Henry at BEA 2025 Shared Task: Improving AI Tutor's Guidance Evaluation Through Context-Aware Distillation

Pagnarith Pit
University of Melbourne
ppit@student.unimelb.edu.au

Abstract

Effective AI tutoring hinges on guiding learners with the right balance of support. In this work, we introduce **CODE** (**CO**ntextually-aware **D**istilled **E**valuator), a framework that harnesses advanced large language models (i.e., GPT-4o and Claude-2.7) to generate synthetic, context-aware justifications for human-annotated tutor responses in the BEA 2025 Shared Task. By distilling these justifications into a smaller open-source model (i.e, Phi-3.5-mini-instruct) via initial supervised fine-tuning and then Group Relative Policy Optimization, we achieve substantial gains in label prediction over direct prompting of proprietary LLMs. Our experiments show that **CODE** reliably identifies strong positive and negative guidance, but like prior work, struggles to distinguish nuanced "middle-ground" cases where partial hints blur with vagueness. We argue that overcoming this limitation will require the development of explicit, feature-based evaluation metrics that systematically map latent pedagogical qualities to model outputs, enabling more transparent and robust assessment of AI-driven tutoring.

1 Introduction

Large language models (LLMs) have opened a promising frontier for education, enabling conversational agents that deliver personalized and adaptive guidance calibrated to a learner's current knowledge state and pace (Tack et al., 2023). Indeed, the main goal of dialectic teaching is to provoke exploration through carefully timed questions, hints, or explanations (Clark and Egan, 2015). If the guidance provided is too little, it frustrates students, while too much erodes learning opportunities and fosters over-reliance (Le, 2019). Although striking this balance is central to effective tutoring, the field still lacks precise operational definitions and automatic metrics for "optimal guidance", making systematic evaluation, and therefore progress, very challenging (Kochmar et al., 2025).

In light of this missing definition, existing assessments rely heavily on individual annotation by human experts (Maurya et al., 2025). However, crafting high-quality, question-specific explanations at the scale needed to train or benchmark modern transformer models is prohibitively expensive. To address this bottleneck, we explore *reasoning distillation*: using stronger LLMs to generate reasoning about a tutor's utterance as to why it matches the gold label. Our study investigates (i) whether synthetically contexts capture meaningful signals of pedagogical quality, and (ii) how well these signals transfer when smaller, student models are trained on them.

As such, our contributions from team Henry are as follows:

- We propose **CO**ntextually-aware **D**istilled **E**valuator (**CODE**) framework, a multi-step finetuning process that distills reasoning from larger LLMs to train smaller open-sourced models to better detect what "good guidance" is. Our method consistently outperforms state-of-the-art (SOTA) proprietary models and aligns reasonably well with expert human judgements.

- We release an enriched dataset with synthetically generated reasoning based on their gold labels for each of the tutor's last utterance across the entire human-annotated set from (Maurya et al., 2025).

2 Related Work

2.1 AI Tutor's Guidance Evaluation

This feature of AI tutor currently lacks a unified definition, but there has been efforts in this area to explore it through various perspectives. Tack

and Piech (2022) in their work evaluates performances of AI tutor based on how much they "help the student" using human participants and expert annotators. While they don't provide a formal definition, their approach to evaluation is closely resembled by Daheim et al. (2024)'s "actionability" where the AI tutor's utterance provides sufficient information for the student to progress the conversation and move closer to the correct answer.

In another work by Wang et al. (2024), this feature is referred to as "usefulness", the degree to which the responses are productive at advancing the student's understanding and helping them learn from their errors, also evaluated through human judgments. These concepts are also reflected in the work of Al-Hossami et al. (2023), where they defined "indirectness", where an effective tutor asks questions that induce critical thinking and not reveal the answer.

2.2 Learning via Distillation

Knowledge distillation transfers the knowledge embedded in large, high-capacity "teacher" models into smaller, more efficient "student" models by having the student match the teacher's softened probability distributions, known as "soft targets", rather than relying solely on hard labels. First introduced by Hinton et al. (2015), this technique has enabled compact language models to approach the performance of much larger LLMs while using reduced architectures and training data (Hsieh et al., 2023).

