

Letting Tutor Personas Speak Up for LLMs: Learning Steering Vectors from Dialogue via Preference Optimization

Jaewook Lee¹, Alexander Scarlatos¹, Simon Woodhead², Andrew Lan¹

¹University of Massachusetts Amherst, ²Eedi

{ajscarlatos, jaewooklee, andrewlan}@cs.umass.edu, simon.woodhead@eedi.co.uk

Abstract

With the emergence of large language models (LLMs) as a powerful class of generative artificial intelligence (AI), their use in tutoring has become increasingly prominent. Prior works on LLM-based tutoring typically learn a single tutor policy and do not capture the diversity of tutoring styles. In real-world tutor–student interactions, pedagogical intent is realized through adaptive instructional strategies, with tutors varying the level of scaffolding, instructional directiveness, feedback, and affective support in response to learners’ needs. These differences can all impact dialogue dynamics and student engagement. In this paper, we explore how tutor personas embedded in human tutor–student dialogues can be used to guide LLM behavior without relying on explicitly prompted instructions. We train a steering vector using preference optimization: an activation-space direction that guides model responses toward specific tutor personas. We find that this steering vector captures tutor-specific variation across dialogue contexts, improving semantic alignment with ground-truth tutor utterances and increasing preference-based evaluations, while largely preserving lexical similarity. Analysis of the learned scaling coefficients further reveals interpretable structure across tutors, corresponding to consistent differences in tutoring behavior. These results demonstrate that activation steering offers an effective and interpretable way for controlling tutor-specific variation in LLMs using signals derived directly from human dialogue data.

1 Introduction

With the emergence of large language models (LLMs) as a powerful class of generative artificial intelligence (AI), their use in educational tutoring has become increasingly prominent, spanning both general-purpose systems (Google, 2023; OpenAI, 2023) and education-specific tutors (Carnegie Learning, 2024; Khan Academy, 2023). Recent

work has also connected LLMs with Knowledge Tracing (KT), using KT-derived or LLM-assisted student-state estimates for tasks such as student error simulation and mastery modeling in tutoring dialogues (Huang et al., 2026; Duan et al., 2026; Scarlatos et al., 2025a).

Approaches to building LLM tutors vary in how pedagogical behavior is specified and optimized. Many approaches simply prompt LLMs with dialogue policies and guardrails to elicit best-practice pedagogy, including reducing solution leakage, encouraging scaffolding, and enforcing Socratic-style questioning (Pal Chowdhury et al., 2024; Puech et al., 2025; Zhang et al., 2024). Other approaches instead train LLMs using supervised fine-tuning (SFT) to mimic human tutor behavior or use reinforcement learning (RL) on synthetic tutoring dialogues. These approaches optimize proxies of pedagogical quality, such as improving feedback correctness and alignment with tutoring goals (Scarlatos et al., 2024), directly optimizing for pedagogical alignment signals derived from preference data over tutoring behaviors (Sonkar et al., 2024), or leveraging pedagogical rubrics and LLM-based simulated students in-the-loop to encourage greater learning outcomes (Dinucu-Jianu et al., 2025; Scarlatos et al., 2025b).

Despite making improvements on pedagogical quality, these approaches only learn a single tutor policy and do not attempt to capture the diversity of tutoring styles. Figure 1 illustrates how real tutors differ in their instructional style when addressing the same problem. One noticeable difference is the use of emojis and affective language, which conveys emotional support and rapport. Another key difference is instructional approach: the first tutor provides increased emotional support and guides the student step-by-step, whereas the other provides minimal rapport-building and asks the student to test a solution, emphasizing student self-completion. Therefore, we ask a research question:

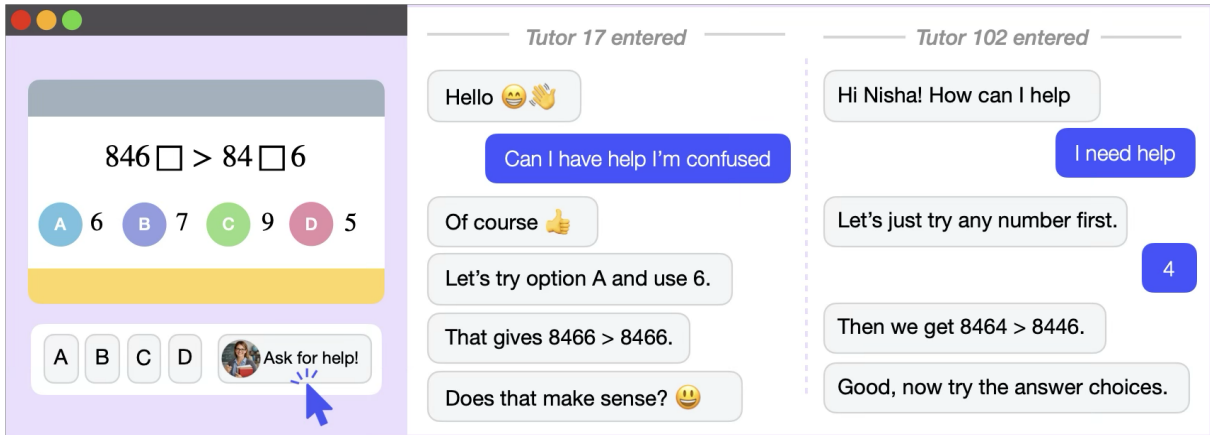


Figure 1: Real math tutoring dialogue on inequalities (Zent et al., 2025), showing how tutors 17 and 102 differ in instructional style when solving the same problem.

“how do we capture differences in tutoring styles and steer LLMs to exhibit these styles?”

1.1 Contributions

In this paper, we explore how implicit tutor personas embedded in human tutor-student dialogues can be used to guide LLM behavior, rather than relying on explicitly prompted instructions. We use preference optimization to learn a steering vector and a scaling coefficient that capture and control the persona direction in the model’s representation space. Using these learned parameters, we then steer the LLM in a tutor-specific manner. We find that our approach enables consistent recovery of tutor-specific steering across dialogue contexts, shifting the model away from population-mean behavior and toward the behavior exhibited by individual tutors. We further show the strength of this effect can be controlled at inference time through a global scaling factor, allowing for smooth adjustment of the intervention. Finally, analysis of the learned scaling coefficients reveals an interpretable structure over tutors that corresponds to variation in tutoring behavior, supporting activation steering as an effective way for capturing and controlling tutor-specific variation.

