

# Stepwise Verification and Remediation of Student Reasoning Errors with Large Language Model Tutors

Nico Daheim<sup>\*1</sup> Jakub Macina<sup>\*2,3</sup>  
Manu Kapur<sup>4</sup> Iryna Gurevych<sup>1</sup> Mrinmaya Sachan<sup>2</sup>


<sup>1</sup>Ubiquitous Knowledge Processing Lab (UKP Lab), Department of Computer Science and Hessian Center for AI (hessian.AI), TU Darmstadt

<sup>2</sup>Department of Computer Science, ETH Zurich <sup>3</sup>ETH AI Center

<sup>4</sup>Professorship for Learning Sciences and Higher Education, ETH Zurich

## Abstract

Large language models (LLMs) present an opportunity to scale high-quality personalized education to all. A promising approach towards this means is to build dialog tutoring models that scaffold students' problem-solving. However, even though existing LLMs perform well in solving reasoning questions, they struggle to precisely detect student's errors and tailor their feedback to these errors. Inspired by real-world teaching practice where teachers identify student errors and customize their response based on them, we focus on verifying student solutions and show how grounding to such verification improves the overall quality of tutor response generation. We collect a dataset of 1K stepwise math reasoning chains with the first error step annotated by teachers. We show empirically that finding the mistake in a student solution is challenging for current models. We propose and evaluate several verifiers for detecting these errors. Using both automatic and human evaluation we show that the student solution verifiers steer the generation model towards highly targeted responses to student errors which are more often correct with less hallucinations compared to existing baselines.

 <https://github.com/eth-lre/verify-then-generate>

## 1 Introduction

The field of dialog tutoring aims to build systems that can teach students by holding a conversation with them (Wollny et al., 2021; Jurenka et al., 2024). Dialog tutors hold the potential to make personalized teaching available to learners anywhere anytime. The increasing capabilities of LLMs have brought renewed hope to this field (Thoppilan et al., 2022; Jurenka et al., 2024). However, real-time tutoring is quite complex, and human teachers bring various intricate capabilities when teaching,

such as identifying student errors in problem solving, picking a pedagogical strategy, and communicating it (Bommasani et al., 2021; Wang et al., 2024b). The same requirements hold for dialog tutoring models which need all these abilities to be effective.

Yet, although research on effective human tutors shows they perform these steps sequentially by first reasoning about the error, then picking a strategy, and then responding (Lepper and Woolverton, 2002), many tutoring models perform all of them in one forward pass. Recent studies (Macina et al., 2023b,a) have shown that this can lead to several deficiencies that can be detrimental to student learning, for example, in math tutoring. Despite impressive performance on math reasoning benchmarks (Cobbe et al., 2021; Hendrycks et al., 2021), dialog tutors often generate hallucinated outputs and present erroneous information to students, for example, because they assess an incorrect solution as correct. We show an example of this in Figure 1.

In this paper, we alleviate this problem by decoupling the verification of student solutions from response generation with a modular approach. As opposed to the common approach, the model does not directly generate the tutor response from the students' utterances, whereby solution assessment is done implicitly in the models' activations, but rather receives the output of an additional verification model that assesses solutions and can therefore also be more specialized. We hypothesize that this increases the correctness of the model as well as makes the response more targeted to the error because the response generation module is already aware of the exact student error. Furthermore, this architecture more closely mimics human tutors.

To test our approach and train verifiers, we collect a dataset of ca. 1k student solutions and their stepwise reasoning chains in the domain of multi-step math problem-solving, which we released publicly. This dataset augments the math dialog tu-

<sup>\*</sup>Equal contribution.

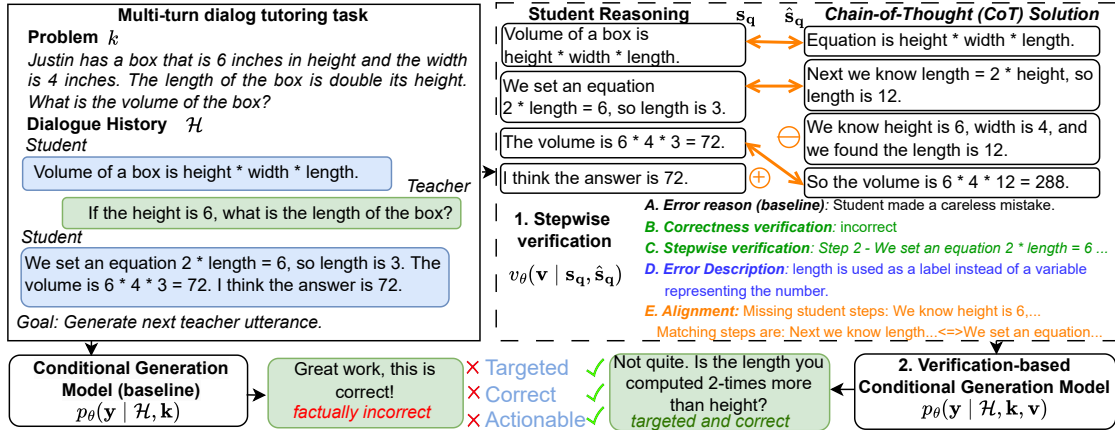


Figure 1: Directly generating a tutor response based on the conversation history can lead to hallucinations (bottom left). To alleviate this, we split this process into two sequential tasks (right): 1) A model identifies the student’s mistake. 2) A different response generation model communicates the identified mistake. We use different verifiers: providing the **Error Reason** (Wang et al., 2024b), **Classification-based Verification**, providing a more detailed **Error Description** and a **Step Alignment** of student and reference solution. Especially the latter two reduce hallucinations and make tutor models more targeted at the student error when verification and generation are combined (Section 6).

toring corpus MathDial (Macina et al., 2023a), which we use for evaluating dialog tutoring models, by teacher-annotated verifications of the first erroneous step in the student solution (Section 4).

We propose three verification approaches based on prompting and finetuning language models. Besides a simple classification-based approach for verification, we also generate a textual verification and notably align student solution steps to steps of a reference solution (Section 3.1) to verify the student solution. We find that using our data for finetuning helps smaller LLMs surpass prompted state-of-the-art LLMs. Furthermore, incorporating the verification output in the response generation step (Section 3.2) clearly improves their performance in terms of both extensive automatic (Section 6) and human evaluation using real teachers (Section 6.3): the generated responses are more targeted to the exact student error, there are less hallucinations, and there is more actionable scaffolding feedback for the student. In general, we find that such improvements are much stronger when the verification output is correct (Section 7.1) indicating a large potential for the community to improve dialog tutors by adding verifiers.

## 2 Background & Related Work

**Problem formulation** Dialog tutoring systems aim to tutor students by holding a conversation with them. Formally, the goal of dialog tutoring is to continue a tutoring dialog consisting of a sequence

of  $T-1$  turns  $\mathcal{H} := (\mathbf{u}_1, \dots, \mathbf{u}_{T-1})$  taken by either student or teacher. Each  $\mathbf{u}_t \in \mathcal{V}^*$  is constructed from a fixed model vocabulary  $\mathcal{V}$ . Continuation then means generating a new utterance  $u_T \in \mathcal{V}^*$ . Often there is also background knowledge that is either required or helpful to tutor a certain concept, such as grammar rules (Stasaski et al., 2020), or textbook knowledge (Wang et al., 2024a), and can be used to ground  $\mathbf{u}_T$ . In this work, we deal with teaching math word problem-solving and therefore use textual knowledge  $\mathbf{k} \in \mathcal{V}^*$ , for instance, the math word problem and background knowledge like math rules or a reference solution.

