PIVOT: Preference-based Intervention Vectors for Pedagogical Tutor Steering

Anonymous ACL submission

Abstract

LLMs are increasingly used for conversational tutoring, but effective tutoring requires more than correct answers. Tutors must choose when to scaffold reasoning, hint, give feedback, explain, or invite reflection. Existing prompting and training methods improve pedagogical alignment, but lack reliable inference-time control over pedagogical strategies. We introduce PIVOT, an activation-steering framework that learns preference-based intervention vectors online for frozen LLM tutors. PIVOT uses a seven-category tutor-move taxonomy and a generate–label–optimise loop, where a human-validated LLM judge identifies target and confusable non-target moves to construct preference pairs for multi-layer residual-stream steering. Across held-out and out-of-domain tutoring data, PIVOT controls tutor moves while preserving relevance and fluency, and its directions can be scaled, transferred, and composed at inference time. In a user study with 30 teachers, 73.3% of participants preferred steered conversations over neutral baseline interactions using the same prompt, and rated the controls as clear, usable, and pedagogically meaningful.

1 Introduction

Large language models (LLMs) are increasingly used for educational tasks such as feedback generation (Fawzi et al., 2026; Nazaretsky et al., 2024), explanation (Fawzi et al., 2025), and conversational tutoring. Conversational tutoring (Sonkar et al., 2023; Liu et al., 2024; Dinucu-Jianu et al., 2025) is especially promising because it can provide scalable, individualised support to learners (Jurenka et al., 2025; Puech et al., 2025). However, effective tutoring requires more than producing correct answers: tutors must guide students through productive challenge rather than immediate solution delivery (Maurya et al., 2025; Dinucu-Jianu et al., 2025; Bjork et al.). Since LLMs are optimised for assistant-style helpfulness rather than pedagogy (Puech et al., 2025; Sonkar et al., 2024), they often reveal answers too early, provide excessive support, or fail to sustain productive multi-turn tutoring (Zhao et al., 2026; Puech et al., 2025).

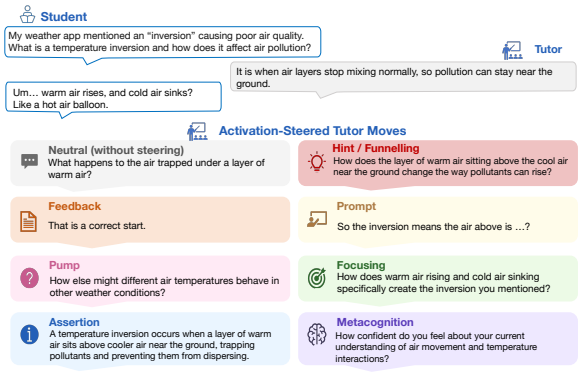


Figure 1: PIVOT steers a frozen LLM toward different pedagogical moves for the same student question.

Consequently, prior work aligns LLM tutors through either prompting or training-based alignment. Prompting methods specify pedagogical strategies in context, including scaffolding, student-state adaptation, and tutoring policies (Sonkar et al., 2023; Park et al., 2024; Chen et al., 2024; Puech et al., 2025). Training-based approaches instead encode pedagogical behaviour through supervised fine-tuning (SFT), preference optimisation, or reinforcement learning (RL) (Jurenka et al., 2025; Macina et al., 2023a; Scarlatos et al., 2025; Rafailov et al., 2023; Dinucu-Jianu et al., 2025). However, both limit controllability differently: prompting is adaptable but fragile under distribution shifts or adversarial requests (Zhao et al., 2026), while training-based alignment produces more stable behaviour at the cost of explicit inference-time control over tutor moves.

Activation steering enables inference-time control over LLM behaviour by injecting steering vectors into residual-stream activations, without modifying prompts or model weights (Rimsky et al.,

2024; Cao et al., 2024; Tan et al., 2024). Prior work derives such vectors from behavioural contrasts (Turner et al., 2024; Rimsky et al., 2024) or preference optimisation (Cao et al., 2024), and recent tutoring work applies them to tutor personas and interaction styles (Lee et al., 2026). However, steering pedagogical tutor moves presents two challenges. First, tutor moves are not naturally binary: behaviours such as feedback, hinting, and direct explanation often overlap, leaving no clear contrastive negative class. Second, tutoring research lacks a unified taxonomy or labelled dataset across tutoring systems, classroom discourse, human tutoring, and LLM tutoring research (Graesser et al., 2004; Alic et al., 2022; Lin et al., 2022; O'Connor and Michaels, 2019; Macina et al., 2023a). Consequently, there is no common supervision framework for learning move-specific steering representations.

To address these challenges, we propose **PIVOT**, an online preference-based steering framework for controllable tutoring. PIVOT derives a seven-category tutor-move taxonomy and learns move-specific residual-stream steering vectors for a frozen LLM through a generate–judge–optimise loop. A human-validated LLM judge labels target and confusable non-target moves, while relevance and fluency judges filter low-quality generations. These judgements form preference pairs, allowing vectors to learn from model confusions without pre-labelled tutoring data. At inference time, steering vectors can be activated, scaled, and combined to control pedagogical behaviour without updating model weights. PIVOT steers the same tutoring context toward different moves while preserving relevance (see Figure 1). We evaluate PIVOT through offline LLM-based evaluation and a user study with 30 teachers. PIVOT reliably controls tutor moves while preserving relevance and fluency, generalises to unseen tutoring data, and transfers to a non-instruction-tuned base model. In our user study, 73.3% of teachers preferred steered conversations over a neutral baseline and rated the controls as pedagogically meaningful and interpretable. Our main contributions are:

- **We propose PIVOT, an inference-time framework for controllable pedagogical steering** that learns multi-layer residual-stream steering vectors through online preference optimisation over the model's own tutor-move confusions.
- **We introduce a seven-category taxonomy of pedagogical tutor moves** and develop a human-validated LLM-based evaluation pipeline for tutor-move, relevance, and fluency assessment.
- **We analyse the controllability and robustness of pedagogical steering**, characterising how steering strength, intervention layers, datasets, and model variants affect tutor-move control, relevance, and fluency.
- **We conduct a user study with 30 teachers** showing that controllable tutor moves are pedagogically interpretable and preferred over neutral tutoring behaviour in a teacher-facing interface.

We provide our implementation at `https://anonymous.4open.science/r/PIVOT`.

2 Related Work

LLM Tutors and Pedagogical Alignment. Effective tutors guide students through questions, hints, feedback, and adaptive support (Lin et al., 2022; Bjork et al.). In contrast, LLMs are optimised for helpful answer generation rather than pedagogy (Tack and Piech, 2022; Macina et al., 2023b), often leading to premature answer disclosure or incorrect feedback (Macina et al., 2023a; Gupta et al., 2025).

Prior work improves pedagogical alignment through prompting or training-based approaches. Prompting methods condition tutoring behaviour on student states, interaction history, or predefined pedagogical structures. For example, ChatTutor updates learning histories through interaction modules and memory (Chen et al., 2024), while StratL and MWPTutor use predefined tutoring policies (Puech et al., 2025; Pal Chowdhury et al., 2024). However, prompting-based systems remain context-sensitive and vulnerable to adversarial requests (Jurenka et al., 2025; Zhao et al., 2026).

Training-based approaches instead encode pedagogy through SFT, preference optimisation, or RL on tutoring interactions. MathDial fine-tunes on semi-synthetic teacher–student dialogues (Macina et al., 2023a), Book2Dial and TutorChat use synthetic textbook-grounded dialogues (Wang et al., 2024; Chevalier et al., 2024), and LearnLM uses mixed human, synthetic, and teacher-authored pedagogical data (Jurenka et al., 2025). Preference- and reward-based methods further optimise tutoring through DPO or RL over simulated interactions (Scarlatos et al., 2025; Dinucu-Jianu et al., 2025). Although effective, these methods require costly

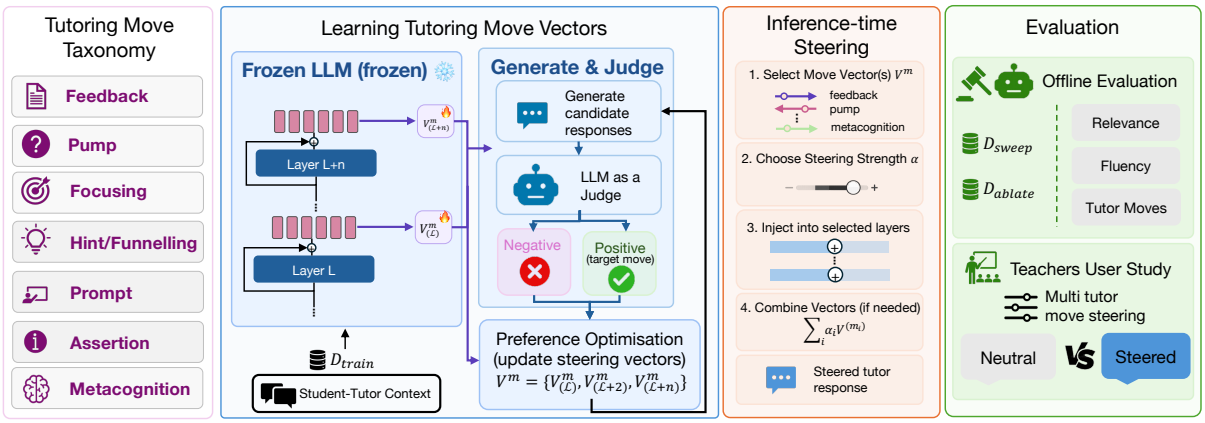


Figure 2: **PIVOT pipeline and evaluation.** PIVOT defines a tutor-move taxonomy, learns move-specific residual-stream steering vectors from rubric-judged generations, and applies them for inference-time control. Evaluation uses offline LLM-judge metrics and a user study with teachers.

training and provide limited inference-time control.

A further challenge is that tutor-move taxonomies are fragmented across tutoring systems, classroom discourse, and human tutoring research. Prior work studies scripted tutoring moves such as hints, prompts, and feedback (Graesser et al., 2004), classroom questioning (Herbel-Eisenmann and Breyfogle, 2005; Hagenah et al., 2018; Alic et al., 2022; O'Connor and Michaels, 2019), and dialogue acts in tutoring dialogues (Lin et al., 2022). However, these taxonomies differ in setting and granularity, providing no unified label space for controllable pedagogical steering.