2 Related Work

2.1 Tutoring Strategies

Effective tutoring relies on a set of practical strategies rather than a single teaching approach, with tutors adjusting what they do in response to the learner’s needs. One core strategy is scaffolding, in which tutors provide support that helps learners solve problems they could not complete on their

own, and then gradually reduce that support as learners become more capable (Wood et al., 1976). Tutors also vary how directly they teach, moving between clear explanation or demonstration and guided practice that helps learners take on more responsibility over time (Rosenshine, 2012). In tutoring conversations, questioning strategies are used to prompt learners to explain their thinking while keeping the interaction focused on the learning goal (Graesser et al., 1995). During early stages of learning, tutors often use worked examples to show how problems can be solved and to make new skills easier to grasp (Sweller and Cooper, 1985). As learners practice, tutors provide formative, error-focused feedback to identify misunderstandings and support improvement (Shute, 2008). Finally, effective tutoring includes affective strategies such as encouragement, building rapport, and managing frustration in order to maintain engagement and persistence (D’Mello and Graesser, 2012).

2.2 Embedding Personas in LLMs

Research on embedding personas into LLMs generally follows two approaches: text-based conditioning and data-driven learning. In the text-based approach, personas are explicitly specified using structured descriptors such as the Five-Factor Model (Big Five) (McCrae and John, 1992), with prior work demonstrating that prompting an LLM with predefined Big Five trait profiles can induce consistent personality-aligned behaviors, which are validated by administering standardized personality questionnaires to the model (Serapio-García et al., 2023). In the context of intelligent tutoring systems, related work adopts explicit persona specifications to construct personality-aware student simulations

that combine cognitive attributes and personality traits, enabling systematic analysis of how different learner personas elicit and evaluate adaptive tutoring strategies (Liu et al., 2024b).

In contrast, data-driven learning induces personas from examples rather than explicit textual descriptions, and is commonly implemented either through SFT or by manipulating representation-level features (Chen et al., 2025; Mazaré et al., 2018). In this paper, we focus on activation steering using steering vectors, which estimate a direction from contrastive behaviors and inject that direction into a model’s hidden states to bias generation toward a target persona. A steering vector can be computed from differences in internal activations produced by paired inputs or outputs that reflect desired versus undesired traits, and then applied at selected layers and token positions during the forward pass to produce consistent behavioral shifts (Liu et al., 2024a; Rimsky et al., 2024; Turner et al., 2023; Wang and Shu, 2024). Our work follows BiPO, which itself builds on Direct Preference Optimization (DPO) by combining preference-based learning to train a steering vector with activation steering to achieve finer-grained control over model behavior (Cao et al., 2024; Rafailov et al., 2023). Like DPO, BiPO uses relative preferences between model outputs rather than absolute likelihood targets, yielding a contrastive learning signal that enables bidirectional steering between preferred and non-preferred behaviors.

3 Methodology

In this section, we detail our approach to learning a steering vector from tutor-student dialogues. We introduce the notation used throughout the paper:

$$\mathcal{D} \triangleq \bigcup_{i=1}^I \{(q_j, d_j^i)\}_{j=1}^{J_i},$$

$$d_j^i \triangleq (s_{j,0}, t_{j,1}^i, s_{j,1}, \dots, t_{j,K_j}^i, s_{j,K_j}).$$

\mathcal{D} denotes a human tutor-student dialogue dataset, where I is the total number of tutors and $i \in \{1, \dots, I\}$ indexes a specific tutor. For each tutor i , J_i denotes the number of dialogues conducted by tutor i , and $j \in \{1, \dots, J_i\}$ indexes an individual dialogue. Each dialogue d_j^i is associated with a math question q_j . Each dialogue d_j^i consists of an alternating sequence of student turns $s_{j,k}$ and tutor turns $t_{j,k}^i$. $k \in \{1, \dots, K_j\}$ indexes (student, tutor) turn pairs in the dialogue, where K_j is the

total number of such turn pairs. The initial student turn $s_{j,0}$ is included when the student initiates the dialogue. We refer to the textual content of each turn as an *utterance*.

A steering vector encodes a direction in an LLM’s activation space, and when added to the activations induces changes in the LLM’s behavior. Learning a tutor-specific vector requires making explicit what the model is being steered away from and toward. In our setting, we define this direction relative to an *average* tutor: for a given dialogue context, the steering vector should shift the model from a population-mean tutor utterance toward the utterance produced by a particular tutor. We first SFT the LLM to condition on the dialogue context and generate an appropriate tutor response. The adaptation ensures that the model can perform the tutoring-response generation task before we estimate tutor-specific directions in activation space. We then use the fine-tuned model to generate a population-mean tutor utterance for each dialogue context and pair this generated utterance with the corresponding ground-truth tutor utterance. Finally, we learn a steering vector from these paired examples, so that the resulting vector captures tutor-specific stylistic variation beyond the population-average tutoring behavior. See Appendix Algorithm 1 for overall procedure to learn the steering vector.

3.1 Constructing Pairs

We use SFT to fine-tune an LLM to serve as a proxy for population-mean behavior, reducing response diversity and encouraging convergence toward a generic average-tutor behavior (Li et al., 2025); the resulting model is used to generate population-mean utterances. Let π_θ denote an LLM parameterized by θ ; we train π_θ via SFT by minimizing the negative log-likelihood of tutor utterances in \mathcal{D} conditioned on the associated question q_j and the preceding dialogue context $(s_{j,\leq k}, t_{j,<k}^i)$:

$$\frac{1}{I} \sum_{i=1}^I \frac{1}{J_i} \sum_{j=1}^{J_i} \frac{1}{K_j} \sum_{k=1}^{K_j} -\log \pi_\theta(t_{j,k}^i | s_{j,\leq k}, t_{j,<k}^i, q_j).$$

For the preceding dialogue context, we construct a pair in which the ground-truth tutor utterance is the preferred response and a response generated by the SFT model is the dispreferred response. The SFT-generated response serves as a proxy for population-mean tutor behavior, so each pair explicitly defines the direction from the average tutor

response $\bar{t}_{j,k}$ toward the tutor-specific response $t_{j,k}^i$. See Appendix Tables 5 and 6 for examples of such contrastive pairs.