**Rule-based dialog tutoring** Tutoring systems should guide students to solving a problem (e.g. an MWP or translation of a phrase) not by telling the solution outright, but rather by using scaffolding techniques that give students space for guided exploration and self-correction. For example, the tutor might elicit the students’ thinking with Socratic questions (Shridhar et al., 2022) or sense-making questions that challenge their understanding of the problem (Reiser, 2004; Anghileri, 2006).

Capturing such intricate tutoring strategies is hard and requires teachers years to master. Due to this complexity, most previous intelligent tutoring systems were human-authored, notably the Auto-Tutor family (Nye et al., 2014), LISP tutor (Anderson et al., 1985) which uses a large set of rules to verify student programming solutions, or any systems built using example-tracing tutors which

requires enumerating all possible solutions or writing complex production rules (Alevin et al., 2016). However, scaling such human-authored systems can quickly explode in both complexity and human effort (Macina et al., 2023b). Due to this and rapid progress in language generation from learning large models based on large amounts of data, LLMs such as LearnLM (Jurenka et al., 2024) have recently become popular in favor of human-authored, rule-based systems.

**LLM-based dialog tutoring** Recently, LLM-based dialog tutors have become more popular as a means to tutor human students through a conversation with an LLM. For example, dialog tutoring has been proposed for second-language acquisition (Stasaski et al., 2020; Caines et al., 2020; Kwon et al., 2024), to answer questions in science (Chevalier et al., 2024), or to solve math word problems (MWP) (Macina et al., 2023a; Pal Chowdhury et al., 2024) To train such models, both human-collected data (Stasaski et al., 2020; Caines et al., 2022; Macina et al., 2023a) and synthetic data (Sonkar et al., 2023; Wang et al., 2024a), as well as a combination of both (Jurenka et al., 2024) can be used. Using these, a simple approach is training a model to generate the teacher response directly by pairing the following model, parameterized by weights  $\theta$ , with a decoding algorithm, such as beam search or greedy decoding:

$$p_{\theta}(y \mid \mathcal{H}, \mathbf{k}) = \prod_{i=1}^{|y|} p_{\theta}(y_i \mid y_{<i}, \mathcal{H}, \mathbf{k}). \quad (1)$$

This model is straightforward to implement and learn from data but prior work has shown that it suffers from generating factually incorrect outputs (Macina et al., 2023a). Therefore, in this work we break down response generation into two steps: verification, where it is first assessed whether the student solution is correct, and generation.

**Verification** Verification is challenging in its own right and has recently been tackled for general reasoning problems. For example, ROSCOE (Golovneva et al., 2023) presents unsupervised metrics to assess the correctness of a models’ chain-of-thought (CoT) reasoning, and (Jacovi et al., 2024) evaluate open-domain question answering for logical errors. The outputs of verifiers have subsequently also been used for self-refinement of LLMs (Madaan et al., 2023; Shinn et al., 2024) and

also allow targeted feedback for the training of student LLMs with teacher LLMs (Saha et al., 2023). Closely related to our work (Wang et al., 2024b) define broad error categories, such as miscalculation, to understand the cause of incorrect reasoning by students and condition on it to generate teacher responses. We compare to this baseline and call it **Error Reason**.

### 3 Verification-based Response Generation

We first introduce the task of verification and different verifiers in Section 3.1. Afterwards, in Section 3.2, we combine verification and response generation for modular tutor response generation.

#### 3.1 Verification

We deal with the verification of student solutions to a given math word problem  $\mathbf{q} \in \mathcal{V}^*$ . The solutions can be described by a sequence of substep solutions  $\mathbf{s}_{\mathbf{q}} = \{\mathbf{s}_1, \dots, \mathbf{s}_N\}$ , where each  $\mathbf{s}_n \in \mathcal{V}^*$  and  $\mathbf{s}_N$  is the final solution. Usually,  $\mathbf{s}_{\mathbf{q}}$  is described by the student in one of the student utterances  $\mathbf{u}_t$ . The task of the model is to assess whether  $\mathbf{s}_N$  is the correct solution to  $\mathbf{q}$  and if not, potentially, to identify which step  $\mathbf{s}_n$  caused the error. Oftentimes, this can be done by comparing to a reference solution  $\widehat{\mathbf{s}}_{\mathbf{q}} = \{\widehat{\mathbf{s}}_1, \dots, \widehat{\mathbf{s}}_M\}$  that is either given or model-generated and might differ in length. All verifiers which we discuss next can then be described by a learned function  $v_{\theta'}(\mathbf{v} \mid \mathbf{s}_{\mathbf{q}}, \widehat{\mathbf{s}}_{\mathbf{q}})$ , usually an LLM. Here,  $\mathbf{v}$  is the verification output and  $\widehat{\mathbf{s}}_{\mathbf{q}}$  may be an empty string if no reference solution is given. In the following, we introduce different verifiers.

**Classification-based Verification** A comparably simple approach to verification is classifying whether the whole student solution  $\mathbf{s}_{\mathbf{q}}$  is correct using a binary classifier. We call this **Overall Verification**. Similarly, identifying the first error step  $\mathbf{s}_n$  can be framed as multi-class classification based on all student solution steps with labels  $\{0, \dots, N\}$ , where 0 means no mistake. We call this **Stepwise Verification**. Alternatively, Stepwise Verification (iterative) can be framed as a binary classification just for each step  $\mathbf{s}_n$  whether it is correct or not. The first error step is the first step classified as an error.

**Error Description** While conceptually easy, classification-based approaches locate the first error without explaining the exact issue. Therefore, we propose to use an LLM to directly describe

---

**Algorithm 1** Modified Needleman-Wunsch.

---

**Require:** Solution attempt  $\mathbf{s}_q = \{s_1, \dots, s_N\}$   
**Require:** Reference solution  $\widehat{\mathbf{s}}_q = \{\widehat{s}_1, \dots, \widehat{s}_M\}$   
**Require:** Gap utility  $c$ , similarity threshold  $t$ , penalty  $p$   
**Ensure:** Optimal alignment of  $\mathbf{s}_q$  and  $\widehat{\mathbf{s}}_q$

```
1:  $F \leftarrow \text{zeros\_matrix}(M + 1, N + 1)$   $\triangleright$  initialize
2:  $F[1 : M + 1, 0] \leftarrow [i \cdot c \text{ for } i \text{ in } 1 \dots M]$ 
3:  $F[0, 1 : N + 1] \leftarrow [i \cdot c \text{ for } i \text{ in } 1 \dots N]$ 
4: for  $i \leftarrow 1$  to  $M$  do
5:    $\mathbf{e}_{\widehat{s}_m} \leftarrow \text{embed}(\widehat{s}_m)$ 
6:   for  $j \leftarrow 1$  to  $N$  do
7:      $\mathbf{e}_{s_n} \leftarrow \text{embed}(s_n)$ 
8:      $F[i, j] \leftarrow \text{cosine\_similarity}(\mathbf{e}_{\widehat{s}_m}, \mathbf{e}_{s_n})$ 
9:     if  $F[i, j] \geq t$  then  $\triangleright$  exact match
10:       $F[i, j] \leftarrow F[i - 1, j - 1] + F[i, j]$ 
11:     else  $\triangleright$  near match or gap
12:       $F[i, j] \leftarrow \max(F[i - 1, j - 1] - p + F[i, j], F[i - 1, j] + c, F[i, j - 1] + c)$ 
13:     end if
14:   end for
15: end for
16:  $\mathbf{a} = \{(\mathbf{a}_1, \widehat{\mathbf{a}}_1), \dots, (\mathbf{a}_L, \widehat{\mathbf{a}}_L)\} \leftarrow \text{backtrack}(F, \mathbf{s}_q, \widehat{\mathbf{s}}_q)$ 
17: return Globally-optimal alignment  $\mathbf{a}$ 
```

---

the error, and the concrete first error step, in a textual format, and call this **Error Description**. We prompt the LLM with the prompt outlined in Appendix G. In comparison to Wang et al. (2024b), this error description is allowed to be free-form and does not map to predefined error types. This makes our method more flexible and easy to apply to new domains, especially if these are captured in the LLM’s pretraining data. The LLM-generated error step description can then be passed to a tutor response generation model.