More recently, distillation has been extended to complex reasoning tasks, spawning the field of reasoning distillation. For example, Li et al. (2025) present Fault-Aware Distillation via Peer-Review (FAIR), in which multiple teacher models critique each other's reasoning chains to improve fidelity. Likewise, Dai et al. (2024) propose training student models on key reasoning steps extracted from dual chain-of-thought explanations. These innovations not only enhance model interpretability but also substantially boost conceptual understanding in educational applications.

3 Methods

3.1 Synthetic Context Generation

Data Preprocessing We focus exclusively on Task 3 of the BEA Shared Task (Kochmar et al., 2025), and so we process the original dataset from Maurya et al. (2025) accordingly. From the provided validation set, we construct a filtered dataset:

$$\mathcal{D} = \{(C_i, R_i, L_i)\}_{i=1}^{N},$$

such that for each sample i:

- C_i: conversation history of each original element,
- R_i: each tutor's response,
- $L_i \in \{Yes, To\ Some\ Extent, No\}$: the gold label provided by "Providing Guidance"

In total, we have $N = 3,589$.

Generating Reasoning with Labels To enrich each response label with contextual justification, we leverage two state-of-the-art models, namely GPT-4o (OpenAI et al., 2024) and Claude-2.7 Sonnet (Anthropic, 2024). For each model, we process batches of 10 examples from our original dataset $\mathcal{D}$[1] alongside the system prompt in Appendix A. Each model then generates a justification J_i for sample i, drawing on the provided label L_i, the conversation history C_i, and the latest tutor response R_i. This yields an expanded dataset

$$\mathcal{D}' = \{(C_i, R_i, L_i, J_i)\}_{i=1}^{N},$$

where J_i is the synthetic justification associated with the ith response.

Selection of Justifications To ensure the quality and utility of the synthetically generated justifications, we conduct a manual selection process to identify the most suitable responses produced by the two models. The selection criteria are as follows:

- **Non-repetition:** Justifications that are repeated within the same batch are excluded to prevent redundant signals, which could lead to overfitting during downstream model training.
- **Linguistic diversity and specificity:** Selected justifications exhibit varied and distinctive vocabulary, reflecting the natural diversity found in human tutor responses. This diversity will then enhance the generalizability of models trained on the data.

[1] Batch size selected after varying from 1 to 50. We find that beyond 10 samples, both models tend to hallucinate or become overly generic and provide low-quality responses.

- **Adequate length and contextual richness:** Justifications are required to provide sufficient explanatory detail to offer meaningful context for the corresponding labeled response.

Extracting Critical Tokens For each synthetic justification J_i, we perform the following preprocessing steps:

1. Convert to lowercase and strip leading/trailing whitespace.
2. Remove all stopwords.
3. Tokenize the resulting string.
4. Apply stemming and lemmatization to each token.

Let

$$\mathcal{T}_i = \{ t_{i1}, t_{i2}, \ldots, t_{iK_i} \}$$

be the set of remaining tokens for sample i. We refer to $\mathcal{T}_i$ as the *context-critical* token set, and we use these tokens as our reward signals. With this, we now proceed to training our student model.

3.2 Expert Alignment Through Reinforcement Learning

To best align the student model's outputs with those of advanced LLMs, and potentially a human tutor expert, we introduce the **CO**ntextually-aware **D**istilled **E**valuator (**CODE**) framework. In **CODE**, reasoning is distilled through a multi-step transfer process, with tailored reward signals that guide the model to generate contextually relevant tokens for downstream classification.

3.2.1 Initial Supervised Learning

We begin by performing supervised fine-tuning (SFT) to teach the student model to generate J_i given the conversation history C_i and last response R_i. This initial stage aims to instill the desired format, tone, and length characteristic of expert-generated justifications. At this point, emphasis is placed not on the semantic quality or reasoning depth of the model's outputs, but rather on aligning the stylistic aspects of the responses to facilitate more efficient convergence during subsequent training phases. We have:

$$J_i = (j_{i,1}, j_{i,2}, \ldots, j_{i,T_i}),$$

where each justification is represented as a token sequence. Under a standard cross-entropy objective, the per-sample loss is

$$\ell_i(\theta) = -\frac{1}{T_i} \sum_{t=1}^{T_i} \log p_\theta\big(j_{i,t} \mid C_i,\, R_i,\, j_{i,<t}\big),$$

where $p_\theta(\cdot)$ is the student model's predicted probability and $j_{i,<t} = (j_{i,1}, \ldots, j_{i,t-1})$. Averaging over all N samples gives the final SFT loss:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell_i(\theta)$$

The system prompt used as part of this instruction tuning is provided in Appendix C.