3.2 Learning a Steering Vector

We now detail what a steering vector is and how we learn it using the pairs. We introduce a steering vector v , shared across tutors, which we use to modify the model’s activation during inference by adding it to the latent activations $A_L(\cdot)$ at layer L . We assign each tutor a positive scaling coefficient δ_i that scales a shared steering direction v . We constrain these coefficients to have unit mean across tutors to eliminate a scale ambiguity, since the intervention depends only on the product $\delta_i v$ and could otherwise be left unchanged by scaling δ_i up while scaling v down (or vice versa) without affecting the loss. To parameterize δ_i while enforcing positivity, we introduce a tutor-specific learnable scalar u_i and define $\delta_i = \exp(u_i) / (\frac{1}{T} \sum_{m=1}^T \exp(u_m))$. We jointly optimize each u_i and the shared steering direction v in training. We initialize $u_i = 0$ for all tutors ($\delta_i = 1$), and initialize v from a standard Gaussian scaled by 0.5 to ensure a sufficiently strong initial steering magnitude. We note that while we learn the steering vector through preference optimization inspired by BiPO (Cao et al., 2024), we differ in that, rather than using bidirectional steering, we learn a scaling coefficient that scales a shared steering direction:

$$\min_{v,u} \mathbb{E}_{i,j,k} \left[-\log \sigma \left(\beta \log \frac{\pi_{\theta}(t_{j,k}^i | A_L(\cdot) + \delta_i v)}{\pi_{\theta}(t_{j,k}^i | A_L(\cdot))} \right. \right. \\ \left. \left. - \beta \log \frac{\pi_{\theta}(\bar{t}_{j,k} | A_L(\cdot) + \delta_i v)}{\pi_{\theta}(\bar{t}_{j,k} | A_L(\cdot))} \right) \right].$$

For tutor i , dialogue j , turn k , our objective includes two relative preference terms. The first log term encourages the steered LLM to have higher likelihood in generating actual tutor utterance $t_{j,k}^i$ compared to the unsteered SFT model. The right term does the exact opposite, encouraging the steered model to have lower likelihood in generating the population-average tutor utterance. All likelihoods in these terms are sequence log-likelihoods computed by summing next-token log-probabilities over the response tokens only, with prompt tokens masked out. The logistic loss encourages learning a shared steering direction v and tutor-specific coefficients δ_i such that applying $\delta_i v$ consistently favors tutor-specific outputs across tutors, dialogues, and turns. The hyperparameter β encourages similarity

to the base model’s behavior without the steering vector applied. We optimize this objective by iterating for T steps until the loss converges.

At test time, we introduce a global scaling factor α and apply activation steering as $A_L(\cdot) \leftarrow A_L(\cdot) + \alpha \delta_i v$. We observe that $\alpha = 1$ leads to longer utterances, because of the objective uses unnormalized sequence log-likelihoods, so the log-ratio accumulates over all response tokens. Consequently, longer tutor utterances can contribute larger preference margins than shorter ones, making the learned intervention at $\alpha = 1$ overly strong on continuing generation rather than stopping. While the training objective identifies a shared steering direction v and relative tutor-specific strengths δ_i with respect to the unsteered SFT model, a separate scaling factor is necessary to control the overall strength of the intervention during inference.

4 Experimental Settings

4.1 Dataset

We use the Question-Anchored-Tutoring-Dialogues-2k dataset (Zent et al., 2025), containing dialogues from web-based conversations between expert tutors and real students in an online math learning platform. The conversations are initiated by tutors or students requesting assistance while working through a lesson. We split the dataset into training, validation, and test sets at the dialogue level using an 80/10/10 split for each of the 21 unique tutors. On average, each tutor has 74.19 ($\sigma = 75.47$) dialogues in the training set, 8.90 ($\sigma = 9.45$) dialogues in the validation set, and 10.10 ($\sigma = 9.42$) dialogues in the test set. The average number of turns per dialogue is 11.75 ($\sigma = 1.10$) in training, 12.06 ($\sigma = 2.43$) in validation, and 12.25 ($\sigma = 1.73$) in test.

4.2 Evaluation Setup and Metrics

We evaluate how closely the steered tutor utterances match the corresponding ground-truth on a per-tutor basis, along two main axes: *lexical similarity* and *semantic similarity*. For each tutor i , we compare steered outputs $t_j^{\delta_i}$ to ground-truth utterances t_j across turns j . For lexical similarity, we measure surface-form overlap using ROUGE-L (R) (Lin, 2004) and BLEU(B) (Papineni et al., 2002). For semantic similarity, we assess semantic alignment using cosine similarity in embedding space (C) and a pairwise preference-based win rate

(W). We define $C(\delta_i) = \mathbb{E}_j [\cos(\phi(t_j^{\delta_i}), \phi(t_j))]$, where $\phi(\cdot)$ denotes a sentence embedding function from a pretrained SentenceBERT encoder (Reimers and Gurevych, 2019). For Win Rate, we use LLM-as-a-judge with Prometheus-Eval, given a dialogue and a ground-truth turn and chooses which of the steered or unsteered utterances more closely aligns with the tutoring style and language used (Kim et al., 2024). We report the win rate over unsteered SFT utterances, where 0.5 indicates the two perform equally. We do not report ties because they are rare. All metrics are computed per tutor and then averaged across tutors.

4.3 Model Parameters

We use Llama-3.1-8B-Instruct as the base model for SFT, which we adapt using LoRA (Hu et al., 2022) with a learning rate of 5×10^{-5} over 3 epochs, setting the LoRA rank to $r = 32$ and the scaling factor to $\alpha = 64$. Our SFT setup follows Scarlatos et al. (2026), adopting their prompts and training procedure. For learning v and u_i , we use the validation set, applying each u_i to the dialogues corresponding to tutor i and selecting hyperparameters by tracking validation loss; we set $\beta = 1.0$ and learning rate = 0.01, and apply activation steering at the final transformer layer (layer 32). Text generation uses a temperature of 1.0 and top- p sampling with $p = 0.95$, yielding a final validation loss of 0.543 at $T = 17$. The learned u_i have a mean of 0.053 and a standard deviation of 0.106. The activation steering mechanism follows the persona vector approach (Chen et al., 2025), where the learned steering vector is added to the layer activations at all token positions. All training and evaluation are conducted on a single NVIDIA A40 GPU with 48 GB of VRAM.