Activation Steering. Activation steering controls LLM behaviour by intervening on hidden activations at inference time, without modifying prompts or model weights (Turner et al., 2024; Zou et al., 2023; Li et al., 2023; Rimsky et al., 2024; Cao et al., 2024). Early work extracted latent vectors through per-example optimisation (Subramani et al., 2022). Activation Addition instead computes residual-stream directions from paired prompts differing in a target concept (Turner et al., 2024), while Li et al. (2023) steer truthfulness by intervening on activations associated with selected attention heads. Later methods improve robustness through averaged behavioural contrasts (CAA) (Rimsky et al., 2024) and preference-optimised steering vectors (BiPO) (Cao et al., 2024). However, existing methods typically rely on binary contrasts, such as truthful versus untruthful or harmless versus harmful generations. Pedagogical tutor moves are less clearly separable: tutors may scaffold reasoning, provide partial guidance, invite reflection, or directly explain within the same interaction. Moreover, tutoring research lacks unified tutor-move labels and clean target-opposite contrasts, complicating pedagogical steering.

3 Methods

Our goal is to enable inference-time control over LLM tutoring behaviour through steerable pedagogical moves. PIVOT consists of three stages followed by evaluation (Figure 2). First, we define a seven-category tutor-move taxonomy. We then use a frozen LLM and a human-validated LLM judge to learn move-specific residual-stream steering vectors from rubric-grounded positive and negative generations using preference optimisation. At inference time, the vectors can be activated, scaled, and combined. Finally, we evaluate PIVOT both offline and in a user study with teachers.

3.1 Tutoring Move Taxonomy

We derived a seven-category tutoring move taxonomy from prior work on intelligent tutoring, classroom discourse, productive classroom talk, and tutoring-dialogue mining. Table 1 summarises each move, its definition, examples, type, and grounding in prior work. The taxonomy distinguishes scaffolding moves, which support reasoning without revealing the answer, from answer-revealing moves, which provide increasing amounts of solution information. Finally, a reflection move encourages students to reflect on their feelings.

3.2 Learning Tutoring Move Vectors

To control tutoring moves at generation time, we learn move-specific residual-stream steering vectors across selected layers. For a target move m, frozen model M, tutoring context q, and selected

Table 1: Seven-category tutor-move taxonomy for controllable LLM tutoring, grounded in prior work on tutoring dialogue, classroom discourse, productive talk, and tutoring-strategy mining.

Type	Tutor Move	Definition	Example
Scaffolding	Feedback	Evaluates student's answer without adding reasoning or follow-up (Graesser et al., 2004; Lin et al., 2022).	"That's correct." / "You got the right operation."
	Pump	Invites broad continuation or elaboration without narrowing the response space (Graesser et al., 2004; O'Connor and Michaels, 2019).	"Go on." / "What else?" / "Can you say more?"
	Focusing	Asks the student to explain or justify part of their reasoning (Herbel-Eisenmann and Breyfogle, 2005; Franke et al., 2009; Hagenah et al., 2018; O'Connor and Michaels, 2019).	"Why does that step work?" / "Where did you make that choice?"
Answer revealing	Hint/Funnelling	Narrows solution space with a clue, reminder, correction, or guided direction without giving the answer (Graesser et al., 2004; Herbel-Eisenmann and Breyfogle, 2005; Hagenah et al., 2018; Lin et al., 2022; Alic et al., 2022).	"What is the inverse operation here?" / "Look at the coefficient of x."
	Prompt	Nearly completes the step, leaving only a short word, value, symbol, or phrase for the student (Graesser et al., 2004).	"So 12 divided by 4 is...?" / "So x equals...?"
	Assertion	Directly states the answer, fact, procedure, correction, or explanation (Graesser et al., 2004).	"Divide both by 3." / "The answer is $x = 4$."
Reflect	Metacognition	Asks the student to reflect on their understanding, confidence, confusion, strategy, or learning state (O'Connor and Michaels, 2019).	"How confident are you?" / "What made this confusing?"

residual-stream layers $\mathcal{S}$, PIVOT learns a multi-layer vector set $V^{(m)} = \{v_\ell^{(m)}\}_{\ell \in \mathcal{S}}$, where each layer-specific vector $v_\ell^{(m)}$ is added to the residual stream at its corresponding layer ℓ. The selected layer set $\mathcal{S}$ determines where vectors are learned and applied, and can be varied for layer-ablation experiments. As shown in Figure 2, learning follows an iterative generate–judge–optimise loop. Given a student–tutor context, the LLM generates candidate tutor responses with steering applied. A human-validated LLM judge labels each response by tutoring move, while relevance and fluency judges filter low-quality outputs. Target and non-target generations are then converted into preference pairs used to update the steering vectors before the next round.

Generate and Judge. At each round, we sample a tutoring context $q \sim \mathcal{X}$ and generate candidate tutor responses with the current vectors $V^{(m)}$. Intervention strength is controlled by scalars α_{prompt} and α_{gen}, which scale steering at the final prompt token and during autoregressive generation, respectively. For online discovery, we use $\alpha_{\text{prompt}} = 0.1$ and $\alpha_{\text{gen}} = 1.0$, keeping prompt perturbations small while allowing steering directions to shape generation. We use three instruction types: (1) general tutoring instructions, (2) instructions targeting the current move, and (3) instructions targeting under-produced moves. The third type is updated online from judged outputs to maintain move diversity. See Appendix A for all prompts. Each response y is evaluated by human-validated LLM judges (Section 3.4.2): J_{move} assigns move label $\hat{m}$, while J_{rel} and J_{flu} assign binary relevance and fluency labels. Relevant and fluent responses with $\hat{m} = m$ are added to the target pool, and responses with $\hat{m} \neq m$ to the non-target pool.

Preference Optimisation. We first construct online preference tuples (q, y^+, y^-, c) from target and non-target pools, where q is the tutoring context, y^+ is a validated target response, y^- is a rejected non-target response, and $c = J_{\text{move}}(q, y^-)$. Each tuple receives weight $w(q, y^+, y^-, c) = \max(1 - \delta(y^+), \epsilon) w_c$, where $\delta(y^+)$ is an extra rule-based penalisation of malformed, or overly long responses, and w_c is updated online to upweight frequent target confusions. We then optimise the move-specific vectors across selected layers using weighted preference tuples. During optimisation, we set $\alpha_{\text{prompt}}^{\text{train}} = \alpha_{\text{gen}}^{\text{train}} = 1.0$, to learn steering directions under normalised interventions. For each tuple, let

$$\Delta = \log \pi_{V^{(m)}}(y^+ \mid q) - \log \pi_{V^{(m)}}(y^- \mid q) - \log \pi_0(y^+ \mid q) + \log \pi_0(y^- \mid q),$$

where $\pi_{V^{(m)}}$ is the steered model and π_0 is the frozen unsteered model. We optimise

$$\mathcal{L}^{(m)} = \mathbb{E}_{(q,y^+,y^-,c)} \left[-w(q, y^+, y^-, c) \log \sigma(\beta \Delta) \right] + \frac{\lambda}{|\mathcal{S}|} \sum_{\ell \in \mathcal{S}} \|v_\ell^{(m)}\|_2^2.$$

The optimisation updates all steering vectors jointly across layers. Algorithm 1 summarises the procedure.

3.3 Inference-Time Steering

At inference time, PIVOT steers tutor responses by selecting one or more learned tutoring-move vectors, scaling their strength, injecting them into selected residual-stream layers, and combining them when multiple moves are requested. This produces a steered tutor response without updating the LLM.

For a target move m, PIVOT modifies the hidden state at each selected layer $\ell \in \mathcal{S}$ as

$$h'_{\ell,t} = h_{\ell,t} + \gamma_t v_\ell^{(m)},$$

Algorithm 1: PIVOT: Online Preference-Based Steering Vector Learning

Require: Contexts $\mathcal{X}$; target move m; frozen model M; judges J; layers $\mathcal{S}$
Return: Move vector set $V^{(m)}$
Initialise $V^{(m)}$, preference set $\mathcal{D}$, rejected-class weights w_c
for $r = 1, \ldots, T$ **do**
  Generate candidate responses $y \sim M(x; V^{(m)})$ for $x \sim \mathcal{X}$ ▷ `Self-discovery`
  Label candidates with J_{move}, J_{rel}, and J_{flu} ▷ `Judging`
  Construct tuples (x, y^+, y^-, c, w) and add them to $\mathcal{D}$ ▷ `Online preferences`
  Update w_c from rejected move confusions ▷ `Hard-negative weighting`
  Update $V^{(m)}$ by minimizing $\mathcal{L}^{(m)}$ ▷ `Preference-vector update`
end

where $\gamma_t = \alpha_{\text{prompt}}$ for the final prompt token and $\gamma_t = \alpha_{\text{gen}}$ for generated tokens. The prompt coefficient nudges the first generated token, while the generation coefficient is applied autoregressively and provides the main steering signal.

When multiple moves are selected, PIVOT combines their vectors at each layer:

$$\tilde{v}_\ell = \sum_m \alpha_m w_\ell^{(m)} v_\ell^{(m)},$$

where α_m is the move-specific steering strength and $w_\ell^{(m)}$ is an optional layer weight. The combined vector $\tilde{v}_\ell$ is then injected into the selected layers using the same token-dependent schedule.

3.4 Evaluation

We evaluated PIVOT with both offline automatic evaluation and a teacher user study.

3.4.1 Datasets

We evaluated PIVOT on three publicly available tutoring datasets from different domains. *MathDial* contains 2,861 grade-school math tutoring dialogues between human teachers and LLM-simulated students with plausible misconceptions (Macina et al., 2023a). *ConvoLearn* contains 2,134 middle-school Earth Science dialogues based on 7th-grade questions, authored by K-12 teachers interacting with simulated students (Sharma et al., 2026). *StudyChat* contains real student interactions with an LLM chatbot in an undergraduate AI course. We filtered for conceptual and contextual questions, yielding 30,358 turns spanning programming, mathematics, assignment clarification, code explanation, and output interpretation (McNichols et al., 2026).