**(Step) Alignment** As our third verification approach, we align the steps in the student’s solution with a reference solution, and compare the steps in the student and reference solution to localize errors. We call this approach **Step Alignment**. As the order of steps in the solutions matters, a greedy algorithm that finds the most similar steps across the two solutions is insufficient. Thus, we frame verification as a sequence alignment problem. Intuitively, our goal is to find an alignment, where the aligned steps are as similar as possible while allowing missing and additional steps in the student solution to capture all error types.

The input to the alignment algorithm is the student solution  $\mathbf{s}_q$  with  $N$  steps and the reference solution  $\widehat{\mathbf{s}}_q$  with  $M$  steps. Note that here we are aligning solution steps which can be long strings. This is different from other sequence alignment problems in NLP, where typically tokens are aligned (Paolini et al., 2021, inter alia). The output is a sequence of tuples  $\{(\mathbf{a}_1, \widehat{\mathbf{a}}_1), \dots, (\mathbf{a}_L, \widehat{\mathbf{a}}_L)\}$  of

length  $L$ , where  $(\mathbf{a}_l, \widehat{\mathbf{a}}_l)$  means that student solution step  $\mathbf{a}_l$  is aligned to reference solution step  $\widehat{\mathbf{a}}_l$ . Each  $\mathbf{a}_l$  and  $\widehat{\mathbf{a}}_l$  can be either a step of  $\mathbf{s}_q$  and  $\widehat{\mathbf{s}}_q$ , respectively, or a special symbol  $\circ$ . Here,  $\mathbf{a}_l = \circ$  indicates a missing step in the student solution ( $-$ ) and  $\widehat{\mathbf{a}}_l = \circ$  indicates an additional step ( $+$ ).

In our implementation, we use the Needleman-Wunsch (NW) algorithm (Needleman and Wunsch, 1970) as it guarantees an optimal alignment with respect to a chosen utility function. We modify the algorithm for semantic sequence alignment and use sentence embeddings (Reimers and Gurevych, 2019) to measure the similarity between steps. We detail our adaptation of the NW algorithm in Algorithm 1 and describe each step in the following.

The NW algorithm iterates over all possible pairs of substeps from  $\mathbf{s}_q$  and  $\widehat{\mathbf{s}}_q$  by constructing a grid over them and calculating a utility for each pair in the grid. The path through this grid from beginning to end with highest utility is then chosen as the alignment. Since each substep is a string, we use semantic string similarity measured by the cosine similarity of the contextual embeddings of the substeps as utility. In our experiments, this performed better than just matching the final numerical solution of the substeps which can struggle to align incorrect steps (cf. Section 7.1 for results and a comparison of embedding models).

We distinguish three cases. If the similarity of a pair is at least as large as a threshold  $t$ , we count it as an exact match and add the pairs’ similarity to the similarity of its predecessors. In such cases, we are certain that the student and reference solution steps match. In some cases, we might be less certain and would ideally reflect the uncertainty in reduced utility. If the similarity is smaller than  $t$ , we still count pairs as a near match if subtracting a penalty (that indicates non-exact match) leads to a larger value than using a predefined gap utility  $c$ . In this case, we add the similarity to the predecessor similarities but discount it by the penalty  $p$  (in our case set to -1). Otherwise, the gap utility  $c$  is added to either a pair of the previous student and current reference solution step, or a pair of the current student and previous reference solution step. This enables missing and additional steps in the student solution but how often they appear can be controlled via the hyperparameter  $c$ . Altogether, this forms a similarity matrix  $F$  of size  $(N + 1) \times (M + 1)$ . The alignment is finally found by backtracking (moving only to adjacent entries with each step) from entry  $F_{N+1, M+1}$ .

Similar to the classification-based approach, the alignment output can not directly be used in a response generation model but has to be converted to a formatted verification output string. For this, we use a preformatted template which is shown in Appendix D. The template groups together the missing, additional and matching steps to produce  $\mathbf{v}$  from the alignment produced by the NW algorithm. The alignment approach can be used with any utility function and therefore can be adapted to a variety of domains.

### 3.2 Response Generation

Direct generation of tutor responses can be challenging because one model has to reason over the student solution, pick a teaching strategy, and generate a response in one step. This has been shown to produce hallucinations (Macina et al., 2023a). We tackle this by incorporating an additional verification step that informs the response generation model, as previously discussed. Our aim is to split the task into two less complex tasks which should reduce errors if each task can be performed well enough and has been shown to reduce hallucinations in document-grounded dialog (Adolphs et al., 2022) and question answering (Press et al., 2023).

The verifier and response generation model are combined in a two-stage approach. First, the verifier outputs a verification  $\mathbf{v}$  of the student solution  $\mathbf{s}_q$  based on a reference solution  $\hat{\mathbf{s}}_q$ . Then, the response generation model is conditioned on  $\mathbf{v}$ , the dialog history  $\mathcal{H}$ , and background knowledge  $\mathbf{k}$ . In our work,  $\mathbf{k}$  consists of the student solution  $\mathbf{s}_q$ , optionally the reference solution  $\hat{\mathbf{s}}_q$ , and the math word problem  $\mathbf{q}$ . If  $v_\theta$  is a distribution over verification labels, the overall model is:

$$p(\mathbf{y}, \mathbf{v} \mid \mathcal{H}, \mathbf{k}) = \underbrace{v_{\theta'}(\mathbf{v} \mid \mathbf{s}_q, \hat{\mathbf{s}}_q)}_{\text{verification}} \cdot \underbrace{p_{\theta}(\mathbf{y} \mid \mathcal{H}, \mathbf{k}, \mathbf{v})}_{\text{generation}} \quad (2)$$

The full model provides us with a verification output and the generated response which makes the internal reasoning of the tutor model in terms of student errors more explicit and controllable.

## 4 Data Collection

We propose and evaluate various verifiers in this work. Since some of them require training data and to evaluate their performance, we collect a dataset of 1,002 human-produced verification outputs to train and evaluate them. This is similar in size to a

related corpus (Jacovi et al., 2024). In this section, we describe the annotation task and data collection.

**Incorrect Student Solutions Source** Our work extends MathDial (Macina et al., 2023a) by having teachers annotate incorrect student solutions from the dataset with their first error step. There, these incorrect student solutions were used to condition a student model (INSTRUCTGPT) to generate responses in a dialogue with a human teacher. MathDial uses names and genders randomly sampled from a diverse set of cultural and ethnical backgrounds and reasoning errors sourced from learning sciences literature during data collection.

