Improving the Validity of Automatically Generated Feedback via Reinforcement Learning

Alexander Scarlatos[1], Digory Smith[2], Simon Woodhead[2], and Andrew Lan[1]

[1] University of Massachusetts Amherst
[2] Eedi
Contact Emails: {ajscarlatos,andrewlan}@cs.umass.edu

Abstract. Automatically generating feedback via large language models (LLMs) in intelligent tutoring systems and online learning platforms has the potential to improve the learning outcomes of many students. However, both feedback generation and evaluation are challenging: feedback content has to be valid especially in subjects like math, which requires models to understand the problem, the solution, and where the student's error lies. Feedback also has to be pedagogically valid to reflect effective tutoring strategies, such as explaining possible misconceptions and encouraging the student, among other desirable features. In this work, we address both problems of automatically generating and evaluating feedback while considering both correctness and alignment. First, we propose a rubric for evaluating math feedback and show that GPT-4 is able to effectively use it to annotate human-written and LLM-generated feedback. Second, we propose a framework for feedback generation that optimizes both correctness and alignment using reinforcement learning (RL). Specifically, we use GPT-4's annotations to create preferences over feedback pairs in an augmented dataset for training via direct preference optimization (DPO). We show that our methods significantly increase the correctness and alignment of generated feedback with Llama 2, an open-source LLM, qualitatively analyze our generation and evaluation systems using case studies, and outline several areas for future work.[3]

Keywords: Feedback Generation · Human Preference Alignment · Math Education · Reinforcement Learning

1 Introduction

Providing students with helpful feedback can be critical to their learning, allowing them to quickly address and learn from mistakes. Prior work has shown that delivering immediate automated feedback to students in intelligent tutoring systems and online learning platforms can improve learning outcomes [12,25]. However, doing so is challenging since generated feedback should satisfy a wide

[3] Our code is available at `https://github.com/umass-ml4ed/feedback-gen-dpo`. The authors thank Schmidt Futures and the NSF (under grants IIS-2118706 and IIS-2237676) for partially supporting this work.

variety of requirements: it should convey an understanding of the question and why the student's response is incorrect, as well as be aligned with *educational goals* and pedagogical theory. For example, identifying student misconceptions, providing hints, and using encouraging language to promote a growth mindset [2,33] can be helpful, but simply giving away the answer could be detrimental.

Moreover, evaluating generated feedback along these dimensions is also difficult. Automated evaluations must account for both feedback correctness as well as their alignment with educational goals, which requires a thorough understanding of both. Additionally, even when expert-written feedback examples are given as reference, text similarity-based metrics may be unreliable since there are many ways to write valid feedback, and text overlap can emphasize irrelevant features while neglecting more significant ones [1,20]. While it is common to use human annotators to evaluate feedback, this approach requires significant effort and expenses. Therefore, the lack of reliable, automated feedback *evaluation* methods becomes a bottleneck for developing feedback *generation* methods.

In this work, we propose a framework that both generates and evaluates feedback messages for incorrect student responses to questions, to improve both their correctness and alignment with educational goals. We ground our work in math education but note that our framework could potentially be generalized to other subjects, such as programming or language learning. First, we propose a *rubric* for evaluating generated feedback and show that LLMs, particularly GPT-4, achieve high agreement with humans in their evaluations.

Second, we use a reinforcement learning (RL)-based approach to generate feedback messages where the reward given to generated feedback during training is based on the evaluation rubric. Moreover, to avoid repeatedly using GPT-4 to evaluate feedback during training, we use direct preference optimization (DPO) [24], an offline RL algorithm, to align the generated feedback with educational goals. This approach is similar to aligning LLMs with human [39] or AI [16] preferences. We experiment on a dataset that consists of feedback messages written by math teachers for each incorrect option in multiple-choice questions. Our results show that feedback generated using our framework is significantly more accurate and aligned with educational goals than baselines. Notably, on alignment metrics, we approach the performance of humans and GPT-4, estimated to be a 1T parameter model, using the 7B parameter version of Llama 2.

2 Related Work

2.1 Feedback Generation

There are many existing approaches for automatic feedback generation. One common method is to use engineered features to detect errors in student responses, and then use a rule-based system to provide relevant feedback or hints [3,12,15,25,29,30]. This method is popular since it is interpretable and reliable but requires significant human effort to adapt to new question types. A recent and more general approach to feedback generation is using large language models (LLMs), either through prompting [1,20,23,32] or fine-tuning [10]. However, prompting pre-trained LLMs requires them to be capable of understanding educational goals, but fine-tuning can yield poor results without significant amounts of aligned training data. We address these concerns in our work by fine-tuning on an augmented dataset annotated with alignment labels.

