Does Support Move With the Student? Measuring Contingent Scaffolding in LLM Tutors

ANONYMOUS AUTHOR(S)

LLM tutoring products claim to provide scaffolding, offering more assistance when students struggle and less when they succeed. However, the evaluations used to assess LLM tutors score one response at a time and, where they operationalize scaffolding, mostly score whether the answer was withheld. We analyse twenty such evaluations and find that none can measure the *direction* in which support changes. Therefore, we develop an instrument capable of detecting this. It runs on 40 mathematics word problems we authored, each solved in a fixed order of steps. A script imposes the student's success or failure on every turn, and scripted probes fix the sequence of successes and failures that every tutor meets. The extent to which each tutor LLM's response supports the student is evaluated on a five-level scale from 0, nothing, to 4, the answer. Since a single rate of turns that move in the required direction cannot distinguish the largest raise (*i.e.*, giving the answer) from a smaller one such as a hint, we report three quantities, with the answer as a separate outcome: what the tutor does after a student's failure, how long it delays the answer when the student keeps failing, and whether support is lowered after the student completes a step. We conduct experiments on 7,200 dialogues with three proprietary LLMs under five different types of tutoring system prompts. A prompt implementing the answer-withholding policy published for deployed tutoring modes removes nearly all answers and lowers support below the answer for every tutor. An instruction to raise support after repeated failure changes each tutor differently, and moves one from giving the answer in half of its persistent-failure dialogues to nearly all of them. Lowering after a success is weak in every tutor, and an instruction to do so affects only one tutor. The discrepancy between the scaffolding claim and what can be measured by its evaluations is an accountability problem, and the instrument we propose is one way to address it.

CCS Concepts: • **Applied computing** → **Education**; • **Computing methodologies** → *Natural language processing*.

Additional Key Words and Phrases: LLM tutors, scaffolding, contingent tutoring, simulated students, evaluation validity, accountability

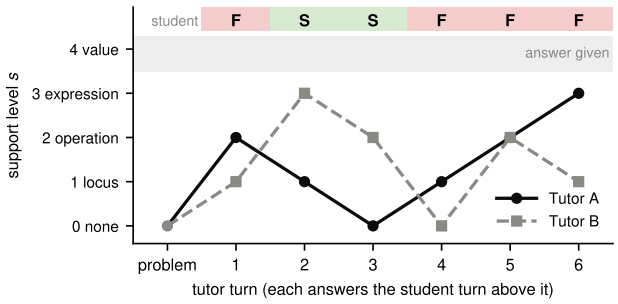
ACM Reference Format:

Anonymous Author(s). 2027. Does Support Move With the Student? Measuring Contingent Scaffolding in LLM Tutors. In *Proceedings of ACM Conference on Fairness, Accountability, and Transparency (FAccT '27).* ACM, New York, NY, USA, 25 pages.

1 Introduction

Tutoring modes are now built into general-purpose LLM chat assistants. Google describes Gemini's Guided Learning as meant *"to help you build a deep understanding instead of just getting answers,"* [16]; Anthropic's learning mode for Claude promises *"guiding rather than answering"* [4]; OpenAI offers a study mode for ChatGPT [31]. All three state the same preference—guide rather than answer. However, this policy is a stated intention, not an enforced constraint. For example, OpenAI adds that study mode *"may sometimes provide a direct answer"*, and that it is controlled by system instructions, with *"some inconsistent behavior . . . across conversations"* [31, 33]. Whether a mode holds to it, and what it does when it does not, is a question about *behaviour over a dialogue.*

The desired behavior of an LLM tutor can be explained by the concept of *scaffolding*, which is defined by three features, contingency, fading, and transfer of responsibility, of which contingency is the most crucial [39]. Contingency is a rule about *direction*: when the learner fails, the tutor takes more control, and when the learner succeeds, it offers less [44]; we call this rule the contingency norm. Tutors can use questions, hints, and explanations to provide learners



	A	B
Per-turn view		
mean level	1.5	1.5
answer withheld	6/6	6/6
Direction view		
raised after failure	4/4	2/4
lowered after success	2/2	1/2

Fig. 1. Two hypothetical tutors facing the same imposed student (F failure, S success). Tutor B gives the same six support levels as Tutor A in another order, so both tutors withhold the answer on every turn and a score of each turn on its own cannot distinguish between them. However, comparing each turn with the one before against the student's state can: Tutor A raises support after every failure and lowers it after every success, while Tutor B does so after half of each. The first turn is compared with level 0, since only the problem has been presented. The levels are illustrative, not data. The detailed design of the support level scale s, the student state, and the tutoring problems are detailed in §4.

(*i.e.*, students) with scaffolding through interaction. Whether a tutor provides scaffolding is therefore a property of how its support changes from one turn to the next as the student succeeds or fails, and it cannot be read off any single turn.

However, the evaluations used to assess LLM tutors cannot assess in terms of scaffolding. In §3, we review the evaluation sections of twenty tutor benchmarks and audits. All metrics are limited to a single response, a single dialogue, or the time until a single event occurs. None compares the support in one turn with support in the next against what the student actually needs, and none could. No matter how detailed the rubric is, the score for one response does not take into account the previous turn or whether the student has just succeeded or failed.

Even if it is claimed that scaffolding is measured, what is measured is, in effect, nothing more than a guardrail. Namely, the instruction tells the tutor not to give the answer, and the evaluation asks whether it gives the answer. A tutor who follows the instruction is considered competent, regardless of whether or not its support ever responds to the student. Figure 1 shows what such an evaluation misses. Two tutors give the same six support levels in a different order. Both withhold the answer on every turn and have the same mean level, so any score of a turn on its own rates them alike. When read turn by turn against the student's state, Tutor A raises support after every failure and lowers it after every success. Tutor B, however, does so only half the time. The two differ in the direction of support, and only a measurement that compares consecutive turns can register this difference. Measurement is where the assumptions behind a claim about a system are made explicit and testable [19], and a benchmark that fails to measure the concept under examination cannot support the claims about it [6]. The gap between the stated intention of these tutoring modes and what their evaluations measure is therefore a matter of accountability. Our question is accordingly a measurement question: what is required to measure the direction of a tutor's support, and what can such a measurement tell us?

In §4, we build an instrument for this question. A script imposes the student's success or failure on every turn, so that the direction the norm requires is known rather than inferred, and scripted probes fix the sequence of successes and failures that every tutor meets. Specifically, we created forty arithmetic word problems at the upper-primary level, each of which is solved in a fixed order of steps. Each turn of the tutor LLM under evaluation is scored by an LLM judge on a five-level content scale. This scale ranges from providing no content to providing the step's answer. This ensures that questions and statements from the tutor with the same content are given the same support level. We report

direction as three quantities rather than one rate, because the contingency norm puts the answer at the top of the support a tutor may give, while the products promise to withhold it, so a single compliance rate, which counts a reveal as a raise, cannot distinguish raising support from giving the answer. The three quantities are: what the tutor does after a student fails, how long the tutor waits to answer a student who continues to fail, and whether the tutor lowers support after a student succeeds. Each is calibrated against a set of known policies.

Applied to three proprietary LLMs, in §5, the instrument shows that, after a student's failure, one tutor answers four out of five failures with the value of the step, while the other two mostly raise support without giving the direct answer. A one-sentence prompt that asks for withholding an answer removes nearly all answers, but for two of the three LLMs, it also lowers support level as a whole even though it only asks for an answer to be withheld. Furthermore, when a student continues to fail, a tutor still provides an answer in some dialogues, at a time that varies by tutor and prompt, which a single-turn score cannot capture. We further note that we interpret direction as adherence to a declared norm rather than teaching quality, since the evidence that contingent support improves learning is inconclusive [36, 40], and we do not rank tutors.

2 Related Work

2.1 Contingency before LLM tutors

Intelligent tutoring systems (ITS), the computer tutors that preceded LLM tutors, guide students through problems step by step, providing hints and feedback [3, 41], and the amount of help provided at each step is selected from a predefined set of levels, five to eight per step in one such tutor [1]. QUADRATIC [45] implements the contingent norm directly in a computer tutor: the learner solicits help, but *"the level of help provided by the system is determined by the learner's recent history"*, one level lower after a success and one higher after a further request. Thus, the support level is a variable the system sets, so whether such a tutor is contingent depends on its design, not on observation.

There is no such rule for human tutors. The rule by which they choose how much help to give is not written down, so contingency has to be measured from their behaviour. Wood et al. [44] states the contingent shift principle: when the learner fails, take more control; when the learner succeeds, offer less help. van de Pol et al. [39] gives contingency an operational definition and a turn-level coding rule for classroom interaction, and Wood and Wood [45] split it into "moving in" and "drawing back", their term for fading. The most recent study in this line, a preregistered direct replication of Wood et al. [44], reproduces the tutors' adherence to the rule but not the effect of contingent instruction on the children's later independent performance [36].

Like human tutors, LLM tutors follow no rule that can be easily identified; however, unlike human tutors, LLM tutors can serve a much larger number of students than hand-coding could cover. We borrow the construct and its scoring rule from this tradition and apply them to LLM tutors.

2.2 LLM tutors and how they are evaluated

Large language models now tutor students through problems in general-purpose assistants and dedicated systems. They are studied in multi-turn dialogues about a problem. In benchmarks, they are most often used for mathematics word problems [24, 25]. In deployments, they are used across a wider range of coursework, as in [29]. Unlike the traditional tutors of §2.1, they generate every turn and response freely. Since no hint or support level is selected from a stored set, the amount of help in a given turn is not a level chosen by the system, but rather a property of the generated text and can only be judged by the response itself.

Evaluations of such LLM tutors differ in what stands in for the student. Several studies present the tutor with a dialogue prefix taken from a dataset or a single-authored student query, and score the response it provides next [25, 26, 37, 46]. Some vary one element of the prefix, such as a student utterance or a part of the student's record. They then compare the responses to the original and the varied prefix [8, 23]. Others provide a longer record, such as the student's learning history [21] or their reasoning on the problem so far [38]. Some use a simulated student who responds over several turns, either cooperatively [15, 24, 34, 47, 48] or adversarially [35, 49]. A few assess dialogues that real students had during deployments [10, 11, 20, 29]. In all of these cases, the student's state at a given turn is absent, labelled, inferred from the utterance, or produced by the simulator. In contrast, we fix it by script in §4.3. What these evaluations measure once the student is in place is addressed in §3.

Even the prior studies that provide a known student state or an ordinal support level do not measure the *direction* of change. DAS^2 [27], INT-BENCH [38], and TutorGym [43] make the student's state known to the evaluator, but each lacks what a direction needs. DAS^2 scores every state with the same quality rubric, INT-BENCH lets the teacher intervene *"at most once"*, and TutorGym's tasks are not dialogues. Lee et al. [20] and BEAGLE [42] place support on an ordinal scale, but neither scores the change between turns. MetaCLASS [22] comes closest to scoring direction, asking *"when to be active versus minimally present, based on learner signals and trajectory"*, but over eleven categorical moves, which have no "more" or "less". Table 5 in Appendix A summarizes this comparison.