Specifically, the problems in MathDial are based on the GSM8k (Cobbe et al., 2021) dataset of multi-step math word problems. In MathDial, the reasoning chains are generated using a 2-shot CoT prompt with GPT-3.5-TURBO, and temperature sampling ( $T = 0.7$ ) is used to get multiple reasoning paths ( $n = 50$ ). Finally, the most common incorrect solution is chosen. Subsequently, their student model is prompted to respond to a human teacher as a student who tries to solve a problem with a particular incorrect solution.

To not skew our dataset to errors, we balance it with rephrased reference solutions from the student model. We reproduce the student model prompt from MathDial to generate student responses using the reference solutions. All reference solutions and student responses with incorrect solutions are part of the dataset. Details are in Appendix A.

**Student Solution Annotation** The objective of the annotation is to mark the exact step of the first error in the student solution. We do not annotate error steps after the first one to decrease ambiguity, as they frequently stem from the first error. We recruit teachers through Prolific<sup>1</sup>, who first read the problem and then mark the precise step of the first error in the student solution. Teachers can access the reference solution to reduce task complexity. Details of the task, the user interface, and examples of collected data are in Appendix A. To compute agreement, 10% of the samples are annotated by one additional annotator with an inter-rater reliability of Cohen’s  $\kappa = 0.75$  indicating substantial agreement (Cohen, 1960). We show the distribution of incorrect student solution steps in Figure 3.

<sup>1</sup><https://www.prolific.com>

## 5 Experiments

We evaluate different verifiers on our dataset and use them to inform response generation models to improve their correctness. Since we extend MathDial with additional annotations we use MathDial dialogues for evaluating tutor response generation. Besides math problem and student solution in a dialog, we either use a model-generated CoT reference solution if marked by “solution” or no reference solution as input to the models. Next, we detail metrics and models.

### 5.1 Metrics

For teacher response generation, we evaluate the generated output  $\mathbf{u}_T$  of each model by comparing it to a human-annotated response  $\hat{\mathbf{u}}_T$  from MathDial. We report standard text generation metrics: the sacrebleu (Post, 2018) implementation of BLEU (sBLEU) to measure word overlap and BERTScore (Zhang et al., 2020) (BF1, using the *all-MiniLM-L6-v2* checkpoint) to measure semantic similarity. Moreover, we report the knowledge F1 (KF1) score with respect to the grounding information (correct solution in the case of MathDial) which has been used as a proxy for faithfulness in prior work (Daheim et al., 2024). Similar to (Zheng et al., 2024; Jurenka et al., 2024), we also prompt LLAMA3-70B and use it complementary to human evaluation (the same task and instructions are used in both) to assess how *targeted*, *correct*, and *actionable* a response is. Details about the LLM-based evaluation are found in Appendix E.

### 5.2 Models

For both verification and response generation, we use different prompted or finetuned models. For verification, we compare the closed-source model GPT-3.5 to the open models LLAMA2 and LLAMA3. For the latter, we prompt the 70B version of the models and finetune LLAMA2-7B using LoRA. For response generation, we evaluate prompting GPT-3.5 and finetuning the encoder-decoder model Flan-T5 with 3B parameters. We finetune Flan-T5 again using LoRA for both the direct modeling and verify-then-generate approach.

## 6 Results

We first show the performance of different verification models in Section 6.1 and then use verification models in response generation in Section 6.2.

Model	Overall Verification			Stepwise
	Corr. F1	Err. F1	F1	micro F1
<b>Few-shot</b>				
GPT3.5	0.66	0.52	0.59	0.42
+ solution	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>	0.61
Llama2-70B	0.69	0.38	0.54	0.17
+ solution	0.78	0.59	0.68	0.48
Llama3-70B	0.74	0.58	0.66	<b>0.56</b>
+ solution	0.90	0.87	0.89	<b>0.70</b>
<b>Finetuned</b>				
Llama2-7B	<b>0.89</b>	0.67	0.78	0.20
+ solution	0.81	<b>0.80</b>	<b>0.80</b>	0.28

Table 1: Verifying student solutions can be challenging even for strong LLMs. Models are worse at verifying erroneous responses (Err. F1) than correct responses (Corr. F1). Providing a reference solution improves all models. Finetuning using our data can make models more robust when no such solution is present and make small models outperform larger prompted ones.

### 6.1 Verification

In this section, we benchmark LLMs on their ability to evaluate the correctness of student solutions using the **Overall Verification** and **Stepwise Verification** approaches from Section 3.1. For **Stepwise Verification** we use the multi-class classification approach, because it performed better than iterative classification in our experiments. A comparison is found in Appendix B. We measure the F1 score (balanced dataset), in particular, micro F1 for **Stepwise Verification** (imbalanced dataset, see Figure 3). We find in Table 1 that **Overall Verification** can be challenging even for state-of-the-art LLMs. All prompted models show comparably low performance when prompted without a reference solution and especially struggle with identifying incorrect responses. Providing a reference solution improves results significantly. However, for **Stepwise Verification** even the reference solution does not improve micro F1 beyond 0.70. This result is consistent with expert educator-based assessment (Yen and Hsu, 2023) and LLM self-correction results (Huang et al., 2024; Tyen et al., 2024).

Interestingly, our dataset can be used effectively for finetuning. Even a smaller LLAMA2-7B model can outperform larger prompted models on **Overall Verification**, especially when no solution is provided. Potentially, the additional finetuning steps make it easier for the model to also solve the problem before verification. The finetuned **Stepwise Verification** model outperforms its larger prompted counterpart LLAMA2-70B when no solution is

Model	Variant	Automatic Metrics			LLM Judge (%)		
		sBLEU	KF1	BF1	Targeted	Correct	Actionable
-	Human	100.0	100.0	100.0	27	82	87
<b>Few-shot</b>							
GPT-3.5	Baseline	2.0	27.0	51.2	29	37	27
	<b>Error Reason</b>	1.5	22.5	46.7	34	40	<b>56</b>
	<b>Error Description</b>	2.8	<b>30.3</b>	52.6	<b>62</b>	<b>66</b>	45
	<b>Step Alignment</b>	2.3	29.8	<b>53.3</b>	42	61	26
<b>Finetuned</b>							
Flan-T5-3B	Baseline	2.6	<b>27.6</b>	<b>56.0</b>	1	89	76
	<b>Error Description</b>	<b>3.0</b>	26.7	<b>56.0</b>	<b>2</b>	<b>92</b>	<b>84</b>

Table 2: Adding an additional verification stage to ground tutor response generation models leads to responses that are more targeted at the student error, less frequently hallucinated, and more actionable for the student, both for finetuned and prompted models. Proving a textual **Error Description** of the student solution performs better than **Step Alignment** of student and reference solution, as well as providing a shorter **Error Reason**.

provided. Results for finetuning show a ten-fold cross-validation. Further details are in Appendix H.

## 6.2 Response Generation

Next, we show in Table 2 that combining verification and tutor response generation models can improve the quality of the generated responses. We compare the **Error Description** and **Step Alignment** verifiers to direct response generation and using the **Error Reason** (Wang et al., 2024b). There, the error is categorized into either: *guess*, *misinterpret*, *right-idea*, *imprecise*, *not sure*, or *careless*. We use a subset of MathDial, where the student describes their solution to the teacher in the dialog, and generate the following teacher utterance.