2.2 Feedback Evaluation

Several recent works have used rubrics to evaluate feedback [10,32], and works in other domains have found success in using LLMs to evaluate open-ended text where their judgements correlate with human judgements [6,13,22]. However, most prior works on feedback generation tend to rely on human annotators for reliable evaluation [1,9,10,32]. One recent work [11] uses GPT-4 to evaluate math feedback with a rubric and finds high agreement with human annotations. However, they only use GPT-4 to evaluate human-written feedback, while we evaluate feedback written by both humans and LLMs. Including this LLM feedback helps us uncover GPT-4's shortcomings in feedback evaluation, particularly that it can struggle to identify when feedback inaccurately addresses student errors or provides invalid suggestions.

3 Methodology

We now detail our framework for the two main tasks of feedback generation and evaluation. Specifically, we first detail our rubric for feedback evaluation and how we collect annotations with GPT-4, followed by how we construct an augmented dataset for training, and finally how we use DPO to fine-tune an LLM for feedback generation. We first define some notations for our tasks. Given a dataset $\mathcal{D}$ of N math questions, we define the i-th question as $(q^{(i)}, c^{(i)}, e^{(i)}, \{d_j^{(i)}, f_j^{(i)} | j \in \{1, \dots, M\})$. Here, $q^{(i)}$ is the question text, $c^{(i)}$ is the correct answer to the question, $e^{(i)}$ is a textual explanation of the question's solution, $d_j^{(i)}$ is an incorrect, student-generated answer to the question, $f_j^{(i)}$ is a textual feedback message to give to a student when their answer is $d_j^{(i)}$, and M is the number of different incorrect answers given for each question. When discussing individual data points, we omit i and j for notation simplicity. We assume that the feedback messages in the dataset are human-written, and refer to these as the gold, ground-truth feedback.

3.1 Feedback Evaluation

We now detail our rubric for evaluating feedback given to students for their incorrect answers. In addition to correctness, we aim to evaluate feedback messages on their alignment with educational goals, including those associated with a growth mindset [2,33]. We take inspiration from prior works using rubrics for feedback evaluation [10,32] and include aspects to target common errors that LLMs make when generating feedback. Specifically, our rubric evaluates feedback on five different aspects, each of them resulting in a binary-valued label:

- **Correct (COR.)** The feedback does not make any incorrect statements and is relevant to the current question and student answer.
- **Revealing (REV.)** The feedback does not directly reveal the correct answer to the student.
- **Suggestion (SUG.)** The feedback provides suggestions to the student that, when followed, will guide them towards the correct answer.
- **Diagnostic (DIA.)** The feedback correctly points out the error the student made or the misconception underlying their answer.
- **Positive (POS.)** The feedback is positive and has an encouraging tone.

We now define the *rubric function*, r, which assigns labels to any feedback given a corresponding question and incorrect answer, and a final scalar-valued *rubric score*, s, which indicates the feedback's overall quality:

$$\mathrm{r}(f|q,c,e,d) = (y_{\mathrm{C}}, y_{\mathrm{R}}, y_{\mathrm{S}}, y_{\mathrm{D}}, y_{\mathrm{P}}) = \mathbf{y} \in \{0,1\}^5$$
$$s = y_{\mathrm{C}} \cdot \frac{y_{\mathrm{C}} + y_{\mathrm{R}} + y_{\mathrm{S}} + y_{\mathrm{D}} + y_{\mathrm{P}}}{5} \in [0,1]$$

where except for correctness, other rubric aspects are equally weighted. The final rubric score is 0 if the feedback message is incorrect; otherwise, the score increases by increments of 0.2 for every rubric aspect the feedback satisfies.

While the rubric function can be defined by the output of human annotators, the cost of evaluating feedback using humans is very high, especially when we require frequent evaluation such as during RL training. To address this issue, we use GPT-4, known for its ability to generalize to new tasks, to define a version of the rubric function, $\mathrm{r}_{\text{GPT-4}}$. Using zero-shot chain-of-thought prompting [14], we ask GPT-4 yes or no questions related to each of the 5 labels, and use its output to get an estimated label $\mathbf{y}'$ and corresponding score s'. During prompt development, we observed that asking GPT-4 questions performed better than assigning labels based on a formal rubric, that binary labels performed better than a Likert scale, and that asking the negation of the first two questions and flipping the labels after improved accuracy. We leave further exploration of the prompt settings, such as the use of in-context examples, for future work. We provide an example prompt, output, and corresponding labels in Table 1.

