Towards the Pedagogical Steering of Large Language Models for Tutoring: A Case Study with Modeling Productive Failure

Romain Puech[1,2], Jakub Macina[1,3], Julia Chatain[4], Mrinmaya Sachan[1], Manu Kapur[4]

[1]ETH Zurich, [2]École polytechnique, [3]ETH AI Center,
[4]Professorship for Learning Sciences and Higher Education, ETH Zurich,
romain.puech@alumni.polytechnique.org

Abstract

One-to-one tutoring is one of the most efficient methods of teaching. With growing popularity of Large Language Models (LLMs), there have been efforts to create LLM-based conversational tutors which can expand the benefits of one-to-one tutoring to everyone. However, current LLMs are trained primarily to be helpful assistants and lack crucial pedagogical skills. For example, they often quickly reveal the solution to the student and fail to plan for a richer multi-turn pedagogical interaction. To use LLMs in pedagogical settings, they need to be steered to use effective teaching strategies: a problem we introduce as *Pedagogical Steering*. We develop StratL, an algorithm to optimize LLM prompts and steer it to follow a predefined multi-turn tutoring plan represented as a transition graph. As a case study, we create a prototype tutor for high school math following *Productive Failure* (PF), an advanced and effective learning design. To validate our approach in a real-world setting, we run a field study with 17 high school students in Singapore and show that StratL succeeds in steering the LLM to follow the PF tutoring strategy. Finally, we highlight challenges in Pedagogical Steering of LLMs and offer opportunities for further improvements by publishing a dataset of PF problems[1] and our code[2].

1 Introduction

One-to-one tutoring is an effective teaching method (Bloom, 1984). Because of the efficiency of tutoring and its prohibitive cost, the challenge of finding scalable and affordable one-to-one tutoring solutions is considered as one of the greatest problems in education: the so-called *Two-Sigma Problem* (Bloom, 1984; Cohen et al., 1982). Following the rising popularity of Large Language

Models (LLMs), there has been a growing interest in using them to make tutoring affordable by building conversational tutoring systems (Jurenka et al., 2024; Khan, 2023). However, previous research has highlighted the lack of pedagogical properties of LLMs (Macina et al., 2023b; Tack and Piech, 2022). Most LLMs are instruction-tuned to optimize for a broad set of human preferences (Ouyang et al., 2022; Zhang et al., 2023), leading to assistant-like responses. However, a tutor's goal is to maximize learning, not user satisfaction.

These two goals can be contradictory. For example, if a student asks for help, providing them directly with the solution would likely maximize their satisfaction and be the response preferred by LLMs (Macina et al., 2023a). Yet, it is more efficient from a learning standpoint to promote students' active engagement and let them cognitively engage with the problem, for example, by providing an indirect hint or asking a guiding question (Freeman et al., 2014; Merrill et al., 1992). Moreover, existing LLM tutors are optimized and mostly evaluated for single-turn requests, while a tutoring dialog should be envisioned as a multi-turn teacher-student interaction where the teacher uses various pedagogical cues (Merrill et al., 1992). As LLMs are not natively made for tutoring, we introduce the shift from their user-serving goal to a pedagogically suitable one as the *Pedagogical Steering* problem.

We present and evaluate StratL, an algorithm to model a multi-turn *tutoring strategy* with LLMs. We define a multi-turn strategy as a succession of single-turn pedagogical goals called tutoring *intents*. StratL uses a transition graph to dynamically redefine the tutoring intents after every student utterance and prompts the LLM to follow these intents for the next turn. We partner with learning scientists to integrate a strategy based on an advanced learning design known as Productive Failure (PF) (Kapur, 2008, 2024), which is more
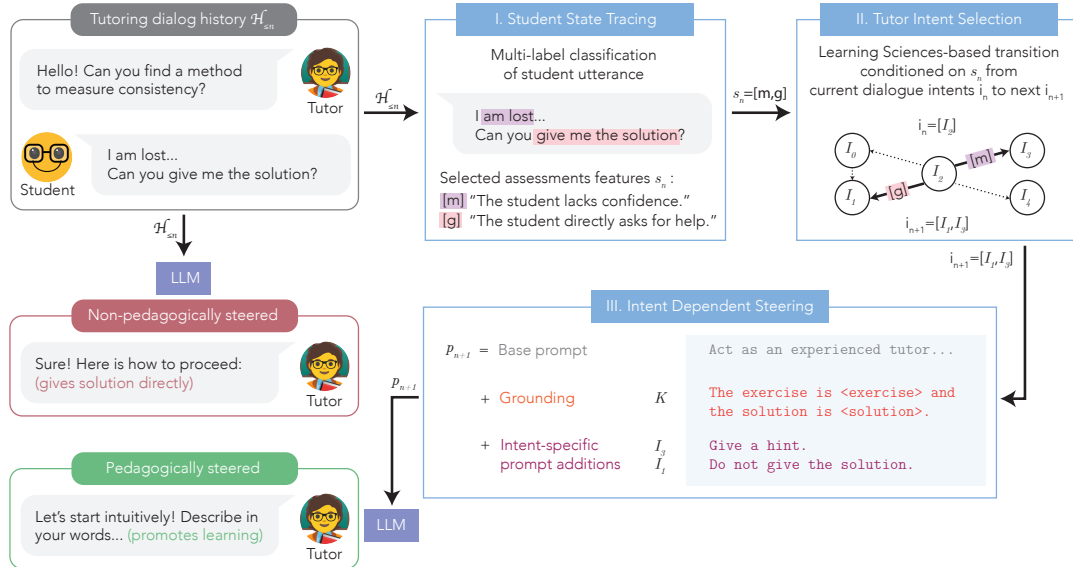
---

[1]Dataset
[2]Code

Figure 1: Schematic representation of StratL (in blue), an algorithm to control an LLM's tutoring strategy.

effective for human learning than other classical teaching methods (Sinha and Kapur, 2021). PF can be seen as the most extreme version of not directly telling the answer to the students. In PF, students are encouraged to explore an ill-structured problem, for example by using Socratic questioning (Shridhar et al., 2022), and are confronted with various suboptimal solutions, even if it leads to failure. While PF works across learning subjects, this work focuses on high school math, as it has been most explored in PF research (Sinha and Kapur, 2021).

We evaluate StratL and uncover practical limitations and insights on the Pedagogical Steering problem through a preliminary study with simulated students and a field study with 17 high school students in Singapore. Our results show that StratL effectively steers the LLM to adhere to the PF tutoring strategy and does so with no negative spillover effects on other desirable properties of the LLM.

We summarize our contributions as follows:

- We define the problem of *Pedagogical Steering* in the context of LLM multi-turn tutoring. We introduce the StratL algorithm to model the concept of *tutoring strategy* and to prompt an LLM towards a given strategy using a sequence of single-turn intents.

- We openly release the first dataset of PF tutoring exercises. These open-ended exercises have multiple suboptimal solutions and require multiple interactions to find the best one.

- We run a field test in a Singaporean high school to evaluate the potential real-world impact of StratL. We measure the extent to which StratL and its ablations control the pedagogy through novel learning sciences–inspired metrics and investigate the limitations.

2 Background and Related Work

2.1 Pedagogical shortcomings of LLMs

Recent advances enable state-of-the-art LLM-based systems to reliably perform middle– to high school–level mathematics, such as using prompting (Wu et al., 2023; Kojima et al., 2022), multiple sampling strategies with consistency sampling (Wang et al., 2023) or refinement (Madaan et al., 2024; Shridhar et al., 2024; Li et al., 2023), fine-tuning (Zelikman et al., 2022; Yuan et al., 2023) or use of external tools like symbolic solvers or code interpreters (He-Yueya et al., 2023; Mialon et al., 2023).

Notwithstanding the promising capabilities of LLMs, several studies exhibited their poor pedagogical properties (Tack and Piech, 2022). Among other flaws, most models often reveal the solution of the exercise to the students, even if prompted not to do so (Macina et al., 2023a). These pedagogical limitations stem from a core problem: LLMs are not optimized for tutoring. After their initial pre-training, an *alignment* step is necessary to turn base language models into useful assistants capable of following human instructions, like ChatGPT. The alignment is often performed by fine-tuning LLMs

on human preferences (Zhang et al., 2023), mainly using a method known as Reinforcement Learning with Human Feedback (RLHF) (Ouyang et al., 2022). During this process, LLMs learn to comply with requests by optimizing user satisfaction.

LLM assistants thus generate texts that optimize for human preferences. This is misaligned with how a tutor should behave. Indeed, when a student asks for help, the tutor should not necessarily maximize the student's satisfaction, but rather their learning. For this, the tutor may have an intent (for example, asking back a question) different from what the student would like them to do (for example, giving them the solution). This mismatch is one of the core limitations of LLMs as tutors.

2.2 LLMs for Tutoring

Several works have attempted to address the lack of pedagogical properties of LLMs. The easiest and most direct approach is to write an extensive prompt (Wang et al., 2024b; Sonkar et al., 2023; Pal Chowdhury et al., 2024). However, enumerating all desired behaviors of a tutor in words is complex, especially for nuanced pedagogical strategies.