3.2.2 Applying GRPO with Semantic Rewards

After supervised fine-tuning, we now refine the student model's output quality via an online reinforcement-learning algorithm known as *Group Relative Policy Optimization* (GRPO) (Shao et al., 2024). We denote the student's policy by

$$\pi_\theta(J \mid C_i, R_i)\,,$$

parameterized by θ, which we adapt efficiently using low-rank adaptation (LoRA) (Hu et al., 2021) updates to the transformer weights.

Group sampling and baseline For each training example (C_i, R_i), the model samples a *group* of M candidate justifications:

$$\{ J_i^1, J_i^2, \ldots, J_i^M \} \sim \prod_{j=1}^{M} \pi_\theta(\cdot \mid C_i, R_i)\,.$$

Each candidate J_i^j is scored by a programmable reward function r_i^j. We then compute the group baseline as the mean reward:

$$b_i = \frac{1}{M} \sum_{j=1}^{M} r_i^j\,.$$

Reward design The total reward r_i^j is a weighted sum of three components:

$$r_i^j = w_{\mathrm{tok}}\, r_i^{\mathrm{tok}}(J_i^j) + w_{\mathrm{sent}}\, r_i^{\mathrm{sent}}(J_i^j) + w_{\mathrm{ppl}}\, r_i^{\mathrm{ppl}}(J_i^j)\,,$$

where:

$$r_i^{\mathrm{tok}}(J) = \sum_{t \in \mathcal{T}_i} \mathbf{1}\{t \text{ appears in } J\},$$

$$r_i^{\mathrm{sent}}(J) = \begin{cases} 1 & \text{if the sentiment of } J \text{ matches the gold label,} \\ 0 & \text{otherwise,} \end{cases}$$

$$r_i^{\mathrm{ppl}}(J) = -\frac{1}{|J|} \sum_{t=1}^{|J|} \log p_{\theta_0}\big(j_t \mid C_i, R_i, j_{<t}\big).$$

Here $\mathcal{T}_i$ is the context-critical token set for example i. In this reward scheme, we do not punish the student model arbitrarily for not generating trivial tokens. Likewise, it is only rewarded if it can generate the critical tokens that would be informative for the response's label based on the context provided.

Next, the sentiment score is given by a finetuned transformer based on DistilBERT (Sanh et al., 2020) (i.e., DistilBERT-based SST-2 classifier). While sentiment alone is not a reliable indicator of guidance quality (Wang et al., 2024), it provides a concrete and readily interpretable signal that can guide model generation. The inclusion of this sentiment-based reward facilitates faster convergence of the student model by offering an easier-to-learn proxy objective compared to directly optimizing for alignment with complex gold labels. Crucially, sentiment is not intended as a hard classification signal but rather a soft reward, encouraging the generation of justifications whose affective tone is consistent with the associated label. This approach helps steer the model's learning trajectory in a meaningful direction in light of GRPO's multiple response generation.

Finally, p_{θ_0} denotes the frozen base model (i.e., untrained student model) used to compute perplexity. Importantly, the perplexity score is added such that the trained model does not exploit the other reward signals by randomly inserting tokens as part of their outputs. This ensures that the final responses produced by the trained student model is still cohesive and human understandable.

Before performing the policy gradient update, these scores are then normalised to zero mean and unit variance to prevent their magnitude from dominating other scores, with the weights $(w_{\text{tok}}, w_{\text{sent}}, w_{\text{ppl}})$ balancing these signals. [2]

3.3 Final Classification

To produce the final label predictions, we append a trainable classification head atop the trained student model. The primary objective of this step is feature selection. That is, the student model has been previously trained to generate justifications containing critical tokens, and as such, this classification head aims to capture and interpret these contextual cues, mapping them effectively to the target label space. In this final training stage, the model is trained to associate its own generated output with the corresponding gold label.