3.2 Data Augmentation

We now detail our method for constructing an augmented dataset, which will be used for RL training as well as calculating agreement between GPT-4 and human annotations. For both of these tasks, we require both *positive* examples, i.e., feedback messages that score highly on the rubric, and *negative* examples, i.e., feedback that score poorly on the rubric. We observe that while most of the gold feedback messages in $\mathcal{D}$ score highly, feedback generated by prompting LLMs can be flawed or unaligned with the evaluation metric, thus scoring poorly. Therefore, we collect 3 LLM-augmented versions of $\mathcal{D}$, where each feedback $f_j^{(i)}$ is replaced with a generated version: $\mathcal{D}_R$, where feedback is generated using

Table 1. Example prompt, output, and labels for GPT-4 feedback evaluation.

System	You are a math education expert.
Prompt	Your job is to evaluate feedback given to students on math problems. Here is the question, the correct solution, the incorrect answer the student gave, and the feedback given to the student: Question: emma works 36 hours a week. write this as a fraction in its simplest form. Correct Answer: $\frac{3}{14}$ Solution: There are $24 \times 7 = 168$ hours in a week, so the fraction is 36/168. This simplifies to $18/84 = 9/42 = 3/14$. Incorrect Answer: $\frac{18}{84}$ Feedback: It looks like you have simplified, but not fully. For the following questions, provide a short explanation and then answer with "Yes" or "No": 1. Does the feedback make any incorrect statements? 2. Does the feedback directly reveal the answer to the question? 3. Does the feedback give suggestions to the student on how to improve the answer? 4. Does the feedback correctly point out the misconception underlying the student's answer? 5. Does the feedback have a positive or encouraging tone?
Output	1. Explanation: The feedback is correct in stating that the student has simplified, but not fully. The student stopped at $\frac{18}{84}$, which can be further simplified to $\frac{3}{14}$. So, the feedback does not make any incorrect statements. Answer: No. 2. Explanation: The feedback does not provide the correct answer. It only indicates that the student's answer is not fully simplified. Answer: No. 3. Explanation: The feedback does not provide specific suggestions on how to improve the answer. It only states that the student has not fully simplified the fraction. Answer: No. 4. Explanation: The feedback correctly identifies that the student's misconception is not fully simplifying the fraction. Answer: Yes. 5. Explanation: The feedback is neutral in tone. It points out the error but does not use any positive or encouraging language. Answer: No.
Result	$(y'_{\text{C}} = 1,\, y'_{\text{R}} = 1,\, y'_{\text{S}} = 0,\, y'_{\text{D}} = 1,\, y'_{\text{P}} = 0) \rightarrow s' = 0.6$

few-shot prompting with random in-context examples, $\mathcal{D}_S$, where feedback is generated using few-shot prompting with the most similar examples, and $\mathcal{D}_Z$, where feedback is generated using zero-shot prompting. We refer to the union of the original dataset and LLM-augmented data $\mathcal{D}' = \bigcup\{\mathcal{D}, \mathcal{D}_R, \mathcal{D}_S, \mathcal{D}_Z\}$ as the augmented dataset. We use GPT-4 to annotate the feedback messages in this set, and we detail how we use these annotations for training in the next section.

3.3 Direct Preference Optimization

In order to generate feedback that scores highly on the rubric, we leverage direct preference optimization (DPO) [24], an offline RL algorithm, due to its simplicity and efficiency. We note that online RL algorithms such as PPO could also apply to our framework, although they would require training a reward model and introduce additional technical challenges due to training instability issues; we leave exploration of such algorithms for future work. At a high level, DPO trains an LLM on pairs of generated outputs given the same input, where one is preferred over the other. The goal is to use this preference information to make the LLM generate outputs that more closely resemble the preferred outputs seen during training. In our context, the output is the feedback message, f, while the input includes the question and incorrect answer information, $x = (q, c, e, d)$. During training, we minimize the DPO objective, i.e.,

$$\min_{\theta} -\mathbb{E}_{(x,f_w,f_l)\sim\mathcal{D}_{\text{DPO}}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(f_w|x)}{\pi_{\text{ref}}(f_w|x)} - \beta\log\frac{\pi_\theta(f_l|x)}{\pi_{\text{ref}}(f_l|x)}\right)\right],$$

where f_w is preferred over f_l as the feedback for x, $\mathcal{D}_{\text{DPO}}$ is a curated dataset containing these feedback pairs and preferences, π_θ is the trained LLM, i.e., a text generation "policy", parameterized by θ, π_{ref} is a frozen *reference* LLM, and β is a hyperparameter to control how far π_θ can deviate from π_{ref}. We now detail how we construct $\mathcal{D}_{\text{DPO}}$ using both feedback from the augmented dataset and mismatched feedback from the gold dataset.

