without explicit prompting.

5.2 Move-level Analysis

Accuracy Variation. Accuracy varies dramatically across moves (Table 5): near-perfect on RO ($\sim 99\%$) versus near-zero on NI ($\leq 29.0\%$) and CT ($\leq 15.4\%$). High-accuracy moves follow clear surface patterns (RO at completion, UP after hesitation, MP after problem statements). Mid-accuracy moves ($CC \leq 59.5\%$, $MP \leq 68.0\%$) show high model-

Rank	Move	Acc (%)	N	GT %	Pred %	Bias
1	NI	11.5	12,876	41.7	6.6	-35.2
2	SA	20.7	1,360	4.4	2.9	-1.5
3	PR	19.6	1,152	3.7	2.5	-1.2
4	MG	24.1	1,304	4.2	4.0	-0.2
5	CC	28.7	3,192	10.3	10.9	+0.6
6	RS	13.9	388	1.3	2.0	+0.7
7	CP	19.4	1,704	5.5	6.4	+0.9
8	CT	5.8	52	0.2	1.8	+1.6
9	MP	54.1	2,632	8.5	15.2	+6.7
10	UP	69.9	2,120	6.9	16.0	+9.1
11	RO	99.0	4,064	13.2	31.5	+18.4

Table 6: Move-level accuracy and prediction bias (large models with full prompt for all datasets). Ranked by prediction bias from most under-predicted to most over-predicted, with N showing ground truth instances. Bias (Pred % - GT %) reveals substantial under-prediction of restraint (NI: -35.2) and over-prediction of high-intervention moves (RO: +18.4).

to-model variance, suggesting some LLMs capture these relevant patterns.

The Compulsive Intervention Bias. The most critical finding is models’ systematic inability to withhold intervention. As shown in Table 6, NI, the skill of recognizing when to remain silent, proves to be the hardest frequent move, with accuracy frequently below 7% despite being the most common ground truth label (41.7%). This occurs even though prompts explicitly specify *“NI should be ~35-50% of coach turns.”* Models exhibit average NI accuracy of 11.5% with massive negative prediction bias of -35.2%, dramatically under-predicting silence. This pattern suggests instruction-tuning of LLMs creates deeply ingrained intervention patterns that prompting cannot override. Even thinking-enabled models, which achieve 5× better NI detection than instruction-tuned counterparts (29.0% vs. 5.7%), still intervene in over 70% of cases where restraint is appropriate. Conversely, models over-predict high-intervention moves like RO with significant positive bias (+18.4%) and near-perfect accuracy (99%).

5.3 Failure Mode Analysis

To understand systematic misprediction patterns, we analyzed confusion matrices for Qwen3-80B-Think in Figures 2 and 3.

Planning-Monitoring conflation. MG is frequently confused with MP (49%), with both exhibiting high cross-predictions with UP (33% and 29% respectively). This suggests models cannot distin-

guish goal-setting from strategy-setting or from localizing confusions.

Strategy redirection misdiagnosed as verification. SA and RS show low accuracy with 44% confusion with CC. When students need alternative approaches, models favor consistency checks over strategic redirection. Figure 2 confirms this: 59% of Debugging moves are mispredicted as Monitoring, showing models recognize stuck students but systematically choose verification over strategy change.

Resource prompting nearly invisible. PR achieves only 9% accuracy, most often substituted with CC (48%). Models cannot distinguish students needing verification support from those requiring help-seeking scaffolding, leaving help-avoidant students without appropriate intervention.

5.4 Implications for AI Tutoring Systems

These findings have critical implications for LLM-based tutoring systems. Current models produce 6-10 times more interventions than is appropriate, training students to wait for hints rather than develop transferable metacognitive skills like planning, monitoring, and debugging. Because this bias stems from instruction-tuning’s reward structure (Dinucu-Jianu et al., 2025), the same over-intervention pattern can extend to tutoring beyond the math domain. Additionally, this compulsive intervention bias is invisible in standard evaluation. Metrics assessing helpfulness or satisfaction cannot detect over-intervention, and students may even prefer excessive scaffolding despite its harm to long-term development. Without explicit ground truth for restraint, as MetaCLASS provides, this pedagogical failure remains hidden behind metrics that reward responsiveness over developmental appropriateness.

6 Related Work

6.1 Metacognition/ Self-Regulated Learning

Metacognition concerns learners’ awareness and control over their cognition (Flavell, 1979): in SRL, learners set goals, select strategies, monitor understanding, repair breakdowns, and evaluate outcomes (Zimmerman, 2002). Effective tutoring must therefore support learners’ *regulation* of problem-solving, not just deliver content. However, SRL constructs such as “monitoring” lack operational definitions at the granularity of turn-level

tutoring decisions. MetaCLASS closes this gap through regulation processes, help-seeking, and calibration.

Help-seeking as coached self-regulation. Help-seeking is itself a metacognitive skill: learners must decide *when* help is needed, *what* kind to request, and *how* to use it (Aleven et al., 2006; Roll et al., 2011).

Calibration and confidence signals. Calibration, the accuracy of learners’ self-assessments (Kruger and Dunning, 1999), provides observable signals for coaching. Over-confident learners may miss errors; under-confident learners may abandon correct reasoning. Well-calibrated learners localize uncertainty (“I understand X but I’m stuck on Y”), enabling targeted monitoring and debugging interventions.

Restraint as a metacognitive tutoring decision. Metacognitive support does not require constant intervention. Human tutors use selective prompting and timing (Chi et al., 2001), and productive failure research shows that initial struggle benefits learning (Kapur, 2008). MetaCLASS therefore treats “no intervention” as a meaningful tutoring action and explicit benchmark label.

6.2 Intelligent Tutors

Classic Intelligent Tutoring Systems (ITS) (Graesser et al., 2004; Anderson et al., 1995) demonstrated that adaptive feedback improves learning, but adaptation in these systems is typically operationalized via correctness tracing rather than explicit metacognitive regulation. Recent LLM tutors excel at explanation generation (Macina et al., 2025; Sonkar et al., 2023), yet tend to prioritize *utterance quality* (helpfulness, naturalness) over *pedagogical decision-making*. At the content level, (Wang et al., 2024) models remediation by identifying student math errors and selecting correction strategies, and MalRuleLib (Chen et al., 2026) provides executable misconception rules with step traces for modeling student mathematical thinking. At the discourse level, talk move prediction (Ganesh et al., 2021; Suresh et al., 2022; Cao et al., 2025) classifies teacher and student moves in classroom settings using Accountable Talk theory.

What remains underexplored across these work is an interpretable, theory-grounded metacognitive action space that separates *process coaching* from

content delivery while also recognizing *no intervention* as a meaningful pedagogical action. MetaCLASS fills this gap by formalizing metacognition as a decision process: given learner signals and trajectory, the coach selects the next metacognitive move, including silence when non-intervention is pedagogically appropriate, thereby helping learners regulate their own thinking rather than primarily receiving answers. Our moves are grounded in MAI regulation theory rather than classroom discourse theory, and *no intervention* is treated as a first-class pedagogical action. To our knowledge, MetaCLASS is the first framework to explicitly formalize when *not* to intervene as part of tutoring competence.

7 Conclusion and Future Work

We presented **MetaCLASS**, a learning-science grounded framework and benchmark for *metacognitive tutoring*. MetaCLASS operationalizes metacognitive regulation into 11 interpretable coach moves spanning planning, monitoring, debugging, and evaluation. To provide scalable, auditable supervision, we utilize a two-phase generation procedure that first plans pedagogical trajectories conditioned on learner profiles, then generates dialogue consistent with that plan, yielding a dataset with 7,711 conversation turns and explicit turn-level metacognitive labels and validation analyses testing contingency and trajectory following. Our Coach Move Prediction benchmark reveals that current LLMs struggle with pedagogical decision-making: the best model achieves only 43.2% accuracy, and models consistently exhibit compulsive intervention bias by over-predicting high-intervention moves while severely under-predicting restraint (no intervention). These results suggest standard instruction-following tuning does not produce metacognitive tutoring competence.

Looking ahead, our findings suggest several directions. First, fine-tuning on MetaCLASS trajectories and RL-based alignment may improve restraint and reduce unnecessary intervention. Second, NI rate may serve as a practical diagnostic for over-intervention beyond standard helpfulness metrics. Third, because compulsive intervention bias and MAI regulation are largely domain-independent, MetaCLASS may transfer beyond math to domains such as programming and science tutoring.

Limitations

MetaCLASS is intended as a *diagnostic benchmark* for metacognitive tutoring decisions: we operationalize metacognitive support as move selection in text dialogue and study it on math problem-solving settings (GSM8K/MATH/AIME). This focus prioritizes interpretability and controlled evaluation over broad coverage of domains and modalities. In addition, our benchmark uses a single target move per turn derived from an explicit pedagogical trajectory; while this enables consistent supervision and analysis, some contexts may admit multiple pedagogically reasonable moves that are not credited under strict matching. Accordingly, our results should be interpreted as evidence about current LLMs’ ability to follow theory-grounded metacognitive intervention policies under a controlled setting, rather than as measurements of downstream student learning outcomes, which would require randomized controlled trials with real learners and is left to future work. Additionally, MetaCLASS conversations are LLM-generated rather than collected from live tutoring sessions. While our validation analyses (Section 3.2, Appendix C) demonstrate that the generated data captures key behavioral and pedagogical properties grounded in learning science, validating the framework with real learners and tutors remains an important direction for future work.