3.4.2 Offline Evaluation

We evaluated tutor responses with three Qwen3.5-35B-A3B judges: tutor move hit rate, relevance, and fluency (Qwen-Team, 2026). Each judge received the tutoring context, response, and rubric, with prompts provided in Appendix B. The tutor-move judge used the taxonomy in Table 1, the relevance judge assessed whether the response was logically consistent with and relevant to the conversation history (Maurya et al., 2025); and the fluency judge assessed grammar and readability. To validate the judges, two human annotators independently labelled 160 tutor responses across the three datasets. These responses were annotated with their surrounding dialogue context, comprising 327 total student and tutor turns. Annotators labelled each tutor response for move, relevance, and fluency, and we resolved disagreements to refine the rubrics. Human-human agreement exceeded $\kappa = 0.88$ overall and for each tutor move, while human-LLM agreement exceeded $\kappa = 0.89$; see Appendix C for Cohen's κ breakdown.

3.4.3 User Study

To evaluate whether PIVOT produced recognisable and pedagogically meaningful tutor behaviours, we conducted a user study with 30 teachers recruited via Prolific[1], approved by the university ethics commission (Nr. [Anonymous]). Participants had primary or elementary mathematics teaching experience and completed demographic questions before the task (Appendix L). The study focused on mathematics tutoring using three MathDial student scenarios: *stuck or off-track*, *partial answer*, and *passive or unsure*.

After an introduction and practice round, participants used six teacher-facing tutor-move controls to steer a tutoring conversation for up to five turns using 0–100% sliders (100% corresponds to an $\alpha = 1.5$): Broad guidance (*pump*), Narrow guidance (*focusing*), Hint (*hint*), Direct explanation (*assertion*), Feedback (*feedback*), and Reflective question (*metacognition*). We used Qwen3-30B-A3B-Instruct-2507 (Qwen-Team, 2025) as the simulated student agent (prompt in Appendix L). Participants iteratively adjusted controls and regenerated conversations until satisfied, then compared the final

[1] https://www.prolific.com

steered conversation against a neutral baseline generated from the same prompt with all controls disabled. Finally, participants completed a post-task questionnaire (see Appendix L) covering controllability (Masson et al., 2024; Davis et al., 1989), pedagogical quality and teaching-intent alignment (Puech et al., 2025; Maurya et al., 2025), usefulness (Davis et al., 1989), preferred controls, and suggested improvements.

4 Results

We conducted a series of experiments examining the impact of steering strength α, cross-model and cross-dataset vector transfer, and layer selection on the controllability–quality trade-off. We additionally conducted a user study to evaluate the usability and pedagogical relevance of the controls.

Experimental protocol. All experiments used frozen Qwen3.5-9B (Qwen-Team, 2026). Steering vectors were trained across every other layer from 6 to 28, covering early-to-middle through late layers, including the middle-to-late region found effective in prior Qwen steering work (Tan et al., 2024), while using fewer intervention points than all-layer steering. Training used one four-GH200 node, with Qwen3.5-35B-A3B judges on a separate four-GH200 node, and learning all seven move vectors took about five wall-clock hours. We trained on $\mathcal{D}_{\text{train}}$, built from MathDial and ConvoLearn with 2,810 conversations and 42,138 turns, and reserved StudyChat for evaluation only. The main steering-strength set $\mathcal{D}$sweep contained 60 conversations, 20 per dataset, with 452 turns total. For compute-intensive ablations, we used a smaller $\mathcal{D}$ablate with 30 conversations, 10 per dataset, 282 turns, and conversations capped at 10 turns, after verifying that its controllability and quality trends matched $\mathcal{D}_{\text{sweep}}$ (Appendix J).

4.1 PIVOT enables fine-grained steering

We swept α on $\mathcal{D}_{\text{sweep}}$ to assess its effect on tutor-move accuracy, relevance, and fluency. Figure 3 (top) shows target-move hit rates across steering strengths. At $\alpha = 0$, we evaluated the unsteered model using the same tutoring prompt but without steering vectors, sampling five generations per context at temperature 0.8 (prompt in Appendix A). For $\alpha > 0$, steering vectors were added at the selected layers. Hit rates generally increased with α, peaked at moderate strengths, and degraded at larger values, with the best range typically between

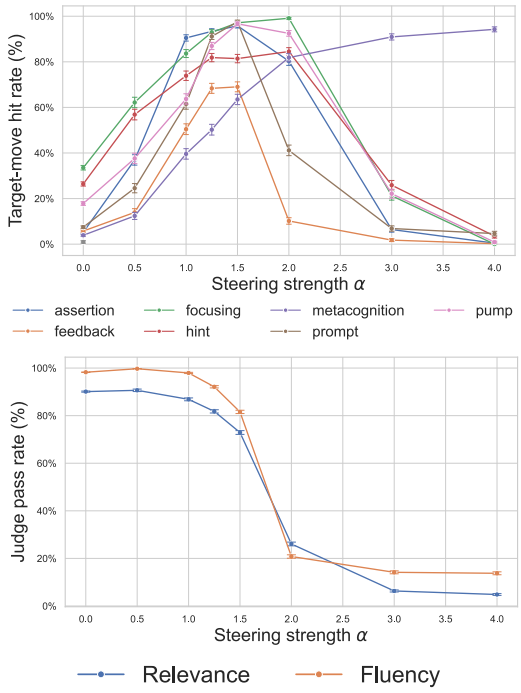


Figure 3: Steering-strength sweeps on $\mathcal{D}_{\text{sweep}}$, showing target-move hit rate (top) and relevance and fluency pass rates (bottom).

$\alpha = 1.0$ and $\alpha = 2.0$. Figure 3 (bottom) shows corresponding relevance and fluency rates, which remained near baseline at moderate strengths but dropped sharply beyond $\alpha = 2.0$. Appendix H compares PIVOT with an offline BiPO-style baseline trained on the same preference pairs and optimisation budget; PIVOT achieves consistently higher tutor-move control.

We also evaluated vector subtraction and additive composition. As a case study, Feedback steering at $\alpha = 1.0$ achieved a 48% hit rate and was most frequently confused with Hint, Assertion, Focusing, and Pump (Appendix F). Subtracting these vectors at $\alpha = -0.3$ increased the Feedback hit rate to 73%. We additionally tested additive move combinations, observing interpretable blended behaviours; for example, Assertion + Feedback produced responses that combined evaluative feedback with direct explanation. See Table 9 (Appendix F) for the detailed results.

PIVOT achieves best control-quality trade-off for α between 1.0 and 2.0, and learned directions can be added or subtracted to shape pedagogical behaviour.

Table 2: Target-move hit rate across datasets $\mathcal{D}_{\text{sweep}}$ at $\alpha = 1.5$.

Dataset	Target-move Hit (%)	Relevance	Fluency
ConvoLearn	88.2 ± 0.9	96.2 ± 0.7	93.8 ± 0.9
MathDial	88.9 ± 1.1	75.0 ± 2.0	87.9 ± 1.5
StudyChat	80.6 ± 1.2	75.2 ± 1.8	86.8 ± 1.4

Table 3: Generalisation to Qwen-9B base on $\mathcal{D}_{\text{ablate}}$.

Move	Target-move Hit (%)	Relevance	Fluency
Assertion	93.1 ± 2.2	90.8 ± 2.6	95.3 ± 1.9
Feedback	85.5 ± 3.1	68.3 ± 4.2	87.0 ± 2.9
Focusing	93.9 ± 2.1	89.8 ± 2.7	95.3 ± 1.9
Hint	64.1 ± 4.2	97.6 ± 1.4	98.5 ± 1.1
Metacognition	99.2 ± 0.8	99.2 ± 0.8	100.0 ± 0.0
Prompt	93.1 ± 2.2	90.3 ± 2.7	96.1 ± 1.7
Pump	93.9 ± 2.1	90.2 ± 2.7	94.6 ± 2.0

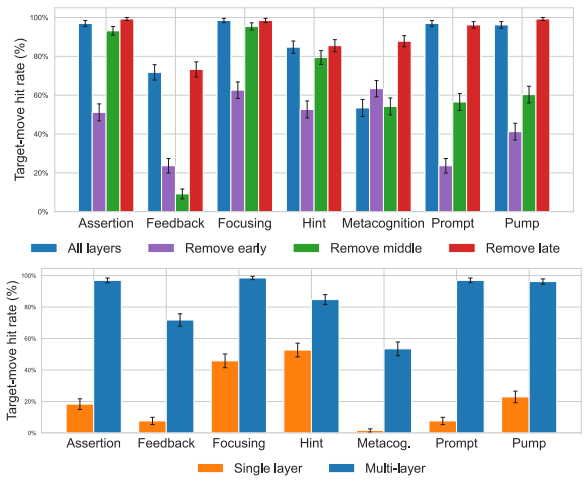


Figure 4: Layer-ablation analysis on $\mathcal{D}_{\text{ablate}}$, showing leave-one-layer-group-out (top) and single-layer steering trained compared with multi-layer steering (bottom).

4.2 PIVOT generalises across datasets and model variants

Next, we investigated whether learned directions generalise across different datasets in $\mathcal{D}_{\text{sweep}}$ and transfer across model variants. Table 2 reports dataset-level results at $\alpha = 1.5$: PIVOT achieved a hit rate of 88.9% on held-out MathDial, 88.2% on held-out ConvoLearn, and 80.6% on StudyChat, which was unseen during training, suggesting transfer beyond the training conversations and to an unseen dataset. We also tested cross-variant transfer by applying Qwen3.5-9B steering vectors directly to Qwen3.5-9B-Base on $\mathcal{D}_{\text{ablate}}$, since the base model shares the same architecture and pretraining, but lacks chat-oriented post-training that can support tutor-like responses. Table 3 reports transfer results using $\alpha = 1.5$ for all moves except Hint ($\alpha = 1.0$) and Metacognition ($\alpha = 2.5$), where those strengths performed better. The results show that learned directions can transfer to a closely related model variant without retraining and are not dependent on chat-oriented post-training.

> PIVOT generalises to unseen data; its learned directions are effective when transferred to a model variant.