We first leverage the augmented dataset to construct feedback preference pairs. For each unique $x \in \mathcal{D}'$, we have 4 feedback messages, 1 human and 3 LLM-generated, from which we construct $\binom{4}{2} = 6$ unique pairs. We then use the score s' for each feedback to determine which feedback is preferred, and exclude pairs that have the same score. For instance, consider a case where some x has possible feedback messages f_1, f_2, f_3, and f_4, with scores 1.0, 0.8, 0.4, and 0.4, respectively. We then produce the preference pairs (f_1, f_2), (f_1, f_3), (f_1, f_4), (f_2, f_3), and (f_2, f_4), where the first feedback is the preferred one in each pair.

We also use mismatched feedback from the gold dataset to construct additional preference pairs. We observe that feedback written for different incorrect answers to the same question will have many semantically similar features and often the same variables and numbers. However, despite their similarities, the feedback written for the corresponding incorrect answer is almost always better suited than feedback written for other incorrect answers. Therefore, these mismatched feedback are excellent *hard* negatives since it is hard for algorithms to distinguish between them and good feedback; finding such hard negatives has been shown to be the key to contrastive learning [27]. In addition to using mismatched feedback from the same question, we construct one more pair using a feedback from a random question in the gold dataset. For instance, for $x_1^{(i)}$ and $M = 3$, we construct the preference pairs $(f_1^{(i)}, f_2^{(i)})$, $(f_1^{(i)}, f_3^{(i)})$, and $(f_1^{(i)}, f_{j'}^{(i')})$, for some random $i' \in [1, N]$ and $j' \in [1, M]$, where $f_1^{(i)}$ is preferred in all pairs.

4 Experiments

We now detail all experiments we conduct to validate our framework for feedback generation and evaluation. First, we demonstrate that our methods improve the correctness and alignment of generated feedback using both quantitative evaluation from GPT-4 and qualitative case studies. Second, we demonstrate that

GPT-4 has high agreement with human annotations on our rubric, justifying its use as an evaluator, and further investigate its shortcomings using case studies.

4.1 Dataset

We validate our framework using a dataset of middle school-level math multiple choice questions from Eedi, a math learning platform. The questions cover a variety of number sense concepts including fractions, exponents, rounding, and many others. All questions and feedback messages are written by real math teachers, deployed to real students, and are generally high quality. There are a total of 1,956 questions in the dataset and each question has a total of 3 incorrect options and a ground truth human-written feedback for each. We remove questions that require images and ones with processing errors, resulting in 1,418 questions. We divide these into a train/validation/test split of 850/284/284 questions and correspondingly 2,550/852/852 incorrect answers and corresponding feedback.

4.2 Experimental Setting

Data Augmentation We use two LLMs to generate feedback for our augmented dataset: `code-davinci-002` (Codex) [4] for $\mathcal{D}_R$ and $\mathcal{D}_S$ since it has strong few-shot prompting ability, and `gpt-3.5-turbo` for $\mathcal{D}_Z$ since its zero-shot ability is much better than `code-davinci-002`. We use 2 in-context examples for few-shot prompts, only select examples from the train set, and use the S-BERT model `all-distilroberta-v1` [26] to measure similarity for $\mathcal{D}_S$. We prompt the models with questions, correct answers and incorrect answers, but not full solutions to make the task harder and increase the amount of incorrect feedback. To reduce costs, we randomly select a subset of $\mathcal{D}'$ to be annotated by GPT-4. Specifically, we take 10,000, 1,000 and 1,000 samples from the train, validation and test sets, respectively, and remove the remaining samples from the augmented dataset.

Feedback Generation Models We primarily use the instruction-tuned Llama-2 7B Chat model [34] from HuggingFace [35] for feedback generation, loaded with 8-bit quantization [7]. For both supervised fine-tuning (SFT) and DPO, we train LoRA adapters [8] on all weight matrices, setting $r = 32$, $\alpha = 16$, and dropout$= 0.05$. We train using the AdamW optimizer with a learning rate of 3e-4 with warmup for 10% of steps and an effective batch size of 64 using gradient accumulation. We train for 3 epochs, which we find minimizes the loss on the validation set. For DPO, we set $\beta = 0.5$ and use the SFT model for initialization and as the reference model, which empirically outperformed using the base model. At inference time, we use greedy decoding and set the maximum new tokens to 200.

Metrics When evaluating feedback, we report the average of each rubric label in $\mathbf{y}'$ and the corresponding scores s' assigned by GPT-4. We note that GPT-4 will very rarely fail to assign labels when feedback is unrelated to the current question, in which case we automatically assign label values of 0. We use a temperature of 0 and 300 maximum tokens for GPT-4 decoding. We also use two popular

Table 2. Quantitative results of feedback generation across methods. Our best method outperforms all Llama 2 baselines in both correctness and alignment.