Another method considered is fine-tuning LLMs to efficient tutoring strategies. However, supervised fine-tuning is hampered by the scarcity of tutoring datasets, as their collection is costly and poses ethical concerns (Kasneci et al., 2023). Macina et al. (2023a) adopt an inverted Wizard-of-Oz approach by asking human teachers to tutor GPT-simulated students to create a comprehensive dataset of high-quality tutoring sessions and show that fine-tuning LLMs on this dataset improves their pedagogy. Similarly, the CLASS framework (Sonkar et al., 2023) and Book2Dial (Wang et al., 2024a) uses an LLM to generate a synthetic conversational dataset with student-tutor interactions to fine-tune a smaller LLM. However, the quality of these conversations is not evaluated, and is prone to hallucinations. Additionally, fine-tuning is expensive and restricts the LLM to the specific teaching style present in the training data, thus requiring multiple datasets for different pedagogical approaches. LearnLM (Jurenka et al., 2024; LearnLM Team, 2024) alleviates this issue by integrating pedagogical instruction following tuning, but still heavily relies on synthetic data and a costly co-training process where LearnLM is trained alongside a generic LLM. The Bridge method (Wang et al., 2024b), which we use as an inspiration for one of our baselines, uses

an LLM to classify student errors and decide tutor interventions on the conversation history. Systems based on Bridge let the LLM choose the tutoring strategy on its own, assuming its ability to make pedagogically sound choices.

In contrast, our approach focused on letting learning scientists control the tutoring behavior. To do so, we employ a structured pedagogical intent selection using a transition graph based on classified student states, enabling experts to constrain the model to more faithfully follow a given pedagogy over multi-turn conversations.

2.3 Productive Failure (PF)

In education, "Instruction followed by Problem Solving" approaches, which directly instruct students on how to approach a problem and then let them practice, are widely used. However, PF (Kapur, 2008, 2016) rather uses the reversed order. PF is a "Problem Solving followed by Instruction" learning design, in which the learner attempts to solve open-ended problems before formally learning how to approach them, to build a deeper understanding of the topic. With PF, the teacher mostly uses questioning to let students engage cognitively with the problem. There is a large body of evidence that PF is effective with meta-analyses reporting an effect size of 0.87 standard deviations (high by the Learning Sciences standards) (Sinha and Kapur, 2021), when compared to traditional teaching.

A successful problem-solving session in PF provides opportunities for the learner to explore the problem and build as many Representation and Solution Methods (RSMs) as possible, even if they fail at solving the problem (Kapur and Bielaczyc, 2012). In simple terms, these RSMs can be seen as mental representations of the problem, or unique solution attempts. The number of RSMs generated is one of the best predictors of how much students learn from PF (Kapur and Rummel, 2012). Our tutoring strategy thus aims to promote the generation of such RSMs, and we will use their number as one of the evaluation metrics.

Overall, PF is a suitable learning design to validate our approach because (1) it is a widely successful pedagogical strategy, and (2) opposed to native LLM behavior as a PF tutor should not directly reveal the solution to the students.

3 Methodology

3.1 Formalism

Dialog Tutoring Task We extend the formalization of the Dialog Tutoring task defined in (Macina et al., 2023b). We consider a tutoring dialog history $\mathcal{H}_{\leq n} = (u_1, \ldots, u_n)$ made of tutoring turns $u_t = (y_t, x_t)$ with y_t the tutor's utterance and x_t the student's utterance at turn t. Each tutor utterance is associated with a set of *tutoring intents* $\mathbf{i}_t$ whose elements are chosen from a taxonomy of intents $\mathcal{I} = \{I_1, \ldots, I_{|\mathcal{I}|}\}$. An *intent* is defined as a goal of the tutor utterance, e.g. `correct the student's mistake`. We denote by K the grounding information, i.e. additional information provided to the LLM to give context. In our case, K includes the tutoring exercise and its solution. The resulting generative conditional model produces output:

$$y_{n+1} \sim p(\, y \mid \mathbf{i}_{n+1}, \mathcal{H}_{\leq n}, K) \qquad (1)$$

Tutoring Strategy Modeling We define a *tutoring strategy* function as a function ϕ that takes as input $(K, \mathcal{H}_{\leq t})$, the grounding information and dialog history up to any turn t, and that outputs a set $\mathbf{i}_{t+1} \subseteq \mathcal{I}$ of single-turn tutoring intents for turn $t + 1$. Abstractly, ϕ represents the decision process by which an expert tutor, given the context of the conversation, selects the best possible intents from a pedagogical standpoint to react to the student's last utterance. This function will be replaced in practice by an approximation $\tilde{\phi}$ introduced in the next paragraph. The resulting conditional tutor model generates y_{n+1} according to:

$$y_{n+1} \sim p(\, y \mid \phi(\mathcal{H}_{\leq n}, K), \mathcal{H}_{\leq n}, K) \qquad (2)$$

Student State Tracing Expert tutors base their interventions on observations about (1) the student's emotional or knowledge *state* (Lepper et al., 1993), and (2) the state of the conversation and their previous interventions (Merrill et al., 1992). For example, a tutor does not react the same way to a computational mistake as to a reasoning one. Therefore, we use *state features* (1) we define in the next paragraph, and the previous tutoring intents (2) as inputs to our intent selection function $\tilde{\phi}$.

Each student turn is associated with a set $\mathbf{s}_t$ of *state features* describing the state of the student and of the conversation at turn t. The state features are chosen from a set $\mathcal{S}$ of possible features. A feature in $\mathcal{S}$ could, for example, be 'the

student is confused' or 'the student made a computational mistake'. We denote ψ a classifier that provides these features based on the dialog history and grounding. Our intent selection function thus writes $\widetilde{\phi}(\mathbf{i}_n, \mathbf{s}_n) = \mathbf{i}_{n+1} \subseteq \mathcal{I}$. As previously noted, $\widetilde{\phi}$ depends on the student state features $\mathbf{s}_n$ and previous tutor intents $\mathbf{i}_n$. Our final conditional generation model produces output:

$$y_{n+1} \sim p(\, y \mid \widetilde{\phi}\,(\mathbf{i}_n, \psi(\mathcal{H}_{\leq n}, K)), \mathcal{H}_{\leq n}, K) \quad (3)$$

3.2 Algorithm

To steer the LLM towards a multi-turn pedagogical strategy, we introduce StratL, an algorithm for the formalization (3) given in Section 3.1. StratL dynamically updates the tutoring intents after every student utterance and prompts the LLM to follow them at the single-turn level. A visual representation of StratL is presented in Figure 1. This algorithm is based on three sub-procedures: I. Student State Tracing, II. Tutor Intent Selection and III. Intent Dependent Steering.

I. The Student State Tracing mirrors the function ψ. This step is a multi-label classification of the dialog history and latest student utterance at turn t. The selected features $\mathbf{s}_n$ are chosen from a pre-defined set of possible features $\mathcal{S}$. These features fall into four categories: (1) mistake types, (2) student requests, (3) conversation states, and (4) emotional states. These categories each provide valuable information to be used in the Intent Selection, i.e.: (1) helps plan error remediation (e.g. a reasoning and a computational mistake should not be dealt with equally) (Daheim et al., 2024; Wang et al., 2024b), (2) is necessary as different student requests call for different reactions (e.g. if the student asks for the solution v.s. if they ask for a definition), (3) helps structure the conversation (e.g. knowing if the problem is solved) and (4) allow for proper emotional responses from the tutor (Lepper et al., 1993). The precise features we use are detailed in Appendix B.2.1. State tracing is performed by an LLM prompted to identify these different features given the set of possible student states $\mathcal{S}$ and the dialog history. Implementation details such as the prompt we used are provided in Appendix B.2.1.