5 Experimental Results

5.1 Quantitative Results

We quantitatively evaluate activation steering for tutoring along two dimensions: (1) how well it recovers tutor-specific behavior at different stages of a dialogue, and (2) how this behavior varies with the strength of the steering intervention. Specifically, we first compare steered tutor utterances \hat{t}_j with population-mean utterances \bar{t}_j , using ground-truth tutor utterances t_j as reference, in different stages of the dialogue (Table 1). We then analyze the effect of varying the global scaling factor α on both stylistic alignment and lexical similarity

Table 1: Comparison of population-mean utterances \bar{t}_j and steered tutor utterances \hat{t}_j against the ground-truth tutor utterances t_j across dialogue stages. Metrics are computed per tutor and then averaged across tutors.

Stage	Count	Method	R [†]	B [†]	C [†]	W [†]
early	326	\bar{t}_j	0.285	0.070	0.392	–
		\hat{t}_j	0.206	0.041	0.321	0.571
mid	1971	\bar{t}_j	0.165	0.019	0.385	–
		\hat{t}_j	0.157	0.018	0.426	0.587
late	326	\bar{t}_j	0.157	0.025	0.330	–
		\hat{t}_j	0.121	0.017	0.349	0.564

Table 2: Comparison of varying the strength of the steering vector by applying a scaling factor α in activation steering for all stages: $A_L(\cdot) \leftarrow A_L(\cdot) + \alpha \delta_i v$.

Count	Method	α	R [†]	B [†]	C [†]	W [†]
2623	\hat{t}_j	0.0	0.179	0.026	0.379	–
		0.3	0.181	0.027	0.400	0.536
		0.5	0.187	0.028	0.407	0.539
		0.7	0.176	0.026	0.404	0.562
		1.0	0.159	0.021	0.403	0.582

(Table 2).

We conduct a stage-wise analysis because different parts of a dialogue serve different purposes. Early and late turns are largely greetings, encouragement, and closing remarks, which tend to follow common conventions and thus provide limited signal about a tutor’s instructional style. In contrast, the middle portion of the dialogue contains the core problem-solving interaction, where tutors explain concepts, provide hints, and respond to student errors. These turns are where tutor-specific strategies and language use are most evident. For this reason, we define the *mid stage* as the 10–90% interval of turns within each dialogue and focus our analysis on this region.

Table 1 shows the results on all metrics. Beyond early dialogue stages, steering leads to a clear improvement in semantic alignment with the ground-truth tutor: in the mid stage, cosine similarity increases from 0.385 for \bar{t}_j to 0.426 for \hat{t}_j , and the LLM judge prefers \hat{t}_j over the unsteered baseline in 58.7% of comparisons. At the same time, lexical overlap is only slightly reduced; ROUGE-L decreases from 0.165 to 0.157 and BLEU remains almost unchanged (0.019 to 0.018). This result suggests that steering improves tutor-specific style primarily by altering phrasing and discourse struc-

ture rather than by matching exact word sequences. Only in the early stage, the cosine similarity for \hat{t}_j is worse than \bar{t}_j . The result is likely due to greetings in the early turns being inherently generic, which means that adding the steering vector over-exaggerates the effect of tutor style in these turns. A solution may be dynamically setting α to lower values for turns that are intended to be more generic.

Table 2 examines the effect of steering strength aggregated across all dialogue stages. Moderate values of α (0.3 and 0.5) slightly improve ROUGE-L and BLEU relative to the population-mean baseline, while also increasing cosine similarity. As α increases, Win Rate rises steadily from 0.536 to 0.582, indicating stronger expression of tutor-specific behavior. However, higher values of α also lead to lower ROUGE-L and BLEU scores, reflecting a trade-off in which stronger steering enhances stylistic alignment at the cost of lexical similarity.

Based on these results, we select $\alpha = 0.5$ for our qualitative analysis, since it provides a good balance between improving semantic and stylistic alignment with the target tutor while largely preserving lexical similarity.

5.2 Qualitative Results

5.2.1 Interpreting the steering direction

We visualize the scaling coefficient δ_i in Figure 2. To examine how δ_i correlates with tutoring styles or language use, we put tutors into three groups that each address a common math question in the dataset¹: from the left end to the middle (Tutors 283, 346, and 102), from the middle to the right (Tutors 283 and 1430), and from the middle to the right end (Tutors 19 and 2488). Table 3 provides an overview of the math questions and ground-truth utterances, where tutors with lower δ_i are placed on the left and those with higher δ_i are placed on the right.

Table 3 shows three example math questions (ratios, linear sequences, and order of operations) together with short tutor-student dialogue snippets. The table is organized to make the learned direction δ_i interpretable: tutors are arranged from left to right according to their position in the heatmap ordering (e.g., 283 \rightarrow 346 \rightarrow 102 \rightarrow 1430 \rightarrow 19 \rightarrow 2488). For each question, the “Lower δ_i ” column contains tutors closer to the left side of this ordering, while the “Higher δ_i ” column contains tutors closer to the right side. The blue-highlighted

text marks the target tutor utterance to be predicted. Therefore, snippets show how the same dialogue context elicits different conversational moves and tutor utterances, as we move along the δ_i direction.

Tutor 283 who appears at the left end shows relational and pedagogical investment through explicit encouragement and step-by-step scaffolding. In the ratios example, Tutor 283 uses affective markers and praise (“:-)”, “Yes, great!”) and then connects the student’s correct operation on the first term to the required operation on the second term: “the 3 in the ratio has been divided by 3, so you now need to divide the 2 by 3 as well.” In the linear sequence example, the tutor validates the student (“Great!”), explains the structural role of $4n$, and checks understanding with a concrete substitution prompt (“When $n = 1$, $4n = ?$ ”). These turns prioritize both student confidence and conceptual clarity.