We first map each gold label:

$$L_i \in \{\text{Yes, To Some Extent, No}\}$$

to a categorical target $y_i \in \{1, 2, 3\}$. Let θ^* denote the student model parameters after merging the LoRA adapters. Here, we freeze all θ^* and add a dense feedforward layer with parameters $\phi = (W, b)$, where

$$W \in R^{3\times d}, \quad b \in R^3.$$

However, instead of training it on one forward pass on each example (C_i, R_i), we first generate the student model's full response by

$$\hat{J}_i \;=\; \arg\max_{J} \; \pi_{\theta^*}(J \mid C_i,\, R_i).$$

We encode this output using the student model's tokenizer, and train ϕ using standard cross-entropy loss on the last token hidden state:

$$\mathcal{L}_{\text{CE}}(\phi) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{3} \mathbf{1}[y_i = c] \;\log \hat{p}_{i,c}.$$

where $\hat{p}_{i,c}$ is the softmax of the classification head's predicted labels.

This pooling design choice to use the last token hidden state as the representation is particularly motivated by the architecture of decoder-only transformers, which lack a dedicated classification token such as [CLS] found in encoder-based models (Fu et al., 2023). The last token in our generated outputs (i.e., each justification) functions as a natural summary or conclusion to the token sequence, providing a meaningful contextual embedding that reflects the entire output. This approach balances computational efficiency, avoiding the increased complexity of attention-based pooling, and mitigates the potential noise or dilution of critical token signals that may arise with mean pooling strategies (Suganthan et al., 2025).

External Benchmarking In addition to comparing against gold labels and the CodaBench leaderboard submission, we also evaluate whether our method could outperform direct prompt-tuning of the proprietary models. For each example, we prompt GPT-4o and Claude-2.7 Sonnet to predict the guidance label without revealing the true label and recorded their accuracy on the validation set. The exact prompts used for this experiment are provided in the Appendix B.

[2] We experimented with several weighting schemes but observed only minor, non-meaningful variations. As such, for our final implementation we adopted uniform weights, assigning a value of 1 to each.

Model	Validation set				CodaBench set			
	Ex. F1	Ex. Acc	Len. F1	Len. Acc	Ex. F1	Ex. Acc	Len. F1	Len. Acc
GPT-4o	56%	69%	75%	83%	49%	58%	70%	75%
Claude-2.7 Sonnet	61%	70%	73%	81%	—	—	—	—
CoDE	**64%**	**74%**	**83%**	**89%**	**53%**	**63%**	**72%**	**78%**

Table 1: Evaluation of all models across both validation set and the CodaBench test set. Due to the limited nature of submission on CodaBench set, results from Claude-2.7 Sonnet were not submitted in the competition. All result reported has been rounded to the nearest percent. The final reported score on the official leaderboard for CoDE is slightly higher than the reported value in this table, but because the baseline score of GPT-4o is not reported there, we report the values of CoDE from the unofficial table for consistency.

3.4 Experimental Setup

We used the Unsloth-provided "Phi-3.5-mini-instruct" (Daniel Han and team, 2023; Abdin et al., 2024) as our student model. The original validation set was further split 80/20 into training and test subsets. All data preprocessing ran on an NVIDIA L40 GPGPU, with model training and evaluation performed on an NVIDIA A100 GPU. In total, preprocessing, training, and evaluation consumed over 70 GPU-hours. Complete details on training hyperparameters, such as GRPO and LoRA parameters, are detailed in the Appendix.

4 Results

As shown in Table 1, **CODE** consistently outperforms state-of-the-art baselines, achieving the tenth position in the final CodaBench ranking. We attribute this improvement both to the quality of the synthetic data and to the student model's ability to capture hidden features from the extended context. Our results suggest that existing SOTA models possess an implicit notion of "good guidance", and their generated outputs can be effectively transferred to smaller models. This observation corroborates prior work demonstrating that large language models can serve as an effective tutors, offering substantial instructional value, albeit not at expert-level proficiency (Wollny et al., 2021).

Notably, on both the validation set and, to a lesser extent, the CodaBench benchmark, **CODE** exhibits a larger gain when evaluated with lenient F1 compared to exact F1, with improvements under strict scoring criteria remain modest. This pattern indicates that fine-tuning renders **CODE** less sensitive to the ambiguous label that is "To some extent". The strong labels, "Yes" or "No", are much easier to deduce, with clearer human definition, but this "middle-ground" is much more nuanced, and since finetuning is known to reduce LLMs' general reasoning (Luo et al., 2025), this drop may be inevitable. When pedagogical value differs only slightly, we see that even among human experts, these are difficult to discern (Macina et al., 2023).