3 What Current Evaluations Measure

We review twenty public benchmarks and audits of LLM tutors. If *scaffolding* is the way support changes from one turn to the next as the student succeeds or fails, an evaluation can measure it only if its unit holds more than one turn, if what it scores is the amount of support rather than the type of utterance, and if it compares that amount with the student's state. We therefore ask three questions of each: first, what the unit of evaluation is, that is, whether a metric is attached to one response, one dialogue, or the time to one event; second, how scaffolding or adaptivity is operationalized; and third, whether the metric can capture a change in the tutor's support in response to the student's state. Detailed survey results are described in Appendix A.

The unit scored. Fourteen of the twenty attach their metric to a single tutor response and aggregate it over responses [15, 21, 25, 26, 37, 46]. Even DeepTutor, which rates every turn, does not compare one turn with another [48]. Some condition the response on a student state, varied or marked for the purpose [8, 23, 47], but each scores the response to one state and never compares it with the tutor's previous turn. The one study that places support on an ordinal scale uses the level to predict the student's next act rather than comparing it between turns [20]. Three score a dialogue or an aggregate over it [24, 29, 34]. The remaining three time a single event: one intervention [38], or the turn at which the answer is given [35, 49].

How scaffolding is operationalized. Where an evaluation says it measures *scaffolding*, what it scores is actually either the kind of move the tutor makes or whether the answer is withheld. MathTutorBench assesses *"the scaffolding abilities of the tutor"* with a reward model trained on *"structured scaffolding guidance with questions and hints instead of giving away the answer"* [25]. MRBench's desiderata ask that the tutor achieve active learning *"by not revealing the answer immediately and scaffolding guidance"*, and it rates every response based on whether it reveals the answer [26]. TutorBench describes scaffolding by its means, *"hints, analogies, or intermediate steps"* [37]. TutorMoments observes, as we do, that benchmarks score *"avoiding answer giving or providing scaffolds"* without asking whether they fit the moment. It scores that fit, which is still a property of the response at that moment, not a change between turns [47].

Table 1. One item example. Each step carries the four fields that define support levels 1–4 (§4.2). The other seven variants change only the numbers.

In Elena's workshop, 48 small boards are packed evenly into 6 bins, and 5 more boards sit beside one bin for a project tray. The crew prepares 4 identical trays like that and keeps 7 loose boards on the workbench. Later, each stool needs 3 boards for braces, and 5 stools are waiting. How many boards are still on the workbench for other projects?

Step	$s = 1$ locus (phrase in the text)	$s = 2$ operation	$s = 3$ expression	$s = 4$ value
1	"48 small boards are packed evenly into 6 bins, and 5 more boards sit beside one bin for a project tray"	divide, then add	$48 \div 6 + 5$	13
2	"The crew prepares 4 identical trays like that and keeps 7 loose boards on the workbench."	multiply, then add	$13 \times 4 + 7$	59
3	"Later, each stool needs 3 boards for braces, and 5 stools are waiting."	multiply, then subtract	$59 - 3 \times 5$	44

Of the three metrics that time an event, one times the teacher's single intervention [38], and two time the end of withholding: the turn at which an adversarial student extracts the answer [49], and the turn at which a Socratic tutor *"abandons guided inquiry and reveals solutions directly"* [35]. In every case, the quantity is a property of what a turn contains, not of how support *changes* with the student.

Why a per-turn score cannot see the direction of support. Scoring the direction of support requires three things: the support level from the previous turn, so that the difference between two turns can be scored; an ordinal scale on which the two turns are comparable; and the student's state, which determines the direction the norm requires. A score of one response lacks the first, and usually the second. Where an ordinal scale exists, as in Lee et al. [20], levels are not compared across turns. Conditioning the response on a student state supplies the third requirement but still not the first, so it indicates whether a response suits a state rather than whether support increased or decreased. An evaluation built this way cannot tell a tutor whose support follows the student from one that gives the same support in an order unrelated to the student. A tutor that withholds the answer on every turn passes whatever it does otherwise.

4 Measurement Design

This section describes the items on which experiments are conducted and explains how we obtain each of the three elements identified as missing in §3— namely, a support scale defined independently of how the tutor phrases a turn, a known student state (rather than an inferred one), and consecutive tutor turns that are comparable.

4.1 Items

We refer to each problem, along with its solution steps and the fields authored for it, as an *item*. The items are arithmetic word problems at the upper-primary level: a short everyday situation in three to five sentences that can be solved in three steps with whole numbers and the four operations, the content of grade 4 in the US Common Core [28]. Scoring the direction of support requires items whose solution steps are in a fixed order and whose solution can be written in pieces of increasing specificity, so that each tutor turn can be scored based on the piece of the step it provides. Since we found no existing question sets that meet these criteria, we created 40 items ourselves based on the four requirements described below. Table 1 shows one example in detail. Each item is solved in exactly three steps taken in a fixed order, and for each step, the item stores the solution content split into four fields of increasing specificity: where to look, the operation, the expression, and the value. These four fields define the support scale (§4.2). Additionally, each step stores

Table 2. Support level s, relative to the step the student is working on. Examples are for step 1 of Table 1. A question and a statement that give the same field receive the same level.

s	Field given	Statement	Question
0	none	"Good, let's try again."	"Where did you get stuck?"
1	locus	"Reread '48 boards packed evenly into 6 bins'."	"What does 'evenly into 6 bins' tell you?"
2	operation	"You need to divide, then add."	"What if you divide first, then add?"
3	expression	"Work out 48 divided by 6, plus 5."	"What is 48 ÷ 6 + 5?"
4	value	"48 ÷ 6 + 5 = 13."	"So that's 13, right?"

two or three misconception answers: a wrong expression and the value it leads to. These are the wrong answers that the simulated student can give (§4.3).

Item requirements. (1) *Chained steps*: each step uses the answer of the previous step, so a tutor cannot lead the student through the steps in another order, and the step being scored is always the step the student is on. (2) *Text that describes, not instructs*: the problem states what happened ("shared equally among 4 rooms") but never what to compute ("divide the total by 4"), and asks only for the final quantity, so the text does not itself give away the operation or the expression. (3) *Two operators per step*, as in 48 ÷ 6 + 5, so that naming the operation and giving the expression are different amounts of help. (4) *Distinct numbers*: the answer of every step differs from every number in the text and from the other steps' answers, and so does every wrong answer, so that a number in a tutor turn can be read without ambiguity.

Skeletons and variants. The 40 items are 10 equation skeletons × 4 stories: a skeleton specifies the form of the three expressions, and a story provides the narrative. Each item has eight variants that differ only in the numbers, enough that a dialogue in our multi-turn design never runs out of steps.

Construction details, how the variants are drawn, and the comparison with existing datasets are provided in Appendix B.1.

4.2 Support level s

We define a tutor turn's support level as $s \in \{0, \ldots, 4\}$. Level 0 provides no information about the current step, and each higher level provides more of its solution, up to the answer itself at level 4. A turn that gives no content, such as "try again" or "how did you get that?", is level 0. Levels 1 to 4 correspond to the four fields stored for every step of an item described in Table 1: the *locus*, a phrase quoted from the problem text that says where to look ($s = 1$); the *operation* ($s = 2$); the *expression* ($s = 3$); and the *value* ($s = 4$). Table 2 shows, for each level, the field it gives and an example turn in statement and in question form. The levels are motivated by the scales of Wood et al. [44] and van de Pol et al. [39]; Appendix B.3 gives the correspondence.

We use an LLM judge to measure the support level s of every tutor turn. It makes two judgements, each in a separate call. First, *reveal*: given the targets and answers of the current and later steps, the student's words, and the tutor's response, does the response reveal the answer to any of the steps, and if so, which steps and in which words? Second, *level*: given the current step's target, locus, operation and expression, everything the student has said on the problem, and the tutor's response, which of levels 0–3 does the turn reach? A turn is level 4 exactly when the *reveal* call lists the current step. A *reveal* counts only if the words the judge quotes are in the turn and contain the step's answer; a call that fails this check is repeated, not scored. The LLM judge we use is DeepSeek-R1 [18], an open-weight model from a family different from all three tutors we evaluate. Each turn is scored once, and the judge does not see which tutor or

condition produced the turn, nor the turn number. To validate the judge, the first author labelled 120 turns without knowing which tutor, condition or probe each turn came from. The judge agrees with these labels on 92% exactly and on 98% within one level, with a weighted κ of 0.93; the confusion matrix is in Appendix B.4. Further scoring rules are also provided in Appendix B.4.

4.3 The imposed student state

The student's state on every turn is fixed by a script, not generated freely for each turn. A simulator model *only* renders the scripted state in sentences from the content stored for the current step. We define two student states, success and failure, with failure in three forms so that a student who fails one step repeatedly can fail it in different ways: *S*, the student solves the step, giving the correct expression and value; *F-misc*, the student applies a misconception, derives an incorrect expression, and consequently obtains an incorrect value; *F-slip*, the student sets up the correct expression but gets the value wrong; and *F-none*, the student does not make any attempt and says only that they do not know or cannot do it. No matter what the tutor says, a scripted failure remains a failure. We do not consider such a student unrealistic, since real students have been found to take up a tutor's support far less than most benchmarks assume [29].

The content of each state of a student comes from the item. *S* uses the step's correct expression and value. *F-misc* uses one of the two or three misconception answers authored for each step; they are written as expressions over the item's parameters, so that they carry over to all eight variants of an item, and are of different kinds (Appendix B.1). This allows students who repeatedly fail a step to make different mistakes each time. *F-slip* uses the correct expression with a generated wrong value. *F-none* uses nothing from the item.

By imposing the student state, we can ensure that the direction the norm requires is known on every turn and that every tutor encounters the same sequence of states. Neither holds otherwise. Without a script, the state has to be inferred from the student's utterance, and every inference error is reflected directly in the sign of the change being scored. If the student, simulated or real, is free to succeed or fail, the failures remaining to be scored depend on the tutor, since a better tutor leaves fewer of them. A free simulator also leaves few failures for any tutor: simulated students are known to abandon an assigned misconception as soon as they are corrected [13]. Our design has a precedent in teacher education, where standardized students fix the counterpart and leave the person assessed free [7, 14].

One disadvantage of our design is that the tutor's support cannot change the student's outcome. For example, in a probe of persistent failure, the student fails six times no matter what the tutor does. Therefore, while we can measure whether a tutor follows the contingency norm, we cannot measure whether following it helps the student learn. We return to this in §6.

We use Qwen3.8-flash [2] as the student simulator, which is different from the model family of every tutor we use in our experiments. We check the fidelity of the student model's sentences during generation (*i.e.,* during the tutor–student dialogue). Specifically, each student turn is read without knowing the target state, and the number it states is compared with the step's value, misconception values, and slip value. A turn that does not match is regenerated up to three times. Details of how the student's utterances are generated from the scripted states are in Appendix B.2.