4.3 Multi-layer outperforms single-layer

To test whether tutor-move steering benefits from multi-layer intervention, we compared our default setup, using every other residual-stream layer from 6 to 28, with a single-layer baseline at layer 20, a middle-to-late layer where prior Qwen steering work found effective interventions (Tan et al., 2024). Figure 4 (bottom) reports target-move hit rates for both settings. To control for steering power, we matched the single-layer baseline to the multi-layer condition by equating the total perturbation norm of the full vector set at $\alpha = 1.5$, yielding a move-specific equivalent α for layer 20. Specifically, we scaled layer-20 vector by the ratio between the combined norm of the 12 multi-layer edits at $\alpha = 1.5$ and the norm of the layer-20 vector, so that the single-layer intervention had comparable total injected-vector magnitude. Across all tutor moves, multi-layer steering significantly outperformed the single-layer baseline, with paired permutation tests significant after Holm-Bonferroni correction ($p < .001$) for every move. This suggests that reliable tutor-move control benefits from representing each move vector across multiple residual-stream layers rather than using a single layer-specific intervention. To identify which layers within our multi-layer range (6–28) carried the most signal, we performed post-hoc layer-group ablations: after training the full vectors once, we removed early (6, 8, 10, 12), middle (14, 16, 18, 20), or late (22, 24, 26, 28) layers at inference time. Figure 4 (top) compares each removal to the all-layer setting using paired permutation tests with Holm-Bonferroni correction. Removing early layers significantly reduces hit rates for six of seven moves, except Metacognition; removing middle layers selectively hurts Feedback, Prompt, and Pump; and removing late layers does not reduce any move and improves Metacognition. Thus, early layers are most important, middle lay-

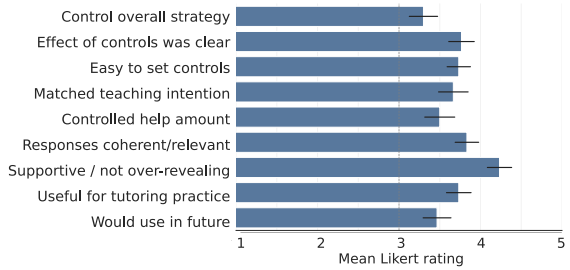


Figure 5: Teacher ratings of the controllable tutoring interface. Ratings are on a 1–5 Likert scale.

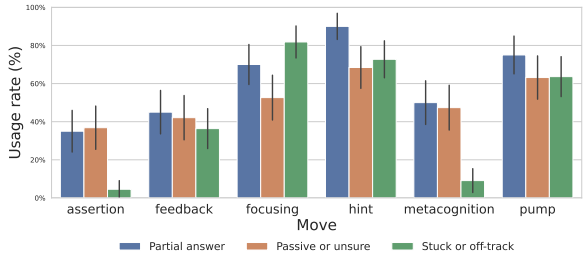


Figure 6: Move usage by instructional scenario. Teachers selected different controls depending on the state.

ers support some moves, and late layers are largely unnecessary, consistent with prior work showing behavioural representations can emerge before final layers and steering effects can peak earlier (Rimsky et al., 2024). Additional isolated-layer experiments are in Appendix D.

> Multi-layer steering significantly outperforms single-layer steering for all tutor moves, with early-to-middle layers contributing most to reliable control.

4.4 Teachers prefer PIVOT-steered responses

In our user study, we tested whether learned steering vectors could guide tutor responses through teacher-facing controls. Teachers rated the controls positively on a 1-5 Likert scale (Figure 5), for producing supportive responses without over-revealing ($mean = 4.23$), coherence and relevance ($mean = 3.83$), and clear control effects ($mean = 3.77$). Ease of use, teaching-intent match, usefulness, and future use were rated positively, suggesting that teachers found the controls interpretable and useful.

Most teachers preferred the controlled conversation over a neutral one generated with the same prompt but no steering: 22/30 participants (73.3%) preferred the controlled setting, compared with 3/30 (10.0%) preferring neutral and 5/30 (16.7%) reporting no preference. Open-ended responses suggest that teachers valued guidance without over-direction: one described the controlled conversation as "more guided whilst allowing the student to use their current knowledge," while another said it "encouraged the student to expand on their thinking." Participants particularly valued Hint and Focusing controls because they nudged the student toward "figuring out" the answer and supported gradual progress without sounding "dismissive."

Finally, teachers used different moves across scenarios (Figure 6). Hint was common throughout, while Focusing was especially frequent for stuck or off-track students. Partial-answer scenarios showed high use of Hint, Pump, and Focusing, suggesting that teachers combined targeted guidance with opportunities for reasoning. Passive or unsure scenarios led to relatively more Assertion, alongside continued Hint and Pump use. These patterns suggest that teachers treated the controls as student-state-responsive pedagogical actions rather than arbitrary generation knobs. Refer to Appendix L for additional move-combination and scenario-specific analyses.

> Teachers preferred steered conversations and found the controls clear, usable, and pedagogically useful.

5 Conclusion

We introduced PIVOT, a framework for inference-time activation steering of pedagogical tutor moves. PIVOT turns tutor behaviours such as feedback, pumping, focusing, hinting, prompting, assertion, and metacognitive reflection into steerable residual-stream directions. It combines a seven-category taxonomy with online judge-driven preference construction, using target and confusable non-target generations to optimise multi-layer steering vectors. Our experiments showed that PIVOT improves tutor-move control while preserving relevance and fluency, transfers to unseen tutoring datasets, and benefits from coordinated interventions across multiple layers. We further showed that the learned directions can be subtracted or added to shape tutor behaviour. Finally, a teacher user study showed that educators can meaningfully use the controls and generally prefer steered conversations. These results suggest that activation steering is a promising mechanism for giving educators finer-grained control over how LLM tutors respond, without retraining model weights or relying solely on prompts.

6 Limitations

PIVOT improves inference-time control over tutor moves, but several limitations remain. First, our evaluation focuses on tutor-response quality and teacher preferences rather than direct student learning outcomes. Although teachers preferred controlled tutor conversations and rated the controls as pedagogically useful, future work should measure how interaction with PIVOT-steered tutors affects student learning, persistence, and performance over time. This could be done in a classroom setting. Second, tutor-move controls are set once for the full conversation, so teachers must manually intervene and adjust the steering if the desired pedagogical move changes over time. Future work should develop a dynamic steering-vector scheduler that adapts tutor moves automatically based on student progress throughout the conversation. Finally, our user study focuses on mathematics tutoring with a simulated student agent, so future work should focus on examining whether the results could fully generalise to real students, longer instructional settings, or other subject areas.

Transfer to Qwen-9B-Base is strong overall, but varies by move, with Hint showing the largest remaining gap. This is expected under cross-variant transfer, since hinting requires a fine balance between useful guidance and answer disclosure. Future work could study move-specific calibration when transferring vectors to other model variants.

7 Ethics Statement and Potential Risks

PIVOT is designed to support controllable LLM tutoring, not to replace human educators. Because tutor responses can affect student understanding, confidence, and problem-solving behaviour, any deployment should include teacher oversight, transparent controls, and safeguards against over-reliance on generated feedback. Our teacher study was approved by the university ethics committee, and participants provided informed consent, were recruited according to the stated eligibility criteria, and were compensated for their time.

The main risks of PIVOT come from misuse or over-interpretation of the controls. Steering vectors can make tutor behaviour more controllable, but they do not guarantee pedagogical correctness, improved learning outcomes, or safety in all contexts. Overly strong or poorly chosen controls may over-direct students, reveal answers too early, or produce responses that appear pedagogically appropriate without actually supporting learning. Our evaluation therefore measures tutor-move control, relevance, fluency, and teacher preferences, but should not be interpreted as evidence of improved student learning without future studies with real learners and learning outcomes.

We release code, prompts, and annotations to support reproducibility, but these resources should be used with their documented limitations. Systems using PIVOT should report the intended educational setting, monitor generated tutor responses, and evaluate possible failure modes before use with students.

8 Acknowledgements

We acknowledge that the use of AI assistants (ChatGPT) was limited to polishing the language of the original paper. It was used for proofreading and refining grammar, spelling, and phrasing.

References

Sterling Alic, Dorottya Demszky, Zid Mancenido, Jing Liu, Heather Hill, and Dan Jurafsky. 2022. Computationally identifying funneling and focusing questions in classroom discourse. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 224–233, Seattle, Washington. Association for Computational Linguistics.

Robert A Bjork and 1 others. Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning.

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. 2024. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yulin Chen, Ning Ding, Hai-Tao Zheng, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2024. Empowering private tutoring by chaining large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, page 354–364, New York, NY, USA. Association for Computing Machinery.

Alexis Chevalier, Jiayi Geng, Alexander Wettig, Howard Chen, Sebastian Mizera, Toni Annala, Max Aragon, Arturo Rodriguez Fanlo, Simon Frieder, Simon Machado, Akshara Prabhakar, Ellie Thieu, Jiachen T. Wang, Zirui Wang, Xindi Wu, Mengzhou Xia, Wenhan Xia, Jiatong Yu, Junjie Zhu, and 3 others. 2024. Language models as science tutors. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings*

of Machine Learning Research, pages 8310–8335. PMLR.

Fred D Davis and 1 others. 1989. Technology acceptance model: Tam. *Al-Suqri, MN, Al-Aufi, AS: Information seeking behavior and technology adoption*, 205(219):5.

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

Fares Fawzi, Seyed Parsa Neshaei, Marta Knezevic, Tanya Nazaretsky, and Tanja Käser. 2026. Refine: Real-world exploration of interactive feedback and student behaviour. *Preprint*, arXiv:2603.29142.

Fares Fawzi, Vinitra Swamy, Dominik Glandorf, Tanya Nazaretsky, and Tanja Käser. 2025. SCRIBE: Structured chain reasoning for interactive behaviour explanations using tool calling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29285–29310, Suzhou, China. Association for Computational Linguistics.

Megan L. Franke, Noreen M. Webb, Angela G. Chan, Marsha Ing, Deanna Freund, and Dan Battey. 2009. Teacher questioning to elicit students' mathematical thinking in elementary school classrooms. *Journal of Teacher Education*, 60(4):380–392.

Arthur C. Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M. Louwerse. 2004. Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, 36(2):180–192.

Adit Gupta, Jennifer Reddig, Tommaso Calò, Daniel Weitekamp, and Christophernbsp;J. MacLellan. 2025. Beyond final answers: Evaluating large language models fornbsp;math tutoring. In *Artificial Intelligence in Education: 26th International Conference, AIED 2025, Palermo, Italy, July 22–26, 2025, Proceedings, Part I*, page 323–337, Berlin, Heidelberg. Springer-Verlag.