First, we prompt GPT-3.5 using the prompt templates from Section 3.1 for comparability. We find that providing only the **Error Reason** does not improve over the direct baseline in simpler automatic metrics (sBLEU, KF1, BF1) but only in terms of the LLM-based judging. Using the more detailed **Error Description** which provides the exact mistake of the student gives larger improvements, both in terms of automatic metrics and LLM-based judging. Similarly, we find **Step Alignment** to be helpful, but to provide less actionable responses. When finetuning with the **Error Description**, we obtain improvements over the finetuned baseline but they are smaller and do not hold for each metric.

Our qualitative analysis shows that both **Step Alignment** and **Error Description** result in responses that better localize the exact student error. For example, the baseline often assesses the solution wrongly or skips the first error step and instead asks for the solution of a later step. Examples

Model	Variant	target $\uparrow$	corr $\uparrow$	act $\uparrow$
<b>Verification</b>				
GPT3.5	<b>Error Description</b>	-	70.6	-
Llama3-70B	<b>Error Description</b>	-	82.4	-
<b>Response Generation</b>				
-	Human	35.0	45.0	42.5
<b>Few-shot</b>				
GPT3.5	Baseline	30.0	37.5	30.0
	<b>Error Reason</b>	27.5	22.5	37.5
	<b>Error Description</b>	<b>57.5</b>	<b>62.5</b>	<b>45.0</b>
	<b>Step Alignment</b>	<b>57.5</b>	<b>60.0</b>	27.5
<b>Finetuned</b>				
Flan-T5-3B	Baseline	7.5	20.0	25.0
	<b>Error Description</b>	<b>20.0</b>	<b>35.0</b>	<b>35.0</b>

Table 3: Human evaluation with four expert annotators shows that verification before generation improves along the targetedness, correctness, and actionability (without telling the solution) of responses. We find that **Error Description** works best and improves both prompted and finetuned models.

are shown in Table 10 and Table 11. Section 6.3 confirms our results by human evaluation.

## 6.3 Human Evaluation

We conduct a human evaluation using teachers as expert annotators. All annotators are recruited on Prolific after manual screening. We assess whether the generated responses are *targeted*, *correct*, and *actionable* without outright telling the solution. Annotators are blind to the model source. The exact questions are as follows. 1) (*Targeted (T)*) Does the Teacher point out the root cause of the student’s mistake? 2) (*Correctness (C)*) Is the Teacher’s response factually correct with respect to the reference solution? 3) (*Actionable (A)*) - Does the Teacher provide actionable steps to let the Student correct the mistake without giving away the full

answer? More details are in Appendix C.

Responses from 6 models and one human response from MathDial were annotated for a random set of 40 conversations. To compute inter-rater reliability, 9 conversations were annotated with at least 2 raters for each response. Cohen’s kappa is 0.21, 0.25, and 0.13 for targeted, correctness, and actionable. For **Error Description** correctness it is  $\kappa = 0.30$ . Next, we describe the results, first on verification and then for response generation.

**Verification** Annotators assess the **Error Description** as correct if the exact mistake of the student is found and incorrect when the model says that the solution is correct when it is not and vice versa, misses the step of the error, or is generic. Results in Table 3 show that LLAMA3-70B outperforms GPT-3.5 but also with 82.4% of the errors being found correctly.

**Response Generation** Next, in Table 3, we evaluate how *targeted*, *correct*, and *actionable* the responses generated by different models are. We find that providing the **Error Reason** improves over the baseline only in terms of how actionable responses are. We hypothesize that conditioning on only the reason is insufficient for a targeted response. **Error Description** and **Step Alignment** provide more information regarding the exact error and therefore improve strongly over the baseline in both targetedness and correctness. Using **Step Alignment** information also does not improve actionability but **Error Description** improves it. The same improvements also hold for using the **Error Description** for a finetuned model. All in all, we find strong evidence that using our verify-then-generate approach improves teacher response generation.

## 7 Ablation Studies

Next, we provide further ablations, first on the utility function used in the NW algorithm (Section 7.1) and then on the impact of verification before response generation based on its correctness and problem difficulty (Section 7.2).

### 7.1 Alignment

We compare different utility functions used for the NW **Step Alignment** algorithm in Table 4. For the comparison, 30 alignments between a student and reference solution were produced by humans and the accuracy of student solution step alignment is measured. As utility functions we use the cosine

<b>Step Alignment</b>	$t^*$	$c^*$	Accuracy(%)
Random	0.6	-0.2	43.6
Solution Match	0.6	-0.3	51.9
SBERT	0.8	-0.1	58.2
Roscoe	0.9	-0.1	61.4

Table 4: We compare different utility functions for **Step Alignment** with the Needleman-Wunsch algorithm based on 30 human-annotated alignments. Semantic-similarity-based utility function (SBERT, Roscoe) performs better than random utility or an indicator function of whether the numerical substep solutions match.

Model	<b>Error Desc.</b>	T	C	A
<b>Few-shot</b>				
GPT3.5	incorrect	50.0	<b>62.5</b>	<b>50.0</b>
	correct	<b>62.5</b>	<b>62.5</b>	41.7
Llama3-70B	incorrect	41.7	33.3	<b>25.0</b>
	correct	<b>82.1</b>	<b>71.4</b>	21.4
<b>Finetuned</b>				
Flan-T5-3B	incorrect	08.3	25.0	<b>41.7</b>
	correct	<b>25.0</b>	<b>39.3</b>	32.1

Table 5: We find that tutor responses are much more often correct and targeted if the **Error Description** is correct. Data from human evaluation.

similarity of Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) embeddings and embeddings from a model trained on Roscoe (Golovneva et al., 2023), as well as a random utility and an indicator function that is 1 when two substeps have the same numerical solution and 0 otherwise. Similarity threshold  $t$  and gap utility  $c$  are optimized via a hyperparameter grid search, as indicated by  $t^*$  and  $c^*$ . We find that cosine similarity works best and training on relevant math data finetuning in Roscoe further improves performance.

### 7.2 Verification

**Verification correctness is important** Table 5 shows how the correctness of the **Error Description** verification influences the response generated by the tutor model. The results are taken from human evaluation, as described in Section 6.3. Overall, we find that a correct **Error Description** leads to more targeted and correct responses. If the **Error Description** is wrong, responses tend to be less targeted and are often incorrect which can be detrimental to student learning, as students are taught wrong information. Nevertheless, the responses can also be less actionable if feedback is correct, potentially due to the model often telling students the solution



Model	Steps	T	C	A	Error Desc.
<b>Few-shot</b>					
GPT3.5	3	<b>62.5</b>	<b>75.0</b>	37.5	50.0
	4	61.1	55.6	<b>55.6</b>	<b>72.2</b>
	5	50.0	64.3	50.0	50.0
Llama3-70B	3	<b>87.5</b>	<b>75.0</b>	12.5	<b>87.5</b>
	4	72.2	72.2	<b>72.2</b>	72.2
	5	57.1	35.7	57.1	50.0
<b>Finetuned</b>					
Flan-T5-3B	3	12.5	25.0	25.0	<b>87.5</b>
	4	<b>27.8</b>	<b>50.0</b>	<b>50.0</b>	72.2
	5	14.3	21.4	14.3	50.0

Table 6: For prompted models, responses for problems with shorter solution lengths are more often correct and targeted, because such problems are likely less complex. For finetuned models we do not find this trend. More steps can decrease description correctness (Error Desc.). Data from human evaluation of Error Description.

or facts without scaffolding.