II. The Tutor Intent Selection mirrors the function $\widetilde{\phi}$. We model tutor intent selection as a deterministic transition from the previous set of intents $\mathbf{i}_n$ to the next set of intents $\mathbf{i}_{n+1}$, conditioned on
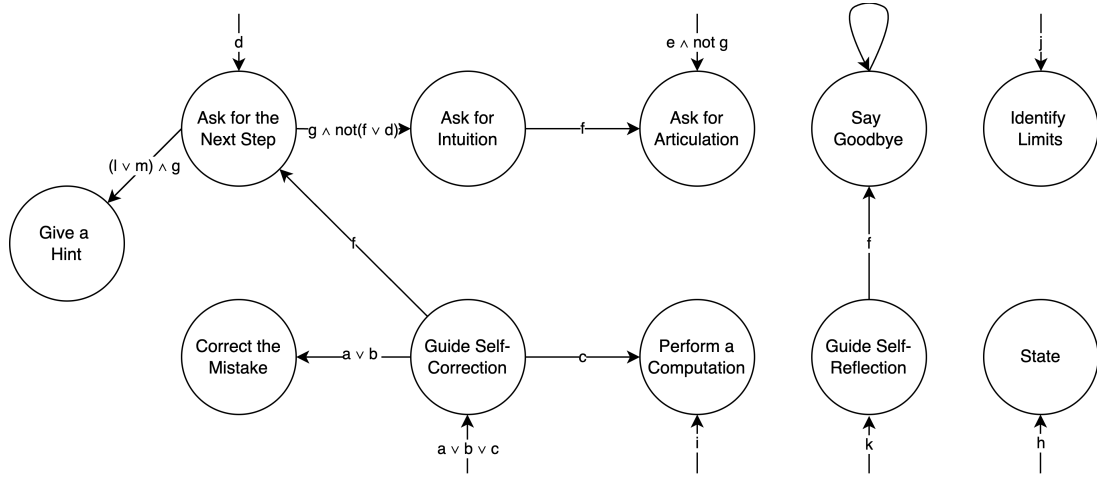
Figure 2: Transition Graph of a PF (Kapur, 2008) Intent Selection. Nodes correspond to intents (listed in Table 5 of Appendix B.1). The variables used in the arrows' transition conditions ('a', ..., 'm') are codes for the state features (displayed in Figure 8 of Appendix B.2.1). At each turn, we transition from one set of intents to the next by following all satisfied arrows. Arrows with no starting node can be taken from any node. At turn 0, we start at node 'Ask for the Next Step'. We describe the Learning Sciences justifications for this PF modeling in Appendix B.3.

the output $\mathbf{s}_n$ of step I. The transition rules are defined by a learning scientist expert. We represent these transitions with a graph. Please note that, unlike traditional deterministic state transition models, our graph allows to transition from a *set* of nodes to another by following multiple concurrent transitions.

We used the theories of Scaffolding (Reiser, 2004), Affective Cognition (Lepper et al., 1993), and PF (Sinha and Kapur, 2019; Kapur and Bielaczyc, 2012) to design a taxonomy of tutoring intents $\mathcal{I}$. These intents are mainly separated into: (1) *scaffolding intents*, helping the student by restricting the solution space (e.g. giving a hint), and (2) *problematizing intents*, helping the student deepen their understanding by exploring the solution space (e.g. asking to formalize an intuition). For our PF strategy, we prioritize scaffolding when the student is stuck and use problematizing intents whenever the student shows an understanding of the problem, to encourage the free exploration of the problem and the generation of multiple RSMs. The exhaustive taxonomy is displayed in Table 5 of Appendix B.1 and a detailed explanation of the Learning Science principles we used to design our PF Intent Selection is given in Appendix B.3.

Our PF Intent Selection graph is represented in Figure 2. Each node represents an intent of $\mathcal{I}$. Arrows correspond to transitions between intents. We initialize start at the 'Ask for the Next Step' node and move at each tutoring turn. The letters used in the logical conditions on the arrows are

codes corresponding to Student State Tracing features (detailed in Figure 8 of Appendix B.2.1). For the sake of clarity, transitions that can be taken from any intent are represented by arrows with no starting node.

III. The Intent Dependent Steering generates the LLM Tutor prompt p_{n+1} conditioned on the selected tutoring intents $\mathbf{i}_{n+1}$. We defined a one-to-one correspondence between the intents in $\mathcal{I}$ and short prompt chunks. The precise correspondence is given in Appendix B.2.2.

4 Experiments

Experimental Setup. As a case study, we use StratL to implement a prototype high school math tutor based on GPT-4o (gpt-4o-2024-08-06) and following a PF–inspired tutoring strategy. Our tutor discusses one problem per tutoring dialog and has access to its solution. The exact prompts we used in our prototype, and the exhaustive list of state features, intent taxonomy, and intent-to-prompt correspondences are given in Appendix B.2.

4.1 Research Questions

We validate the ability of our system to control the tutoring strategy employed by the LLM by assessing the capacity of our prototype to follow the PF paradigm. Ideally, without hindering the factual correctness or desirable conversational properties of LLMs such as their coherence. We formulate these goals in two research questions:

RQ1. To what extent does StratL make the LLM Tutor follow PF?

RQ2. Does StratL influence on other desirable properties of the LLM Tutor?

We chose to handcraft the Intent Selection mechanism with the help of PF experts. This is different from Wang et al. (2024b) which lets an LLM choose how to react. We thus formulate a design question to test our assumption: **DQ1.** How does the intent selection by LLM compare to our approach?

4.2 Dataset

We compiled a dataset of 15 PF problems. These problems are open-ended, ill-structured, and engaging. They feature multiple suboptimal solutions (and a single optimal one) and require multiple interactions from students to explore their solution space. These serve as high-quality testing problems for evaluating a tutoring system's pedagogical skills. Our algorithm is tested on two 9th-grade math problems (successfully solved by 62% of our students in the field test). The first problem, `Consistency`, is adapted from Kapur and Toh (2015) and is a well-studied *invention problem* where students devise a consistency measure (typically variance) to compare football players' performance. An effective PF tutor for this problem would help the students construct as many ways to measure consistency as possible. The second problem, entitled `Country`, is an open-ended problem taken from Kerrigan (2018) requiring students to fairly allocate voting seats across states based on their population. On this problem, an effective PF tutor would guide reasoning without explicitly providing key steps like the idea of using proportionality. The exact problem statements, solutions, and the dataset specifications, intended use, and license are provided in Appendices A.1 and A.2.

4.3 Baselines

We test our StratL-controlled tutor LLM (**V1**) against three ablation versions. **V2: No Intent** is a baseline LLM prompted to act like a tutor with our base prompt (but no intent-dependent prompt). **V3: Constant Intent** always uses the same intent ("Ask for Next Step" in Table 5 of Appendix B.1) that only asks the student to proceed with the exercise. We use this baseline to assess the importance of the Intent Selection procedure. **V4: LLM Intent** use an LLM to perform the Intent Selection

(with the prompt given in Appendix B.2.3) Wang et al. (2024b), and we use it to answer **DQ1**.

4.4 Evaluation Metrics

To quantitatively assess the *PF fidelity* of the different versions (their success in following PF), we have to rely on proxy variables. We propose as a main proxy the **number of student-generated Representation and Solution Methods** (RSMs), which can be seen as the different ways students represent the problem and try to solve it. As explained in Section 2.3, the diversity of student-generated RSMs is the first predictor of learning gains with PF (Kapur and Bielaczyc, 2012; Kapur and Rummel, 2012). We thus use this number to measure the PF fidelity of the tutor.

However, not every problem easily admits multiple RSMs. In our case, only `Consistency` does. We thus define another proxy, the **PF Score**. One of the core ideas of PF is to encourage the student to construct their own understanding of the problem by exploring it by themselves, even if they do not succeed in solving it. Building on this idea, we define for each problem a grading rubric used to compute the tutor's PF Score: For each problem, we identify a list of critical reasoning steps that represent different conceptual understandings of the problem. To assess a tutoring conversation, we give a point to the tutor for every crucial reasoning step that the student found on their own. We do not award the point if the tutor reveals it or directly hints to it. This metric can be thought of as an adjusted sub-step success rate. An efficient PF tutor would help the student explore the problem and thus achieve these conceptual understandings, without giving them any direct hint or revealing them, resulting in a high PF Score. We display the precise sub-steps for both testing problems in Appendix A.3.

4.5 Preliminary Study with Simulated Students

Due to ethical concerns (Kasneci et al., 2023), we first tested our prototype with GPT-4–simulated students before involving humans. This method of using simulated students, known as interactive tutoring, is used in the literature (Macina et al., 2023a; Sonkar et al., 2023). Importantly, we do not assume that the LLM can accurately model students' learning. However, this assumption is not necessary for our preliminary study as we are interested in the ability of our algorithm to follow a

PF tutoring strategy, rather than the learning gains it produces. We simulated three conversations for each of two problems across four tutor versions (V1–4) and used a scoring rubric to annotate conversations, computing the PF Score and RSM count (as described in Section 4.4).

4.6 Field Test with Students

We ran the field test with 17 9th-grade students from a high school in Singapore. Each student was randomly (uniformly) assigned to one of two conditions (V1 or V2, chosen as described in Section 5.1) and one of the two exercises. They had 15 minutes to interact with the tutor. We manually annotated their conversations using our rubric to compute the PF score and the number of generated RSMs.

Post-study Questionnaire. After the tutoring session, students were asked about their experience to assess potential spillover effects of StratL on the tutor LLM's desirable properties. Using a 3-point Likert scale, they rated the AI tutor on: a) coherence, b) empathy, and c) helpfulness.

5 Results and Discussions

5.1 Preliminary Study with Simulated Students

	# RSMs	PF Score	
		Consistency	Country
V1. StratL	**2.34**	3.67	**4.00**
V2. No Int.	1	1.33	0.44
V3. Constant Int.	1.34	**4.00**	**4.00**
V4. LLM Int.	1.34	3.67	3.11

Table 1: PF fidelity metrics for different tutor versions and different problems in the preliminary study with simulated students. Best values in bold.