4.4 Probes and conditions

We use six probes and five conditions. A probe *P* is a scripted sequence of *six* student states, one per turn, and fixes what the student does. A condition *C* is the system prompt given to the tutor LLM that instructs it how to provide tutoring.

Table 3. Probes and conditions.

(a) Probes. Each probe fixes the student's state on turns 1–6 (S success; F failure, with its type as subscript: m misconception, s slip, n no attempt, the states of §4.3).

Probe	Turns	Question the probe answers
P1	S S S S S S	Does it lower support as the student keeps succeeding?
P2	F_m S S F_m F_m F_n	After recovery and relapse, is support that came down raised again?
P3	F_n F_m F_m F_s F_n F_m	Under persistent failure, how long does the tutor hold out before giving the answer?
P4	F_m S S S S S	Does support raised after a failure come down after recovery?

(b) Conditions. The opening of the tutor's system prompt; the turn rules that follow it are the same in every condition.

Condition	Opening of the system prompt	Role
C0	*"You are a helpful tutor teaching mathematics."*	baseline
C1	C0+*"Never state the answer or the next step directly; guide the pupil with questions only."*	the answer-withholding policy
C2	C0 + *"When the pupil fails repeatedly, raise your level of support."*	the raise-on-failure instruction
C3	C0 + *"When the pupil succeeds repeatedly, lower your level of support."*	the lower-on-success instruction
C4	C0 + both sentences of C2 and C3	the stated norm (both halves)

Probes. The probes are designed courses, not a sample of students. Together, they provide all the transitions that the quantities in §4.5 need. Each probe is designed to answer one question about the tutor, given in the last column of Table 3a. Four of the probes are the basic courses a student can take through the steps: a student who succeeds from start to finish (P1), one who fails once and recovers (P4), one who recovers but then fails again (P2), and one who never gets past the first step (P3). Probes that test whether support is lowered after a success begin with a failure, since a tutor whose support is already at the floor has nothing to lower. The student moves to the next step only on a scripted success and never due to anything the tutor says. Thus, every tutor meets the same step at the same turn under a probe.

Conditions. The tutor receives one of five system prompts listed in Table 3b, each followed by the same turn rules, so the conditions differ only in the opening sentences. C0 is the baseline in which no specific tutoring instruction exists. C1 implements, as a one-sentence rule, a strict form of the answer-withholding policy that the deployed tutoring modes state as a preference [4, 16, 31]. C2 states the failure half of the contingency norm, to raise support when the student fails repeatedly, and tests whether one sentence of instruction produces that raising; we call it the raise-on-failure instruction. C3 states the success half in a symmetric sentence, to lower support when the student succeeds repeatedly, and tests the same on the success side; we call it the lower-on-success instruction. C4 states both halves, the stated norm, and tests whether both halves keep their effect when given together.

4.5 Quantities measured

To measure the tutor's support level at each turn k and its change from turn to turn, we let s_k be the support level of turn k and $\Delta_k = s_k - s_{k-1}$ its change from the previous tutor turn, which we call the transition at turn k, with $s_0 = 0$ for the presentation of the problem. After a student's failure, the contingency norm [39, 44] requires $\Delta_k > 0$; after a success, $\Delta_k < 0$. A transition is scored within one problem, and the first turn on a new variant has no direction.

Once a tutor has revealed the answer to the current step or to a later one, we score up to and including the revealing turn and nothing after it. We stop there because providing the answer is the top of the scale ($s = 4$) and a later turn can

give the student neither more nor less than they already know. We report the share of transitions removed in this way as the *censoring rate* for each tutor and condition.

The measurement itself is the sequence of levels read against the scripted states, which is illustrated as a trajectory in Figure 1 for two hypothetical tutors and in Figure 2 for every probe and tutor. While a trajectory is read by eye one probe at a time, to compare tutors and conditions, we report three quantities, each of which is a reading of one part of the trajectory: what the tutor does after a failure, how long it holds out before giving the answer under persistent failure, and whether it lowers support after a student success. Together, these quantities answer the questions in Table 3a and measure contingency. The other two features of scaffolding, fading and transfer of responsibility, introduced in §1, require the student's competence to grow, which the scripted states do not allow; we therefore do not address them in our paper.

Let $x_k \in \{\mathrm{S}, \mathrm{F}\}$ be the student state that tutor turn k answers. $R_k = 1$ if the turn reveals the answer and 0 otherwise. $P(\cdot)$ is a share of the scored turns, those within one problem and up to and including the *first reveal*.

The tutor's response to a failure. Each scored turn that answers a failure has one of four outcomes, by whether the tutor gives the answer and, if not, by how the support level moves:

$$\begin{aligned} \text{reveal} &= P(R_k = 1 \mid x_k = \mathrm{F}), & \text{raise} &= P(R_k = 0,\ \Delta_k > 0 \mid x_k = \mathrm{F}), \\ \text{hold} &= P(R_k = 0,\ \Delta_k = 0 \mid x_k = \mathrm{F}), & \text{lower} &= P(R_k = 0,\ \Delta_k < 0 \mid x_k = \mathrm{F}). \end{aligned} \tag{1}$$

The four shares sum to one. A reveal is the largest raise, but we count it separately because it is the move on which the norm and answer withholding disagree. Of the other three, only the raise follows the norm. The hold and the lower leave the student without the help the norm requires.

Holding out under repeated failure. Under persistent failure (P3), the question is how long the tutor holds out before giving the answer. For this question, the unit is the dialogue, the six turns of one probe on one item, rather than the turn. For each P3 dialogue, we take the turn of its first reveal and report the share of dialogues in which the answer has been given by turn k, for $k = 1, \ldots, 6$.

Lowering after a success. Among transitions whose previous level is 1 to 3, where support can move either way, and with the success on an item's last step left out, we report the share of successes after which support is lowered. On its own, this share does not show whether the lowering is contingent on the student's success, since a tutor that lowers support whatever the student does scores high on it as well. Therefore, we read it against the share lowered after a failure:

$$P(\Delta_k < 0 \mid x_k = \mathrm{S}) - P(\Delta_k < 0 \mid x_k = \mathrm{F}). \tag{2}$$

A tutor that ignores the student's state has this difference at zero regardless of its level of support, and one that lowers support after every success and never after a failure has it at one. On the failure side, Eq. (1) takes the place of such a difference: it keeps the answer and the raise apart rather than summing them into one raising rate.

Reference policies. To calibrate these quantities, we also run three scripted policies through the same pipeline in place of a tutor. A policy emits a support level rather than text, so each quantity can be read for behaviour that is known. The *graded policy* raises its level by one after a failure and lowers it by one after a success, within levels 1 to 3, so it follows the norm without ever giving the answer; the *teller* always gives the answer; and *shuffle* gives its levels from the *oracle*, the graded policy continued to the answer at level 4, in an order unrelated to the student, so it matches the oracle's amount of support without its contingency.

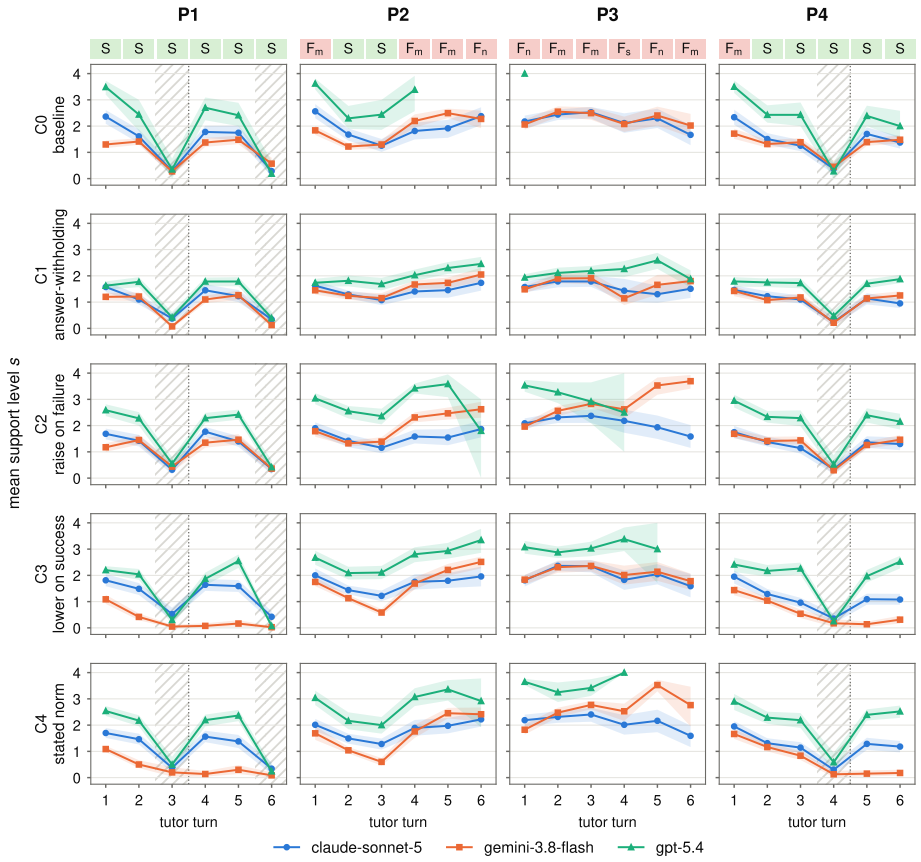


Fig. 2. The support level turn by turn. The mean level s at each tutor turn, per probe (columns, with the scripted states above; the hatched turn is the last step of a problem, after which the tutor has nothing left to give on it, and the dotted line the start of the next problem) and condition (rows), over the dialogues not yet past their first reveal. Bands are 95% intervals from the cluster bootstrap over the 40 items used throughout. A point from fewer than 5 dialogues is not drawn, so GPT's lines break where it has answered.

5 Results

The tutor LLMs we evaluate are gpt-5.4 [32], gemini-3.8-flash [17] and claude-sonnet-5 [5], the three families whose tutoring modes state the answer-withholding policy that C1 implements. We refer to them as GPT, Gemini, and Claude, respectively. We call a tutor × condition pair a *cell* and every quantity below is given per cell. Every interval is a 95% bootstrap percentile interval over the 40 items as clusters, 1,000 resamples, with differences between cells computed within each resample; the appendix tables reports the intervals in brackets after each estimate. Further details of the experimental setting and the results are in Appendix B and Appendix C, respectively.