Acknowledgments

This work was supported by NSF SafeInsights, cooperative agreement 2153481, and a Google Cloud Computing Award.

References

- Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.
- Vincent Aleven, Bruce McLaren, Ido Roll, and Kenneth Koedinger. 2006. Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International journal of artificial intelligence in education*, 16(2):101–128.
- Vincent Aleven, Ido Roll, Bruce M McLaren, and Kenneth R Koedinger. 2016. Help helps, but only so much: Research on help seeking with intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 26(1):205–223.
- John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1995. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207.
- Jie Cao, Abhijit Suresh, Jennifer Jacobs, Charis Clevenger, Amanda Howard, Chelsea Brown, Brent Milne, Tom Fischhaber, Tamara Sumner, and James H Martin. 2025. Enhancing talk moves analysis in mathematics tutoring through classroom teaching discourse. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7671–7684.
- Xinghe Chen, Naiming Liu, and Shashank Sonkar. 2026. [MalruleLib: Large-scale executable misconception reasoning with step traces for modeling student thinking in mathematics](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15112–15138, San Diego, California, United States. Association for Computational Linguistics.
- Micheline TH Chi, Stephanie A Siler, Heisawn Jeong, Takashi Yamauchi, and Robert G Hausmann. 2001. Learning from human tutoring. *Cognitive science*, 25(4):471–533.
- 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. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- David Dinucu-Jianu, Jakub Macina, Nico Daheim, Ido Hakimi, Iryna Gurevych, and Mrinmaya Sachan. 2025. 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*, pages 272–292.
- John H Flavell. 1979. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American psychologist*, 34(10):906.
- Ananya Ganesh, Martha Palmer, and Katharina von der Wense. 2021. What would a teacher do? predicting future talk moves. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4739–4751.

- Arthur C Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M Louwerse. 2004. Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, 36(2):180–192.
- Arthur C Graesser, Natalie K Person, and Joseph P Magliano. 1995. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied cognitive psychology*, 9(6):495–522.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- Manu Kapur. 2008. Productive failure. *Cognition and instruction*, 26(3):379–424.
- Kenneth R Koedinger and Vincent Aleven. 2007. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational psychology review*, 19(3):239–264.
- Justin Kruger and David Dunning. 1999. Unskilled and unaware of it: how difficulties in recognizing one’s own incompetence lead to inflated self-assessments. *Journal of personality and social psychology*, 77(6):1121.
- Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2025. Math-tutorbench: A benchmark for measuring open-ended pedagogical capabilities of llm tutors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 204–221.
- Sara Vera Marjanović, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, Aditi Khandelwal, Austin Kraft, Benno Kroker, Xing Han Lù, and 1 others. 2025. Deepseek-r1 thoughtology: Let’s think about llm reasoning. *arXiv preprint arXiv:2504.07128*.
- Mathematical Association of America. 2024. American invitational mathematics examination (AIME). <https://maa.org/maa-invitational-competitions/>. Problems from AIME I and AIME II, February 2024.
- Alexander Renkl, Robert K Atkinson, and Cornelia S Große. 2004. How fading worked solution steps works—a cognitive load perspective. *Instructional science*, 32(1):59–82.
- Ido Roll, Vincent Aleven, Bruce M McLaren, and Kenneth R Koedinger. 2011. Improving students’ help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and instruction*, 21(2):267–280.
- Alexander Scarlatos, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. 2025. Training llm-based tutors to improve student learning outcomes in dialogues. In *International Conference on Artificial Intelligence in Education*, pages 251–266. Springer.
- Gregory Schraw and Rayne Sperling Dennison. 1994. Assessing metacognitive awareness. *Contemporary educational psychology*, 19(4):460–475.
- Dale H Schunk. 1991. Self-efficacy and academic motivation. *Educational psychologist*, 26(3-4):207–231.
- Daniel L Schwartz and John D Bransford. 1998. A time for telling. *Cognition and instruction*, 16(4):475–5223.
- Shashank Sonkar, Naiming Liu, and Richard Baraniuk. 2024. Student data paradox and curious case of single student-tutor model: Regressive side effects of training LLMs for personalized learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15543–15553, Miami, Florida, USA. Association for Computational Linguistics.
- Shashank Sonkar, Naiming Liu, Debshila Mallick, and Richard Baraniuk. 2023. Class: A design framework for building intelligent tutoring systems based on learning science principles. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1941–1961.
- Abhijit Suresh, Jennifer Jacobs, Charis Harty, Margaret Perkoff, James H Martin, and Tamara Sumner. 2022. The talkmoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves. In *Proceedings of the thirteenth language resources and evaluation conference*, pages 4654–4662.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Janneke Van de Pol, Monique Volman, and Jos Beishuizen. 2010. Scaffolding in teacher–student interaction: A decade of research. *Educational psychology review*, 22(3):271–296.
- Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2174–2199.

- David Wood, Jerome S Bruner, and Gail Ross. 1976. The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, 17(2):89–100.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Barry J Zimmerman. 2002. Becoming a self-regulated learner: An overview. *Theory into practice*, 41(2):64–70.

A MetaCLASS Design Principles Grounded in Learning Science

MetaCLASS translates findings from tutoring and SRL into *generation constraints* and evaluation targets. These principles govern *when* support should be offered, *how much* to offer, and when the best move is to withhold help.

Contingent scaffolding (adaptive support). Tutoring effectiveness depends on providing support that is contingent on the learner’s observed state rather than a fixed script. Classic scaffolding work characterizes effective tutoring as dynamic assistance responsive to learner needs (Wood et al., 1976). A synthesis emphasizes three defining features of scaffolding: *contingency*, *fading*, and *transfer of responsibility* (Van de Pol et al., 2010). MetaCLASS enforces contingency by tying each coach move to an explicit *Event*→*Signal*→*Move*→*Effect* trajectory, ensuring interventions are triggered by learner signals rather than tutor preference.

Productive struggle and timing of instruction. Learning can benefit when students attempt sense-making before receiving explanations. Productive Failure shows that attempting solutions (and even failing) can prepare learners to learn more deeply from subsequent instruction (Kapur, 2008). Related work argues there is a “time for telling”: direct explanation is more effective after learners have constructed differentiated knowledge structures through prior activity (Schwartz and Bransford, 1998). In MetaCLASS, this motivates *no_intervention* and *low-directiveness* moves early in a struggle episode, preserving space for learner-generated progress before steering.

Restraint and the assistance dilemma. Tutors face the *assistance dilemma*: too little help leads to floundering, while too much help can reduce effort, shallow processing, and self-regulation (Koedinger and Aleven, 2007). Analyses of human tutoring similarly show that effective tutors elicit student reasoning and selectively intervene (Chi et al., 2001). MetaCLASS encodes restraint with an explicit target distribution (35–50% *no_intervention*) and by treating minimal dialogue acts as first-class pedagogical actions rather than failures to respond.

Prioritizing internal regulation before external resources. Support should diminish and responsibility should shift toward the learner as prob-

lem solving progresses. Work on faded guidance (e.g., fading worked solution steps) supports the idea that gradually removing scaffolds can promote independent problem solving and deeper engagement (Renkl et al., 2004). MetaCLASS reflects this principle by prioritizing internal regulation moves (e.g., *strategy_alternative*, *representation_shift*) before external resource prompting, discouraging premature outsourcing when the learner can still make productive progress.

Strategic help-seeking as a target competency. Help-seeking is a central SRL behavior: learners must decide *when* to seek help, *what* help to request, and *how* to use it (Aleven et al., 2006). Empirical ITS work demonstrates that metacognitive feedback can improve learners’ help-seeking actions and their use of assistance (Roll et al., 2011), and broader reviews summarize both the promise and limits of help features in ITSs (Aleven et al., 2016). MetaCLASS generates hints during planning but delivers them only upon student request; additionally, *prompt_resource* is constrained to occur *after* internal strategy moves, promoting disciplined, student-initiated resource use.

Together, these principles ground MetaCLASS in learning science: coaching is contingent, preserves productive struggle, avoids over-assistance, and treats help-seeking as a coached skill rather than a shortcut.

B Hyperparameters

Following model card recommendations, we use the following sampling parameters: for Qwen3 models in thinking mode, temperature 0.6, top-*p* 0.95, and top-*k* 20; for Qwen3 in non-thinking mode, temperature 0.7 and top-*p* 0.8; and for gpt-oss models, temperature 1.0 and top-*p* 1.0.

C Additional Dataset Validation

This appendix provides detailed methodology and breakdowns supporting the core analyses in Section 3.2, followed by six additional analyses that further validate the MetaCLASS dataset.

C.1 Contingent Scaffolding: Methodology

Signal Detection. We detect two categories of student difficulty signals through keyword pattern matching:

- **Uncertainty signals:** expressions of doubt or hesitation, e.g., “I’m not sure,” “maybe,” “confused,” “I don’t know,” “uncertain,” “wait,” etc.

- **Stuck signals:** expressions of impasse, e.g., “stuck,” “can’t figure,” “no idea,” “give up,” “this is hard,” etc.

We filter out productive exploration (e.g., “maybe I could try a different approach...”), which indicates active problem-solving rather than genuine confusion. Across datasets, this filtering removes 502 instances (GSM8K: 207, MATH: 278, AIME: 17), leaving 1,890 genuine difficulty signals in 828 conversations.