5 Conclusion

In this paper, we have investigated the potential of modern large language models to both generate and train on synthetic data that emulate expert human reasoning in educational guidance through our **CODE** framework. Across both our validation and CodaBench test sets, our approach consistently outperforms SOTA baselines and aligns reasonably well with human judgments. However, our findings also underscore the persistent challenge of the absence of a formal, operational definition of this pedagogical quality. In particular, nuances embodied by the "middle-ground" label appear too subtle or demand too much data for current LLMs to learn reliably.

As future direction, we advocate for the continued development of explicit metrics that systematically map these latent pedagogical features to models' outputs. By grounding fine-tuning in a well-defined, feature-based evaluation framework, we can move beyond black-box learning of hidden signals and instead foster more robust, transparent, and interpretable AI tutoring systems.

Limitations

5.1 Models Faithfulness and Prompt Sensitivity

The dataset created is not guaranteed to match reasoning provided by expert tutors. While we have conducted manual inspections on samples in the synthetic data to ensure some level of consis-

tency between reasoning and the label provided, this cannot be assured.

Furthermore, the SFT prompt used for training may not be optimal. This was chosen only after a few iterations of prompt tuning on a sample of the synthetic data.

5.2 Distillation Cost

Both models used are not open source nor free. Generating these takes extensive time on paid models, limiting the number of reasoning responses to one per sample. Ideally, we would like to expand this dataset further by generating multiple responses under various prompts to better simulate the diversity in thinking among real human tutors.

Ethical Consideration

Our research adheres strictly to ethical standards, using publicly available datasets as well as following distillation restriction carefully. We uphold the principles of fairness, accountability, and academic integrity throughout the research process.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Erfan Al-Hossami, Razvan Bunescu, Ryan Teehan, Laurel Powell, Khyati Mahajan, and Mohsen Dorodchi. 2023. Socratic questioning of novice debuggers: A benchmark dataset and preliminary evaluations. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 709–726, Toronto, Canada. Association for Computational Linguistics.

Anthropic. 2024. Claude 2.7 sonnet. [Large language model].

Gavin Clark and Sarah Egan. 2015. The socratic method in cognitive behavioural therapy: A narrative review. *Cognitive Therapy and Research*, pages 1–17.

Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. Stepwise verification and remediation of student reasoning errors with large language model tutors. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8386–8411, Miami, Florida, USA. Association for Computational Linguistics.

Chengwei Dai, Kun Li, Wei Zhou, and Songlin Hu. 2024. Beyond imitation: Learning key reasoning steps from dual chain-of-thoughts in reasoning distillation. *Preprint*, arXiv:2405.19737.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Zihao Fu, Wai Lam, Qian Yu, Anthony Man-Cho So, Shengding Hu, Zhiyuan Liu, and Nigel Collier. 2023. Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder. *Preprint*, arXiv:2304.04052.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *Preprint*, arXiv:2305.02301.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Ekaterina Kochmar, Kaushal Kumar Maurya, Kseniia Petukhova, K. V. Aditya Srivatsa, Anaïs Tack, and Justin Vasselli. 2025. Findings of the bea 2025 shared task on pedagogical ability assessment of AI-powered tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*.

Nguyen-Thinh Le. 2019. How do technology-enhanced learning tools support critical thinking? *Frontiers in Education*, 4.

Zhuochun Li, Yuelyu Ji, Rui Meng, and Daqing He. 2025. Learning from committee: Reasoning distillation from a mixture of teachers with peer-review. *Preprint*, arXiv:2410.03663.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *Preprint*, arXiv:2308.08747.

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

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.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Paul Suganthan, Fedor Moiseev, Le Yan, Junru Wu, Jianmo Ni, Jay Han, Imed Zitouni, Enrique Alfonseca, Xuanhui Wang, and Zhe Dong. 2025. Adapting decoder-based language models for diverse encoder downstream tasks. *Preprint*, arXiv:2503.02656.

Anaïs Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. 2023. The BEA 2023 shared task on generating AI teacher responses in educational dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 785–795, Toronto, Canada. Association for Computational Linguistics.

Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. *Preprint*, arXiv:2205.07540.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2174–2199, Mexico City, Mexico. Association for Computational Linguistics.

Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachsler. 2021. Are we there yet? - a systematic literature review on chatbots in education. *Frontiers in Artificial Intelligence*, 4:654924.