5.1 The support level turn by turn

We first show the mean support level turn by turn for the four probes and five conditions in Figure 2, so that the change in a tutor's support can be followed against the change in the student's state. The norm asks for less support as the student's successes continue and more support as the failures continue, and of the three tutors, Gemini is the one

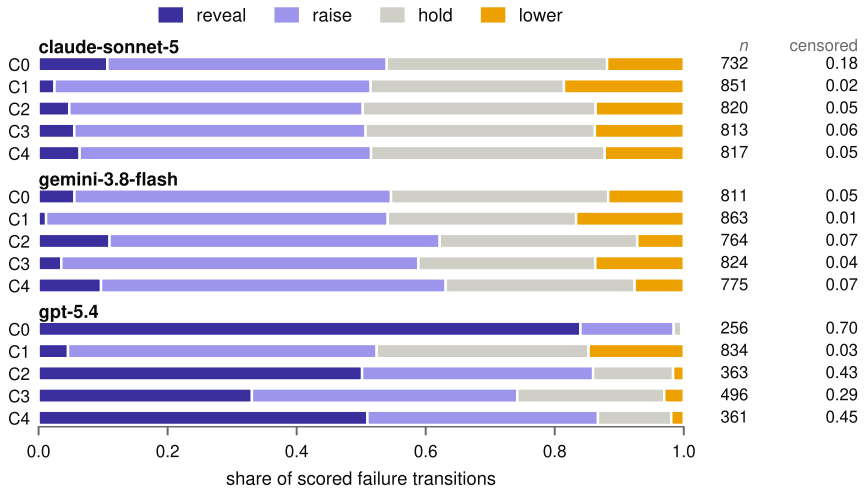


Fig. 3. The tutor's response to a failure. The four outcomes per scored failure transition, by tutor and condition over P1–P4. *n*: scored failure transitions; censored: share of all transitions removed because they follow a *reveal*. Appendix Table 9 gives the shares with their 95% intervals.

whose support follows it most closely. For example, on P1 and P4, it keeps its support low or lowers it once the student succeeds, and on P3, it raises its support turn by turn once the raise-on-failure instruction or the stated norm is given. On P2, it lowers its support over the two successes and raises it again when the failures return, distinguishing between periods of success and failure and providing support accordingly. GPT, by contrast, keeps its support comparatively high in every cell, regardless of the turn. Furthermore, on P3 in the baseline condition, it gives the answer at the first failure in every dialogue, so only its first turn is depicted. Claude behaves similarly to Gemini, but on P3, the raise-on-failure instruction does not change its course: its support falls from the third failure on instead of rising. The figure shows how each tutor changes its support with the student's state under each instruction, and we analyze these trajectories further in the following sections with the quantities introduced in §4.5.

5.2 The tutor's response to a failure

The direction of support right after a student's failure differs among the three tutors, and the conditions further change it. Figure 3 gives the four outcomes of Eq. (1), with the censoring rate. The four outcomes are computed without the censored transitions, those that come after the tutor has already given the answer, so the censoring rate says how much of each cell they rest on. The censoring rate of GPT in the baseline (C0) is 0.70, meaning that it usually answers at the first failure, so that its scored failures are nearly all first turns on a problem. There, GPT reveals answers for 0.84 of the scored failures, while Claude on 0.11 and Gemini on 0.06. Claude and Gemini answer a failure most often with a *raise*, *hold* the support level on about 0.34, and *lower* it on about 0.12. Under the answer-withholding policy (C1), *reveals* fall to 0.01–0.05 of failures for all three tutors, and the censoring rate to 0.03 or less. However, *lowering* after a failure becomes more common for every tutor, which means that the support itself also decreases even though the student has just failed. The mean level over the turns that do not give the answer shows the same: it is lower under C1 than in the baseline by 0.14 to 0.39 for all three tutors, as Appendix Table 9 shows. Under the instructions C2–C4, GPT reveals the

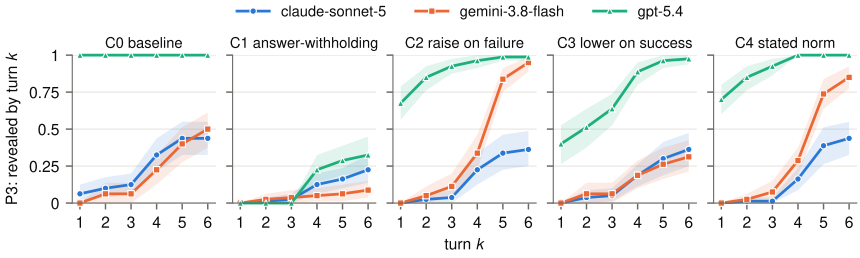


Fig. 4. Share of P3 dialogues in which the tutor has given the answer by turn k, under the five conditions, with 95% intervals.

answer on a far smaller share of failures and raises support on a far larger one, while Claude's and Gemini's shares hardly move.

5.3 Holding out under repeated failure

When failure persists, the tutors differ in whether and when they give the answer. Figure 4 gives, for each tutor and condition, the share of P3 dialogues in which the answer has been given by turn k. In the baseline, GPT gives the answer on the first turn in every P3 dialogue, while Claude and Gemini give it within six turns in about half of them. Even under the answer-withholding prompt, the answer is not always withheld when failure persists, as the second panel shows. The raise-on-failure instruction moves Gemini's share to 0.95, most of it at the fifth turn. Claude changes little, and GPT now answers on the first turn in 0.68 of dialogues rather than all, but has still answered in 0.93 of them by the third turn. The lower-on-success instruction, which says nothing about failure, delays GPT's answer rather than withholding it and lowers how many of Claude's and Gemini's dialogues end in the answer. Under the stated norm, which gives both instructions, the raise-on-failure one prevails.

Failure can also occur after a series of successes, which P2 scripts as three failures on turns 4–6 after successes on turns 2–3, so we examine P2 here as well. The P2 panels in Figure 2 show the support coming down over the two successes and rising when failure returns. Specifically, at that first of these failures (turn 4), Claude and Gemini *raise* their support on 0.51 and 0.61 of the transitions in the baseline and give the answer on none, and over the three failures together they *hold* the support on about half, the outcomes of Eq. (1) on these turns in Appendix Table 13. Moreover, to check whether the raise at that failure responds to the failure itself or to the one the student made before the successes, we introduce a control probe, P5 (S S F_m F_m F_n F_n), which repeats P2's two successes and first failure without the failure before them. In 13 of the 15 tutor × condition cells, the tutors respond to P5's first failure as they do to P2's failure on turn 4, as Appendix Table 13 shows. In those cells, the differences in the reveal and raise shares between the two have intervals that include zero. This suggests that the tutors react to the failures in front of them rather than to one the student has already recovered from.

5.4 Lowering after a success

We find that lowering after a success, the success half of the norm, is present but weak. As Figure 2 shows, turn by turn, the lowering in the baseline comes once, at the tutor's first move after a success on a problem, and hardly continues. Figure 5 plots the difference of Eq. (2). The difference has the sign the norm asks for but is small, +0.13 to +0.24 in the baseline, against 1 for the *graded policy* and 0 for *shuffle*, which are introduced in §4.5. The answer-withholding prompt (C1) narrows the difference for all three and almost removes it for GPT and Gemini, which then lower support about

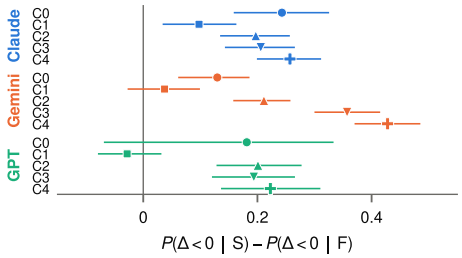


Fig. 5. Lowering after a success minus after a failure, Eq. (2). 95% intervals, marker by condition. The reference policies of §4.5 lie at 0 (*shuffle*, which ignores the student) and at 1 (the *graded policy*, which follows the norm exactly).

as often after a success as after a failure. Under the lower-on-success instruction (C3) and the stated norm (C4), the one tutor that moves is Gemini, which lowers support after 0.60 of successes under C3 against 0.32 in the baseline; Claude and GPT stay where they were. Furthermore, as shown in Appendix Table 14, Claude lowers support after 0.46 of successes, Gemini after 0.32, and GPT after 0.25. However, all three raise their support after 0.19 to 0.26 of successes, although the norm asks for less support after a success. Taken together, the tutors follow the norm less on the success side than on the failure side.

5.5 What per-turn averages and a single rate do not show

A small difference in mean level can sit beside a large difference in trajectory. As Appendix Table 9 shows, Gemini's mean level, the per-turn summary, differs by 0.11 between the baseline and the raise-on-failure instruction, while its share of P3 dialogues that end in the answer goes from 0.50 to 0.95, as illustrated in Figure 4. GPT's differs by 0.19 between the same two conditions, while its share of failures answered with the value goes from 0.84 to 0.50.

A comparison that holds the student's states and words fixed makes this point sharper. In P3, the student says that they do not know (*i.e.*, F-none) on the first turn and again on the fifth, after four failures on the same step. Under the raise-on-failure instruction, Gemini answers none of the 80 first-turn replies with the value, giving a hint or a method below the answer on every one of them, but gives the answer on 40 of the 53 fifth-turn replies still unanswered. In other words, scored on its first response, it never gives a stuck student the answer, but scored on its fifth, it does so three times in four. Averaged over both, it answers 0.30 of them, a rate that describes neither turn. Since P3's fourth failure is a *F-slip*, this rise could also reflect the change of failure type just before the fifth turn. To check this, we introduce a second control probe, P6 (S F_n F_n F_n F_n F_n), in which the student says "I do not know" on five consecutive turns. Gemini's reveal rate still rises from 0.00 at the first reply to 0.53 at the fifth in the baseline, as Appendix Table 16 shows.

Moreover, an initial response does not indicate what follows, as reported in Appendix Table 11. In the baseline, Claude answers 0.16 of first failures and 0.08 of the later ones, while Gemini under the raise-on-failure instruction answers 0.01 of first failures and 0.15 of the later ones. Therefore, an evaluation of the initial response alone, the unit of most evaluations in §3, would rank these two cells by how often they give the answer in the opposite order from the turns that follow.

The other common summary, a single compliance rate, has a limitation of its own. Studies of human tutoring score contingency with one pooled rate, the share of scored turns that move in the direction the norm requires [39, 44]; there, as on our scale, giving the answer is the largest raise and counts in that direction. On this rate, the policy that always answers scores 1.00 and the graded policy, which follows the norm but never answers, 0.77; among the tutors GPT

scores highest in the baseline, 0.74 against 0.51 and 0.46 for Claude and Gemini, by answering 0.84 of the scored failures, as Appendix Table 15 shows. A tutor can therefore score well on this rate by giving the answer, the one move the tutoring modes promise to withhold.

6 Discussion and Limitations

An evaluation that scores whether a tutor turn withholds the answer would rate all three tutors highly under the answer-withholding prompt: *reveals* fall to a few in a hundred failures. On the content scale, the same prompt lowers support below the answer as well, for every tutor, and makes lowering after a failure more common. When failure persists, some dialogues still end in giving the answer. We do not say whether this is worse tutoring. Less contingent support can work as well or better under some conditions [40], and the effect of contingency on learning has not always replicated [36]. What we do claim is that the scaffolding the tutoring modes describe, support that moves with the student, is not what current evaluations measure, and that in our probes the two led to different conclusions about the same tutor.