**Problem difficulty influences verification** Finally, we show in Table 6 that the performance of both verification and our verify-then-generate approach is heavily correlated with the number of reasoning steps that are used in the reference solution of a given math word problem. We use this as a proxy for problem difficulty. First of all, the performance of the LLAMA3 70B Error Description decreases with the number of steps. This is reflected in the decreased correctness and targetedness of the responses of the few-shot prompted LLAMA3 model. For GPT-3.5 we do not find a similar conclusion for the Error Description model but at least targetedness still decreases with the number of steps. For the finetuned model we do not see similar trends but instead find the best performance for problems with four steps, likely because these are more common in the training data.

## 8 Discussion

Student errors are key learning opportunities. Tutors should recognize them and precisely guide students with targeted feedback without telling the full solution. Our results show that even state-of-the-art LLMs can struggle with this but that adding a separate verification step can help. Still, while verification can make responses more targeted, they sometimes tend to be less actionable (cf. Table 5, for example). This suggests that there is an inherent trade-off between targeted and actionable responses that future work could explore further.

Related to this, just prompting models might

not be enough to elicit pedagogy if LLMs are optimized mainly for solving accuracy, as also reported by Jurenka et al. (2024). Rather, training on high-quality data, which still is hard to obtain, might be required. This is also supported by our results in Table 1, where smaller finetuned models can even outperform their larger counterparts.

High-quality data is also important for the reliable evaluation of dialog tutors. In particular, it is important to understand different fine-grained quality criteria, such as correct verification that the student’s solution is correct or identification of which step of the student’s solution is incorrect. Both are important prerequisites to understanding how effectively dialog tutors can foster learning. Future work can therefore also build on our dataset and evaluation protocols to improve the assessment of dialog tutors.

## 9 Conclusion

Motivated by effective teaching practice, we split the task of tutor response generation into two separate steps of verifying the student solution and generating a response. To evaluate our approach, we collect a dataset of around 1k teacher-annotated solutions to augment an existing math tutoring corpus. Our results show that splitting response generation into two steps can result in more targeted and correct responses that better scaffold human learning. We showcase this using both automatic evaluation and human evaluation annotated by teachers, both for prompted and finetuned models.

## 10 Limitations

**Focus on scaffolding problem-solving** The tutoring scenarios which are considered are centered around the student problem-solving stage. In this case, students have prior knowledge, mostly understand the learning topics and practice them. However, different learning scenarios such as direct instruction, building rapport with students, or open-ended discussions are not considered in this work.

Evaluating student solutions and responding appropriately to a student’s mistakes is inherently challenging, even for human teachers. Furthermore, teachers should ideally give adaptive feedback depending on the problem-solving strategy chosen by the student and treat different errors in different ways to uncover any misconceptions (Nye et al., 2014; Ross and Andreas, 2024). For example, in math, productive errors present important learning

opportunities for students to learn from them (Kapur, 2016; Shaughnessy et al., 2021; Sinha and Kapur, 2021), e.g. by teacher-guided self-correction or targeted instruction, while unproductive errors, such as numerical miscalculations, could be easily resolved using a calculator (Lepper and Woolverton, 2002).

### **Difficulty of obtaining student reasoning chains**

Model-generated reasoning chains might contain the same biases as human students (Opedal et al., 2024). On the other hand, there might be many additional differences from human student reasoning, e.g. students might not always stick to exact math notations or skip some steps in the explanations. However, because such data from students is sensitive, we work with model-generated reasoning solutions and responses.

**Focus on multi-step problems** Procedural or multi-step problems are the basis of most of the scientific disciplines, therefore we believe our approach should be general enough to work across any science subject, especially by including retrieval-augmented generation (RAG) from textbooks. However, it is still an open research question whether a similar solution would work for language learning or fact-based problems, and how models perform in languages other than English.

**Evaluation is teacher-centered and complemented with an LLM-judge** Future work should focus on student user studies with AI tutors. However, this requires careful experimental consideration and safety mechanisms. Moreover, assessment of the responses is done exclusively by teachers and therefore future work should consider running assessments of the responses by students.

**Potential biases in data collection** Our work bases on MathDial (Macina et al., 2023a). While MathDial aims to capture a variety of student personas and reasoning errors by prompting the LLM with names and genders sampled from various ethnic backgrounds and reasoning errors based on categories from learning sciences literature, our data may still exhibit biases. In particular, the diverse backgrounds in MathDial are promoted by prompting LLMs with names from culturally and ethnically diverse set. This, however, bears the risk of exacerbating biases in LLMs towards such groups and could also fail due to the LLM not capturing the personas’ cultural aspects well. Furthermore, the majority of annotators in MathDial

stem from western countries which could further bias the data.

## **11 Acknowledgements**

This research work has been funded by a Swiss National Science Foundation award (#201009), a Responsible AI grant by the Haslerstiftung, by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE. Nico Daheim acknowledges travel support from ELISE (GA no 951847). Jakub Macina acknowledges funding from the ETH AI Center Doctoral Fellowship, Asuera Stiftung, and the ETH Zurich Foundation. We thank Sankalan Pal Chowdhury, Kumar Shridhar, Shehzaad Dhuliawala, and Justus-Jonas Erker for valuable feedback and discussions.

## **12 Ethics Statement**

**Intended usage** The benefits of our dataset are in understanding and designing AI technology to assist teachers and students during the problem-solving stage. Most importantly, the goal of such systems is to not replace human teachers, but rather enhance their capabilities and make them focus on important and human aspects of teaching. We release the dataset under CC-BY-4.0 license<sup>2</sup> for further usage and exploration by the community. This also adheres to the licensing of MathDial, which we extend.

**Data Anonymization and Privacy** As the data in education are strictly confidential we obtained approval on the proposal<sup>3</sup> of the collection interface, questions and how long the data will be stored. All participants fill informed consent at the beginning of the annotation and may withdraw without reason at any time. We store only the necessary data and do not store any personally identifiable information. The collected data are stored anonymously and securely. Moreover, no student data are used in this work.

**Accessibility and Potential Misuse** Our work focus on addressing hallucinations of LLM Tutors and their generation of factually incorrect re-

<sup>2</sup><https://creativecommons.org/licenses/by/4.0/deed.en>

<sup>3</sup>The study was approved by the ETH Zurich Ethics Committee (IRB) under EK-2024-N-97.

sponses. This directly addresses one of the important aspects of responsible use of AI which does not spread incorrect information, especially in education. We encourage the community to work on this important topic by open-sourcing the dataset, the code for running the benchmarks, and the methods used in this work. These are primarily intended for research purposes. As with any AI technology, the methods and dataset could be misused. However, we believe by open-sourcing the work we inform about the risks and capabilities of the technology a wider research community which then leads to further improvements.