Tutors positioned toward the middle of the δ_i direction, including Tutors 102, 1430, and 19, share a more efficiency-oriented style. Across examples, these tutors rely on short, task-focused prompts and corrective feedback, with reduced affective language and limited elaboration. Tutor 102 narrows the student’s response space through a fill-in-the-blank equation (“ $3 \div ___ = 1$ ”) and provides brief correction (“Almost, try again?”). Tutor 1430 advances the linear sequence problem through a series of concise checkpoints (“Great, now pattern 2?”, “Great, now pattern 3?”) and a minimal continuation of the sequence (“6, 10, 14,”). Tutor 19 similarly emphasizes efficient progression by providing succinct procedural guidance (“we multiply before we add”) and a short conceptual reminder (“Remember the fraction line means divide”), occasionally accompanied by a small positive signal with thumbs up emoji. Together, these tutors demonstrate consistent pedagogical engagement while prioritizing error diagnosis and progress over extended explanation or rapport-building.

At the right end of the δ_i direction, Tutor 2488 closely aligns with a low-investment style, focusing on just getting the task completed. The tutor gives short answers (“It means divide”) and follows up with a confusing justification (“it’s a weird one because you’d normally do division before the addition”), which does not clearly resolve the student’s uncertainty about expression structure. Compared to Tutor 19’s concise but accurate guidance, Tutor 2488’s response prioritizes moving the interaction forward, even at the expense of clarity in their explanation. Taken together, these examples

¹Question IDs: 104637, 91697, and 77628, respectively.

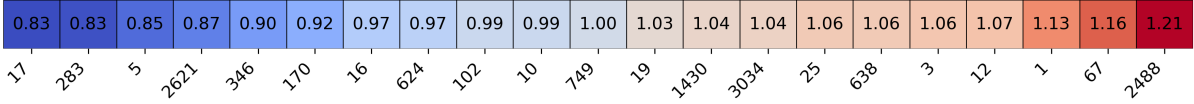


Figure 2: Heatmap of δ_i across 21 tutors with different tutoring styles.

Table 3: Examples of tutor-student dialogue across different math topics. **Blue** denotes our target utterance.

Question	Lower δ_i	Higher δ_i
Ratios <i>Lila and Elijah share some sweets in the ratio 3 : 2. What is this ratio in the form 1 : n?</i> A. 1 : 1.5 B. 1 : $\frac{2}{3}$ C. 1 : 0 D. 1 : 6	Tutor 283 T: :-) So, you need to find an equivalent ratio to 3 : 2. That is, instead 1 : n S: im not sure T: So, what would you divide 3 by, to get the answer 1? S: 3 divide by 3 T: Yes, great! So, the 3 in the ratio has been divided by 3, so you now need to divide the 2 by 3 as well. *well	Tutor 102 T: Ok let's take a look. What do we need to divide by to get the start of the ratio down from 3 to 1? S: i dont know T: $3 \div ___ = 1$ S: 1 T: Almost, try again?
Linear sequence <i>How many dots would there be in Pattern n ? [Image: 6 dots, 10 dots, 14, and 18 dots]</i> A. $4n + 2$ B. $4n - 2$ C. $2n + 4$ D. $n + 2$	Tutor 283 T: Yes of course, let's have a look So, how many dots are being added each time? S: 4? T: Great! So that tells us that the first part of our formula will be $4n$ S: so does the $4n$ refer to how many dots are being added each time? T: Yes, that's how we work out that bit Then, we need to see if that works, or if we need to add/subtract anything When $n=1$, $4n = ?$	Tutor 1430 T: Great, now pattern 2? S: 10 T: Great, now pattern 3? S: 14 T: So what is the pattern going up in? S: 4 T: 6, 10, 14,
Order of operations <i>What is the value of $\frac{1+3 \times 4}{2}$?</i> A. 12.5 B. 8 C. 7 D. 6.5	Tutor 19 T: Yep no worries With a fraction like this we would need to calculate the value of the numerator first What is $1 + 3 \times 4$ S: well 1 add 3 is 4 and 4 times 4 is 16 so will it be $16 <$ T: Not quite. Remember with the order of operations we multiply before we add so we need to do 3×4 first then add 1 S: oh 3 times 4 is 12 add the 1 is 13 T: 👍 So now we have $13/2$ Remember the fraction line means divide	Tutor 2488 S: What do I do after 3×4 T: With the fraction line what does that actually mean? S: I'm not sure what it means T: It means divide S: So 12 divided by 2 T: So it's a weird one because you'd normally do division before the addition but because it's all of the $1+12$ divided by 2

indicate that the learned direction δ_i corresponds to changes in tutoring style, from rapport-rich scaffolding on the left, through diagnostic instruction in the middle, to procedural assistance that emphasizes task completion over teaching on the right.

5.2.2 Case studies

Table 4 shows predicted utterances from unsteered SFT and steered generations with $\alpha = 0.5$. Overall, steering makes the generations better aligned with the ground-truth tutor utterances in Table 3, particularly in how the LLM matches the tutoring approach associated with δ_i . In many cases, the unsteered model produces reasonable tutoring moves, but it often misses the specific dialogue act or tone that distinguishes lower- δ_i tutors (rapport and scaffolding) from higher- δ_i tutors (corrective or short/concise continuations). With $\alpha = 0.5$, the model

more consistently reproduces the intended style difference between the two ends, yielding utterances that are closer to the ground-truth utterances in both tutoring style and language use.

In the ratios example, the ground-truth utterance of Tutor 283 first affirms the student and then explains that whatever is done to the 3 must also be done to the 2. The unsteered prediction (“Amazing!! And 2 divide by 3?”) has a friendly tone, but moves directly to $2/3$ without introducing additional steps. The steered prediction, however, provides a conceptual step while maintaining a friendly tone, thereby exhibiting a tutoring style similar to the ground truth. The ground-truth utterance of Tutor 102 is a brief correction (“Almost, try again?”) following an incorrect student answer. The unsteered prediction instead shifts into an elaborated instructional explanation (“So we need to divide

Table 4: Predicted utterances for the target utterances shown in Table 3.