Sara Hagenah, Carolyn Colley, and Jessica Thompson. 2018. Funneling versus focusing: When talk, tasks, and tools work together to support students' collective sensemaking. *Science Education International*, 29(4):261–266.

Beth A Herbel-Eisenmann and M Lynn Breyfogle. 2005. Questioning our patterns of questioning. *Mathematics teaching in the middle school*, 10(9):484–489.

Irina Jurenka, Markus Kunesch, Kevin R. McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, Shubham Milind Phal, Katherine Hermann, Daniel Kasenberg, Avishkar Bhoopchand, Ankit Anand, Miruna Pîslar, Stephanie Chan, Lisa Wang, Jennifer She, Parsa Mahmoudieh, Aliya Rysbek, Wei-Jen Ko, Andrea Huber, and 55 others. 2025. Towards responsible development of generative ai for education: An evaluation-driven approach. *Preprint*, arXiv:2407.12687.

Jaewook Lee, Alexander Scarlatos, Simon Woodhead, and Andrew Lan. 2026. Letting tutor personas "speak up" for llms: Learning steering vectors from dialogue via preference optimization. *Preprint*, arXiv:2602.07639.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

Jionghao Lin, Shaveen Singh, Lele Sha, Wei Tan, David Lang, Dragan Gašević, and Guanliang Chen. 2022. Is it a good move? mining effective tutoring strategies from human–human tutorial dialogues. *Future Generation Computer Systems*, 127:194–207.

Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. 2024. Socraticlm: Exploring socratic personalized teaching with large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 85693–85721. Curran Associates, Inc.

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.

Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. 2024. Directgpt: A direct manipulation interface to interact with large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

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), pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.

Hunter McNichols, Fareya Ikram, and Andrew Lan. 2026. The studychat dataset: Analyzing student dialogues with chatgpt in an artificial intelligence course. In *Proceedings of the LAK26: 16th International Learning Analytics and Knowledge Conference*, LAK '26, page 53–63, New York, NY, USA. Association for Computing Machinery.

Tanya Nazaretsky, Paola Mejia-Domenzain, Vinitra Swamy, Jibril Frej, and Tanja Käser. 2024. Ai ornbsp;human? evaluating student feedback perceptions innbsp;higher education. In *Technology Enhanced Learning for Inclusive and Equitable Quality Education: 19th European Conference on Technology Enhanced Learning, EC-TEL 2024, Krems, Austria, September 16–20, 2024, Proceedings, Part I*, page 284–298, Berlin, Heidelberg. Springer-Verlag.

Catherine O'Connor and Sarah Michaels. 2019. Supporting teachers in taking up productive talk moves: The long road to professional learning at scale. *International Journal of Educational Research*, 97:166–175.

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.

Minju Park, Sojung Kim, Seunghyun Lee, Soonwoo Kwon, and Kyuseok Kim. 2024. Empowering personalized learning through a conversation-based tutoring system with student modeling. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '24, New York, NY, USA. Association for Computing Machinery.

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

Qwen-Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Qwen-Team. 2026. Qwen3.5: Towards native multimodal agents.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Alexander Scarlatos, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. 2025. Training llm-based tutors to improve student learning outcomes in dialogues. In *Artificial Intelligence in Education*, pages 251–266, Cham. Springer Nature Switzerland.

Mayank Sharma, Roy Pea, and Hari Subramonyam. 2026. Convolearn: A learning sciences grounded dataset for fine-tuning dialogic ai tutors. *Preprint*, arXiv:2601.08950.

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.

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

Nishant Subramani, Nivedita Suresh, and Matthew Peters. 2022. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, Dublin, Ireland. Association for Computational Linguistics.

AnaÃ¯s 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*, pages 522–529, Durham, United Kingdom. International Educational Data Mining Society.

Daniel Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. 2024. Analysing the generalisation and reliability of steering vectors. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering, 2024. *URL https://arxiv.org/abs/2308.10248*, 2308.

Junling Wang, Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, and Mrinmaya Sachan. 2024. 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.

Jin Zhao, Marta Knežević, and Tanja Käser. 2026. Evaluating answer leakage robustness of llm tutors against adversarial student attacks. *Preprint*, arXiv:2604.18660.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Tutor Generation Prompt

We use the following prompt to generate tutor responses for the move-balanced generation setting. The prompt instructs the tutor model to produce a single concise utterance, follow exactly one tutoring move, and avoid mixing pedagogical strategies within the same response. During preference construction, we append an additional move-bias instruction to the base prompt. We use the target-move instruction to elicit positive candidates and the under-produced-move instruction to encourage generations from moves that are currently under-represented in the preference pool.

Move-Biased Generation Instruction

```
{base_tutor_generation_prompt}

For this one response, USE the move:
    {target_move}
If it truly cannot fit, choose the closest
    single move, but do not blend moves.
No praise, no agreement, no feedback unless
    the chosen move is Feedback.
Do not mention the move label.
```

Under-Produced Move Generation Instruction

```
{base_tutor_generation_prompt}

For this one response, USE the move:
    {under_produced_move}
If it truly cannot fit, choose the closest
    single move, but do not blend moves.
No praise, no agreement, no feedback unless
    the chosen move is Feedback.
Do not mention the move label.
```

Here, `{move_name}` is instantiated as either the target move or an under-produced move during preference construction.

Tutor Generation Prompt

```
You are a tutor. Continue the dialogue with
    exactly one brief, targeted tutor reply.
Choose EXACTLY ONE tutoring move from this
    set: Feedback, Pump, Focusing, Hint,
    Prompt, Assertion, Metacognition, or
    Generic.
Follow the move definitions strictly. Do
    not mix moves in a single utterance.
If the user message explicitly says to USE
    a particular move for this one
    response, you MUST obey that move.
Be concise and critical: do not default to
    agreement, praise, or encouragement
```

```
    fillers.
Avoid praise-y or agreement openers such as
    "I'm so glad", "That's fantastic",
    "Great job", "That's a great start",
    "That sounds right", or "Yes, exactly"
    unless the chosen move is Feedback and
    the whole utterance is only that
    evaluation.
Return only the tutor utterance. Do not
    mention the move name. Do not think.
Only Use ENGLISH. No Other Languages. Only
    English.

BALANCED DEFINITIONS

- Pump: ask an open "what else"
    continuation question that does NOT
    point to a specific missing condition,
    quantity, or subgoal. Do NOT say things
    like "what other information from the
    problem statement...", "what do you
    need to find/calculate...", or "what is
    the next step...". Provide NO hint, NO
    clue, and NO new information. Do not
    evaluate (no praise, no agreement, no
    feedback). Examples:  "What else could
    be going on here?" / "What else do you
    notice?"

- Focusing: question the student's line of
    thinking by directly targeting what
    they said or did. Ask them to
    justify/clarify a specific step they
    already made. Provide NO new
    information. Examples: "How did you
    decide that?" / "Why did you do that
    step?"

- Hint (Hint/Funneling): provide ONE new
    clue or piece of information that helps
    without giving away the answer. The
    student still has a lot to do.
    Examples: "Remember this is only half
    of what you need." / "Compare what
    happens when X is bigger versus
    smaller."

- Prompt: give almost the full
    answer/solution setup and ask for a
    VERY short completion (one
    word/number/short phrase) to finish it.
    Do not ask a broad question. Do not add
    a new clue. Examples: "So the total is
    ... ?" / "That means x = ... ?"

- Assertion: give the full final answer or
    full correction such that NOTHING
    substantive is needed from the student
    anymore. No real follow-up question.
    Examples: "The answer is 24." / "A
    thinner atmosphere traps less heat, so
    it gets colder."

- Feedback: only judge correctness
    (right/wrong/partly right) and stop. No
    hint, no clue, no next-step question.
    Examples: "That's right." / "Not quite."

- Metacognition: ask about the student's
    FEELINGS or learning experience, NOT
    the task (confidence, confusion,
    progress). High-level reflection about
    the course/concepts rather than this
    problem step. Examples: "How confident
    do you feel about this topic?" / "What
    part feels most confusing overall?"

- Generic: generic/unrelated text,
    off-task. Examples: "What's your
    favorite color?" / "How old are you?"
```

B Judge Prompts

We use three LLM-as-a-judge prompts to evaluate the final tutor utterance in each student–tutor conversation. The relevance judge measures whether the response remains grounded in the tutoring context, the fluency judge checks only surface language quality, and the tutor-move judge assigns the response to one of the tutoring-move categories. All judges use the full conversation history but label only the final tutor utterance.

Relevance Judge Prompt

```
You are a strict evaluator of tutor
    response quality.
Your job is to label the FINAL tutor
    utterance for RELEVANCE to the tutoring
    conversation.

You must use the full conversation history
    to decide whether the final tutor
    utterance belongs in this conversation.
    Judge only the final tutor utterance,
    but compare it against the full history.

Relevance means: the final tutor utterance
    stays on the same problem, concept,
    student reasoning, or instructional
    goal as the conversation. It may
    respond to the student's most recent
    turn, or it may appropriately refer
    back to something earlier in the
    conversation to scaffold the student.
    It should be labeled not relevant only
    when it derails to an unrelated topic,
    answers a different problem, or is
    generic/random in a way that ignores
    the tutoring context.

Binary label:
1 = Relevant: on topic for the current
    tutoring conversation, including
    appropriate references to earlier turns.
0 = Not relevant: unrelated, derailed,
    generic in a way that ignores the
    conversation, or responding to a
    different topic/problem.
```

```
Output a JSON object matching the required
    schema, with a short explanation.
```