## References

- Leonard Adolphs, Kurt Shuster, Jack Urbanek, Arthur Szlam, and Jason Weston. 2022. [Reason first, then respond: Modular generation for knowledge-infused dialogue](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7112–7132, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Vincent Aleven, Bruce M McLaren, Jonathan Sewall, Martin Van Velsen, Octav Popescu, Sandra Demi, Michael Ringenber, and Kenneth R Koedinger. 2016. [Example-tracing tutors: Intelligent tutor development for non-programmers](#). *International Journal of Artificial Intelligence in Education*, 26:224–269.
- John R Anderson, C Franklin Boyle, and Brian J Reiser. 1985. [Intelligent tutoring systems](#). *Science*, 228(4698):456–462.
- Julia Anghileri. 2006. [Scaffolding practices that enhance mathematics learning](#). *Journal of Mathematics Teacher Education*, 9:33–52.
- Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khat-tab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Muniyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R’e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. [On the opportunities and risks of foundation models](#). *ArXiv*.
- Andrew Caines, Helen Yannakoudakis, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2022. The teacher-student chatroom corpus version 2: more lessons, new annotation, automatic detection of sequence shifts. In *Swedish Language Technology Conference and NLP4CALL*, pages 23–35.
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. [The teacher-student chatroom corpus](#). In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 10–20.
- Alexis Chevalier, Jiayi Geng, Alexander Wettig, Howard Chen, Sebastian Mizera, Toni Annala, Max Aragon, Arturo Rodriguez Fanlo, Simon Frieder, Simon Machado, Akshara Prabhakar, Ellie Thieu, Jiaachen T. Wang, Zirui Wang, Xindi Wu, Mengzhou Xia, Wenhan Xia, Jiatong Yu, Junjie Zhu, Zhiyong Ren, Sanjeev Arora, and Danqi Chen. 2024. [Language models as science tutors](#). In *Forty-first International Conference on Machine Learning*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*.
- Jacob Cohen. 1960. [A coefficient of agreement for nominal scales](#). *Educational and psychological measurement*, 20(1):37–46.
- Nico Daheim, Nouha Dziri, Mrinmaya Sachan, Iryna Gurevych, and Edoardo Ponti. 2024. [Elastic weight removal for faithful and abstractive dialogue generation](#). 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 7096–7112, Mexico City, Mexico. Association for Computational Linguistics.
- Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. [ROSCOE: A suite of metrics for scoring step-by-step reasoning](#). In

- The Eleventh International Conference on Learning Representations.*
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. [Large language models cannot self-correct reasoning yet](#). In *The Twelfth International Conference on Learning Representations*.
- Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roei Aharoni, and Mor Geva. 2024. [A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4615–4634, Bangkok, Thailand. Association for Computational Linguistics.
- Irina Jurenka, Markus Kunesch, Kevin R McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, Shubham Milind Phal, Katherine Hermann, Daniel Kasenberg, Avishkar Bhoopchand, et al. 2024. Towards responsible development of generative ai for education: An evaluation-driven approach. *arXiv preprint arXiv:2407.12687*.
- Manu Kapur. 2016. [Examining productive failure, productive success, unproductive failure, and unproductive success in learning](#). *Educational Psychologist*, 51(2):289–299.
- Soonwoo Kwon, Sojung Kim, Minju Park, Seunghyun Lee, and Kyuseok Kim. 2024. [BIPED: Pedagogically informed tutoring system for ESL education](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3389–3414, Bangkok, Thailand. Association for Computational Linguistics.
- Mark R. Lepper and Maria Woolverton. 2002. [Chapter 7 - the wisdom of practice: Lessons learned from the study of highly effective tutors](#). In Joshua Aronson, editor, *Improving Academic Achievement*, Educational Psychology, pages 135–158. Academic Press, San Diego.
- Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023a. [MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.
- Jakub Macina, Nico Daheim, Lingzhi Wang, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023b. [Opportunities and challenges in neural dialog tutoring](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2357–2372, Dubrovnik, Croatia. Association for Computational Linguistics.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Saul B Needleman and Christian D Wunsch. 1970. [A general method applicable to the search for similarities in the amino acid sequence of two proteins](#). *Journal of molecular biology*, 48(3):443–453.
- Benjamin D Nye, Arthur C Graesser, and Xiangen Hu. 2014. [Autotutor and family: A review of 17 years of natural language tutoring](#). *International Journal of Artificial Intelligence in Education*, 24:427–469.
- Andreas Opedal, Alessandro Stolfo, Haruki Shirakami, Ying Jiao, Ryan Cotterell, Bernhard Schölkopf, Abulhair Saparov, and Mrinmaya Sachan. 2024. [Do language models exhibit the same cognitive biases in problem solving as human learners?](#) In *Forty-first International Conference on Machine Learning*.
- 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.
- Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. [Structured prediction as translation between augmented natural languages](#). In *International Conference on Learning Representations*.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. [Measuring and narrowing the compositionality gap in language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.
- 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*. Association for Computational Linguistics.

- Brian J. Reiser. 2004. [Scaffolding complex learning: The mechanisms of structuring and problematizing student work](#). *Journal of the Learning Sciences*, 13(3):273–304.
- Alexis Ross and Jacob Andreas. 2024. [Toward in-context teaching: Adapting examples to students’ misconceptions](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13283–13310, Bangkok, Thailand. Association for Computational Linguistics.
- Swarnadeep Saha, Peter Hase, and Mohit Bansal. 2023. [Can language models teach? teacher explanations improve student performance via personalization](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Meghan Shaughnessy, Rosalie DeFino, Erin Pfaff, and Merrie Blunk. 2021. [I think i made a mistake: How do prospective teachers elicit the thinking of a student who has made a mistake?](#) *Journal of Mathematics Teacher Education*, 24:335–359.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. [Reflection: Language agents with verbal reinforcement learning](#). *Advances in Neural Information Processing Systems*, 36.
- Kumar Shridhar, Jakob Macina, Mennatallah El-Assady, Tanmay Sinha, Manu Kapur, and Mrinmaya Sachan. 2022. [Automatic generation of socratic subquestions for teaching math word problems](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4136–4149, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Tanmay Sinha and Manu Kapur. 2021. [When problem solving followed by instruction works: Evidence for productive failure](#). *Review of Educational Research*, 91(5):761–798.
- Shashank Sonkar, Naiming Liu, Debshila Mallick, and Richard Baraniuk. 2023. [Class: A design framework for building intelligent tutoring systems based on learning science principles](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1941–1961.
- Katherine Stasaski, Kimberly Kao, and Marti A. Hearst. 2020. [CIMA: A large open access dialogue dataset for tutoring](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. [Lamda: Language models for dialog applications](#). *arXiv preprint arXiv:2201.08239*.
- Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. 2024. [LLMs cannot find reasoning errors, but can correct them given the error location](#). In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13894–13908, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.
- Junling Wang, Jakob Macina, Nico Daheim, Sankalan Pal Chowdhury, and Mrinmaya Sachan. 2024a. [Book2Dial: Generating teacher student interactions from textbooks for cost-effective development of educational chatbots](#). In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9707–9731, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.
- Rose E. Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024b. [Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachler. 2021. [Are we there yet?-a systematic literature review on chatbots in education](#). *Frontiers in artificial intelligence*, 4.
- An-Zi Yen and Wei-Ling Hsu. 2023. [Three questions concerning the use of large language models to facilitate mathematics learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3055–3069, Singapore. Association for Computational Linguistics.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). In *International Conference on Learning Representations*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). *Advances in Neural Information Processing Systems*, 36.

**Study progress**  
20%

**Math Problem:**  
Bertha plays tennis. Every ten games, one of her tennis balls wears out. Every five games, she loses a ball. Every four games, she buys a canister of three balls. She started with two balls and gave one to her partner. After twenty games, how many tennis balls does she have?