α	Lower δ_i	Higher δ_i
0.0	Amazing!! And 2 divide by 3?	So we need to divide both sides by 3 $3 \div 3 = 1$ What would the ratio become?
0.5	And what would happen to 2? 😊 $3 \div 3 = 1$ So, we would have to divide both parts by 3 $2 \div 3 = ?$	Nearly, not 1... What do we need to divide by to get from 3 to 1?
0.0	Exactly, so if it's pattern 1 it will have $4n$ dots in the pattern	Great, so to get pattern n , we have $4(n)$ How many is it missing?
0.5	$4n$ refers to the number of times 4 is being added So, $4n$ means the number of terms we're adding 4 to And then, we have to account for the fact that in the first pattern ($n=1$) there are 6 dots, not 4 So, we start with $4n$ and then add 2	So in pattern 1 how many were there and what is this times by??
0.0	So when we have $1 + 3 \times 4$, this is different from $3 \times (1 + 4)$ Are you ok with why that's different?	You have to carry out both operations
0.5	brilliant Now we have $13 / 2$	Yes 👍

both sides by 3 ... What would the ratio become?"), which does not match Tutor 102's concise corrective move. The steered prediction "Nearly, not 1... What do we need to divide by to get from 3 to 1?" is much closer in style: it is explicitly corrective ("Nearly, not 1...") and prompts the student to retry the same subproblem, matching the ground-truth tutor strategy even if the exact wording differs.

In the linear sequence example, there is a similar pattern. The ground-truth utterance of Tutor 283 answers the student's clarification question about what $4n$ represents and then immediately transitions into a structured check ("When $n = 1$, $4n = ?$ "). The unsteered prediction ("Exactly, so if it's pattern 1 it will have $4n$ dots ...") moves the problem forward but does not directly address the student's question, which differs from the tutor's explanatory move. The steered prediction shows an explicitly explanatory turn that attempts to clarify what $4n$ means and then addresses it by referencing the mismatch at $n = 1$, introducing an offset ("we start with $4n$ and then add 2"). While the steered text is more verbose and uses some slightly awkward phrasing, it is closer in tutoring strategy because it responds to the student's conceptual query and uses a concrete consistency check to motivate the adjustment, which is the same pedagogical move that Tutor 283 uses. The ground-truth utterance of Tutor 1430 is extremely concise ("6, 10, 14,"), functioning more like a checkpoint or continuation than an explanation. The unsteered prediction introduces additional scaffolding ("How many is it missing?"), and the steered prediction becomes even more diagnostic ("So in pattern 1 how many were there and what is this times by??"). Neither matches the concise tutoring style in the ground truth, but the steered utterance aligns better in that it remains concise and task-driven.

In the order of operations example, steering yields the clearest improvement. The ground-truth utterance of Tutor 19 ends with a short affirmation with a procedural reminder, "(thumbs up) So now we have $13/2$ Remember the fraction line means divide." The unsteered prediction instead introduces a conceptual aid, contrasting two different expressions ("this is different from $3 \times (1 + 4)$ "), which is pedagogically plausible but does not match the ground-truth target's concise confirmation-and-next-step structure. The steered prediction shows "brilliant Now we have $13 / 2$," which is substantially closer in both tone and function, echoing the short praise and the immediate transition to the next step. The ground-truth utterance of Tutor 2488 is a long, somewhat muddled explanation about why the fraction is "weird," a term rarely used to explain order of operations. Therefore, both predictions struggle, and the steered one incorrectly assesses the student response as correct. The unsteered prediction is overly generic while the steered shows an acknowledgment, saying yes.

Across all examples, steering consistently improves alignment between model-generated utterances and ground-truth ones, capturing the specific tutoring strategy, tone, and style along the δ_i spectrum that unsteered SFT often misses. With $\alpha = 0.5$, steered outputs more reliably reflect the intended tutoring stance and underlying tutoring function, such as rapport-building and scaffolding, affirming before explaining, or concise, corrective continuations, suggesting steering effectively controls pedagogical intent beyond surface style.

6 Conclusions and Future Work

In this paper, we explored how implicit signals in human tutor-student dialogues can be used to guide LLMs towards different tutoring styles. We proposed an activation steering method that learns a shared steering direction and tutor-specific coefficients from real tutor-student dialogues, relative to a population-mean reference, enabling personalized, controlled tutor-specific behavior. We show that the learned steering direction and strengths lead to improved replication of individual tutor behavior. We also qualitatively show that it encodes changing tutoring styles from scaffolded, rapport-building through efficient, task-focused guidance to minimal, task-completion-oriented instruction.

There are several directions for future work. First, although our objective captures tutor-specific behavior via preferences, it does not explicitly encode pedagogical principles. Integrating pedagogical rubrics, learning-theoretic goals, or student outcome-based rewards could better align models with effective tutoring practices. Second, beyond steering tutor behavior, future work could learn complementary steering vectors for student models, enabling simulated tutor-student interactions driven by latent student personas. This would allow analysis of how interaction dynamics and tutor-student compatibility influence learning outcomes. Finally, extending the method to learn multiple steering directions, for example through low-rank subspaces from dimensionality reduction, could enable more fine-grained and disentangled control over distinct pedagogical attributes.

Acknowledgments

This work is partially supported by Renaissance Philanthropy under the learning engineering virtual institute (LEVI) initiative and the NSF under grants 2153481 and 2237676.

Limitations

There are several practical limitations to our work. First, we evaluate our method on a single math dialogue dataset, so it is unclear how well the results generalize to other datasets, domains, or tutoring settings. Second, our baselines are limited to supervised fine-tuning methods; future work should compare against prompting-based approaches as well. Third, we do not include human evaluation, so our findings rely on automatic metrics and model-based judgments.

Ethical Considerations

Our experiments use the Eedi dataset (Zent et al., 2025), in which dialogues are associated with pseudonymous identifiers rather than real names. We therefore do not identify any individual tutors or students in this work. Beyond this, we do not foresee additional ethical concerns specific to our experimental setup.

References

- 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. *Advances in Neural Information Processing Systems*, 37:49519–49551.
- Carnegie Learning. 2024. Livehint overview. Online: <https://support.carnegielearning.com/help-center/math/livehint/article/livehint-overview/>.
- Runjin Chen, Andy Ardit, Henry Sleight, Owain Evans, and Jack Lindsey. 2025. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*.
- 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.
- Zhangqi Duan, Nigel Fernandez, and Andrew Lan. 2026. Kaser: Knowledge-aligned student error simulator for open-ended coding tasks. *arXiv preprint arXiv:2601.06633*.
- Sidney D’Mello and Art Graesser. 2012. Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2):145–157.
- Google. 2023. Gemini. <https://deepmind.google/technologies/gemini/>. Accessed: 2026-01-21.
- Arthur C Graesser, Natalie K Person, and Joseph P Magliano. 1995. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied cognitive psychology*, 9(6):495–522.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Shuyan Huang, Alexander Scarlatos, Jaewook Lee, and Andrew Lan. 2026. Interpretable difficulty-aware knowledge tracing in tutor-student dialogues. *Preprint*, arXiv:2605.01097.