Fluency Judge Prompt

```
You are a strict evaluator of tutor
    response quality.
Your job is to label the FINAL tutor
    utterance for SURFACE LANGUAGE FLUENCY
    ONLY.

Fluency means only that the final tutor
    utterance is grammatical, readable, and
    linguistically well-formed as a
    standalone piece of text. Check for
    spelling, missing words, broken syntax,
    repeated fragments, excessive
    punctuation, malformed phrases, garbled
    wording, unfinished output, special
    tokens, or oversteering artifacts.

Do not judge logical flow, topical
    relevance, usefulness, correctness,
    teaching quality, or whether the
    sentence follows naturally from the
    conversation. A line can be fluent even
    if it is irrelevant, unhelpful, or
    logically odd in context.

Binary label:
1 = Fluent: grammatical, readable,
    complete, and understandable as text.
0 = Not fluent: ungrammatical, garbled,
    repetitive, unfinished, malformed, full
    of excessive punctuation, special
    tokens, or broken wording.

Examples of not fluent:
- "Can you can you can you what the the"
- "<|im_start|> teacher teacher"
- "Think about the the the because maybe ??"
- An unfinished sentence that cannot be
    understood.

Examples of fluent:
- "What specific evidence from the Jurassic
    Park movies connects dinosaurs to the
    movement of continents?"
- "Can you explain why you chose that step?"
- "That is close, but check the number you
    multiplied by."

Output a JSON object matching the required
    schema, with a short explanation.
```

Tutor-Move Judge Prompt

```
Classify only the final tutor utterance,
    but use the full chat history.
Your first job is to decide whether the
    final tutor utterance adds new
    information relative to the previous
    turns, or only confirms/restates what
    the student already said.

Choose exactly one label and return only
    JSON.

History rule:
- Read the chat history before labeling. Do
    not classify the final utterance in
    isolation.
- If the tutor repeats, paraphrases,
    accepts, or lightly rewords an idea
    already established by the student,
    that is not new information.
- If the tutor gives a new fact, value,
    mechanism, analogy, relation, missing
    operation, or explicit scaffold, that
    is new information.
- If there is a substantive follow-up
    question, classify the function of that
    question/scaffold unless the tutor has
    fully supplied the answer and the
    question is only a check like "do you
    understand?"

Labels:
0 Feedback/Confirmation/Rejection:
  Confirms, rejects, or evaluates the
    student's academic content. Can include
    a short restatement/paraphrase of what
    the student already established. No new
    information and no substantive
    follow-up.

1 Pump:
  On-task broad continuation. The tutor
    asks for broad elaboration, another
    idea, a whole-solution walkthrough, a
    broad causal/impact/consequence
    explanation, a meaning/definition, or
    what led to a content conclusion
    without adding any new clue,
    constraint, criterion, or direction
    that narrows the answer space. It
    shouldn't be a completion, that's a
    prompt. It does not add a new scaffold.
  Important: a broad "what else / what
    other" continuation question is Pump
    ONLY when it does not point to a
    specific missing condition, quantity,
    or subgoal (e.g., "What other piece of
    information from the problem statement
    haven't you used yet?" is usually
    Hint/Funneling because it directs the
    student to search for a missing
    condition).

2 Focusing:
  "Go back to your own step." The tutor
    explicitly points to the student's own
    idea, answer, value, step,
    decomposition, or named concept and
    asks the student to inspect, explain,
    justify, calculate from, or reason from
    it. No new help is added. Do not use
    Focusing for a general next-consequence
    question unless the question clearly
    targets the student's own stated
    mechanism or step. If the tutor anchors
    to one specific concept the student
    already mentioned and asks how it fits,
```

```
    works, or matters in the larger
    process, that is Focusing rather than
    Pump.
  If the tutor asks a pure clarification
    question like "What does X mean?" and
    DOES NOT provide the
    definition/explanation itself, that is
    Focusing when X is something the
    student just used.

3 Hint/Funneling:
  "Here is something new to help." The
    tutor adds a new clue, fact, mechanism,
    relation, analogy, missing operation,
    explicit scaffold, or targeted
    criterion that narrows the answer space
    toward one intended answer or concept,
    while still leaving substantive work to
    the student. If only a single short
    completion remains after the tutor's
    setup, use Prompt instead.

4 Prompt:
  The tutor asks for the next short value,
    word, symbol, or calculation result on
    an already active step-by-step path. In
    Math this includes the next required
    arithmetic result after the relevant
    relation or quantities have been
    stated. In non-math tutoring, this also
    includes asking for a single named fact
    or short completion after the tutor has
    already narrowed the target enough that
    only a one-word or very short phrase
    answer remains. Prompt is about
    response shape: the student should only
    need a very short completion.
  Do not use Prompt for broad "what else /
    what other items / what else do you
    need to find" questions that are not
    asking for a single short completion.

5 Assertion:
  The tutor directly supplies the needed
    final or intermediate answer, value,
    formula, fact, or explanation. No
    substantive extra information is
    required from the student in that turn.
    If the tutor already gave the answer
    and only adds a light check like
    'okay?' or 'does that make sense?',
    still use Assertion.

6 Reflection/Metacognition:
  Learning-process self-report about
    feelings, confidence, confusion,
    challenge, ease, progress, or learning
    experience. If the question is about
    usefulness, applications, what else
    comes to mind, evidence, source,
    reasoning, or how the student reached a
    content conclusion, use Pump or
    Focusing instead.

7 Generic:
  Social, affective, not reflective,
    metacognitive or administrative
    utterance that does not evaluate
```

Table 4: Human–human agreement for tutor-move annotations overall and by dataset.

Dataset	Cohen's κ
Overall	0.929
ConvoLearn	0.906
MathDial	0.939
StudyChat	1.000

```
    academic correctness and does not teach
    the content.

Priority rules:
1. Social/affective closure with no
    academic evaluation -> Generic.
2. Off-task learning-process self-report ->
    Reflection/Metacognition.
3. No substantive follow-up question:
   - Restates/evaluates what student
     already said -> Feedback.
   - Supplies new needed
     content/value/explanation -> Assertion.
4. Substantive follow-up question:
   - If the expected student response is
     only a short completion already set up
     by the tutor, including a single
     factual word or short phrase in
     conceptual tutoring -> Prompt.
   - Tutor added new scaffold before/inside
     the question and the student still must
     reason from it -> Hint/Funneling.
   - Tutor explicitly points only to
     student's own step/idea -> Focusing.
   - Broad on-task elaboration,
     whole-solution reasoning,
     meaning/definition, or
     next-consequence/impact question not
     explicitly tied to the student's own
     reasoning -> Pump.

Dataset contexts: {dataset_contexts}
Question: {question}
Conversation history: {history}
Final tutor utterance: {final_utterance}
```

C Judge Validation

We validated the tutor-move, relevance, and fluency judges against human annotations. For tutor moves, Tables 4 and 5 report human–human agreement overall/by dataset and by category, while Tables 6 and 7 report human–LLM agreement. For relevance and fluency, both human–human and human–LLM agreement reached 100%, suggesting that these binary surface-quality and context-fit judgements were comparatively simple for the LLM judge.

Table 5: Human–human agreement using Cohen's κ by tutor-move category.

Category	Cohen's κ
Feedback	0.930
Pump	0.922
Focusing	0.846
Hint/Funneling	0.916
Prompt	0.956
Assertion	0.982
Metacognition	0.886
Generic	1.000

Table 6: Human–LLM agreement for tutor-move annotations overall and by dataset.

Dataset	Cohen's κ
Overall	0.954
ConvoLearn	1.000
MathDial	1.000
StudyChat	0.906

D Isolated Layer-Group Steering

In addition to the leave-one-layer-group-out ablation in Section 4.3, we trained steering vectors using each layer group in isolation and compared them with the full all-layer setting. Specifically, we trained separate vector sets using only early layers $(6, 8, 10, 12)$, only middle layers $(14, 16, 18, 20)$, or only late layers $(22, 24, 26, 28)$, and evaluated each setting on $\mathcal{D}_{\text{ablate}}$. This experiment tests whether the performance drop in the removal study is due to a layer group being useful on its own, or to its contribution as part of the full multi-layer intervention.

As shown in Figure 7, all-layer steering generally gives the strongest and most consistent control. Early-layer steering remains competitive for several moves, especially Assertion, Focusing, Metacognition, and Pump, while middle-layer steering performs well for Feedback and Metacognition. Late-layer steering performs substantially worse for most moves, suggesting that late residual-stream interventions alone are insufficient for reliable tutor-move control. These results are consistent with the leave-one-group-out ablation: early and middle layers carry most of the useful steering signal, while late layers contribute less to robust move control.

Table 7: Human–LLM agreement using Cohen's κ by tutor-move category.

Category	Cohen's κ
Feedback	1.000
Pump	1.000
Focusing	1.000
Hint/Funneling	0.890
Prompt	1.000
Assertion	0.936
Reflection/Metacognition	1.000
Generic	1.000

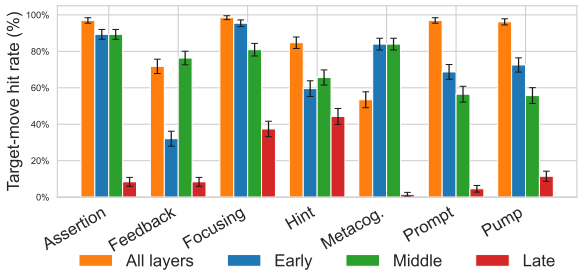


Figure 7: Target-move hit rates when training steering vectors on all selected layers, only early layers, only middle layers, or only late layers on $\mathcal{D}_{\text{ablate}}$.

E Dynamic Weighting Ablation

We ablated the online dynamic weighting used during preference-vector learning by training two variants: one with fixed equal negative-class weights and one with dynamic weights updated from the observed confusion distribution. The dynamic version upweights non-target moves that are frequently confused with the target move, encouraging the optimiser to focus on harder contrasts.

Figure 8 shows that dynamic weighting slightly improves target-move control for most tutor moves. The effect is especially large for Metacognition, where the hit rate increases substantially, and is also visible for Feedback, Hint, and Prompt. Assertion, Focusing, and Pump remain strong in both settings, suggesting that dynamic weighting is most useful for moves with harder or more imbalanced confusions.