	COR.	**REV.**	**SUG.**	**DIA.**	**POS.**	**Score**	**ROU.**	**BER.**
Human	0.91	0.98	0.67	0.82	0.41	0.73	1.00	1.00
GPT-4	0.95	0.96	0.99	0.93	1.00	0.94	0.19	0.57
Zero-shot	0.63	0.63	0.74	0.43	**1.00**	0.49	0.16	0.55
SFT	0.65	**0.98**	0.49	0.68	0.19	0.49	**0.29**	**0.61**
DPO (Score)	0.70	0.93	**0.95**	0.82	0.66	0.65	0.22	0.57
DPO (Score + Mismatch)	**0.77**	0.96	**0.95**	**0.86**	0.57	**0.71**	0.23	0.57

reference-based metrics with the human-written feedback as reference: ROUGE-L (**ROU.**) [17] which is based on textual overlap, and the F1 of the BERTScore (**BER.**) [38] using the recommended `microsoft/deberta-xlarge-mnli` model, which is based on token-level semantic similarity.

4.3 Feedback Generation

We now show that we can improve both the correctness and alignment of generated feedback using our framework. We primarily focus on using Llama 2 Chat, an open-source LLM with 7B parameters, where we compare several versions of the model: **Zero-Shot**, i.e., simply prompting the base LLM, **SFT**, i.e., fine-tuning the base LLM on the gold feedback set, **DPO (Score)**, i.e., training the LLM with DPO only on the augmented dataset, and **DPO (Score + Mismatch)**, i.e., training the LLM with DPO on the augmented dataset and mismatched feedback. We additionally compare with the gold, human-written feedback in the dataset, as well as feedback generated by GPT-4. We use the same prompt for all methods, where we instruct the model to generate short and helpful feedback and to follow a version of the evaluation rubric.

Quantitative Analysis Table 2 shows the average rubric labels and scores assigned by GPT-4 on all feedback in the test set, as well as the ROUGE-L and BERTScore values for reference. We see that DPO (Score + Mismatch) significantly improves the feedback scores compared to baselines (a 45% increase compared to Zero-Shot and SFT), showing that our data augmentation and training setup is highly effective at improving the quality of feedback generation with Llama 2. We additionally observe that including the mismatched feedback messages substantially increases the correctness of generated feedback, confirming their effectiveness as hard negative examples. Surprisingly, SFT does not outperform Zero-Shot on score, which shows that the standard fine-tuning setup is not effective for feedback generation. We can also see that ROUGE-L and BERTScore are unreliable estimates of feedback quality since they are highest on SFT, primarily because it copies the style of the gold feedback the closest.

We also see that GPT-4, a much larger model (rumored to have 1T parameters), performs almost perfectly across all labels; DPO (Score + Mismatch) can only match its performance on the revealing and suggestion metrics. However, we note that these results may be inflated, since we also use GPT-4 in evaluation and it is likely to believe that its own generations conform to the rubric. Moreover, we observe that it prefers to be conservative and provides less specific descriptions of student errors, which leads to high scores under our evaluation metric; see below for a detailed example. Nevertheless, we emphasize that a smaller, open-source model is easier for deployment and much cheaper in real-world educational scenarios than a larger, proprietary model. Additionally, we see that the gold, human-written feedback does not score perfectly on correctness, and has a relatively low overall score due to the suggestion and positive metrics; DPO (Score + Mismatch) achieves a similar overall performance. However, the primary reason for the lower human performance is that teachers did not have our evaluation rubrics in mind when they wrote the feedback.

Qualitative Analysis We also performed qualitative studies to compare the outputs of the different methods and find cases where they succeed or fail; the main findings are: 1) Zero-Shot produces feedback with the right style but struggles to follow instructions, particularly by not identifying the error or revealing the correct answer, and is prone to hallucinations and numerical errors. 2) SFT produces feedback that is generally short and blunt, and usually attempts to identify the error although is often incorrect. 3) DPO (Score) produces feedback that attempts to identify the error, adding details and questions to provide implicit suggestions and increase positivity. It also produces incorrect outputs although less so than SFT. 4) DPO (Score + Mismatch) is more accurate than DPO (Score) in identifying the error. 5) GPT-4 produces feedback with smooth and coherent language but tends to avoid mistakes by not clearly pointing out the error. When it does detail the error, it is often incorrect, although less often than the other methods. 6) For all methods, most failures occur because the LLM cannot correctly identify the error in the incorrect answer. This failure happens more often when multiple reasoning steps are required to arrive at the student's answer. This observation suggests that although state-of-the-art LLMs claim to excel at mathematical reasoning, they cannot conceive how students may make errors [11,18,20]. One solution could be prompting the LLM to evaluate relevant calculations via code execution before writing the feedback [5,31].