DQ1. *Using an LLM prompted with a natural language description of the tutoring strategy is not enough for the Intent Selection.* The preliminary study in Table 1 suggests that StratL (V1) is successful in steering the LLM tutor towards PF, as evidenced by the greater number of RSMs elicited by this version. We observe that the version using StratL with an LLM responsible for the Intent Selection (V4) did perform better than the baseline LLM (V2), but not as well as StratL with our hand-crafted Intent Selection mechanism. This result contributes to an answer to **DQ1** by showing that providing a natural language description of the tutoring strategy to follow and letting an LLM choose

the intents accordingly does not yield satisfactory results in terms of tutoring strategy fidelity.

RQ1. *The strategy followed by an LLM controlled by StratL is determined by the Intent Selection procedure.* Overall, V3 achieves a rather high PF Score on the two problems, since it never provides any hint to students. However, this result is more a consequence of the way we defined the PF score rather than a sign of PF fidelity, as V3 does not perform as well on our second proxy: It does not manage to elicit as many RSMs from students as V1, with an average of only 1.34. This shows that the strategy followed by an LLM controlled by StratL is determined by the Intent Selection procedure as it is the only part that differs (V1 vs. V3).

Based on our preliminary answers to **DQ1** and **RQ1**, **we decided not to include V3 and V4 in our field test** with real students to focus on the best-performing version (V1) while keeping the non-StratL baseline (V2) for comparison.

Accuracy of Student State Tracing The functioning of StratL as a whole is contingent upon the functioning of the Student State Tracing. To test its reliability, we generate ten tutoring conversations for both problems and the first author manually annotates them with ground truth state features s_t. We compare the outputs of the Student State Tracing with the annotated data. As a multi-label classification task, we compute for each state feature the precision, recall, and F1 score. We compare three different implementations of the State Tracing step and provide more details about our methodology and results in Table 4 of Appendix A.4. Our implementation achieves a satisfactory micro F1 score of 0.77. The Student State Tracing is a task of its own, and we encourage researchers to try different classification models to improve its reliability.

5.2 Field Test with Students: Measuring end-to-end Effectiveness

RQ1. Tutoring Strategy Fidelity *StratL is successful in steering the LLM to follow the PF tutoring strategy.* We report in Figure 3 the results of the protocol defined in Section 4.4. We note the PF score (out of 4) is used as a proxy for the extent to which the tutor followed the PF tutoring strategy. We also report the average number of student-generated RSMs (partial solutions) on the problem Consistency. This metric is best proxy for PF fidelity, as one of the goals of the PF tutoring strategy is to maximize this number, as explained in Section 2.3. Out of the two test problems, the number of
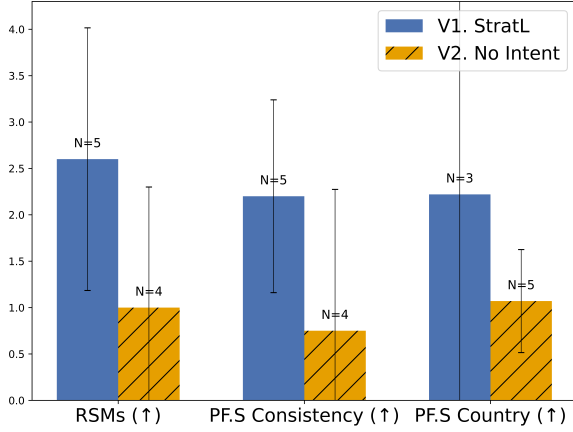
Figure 3: Field test results and their confidence intervals. 'PF.S' refers to the 'PF Score' as defined in Section 4.4 and is out of 4. 'RSMs' denotes the number of student-generated RSMs. StratL (V1) succeeds in steering the LLM to follow PF as compared to V2.

	N	Coherence	Empathy	Helpful.
Problem: Consistency				
V1.	5	1.80	1.60	1.20
V2	4	**2.25**	**1.75**	**2.00**
Problem: Country				
V1	3	**3.00**	**2.67**	2.67
V2.	5	**3.00**	2.20	**2.80**

Table 2: Feedback questionnaire results (out of 3). 'N' refers to the number of answers to the questionnaire. We do not detect any harmful effect of StratL (V1) on the LLM tutor's coherence, empathy, or perceived helpfulness.

RSMs can only be measured on `Consistency` as it is the only invention problem.

On both test problems, the StratL-controlled LLM (V1) achieved a higher (statistically significant for problem `Consistency`, $p = .046$ with a 2-sample t-test) PF score than the baseline LLM (V2), signifying a higher fidelity to the PF tutoring strategy. Furthermore, the baseline LLM (V2) failed to elicit multiple RSMs from students, while a student tutored by the StratL-controlled LLM produced on average 2.6 RSMs during the tutoring session (the difference is statistically significant, $p = .05$ with a 2-sample t-test). In that regard, V1 succeeded in the PF process of making students explore as many solutions as possible.

RQ2. Student Feedback Questionnaire The feedback questionnaire did not reveal particular

spillover effects of StratL on the coherence, empathy or perceived helpfulness of the tutor LLM.

StratL has no or little influence on the capacity of the tutor LLM to generate coherent responses. StratL (V1) scores similarly or slightly less than an LLM with no intent (V2) on coherence across the two problems: 1.80 out of 3 for (V1) *vs.* 2.25 for (V2) when used on problem `Consistency`, and 3.00 for both on `Country`. We do not detect any statistically significant difference.

StratL does not influence the capacity of the LLM to generate empathetic answers. None of the two versions is perceived as significantly more empathetic than the other (1.60 for V1 *vs.* 1.75 for V2 on `Consistency` and 2.67 *vs.* 2.20 on `Country`).

StratL can influence the perceived helpfulness of the tutor LLM, through the tutoring strategy it makes it follow. We find that the baseline with no intents (V2) is perceived by the students as more helpful than V1 (2.00 *versus* 1.20 respectively on `Consistency` and 2.80 *versus* 2.67 on `Country`). This is not surprising, as the PF tutoring strategy can be frustrating or misunderstood by students. Indeed, the PF strategy is designed to only give help when it is strictly necessary, which means that students spend more time being stuck and exploring suboptimal solutions. This can lead students to perceive the tutor as less helpful compared to a version that would give more help. We report the detailed results in Table 2.

5.3 Questionnaire Feedback

Our field test demonstrated the ability of the StratL tutor to follow PF. This section highlights some qualitative insights from the field study feedback.

Robustness As discussed in Section 5.1, the Student State Tracing procedure can make labeling mistakes. These mistakes can then compromise the Intent Selection and result in the selection of contradictory intents. This is exemplified by the feedback of a student: *"The AI keeps on asking me to check my answer although it was correct"*.

Solution constraint A student assigned to the no intents baseline (V2) complained that the tutor would not let them use a solution entirely different from the expected one: *"The AI only listens to key points that it wants, and ignores everything else (e.g. my idea of using [...])"* A good tutor should encourage original solutions. This highlights a pedagogical problem, as in this case, the tutor is harmful to the learning by restricting the student.

6 Conclusion

We present the pedagogical limitations of LLMs in tutoring as the Pedagogical Steering problem. We introduce StratL, an algorithm addressing this problem. StratL integrates ideas from Learning Sciences with LLMs, utilizing a Student State Tracing and a Tutoring Intent Selection procedure based on expert-defined transition graphs to dynamically steer the LLM to follow a modeled pedagogy. A field study with high school students demonstrates StratL's ability to follow the Productive Failure learning design.

Limitations

Evaluation aspects. The field study uses a rubric-based evaluation and learner-perceived usefulness in an end-to-end scenario. We argue additional evaluation metrics such as multiple human teachers evaluating the quality across the dimensions such as engaging the student or quality of questioning. Moreover, as we evaluated StratL through a field study, learners might have taken the exercise less seriously than they would for an exercise that is officially part of their curriculum. Further studies with increased ecological validity should be conducted.

Integration. While tutoring can also be explored as a standalone experience, it is important to consider how systems like StratL could be integrated into a classroom or as part of a formal education context. As such, further research should explore (1) how to support teachers in adjusting and personalizing the tutor's strategy, (2) how to provide relevant feedback to the teacher in order to ensure continuity with classroom content, and (3) how such tutors should be best integrated in existing pedagogical structures.

Risks. As students increasingly utilize AI tools in their learning process, we must ensure they are beneficial to learning. Some studies empirically demonstrated that generative AI tools could harm learning, through mechanisms such as over-reliance (Zhai et al., 2024). With our work, we mitigate these effects by empowering teachers and learning scientists to control their student's interactions with generative AI through the use of StratL. However, further field studies on the reliance behavior of students with a StratLLLM tutor should be conducted before large-scale deployment. Wang et al. (2024c) discusses more broadly the risks associated with the use of LLMs in education.

Social aspects. The social context of a learning experience plays an important role in learning, and "social surround" is even one of the three design layers of PF (Kapur and Bielaczyc, 2012). Generally, supportive and coordinated relationships play an important role in learning (Roschelle and Teasley, 1995), and research on how students perceive these tutors at the social level should be explored to paint a complete picture. Beyond cognitive learning outcomes, it is also important to consider affective outcomes of learning activities, such as social-emotional competencies (Alzahrani et al., 2019), and how these are impacted by the interaction with digital tutors over human ones.

Data Privacy. The field study was approved by the university IRB (Proposal EK-2024-N-256). All data collected were anonymized and no sensitive student information was stored.