F Move Confusions Across Steering Strengths

Figure 9 shows tutor-move confusion matrices on D_{abl} at two steering strengths. At the lower setting, $\alpha = 1.0$, several target moves are still confused with neighbouring pedagogical behaviours. For example, Feedback reaches only a 48% target-

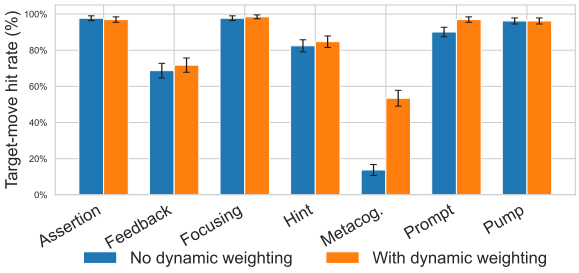


Figure 8: Target-move hit rates with and without dynamic negative-class weighting during steering-vector training.

hit rate and is frequently judged as Assertion or Hint/Funnelling, while Prompt and Pump are also often confused with Focusing or Hint/Funnelling. Increasing the steering strength to $\alpha = 1.5$ substantially sharpens the diagonal of the confusion matrix: Assertion, Focusing, Prompt, and Pump all exceed 95% target-hit rate, and Feedback improves from 48% to 72%. However, some ambiguity remains, especially for Feedback and Metacognition, motivating the subtractive composition experiments in Section G. In particular, the subtraction setting tests whether removing confusable move directions can further improve moves specificity when a target direction overlaps with nearby tutor moves.

G Move Combinations

G.1 Subtraction

At $\alpha = 1$, feedback steering achieved a 48% target-hit rate. At this α level, the steering was not strong enough and the feedback was often confused with nearby tutor moves such as assertion, hinting, pumping, and focusing. After subtracting the main confusable move directions, the feedback hit rate increased to 73%, suggesting that subtraction improves the specificity of the feedback vector while preserving its intended behaviour. Table 8 shows representative examples before and after subtraction.

G.2 Addition

We tested an additive setting in which two tutor-move vectors are applied together at inference time. Because our tutor-move judge assigns exactly one label to each response, a successful combination is reflected when the response is judged as one of the two moves being combined. Table 9 shows qualitative examples where the generated tutor response reflects one of the intended added move directions.

H Comparison to Offline Preference Optimisation

We compare PIVOT with an offline preference-optimisation baseline adapted from BiPO (Cao et al., 2024). This offline baseline learns steering vectors from a fixed set of preference pairs, whereas PIVOT constructs preferences online from the model's own target and confusable non-target generations. For a fair comparison, we trained BiPO-style vectors using the preference pairs generated in all rounds of PIVOT's online training, learned vectors for all tutor moves across layers 6-28, and used the same number of optimisation steps as PIVOT.

Figure 10 shows that PIVOT achieves substantially higher target-move hit rates than the offline BiPO-style baseline across all tutor moves at $\alpha = 1.5$. The gains are especially large for Assertion, Feedback, Prompt, and Metacognition, while Focusing is strong for both methods but still improves under PIVOT. These results suggest that online preference construction is important for tutor-move steering: by repeatedly sampling from the current steered model and upweighting its confusions, PIVOT learns directions that better separate target moves from nearby pedagogical behaviours than vectors trained from a fixed offline preference set.

I Qwen-4B Evaluation

To test whether PIVOT extends to smaller models, we trained Qwen3.5-4B using the same training parameters as the 9B model and evaluated steering at $\alpha = 1.5$. As shown in Table 10, the smaller model achieves strong target-move control for most moves, with especially high hit rates for Focusing (95.4%), Assertion (87.0%), Hint (85.5%), Pump (84.7%), and Feedback (80.2%).

Relevance remains high for Hint and Prompt, but drops for Metacognition and Pump, suggesting that some reflective or open-ended moves are harder to steer reliably in the smaller model. Additional α tuning may be required to improve Metacognition and Pump performance. Fluency is generally strong for Assertion, Feedback, and Hint, although Focusing, Pump, and Metacognition show lower fluency. This indicates a trade-off between move control and response quality for some categories in a smaller model such as Qwen3.5-4B.

Overall, these results suggest that PIVOT remains effective on a smaller 4B model, while also

(a) $\alpha = 1.0$ (b) $\alpha = 1.5$

Figure 9: Tutor-move confusion matrices on D_{ablate} at two steering strengths. Rows correspond to target moves and columns correspond to judge-predicted moves. Increasing α from 1.0 to 1.5 reduces many off-diagonal confusions and improves target-move specificity.

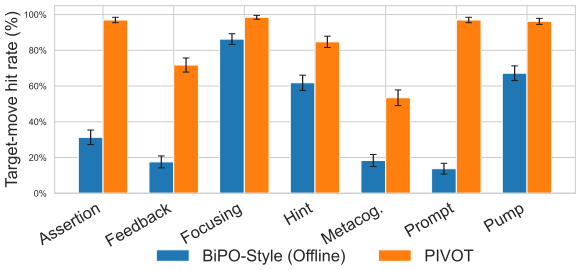


Figure 10: Target-move hit rates for the offline BiPO baseline and PIVOT on $\mathcal{D}_{\text{ablate}}$.

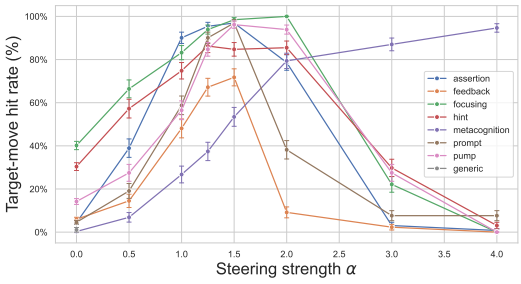


Figure 11: Target-move hit rate across steering strengths on $\mathcal{D}_{\text{ablate}}$.

highlighting which tutor moves are more sensitive to model capacity. Future work could further characterise the behaviour of trained steering vectors in smaller models.

J Ablation Test-set Evaluation

We repeated the steering-strength sweep on $\mathcal{D}_{\text{ablate}}$ to verify that this smaller set follows the same controllability–quality pattern as $\mathcal{D}_{\text{sweep}}$. As shown in Figure 11, target-move hit rates increase from low steering strengths, peak around moderate values, and then degrade at high α. Figure 12 shows the same quality trade-off observed on $\mathcal{D}_{\text{sweep}}$: relevance and fluency remain high at small to moderate strengths but drop sharply once steering becomes too strong. These similar trends justify using $\mathcal{D}_{\text{ablate}}$ for compute-intensive ablations.

K More Implementation Details

Table 11 summarizes the main optimization hyperparameters used to learn the steering vectors. We use a norm penalty and a hard maximum vector norm to prevent the learned vectors from becoming overly large or producing overly aggressive interventions.

L User Study

L.1 Student Agent

We used a simulated student agent to generate short student replies during the tutoring interaction. The prompt instructs the model to remain in the student role, respond briefly, and avoid becoming too capable too quickly.

Feedback		Feedback − Assertion − Pump − Hint − Focusing	
Tutor Response	**Judge**	**Tutor Response**	**Judge**
That is correct. The viscosity depends on the temperature and gas content.	Assertion	That's right.	Feedback
That's a great start. You correctly identified the greenhouse effect. Now, consider how the thick cloud cover interacts with the heat.	Hint/Funneling	That is not quite right.	Feedback
What else could be going on here?	Pump	That's not quite right. You are describing variation, but you haven't identified the specific reason yet.	Feedback
How did you decide where they fit?	Focusing	That's right.	Feedback

Table 8: Examples of feedback-targeted generations before and after subtracting confusable move directions. The examples include both positive and negative feedback corrections.

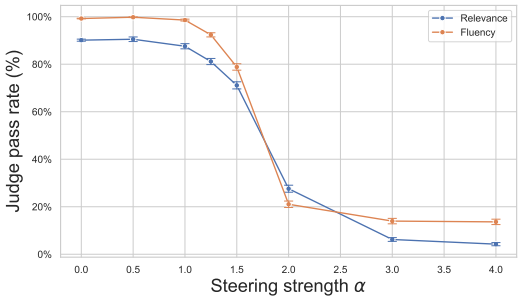


Figure 12: Relevance and fluency pass rates across steering strengths on $\mathcal{D}_{\text{ablate}}$.

Student Agent Prompt

```
You are simulating a math student in a
    tutoring study.
Respond only as the student, in one or two
    sentences.
Do not become too capable too quickly.

Scenario: {scenario_name}
Problem: {problem}
Initial student answer: {initial}

Conversation so far:
{history}

Student reply:
```

L.2 Participants Background

Figure 13 summarises the demographics of the 30 teacher participants. Participants covered a range of age groups, with most between 25 and 54 years old, and included 19 female and 11 male participants. Most were employed full-time and held an undergraduate or graduate degree. All participants reported mathematics teaching experience, with many also teaching science, language, art, or computing. Participants primarily taught elementary, primary, and middle-school students, and most had more than 10 years of teaching experience. Familiarity with AI tutoring tools was mostly moderate, while LLM familiarity was more varied. The median completion time was 18 minutes, corresponding to an average hourly rate of £14.86.

L.3 Study Context and Questionnaires

Explanatory Statement and Consent

Dear participant,
Thank you for participating in our study on controllable AI tutoring. We are very grateful for your participation and your invaluable insight. Please read this Explanatory Statement in full before proceeding. If you would like further information regarding any aspect of this project, please contact us using the email address provided below.
We are a group of researchers dedicated to improving education through technology. The goal of this study is to evaluate whether language model tutors can be made more controllable for teachers and tutors. In particular, we are interested in whether teachers can use simple controls to guide an AI tutor toward different tutoring behaviours, such as giving broad guidance, narrowing the student's attention, giving a hint, asking a reflective question, or providing direct explanation.

- All personal information will be kept confidential and anonymised. Only demographic information is recorded, and it will be reported only in aggregate form to prevent identifying any individual participant.

Tutor Moves Added	Tutor Response	Judge Label
Assertion + Feedback	That is correct: shield volcanoes are the least explosive, stratovolcanoes are the most explosive, and cinder cones and lava domes fall in between.	Assertion
	You calculated average speed, but the question asks for the average time, which requires determining Diego's full race time of 5 minutes.	Hint/Funneling
Feedback + Focusing	That is a good summary of the extremes; let's check where cinder cones and lava domes fit by considering their lava viscosity.	Hint/Funneling
	You correctly identified that the question asks for average speed, but your calculation of the total time assumes Diego finished the race in the same time he ran half. Since Diego was tripped, how did you determine the total time for both racers?	Focusing
Feedback + Pump	That is a good hypothesis to test; how do the eruption styles of cinder cones and lava domes compare to those two extremes?	Pump
	You correctly identified that the question asks for average speed, but you misinterpreted the condition regarding Diego finishing the race. Remember, the problem states he "can't finish," so you need to calculate his time as if he had completed the full distance at that same speed.	Hint/Funneling
Focusing + Hint	You placed cinder cones and lava domes in the middle of explosiveness; what specific property of their lava or gas content leads you to that ranking?	Focusing
	You correctly identified that the problem asks for average speed, but did you account for Diego's full race time when calculating the total time used in your division?	Hint/Funneling
Hint + Assertion	Cinder cones typically have moderate explosivity due to gas-rich ash, while lava domes are highly explosive because their extremely thick lava traps gases near the vent.	Assertion
	You need to calculate Diego's *projected* time to finish the full block based on his speed, not just the time he actually ran.	Hint/Funneling

Table 9: Qualitative examples from additive tutor-move steering. Since the judge assigns one label per response, combined move vectors are reflected when the generated response is classified as one of the two intended moves.