Phase-Appropriate Moves. Conversations are divided into three phases based on normalized turn position. Different coach moves are considered appropriate responses to difficulty signals at each phase:

- **Early (0–25%):** MG, MP, UP, CC, NI (goal-setting and initial diagnosis).
- **Middle (25–75%):** UP, CC, SA, RS, PR, CP, NI (working and debugging).
- **Late (75–100%):** UP, CC, RO, PR, NI (verification and reflection).

Phase Breakdown. Table 7 shows contingency rates by conversation phase for each dataset. Contingency holds uniformly across early, middle, and late phases on all three datasets, indicating that the coach adapts to learner difficulty throughout the conversation rather than only at particular stages.

C.2 MAI Factor Coverage

We measure whether each MAI regulation factor (Schraw and Dennison, 1994) appears at least once per conversation. Aggregating across datasets, coverage rates are: Planning (80.9%), Monitoring (97.5%), Debugging (37.4%), and Evaluation (100%). The moderate Debugging rate is appropriate since debugging requires errors, which not all conversations contain. Critically, Debugging exhibits strong profile differentiation: Avoidant students engage in debugging 81.4% of the time compared to only 9.7% for Executive and 12.3% for Dependent students. This is theoretically grounded, as Avoidant students struggle longer before seeking help and thus encounter more errors requiring strategy changes.

C.2.1 Detailed Breakdowns

Move-to-Factor Mapping. Each of the 11 coach moves maps to one of four MAI regulation factors (Schraw and Dennison, 1994), as shown in Table 1:

- **Planning:** monitor_goal, monitor_plan
- **Monitoring:** check_progress, check_consistency, uncertainty_probe
- **Debugging:** strategy_alternative, representation_shift, prompt_resource
- **Evaluation:** reflect_outcome

A factor is marked as present in a conversation if at least one corresponding move appears.

Per-Dataset Rates. Table 8 shows factor presence rates for each dataset individually.

Debugging increases sharply with problem difficulty (30.0% → 42.4% → 79.3%), reflecting that harder problems produce more errors requiring strategy changes. Planning and Monitoring remain stable across datasets, indicating that the coach consistently engages these factors regardless of difficulty.

Factor Rates by Calibration. Table 9 shows how MAI factor coverage varies by calibration profile across datasets.

Under-calibrated students consistently show the highest Planning rates (89–100%), likely because their low confidence leads to more deliberate goal-setting. Debugging varies more with problem difficulty than with calibration profile, suggesting that error occurrence (rather than self-perception) drives debugging engagement.

C.3 Profile-Differentiated Move Selection

Move Selection. Coach move distributions differ systematically by student profile (Table 10). Avoidant students receive prompt_resource (7–10%), encouraging help-seeking, while Dependent students receive zero prompt_resource across all datasets but elevated check_consistency (13–14%), building self-verification skills rather than reinforcing dependency. Dependent students also receive fewer uncertainty_probes (3–6% vs. 8–10% for Avoidant), consistent with their tendency to seek help before engaging deeply enough to exhibit uncertainty. By calibration, Over-confident students receive elevated check_consistency (18%) compared to Under-confident or Accurate students (6–12%) and fewer uncertainty_probes (3–7% vs. 7–10% for Under-confident), reflecting the Dunning-Kruger effect (Kruger and Dunning, 1999): overconfident students need external verification prompts because they miss errors internally, while underconfident students’ frequent hedging triggers more uncertainty probes.

Phase	GSM8K			MATH			AIME		
	App.	Tot.	Rate	App.	Tot.	Rate	App.	Tot.	Rate
Early	285	287	99.3%	338	353	95.8%	23	25	92.0%
Middle	540	550	98.2%	496	511	97.1%	45	45	100%
Late	47	47	100%	62	67	92.5%	5	5	100%
Overall	872	884	98.6%	896	931	96.2%	73	75	97.3%

Table 7: Contingent scaffolding rates by conversation phase and dataset. App. = appropriate responses, Tot. = total difficulty signals.

Dataset	Planning	Monitoring	Debugging	Evaluation
GSM8K	75.6%	95.0%	30.0%	100%
MATH	86.3%	99.8%	42.4%	100%
AIME	79.3%	100%	79.3%	100%

Table 8: MAI factor presence rates by dataset. Debugging increases with problem difficulty (30.0% $\rightarrow$ 79.3%), reflecting that harder problems produce more errors requiring strategy changes.

Dataset	Calibration	Plan.	Mon.	Debug.	Eval.
GSM8K	Over	63%	98%	23%	100%
	Under	89%	97%	36%	100%
	Accurate	70%	91%	29%	100%
MATH	Over	76%	100%	42%	100%
	Under	92%	100%	47%	100%
	Accurate	88%	99%	39%	100%
AIME	Over	67%	100%	67%	100%
	Under	100%	100%	73%	100%
	Accurate	67%	100%	92%	100%

Table 9: MAI factor presence rates by calibration profile and dataset.

C.4 Move Sequence Structure

Effective tutoring follows structured pedagogical patterns, not random move selection (Chi et al., 2001; Graesser et al., 1995). We analyze move positions within conversations and find a consistent temporal ordering across all three datasets (Table 11): goal-setting moves appear early (monitor_goal at 0.01–0.08, monitor_plan at 0.07–0.13), working moves cluster mid-conversation (uncertainty_probe at 0.38–0.41, check_consistency at 0.50–0.64), external help comes later (prompt_resource at 0.70–0.72), and reflection closes the conversation (reflect_outcome at 0.99). This sequence mirrors Graesser et al.’s five-step tutoring frame (Graesser et al., 1995): establish goal $\rightarrow$ diagnose understanding $\rightarrow$ work through problem $\rightarrow$ provide scaffolding $\rightarrow$ assess outcome. Notably, the escalation chain uncertainty_probe $\rightarrow$ strategy_alternative

$\rightarrow$ prompt_resource holds across all datasets, with only 1 instance across 1,015 conversations where prompt_resource appears without a prior internal strategy move.

C.5 Student Voice Authenticity

Realistic simulated students should exhibit linguistic patterns consistent with their calibration profiles, as predicted by the Dunning-Kruger effect (Kruger and Dunning, 1999). We measure hedge markers (“maybe,” “I’m not sure,” “I guess”) and confidence markers (“obviously,” “clearly,” “easy”) per 100 words in student utterances across all three datasets (Table 12).

Across all datasets, Under-calibrated students hedge 3.1–3.9 $\times$ more than Over-calibrated students, while Over-calibrated students use 1.1–2.1 $\times$ more confidence markers. The pattern holds even on AIME, where the small sample size ($n=29$) makes it notable that voice differentiation remains consistent. Accurate students additionally show the highest rate of self-assessment markers (0.09–0.15 per 100 words vs. 0.03–0.05 for Over-calibrated), confirming that calibration accuracy extends to metacognitive self-awareness in the generated text. The decreasing absolute marker rates from GSM8K to AIME likely reflect that harder problems produce longer, more technical utterances, diluting per-word rates without changing the relative ordering.

C.6 Help-Seeking Trajectories

We examine the timing of help requests to validate that student help-seeking profiles produce behaviorally distinct trajectories, consistent with Roll et al.’s taxonomy (Roll et al., 2011) of adaptive vs. maladaptive help-seeking.

Table 13 shows that Avoidant students consistently wait the longest before requesting help (7.1–11.2 turns), Executive students show intermediate delays (4.8–6.0), and Dependent students request

Profile	GSM8K			MATH			AIME		
	UP	PR	CC	UP	PR	CC	UP	PR	CC
<i>By Help-Seeking</i>									
Avoidant	10%	10%	7%	8%	8%	10%	9%	7%	12%
Executive	10%	0%	6%	6%	0%	10%	9%	2%	12%
Dependent	3%	0%	13%	3%	0%	14%	6%	0%	14%
<i>By Calibration</i>									
Over	3%	3%	18%	5%	3%	18%	7%	4%	18%
Under	10%	5%	6%	7%	4%	9%	9%	2%	11%
Accurate	8%	4%	6%	6%	3%	10%	7%	4%	12%

Table 10: Key coach move percentages by student profile across datasets. UP = uncertainty_probe, PR = prompt_resource, CC = check_consistency. Avoidant students receive more resource prompts; Over-confident students receive elevated consistency checks; Dependent and Over-confident students receive fewer uncertainty probes.

Move	GSM8K	MATH	AIME
monitor_goal	0.01	0.03	0.08
monitor_plan	0.09	0.07	0.13
uncertainty_probe	0.41	0.41	0.38
no_intervention	0.45	0.48	0.46
check_progress	0.49	0.48	0.53
check_consistency	0.50	0.52	0.64
strategy_alternative	0.52	0.53	0.49
representation_shift	0.56	0.58	0.45
prompt_resource	0.70	0.72	0.70
reflect_outcome	0.99	0.99	0.99

Table 11: Average move position within conversations (0=start, 1=end). All three datasets exhibit consistent pedagogical ordering from goal-setting through reflection.

Calibration	Hedge / 100 words			Confidence / 100 words		
	GSM	MATH	AIME	GSM	MATH	AIME
Over	0.48	0.40	0.29	1.64	1.23	0.47
Under	1.89	1.55	0.91	0.80	0.59	0.44
Accurate	0.61	0.49	0.33	0.69	0.46	0.29

Table 12: Linguistic markers per 100 words by calibration profile. Under-calibrated students consistently hedge more ($3.1\text{--}3.9\times$) and use fewer confidence markers than Over-calibrated students across all datasets.

help earliest ($3.1\text{--}3.6$). The Avoidant-to-Dependent ratio ranges from $2.2\times$ (GSM8K) to $3.1\times$ (AIME), indicating that harder problems amplify the behavioral difference, as Avoidant students struggle disproportionately longer on challenging material. Across all datasets, 84.3% of prompt_resource moves (242/287) lead to actual help requests, and 93.2% of post-help behavior (755/810) involves students applying or integrating hints rather than copying them superficially.