- Khan Academy. 2023. Supercharge your teaching experience with khanmigo. Online: <https://www.khanmigo.ai/>.
- Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353.
- Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. 2025. Preserving diversity in supervised fine-tuning of large language models. In *The Thirteenth International Conference on Learning Representations*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2024a. In-context vectors: making in context learning more effective and controllable through latent space steering. In *Proceedings of the 41st International Conference on Machine Learning, ICML'24*. JMLR.org.
- Zhengyuan Liu, Stella Xin Yin, Geyu Lin, and Nancy Chen. 2024b. Personality-aware student simulation for conversational intelligent tutoring systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 626–642.
- Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. *arXiv preprint arXiv:1809.01984*.
- Robert R McCrae and Oliver P John. 1992. An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215.
- OpenAI. 2023. Chatgpt. <https://chat.openai.com>. Accessed: 2026-01-21.
- 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*, pages 5–15.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- 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.
- 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.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.
- 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.
- Barak Rosenshine. 2012. Principles of instruction: Research-based strategies that all teachers should know. *American educator*, 36(1):12.
- Alexander Scarlatos, Ryan S Baker, and Andrew Lan. 2025a. Exploring knowledge tracing in tutor-student dialogues using llms. In *Proceedings of the 15th international learning analytics and knowledge conference*, pages 249–259.
- Alexander Scarlatos, Jaewook Lee, Simon Woodhead, and Andrew Lan. 2026. Simulated students in tutoring dialogues: Substance or illusion? *arXiv preprint arXiv:2601.04025*.
- Alexander Scarlatos, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. 2025b. Training llm-based tutors to improve student learning outcomes in dialogues. In *International Conference on Artificial Intelligence in Education*, pages 251–266. Springer.
- Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. Improving the validity of automatically generated feedback via reinforcement learning. In *International Conference on Artificial Intelligence in Education*, pages 280–294. Springer.
- Gregory Serapio-García, Mustafa Safdari, Clément Crepy, Luning Sun, Stephen Fitz, Marwa Abdulhai, Aleksandra Faust, and Maja Matarić. 2023. Personality traits in large language models.
- Valerie J Shute. 2008. Focus on formative feedback. *Review of educational research*, 78(1):153–189.
- 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.
- John Sweller and Graham A Cooper. 1985. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and instruction*, 2(1):59–89.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Haoran Wang and Kai Shu. 2024. Trojan activation attack: Red-teaming large language models using steering vectors for safety-alignment. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM '24*, page 2347–2357. Association for Computing Machinery.

David Wood, Jerome S Bruner, and Gail Ross. 1976. The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, 17(2):89–100.

Matthew Zent, Digory Smith, and Simon Woodhead. 2025. PIIvot: A lightweight NLP anonymization framework for question-anchored tutoring dialogues. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27467–27476.

Liang Zhang, Jionghao Lin, Ziyi Kuang, Sheng Xu, and Xiangen Hu. 2024. Spl: a socratic playground for learning powered by large language model. *arXiv preprint arXiv:2406.13919*.

A Appendix

As mentioned earlier, our prompting and supervised fine-tuning (SFT) approach builds on the framework of Scarlatos et al. (2026), using the code available on <https://github.com/umass-ml4ed/sim-student-eval>. In this section, we outline the setup required to train a steering vector.

A.1 Implementation Details

We implement the method in two stages: (i) construction of turn-level preference pairs from tutor-student dialogues, and (ii) optimization of a shared steering direction v together with tutor-specific scaling coefficients $\{\delta_i\}_{i=1}^I$. The implementation follows the formulation in the main text, but includes several practical choices for efficient training and evaluation.

Sequence log-likelihoods. Given a prompt-response pair (q, r) , we compute the conditional sequence log-likelihood

$$\log \pi_{\theta}(r | q)$$

by summing next-token log-probabilities over the response tokens only. Prompt tokens are masked out and do not contribute to the score. These log-likelihoods are computed both for the unsteered

model and for the steered model. Since the baseline values

$$\log \pi_{\theta}(t_{j,k}^i | q_{j,k}^i), \quad \log \pi_{\theta}(\bar{t}_{j,k} | q_{j,k}^i)$$

are reused throughout optimization, they are pre-computed and cached for efficiency.

Parameterization of tutor-specific strength. As in the main text, we learn a shared steering direction v and tutor-specific coefficients δ_i . To enforce positivity and remove scale ambiguity, we optimize unconstrained parameters $u_i \in \mathbb{R}$ and set

$$\delta_i = \frac{\exp(u_i)}{\frac{1}{I} \sum_{m=1}^I \exp(u_m)}.$$

This ensures $\delta_i > 0$ and $\frac{1}{I} \sum_{i=1}^I \delta_i = 1$. In practice, we initialize $u_i = 0$ for all tutors, so that $\delta_i = 1$ initially, and initialize v from a Gaussian distribution.

Training loss. While the main text presents the objective at the turn level, in implementation we group examples by (j, i) , i.e., all tutor turns from dialogue d_j^i associated with tutor i . For each turn k in such a group, we compute the margin

$$m_{j,k}^i = [\log \pi_{\theta}(t_{j,k}^i | C_i) - \log \pi_{\theta}(\bar{t}_{j,k} | C_i)] - [\log \pi_{\theta}(t_{j,k}^i | C) - \log \pi_{\theta}(\bar{t}_{j,k} | C)],$$

where $C_i = A_L(\cdot) + \delta_i v$ and $C = A_L(\cdot)$. The corresponding turn loss is

$$\ell_{j,k}^i = -\log \sigma(\beta m_{j,k}^i).$$

We then average $\ell_{j,k}^i$ across turns k within a given (j, i) group, and average again across groups in the minibatch. Thus, the implementation preserves the same BiPO-style preference structure as the main objective, while stabilizing optimization by aggregating at the dialogue-tutor level.

Steering intervention. During training and inference, the intervention is applied by adding the scaled steering vector to the hidden activations at a designated transformer layer:

$$A_L(\cdot) \leftarrow A_L(\cdot) + \alpha \delta_i v.$$

During training, α is absorbed into the learned parameterization, whereas at test time we expose a global scaling factor α to control overall steering strength independently of the relative tutor-specific coefficients δ_i .