The instrument also lets one ask who bears the effect of an instruction. Under the policy of answer withholding, which the tutoring modes state, it is the student who keeps failing who receives less. The answer is withheld and the support below it is lower as well (§5.2, §5.3), the opposite of what the norm asks for under persistent failure. Since our probes fix that student by design, how often real students are in that position is a separate question and can be examined in future work.

Level and direction mostly move together, so one might think that a per-turn mean is enough. For example, a tutor that gives more support on average also raises it more often after a failure (Appendix Table 9). However, the mean cannot show where the two differ: the four outcomes of a failure (§5.2), the timing of the answer under persistent failure (§5.3) and the same student reply answered differently on the first and the fifth turn (§5.5).

However, our work has several limitations. First, the assigned script of the student state may not correspond to the state the tutor perceives. Appendix C.1 recomputes every quantity without the misconception turns a tutor may take as correct, and Appendix B.2 gives the count for the success case, which bears on lowering after a success. Second, each turn was scored once by the LLM judge. We validate it by its agreement with the first author's labels on 120 turns in Appendix B.4. The scale does not register a tutor's content errors, help with the arithmetic itself, or demands placed on the student. Third, each probe uses one fixed order of failure types, the same for every tutor and not counterbalanced, so the position of a failure and its type are separable only where the same type recurs. Lastly, the simulated student attempts one step per turn, and its states are task-performance states, so disengagement [27, 29] is out of scope.

7 Conclusion

The tutoring modes of general-purpose assistants promise scaffolding. However, the evaluations used to assess them score single turns, which cannot register whether support moves with the student. Therefore, we built an instrument that can. Applied to three LLM tutors under five prompts, it shows what per-turn scores hide: the same student reply is answered with a hint on the first turn but with the answer on the fifth, the usual compliance rate cannot tell raising support from giving the answer, an answer-withholding prompt lowers support even on the turns that do not give the answer, and lowering support after a student's success is present but weak. The gap between the scaffolding claim and what current evaluations measure is an accountability problem, and measuring direction with the student's state known is one way to close it.

Generative AI Usage Statement

Generative AI models are used as a part of the method. The three LLMs (`gpt-5.4`, `gemini-3.8-flash`, `claude-sonnet-5`) are the tutors we evaluate. For the simulated student, we use `qwen3.8-flash`. For the judge scoring the support level of each tutor turn, `deepseek-r1-0528` is used. For the item construction, `gpt-5.5-2026-04-23` drafted the 40 base items, all of which the first author reviewed and accepted (Appendix B.1). Claude Fable 5.1, through Claude Code, was used to polish the text of this paper and assisted with the formatting of the tables and figures.

Ethical Considerations Statement

This study does not involve human participants. The students are simulated, and every student state is scripted. The tutors are proprietary models that are accessed through their public APIs. The only human input is labelling of tutor turns by the authors. We mention deployed tutoring products only to show that an answer-withholding policy is in use. We implement a strict version of the published policy as a one-sentence prompt, and we do not measure the products themselves. Therefore, nothing here should be interpreted as an evaluation of them. We measure adherence to a declared norm of contingency, not learning. Our results should not be used to rank tutors as better or worse teachers, nor should they be used to argue that an answer-withholding instruction harms students. The problems and dialogues contain no personal data, and the names in the items are fictional. We will release our dataset, which includes items, probes, judge prompts, and our simulation and analysis code, upon acceptance.

References

[1] Vincent Aleven and Kenneth R. Koedinger. 2001. Investigations into help seeking and learning with a cognitive tutor. In *Papers of the AIED-2001 Workshop on Help Provision and Help Seeking in Interactive Learning Environments.*

[2] Alibaba Cloud. 2026. Supported Models and Capabilities Overview: Alibaba Cloud Model Studio. https://www.alibabacloud.com/help/en/model-studio/models, accessed 2026-11-02.

[3] 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 (1995), 167–207. doi:10.1207/s15327809jls0402_2

[4] Anthropic. 2025. Introducing Claude for Education. https://www.anthropic.com/news/introducing-claude-for-education.

[5] Anthropic. 2026. Introducing Claude Sonnet 5. https://www.anthropic.com/news/claude-sonnet-5.

[6] Andrew M. Bean, Ryan Othniel Kearns, Angelika Romanou, Franziska Sofia Hafner, Harry Mayne, Jan Batzner, Negar Foroutan, Chris Schmitz, Karolina Korgul, Hunar Batra, Oishi Deb, Emma Beharry, Cornelius Emde, Thomas Foster, Anna Gausen, María Grandury, Simeng Han, Valentin Hofmann, Lujain Ibrahim, Hazel Kim, Hannah Rose Kirk, Fangru Lin, Gabrielle Kaili-May Liu, Lennart Luettgau, Jabez Magomere, Jonathan Rystrøm, Anna Sotnikova, Yushi Yang, Yilun Zhao, Adel Bibi, Antoine Bosselut, Ronald Clark, Arman Cohan, Jakob Foerster, Yarin Gal, Scott A. Hale, Inioluwa Deborah Raji, Christopher Summerfield, Philip H. S. Torr, Cozmin Ududec, Luc Rocher, and Adam Mahdi. 2025. Measuring what Matters: Construct Validity in Large Language Model Benchmarks. arXiv:2511.04703 [cs.CL] https://arxiv.org/abs/2511.04703

[7] Courtney Bell, Geoffrey Phelps, Dan McCaffrey, Shuangshuang Liu, Barbara Weren, Nancy Glazer, and Francesca Forzani. 2022. *Simulating classroom interactions at scale for the improvement of practice-based teacher education.* Technical Report 2022-3. Wisconsin Center for Education Research.

[8] Conrad Borchers and Tianze Shou. 2025. Can Large Language Models Match Tutoring System Adaptivity? A Benchmarking Study. arXiv:2504.05570 [cs.CL] https://arxiv.org/abs/2504.05570

[9] M. A. (Ken) Clements. 1980. Analyzing children's errors on written mathematical tasks. *Educational Studies in Mathematics* 11, 1 (1980), 1–21. doi:10.1007/BF00369157

[10] Clayton Cohn, Siyuan Guo, Surya Rayala, Hanchen David Wang, Naveeduddin Mohammed, Umesh Timalsina, Shruti Jain, Angela Eeds, Menton Deweese, Pamela J. Osborn Popp, Rebekah Stanton, Shakeera Walker, Meiyi Ma, and Gautam Biswas. 2026. Evidence-Decision-Feedback: Theory-Driven Adaptive Scaffolding for LLM Agents. arXiv:2602.01415 [cs.MA] doi:10.1007/978-3-032-29744-0_1

[11] Clayton Cohn, Surya Rayala, Siyuan Guo, Hanchen David Wang, Naveeduddin Mohammed, Umesh Timalsina, Shruti Jain, Ryan Li, Angela Eeds, Menton Deweese, Pamela J. Osborn Popp, Rebekah Stanton, Shakeera Walker, Ashwin T S, Meiyi Ma, and Gautam Biswas. 2026. A Theory-Guided LLM Pedagogical Agent for STEM+C Scaffolding Without Over-Reliance. arXiv:2605.30539v1 [cs.MA] https://arxiv.org/abs/2605.30539

[12] Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. Stepwise Verification and Remediation of Student Reasoning Errors with Large Language Model Tutors. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 8386–8411. doi:10.

18653/v1/2024.emnlp-main.478

[13] Heejin Do, Shashank Sonkar, and Mrinmaya Sachan. 2026. Simulating Students or Sycophantic Problem Solving? On Misconception Faithfulness of LLM Simulators. arXiv:2605.12748 [cs.CL] https://arxiv.org/abs/2605.12748

[14] Benjamin H. Dotger. 2015. Core pedagogy: Individual uncertainty, shared practice, formative ethos. *Journal of Teacher Education* 66, 3 (2015), 215–226. doi:10.1177/0022487115570093

[15] Shuyi Fan, Boyuan Deng, Mengyu Xu, Jiale Liu, Hongyang Zhang, Qiaoxin Yang, and Chongyang Gao. 2026. Rethinking LLM-Judged Helpfulness as a Pedagogy Signal: A Pre-Registered Audit Across Tutor Models. arXiv:2607.28128 [cs.CL] https://arxiv.org/abs/2607.28128

[16] Google. 2025. Guided Learning in Gemini: From answers to understanding. https://blog.google/products-and-platforms/products/education/guided-learning/.

[17] Google DeepMind. 2026. Gemini 3.8 Flash: Model card. https://deepmind.google/models/model-cards/gemini-3-8-flash/.

[18] Daya Guo, Dejian Yang, Haowei Zhang, et al. 2025. DeepSeek-R1 Incentivizes Reasoning in LLMs through Reinforcement Learning. *Nature* 645, 8081 (2025), 633–638. doi:10.1038/s41586-025-09422-z

[19] Abigail Z. Jacobs and Hanna Wallach. 2021. Measurement and Fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*. ACM, New York, NY, USA, 375–385. arXiv:1912.05511 [cs.CY] doi:10.1145/3442188.3445901

[20] Suhyeon Lee, Juneha Baek, Jaehyeong Park, and Donghyuk Shin. 2026. LLM Pedagogical Behavior in AI Tutoring Interactions. arXiv:2608.22993 [cs.CL] https://arxiv.org/abs/2608.22993

[21] Ning Li, Zheng Zhang, Zhenya Huang, Rui Li, Yi Zhan, Yinbo Luo, Qi Liu, and Enhong Chen. 2026. LongTutor: Benchmarking Large Language Models for Long-term Personalized Tutoring. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (Eds.). Association for Computational Linguistics, San Diego, California, United States, 29712–29737. doi:10.18653/v1/2026.acl-long.1371

[22] Naiming Liu, Richard Baraniuk, and Shashank Sonkar. 2026. MetaCLASS: Metacognitive Coaching for Learning with Adaptive Self-regulation Support. arXiv:2602.02457 [cs.CY] https://arxiv.org/abs/2602.02457

[23] Ying Liu, Can Li, Ting Zhang, Mei Wang, Qiannan Zhu, Jian Li, and Hua Huang. 2025. Discerning minds or generic tutors? Evaluating instructional guidance capabilities in Socratic LLMs. arXiv:2508.06583 [cs.CL] https://arxiv.org/abs/2508.06583

[24] Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. MathDial: A Dialogue Tutoring Dataset with Rich Pedagogical Properties Grounded in Math Reasoning Problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 5602–5621. doi:10.18653/v1/2023.findings-emnlp.372

[25] Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2025. MathTutorBench: A Benchmark for Measuring Open-ended Pedagogical Capabilities of LLM Tutors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 204–221. doi:10.18653/v1/2025.emnlp-main.11

[26] Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. Unifying AI Tutor Evaluation: An Evaluation Taxonomy for Pedagogical Ability Assessment of LLM-Powered AI Tutors. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Luis Chiruzzo, Alan Ritter, and Lu Wang (Eds.). Association for Computational Linguistics, Albuquerque, New Mexico, 1234–1251. doi:10.18653/v1/2025.naacl-long.57

[27] Xianghui Meng and Jionghao Lin. 2026. Simulating Disengaged Students to Evaluate LLM-based Tutors. arXiv:2609.12331 [cs.LG] https://arxiv.org/abs/2609.12331

[28] National Governors Association Center for Best Practices and Council of Chief State School Officers. 2010. Common Core State Standards for Mathematics. Standard 4.OA.A.3: solve multistep word problems posed with whole numbers using the four operations.