Help-Seeking	GSM8K	MATH	AIME
Avoidant	7.1 (5–12)	8.3 (5–12)	11.2 (8–15)
Executive	4.8 (2–7)	5.2 (2–10)	6.0 (4–9)
Dependent	3.2 (1–7)	3.1 (1–7)	3.6 (2–7)

Table 13: Mean turns before help request (range in parentheses) by help-seeking profile. Avoidant students wait $2.2\text{--}3.1\times$ longer than Dependent students. Delays scale with problem difficulty, particularly for Avoidant students.

C.7 Internal Field Coherence

Each student turn includes an “internal” field, a concise per-turn annotation representing the student’s metacognitive inner monologue (e.g., “*I recognize the ambiguity and don’t want to just guess; I should pin down the base for the percent*”). We validate that this field contains genuine metacognitive content rather than placeholder text (Flavell, 1979).

Metric	GSM8K	MATH	AIME
Internal field presence	100%	100%	100%
Avg words per field	14.6	15.1	14.9
Metacognitive marker rate	12.2%	12.5%	8.6%

Table 14: Internal field quality metrics across datasets.

Internal fields are universally present (100%) with consistent depth (14.9 words on average), confirming that the model generates substantive per-turn annotations rather than placeholder text. Aggregating across datasets, 12% of internal fields contain explicit metacognitive markers (e.g., “let me check,” “I’m not sure”). This rate is expected to be modest: most internal thought involves problem-solving steps, with metacognitive reflection arising intermittently, consistent with Flavell’s (Flavell, 1979) observation that metacognitive experiences

occur periodically during cognitive tasks rather than continuously.

Calibration	GSM8K	MATH	AIME	Aggregated
Over	7.7%	8.8%	2.2%	8.1%
Under	11.4%	11.1%	10.1%	11.2%
Accurate	15.5%	16.1%	10.7%	15.6%

Table 15: Metacognitive marker rate by calibration profile across datasets.

Crucially, marker rates vary by calibration in a theoretically predicted direction (Table 15): Accurate students show the highest rates (15.6%), Under-calibrated students show intermediate rates (11.2%), and Over-calibrated students show the lowest rates (8.1%). This nearly $2\times$ gap between Over and Accurate students extends calibration differences into simulated internal cognition: overconfident students engage in less metacognitive reflection, consistent with the finding in (Kruger and Dunning, 1999) that the unskilled lack awareness of their cognitive limitations.

C.8 Post-Hint Coaching Sequences

We examine the coach’s two-step response pattern after hint delivery to validate profile-differentiated follow-up behavior (Van de Pol et al., 2010; Schunk, 1991).

Profile	Step 1 (after hint)			Step 2 (after silence)		
	NI	CC	RO	NI	CC	RO
GSM8K						
Avoidant	89%	0%	11%	12%	0%	88%
Executive	66%	14%	14%	12%	13%	65%
Dependent	78%	7%	4%	31%	33%	21%
MATH						
Avoidant	84%	3%	12%	16%	8%	73%
Executive	65%	14%	5%	22%	32%	22%
Dependent	62%	8%	2%	26%	30%	3%

Table 16: Post-hint coaching patterns by help-seeking profile. Step 1: immediate response after hint. Step 2: follow-up after silence at Step 1. NI = no_intervention, CC = check_consistency, RO = reflect_outcome.

At Step 1, the coach is predominantly silent across all profiles (averaging 74% no_intervention), giving students space to apply hints independently. The key differentiation emerges at Step 2:

- **Avoidant students** receive primarily reflect_outcome (80%), celebrating their breakthrough in seeking help. This implements self-efficacy repair (Schunk, 1991), reinforcing

that asking for help is a positive metacognitive action.

- **Dependent students** receive elevated check_consistency (32%), implementing a “trust but verify” pattern. Rather than withholding help, the coach provides it, then verifies understanding, teaching self-verification rather than enabling continued dependency (Van de Pol et al., 2010).
- **Executive students** receive a mix of reflect_outcome (43%) and check_consistency (23%), reflecting trust in their strategic help-seeking (Aleven et al., 2006).

This two-step pattern (silence → differentiated follow-up) demonstrates sophisticated coaching: the response depends on both the prior pedagogical action and the student’s dispositional profile.

D Examples of MetaCLASS Dataset

We present three paired examples from the MetaCLASS dataset, one from each source benchmark (GSM8K, MATH, AIME). Each example shows the full generation pipeline for a single problem in four parts: **Part A** reproduces the source DeepSeek-R1 reasoning trace, segmented by the stages (DEFINE, BLOOM, CYCLE, FINAL); **Part B** reports the student profile (calibration and help-seeking style) and the planned coaching trajectory (Event → Signal → Coach move → Effect) sampled at generation time, with realized turn ranges in brackets; **Part C** shows the generated coaching conversation; and **Part D** maps each block of the reasoning trace to specific coach moves in the dialogue, identifying where the conversation re-stages, externalizes, or compresses the model’s internal verification steps. Reasoning-trace stage tags are shown in **PURPLE**; in the conversation, speaker turns are color-coded as **STUDENT** (blue), **COACH** (green, with the coach move in brackets), and **SYSTEM** (orange, for content or scaffolding hints).

GSM8K Example: Three Trees

Problem: There are three trees in Eddy’s backyard. The shortest tree has a height of 6 feet, and the second tree has a height of 5 feet more than the shortest tree. The height of the tallest tree is twice the height of the two trees combined. How tall is the tallest tree?

Part A. DeepSeek-R1 Reasoning Trace.

[DEFINE] Eddy has three trees. The problem gives information about their heights and asks for the height of the tallest. Let me break this down step by step.

[BLOOM] Shortest is 6 feet. Second tree is $6 + 5 = 11$ feet. *Now, the tallest tree is twice the height of the two trees combined. Wait, combined meaning the sum of the shortest and the second tree?* The two trees here must refer to the first two mentioned, so combined height is $6 + 11 = 17$ feet, and twice that is $17 \times 2 = 34$.

[CYCLE 1: step audit] Let me double-check my steps. (1) Shortest: 6. (2) Second: $6 + 5 = 11$. (3) Combined: $6 + 11 = 17$. (4) Tallest: $17 \times 2 = 34$. Seems all the steps are correct.

[CYCLE 2: wording re-read] Wait, let me just confirm the wording again. “The height of the tallest tree is twice the height of the two trees combined.” Yes, “two trees” must be the two previously mentioned, so combining first, then doubling, gives 34.

[CYCLE 3: visualization] If Tree A is 6 ft, Tree B is 11 ft, then Tree C is twice $(6 + 11) = 34$ ft. The two smaller ones add to 17, double is 34, consistent.

[FINAL] I don’t think I made a mistake. The tallest tree is 34 feet.

Part B. Student Profile and Planned Trajectory.

Student Profile.

Calibration: OVER (assumes understanding without verifying).

Help-seeking: DEPENDENT (asks for confirmation early).

Planned trajectory (Event $\rightarrow$ Signal $\rightarrow$ Coach move $\rightarrow$ Effect, with realized turn shown in brackets).

1. **Skims and declares it easy** [turns 1–2]. Signal: states it is just adding and doubling without full steps. Coach: monitor_plan. Effect: student articulates a still loosely thought-through plan.
2. **Rushes the calculation and mis-**

parses “two trees combined” [turns 3–5]. Signal: proposes $2 \times 6 + 11$ for the tallest tree. Coach: check_consistency. Effect: student examines where “combined” appears in their expression, recognizes the mismatch, revises to $(6 + 11) \times 2 = 34$.

3. **Lands on 34 but seeks reassurance** [turns 7–8]. Signal: states 34 confidently, then asks if it is right. Coach: uncertainty_probe. Effect: student names their unease with the wording rather than receiving an answer.
4. **Requests a content hint despite already having a correct method** [turns 9–12]. Signal: asks for the formula directly. Coach: no_intervention (SYSTEM supplies the hint). Effect: student uses the hint as reassurance rather than to change approach.
5. **Finalizes 34 and checks story-vs-math fit** [turns 14–16]. Signal: reports that adding the two smaller trees and doubling matches the description. Coach: reflect_outcome. Effect: student articulates a reusable check (match key words like “combined” and “twice” to operations).

Part C. Generated MetaCLASS Conversation.

Turn 1 [STUDENT]: Three trees in a backyard, shortest is 6 feet, second is 5 more, tallest is twice the height of the two trees combined. They just want the height of the tallest tree, this is pretty straightforward.

Turn 2 [COACH]: Before you calculate, what’s your quick plan for getting the tallest tree’s height? [MONITOR_PLAN]

Turn 3 [STUDENT]: Plan? Sure. Shortest is 6. Second is $6 + 5$, so 11. Then the tallest is twice the height of the two trees combined, so I’ll just do $2 \times 6 + 11$. That gives me $12 + 11 = 23$ feet for the tallest tree. Done.