To provide more specific examples of several of our findings, we show a representative question and incorrect answer along with corresponding feedback generated by each method in Table 3. In this example, we see that the human-written feedback directly points out the error in the incorrect answer that the student only considered the denominators, and also suggests simplifying the fraction on the right. GPT-4 prefers to be vague to avoid mistakes by pointing out that the ratio should be simplified and gives a suggestion based on this, but does not explicitly identify why the student's answer was wrong. Zero-Shot does not point out the student's error and directly answers the question, even though it has been instructed to not do so; it then hallucinates an incorrect and irrelevant multiplication before finishing with a positive message. SFT makes a very generic

Table 3. Feedback generated by all methods for one question and incorrect answer.

Input	
Question: Erica and Rebecca share some marbles in the ratio $\frac{1}{4}$: $\frac{6}{8}$. Erica has □ marbles than Rebecca. Which of the following belongs in the gap? Correct Answer: a third as many Incorrect Answer: twice as many	
Source	**Feedback**
Human	Not quite, the denominator has doubled but the fractions don't have that relationship. Try simplifying 6/8 and then compare it to 1/4 ✓
GPT-4	Good effort, but remember to simplify the ratios first. Erica's share is not twice as much, but a fraction of Rebecca's. Try to compare 1/4 to 3/4 again. Keep going! ✓
Zero-shot	Great effort! You are on the right track. Since the ratio is 1/4: 3/4, Erica has 1/4, which is a third of 3/4. To find the number of marbles Erica has, you can multiply 3/4 by 3, which gives you 9. So, Erica has 9 marbles. Keep practicing, and you'll get it right next time! ✗
SFT	I think you have simplified the second fraction incorrectly. ✗
DPO (Score)	I think you have simplified 6/8 to 1/2. If we simplify 6/8 to 3/4, how many marbles does Erica have? How many does Rebecca have? What is the difference between them? ✗
DPO (Score + Mismatch)	I think you have found 8/4 = 2. But we need to simplify 6/8 to get the same units as 1/4. Then we can see how many more or less Erica has. ✓

Table 4. GPT-4's agreement with human annotations across all rubric labels.

Label	**Acc.**	**Prec.**	**Rec.**	**F1**
COR.	0.77	0.80	0.82	0.81
REV.	0.91	0.94	0.97	0.95
SUG.	0.73	0.76	0.73	0.73
DIA.	0.68	0.56	0.85	0.68
POS.	0.72	0.78	0.28	0.41
Avg.	0.76	0.77	0.73	0.71

guess at what the error is and does not give a suggestion for improvement. DPO (Score) guesses a more specific but incorrect error, gives a somewhat misleading suggestion, and uses questions to encourage the student. Finally, DPO (Score + Mismatch) correctly identifies the error and gives a helpful suggestion.

4.4 Feedback Evaluation

Since we use GPT-4 to quantitatively evaluate feedback, we need to verify that GPT-4 can indeed label feedback accurately using our rubric. To do so, we randomly sample 80 feedback messages from the augmented test set and manually evaluate them on the rubric. We also recruited a graduate student with extensive teaching experience to evaluate these feedback messages for an additional set of human annotations. To reduce bias, we do not show human annotators GPT-4's annotations or tell them whether a feedback message is human-written or LLM-generated. We compute agreement between human annotators using the Cohen's

Table 5. Incorrect GPT-4 rubric annotation examples with explanations.

Label	**Input**	**Generated Explanation**
COR. (FP)	Question: a jacket is reduced in a sale by 20%. a customer has a loyalty card for a further 10% off the price. what is the total percentage reduction in the price of the jacket? Correct Answer: 28% Incorrect Answer: 8% Feedback: I think you have added the two discounts together. We need to use multiplication to find the overall discount.	The feedback does not make any incorrect statements. It correctly identifies the student's mistake of adding the discounts together instead of multiplying them. ✗
COR. (FN)	Question: i am thinking of a number. i divide it by 0.5 and my answer is 10. what number am i thinking of? Correct Answer: 5 Incorrect Answer: 20 Feedback: You multiplied by 0.5 instead of dividing by 0.5.	The feedback is incorrect because the student actually multiplied by 2, not 0.5. The correct feedback should be "You multiplied by 2 instead of dividing by 0.5." ✗
DIA. (FP)	Question: which fraction is equivalent $\frac{3}{7}$ Correct Answer: $\frac{9}{21}$ Incorrect Answer: $\frac{9}{7}$ Feedback: You have multiplied the top and bottom by 3, but you have to multiply by 3 again to get the correct answer	The feedback correctly identifies that the student has not multiplied the denominator by the same number as the numerator, which is the underlying misconception. However, the suggested correction is incorrect. ✗

kappa statistic, resulting in 0.53, 0.46, 0.35, 0.53 and 0.29 on the COR., REV., SUG., DIA. and POS. labels, respectively. This moderate agreement shows that while humans generally agree, they occasionally make errors due to the difficulty of the task, and have lower agreement on more subjective labels such as POS.