References

Mona Alzahrani, Manal Alharbi, and Amani Alodwani. 2019. The effect of social-emotional competence on children academic achievement and behavioral development. *International Education Studies*, 12(12):141–149.

Benjamin Samuel Bloom. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13:16–4.

Peter A Cohen, James A. Kulik, and Chen-Lin C. Kulik. 1982. Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal*, 19(2):237–248.

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*, pages 8386–8411, Miami, Florida, USA. Association for Computational Linguistics.

Scott Freeman, Sarah L. Eddy, Miles McDonough, Michelle K. Smith, Nnadozie Okoroafor, Hannah

Jordt, and Mary Pat Wenderoth. 2014. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23):8410–8415.

Joy He-Yueya, Gabriel Poesia, Rose Wang, and Noah Goodman. 2023. Solving math word problems by combining language models with symbolic solvers. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*.

Irina Jurenka, Markus Kunesch, Kevin R McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, Shubham Milind Phal, Katherine Hermann, Daniel Kasenberg, Avishkar Bhoopchand, et al. 2024. Towards responsible development of generative ai for education: An evaluation-driven approach. *arXiv preprint arXiv:2407.12687*.

Manu Kapur. 2008. Productive failure. *Cognition and Instruction*, 26(3):379–424.

Manu Kapur. 2010. Productive failure in mathematical problem solving. *Instructional Science*, 38(6):523–550.

Manu Kapur. 2016. Examining productive failure, productive success, unproductive failure, and unproductive success in learning. *Educational Psychologist*, 51(2):289–299.

Manu Kapur. 2024. *Productive Failure: Unlocking Deeper Learning Through the Science of Failing*. Wiley.

Manu Kapur and Katerine Bielaczyc. 2012. Designing for productive failure. *Journal of the Learning Sciences*, 21(1):45–83.

Manu Kapur and Nikol Rummel. 2012. Productive failure in learning from generation and invention activities. *Instructional Science*, 40(4):645–650.

Manu Kapur and Leslie Toh. 2015. *Learning from Productive Failure*, pages 213–227. Springer Singapore, Singapore.

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Sandra Katz, Gabriel O'Donnell, and Heather Kay. 2000. An Approach to Analyzing the Role and Structure of Reflective Dialogue. *International Journal of Artificial Intelligence in Education*, 11:320–343. Part I of the Special Issue on Analysing Educational Dialogue Interaction (editor: Rachel Pilkington).

John Kerrigan. 2018. Productive failure in the flipped mathematics classroom. *Rutgers University*.

John Kerrigan, Keith Weber, and Clark Chinn. 2021. Effective collaboration in the productive failure process. *The Journal of Mathematical Behavior*, 63:100892.

Sal Khan. 2023. Harnessing gpt-4 so that all students benefit. a nonprofit approach for equal access.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

LearnLM Team. 2024. Learnlm: Improving gemini for learning. *Preprint*, arXiv:2412.16429.

Marc R. Lepper, Maria Woolverton, Donna L. Mumme, and Jean-Luc Gurtner. 1993. *Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors.*, pages 75–105. Technology in education. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, US.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Katharina Loibl and Nikol Rummel. 2014. Knowing what you don't know makes failure productive. *Learning and Instruction*, 34:74–85.

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

Jakub Macina, Nico Daheim, Lingzhi Wang, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023b. Opportunities and challenges in neural dialog tutoring. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2357–2372, Dubrovnik, Croatia. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Claudia Mazziotti, Nikol Rummel, and Anne Deiglmayr. 2016. Comparing students' solutions when learning

collaboratively or individually within productive failure. Singapore: International Society of the Learning Sciences.

Douglas C. Merrill, Brian J. Reiser, Michael Ranney, and J. Gregory Trafton. 1992. Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences*, 2(3):277–305.

Grégoire Mialon, Roberto Dessi, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Roziere, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *Transactions on Machine Learning Research*. Survey Certification.

Phillip M. Newman and Marci S. DeCaro. 2019. Learning by exploring: How much guidance is optimal? *Learning and Instruction*, 62:49–63.

OpenAi. 2024. Prompt engineering. https://platform.openai.com/docs/guides/prompt-engineering/. Accessed: 2024-03.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Sankalan Pal Chowdhury, Vilém Zouhar, and Mrinmaya Sachan. 2024. Autotutor meets large language models: A language model tutor with rich pedagogy and guardrails. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, L@S '24, page 5–15, New York, NY, USA. Association for Computing Machinery.

Brian J. Reiser. 2004. Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *Journal of the Learning Sciences*, 13(3):273–304.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA. Association for Computing Machinery.

Jeremy Roschelle and Stephanie D Teasley. 1995. The construction of shared knowledge in collaborative problem solving. In *Computer supported collaborative learning*, pages 69–97. Springer.

Kumar Shridhar, Jakub Macina, Mennatallah El-Assady, Tanmay Sinha, Manu Kapur, and Mrinmaya Sachan. 2022. Automatic generation of socratic subquestions for teaching math word problems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4136–4149, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kumar Shridhar, Koustuv Sinha, Andrew Cohen, Tianlu Wang, Ping Yu, Ramakanth Pasunuru, Mrinmaya Sachan, Jason Weston, and Asli Celikyilmaz. 2024. The ART of LLM refinement: Ask, refine, and trust. 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 5872–5883, Mexico City, Mexico. Association for Computational Linguistics.

Tanmay Sinha and Manu Kapur. 2019. When productive failure fails. In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society (CogSci 2019): Creativity + Cognition + Compulation*, pages 2811 – 2817, s.l. COGSCI. 41st Annual Meeting of the Cognitive Science Society (CogSci 2019); Conference Location: Montreal, Canada; Conference Date: July 24-27, 2019.

Tanmay Sinha and Manu Kapur. 2021. When problem solving followed by instruction works: evidence for productive failure. *Review of Educational Research*, 91(5):761–798.

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, Singapore. Association for Computational Linguistics.

Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. In *Proceedings of the 15th International Conference on Educational Data Mining*, page 522.

Junling Wang, Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, and Mrinmaya Sachan. 2024a. Book2Dial: Generating teacher student interactions from textbooks for cost-effective development of educational chatbots. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9707–9731, Bangkok, Thailand. Association for Computational Linguistics.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024b. 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, Mexico City, Mexico. Association for Computational Linguistics.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024c. Large language models for education: A survey and outlook. *CoRR*, abs/2403.18105.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowd-

hery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *ICLR 2023*.

Dingjun Wu, Jing Zhang, and Xinmei Huang. 2023. Chain of thought prompting elicits knowledge augmentation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6519–6534, Toronto, Canada. Association for Computational Linguistics.

Tongshuang Wu, Michael Terry, and Carrie J. Cai. 2021. AI chains: Transparent and controllable human-ai interaction by chaining large language model prompts. *CoRR*, abs/2110.01691.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *Preprint*, arXiv:2308.01825.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.

Chunpeng Zhai, Santoso Wibowo, and Lily D Li. 2024. The effects of over-reliance on ai dialogue systems on students' cognitive abilities: a systematic review. *Smart Learning Environments*, 11(1):28.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

A Appendix Evaluation Details

A.1 Dataset

We collected 15 problems from the PF literature (Kapur, 2008; Kapur and Bielaczyc, 2012; Kapur, 2010; Kerrigan et al., 2021; Kerrigan, 2018; Kapur and Toh, 2015; Loibl and Rummel, 2014; Newman and DeCaro, 2019; Mazziotti et al., 2016). We present the columns of the dataset in Table 3. We curated three types of problems: *Invention* problems ask the student to invent a new way to solve a problem. They typically admit several correct but suboptimal answers, as long as the student can justify them. Students are not expected to have previous exposure to the tools canonically used to solve the exercise. The problem `Consistency` is a good example of *invention* problem. *Ill-structured* problems are open-ended problems that contain some non-relevant information and a convoluted story. These problems often admit a single solution, but there can be several ways to reach it. The problem

`Country` is an example of such a problem. *Well-structured* problems are more direct versions of the *ill-structured* problems. They resemble classical exercises that students usually solve in a mathematics class. More information about the different types of exercise in PF and their role can be found in (Kapur, 2008) and (Kapur and Bielaczyc, 2012). Our dataset is intended to be used as a set of high-quality testing problems for evaluating a tutoring system's pedagogical skills.

We release it under CC-BY-4.0 license[3].

A.2 Test Problems

Figures 4 and 5 display the two test problems we used in our field test. They are also included in the dataset we release.

A.3 PF Score and RSM Details

PF Score We present in the following paragraphs the precise reasoning sub-steps used for the PF Score grading as explained in Section 4.4.

For problem `Consistency`, we define these reasoning steps as 1. Giving an intuitive definition of consistency related to the variation of scores, 2. Thinking about comparing the scores to a baseline (central value or previous year), 3. Considering absolute differences (by taking the absolute value or squaring) since only the magnitude of the change matters, and 4. Thinking about a way of aggregating all the differences into a single score, for example by summing or averaging them.