Optimization and model selection. We optimize only the steering parameters $\{v, u_1, \dots, u_I\}$, keeping the underlying language model fixed. Validation loss is computed using the same grouped objective, with cached baseline log-likelihoods for the unsteered model. The best checkpoint is selected according to validation performance, and the learned parameters are then used for test-time generation.

Inference. At test time, for a target tutor i , generation is performed with the intervention $A_L(\cdot) + \alpha \delta_i v$. This yields tutor-conditioned outputs while retaining a single shared steering direction across all tutors. The unsteered SFT model serves as the baseline condition.

Algorithm 1: Learning a steering vector and activating steering using the vector

Input: Dialogue dataset \mathcal{D} , SFT model π_θ , steering layer L , temperature β , training steps T

Output: Shared steering vector v , tutor-specific coefficients $\{\delta_i\}_{i=1}^I$

Pair construction:

foreach tutor i , dialogue d_j^i , tutor turn k **do**

 Construct prompt $q_{j,k}^i$ from the dialogue context

 Set chosen response to $t_{j,k}^i$

 Generate rejected response $\bar{t}_{j,k} \sim \pi_\theta(\cdot | q_{j,k}^i)$

 Store pair $(q_{j,k}^i, t_{j,k}^i, \bar{t}_{j,k})$

Precompute and cache $\log \pi_\theta(t_{j,k}^i | q_{j,k}^i)$ and $\log \pi_\theta(\bar{t}_{j,k} | q_{j,k}^i)$

Initialization:

Initialize $u_i \leftarrow 0$ for all i , and initialize v randomly

Set

$$\delta_i \leftarrow \frac{\exp(u_i)}{\frac{1}{I} \sum_{m=1}^I \exp(u_m)}.$$

Training:

for $t = 1, \dots, T$ **do**

 Group examples by (j, i)

foreach group (j, i) **do**

foreach turn k in d_j^i **do**

 Compute steered log-likelihoods

$$\log \pi_\theta(t_{j,k}^i | A_L(\cdot) + \delta_i v), \quad \log \pi_\theta(\bar{t}_{j,k} | A_L(\cdot) + \delta_i v)$$

 and retrieve cached baseline log-likelihoods

$$\log \pi_\theta(t_{j,k}^i | A_L(\cdot)), \quad \log \pi_\theta(\bar{t}_{j,k} | A_L(\cdot)).$$

 Form margin

$$m_{j,k}^i = \left[\log \pi_\theta(t_{j,k}^i | A_L + \delta_i v) - \log \pi_\theta(\bar{t}_{j,k} | A_L + \delta_i v) \right] - \left[\log \pi_\theta(t_{j,k}^i | A_L) - \log \pi_\theta(\bar{t}_{j,k} | A_L) \right].$$

 Compute turn loss

$$\ell_{j,k}^i = -\log \sigma(\beta m_{j,k}^i).$$

 Average $\ell_{j,k}^i$ over turns k in group (j, i)

 Average loss over groups in the minibatch and update v and $\{u_i\}$

 Recompute $\delta_i \leftarrow \exp(u_i) / (\frac{1}{I} \sum_m \exp(u_m))$

Inference:

for target tutor i **do**

 Apply activation steering

$$A_L(\cdot) \leftarrow A_L(\cdot) + \alpha \delta_i v$$

 during autoregressive generation

Table 5: Examples of our DPO dataset to train a steering vector (Part 1).

ID	Prompt	Chosen	Rejected
12024	<p>[system] Cutting Knowledge Date: December 2023 Today Date: 26 Jul 2024</p> <p>You are a tutor guiding a student through a math problem.</p> <p>[user] Question: Work out the area of the shape [Image: A composite shape made up of a rectangle and a triangle with some side measurements labelled] A. 68 m² B. 34 m² C. 36 m² D. 44 m²</p> <p>(No First Student Turn)</p> <p>[assistant] Hi! How can I help?</p> <p>[user] I dont really understand this question can you please help me with it</p> <p>[assistant] Of course, can you tell me how you started? Or were you just not sure at all?</p> <p>[user] im not sure how i started it</p> <p>[assistant] Ok, well I would start by splitting up the shape into a rectangle and a triangle</p> <p>[user] ok</p> <p>[assistant] Can you tell me the area of that rectangle on the left?</p> <p>[user] 24m squared</p> <p>[assistant]</p>	<p>Fab. Now for the triangle, the sides aren't labelled yet. So if the whole width of the shape is 10m, and the rectangle is 4m, could you work out how wide the triangle is?</p>	<p>Great!</p>

Table 6: Examples of our DPO dataset to train a steering vector (Part 2).

ID	Prompt	Chosen	Rejected
1988	<p>[system] Cutting Knowledge Date: December 2023 Today Date: 26 Jul 2024</p> <p>You are a tutor guiding a student through a math problem.</p> <p>[user] Question: $\frac{1}{2} \times 3 =$ A. $\frac{3}{2}$ B. $3\frac{1}{2}$ C. $\frac{3}{6}$ D. $\frac{1}{6}$</p> <p>First Student Turn: hi</p> <p>[assistant] hello 🙌 how can I help you?</p> <p>[user] all of this</p> <p>[assistant] I'm just reading the question let's think about 3 lots of $\frac{1}{2}$ so first I'm going to think about what I get if I add 3 of them together what is $\frac{1}{2} + \frac{1}{2}$?</p> <p>[user] 1</p> <p>[assistant] great! and then add another $\frac{1}{2}$. so what is $1 + \frac{1}{2}$?</p> <p>[user] 1 and $\frac{1}{2}$</p> <p>[assistant] 👍😊 so the answer to three times $\frac{1}{2}$ must be 1 and $\frac{1}{2}$ oh, look at the options in the question. which is the same as 1 and $\frac{1}{2}$</p> <p>[user] $\frac{1}{6}$</p> <p>[assistant]</p>	<p>no sorry, $\frac{1}{6}$ is much smaller than $\frac{1}{2}$ it's like you share something between 6 people instead of just 2 people. Each person gets less when there's 6 people to share something between instead of just 2 people sharing it have another look at the ABCD options we now know it's not D</p>	<p>A is $\frac{3}{2}$ 😊😞😞</p>