A Synthetic Data Generation Prompt

For both GPT-4o and Claude, we used the following prompt to create our dataset:

Data Creation Prompt

You are an expert evaluator of Socratic-style tutoring dialogs in programming education. Your task is to justify the human-supplied quality label for the tutor's last response You will receive: conversation_history (full transcript up to, but NOT including, the tutor's latest reply, and the speaker turns are prefixed with "Student:" or "Tutor:", last_response (the tutor's latest reply, to be evaluated), and label (one of Yes, To some extent, No indicating whether the reply provides adequate, partial, or no helpful guidance to the student.These labels correspond to:

- Yes: The reply gives sufficient, specific, actionable guidance or hints that directly help the student correct their error or deepen understanding.
- To some extent: Contains some guidance, but it is vague, incomplete, or only tangentially helpful. Student would likely still struggle.
- No: Gives the answer outright without guidance, or offers no meaningful help, such as generic reassurance, topic change, or silence.

Return your result explaining why the provided label is appropriate as structured JSON with these keys: {"label_justification": string}

B Proprietary Model Label Prompt

To produce labels from both GPT-4o and Claude as our external baselines, we used the following prompt:

Data Label Prompt

You are an expert evaluator of Socratic-style tutoring dialogs in programming education. You will receive: conversation_history (full transcript up to, but NOT including, the tutor's latest reply, and the speaker turns are prefixed with "Student:" or "Tutor:", and last_response (the tutor's latest reply, to be evaluated). Your job is to provide a label (one of Yes, To some extent, No indicating whether the reply provides adequate, partial, or no helpful guidance to the student. These labels correspond to:

- Yes: The reply gives sufficient, specific, actionable guidance or hints that directly help the student correct their error or deepen understanding.
- To some extent: Contains some guidance, but it is vague, incomplete, or only tangentially helpful. Student would likely still struggle.
- No: Gives the answer outright without guidance, or offers no meaningful help, such as generic reassurance, topic change, or silence.

Return your result explaining why the provided label is appropriate as structured JSON with these keys: { "label": string }

C SFT Training System Prompt

The system prompt used to align the model's behaviour to that of a professional tutor is as follows:

System Prompt

You are a professional tutor. Your goal is to focus on whether the Last Response from the example is providing enough guidance (i.e, explaination, hints, guidance) to the student to act upon, progressing the conversation based on the conversation history. DO NOT continue the conversation, and you MUST use the Last Response provided. Focus on these characteristics:

1. If the Last Response provides specific, actionable guidance that identifies exactly where errors occur and offers clear steps forward, balancing encouragement with targeted correction while addressing misconceptions without giving away complete answers.
2. If the Last Response acknowledges problems but offer incomplete guidance—they might identify errors without explaining how to fix them, use ambiguous language, or address only part of the misconception, leaving students without clear direction on how to proceed.
3. If the Last Response fails to provide meaningful guidance by offering empty praise without addressing errors, changing the subject, reinforcing incorrect understanding, giving answers without explanation, or presenting completely irrelevant information that leaves students with no actionable path forward in solving their problem.

D LoRA Training Arguments

This details the full LoRA training parameters:

- `max_seq_length`: 2048
- `dtype`: `cuda`
- `load_in_4bit`: False
- `device`: `cuda`
- `device_map`: `cuda:0`
- `r`: 64
- `target_modules`: {`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`}
- `lora_alpha`: 64
- `lora_dropout`: 0
- `bias`: `none`
- `use_gradient_checkpointing`: `unsloth`
- `random_state`: 42
- `use_rslora`: True
- `loftq_config`: None

E GRPO Training Arguments

This details the full GRPO training arguments:

- `use_vllm`: **True**
- `learning_rate`: 5×10^{-6}
- `adam_beta1`: 0.9
- `adam_beta2`: 0.99
- `weight_decay`: 0.1
- `warmup_ratio`: 0.1
- `lr_scheduler_type`: `cosine`
- `optim`: `paged_adamw_8bit`
- `logging_steps`: 10
- `bf16`: True
- `per_device_train_batch_size`: 1
- `gradient_accumulation_steps`: 1
- `num_generations`: 6
- `max_prompt_length`: 2048
- `max_completion_length`: 256
- `num_train_epochs`: 5
- `max_steps`: -1
- `save_strategy`: `steps`
- `save_steps`: 250
- `max_grad_norm`: 0.1