For problem `Country`, we define these steps as 1. Thinking about using a 'standard divisor' which represents the number of seats per inhabitant, 2. Thinking about rounding down the number of seats not to attribute more seats than the total, and 3. Thinking about using the remainders to distribute the additional seats. We linearly scale the PF score to 4 in this case.

RSM We tracked the number of student-generated Representation and Solution Methods (RSMs) in problem `Consistency`. We observed students proposing various solutions to the problem, including comparing the three soccer players based on the variance of their number of goals scored or on their mean absolute year-to-year differences.

---

[3] https://creativecommons.org/licenses/by/4.0/deed.en

type	grade	name	reference	exercise	solution
Type between invention, ill-structured, and well-structured	Recommended school grade	Name to quickly identify the exercise	Article the exercise was curated from	Exercise statement	Solution if provided in article

Table 3: Description of the fields in the dataset.

A new country has recently been founded. The country is split into six states, call them A, B, C, D, E, and F.

The population of state A is 1,646,000 people, the population of state B is 6,936,000 people, the population of state C is 154,000 people, the population of state D is 2,091,000 people, the population of state E is 685,000 people, and the population of state F is 988,000 people.

There are 250 seats available on a legislative body to govern the new country. How many seats should be assigned to each state so that each state would receive a fair representation?

Show your work and justify why you think your method is correct.

Solution:
We assign sits proportionally to the population of each state. Since the results of the divisions are not integers, we round down the number and then distribute the remaining sits to stats having the largest remainders.

Total population = 1,646,000 (A) + 6,936,000 (B) + 154,000 (C) + 2,091,000 (D) + 685,000 (E) + 988,000 (F) = 12,500,000

Standard divisor = Total population / Number of seats = 12,500,000 / 250 = 50,000

Initial quotas:
- A: 1,646,000 / 50,000 = 32.92 → 32 seats
- B: 6,936,000 / 50,000 = 138.72 → 138 seats
- C: 154,000 / 50,000 = 3.08 → 3 seats
- D: 2,091,000 / 50,000 = 41.82 → 41 seats
- E: 685,000 / 50,000 = 13.70 → 13 seats
- F: 988,000 / 50,000 = 19.76 → 19 seats

Total initial seats assigned = 32 + 138 + 3 + 41 + 13 + 19 = 246

Seats left to distribute = 250 - 246 = 4

Distribute the surplus seats based on largest remainders:

Remainders (from the divisions above):
- A: 0.92
- B: 0.72
- C: 0.08
- D: 0.82
- E: 0.70
- F: 0.76

The four highest remainders are from states A, B, D, and F. Give one extra seat to each.

Figure 4: Problem `Country` and its solution.

The organizers of the Premier League Federation have to decide which one of the three players Mike Arwen, Dave Backhand and Ivan Right - should receive the "The Most Consistent Player for the Past 5 Years" award. Table 1 shows the number of goals that each striker scored between 2019 and 2023.

The organizers agreed to approach this decision mathematically by designing a measure of consistency. They decided to get your help. Here is what you must do:
(1) Design as many different measures of consistency as you can.
(2) Your measure of consistency should make use of all data points in the table.

Table 1. Number of goals scored by the three players in the Premier League between 2019 and 2023.

Year	Mike Arwen	Dave Backhand	Ivan Right
2007	13	12	14
2008	12	14	10
2009	15	16	18
2010	17	15	18
2011	13	13	15

Solution:
The concept of variance and standard deviation is unknown to students. Any measure proposed by the student is acceptable as long as it can be justified to measure consistency. The goal is for them to construct their own measure of consistency and justify it based on the data provided. Example of canonical solution: computing the variance (or standard deviation) for each player (standard deviation is also valid):

First, compute the mean:
Mean number of goals for Mike: 14
Mean number of goals for Dave: 14
Mean number of goals for Ivan: 15

Then, compute the sum of square deviations from the mean for each player:
Sum squared deviation for Mike: 16
Sum squared deviation for Dave: 10
Sum squared deviation for Ivan: 44

Then divide by the number of data points to get the variance:
Variance for Mike: 12/5 = 3.2
Variance for Dave: 10/5 = 2
Variance for Ivan: 44/5 = 8.8

So according to the variance, Dave is the most consistent player.

Figure 5: Problem `Consistency` and its solution.

A.4 Student State Tracing Evaluation

As explained in Section 5.1, we evaluated the accuracy of our Student State Tracing procedure, including a comparison against baselines.

Methodology We test the procedure using two tutoring conversations of about ten dialog turns for each of the two test problems detailed in Appendix A.2. These conversations are manually annotated to obtain the reference data. As detailed in Appendix B.2, we use an LLM to perform this task in our implementation of StratL. This LLM is prompted with the full conversation history and is asked to output a justification for the labels it selected. We

compare this implementation to a version not outputting justifications and a version that only keeps the three last utterances in the prompts' history to increase the processing speed.

As a multi-label classification task, we compute for each label the precision, recall, and F1 score. We aggregate these metrics across labels using an average weighted by the relative cardinality of each label, known as the micro average (μ-average). We report the results in Table 4.

Results Our simple Student State Tracing already achieves satisfactory results. The use of justification indeed seems to increase the State Tracing's performance as we can observe a higher precision

	μ-average precision	μ-average recall	μ-average F1
Full version	**0.82**	**0.75**	**0.77**
No Justification	0.71	0.72	0.70
Short Memory	0.64	**0.75**	0.68

Table 4: Test micro-averages for different versions of the Student State Tracing.

and F1 score for the version using it compared to the version not justifying its answers. It is worth remembering that this increased performance comes at a higher cost, since the LLM has to generate additional tokens for the justification. The State Tracing model keeping in memory only the last three dialog pairs does not perform as well as the full version: it thus appears important to provide the whole context of the conversation to obtain reliable results.

A.5 Instructions given to the Field Test Participants

Figure 6 displays the instructions and consent form given to the field test participants before entering the study.

A.6 User interface

Figure 7 displays the chat interface of our prototype tutor.

Participant ID: [ID]. **Please write down and keep this ID for future reference.**
Contact project team: [REDACTED]
Data Protection Officer: [REDACTED]

We would like to ask you if you are willing to participate in our research project. Your participation is voluntary. Please read the text below carefully and ask the contact person about anything you do not understand or would like to know.

What is investigated and how?
You will have the opportunity to test our AI math tutor. We will investigate whether our AI tutor is able to follow a given pedagogical strategy. You will be asked to solve a math exercise with the help of the tutor. We will analyze your tutoring session to extract some numerical metrics related to how good the tutor was. The results will be aggregated into statistics. You will then give some feedback on the tutor.

Who can participate?
You need to be able to write early high school-level mathematics in English to participate.

What am I supposed to do as a participant?
As a participant, you will connect to the website provided by your teacher. You will find a chat interface with an AI math tutor on this website. You will first read the AI tutorial, and then chat by text with the AI tutor to solve the exercise. After 15 minutes with the tutor, you will then fill in a feedback form on your experience with the tutor.

What are my rights during participation?
Your participation in this study is voluntary. You may withdraw your participation at any time without specifying reasons and without any disadvantages.

What risks and benefits can I expect?
We don't expect any risk for you. Your conversation with the tutor will be recorded in an anonymous fashion and won't be shared with anyone beyond the researchers involved in the project. The AI tutor might sometimes make mistakes. The tutor is programmed to follow a pedagogy that aims to improve your understanding of math. This session can thus be a very good practice session for math.

Will I be compensated for participating?
You will not be compensated for participating in this study. This educational activity will help you learn math.

What data is collected from me and how is it used?
Your tutoring session (the messages you send to the AI and the ones you receive) will be recorded. The post-test feedback form will ask you your gender, if you like math, and will ask you to rate your experience with the AI tutor. Only the authors will have access to this data, and it will be anonymous. Members of the [REDACTED] may access the original data for examination purposes. Strict confidentiality will be observed at any time.
Results will be published in conferences in a completely anonymous fashion. The conversations with the tutor will not be published nor shared beyond the project team. The data will only be published through aggregated statistics and in anonymized form and thus no conclusions can be drawn about individuals. The anonymized data will be stored on a password-protected encrypted computer and a private, password-protected online repository. All anonymized data will be stored for a maximum of two years after the end of the study.

What are my rights to my personal data?
Before the irrevocable anonymization of the collected data, you can request information about the personal data collected from you at any time and without giving reasons. You can also request that it be rectified, handed over to you, barred for processing or erased. To do so, please contact the person indicated above. Please write down and remember your participant ID indicated at the top of this form as it will be required to identify your anonymous data.
The Federal Data Protection Act (FADP), and if applicable the European General Data Protection Regulation (GDPR)) are complied with when collecting and processing personal data.

How am I insured?
You are insured for adverse health effects that are directly caused by the study and can be demonstrated to be attributable to fault on the part of the project team or [REDACTED]. You are responsible for insuring yourself against any other adverse health effects such as might occur, for instance, in connection with the trip to or from the place where the study is conducted.