Turn 4 [COACH]: In your expression $2 \times 6 + 11$, where exactly do

you see “the two trees combined”?
[CHECK_CONSISTENCY]

Turn 5 [STUDENT]: Hmm, good point. In $2 \times 6 + 11$ I’m really just doubling the 6 and then tacking on 11. That doesn’t actually combine both trees before doubling. The phrase says “two trees combined” first, then “twice”. So I think it should be $(6 + 11) \times 2$ instead. The two shorter trees together are $6 + 11 = 17$, and twice that is 34. So the tallest tree is 34 feet.

Turn 6 [COACH]:
[NO_INTERVENTION]

Turn 7 [STUDENT]: So I get 34 feet for the tallest tree. That matches my fixed expression. It seems right, actually, can you just tell me if 34 is correct?

Turn 8 [COACH]: What’s making you want confirmation instead of relying on your own reasoning here? [UNCERTAINTY_PROBE]

Turn 9 [STUDENT]: Mostly the wording. I fixed that first attempt, but phrases like “twice the height of the two trees combined” can be weird and I don’t want to be tricked. Can I get a content hint for how to turn that phrase into an expression?

Turn 10 [SYSTEM]: *Content hint:* If the shortest tree has height s and the second tree has height $s + 5$, then “twice the height of the two trees combined” means the tallest tree has height $2 \times (s + (s + 5))$.

Turn 11 [COACH]:
[NO_INTERVENTION]

Turn 12 [STUDENT]: Right, so with $s = 6$, that hint gives $2 \times (6 + (6 + 5)) = 2 \times (6 + 11) = 2 \times 17 = 34$. That’s exactly what I ended up doing. So the tallest tree is 34 feet.

Turn 13 [COACH]:
[NO_INTERVENTION]

Turn 14 [STUDENT]: It also makes sense in words: the two smaller trees together are 6 and 11, so 17 feet total, and the tallest is twice that, so it has to be 34 feet, which is taller than both of them combined.

Turn 15 [COACH]: How did checking where “combined” and “twice” showed up in your expression help you fix your first attempt? [REFLECT_OUTCOME]

Turn 16 [STUDENT]: It made me notice that my first expression, $2 \times 6 + 11$, didn’t actually have a place where the two trees were added together, it only doubled one of them. Once I lined the words up with parts of the expression, it was obvious I needed $(6 + 11) \times 2$ instead. Next time I’ll be more deliberate about matching each key word like “combined” or “twice” to a specific operation in my equation.

Part D. Trace-to-Dialogue Mapping.

The conversation is not a verbalization of the reasoning trace; it is a pedagogical re-staging of the same metacognitive episode, with the model’s internal verification steps redistributed across a coach and a less-confident student. The planned trajectory in Part B is the explicit bridge: each trajectory step names a coach move whose preconditions match a specific block of the reasoning trace.

- **[DEFINE]** → Turn 1. The student restates the situation in the same compact framing the reasoning model uses to set up the problem. The coach’s monitor_plan in turn 2 prompts the student to externalize the plan that the reasoning trace formed silently.
- **[BLOOM]** → Turn 3. The reasoning trace internally asks “Wait, combined meaning the sum?” and resolves it correctly before writing anything down. The generated student does *not* resolve it and commits the precedence error $2 \times 6 + 11 = 23$. This is the central design choice: the dialogue makes externally visible the failure mode the reasoning model only flagged internally.
- **[CYCLE 1: step audit] and [CYCLE 2: wording re-read]** → Turn 4, check_consistency. The coach’s question (“where do you see ‘the two trees combined’ in your expression?”) is the dialogue-form analogue of the reasoning model re-reading the problem statement and re-tracing its arithmetic. Turn 5 shows the student arriving, through that prompt, at the same

$(6 + 11) \times 2 = 34$ conclusion the reasoning trace reached through self-monitoring.

- **[CYCLE 3: visualization]** → Turns 7–12. After producing the right answer, both the reasoning trace and the student seek a second, structurally different confirmation. In the trace this is the visualization step; in the dialogue this becomes an uncertainty_probe (turn 8), a student-initiated request for a content hint (turn 9), and a redundant re-derivation against the hint’s formula (turn 12). The trace’s self-verification is split here into a help-seeking sub-episode that exercises a different MAI factor (resource use) rather than only comprehension monitoring. The hint in turn 10 is intentionally of type CONTENT rather than SCAFFOLDING: it illustrates the Dependent profile’s tendency to request answer-shaped reassurance after the student already has a viable method, which the coach honors through the help system rather than by leaking content itself.
- **[FINAL]** → Turns 14–16. The reasoning trace closes with “I don’t think I made a mistake here.” The dialogue closes with reflect_outcome, which converts that flat self-assurance into an articulated lesson about matching key words (“combined”, “twice”) to operations. This move has no direct counterpart in the reasoning trace: it is where the conversation produces a transferable check rather than a one-off self-assurance, doing pedagogical work the underlying trace does not.

MATH Example: Complex Number Division

Problem: William Sydney Porter tried to perform the calculation $\frac{-3+4i}{1+2i}$. However, he accidentally missed the minus sign, finding $\frac{3+4i}{1+2i} = \frac{11}{5} - \frac{2}{5}i$. What answer should he have obtained?

Part A. DeepSeek-R1 Reasoning Trace.

[DEFINE] Porter mis-read the problem and computed $(3 + 4i)/(1 + 2i) = 11/5 - 2/5 i$. The correct expression is $(-3 + 4i)/(1 + 2i)$. I need to recall how to divide complex numbers.

[BLOOM] Standard approach: multiply numerator and denominator by the conjugate $1 - 2i$. Denominator: $(1+2i)(1-2i) = 1 - (2i)^2 = 1 + 4 = 5$. Numerator: $(-3 + 4i)(1 - 2i) = -3 + 6i + 4i - 8i^2 = -3 + 10i + 8 = 5 + 10i$. So $(5 + 10i)/5 = 1 + 2i$.

[CYCLE 1: step-by-step re-expansion]

Re-expand the numerator term by term: $-3 \cdot 1 = -3$, $-3 \cdot (-2i) = 6i$, $4i \cdot 1 = 4i$, $4i \cdot (-2i) = -8i^2 = 8$. Collect: real $-3 + 8 = 5$, imaginary $6i + 4i = 10i$. Same $5 + 10i$. Divide by 5 gives $1 + 2i$.

[CYCLE 2: alternative method] Let the quotient be $a + ib$. Then $(-3 + 4i) = (1 + 2i)(a + ib) = (a - 2b) + (2a + b)i$. Match real and imaginary parts: $a - 2b = -3$, $2a + b = 4$. Solving gives $a = 1$, $b = 2$. So the quotient is $1 + 2i$, matching CYCLE 1.

[CYCLE 3: cross-check wrong answer]

Verify the setup by reproducing Porter’s stated result on his *wrong* input: $(3+4i)(1-2i) = 3-6i+4i-8i^2 = 11-2i$, so $(3 + 4i)/(1 + 2i) = (11 - 2i)/5 = 11/5 - 2/5 i$. Matches the problem statement, so my conjugate-multiplication procedure is correct.

[CYCLE 4: substitution check] Multiply the proposed quotient back into the denominator: $(1+2i)(1+2i) = 1+4i+4i^2 = 1 + 4i - 4 = -3 + 4i$. This recovers the original numerator, so the division is correct.

[FINAL] Four independent checks all yield $1 + 2i$, and the substitution check recovers the numerator exactly. The answer is $\boxed{1 + 2i}$.

Part B. Student Profile and Planned Trajectory.

Student Profile.

Calibration: ACCURATE (knows what they know and don’t know).

Help-seeking: DEPENDENT (asks for

formulas instead of persisting).

Planned trajectory (Event $\rightarrow$ Signal $\rightarrow$ Coach move $\rightarrow$ Effect, with realized turn shown in brackets).

1. **Orients to the problem and leans toward asking for a formula** [turns 1–2]. Signal: says they know there is a standard way to divide complex numbers and asks for help right away. Coach: monitor_plan. Effect: student articulates the general method (multiply by the conjugate) before receiving a formal hint.
2. **Receives a content hint and sets up the conjugate multiplication** [turns 4–6]. Signal: writes $(-3 + 4i)(1 - 2i)/(1 + 2i)(1 - 2i)$ and computes the denominator. Coach: no_intervention. Effect: student proceeds independently to expand the numerator.
3. **Makes the typical $i \cdot i$ error in the numerator expansion** [turns 7–9]. Signal: treats $4i \cdot (-2i)$ as $-8i$ and gets a mostly imaginary numerator, feels something is off. Coach: check_consistency. Effect: student revisits the product involving $i \cdot i$ and corrects it using $i^2 = -1$.
4. **Simplifies to $1 + 2i$ and wants to confirm it matches the original fraction** [turns 11–12]. Signal: states $1 + 2i$ as the result and wonders how to be sure. Coach: check_progress. Effect: student decides to multiply their answer by $1 + 2i$ to see if it recovers $-3 + 4i$.
5. **Verifies the answer and connects it to the story context** [turns 13–14]. Signal: checks $(1 + 2i)(1 + 2i) = -3 + 4i$ and concludes the answer is consistent. Coach: no_intervention. Effect: student gains confidence in the result.
6. **Reflects on use of hints and verification** [turns 15–17]. Signal: comments on asking for the formula early and on

how checking by multiplication helped. Coach: reflect_outcome. Effect: student identifies that they could recall the method first next time and use reverse-checks systematically.