We measure the accuracy (**Acc.**), precision (**Prec.**), recall (**Rec.**), and **F1** of GPT-4's annotations with respect to human annotations on this set, and report the results averaged across both human annotators in Table 4. We observe that GPT-4 generally agrees with human annotations, with an average accuracy of 76% across labels. We also compute the Pearson correlation coefficient of the final rubric scores between GPT-4 and human annotations, resulting in 0.56 on average, indicating moderate overall correlation. In contrast, ROUGE-L and BERTScore both have average correlations of 0.40 with human-annotated rubric scores. Not only are these correlations smaller, but they are biased upward since the human-written feedback messages, which generally have high rubric scores, automatically get ROUGE-L and BERTScore values of 1.

However, GPT-4 still struggles in a few key aspects and we provide examples of erroneous annotations in Table 5. Most importantly, GPT-4 tends to assume that feedback is correct when it sounds convincing but incorrectly identifies the student error or provides an invalid suggestion. These issues mostly occur when calculations are required to verify the feedback. Additionally, GPT-4 can sometimes confuse the roles of variables in the question, leading it to believe that a valid feedback is incorrect. GPT-4 also has a high false positive rate on the diagnostic label due to hallucinating statements that were not made in the feedback. We note that it may be possible to resolve these issues using additional prompt engineering or tools such as self-reflection [28] and code execution to evaluate math expressions [5,31]. Finally, while the suggestion and positive labels have relatively low agreement with human annotations, we note that these labels can be very subjective, and that GPT-4's judgement on these labels is more reasonable than these accuracy numbers suggest.

5 Conclusions and Future Work

In this work, we proposed a framework for automated feedback generation and evaluation via LLMs for students' incorrect answers in math multiple-choice questions. Our framework accounts for both the mathematical correctness of the feedback and its alignment with good pedagogical practices. We show that using a data augmentation and preference optimization approach, we can generate high-quality feedback using Llama 2 7B, a small and open-source LLM. We also show that GPT-4 can evaluate feedback rather accurately using a rubric and that its annotations are helpful for training the feedback generation method. There are many avenues for future work. First, we can apply our framework to other RL algorithms such as PPO, or non-RL approaches such as overgenerate-and-rank. Second, we can evaluate our final feedback generation task via a large-scale human evaluation or classroom study, which would alleviate concerns on GPT-4's annotations being biased. Third, we can test our framework's generalizability by applying it to other domains such as programming or language learning, or other scenarios such as hint generation or student-instructor conversations. Finally, we can consider tailoring feedback to each student according to their knowledge levels [19], especially for open-ended questions, since student errors can likely be detected from these responses [21,36,37].

References

1. Al-Hossami, E., Bunescu, R., Teehan, R., Powell, L., Mahajan, K., Dorodchi, M.: Socratic questioning of novice debuggers: A benchmark dataset and preliminary evaluations. In: Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications (BEA2023@ACL). pp. 709–726 (Jul 2023)
2. Boaler, J.: Ability and mathematics: The mindset revolution that is reshaping education. The Forum **55**, 143–152 (2013)
3. Botelho, A., Baral, S., Erickson, J.A., Benachamardi, P., Heffernan, N.T.: Leveraging natural language processing to support automated assessment and feedback for student open responses in mathematics. Journal of Computer Assisted Learning **39**(3), 823–840 (2023)
4. Chen, M., Others: Evaluating large language models trained on code (2021)
5. Chen, W., Ma, X., Wang, X., Cohen, W.W.: Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. arXiv preprint arXiv:2211.12588 (2022)