Who funds this study? Who reviewed this study?
The study is funded by [REDACTED].
This study was examined by the [REDACTED].
The secretariat of the [REDACTED] is available to help you with complaints in connection with your participation. Contact: [REDACTED]

Consent Form:
I, the participant, confirm by clicking the following button that:
- I have read and understood the study information. My questions have been answered completely and to my satisfaction.
- I comply with the inclusion criteria for participation described above. I am aware of the requirements and restrictions to be observed during the study.
- I have had enough time to decide about my participation.
- I participate in this study voluntarily and consent that my personal data be used as described above.
- I understand that I can stop participating at any moment.
- I have written down my participant ID and understand that it is the only way that the authors can identify my data for any request I can have, as my data is anonymized.

Participant ID: [ID]

By clicking the button, I confirm that I have read the above information and agree to it.

Figure 6: Instructions and consent form signed by the field test participants

Problem

A new country has recently been founded. The country is split into six states, call them A, B, C, D, E, and F. The population of state A is 1,646,000 people, the population of state B is 6,936,000 people, the population of state C is 154,000 people, the population of state D is 2,091,000 people, the population of state E is 685,000 people, and the population of state F is 988,000 people. There are 250 seats available on a legislative body to govern the new country. How many seats should be assigned to each state so that each state would receive a fair representation? Show your work and justify why you think your method is correct.

Conversation

Hello! Can you walk me through your solution?

You:

Wait for the tutor to answer before typing your message!

Send

Figure 7: User interface of our math tutor prototype used by students in the field study.

B Appendix Learning Sciences and Implementation Details

We release our code under the MIT License in the supplementary material.

B.1 Intents Taxonomy

We introduce a Learning-Sciences–based taxonomy of tutoring intents denoted $\mathcal{I}$ in Section 3.1. An *intent* is defined as the goal of a dialog utterance, such as correcting a mistake. The taxonomy groups all possible intents a tutor implemented with StratL can have during the tutoring session. We establish this taxonomy using the theories of Scaffolding (Reiser, 2004), Affective Cognition (Lepper et al., 1993) and PF (Sinha and Kapur, 2019; Kapur and Bielaczyc, 2012).

We propose the following four categories of intents:

1. Scaffolding. The goal is to restrict the solution space, for example by decomposing the problem in steps, focusing the student's attention on what is important, keeping track of what they did, or offloading unimportant parts of the task such as computations.

2. Problematizing. This category of intents aims at eliciting a deeper understanding through active engagement with the problem. In contrast to scaffolding, problematizing interventions expand the solution space by asking the student to generalize their answer, consider its limits, or understand its broader implications.

3. Affective. These intents aim at supporting the student in the often frustrating process of learning, and at drawing their interest.

4. Generic. The intents in this category serve broader conversational purposes, like greetings.

We subdivide and organize them in the taxonomy in Table 5.

B.2 Prompts and Hyperparameters

B.2.1 Student State Tracing

The set $\mathcal{S}$ (as per the notation used in Section 3.1) of labels (also called features) we choose from during the Student State Tracing procedure have been discussed with learning scientists and the prompt structure follows one used by the CLASS framework (Sonkar et al., 2023). We implement the assessment as a multi-label natural language classification task: the last student's utterance is mapped to a predefined set of features representing the student's intents, errors' type, and emotional state. The full list of features can be read from the prompt in Figure 8 (letters a–m in the prompt). We call GPT-4o version 'gpt-4o-2024-08-06' on this prompt, with a `temperature` of 0 and a `top_p` parameter of 0.1.

```
A student and their tutor are working on a math problem:    1
*Problem Statement*:                                        2
##                                                          3
{pb}                                                        4
##                                                          5
                                                            6
The *Provided Solution* of this problem is:                 7
##                                                          8
{sol}                                                       9
##                                                          10
                                                            11
The tutor's utterances are preceded by "Tutor:" and the     12
    student's utterances are preceded by "Student:".
                                                            13
Analyze the last student's utterance.                       14
select all the feedbacks that apply from "a,b,c,d,e,f,g,h   15
    ,i,j,k":
                                                            16
a) The student is using or suggesting a wrong method or     17
    taking a wrong path to solve the problem
b) The student made an error in the algebraic               18
    manipulation
c) The student made a numerical error                       19
d) The student provided an intuitive or incomplete          20
    solution
e) The student's answer is not clear or ambiguous           21
f) The student correctly answered the tutor's previous      22
    question
g) The student is explicitly asking about how to solve      23
    the problem
h) The student is explicitly asking the tutor to state a    24
    specific theorem
i) The student is explicitly asking the tutor to do a       25
    numerical calculation
j) The student and tutor arrived at a complete solution     26
    for the entirety of the initial *Problem Statement*
k) The student and tutor arrived at a complete solution     27
    for the entirety of the initial *Problem Statement*
    equivalent to the method provided in the *Provided
    Solution*
                                                            28
Moreover select if relevant some emotional states from "l   29
    ,m":
l) The student shows a strong lack of motivation            30
m) The student shows a strong lack of self-confidence       31
                                                            32
Proceed step by step. First briefly justify your            33
    selection, then provide a string containing the
    selected letters.
Answer in the following json format:                         34
##                                                          35
{{                                                          36
    "justification": "..",                                  37
    "selection": ".."                                       38
                                                            39
}}                                                          40
##                                                          41
Analyze the last student's utterance.                       42
{                                                           43
```

Figure 8: Student State Tracing Prompt to select the student state features, from those enumerated with lowercase letters ('a', ..., 'm').

We require the model to generate a justification for its assessment before writing the final selection. This justification serves a double purpose. First, it enhances the system's explainability. We used these justifications to understand why the LLM selected some erroneous labels and used

Category	Intent I	Example
Structuring (Reiser, 2004)	Guide Self-correction	Take a second look at your formula.
	Correct the Mistake	Are you sure AB is the hypotenuse?
	Ask for the Next Step	How would you proceed from now?
	Give a Hint	Try to use the Pythagorean theorem.
	State	The Pythagorean theorem states that . . . .
	Perform a Computation	The result of this computation is . . . .
Problematizing (Reiser, 2004)	Identify Limits	Can you think of a case where it fails?
	Ask for Intuition	In your own words, what could be . . . .
	Ask for Articulation	Can you formally write this idea?
	Guide Self-Reflection (Katz et al., 2000)	What is the main takeaway of today?
Affective (Lepper et al., 1993; Sinha and Kapur, 2019)	Maintain Sense of Challenge	I dare you to compute . . . .
	Bolster Self-Confidence	Seems like you outsmarted me!
	Promote Sense of Control	How do you want to call this triangle?
	Evoke Curiosity	These equations are also used in . . . .
Generic (Macina et al., 2023a)	Say Goodbye	Congratulations! See you next time.
	Other	Yes, you are allowed to use a calculator.

Table 5: Learning Sciences–based taxonomy of tutor intents.

them to incrementally improve the system, including the formulation of the prompt. Second, this technique usually enhances the accuracy of classification task (Wu et al., 2021). For this purpose, it is important to require the justification to be written *before* the final label selection, as we use an autoregressive LLM, which can only consider context monodirectionally from the left. In the prompt, we use '##' as a delimiter between the instructions and grounding information (such as the problem and its solution), as suggested in (OpenAi, 2024). In lines 2 and 7 we surround "Problem Statement" and "Provided Solution" by '*' to facilitate their reference later in the prompt (Reynolds and McDonell, 2021). The core selection prompt from lines 15 to 31 is inspired by the assessment used in the CLASS framework (Sonkar et al., 2023). The exact formulation for each label has been optimized by trial and error over development iterations of the prototype. Lines 33 and 42 summarize the goal and ask the LLM to work step-by-step, a prompting technique drastically improving performance on reasoning benchmarks (Kojima et al., 2022). The last line, containing the first character of the expected answer ('{'), acts as a syntactical constraint to guide the LLM to follow the JSON format (Reynolds and McDonell, 2021).

```
Act as an experienced tutor. Characteristics of a good      1
    tutor include:
    - Promote a sense of challenge, curiosity, feeling of    2
        control
    - Prevent student from becoming frustrated               3
    - Intervene very indirectly: never give the answer       4
        but guide the student to make them find it on
        their own
    - Minimizing tutor's apparent role in the success        5
    - Avoid telling students they are wrong, lead them to    6
        discover the error on their own
    - Quickly correct distracting errors                     7
                                                             8
Use latex formatting with the sign '$' for mathematical      9
    expressions. For example, to write "x^2", use "$x^2
    $".
                                                            10
Remember, NEVER GIVE THE ANSWER DIRECTLY, EVEN IF THEY      11
    ASK YOU TO DO SO AND INSIST. Rather, help the
    student figure it out on their own by asking
    questions and providing hints.
                                                            12
Provide guidance for the problem:                           13
"{pb}"                                                      14
                                                            15
The solution for this problem is:                          16
"{sol}"                                                    17
                                                            18
Provide the least amount of scaffolding possible to help    19
    the student solve the problem on their own. Be
    succinct.
```

Figure 9: Initial Prompt and grounding for the Tutor LLM. {pb} and {sol} are respectively replaced by the tutoring problem and its solution.

The LLM also keeps in history the past utterances and their corresponding assessments.