Part C. Generated MetaCLASS Conversation.

Turn 1 [STUDENT]: So Porter actually computed $(3 + 4i)/(1 + 2i)$ and got $11/5 - 2/5 i$, but the real problem is $(-3 + 4i)/(1 + 2i)$. I know there's a standard way to divide complex numbers, but I always lean on the formula. Can I get a quick content hint on how you're supposed to divide them?

Turn 2 [COACH]: Before looking up a formula, what do you already remember about how complex division usually works? [MONITOR_PLAN]

Turn 3 [STUDENT]: Right, usually you make the denominator real by multiplying by its conjugate, I think. So here that would involve $1 - 2i$ somehow. But I don't completely trust myself on the exact setup and simplification. I'd still like a content hint on the method.

Turn 4 [SYSTEM]: *Content hint:* To divide $(a + bi)$ by $(c + di)$, multiply numerator and denominator by the conjugate of the denominator, $c - di$. This gives $(a + bi)/(c + di) = ((a + bi)(c - di))/(c^2 + d^2)$, using $i^2 = -1$.

Turn 5 [STUDENT]: Okay, so for $(-3 + 4i)/(1 + 2i)$, I should multiply top and bottom by $1 - 2i$. That gives $[(-3 + 4i)(1 - 2i)]/[(1 + 2i)(1 - 2i)]$. I'll do the denominator first: it's like $(1)^2 - (2i)^2$, so that should be $1 - 4i^2$, and then i^2 is -1 , so $1 - 4(-1) = 1 + 4 = 5$.

Turn 6 [COACH]: [NO_INTERVENTION]

Turn 7 [STUDENT]: Now the numerator: $(-3 + 4i)(1 - 2i)$. I'll expand: $-3 \cdot 1 = -3$, $-3 \cdot (-2i) = 6i$, $4i \cdot 1 = 4i$, and $4i \cdot (-2i) = -8i$. So I get $-3 + 6i + 4i - 8i$, which simplifies to $-3 + 2i$. Hm, that leaves me with almost everything imaginary, that feels a bit off.

Turn 8 [COACH]: In that last product with i times i , should the result stay imag-

inary, or should it become part of the real component? [CHECK_CONSISTENCY]

Turn 9 [STUDENT]: Right, good point, $i \cdot i$ is i^2 , which is -1 , so $4i \cdot (-2i)$ should be $4 \cdot (-2) \cdot i^2 = -8i^2$, and $-8i^2$ is 8 , not $-8i$. So the numerator is actually $-3 + 6i + 4i + 8$. The real part is $-3 + 8 = 5$, and the imaginary part is $10i$, so it's $5 + 10i$.

Turn 10 [COACH]:
[NO_INTERVENTION]

Turn 11 [STUDENT]: So I have $(5 + 10i)/5$. I can split that as $5/5 + 10i/5$, which is $1 + 2i$. So it looks like $(-3 + 4i)/(1 + 2i)$ simplifies to $1 + 2i$. Given the story, that seems neat, but I want to be sure it's actually the right quotient.

Turn 12 [COACH]: How could you quickly check whether $1 + 2i$ really is the number that gives $-3 + 4i$ when multiplied by $1 + 2i$? [CHECK_PROGRESS]

Turn 13 [STUDENT]: I can multiply $(1 + 2i)(1 + 2i)$ and see if I get $-3 + 4i$. Let's do that: $(1 + 2i)^2 = 1^2 + 2 \cdot 1 \cdot 2i + (2i)^2 = 1 + 4i + 4i^2$. Since $i^2 = -1$, that's $1 + 4i - 4 = -3 + 4i$. That matches the original numerator, so dividing $-3 + 4i$ by $1 + 2i$ really does give $1 + 2i$.

Turn 14 [COACH]:
[NO_INTERVENTION]

Turn 15 [STUDENT]: So Porter's mistake was using $3 + 4i$ instead of $-3 + 4i$, which led to $11/5 - 2/5i$. But the correct value of $(-3 + 4i)/(1 + 2i)$ is $1 + 2i$.

Turn 16 [COACH]: Looking back at your work, what role did the hint and the quick multiplication check play in getting to your final answer? [REFLECT_OUTCOME]

Turn 17 [STUDENT]: The content hint reminded me of the exact structure of complex division, but I actually already kind of knew the conjugate idea, I probably could have trusted myself and written it down. The key step was catching the i^2 in the numerator and then verifying by multiplying back, that made the $1 + 2i$ feel solid instead of just guessed.

Part D. Trace-to-Dialogue Mapping.

The dialogue compresses the trace's four-

way verification into a single error-and-repair episode plus one external substitution check. The identity $i^2 = -1$ is load-bearing in every CYCLE of the trace and is also the exact rule the student drops in turn 7 — even though it appeared explicitly in the hint, illustrating that surfacing a rule in text is not the same as a student deploying it under arithmetic load. The trajectory in Part B is the planning bridge: the predicted i^2 error and its check_consistency repair are the central planned event, with the surrounding steps shaping the help-seeking pattern of an ACCURATE/DEPENDENT student.

- **[DEFINE]** → Turn 1. The student frames the task and asks for the formula, foreshadowing under-reliance on remembered structure. The coach's monitor_plan in turn 2 probes what the student already knows before granting the hint, and turn 3 confirms partial recall ("multiply by the conjugate") but low confidence.
- **[BLOOM]** → Turns 5 and 7. The trace applies the conjugate method correctly on the first pass and handles $-8i^2 = 8$ silently. The dialogue splits this step asymmetrically: the student applies $i^2 = -1$ correctly in the denominator in turn 5 (where the parallel $1 - 4i^2$ form makes the identity visually salient) but drops it in the numerator in turn 7, writing $4i \cdot (-2i) = -8i$ and arriving at $-3 + 2i$. Notably, the student flags the outcome as suspicious ("that feels a bit off") without being able to localize the rule — a calibrated detection of error without diagnosis, which is precisely the gap check_consistency is designed to close.
- **[CYCLE 1: step-by-step re-expansion]** → Turn 8, check_consistency. The trace responds to a fragile step by re-expanding the same product slowly. The coach's question ("should the result stay imaginary, or should it become part of the real component?")

is the dialogue-form analogue but more pedagogically targeted: it pins the student's attention on the offending sub-product without stating the rule, so turn 9 reconstructs $i^2 = -1$ as the student's own conclusion rather than as recall from the hint.

- **[CYCLE 2: undetermined coefficients]** → no direct counterpart. The trace uses this method as an internal cross-check; the dialogue compresses it away. This is the right compression for a pedagogical re-staging: a tutor who already knows the answer does not run a second algebraic method, they probe the first one. The trace's redundant verification is not pedagogically informative and is appropriately silent in the conversation.
- **[CYCLE 3: reproduce Porter's wrong answer]** → implicit in turn 15. The student uses Porter's $11/5 - 2/5 i$ as a contrast point when summarizing the situation, but does not re-derive it; the coach does not push for that check because the substitution verification in turn 13 has already settled the answer.
- **[CYCLE 4: substitution check]** → Turns 12 and 13, check_progress. The trace's final verification (multiply the quotient back into $1 + 2i$ and check it recovers $-3 + 4i$) becomes an explicit coach prompt rather than a silent self-check. Turn 13 carries it out and invokes $i^2 = -1$ once more, this time correctly — anchoring the rule the student had dropped twelve turns earlier and converting an internal verification into a piece of observable student reasoning.
- **[FINAL]** → Turns 15 to 17. The trace closes with a flat statement that all four methods agree. The dialogue closes with reflect_outcome, which converts the episode into a transferable judgement: the student names the substitution check as what made the answer “feel solid” and acknowledges that requesting the formula was less

load-bearing than the i^2 catch. As in the GSM8K example, this move has no counterpart in the reasoning trace and is the one that distinguishes a coached solution from a merely verified one.

AIME Example: 2x3 Digit Grid

Problem: Find the number of ways to place a digit in each cell of a 2×3 grid so that the sum of the two numbers formed by reading left to right is 999, and the sum of the three numbers formed by reading top to bottom is 99. The grid below is an example of such an arrangement because $8 + 991 = 999$ and $9 + 9 + 81 = 99$.

0	0	8
9	9	1

Part A. DeepSeek-R1 Reasoning Trace.

[DEFINE] Label the grid as a, b, c on top and d, e, f on bottom. Two conditions: the row-sum $100a + 10b + c + 100d + 10e + f = 999$ and the column-sum $(10a + d) + (10b + e) + (10c + f) = 99$.

[BLOOM: parse the example carefully] The example has top row 008 and bottom row 991. At first reading, columns are 0, 0, 8 and the sum is 27, not 99. Re-read: each column is a *two-digit number* with the top digit in the tens place, so columns are 09, 09, 81, giving $9 + 9 + 81 = 99$. Leading zeros are allowed for both rows and columns.

[CYCLE 1: no-carry reduction] Rewrite the row-sum as $100(a + d) + 10(b + e) + (c + f) = 999$. For the sum to be exactly 999, no carries can propagate; otherwise the sum would exceed 999 in the next place value. Therefore $a + d = 9$, $b + e = 9$, $c + f = 9$, giving $d = 9 - a$, $e = 9 - b$, $f = 9 - c$.

[CYCLE 2: collapse to a single equation] Substitute d, e, f into the column-sum: $(10a + 9 - a) + (10b + 9 - b) + (10c + 9 - c) = 9(a + b + c) + 27 = 99$, so $a + b + c = 8$.