6. Chiang, C.H., Lee, H.y.: Can large language models be an alternative to human evaluations? arXiv preprint arXiv:2305.01937 (2023)
7. Dettmers, T., Lewis, M., Belkada, Y., Zettlemoyer, L.: Llm.int8(): 8-bit matrix multiplication for transformers at scale (2022)
8. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models (2021)
9. Jia, Q., Cui, J., Xiao, Y., Liu, C., Rashid, P., Gehringer, E.F.: All-in-one: Multi-task learning bert models for evaluating peer assessments. arXiv preprint arXiv:2110.03895 (2021)
10. Jia, Q., Young, M., Xiao, Y., Cui, J., Liu, C., Rashid, P., Gehringer, E.: Insta-reviewer: A data-driven approach for generating instant feedback on students' project reports. International Educational Data Mining Society (2022)
11. Kakarla, S., Thomas, D., Lin, J., Gupta, S., Koedinger, K.R.: Using large language models to assess tutors' performance in reacting to students making math errors. arXiv preprint arXiv:2401.03238 (2024)
12. Kochmar, E., Vu, D.D., Belfer, R., Gupta, V., Serban, I.V., Pineau, J.: Automated personalized feedback improves learning gains in an intelligent tutoring system. In: International Conference on Artificial Intelligence in Education. pp. 140–146 (2020)
13. Kocmi, T., Federmann, C.: Large language models are state-of-the-art evaluators of translation quality. arXiv preprint arXiv:2302.14520 (2023)
14. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. Advances in neural information processing systems **35**, 22199–22213 (2022)
15. Lan, A.S., Vats, D., Waters, A.E., Baraniuk, R.G.: Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In: Proceedings of the ACM conference on learning@scale. pp. 167–176 (2015)
16. Lee, H., Phatale, S., Mansoor, H., Lu, K., Mesnard, T., Bishop, C., Carbune, V., Rastogi, A.: Rlaif: Scaling reinforcement learning from human feedback with ai feedback. arXiv preprint arXiv:2309.00267 (2023)
17. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004)
18. Liu, N., Sonkar, S., Wang, Z., Woodhead, S., Baraniuk, R.G.: Novice learner and expert tutor: Evaluating math reasoning abilities of large language models with misconceptions. arXiv preprint arXiv:2310.02439 (2023)
19. Liu, N., Wang, Z., Baraniuk, R., Lan, A.: Open-ended knowledge tracing for computer science education. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 3849–3862 (2022)
20. McNichols, H., Feng, W., Lee, J., Scarlatos, A., Smith, D., Woodhead, S., Lan, A.: Automated distractor and feedback generation for math multiple-choice questions via in-context learning. NeurIPS'23 Workshop on Generative AI for Education (2023)
21. McNichols, H., Zhang, M., Lan, A.: Algebra error classification with large language models. In: International Conference on Artificial Intelligence in Education. pp. 365–376 (2023)
22. Naismith, B., Mulcaire, P., Burstein, J.: Automated evaluation of written discourse coherence using GPT-4. In: Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023). pp. 394–403. Association for Computational Linguistics, Toronto, Canada (Jul 2023)

23. Nguyen, H.A., Stec, H., Hou, X., Di, S., McLaren, B.M.: Evaluating chatgpt's decimal skills and feedback generation in a digital learning game. In: Responsive and Sustainable Educational Futures. pp. 278–293 (2023)
24. Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C.D., Finn, C.: Direct preference optimization: Your language model is secretly a reward model (2023)
25. Razzaq, R., Ostrow, K.S., Heffernan, N.T.: Effect of immediate feedback on math achievement at the high school level. In: International Conference on Artificial Intelligence in Education. pp. 263–267. Springer (2020)
26. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019)
27. Robinson, J.D., Chuang, C.Y., Sra, S., Jegelka, S.: Contrastive learning with hard negative samples. In: International Conference on Learning Representations (2021)
28. Shinn, N., Cassano, F., Labash, B., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning. arXiv preprint arXiv:2303.11366 **14** (2023)
29. Singh, R., Gulwani, S., Solar-Lezama, A.: Automated feedback generation for introductory programming assignments. In: Proceedings of the 34th ACM SIGPLAN conference on Programming language design and implementation. pp. 15–26 (2013)
30. Song, D., Lee, W., Oh, H.: Context-aware and data-driven feedback generation for programming assignments. In: Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. pp. 328–340 (2021)
31. Sonkar, S., Le, M., Chen, X., Liu, N., Mallick, D.B., Baraniuk, R.G.: Code soliloquies for accurate calculations in large language models. arXiv preprint arXiv:2309.12161 (2023)
32. Steiss, J., Tate, T., Graham, S., Cruz, J., Hebert, M., Wang, J., Moon, Y., Tseng, W., et al.: Comparing the quality of human and chatgpt feedback on students' writing (2023)
33. Sun, K.L.: Brief report: The role of mathematics teaching in fostering student growth mindset. Journal for Research in Mathematics Education **49**(3), 330–335 (2018)
34. Touvron, H., Others: Llama 2: Open foundation and fine-tuned chat models (2023)
35. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
36. Zhang, M., Baral, S., Heffernan, N., Lan, A.: Automatic short math answer grading via in-context meta-learning. International Educational Data Mining Society (2022)
37. Zhang, M., Wang, Z., Baraniuk, R., Lan, A.: Math operation embeddings for open-ended solution analysis and feedback. International Educational Data Mining Society (2021)
38. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. In: International Conference on Learning Representations (2020)
39. Ziegler, D.M., Stiennon, N., Wu, J., Brown, T.B., Radford, A., Amodei, D., Christiano, P., Irving, G.: Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593 (2019)