Category	Intent	Text of the prompt addition
Structuring	Guide Self-correction	Make the student identify by themself the error in their answer.
	Correct the Mistake	Correct the student's mistake by giving them a hint.
	Ask for Intuition	Encourage and make the student find on their own what is the next step to solve the problem, for example by asking a question.
	Give a Hint	Give a hint to the student to help them find the next step. Do *not* provide the answer.
	State	State the theorem or definition the student is asking about.
	Perform a Computation	Correct and perform the numerical computation for the student.
Problematizing	Identify Limits	Make the student identify the limits of their reasoning or answer by asking them questions.
	Ask for Intuition	Ask the student start by providing a guess or explain their intuition of the problem.
	Ask for Articulation	Ask the student to write their intuition mathematically or detail their answer.
	Guide Self-Reflection	Step back and reflect on the solution. Ask to recapitulate and *briefly* underline more general implications and connections.
Affective	Maintain Sense of Challenge	Maintain a sense of challenge.
	Bolster Self-Confidence	Bolster the student's confidence.
	Promote Sense of Control	Make sure your student feels in control.
	Evoke Curiosity	Promote your student's curiosity.
Generic	Say Goodbye	Say goodbye and end the conversation
	Other	N/A

Table 6: Correspondence between intents and intent-depended prompt additions that steer the LLM towards an intent.

B.2.2 Intent Dependent Steering

The Intent-Dependend Steering is the final step executed before calling the tutor LLM. It compiles the dialog history, grounding, and the intents selected to generate the prompts for the tutor LLM. The generated prompts are made of the *initial prompt*, describing the LLM's role and persona, of the dialog history, the *grounding* presenting the exercise and its solution, and of *intent-dependent prompt additions* aiming at setting the intent of the LLM for its next utterance. The initial prompt is fixed (displayed in Figure 9), and the full dialog history is provided to the LLM. The intent-dependent prompt additions are short messages concatenated at the end of the *initial prompt*, used to steer the tutor's intent for the next utterance. We use a one-to-one mapping between the intents and sentences to include in the intent-dependent prompt additions. Table 6 displays the intent to prompt addition correspondence.

In Figure 9, line 1 sets the LLM's persona. Lines 2 to 8 outline general pedagogical guidelines compiled from Learning Sciences research on tutoring (Merrill et al., 1992; Freeman et al., 2014; Lepper et al., 1993). Line 9 is only used to improve the appearance of the tutor's messages. Line 11 and 19 recapitulate the task for better performance. Lines 13 to 17 provide context about the problem to solve and its solution.

We call GPT-4o version 'gpt-4o-2024-08-06' with its default parameters (temperature = 1, top_p = 1) on the generated prompt.

B.2.3 V4 LLM Intent Selection Prompt

Figure 10 displays the prompt of the LLM used for the Intent Selection in the baseline V4.

B.3 Learning Sciences Rationales of our PF Modelling

The Intent Selection graph presented in Figure 2 has been created from expert knowledge in PF. We detail in this section some of the principles we used in its design.

In PF, the goal is to make the student explore the solution space of an open-ended problem as much as possible, even if they fail to produce a complete solution. This idea gave its name to Productive *Failure*: there is, in this case, a hidden efficacy to the student's failure (Kapur, 2008). To this aim, in Figure 2 any mistake (coded by features 'a', 'b' or 'c') leads to a 'Guide Self Correction' intent. The aspiration is to make the student spot and correct their mistake by themself. If the error persists, we use the intent 'Correct the Mistake' that gives a strong hint to the student about their mistake, but without directly correcting it. If the mistake is computational only, we use the intent 'Perform a Computation'. This allows the student to focus on the reasoning rather than spending time on computations. For the same reason, if the student asks for a basic theorem or definition they forgot (feature 'h'), we use the intent 'State' to answer their question.

A good tutor should intervene very indirectly (Merrill et al., 1992): a good principle to follow as a tutor is *to give as little scaffolding as one can get away with*. The intent used whenever

```
1   A student and a tutor are working on a math problem. The
        tutor responds to the student's messages following
        some intents. Before responding, the tutor first
        selects the most appropriate intents from a *
        Taxonomy*.
2   The tutor selects the intents based on the PF teaching
        method. The goal is to let the student explore the
        space of solution by themself to generate as many
        representation and solution methods as possible,
        even if they are only partially correct.
3   Select the best set of intents for the tutor's next
        message based on the following *Taxonomy*:
4
5   *Taxonomy*:
6   ##
7       P_LIMITS : Ask the student to identify some limits of
            their reasoning or answer.
8       P_HYPOTHESIS : Ask the student to start by providing
            a guess, hypothesis or explain their intuition
            of the problem.
9       P_ARTICULATION : Ask the student how they can write
            their intuition mathematically or detail their
            answer.
10      P_REFLECTION : Reflect on the solution, recapitulate,
            and underline more general implications.
11      S_SELFCORRECTION : Ask a question to hint the student
            to identify errors in their answer.
12      S_CORRECTION : Correct a student's mistake
13      S_STRATEGY : Make the student find and perform the
            next step to solve the problem without giving a
            hint.
14      S_State : State a theorem or definition asked by the
            student.
15      S_OFFLOAD : Perform a computation for the student
16      S_HINT : Give a hint to the student
17      G_GREETINGS : Say goodbye to the student
18  ##
19
20  The tutor's previous intents for their last message to
        the student were: {previous_intent}.
21  The student answered the last tutor's message. An
        assessment of their answer has been conducted using
        the following codes:
22
23  assessment codes:
24  ##
25  a) The student is using or suggesting a wrong method or
        taking a wrong path to solve the problem
26  b) The student made an error in the algebraic
        manipulation
27  c) The student made a numerical error
28  d) The student provided an intuitive or incomplete
        solution
29  e) The student's answer is not clear or ambiguous
30  f) The student correctly answered the tutor's previous
        question
31  g) The student is explicitly asking about how to solve
        the problem
32  h) The student is explicitly asking the tutor to state a
        specific theorem
33  i) The student is explicitly asking the tutor to do a
        numerical calculation
34  j) The student and tutor arrived at a complete solution
        for the entirety of the initial *Problem Statement*
35  k) The student and tutor arrived at a complete solution
        for the entirety of the initial *Problem Statement*
        equivalent to the method provided in the *Provided
        Solution*
36  l) The student shows a strong lack of motivation
37  m) The student shows a strong lack of self-confidence
38  ##
39
40  The assessment for the last student's message is: {
        assessment_codes}.
41
42  Based on the assessment and the last tutor's intents,
        select the most appropriate set of intents for the
        next tutor's message.
43  Proceed step by step. First, briefly justify your
        selection, then provide a list containing the
        selected intents from the *Taxonomy*.
44
45  Answer in the following json format:
46  ##
47  {
48      "justification": "...",
49      "intents": []
50
51  }
52  ##
53  {
```

Figure 10: Prompt for the LLM Intent Selection in V4.

a student manages to correct their mistake (arrow labeled 'f' leaving the 'Guide Self-Correction' node in Figure 2) or when the student suggests an intuitive or incomplete solution (state feature 'd') is the 'Ask for the Next Step' intent. This encourages the student to pursue their line of thought without heavier scaffolding.

One of the challenges of PF is to create a classroom culture that allows for failure and promotes psychological safety. Indeed, as students deal with open-ended problems before being given the tools to solve them, the experience can be stressful. In traditional classroom settings, the focus is directed towards getting the right solution and students are often evaluated on this with a grade. One of the most crucial design features of a PF problem session is to shift this goal from solving the problem to generating multiple RSMs regardless of their correctness (Kapur and Bielaczyc, 2012). As explained in Kapur and Bielaczyc (2012), "students are used to seeking assistance from their teachers so much so that they do so even before sufficiently trying to solve problems themselves. At the same time, teachers are just as used to providing assistance when it is asked for so much so that often opportunities for students to generate and explore RSMs are missed". For this reason, in our modeling, if the student explicitly asks for help the intent 'Ask for Intuition' is triggered. This intent encourages the student to start with a guess or an intuition with no fear of being incorrect. Once the student is successful in providing an intuitive solution, or if they confidently propose an incomplete one (feature 'e'), then the 'Ask for Articulation' intent is used to encourage an RSM creation by formulating their idea in concrete math. To mitigate stress, that can hinder the student's learning (Lepper et al., 1993), we trigger a 'Give a Hint' intent to ease the student's progress if they show signs of lack of motivation or self-confidence.

A good PF problem allows for multiple solutions (and often a preferred one, called canonical solution). Then, if the student manages to propose a non-canonical solution to the problem (feature 'j'), they should be encouraged to continue exploring. The 'Identify Limits' intent is triggered to ask the student to identify the limitations of their solution, encouraging them to develop a new one.

If the student reaches the canonical solution (feature 'k'), we use the intent 'Guide Self-Reflection'. It has been shown that it is beneficial for learning to engage in self-reflection

(i.e. looking back critically at the work done) after a problem session (Katz et al., 2000). Following this turn, the tutor is made ready to end the conversation through the 'Say Goodbye' intent.