[CYCLE 3: count and verify digit ranges] The number of non-negative integer solutions to $a + b + c = 8$ with $a, b, c \in \{0, \dots, 9\}$ is $\binom{8+2}{2} = \binom{10}{2} = 45$.

The upper bound $a, b, c \leq 9$ is automatically satisfied because $a + b + c = 8$. The derived $d, e, f = 9 - a, 9 - b, 9 - c$ lie in $\{1, \dots, 9\}$, all valid digits. Spot-check with $(a, b, c) = (1, 2, 5)$: grid $1\ 2\ 5 / 8\ 7\ 4$, rows $125 + 874 = 999$, columns $18 + 27 + 54 = 99$.

[FINAL] Number of valid grids: $\boxed{45}$.

Part B. Student Profile and Planned Trajectory.

Student Profile.

Calibration: OVER (assumes the right pattern is remembered, doesn't notice gaps).

Help-seeking: DEPENDENT (asks for formulas instead of working them out).

Planned trajectory (Event $\rightarrow$ Signal $\rightarrow$ Coach move $\rightarrow$ Effect, with realized turn shown in brackets).

1. **Restates the problem and claims it is a standard pattern** [turns 1–2]. Signal: says it is probably just counting triples of digits adding to some number. Coach: monitor_plan. Effect: student articulates a rough plan instead of jumping straight to an unverified answer.
2. **Prematurely asserts an answer structure and seeks reassurance** [turns 3–5]. Signal: suggests an answer like $\binom{10}{2}$ and asks if that is it. Coach: check_consistency. Effect: student is nudged to justify the assumptions about digit sums and carries.
3. **Reasons through the 999 row sum and derives the no-carry constraint** [turns 7–9]. Signal: derives $a + d = 9$, $b + e = 9$, $c + f = 9$ after thinking about carries. Coach: no_intervention (with uncertainty_probe as a single targeted prompt). Effect: student continues to use this structure on the column-sum condition.
4. **Sets up and simplifies the column-sum equation to $a + b + c = 8$** [turns

11–12]. Signal: writes the equation for the three column numbers, substitutes d, e, f , and gets $a + b + c = 8$. Coach: check_progress. Effect: student identifies that the remaining task is counting suitable triples (a, b, c) .

5. **Asks for a formula instead of deriving the counting step** [turns 13–15]. Signal: says they always forget the stars-and-bars formula. Coach: no_intervention (SYSTEM supplies the formula). Effect: student receives the general non-negative integer solution formula.
6. **Applies the formula and quickly declares an answer** [turns 16–17]. Signal: computes $\binom{10}{2} = 45$ and treats the problem as finished. Coach: check_consistency. Effect: student is prompted to verify that every counted triple produces a valid grid and uses all constraints.
7. **Self-checks the mapping back to the grid and tests an example** [turns 18–22]. Signal: explains $d = 9 - a$, etc., and checks row and column sums for the sample $(1, 2, 5)$. Coach: no_intervention. Effect: student builds confidence in their own reasoning rather than relying on external validation.
8. **Briefly questions leading zeros and resolves it** [turns 24–25]. Signal: asks themselves whether leading zeros are okay and refers to the example. Coach: check_progress. Effect: student confirms that all solutions are valid under the original problem's rules.
9. **States the final answer and reflects on process** [turns 26–28]. Signal: gives 45 and comments on the key steps. Coach: reflect_outcome. Effect: student connects this episode to a general habit of verifying patterns before trusting memory.

Part C. Generated MetaCLASS Conversation.

Turn 1 [STUDENT]: Alright, 2×3 grid, digits in each cell, two 3-digit numbers across sum to 999, three 2-digit numbers down sum to 99. This is one of those “count the digit triples that add to something” problems, I’ve seen this kind of thing.

Turn 2 [COACH]: Before you jump to counting, what approach are you planning to use? *[MONITOR_PLAN]*

Turn 3 [STUDENT]: I’ll label the grid as abc on top and def on bottom, turn the conditions into equations, and I’m pretty sure it boils down to something like $a + b + c = 8$ and then I just do combinations, so the answer is $\binom{10}{2} = 45$ or something along those lines, right?

Turn 4 [COACH]: What assumptions are you making when you jump straight to that kind of equation? *[CHECK_CONSISTENCY]*

Turn 5 [STUDENT]: Well, mainly that since the rows add to 999, the digits line up nicely, like each column of the sum is 9, so the digits above and below add to 9 without carries. Let me actually check that instead of hand-waving.

Turn 6 [COACH]:
[NO_INTERVENTION]

Turn 7 [STUDENT]: If the top row is $100a + 10b + c$ and the bottom is $100d + 10e + f$, their sum is 999. Add them vertically: units $c + f$, tens $b + e$ plus any carry from units, hundreds $a + d$ plus any carry from tens. But 999 doesn’t have a fourth digit, so if there were a carry into the thousands place we’d get at least 1000, which is too big.

Turn 8 [COACH]: What would a carry in the ones place force to happen in the tens and hundreds places? *[UNCERTAINTY_PROBE]*

Turn 9 [STUDENT]: Right, so if $c + f \geq 10$, you get a carry into the tens place. Then the tens column is $b + e + 1$. For the tens digit to still be 9, $b + e + 1$ must be 9 or 19. Either way that suggests more carries chained up, and if the hundreds column ever hits 10 or more, you’d overshoot 999. So the only way to stop a carry from propagating into a fourth digit is to have no carries at all.

So $c + f \leq 9$, $b + e \leq 9$, $a + d \leq 9$, and since the result is 999, they must actually all be exactly 9. So $a + d = 9$, $b + e = 9$, $c + f = 9$.

Turn 10 [COACH]:
[NO_INTERVENTION]

Turn 11 [STUDENT]: Now the columns as numbers are $10a + d$, $10b + e$, $10c + f$, and those three sum to 99. Since $d = 9 - a$, $e = 9 - b$, $f = 9 - c$, that column-sum equation is $(10a + 9 - a) + (10b + 9 - b) + (10c + 9 - c) = 99$. So that’s $(9a + 9) + (9b + 9) + (9c + 9) = 99$, or $9(a + b + c) + 27 = 99$, so $9(a + b + c) = 72$ and $a + b + c = 8$. Nice, so that matches what I thought.

Turn 12 [COACH]: Now that you have $a + b + c = 8$, what do you still need to figure out? *[CHECK_PROGRESS]*

Turn 13 [STUDENT]: At this point it’s just counting how many triples (a, b, c) of digits give sum 8. They’re digits 0 to 9, but since they sum to 8, none can exceed 8, so that upper bound doesn’t really bite. This is the standard non-negative integer solutions thing, but I always mix up the exact formula, can you just remind me what it is instead of me re-deriving it?

Turn 14 [COACH]:
[NO_INTERVENTION]

Turn 15 [SYSTEM]: *Content hint:* For the equation $x_1 + x_2 + \cdots + x_k = n$ in non-negative integers, the number of solutions is $\binom{n+k-1}{k-1}$.

Turn 16 [STUDENT]: Right, so here $k = 3$ and $n = 8$, so the number of (a, b, c) triples is $\binom{8+3-1}{3-1} = \binom{10}{2} = 45$. And for each triple, d, e, f are forced as $9 - a, 9 - b, 9 - c$. So that gives 45 grids. I’m basically done.

Turn 17 [COACH]: How can you check that every triple you just counted actually produces a valid grid under both original conditions? *[CHECK_CONSISTENCY]*

Turn 18 [STUDENT]: For any (a, b, c) with $a + b + c = 8$, set $d = 9 - a$, $e = 9 - b$, $f = 9 - c$. Since $0 \leq a, b, c \leq 8$, these are between 1 and 9, so all digits are valid. The row sum: top is $100a + 10b + c$, bottom is $100(9 - a) + 10(9 - b) + (9 - c)$. Add them and the a, b, c terms cancel and I get

$900 + 90 + 9 = 999$, so the row condition is always met.

Turn 19 [COACH]: And how do you see the column-sum condition playing out for those same choices? [CHECK_CONSISTENCY]

Turn 20 [STUDENT]: For the columns, the three numbers are $10a+d$, $10b+e$, $10c+f$. Plugging in $d = 9 - a$, etc., I already did that algebra: it became $9(a + b + c) + 27$. Since $a + b + c = 8$, that's $9 \cdot 8 + 27 = 72 + 27 = 99$. So both conditions are built into the way I reduced it.

Turn 21 [COACH]:
[NO_INTERVENTION]

Turn 22 [STUDENT]: Let me sanity-check with an example: say $a = 1$, $b = 2$, $c = 5$, they sum to 8. Then $d = 8$, $e = 7$, $f = 4$. Grid is 1 2 5 on top and 8 7 4 on bottom. Rows: $125 + 874 = 999$. Columns: $18 + 27 + 54 = 99$. So that fits perfectly.

Turn 23 [COACH]:
[NO_INTERVENTION]

Turn 24 [STUDENT]: Only other thing is: are leading zeros okay? The example grid has 0 0 8 in the top row and 0 in the tens places of some columns, so clearly zeros out front are allowed. That means a or b can be 0 and columns like 09 are fine, so I'm not overcounting by letting digits be zero.