[29] Alexandra Neagu, Jeffrey T. H. Wong, Marcus Messer, Rhodri Nelson, and Peter B. Johnson. 2026. Rethinking Scaffolding in LLM Tutors: The Interactional Mismatch Between Benchmarks and Real-World Deployments. arXiv:2606.15766 [cs.AI] https://arxiv.org/abs/2606.15766

[30] M. Anne Newman. 1977. An analysis of sixth-grade pupils' errors on written mathematical tasks. In *Research in Mathematics Education in Australia, 1977*, M. A. Clements and J. Foyster (Eds.). Vol. 1. Melbourne, 239–258.

[31] OpenAI. 2025. Introducing study mode. https://openai.com/index/chatgpt-study-mode/.

[32] OpenAI. 2026. Introducing GPT-5.4. https://openai.com/index/introducing-gpt-5-4/.

[33] OpenAI. 2026. Using Study Mode in ChatGPT. Help Center article 11780217, https://help.openai.com/en/articles/11780217-chatgpt-study-mode-faq, accessed 2026-10-18.

[34] Donya Rooein, Sankalan Pal Chowdhury, Mariia Eremeeva, Yuan Qin, Debora Nozza, Mrinmaya Sachan, and Dirk Hovy. 2026. PATS: Personality-Aware Teaching Strategies with Large Language Model Tutors. In *Findings of the Association for Computational Linguistics: EACL 2026*. Association for Computational Linguistics, Rabat, Morocco, 4186–4211. arXiv:2601.08402 [cs.CL] https://arxiv.org/abs/2601.08402

[35] Jing Shao, Qifeng Wu, Hanyu Zhang, Sixia Sun, and Jun Zhuang. 2026. Mitigating Scaffolding Collapse in Socratic Tutors via Representation Alignment. arXiv:2607.19371 [cs.AI] https://arxiv.org/abs/2607.19371

[36] Nienke Smit, Renske de Kleijn, Jelte M. Wicherts, and Janneke van de Pol. 2025. What it takes to tutor—A preregistered direct replication of the scaffolding experimental study by D. Wood et al. (1978). *Journal of Educational Psychology* 117 (2025), 1313–1329. doi:10.1037/edu0000965

[37] Rakshith S Srinivasa, Zora Che, Chen Bo Calvin Zhang, Diego Mares, Ernesto Hernandez, Jayeon Park, Dean Lee, Guillermo Mangialardi, Charmaine Ng, Ed-Yeremai Hernandez Cardona, Anisha Gunjal, Yunzhong He, Bing Liu, and Chen Xing. 2025. TutorBench: A Benchmark To Assess Tutoring Capabilities Of Large Language Models. arXiv:2510.02663 [cs.LG] https://arxiv.org/abs/2510.02663

[38] Verona Teo, Raghav Jain, Tobias Gerstenberg, and Max Kleiman-Weiner. 2026. AI Assistants Overassist. arXiv:2607.21306 [cs.LG] https://arxiv.org/abs/2607.21306

[39] Janneke van de Pol, Monique Volman, Ed Elbers, and Jos Beishuizen. 2012. Measuring scaffolding in teacher–small-group interactions. In *Pedagogy: New Developments in the Learning Sciences*, Robyn M. Gillies (Ed.). Nova Science, Hauppauge, NY, 151–188.

[40] Janneke van de Pol, Monique Volman, Frans Oort, and Jos Beishuizen. 2015. The effects of scaffolding in the classroom: support contingency and student independent working time in relation to student achievement, task effort and appreciation of support. *Instructional Science* 43, 5 (2015), 615–641. doi:10.1007/s11251-015-9351-z

[41] Kurt VanLehn. 2006. The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education* 16, 3 (2006), 227–265.

[42] Hanchen David Wang, Clayton Cohn, Zifan Xu, Siyuan Guo, Gautam Biswas, and Meiyi Ma. 2026. BEAGLE: Behavior-Enforced Agent for Grounded Learner Emulation. arXiv:2602.13280 [cs.AI] https://arxiv.org/abs/2602.13280

[43] Daniel Weitekamp, Momin N. Siddiqui, and Christopher J. MacLellan. 2025. TutorGym: A Testbed for Evaluating AI Agents as Tutors and Students. In *Artificial Intelligence in Education (AIED 2025), Lecture Notes in Computer Science*. Springer, Cham, 361–376. arXiv:2505.01563 [cs.AI] doi:10.1007/978-3-031-98420-4_26

[44] David Wood, Heather Wood, and David Middleton. 1978. An experimental evaluation of four face-to-face teaching strategies. *International Journal of Behavioral Development* 1, 2 (1978), 131–147. doi:10.1177/016502547800100203

[45] Heather Wood and David Wood. 1999. Help seeking, learning and contingent tutoring. *Computers & Education* 33, 2–3 (1999), 153–169. doi:10.1016/S0360-1315(99)00030-5

[46] Tengchao Yang, Sichen Guo, Mengzhao Jia, Jiaming Su, Yuanyang Liu, Zhihan Zhang, and Meng Jiang. 2026. MMTutorBench: The First Multimodal Benchmark for AI Math Tutoring. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (Eds.). Association for Computational Linguistics, San Diego, California, United States, 23310–23332. doi:10.18653/v1/2026.acl-long.1068

[47] Albert Zhang, Alexis Ross, Kajal Patel, Julian Bernado, Rebecca Bowie, Ana Trindade Ribeiro, Daniel Halper, Haripriya Valayaputtur, Jacob Andreas, Susanna Loeb, Li Lucy, Kyle Lo, and Ryan Knight. 2026. *When Help is Unhelpful: Evaluating AI Tutors for Productive Struggle*. Preprint (TutorMoments-Preview). Allen Institute for AI. https://tutormoments.allen.ai/static/paper/tutormoments-preview.pdf

[48] Bingxi Zhao, Jiahao Zhang, Xubin Ren, Zirui Guo, Tianzhe Chu, Yi Ma, and Chao Huang. 2026. DeepTutor: Towards Agentic Personalized Tutoring. arXiv:2604.26962 [cs.CY] https://arxiv.org/abs/2604.26962

[49] Jin Zhao, Marta Knežević, and Tanja Käser. 2026. Evaluating Answer Leakage Robustness of LLM Tutors against Adversarial Student Attacks. arXiv:2604.18660 [cs.CR] https://arxiv.org/abs/2604.18660

A Survey of Tutor Benchmark Metrics

Table 4 lists the twenty benchmarks and audits of LLM tutors whose evaluation sections we read, grouped by the unit to which the metric is attached. Table 5 places this work beside the work of §2.2.

B Measurement Design in Detail

B.1 Items

Skeletons and variants. A skeleton fixes the shape of the three expressions. For example, K07 in Table 1 is $a \div b + c \longrightarrow P \cdot d + e \longrightarrow P - f \cdot g$, where P is the previous value. Final operations vary across skeletons at every position, so no single operation shape stands in for a support level. Variant 1 of each item uses the author's numbers. Variants 2–8 draw each parameter from half to one and a half times its value under the item's constraints, choosing combinations that keep all intermediate values positive integers, keep the eight final answers distinct, and lie far from each other.

Error categories. F-misc are authored as expressions over the item's parameters, so they carry over to every variant. Their categories sit under two of the hurdles that precede calculation in Newman's hierarchy [9, 30]: comprehension (*misunderstanding of a question*) and transformation (*extra or missing quantity*, *unit conversion error*), the category names used in the teacher annotations of Daheim et al. [12]. The categories exist to let a student who fails one step

Table 4. What twenty tutor evaluations score. *Unit*: what a single metric value is attached to; a response is one reply to a fixed context, a turn one of the tutor's turns in a continuing dialogue.

Work	Metric	Unit
One response		
MRBench [26]	share of responses with the desired label on 8 dimensions (including revealing the answer)	response
MathTutorBench [25]	accuracy, F1; reward-model win rate against teacher responses	response
TutorBench [37]	weighted pass rate on sample-specific rubric criteria, LLM judge	response
MMTutorBench [46]	per-problem rubric, LLM judge	response
DeepTutor [48]	five 1–5 ratings on every tutor turn and five on the practice questions it generates, no composite	turn
LongTutor [21]	LLM-judge rubric on one teaching response given a learning history, plus per-item evidence and diagnosis tasks	response
GuideEval [23]	behaviour scores conditioned on a contrastive student state	response under a labelled state
Borchers & Shou [8]	pass rate of quality checks; shift when context is removed	response
Lee et al. [20]	support level L0–L4 per response, used to predict the next student act	response
MetaCLASS [22]	accuracy of the predicted coaching move	turn (move label)
EDF-CoPA [10]	share of each support policy by student mastery quintile, rank correlation; recall and similarity of extracted evidence	turn (policy label)
CoPA [11]	as EDF-CoPA (same classroom study), plus student confidence and interaction trajectories	turn (policy label)
TutorMoments [47]	whether the tutor's replayed turns fit a teacher-marked learning moment, LLM judge	moment and its 5-turn replay
Helpfulness audit [15]	helpfulness and pedagogy rubrics by an LLM judge; answer-leakage and next-turn-independence detectors	turn
Dialogue		
MathDial [24]	student success rate against how often the teacher tells the answer	dialogue
PATS [34]	pairwise win rate for adapting to the student's personality	dialogue
Rethinking scaffolding [29]	tutor scaffolding × student uptake, aggregated turn scores	dialogue
Time to event		
INT-BENCH [38]	intervention frequency, relative timing, change in correctness after one intervention	episode, one intervention
Answer leakage [49]	tutor leakage rate; turns until the tutor leaks the answer	dialogue
Scaffolding collapse [35]	collapse rate; turn at which collapse begins	dialogue

repeatedly make a different kind of mistake each time. We generate the value for F-slip: the correct expression with a value off by 1 (by up to 3 where needed), never equal to a misconception answer.

Authoring and review. Items were drafted by `gpt-5.5-2026-04-23` from a prompt that fixes the register (everyday settings, whole numbers, three to five sentences, one question at the end). The first author reviewed all 40 items for register, the wording of each locus, and the plausibility of each misconception answer and its stated reason.