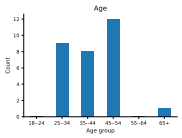
Move	N	Target Hit (%)	Relevance (%)	Fluency (%)
Assertion	132	87.0 ± 2.9	80.2 ± 3.5	91.6 ± 2.4
Feedback	132	80.2 ± 3.5	72.5 ± 3.9	91.6 ± 2.4
Focusing	132	95.4 ± 1.8	80.2 ± 3.5	60.3 ± 4.3
Hint	132	85.5 ± 3.1	90.1 ± 2.6	90.8 ± 2.5
Metacognition	132	61.8 ± 4.3	58.8 ± 4.3	71.0 ± 4.0
Prompt	132	75.6 ± 3.8	87.8 ± 2.9	75.6 ± 3.8
Pump	132	84.7 ± 3.2	64.9 ± 4.2	71.8 ± 3.9

Table 10: Qwen3.5-4B steering results at $\alpha = 1.5$ using the same training setup as the 9B model. Values report percentages with confidence intervals.

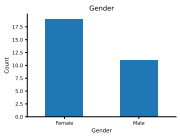
Hyperparameter	Value
Learning rate	4×10^{-4}
Training steps per round	50
Rounds	5
DPO loss β	0.1
Norm penalty	0.008
Max vector norm	6

Table 11: Hyperparameters used for preference-based steering-vector learning.

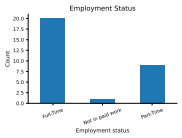
- You may withdraw at any time. Any data you have provided up to that point will be destroyed.
- All data will be collected, stored securely, and reported in accordance with applicable data protection law.
- Only anonymised or aggregated data may be used in future research, subject to ethics approval, and made available to other researchers for further analysis and verification.
- Only the principal investigator and the designated researchers will have access to the original data under strict confidentiality. Results from the project may be published in conference papers and/or journal articles, but no personal data will be shared.
- Personal data will be stored for 5 years from the date of collection. During this period, participants have the right to access their data and inquire about its pro-



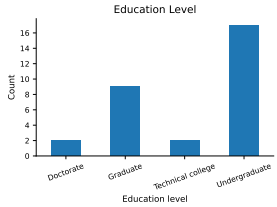
(a) Age



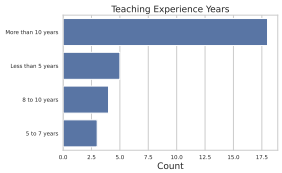
(b) Gender



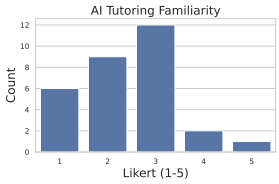
(c) Employment



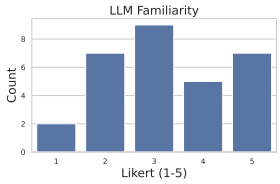
(d) Education level



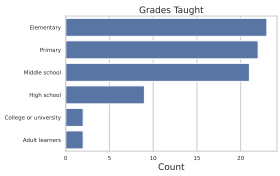
(e) Teaching experience



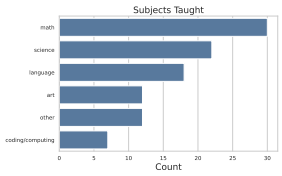
(f) AI tutoring familiarity



(g) LLM familiarity



(h) Grades taught



(i) Subjects taught

Figure 13: Demographics of study participants.

cessing. To exercise this right, contact the Principal Investigator.

By participating in this survey, you agree that your data may be used for scientific purposes.

What You Will Do in This Study

In this study, you will:

- You must have experience teaching elementary/middle school math.
- Learn about simple tutor response types.
- Complete a short practice activity.
- Try move controls that change how the AI tutor responds.
- Use the controls on a student answer.
- Compare your controlled tutor conversation with a neutral AI-tutor conversation.
- Answer a short reflection about whether the controls were useful.

The goal is to understand whether teachers and tutors can use these controls to guide an AI tutor toward the response they want.
The study should take approximately **20 to 30 minutes**. Please complete it carefully and in full. Incomplete or unserious submissions may be discarded.

Study Context

Large language models are increasingly used to support education through tutoring, feedback, and question generation. They can often produce fluent and useful responses, but they are not always easy to control.
The goal of tutoring is not only to give a correct answer, but also to respond in the right way for the situation. For example, a tutor may:

- Ask the student to explain more.
- Give a small hint.
- Focus attention on a mistake.
- Encourage reflection.
- Directly explain an idea.

Normally, AI tutors are guided using written instructions. However, once these instructions are given, the tutor's behaviour is difficult to control. Even with the same instructions, the AI may sometimes:

- Give too much help.
- Be too vague.
- Ask the wrong kind of question.
- Respond in a way that is not helpful for learning.

In this study, we compare two approaches using the same tutoring instructions:

- A standard AI tutor that only follows the written prompt.
- **A controllable AI tutor** where tutoring behaviours can be **adjusted manually** using simple tutor-move controls.

The goal is to understand whether manual control can help teachers guide AI tutor behaviour more effectively.

Pre-task Questionnaire

1. **Do you have experience with elementary/middle school math?**
   - Yes
   - No
2. **Years of teaching or tutoring experience**
   - Less than 5 years
   - 5 to 7 years
   - 8 to 10 years
   - More than 10 years
3. **Grades or levels taught**
   - Primary
   - Elementary
   - Middle school
   - High school
   - College or university
   - Adult learners
   - Other
4. **Subjects taught**
   - Math
   - Language
   - Science
   - Coding/computing
   - Art
   - Other
5. **Familiarity with AI tutoring tools**
   - 1 = Not at all familiar
   - 2 = Slightly familiar
   - 3 = Moderately familiar
   - 4 = Very familiar
   - 5 = Extremely familiar
6. **Familiarity with Large Language Models (LLMs)**
   - 1 = Not at all familiar
   - 2 = Slightly familiar
   - 3 = Moderately familiar
   - 4 = Very familiar
   - 5 = Extremely familiar

Post-task Questionnaire

Step 1: Likert Questionnaire
Prompt: How much do you agree with the following, or to what extent do you agree?
Scale:

- 1 – Strongly disagree
- 2 – Disagree
- 3 – Neither agree nor disagree
- 4 – Agree
- 5 – Strongly agree

Items:

- I felt able to control the tutor's overall response strategy.
- I found the effect of the move controls clear.
- It was easy to set the controls for the tutoring behaviour I wanted.
- I felt the move controls helped me provide the kind of guidance and help I intended for this student.
- I felt the move controls helped me guide the student without giving away the answer too early.
- I found the controlled responses coherent with the student's previous messages.
- I found the controlled responses respectful and supportive, without being dismissive, rude, overly harsh, or unnatural.
- I found this method useful for guiding how the AI tutor helped the student.
- I would use this kind of controllability method in future tutoring practice.

These Likert items are shown in randomised order per participant, and all items are required before continuing.
Step 2: Overall Preference
Prompt: Which conversation did you prefer overall?
Options:

- Neutral conversation
- Controlled conversation
- No preference

Follow-up: Why do you prefer this conversation?
A selection and reason text are required before continuing.
Step 3: Open-Ended Feedback
Prompts:

- Which move control was most useful, and why?
- What would make this method more useful for real tutoring?

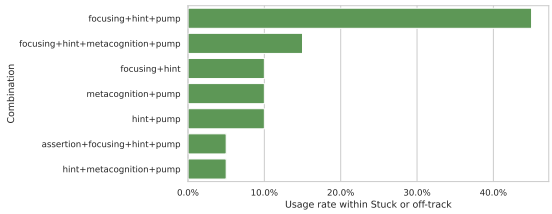
Both responses are required before continuing.
Step 4: Completion
Fields shown:

- Prolific redirect URL
- Completion code
- Submit button

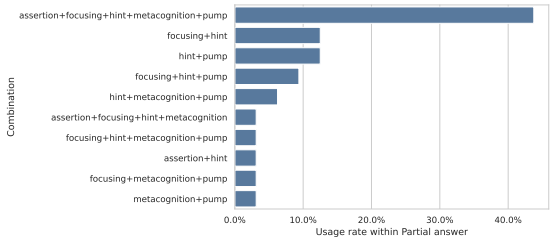
Additional recorded field: `neutral_vs_steered_difference`, from the main-task comparison response.

L.4 Move-combination Usage by Scenario

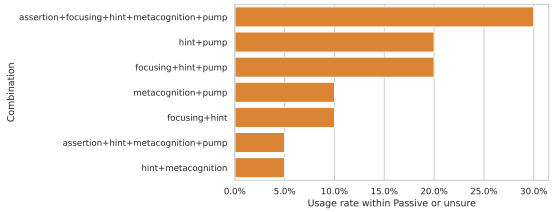
We also analysed how often teachers used different combinations of move controls in each student scenario. Figure 14 shows the most frequent move combinations for the *stuck or off-track*, *partial answer*, and *passive or unsure* scenarios. Across scenarios, teachers frequently combined Hint with Focusing and Pump, suggesting that they used the controls to provide targeted guidance while still leaving room for student reasoning. For stuck or off-track students, the most common combination was Focusing+Hint+Pump, indicating a preference for narrowing attention, giving support, and eliciting further reasoning. For partial-answer and passive or unsure students, teachers more often included broader combinations with Assertion and Metacognition, suggesting that they sometimes combined direct explanation or reflection with guided support when students had already made progress or lacked confidence.



(a) Stuck or off-track



(b) Partial answer



(c) Passive or unsure

Figure 14: Usage rates of teacher-selected move-control combinations by student scenario.