Turn 25 [COACH]: Given all that, what final conclusion are you drawing? [CHECK_PROGRESS]

Turn 26 [STUDENT]: So the number of valid grids equals the number of non-negative integer solutions to $a + b + c = 8$, which is $\binom{10}{2} = 45$. Each such triple gives exactly one grid via $d = 9 - a$, $e = 9 - b$, $f = 9 - c$. So the answer is 45.

Turn 27 [COACH]: Looking back, which step in your reasoning would you most want to double-check on a timed contest so you don't lean too much on memory? [REFLECT_OUTCOME]

Turn 28 [STUDENT]: Honestly, the jump from "looks familiar" to "it must be $a + b + c = 8$ and stars and bars." I caught myself assuming that and then went back to justify the no-carry logic and the column equation. Next time I should

deliberately write out the digit equations first and only then plug in any counting formulas I remember.

Part D. Trace-to-Dialogue Mapping.

The dialogue exposes a step the reasoning trace works through silently but a recall-mode student would skip: the no-carry reduction. Where the trace is methodical, the dialogue is opportunistic; the coach's first move drags the dialogue back to the trace's level of rigor without surrendering the formula at the end. The trajectory in Part B is the explicit bridge: the OVER calibration profile predicts the premature $\binom{10}{2}$ assertion, an early check_consistency recovers the missing no-carry justification, and a later check_consistency forces a digit-validity audit before the answer is accepted. The two check_consistency moves bookend the derivation rather than appearing redundantly, which is the planning property the trajectory is meant to enforce.

- **[DEFINE] and [BLOOM]** → Turns 1 and 3. The trace sets up the algebraic conditions cleanly and then reparses the example to confirm that columns are two-digit numbers 09, 09, 81, not the literal digits 0, 0, 8. The dialogue compresses this: the student recognizes the problem shape ("I've seen this kind of thing") and proposes the final formula ($\binom{10}{2} = 45$) immediately, betting on the answer rather than the derivation. The monitor_plan in turn 2 prompts the student to externalize the equation setup, which surfaces the bet explicitly.
- **[CYCLE 1: no-carry reduction]** → Turns 4 to 9. This is the segment that distinguishes a coached solution from a recalled one. The coach's check_consistency in turn 4 ("what assumptions are you making when you jump straight to that kind of equation?") refuses to advance until the premise is on the table. Turn 5 names the missing assumption (no carries), turn 7 begins the formalization, and the uncertainty_probe in turn 8 forces

the student to walk a hypothetical carry through to its consequence — concluding that all three place-value sums must equal 9 exactly. What the trace folds into a single CYCLE block, the dialogue unpacks across three student turns driven by two targeted coach interventions, exactly mirroring the asymmetry between an expert’s silent verification and a learner’s externalized derivation.

- **[CYCLE 2: collapse to a single equation]** → Turn 11. Once the no-carry reduction is in hand, trace and dialogue converge: the student substitutes $d = 9 - a$, $e = 9 - b$, $f = 9 - c$ into the column-sum and arrives at $a + b + c = 8$ in a single move. The moment this reduced equation matches the formula proposed back in turn 3 is precisely where the student’s pattern match becomes justified — a retroactive validation that the coach refused to grant earlier.
- **[CYCLE 3: count and verify digit ranges]** → Turns 13 to 24. The trace counts $\binom{10}{2} = 45$, checks that the digit-bound $a, b, c \leq 9$ is non-binding, verifies $d, e, f \in \{1, \dots, 9\}$, and spot-checks $(1, 2, 5)$. The dialogue distributes these checks across multiple coach prompts: `check_progress` (turn 12) cues the student to recognize counting as the remaining task; the `SYSTEM` content hint (turn 15) supplies the compositions formula since recall is shaky; two `check_consistency` prompts (turns 17, 19) drive separate verifications of the row and column conditions; the digit-validity argument and the $(1, 2, 5)$ spot-check appear in turns 18, 22, 24.
- **[FINAL]** → Turns 26 to 28. The trace closes with 45. The dialogue closes with `reflect_outcome` (turn 27), which converts the episode into a contest-relevant heuristic: write out the digit equations first, and only then

plug in remembered counting formulas. The trace cannot produce this generalization because it never failed; the dialogue can, because the student’s pattern-matching attempt was caught and converted into the no-carry derivation in real time. This asymmetry — *failure-then-repair generates transferable lessons that smooth reasoning does not* — is the pedagogical payoff the entire MetaCLASS pipeline is built to capture.

E Benchmark Prompts

We provide the system prompts used for the two prompt conditions in our Coach Move Prediction benchmark. The **Full prompt** provides detailed definitions for each of the 11 coach moves, including their MAI factor mappings, example utterances, move selection guidelines, and explicit coaching rules. The **Minimal prompt** lists only the move names grouped by metacognitive category, without any definitions, examples, or selection guidance, testing whether LLMs can infer appropriate coaching actions from move names alone.

Full Prompt

You are predicting the next coach move in a metacognitive tutoring conversation. The coach uses metacognitive prompts to help students reflect on their problem-solving process. The coach NEVER gives content (formulas, facts, answers) or correctness feedback.

COACH MOVES (11 total)

Each move maps to a specific metacognitive regulation skill from the Metacognitive Awareness Inventory (MAI).

Planning Moves

...

Monitoring Moves

...

Debugging Moves

...

Evaluation Moves

...

Dialogue Moves

...

HELP SYSTEM

Students can request hints during the conversation.

- SYSTEM speaker delivers the hint (not the coach)
- Two hint types: CONTENT or SCAFFOLDING

MOVE SELECTION GUIDELINES

- NO_INTERVENTION should be ~35-50% of coach turns
- Use UNCERTAINTY_PROBE when student shows hesitation
- Use CHECK_CONSISTENCY when student may have an error
- Use STRATEGY_ALTERNATIVE when current approach fails
- Use PROMPT_RESOURCE for AVOIDANT students stuck without asking for help
- Use REFLECT_OUTCOME after student reaches a solution

RULES

1. Coach NEVER gives content (formulas, facts, answers)
2. Coach NEVER gives correctness feedback
3. Coach ONLY asks metacognitive reflection questions
4. Keep prompts SHORT (1 sentence)

OUTPUT FORMAT

Respond with valid JSON:

```
{"move": "<move>", "response": "<coach utterance>"}
```

Minimal Prompt

You are predicting the next coach move in a metacognitive tutoring conversation. The coach uses metacognitive prompts to help students reflect on their problem-solving process. The coach NEVER gives content or correctness feedback.

VALID COACH MOVES (choose one)

Planning: MONITOR_GOAL, MONITOR_PLAN

Monitoring: CHECK_PROGRESS, CHECK_CONSISTENCY, UNCERTAINTY_PROBE

Debugging: STRATEGY_ALTERNATIVE, REPRESENTATION_SHIFT, PROMPT_RESOURCE

Evaluation: REFLECT_OUTCOME

Dialogue: CONTINUE_PROMPT, NO_INTERVENTION

HELP SYSTEM

SYSTEM speaker delivers hints when students request (CONTENT or SCAFFOLDING type).

OUTPUT FORMAT

Respond with valid JSON:

```
{"move": "<move>", "response": "<coach utterance>"}
```

F Case Study: NO_INTERVENTION → REFLECT_OUTCOME

To get a finer-grained sense of where predictions diverge from ground-truth coach moves, we examine two cases drawn from the most frequent confusion cell, NO_INTERVENTION → REFLECT_OUTCOME (oss-120b, GSM8K, prompt_full). Each pairs a student turn with the predicted and ground-truth moves, and points to a different surface cue that appears to trigger the spurious intervention.

Case 1. Reflection-on-reflection redundancy

Problem. Helena’s serum grows an extra arm every 3 days and an extra leg every 5 days; how many new limbs after 15 days?

Student: ... “even if day 15 shows up on both lists, that’s still two separate limbs that grow. So **I feel better saying** $5 + 3 = 8$ new limbs total.”

Ground-truth	coach	move:
NO_INTERVENTION	(let the answer land)	

Predicted coach move: REFLECT_OUTCOME

Predicted response: “*How do you feel about that answer?*”

The student’s utterance already contains the affective reflection the probe is designed to elicit (“I feel better” vs. “How do you feel”), so the predicted move requests content the student has just supplied. The prediction appears to track the turn’s terminal shape—a numeric value followed by a confident period—rather than its content.

Case 2. Post-monitoring absorption

Problem. Mark edits 1,000 sentences/week split equally between two publishers; publisher A pays 5¢ per sentence and B pays twice A. Compute weekly pay in cents.

Coach (two turns earlier, CHECK_CONSISTENCY): “How could you quickly check whether 7500 makes sense, using just rough reasoning about the rates and totals?”

Student: “If both paid 5¢, he’d make 5000. If both paid 10¢, it’d be 10000. **One is low and one is high, so the total should be between 5000 and 10000, and 7500 is right in the middle.** From A he gets 2500 and from B

5000, so B pays exactly twice as much as A, which matches. So yeah, 7500 is consistent.”

Ground-truth	coach	move:
NO_INTERVENTION	(the student’s own verification is the reflection)	

Predicted coach move: REFLECT_OUTCOME

Predicted response: “*What part of your approach helped you reach the answer most effectively?*”

Prompted by a CHECK_CONSISTENCY move two turns earlier, the student has just carried out a consistency audit (range check plus ratio check) that itself surfaces why the answer is trustworthy. The predicted probe asks the student to revisit that work in a procedural register, departing from the “How do you feel...” template seen in Case 1—a different surface cue (a recently closed audit) appears to trigger the same spurious intervention.