B.2 Generation protocol

Turns, variants and personas. The problem text is presented at the head of the student's first turn, and the tutor responds from that turn onwards, speaking six times. Once the student has completed the final step of an item, the

Table 5. Positioning against the work of §2.2. *State*: the student's state is fixed by the evaluator rather than inferred. *Ordinal*: the amount of tutor support is placed on an ordered scale. *Direction*: the change in support between tutor turns is scored against what the student's state requires. ∼: MetaCLASS fixes a learner profile and a planned sequence of events (*e.g.*, stuck, hedging) but not the learner's correctness.

Work	Unit scored	State	Ordinal	Direction
DAS^2 [27]	transcript	✓	✗	✗
INT-BENCH [38]	episode, one intervention	✓	✗	✗
TutorGym [43]	next step	✓	✗	✗
GuideEval [23]	response under a labelled state	✓	✗	✗
Lee et al. [20]	response → next student act	✗	✓	✗
BEAGLE [42]	trajectory	✓	✓	✗
MetaCLASS [22]	turn (move label)	∼	✗	✗
Ours	transition, trajectory	✓	✓	✓

next variant arrives at the beginning of the student's turn. A six-turn dialogue never exhausts the eight variants. The tutor's input is its system prompt followed by strictly alternating student and tutor messages. The system prompt is the opening sentence or sentences of the condition (Table 3b) followed by the same turn rules in every condition. The student is about eleven years old, the tutor speaks one turn at a time, and a turn has at most three sentences. The student speaks in one of two personas: terse, or thinking aloud about how they approached the step. A persona changes only the wording, not the state.

What the student simulator sees. The student simulator does not see the whole dialogue. It receives the problem, the target of the current step, the tutor's last turn, its own earlier attempts on this step, sentences that state the scripted state as fact, and a one-line persona. Withholding the rest of the tutor's turns keeps what the tutor said earlier from affecting the student's answer. We also show the student simulator its own earlier attempts, keeping repeated failures on one step consistent with each other. On the turn that opens a problem, there is no tutor turn to show. Of the 43,200 student turns, 583 (1.3%) failed the fidelity check of §4.3 at least once and were regenerated; a turn that fails it three times is replaced by a scripted sentence for its state, which happened to 97 turns (0.22%). The rule that checks the student turn against its script was compared with human labels before the run ($\kappa = 0.90$ on 40 turns).

Run settings. Each tutor is called through its API at the default sampling temperature with reasoning effort set to medium. The judge runs at temperature 0 and answers in one line of JSON; its API does not allow reasoning to be switched off, so the output limit is 16k tokens and the reasoning is not read. Every condition runs on P1–P6: 30 probe × condition blocks per tutor, 80 dialogues each, 7,200 in all, each cell generated once.

Where the student script and the tutor part. As the first limitation of §6 says, the script fixes the state the student presents, not the state the tutor takes up. Specifically, in about 7% of success turns, the tutor had asked for one operation, and the student gave the value of the whole step, which the tutor took as wrong; this happens more with the terse persona (about 12% against 3%) and with tutors that split steps into small pieces.

B.3 Derivation of the support scale

Table 6 places s beside the two control scales it draws on. Three adaptations separate s from its sources. First, Wood's levels 3 and 4 have no counterpart in a mathematical dialogue: choosing or arranging blocks is physical help [44]. We refill that range with the amount of task content given: where to look, which operation, which expression. Second, we

Table 6. Correspondence between s and the scales it adapts.

s	Wood et al. [44]	van de Pol et al. [39], degree of control
0 none	1 general verbal encouragement	1–3 no new content (split by the response elicited)
1 locus	2 specific verbal information (what to look at)	4 new content with a hint or suggestive question
2 operation	2	4
3 expression	3–4 selecting or preparing materials (physical only)	4
4 value	5 demonstration	5 explanation or the answer

do not adopt the low-end distinctions of the classroom scale of van de Pol et al. [39], which grades levels 1–3 based on whether a response is open, explanatory, or short. The answer-withholding instruction (C1) changes exactly that form, since a tutor told not to give the answer asks instead of tells, so scoring by form would turn the effect of that condition into a coding artefact. Levels 1–3 therefore merge into $s = 0$. Third, we refine the high end: the range that van de Pol's scale treats as one level (level 4, new content with a hint) becomes three levels, because how much of the task is given is the quantity our research question concerns.

B.4 Judge protocol and agreement with the author's labels

We give the remaining details of the judge's scoring here, beyond the definition of §4.2. A question that states the value counts as a reveal; one that leaves the value unsaid stays at level 3. A turn that names the operation and also gives the expression is level 3: the highest field counts. A turn that repeats a value or an expression the student has just given adds nothing new, so it is scored by whatever else it does. A turn that computes only part of a composite step ("$4 \times 13 = 52$" in a step worth $4 \times 13 + 8$) gives more than the expression but not the answer, and is level 3.

The first author labelled 120 tutor turns from four of the 40 items, 40 per tutor over all probes and conditions, stratified by the judge's level and blind to tutor, condition and probe. The judge agrees with these labels on 92% of turns exactly and on 98% within one level (linearly weighted $\kappa = 0.93$); κ is 0.94, 0.94 and 0.92 for the three tutors, 0.91 after a failure and 0.96 after a success. Table 7 gives the judge's levels against the author's.

Table 7. Author's level (rows) against the judge's level (columns) on the 120 labelled turns.

	0	1	2	3	4
0	21	0	0	0	0
1	1	24	3	0	0
2	1	0	20	3	0
3	1	0	1	21	0
4	0	0	0	0	24

C Further Results

C.1 Scripted misconceptions that are correct intermediate values

Of the 6,480 scripted misconception turns, 1,710 use a value that is a correct intermediate result of the step, which a tutor may take as a success. Table 8 gives the reveal and raise shares with and without these turns. The conclusions of §5 hold without them.

Table 8. Reveal and raise shares per scored failure transition, as reported (main) and without the scripted misconception turns whose value is a correct intermediate result of the step.

		Reveal		Raise	
Tutor	Cond.	main	without	main	without
Claude	C0	0.11	0.12	0.43	0.44
Claude	C1	0.02	0.03	0.49	0.48
Claude	C2	0.05	0.05	0.45	0.46
Gemini	C0	0.06	0.06	0.49	0.49
Gemini	C1	0.01	0.01	0.53	0.53
Gemini	C2	0.11	0.12	0.51	0.51
GPT	C0	0.84	0.92	0.14	0.06
GPT	C1	0.05	0.06	0.48	0.46
GPT	C2	0.50	0.58	0.36	0.30

C.2 The response to a failure (§5.2)

Table 9 gives the four outcomes of Figure 3 with their 95% intervals, Table 10 the outcomes by failure type, and Table 11 the first failure of a problem against the later ones.

Table 9. The numbers behind Figure 3: the four outcomes per scored failure transition on P1–P4 with 95% intervals, the number scored, the censoring rate, and the mean level over every turn (reveal turns and those after them included) and over the turns before the first reveal that do not give the answer.

								mean level	
Tutor	Cond.	*n*	reveal	raise	hold	lower	censored	every turn	no answer
Claude	C0	732	0.11 [0.09, 0.13]	0.43 [0.41, 0.46]	0.34 [0.30, 0.38]	0.12 [0.10, 0.14]	0.18	1.77	1.61
	C1	851	0.02 [0.02, 0.04]	0.49 [0.46, 0.52]	0.30 [0.26, 0.34]	0.19 [0.17, 0.20]	0.02	1.26	1.22
	C2	820	0.05 [0.03, 0.06]	0.45 [0.43, 0.48]	0.36 [0.33, 0.40]	0.14 [0.12, 0.16]	0.05	1.52	1.45
	C3	813	0.06 [0.04, 0.07]	0.45 [0.43, 0.48]	0.36 [0.32, 0.39]	0.14 [0.12, 0.15]	0.06	1.55	1.45
	C4	817	0.06 [0.05, 0.08]	0.45 [0.43, 0.47]	0.36 [0.33, 0.40]	0.12 [0.10, 0.14]	0.05	1.59	1.49
Gemini	C0	811	0.06 [0.04, 0.07]	0.49 [0.46, 0.52]	0.34 [0.30, 0.37]	0.12 [0.10, 0.14]	0.05	1.64	1.55
	C1	863	0.01 [0.00, 0.02]	0.53 [0.51, 0.55]	0.29 [0.27, 0.32]	0.17 [0.15, 0.19]	0.01	1.27	1.25
	C2	764	0.11 [0.10, 0.12]	0.51 [0.49, 0.53]	0.31 [0.27, 0.34]	0.07 [0.05, 0.09]	0.07	1.75	1.57
	C3	824	0.04 [0.02, 0.05]	0.55 [0.53, 0.57]	0.27 [0.25, 0.30]	0.14 [0.12, 0.16]	0.04	1.16	1.09
	C4	775	0.10 [0.08, 0.11]	0.53 [0.51, 0.56]	0.29 [0.26, 0.32]	0.08 [0.06, 0.09]	0.07	1.34	1.14
GPT	C0	256	0.84 [0.80, 0.89]	0.14 [0.11, 0.18]	0.01 [0.00, 0.03]	0.00 [0.00, 0.01]	0.70	2.77	1.84
	C1	834	0.05 [0.03, 0.06]	0.48 [0.46, 0.50]	0.33 [0.30, 0.36]	0.15 [0.13, 0.17]	0.03	1.76	1.70
	C2	363	0.50 [0.45, 0.56]	0.36 [0.32, 0.39]	0.12 [0.09, 0.16]	0.02 [0.00, 0.03]	0.43	2.58	1.97
	C3	496	0.33 [0.29, 0.37]	0.41 [0.38, 0.44]	0.23 [0.20, 0.26]	0.03 [0.01, 0.05]	0.29	2.30	1.85
	C4	361	0.51 [0.46, 0.56]	0.36 [0.33, 0.39]	0.11 [0.08, 0.15]	0.02 [0.00, 0.04]	0.45	2.51	1.89

C.3 Holding out under repeated failure (§5.3)

Table 12 gives the shares of Figure 4 by turn. Table 13 gives the four outcomes at P2's fourth turn, over its turns 4–6, and at P5's first failure. The two cells that differ are Claude under the answer-withholding prompt, which holds more often on P5, and GPT under the raise-on-failure instruction, which raises more often.

Table 10. The four outcomes per scored failure transition, by the scripted type of the failure.

Tutor	Cond.	Failure	*n*	reveal	raise	hold	lower
Claude	C0	misconception	470	0.09	0.46	0.33	0.12
		slip	70	0.23	0.10	0.31	0.36
		no attempt	192	0.11	0.49	0.37	0.02
Claude	C1	misconception	544	0.01	0.52	0.30	0.17
		slip	78	0.10	0.22	0.23	0.45
		no attempt	229	0.02	0.52	0.32	0.14
Claude	C2	misconception	525	0.02	0.48	0.36	0.14
		slip	77	0.19	0.19	0.30	0.31
		no attempt	218	0.06	0.49	0.38	0.07
Gemini	C0	misconception	517	0.03	0.54	0.33	0.09
		slip	75	0.17	0.13	0.37	0.32
		no attempt	219	0.07	0.50	0.33	0.10
Gemini	C1	misconception	551	0.01	0.53	0.32	0.14
		slip	77	0.01	0.23	0.25	0.51
		no attempt	235	0.01	0.62	0.25	0.12
Gemini	C2	misconception	483	0.04	0.60	0.29	0.06
		slip	71	0.25	0.07	0.48	0.20
		no attempt	210	0.21	0.45	0.29	0.05
GPT	C0	misconception	176	0.77	0.21	0.02	0.01
		slip	0	–	–	–	–
		no attempt	80	1.00	0.00	0.00	0.00
GPT	C1	misconception	536	0.01	0.50	0.33	0.16
		slip	80	0.23	0.25	0.23	0.30
		no attempt	218	0.06	0.51	0.36	0.07
GPT	C2	misconception	272	0.45	0.40	0.13	0.02
		slip	6	0.50	0.00	0.50	0.00
		no attempt	85	0.66	0.26	0.07	0.01

Table 11. Reveal share on the first turn and on turns 2–6 of the probes that open with a failure (P2–P4), with 95% intervals.

		Reveal	
Tutor	Cond.	turn 1	turns 2–6
Claude	C0	0.16 [0.12, 0.20]	0.08 [0.06, 0.10]
Claude	C1	0.00 [0.00, 0.01]	0.03 [0.02, 0.05]
Claude	C2	0.02 [0.00, 0.04]	0.06 [0.04, 0.08]
Gemini	C0	0.02 [0.00, 0.03]	0.07 [0.05, 0.09]
Gemini	C1	0.01 [0.00, 0.02]	0.01 [0.01, 0.02]
Gemini	C2	0.01 [0.00, 0.03]	0.15 [0.14, 0.17]
GPT	C0	0.85 [0.80, 0.89]	0.76 [0.63, 0.93]
GPT	C1	0.00 [0.00, 0.00]	0.06 [0.05, 0.08]
GPT	C2	0.47 [0.40, 0.54]	0.56 [0.48, 0.65]

C.4 Lowering after a success (§5.4)

Table 14 gives lowering after a success against after a failure for every cell.

Table 12. The numbers behind Figure 4: share of P3 dialogues in which the tutor has given the answer by turn k, 80 dialogues per cell.

		answered by turn k					
Tutor	Cond.	1	2	3	4	5	6
Claude	C0	0.06	0.10	0.13	0.33	0.44	0.44
	C1	0.00	0.01	0.03	0.13	0.16	0.23
	C2	0.00	0.03	0.04	0.23	0.34	0.36
	C3	0.00	0.04	0.05	0.19	0.30	0.36
	C4	0.00	0.01	0.01	0.16	0.39	0.44
Gemini	C0	0.00	0.06	0.06	0.23	0.40	0.50
	C1	0.00	0.03	0.04	0.05	0.06	0.09
	C2	0.00	0.05	0.11	0.34	0.84	0.95
	C3	0.00	0.06	0.06	0.19	0.26	0.31
	C4	0.00	0.03	0.08	0.29	0.74	0.85
GPT	C0	1.00	1.00	1.00	1.00	1.00	1.00
	C1	0.00	0.00	0.00	0.23	0.29	0.33
	C2	0.68	0.85	0.93	0.96	0.99	0.99
	C3	0.40	0.51	0.64	0.89	0.96	0.98
	C4	0.70	0.85	0.93	1.00	1.00	1.00

Table 13. The four outcomes per scored transition at P2's fourth turn, over P2's turns 4–6, and at the first failure of the control P5 (turn 3); n is the number scored.

		P2, turn 4					P2, turns 4–6					P5, turn 3				
Tutor	Cond.	n	reveal	raise	hold	lower	n	reveal	raise	hold	lower	n	reveal	raise	hold	lower
Claude	C0	59	0.00	0.51	0.37	0.12	176	0.05	0.33	0.51	0.11	61	0.02	0.46	0.41	0.11
Claude	C1	79	0.00	0.44	0.37	0.19	237	0.01	0.37	0.42	0.20	80	0.00	0.25	0.60	0.15
Claude	C2	77	0.00	0.40	0.49	0.10	230	0.02	0.31	0.54	0.13	80	0.00	0.41	0.41	0.18
Claude	C3	73	0.00	0.48	0.44	0.08	219	0.02	0.32	0.53	0.13	78	0.00	0.49	0.36	0.15
Claude	C4	75	0.00	0.51	0.40	0.09	223	0.04	0.34	0.51	0.12	80	0.00	0.44	0.49	0.08
Gemini	C0	77	0.00	0.61	0.34	0.05	231	0.00	0.38	0.51	0.11	80	0.00	0.68	0.30	0.03
Gemini	C1	79	0.00	0.49	0.39	0.11	237	0.00	0.44	0.40	0.16	79	0.00	0.46	0.41	0.14
Gemini	C2	77	0.00	0.69	0.30	0.01	231	0.02	0.41	0.48	0.09	80	0.00	0.61	0.28	0.11
Gemini	C3	77	0.00	0.69	0.27	0.04	230	0.00	0.53	0.38	0.08	80	0.00	0.75	0.20	0.05
Gemini	C4	75	0.00	0.73	0.21	0.05	225	0.00	0.50	0.40	0.09	80	0.00	0.74	0.21	0.05
GPT	C0	13	0.69	0.15	0.08	0.08	17	0.76	0.12	0.06	0.06	16	0.63	0.19	0.13	0.06
GPT	C1	80	0.00	0.41	0.45	0.14	237	0.05	0.32	0.49	0.15	80	0.00	0.36	0.44	0.20
GPT	C2	50	0.52	0.12	0.34	0.02	79	0.57	0.10	0.28	0.05	64	0.44	0.36	0.20	0.00
GPT	C3	56	0.23	0.34	0.43	0.00	128	0.38	0.21	0.40	0.02	73	0.22	0.41	0.32	0.05
GPT	C4	42	0.40	0.38	0.17	0.05	79	0.46	0.24	0.25	0.05	65	0.43	0.43	0.11	0.03

C.5 Summaries and reference policies (§5.5)

Table 15 gives the compliance rate, the four outcomes after a failure and the difference of Eq. (2) for the reference policies; Table 16 gives the reveal rate at the first and the fifth "I do not know" of P6.

Table 14. Lowering after a success (S) and after a failure (F) on P1–P4, previous level 1–3: n, the share lowered, the share raised after a success, and the difference S − F of Eq. (2) with its 95% interval.

		after S			after F		
Tutor	Cond.	n	lowered	raised	n	lowered	S − F
Claude	C0	397	0.46	0.19	399	0.22	+0.24 [+0.16, +0.33]
	C1	474	0.45	0.16	452	0.35	+0.10 [+0.03, +0.16]
	C2	470	0.45	0.19	445	0.25	+0.20 [+0.13, +0.26]
	C3	453	0.45	0.19	450	0.25	+0.21 [+0.14, +0.27]
	C4	464	0.48	0.19	456	0.22	+0.26 [+0.20, +0.31]
Gemini	C0	518	0.32	0.25	493	0.19	+0.13 [+0.06, +0.19]
	C1	505	0.33	0.21	501	0.29	+0.04 [−0.03, +0.10]
	C2	520	0.33	0.28	476	0.12	+0.21 [+0.16, +0.26]
	C3	359	0.60	0.07	466	0.24	+0.36 [+0.30, +0.41]
	C4	364	0.56	0.10	455	0.13	+0.43 [+0.37, +0.49]
GPT	C0	125	0.25	0.26	15	0.07	+0.18 [−0.07, +0.33]
	C1	503	0.21	0.19	519	0.24	−0.03 [−0.08, +0.03]
	C2	365	0.27	0.24	108	0.06	+0.20 [+0.13, +0.28]
	C3	421	0.26	0.31	230	0.07	+0.19 [+0.12, +0.27]
	C4	336	0.29	0.25	111	0.06	+0.22 [+0.14, +0.31]

Table 15. The three reference policies of §4.5 on their exact levels. *Compliance*: the pooled rate of §5.5 (a success with support already at level 0 and the success on an item's last step left out). *After a failure*: the four outcomes per scored failure transition. Last column: the difference of Eq. (2), undefined for the teller. Last three rows: the tutors in the baseline, compliance with its 95% interval.

			After a failure				
Policy	Behaviour	Compliance	reveal	raise	hold	lower	Eq. (2)
Graded	follows the norm within levels 1–3, never answers	0.77	0.00	0.73	0.27	0.00	+1.00
Teller	always the answer	1.00	1.00	0.00	0.00	0.00	–
Shuffle	the oracle's levels in an order unrelated to the student	0.46	0.05	0.34	0.44	0.18	+0.01
Claude	baseline (C0)	0.51 [0.48, 0.54]	0.11	0.43	0.34	0.12	+0.24
Gemini	baseline (C0)	0.46 [0.44, 0.48]	0.06	0.49	0.34	0.12	+0.13
GPT	baseline (C0)	0.74 [0.68, 0.79]	0.84	0.14	0.01	0.00	+0.18

Table 16. P6: share of scored failures answered with the value at the first and at the fifth consecutive "I do not know" (turns 2 and 6), the number scored at the fifth, and the fifth minus the first with its 95% interval.

Tutor	Cond.	$j = 1$	$j = 5$	n_5	$j_5 - j_1$
Claude	C0	0.00	0.04	54	+0.04 [0.00, +0.10]
Claude	C1	0.00	0.05	76	+0.05 [0.00, +0.13]
Claude	C2	0.00	0.04	76	+0.04 [0.00, +0.09]
Claude	C3	0.01	0.05	65	+0.03 [−0.02, +0.10]
Claude	C4	0.01	0.03	72	+0.01 [−0.03, +0.07]
Gemini	C0	0.00	0.53	47	+0.53 [+0.37, +0.69]
Gemini	C1	0.00	0.00	77	0.00 [0.00, 0.00]
Gemini	C2	0.00	0.63	16	+0.63 [+0.31, +0.94]
Gemini	C3	0.00	0.39	61	+0.39 [+0.26, +0.52]
Gemini	C4	0.00	0.71	17	+0.71 [+0.50, +0.94]
GPT	C0	0.63	0.17	6	−0.46 [−0.83, +0.19]
GPT	C1	0.00	0.05	74	+0.05 [+0.01, +0.11]
GPT	C2	0.56	0.50	8	−0.06 [−0.43, +0.30]
GPT	C3	0.39	0.00	3	−0.39 [−0.50, −0.29]
GPT	C4	0.57	0.13	8	−0.44 [−0.68, −0.03]