**Correct Solution:**  
Bertha wears out  $20 / 10 = 2$  tennis balls in 20 games.  
She loses  $20 / 5 = 4$  tennis balls in 20 games.  
She buys canisters of  $3 * 20 / 4 = 15$  tennis balls in 20 games.  
Bertha started with 2 balls and gave 1 away, so she has  $2 + 15 - 2 - 4 - 1 = 10$  tennis balls after twenty games.  
10

Analyze the student answer first by comparing it to the correct solution:

**Student Solution:** (mark the exact step of the first error)

- After 20 games, Bertha has lost  $20/5 = 4$  balls.
- She has bought  $20/4 = 5$  canisters of balls.
- So she has bought  $5 * 3 = 15$  balls.
- She started with 2 and gave one away, so she had  $2 - 1 = 1$  ball left before playing 20 games.
- With the new balls she bought, she now has  $1 + 15 = 16$  balls.
- But she also lost 4, so she has  $16 - 4 = 12$  balls left after 20 games.

None of the above

When in doubt, prioritize category

Please describe the category of the error

Write your description here

- Reached correct solution but proceeded further
- Extra quantity or missing quantity
- Unit conversion error
- Calculation error easily solved by a calculator
- Missing / Wrong factual knowledge
- Misunderstanding of a question
- None of the above

Figure 2: User interface for annotating the step of the first error, their categorization, and description of the error.

### A Data Collection Details

The annotators are screened through Prolific to be teachers native in English with an overall acceptance rate of more than 98% and with at least 500 submissions. We paid a minimum of \$20 per hour. Annotators are from the US, Canada, and the UK, with a balanced gender ratio, and their age range is from 25 to 53 years. All annotators have K12 experience and on average they have 12 years of experience in teaching.

The annotators are first trained for the task with an interactive practice problem and then annotate student solutions. In one session one annotator performs 5 stepwise error verifications where they first pick the exact step with the error and then classify the error into 8 categories, with separate descriptions of the error for each category. We filter out all error descriptions not following the prescribed format to remove low-quality annotations.

The interface is shown in Figure 2. The categories are: *missing or incorrect factual knowledge*, *misunderstanding of the question*, *the reference solution reached but proceed further*, *missing quantity*, *extra quantity*, *unit conversion error*, *numerical calculation*, *other*.

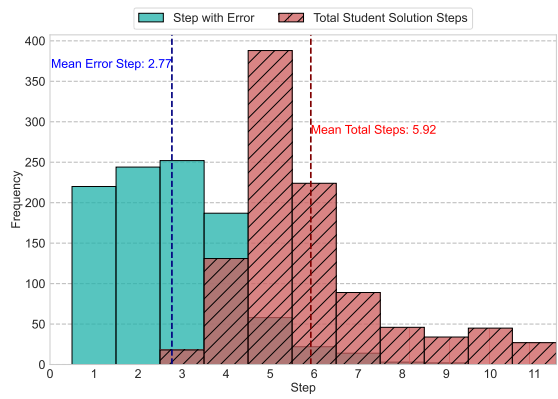


Figure 3: Dataset Distribution. The index of the step with the first error annotated by teachers and the total student solution steps.

### A.1 Dataset Details

The collected dataset is in English and from the domain of K12 math word problem-solving. Examples from the dataset are shown in Table 7. The dataset consists of 1002 data points with 612 unique math problems. The distribution of total student steps and the location of the first incorrect steps are shown in Figure 3. Notice the student solutions contain up to 11 steps with a mean of ca. 6 steps. The location of the first error ranges from 1 to 8 steps with majority of the errors between the first and third steps.

The incorrect student solutions and reference solutions are part of the MathDial dataset (Macina et al., 2023a). The prompt used to generate correct student responses to balance the dataset with correct student responses are based on the Student model from (Macina et al., 2023a).

### B Details of Overall Verification and Stepwise Verification

For **Stepwise Verification**, we compare multi-class classification and iterative approach on our dataset and the results are in Table 8. The iterative approach classifies each step  $s_n$  whether it is correct and therefore is more resource-intensive than multi-class classification. The multi-class classification directly predicts the label  $\{0, \dots, N\}$  where 0 represents the solution is correct. The results indicate no improvements (with the exception of Llama2-70B) by using the iterative approach and in the main paper we therefore report multi-class classification results.

Moreover, to confirm the quality of our collected dataset, we run the same models on the smaller and

**Problem:** Carl has been selling watermelons on the side of the road for \$3 each. This evening he went home with \$105 in profit and 18 watermelons. How many watermelons did he start out with this morning?

**Reference solution:** Carl sells his watermelons for \$3 each so today he sold  $\$105 / \$3$  per watermelon = 35 watermelons. He had 18 watermelons left over, so this morning he started with  $18 + 35 = 53$  watermelons.

**Incorrect solution:** Let's start by finding out how much money Carl made selling the 18 watermelons. Since he sold each watermelon for \$3, he made  $18 \times \$3 = \$54$ . We know that his total profit for the day was \$105, so he must have started with  $\$105 - \$54 = \$51$  worth of watermelons. Since each watermelon costs \$3, he must have started with  $51 / \$3 = 17$  watermelons.

**Student response:** I started by finding out how much money Carl made selling the 18 watermelons. Since he sold each watermelon for \$3, he made  $18 \times \$3 = \$54$ . Then I realized that his total profit for the day was \$105, so he must have started with  $\$105 - \$54 = \$51$  worth of watermelons. Since each watermelon costs \$3, I concluded that he must have started with  $51 / \$3 = 17$  watermelons.

**Error category:** Misunderstanding of a question

**Error description:** Carl did not sell 18 watermelons, but 18 watermelons are left unsold.

**Problem:** Dakota gets hit by a bus and has to spend 3 days in the hospital. The hospital charges her \$900/day for her bed, \$250/hour for two specialists to look at her 15 minutes each, and \$1800 for the ambulance ride. How much is Dakota's medical bill?

**Reference solution:** First find the total cost of the hospital bed by multiplying the cost per day by the number of days:  $\$900/\text{day} \times 3 \text{ days} = \$2700$ . Then figure out how many minutes total the two specialists looked at Dakota:  $15 \text{ minutes}/\text{specialist} \times 2 \text{ specialists} = 30 \text{ minutes}$  Then divide the number of minutes by the number of minutes per hour and multiply it by the specialists' hourly rate to find their total charge:  $30 \text{ minutes} / 60 \text{ minutes}/\text{hour} \times \$250/\text{hour} = \$125$ . Finally, add up the cost of the specialists, ambulance and bed to find Dakota's total bill:  $\$2700 + \$1800 + \$125 = \$4625$

**Incorrect solution:** The hospital charges Dakota  $\$900 \times 3 = \$2700$  for her bed. Each specialist charged her  $\$250/\text{hour} \times 2 = \$500$  for their 15 minutes each. So, Dakota was charged  $\$500 \times 2 = \$1000$  for the two specialists. Therefore, her medical bill is  $\$2700 + \$1000 + \$1800 = \$5500$

**Student response:** I started by calculating the cost of the bed, which was  $\$900 \times 3 \text{ days} = \$2700$ . Then I calculated the cost of the two specialists, which was  $\$250/\text{hour} \times 2 \text{ specialists} \times 15 \text{ minutes each} = \$500$ . Then I added all the costs together to get the total cost of  $\$2700 + \$1000 + \$1800 = \$5500$

**Error category:** Misunderstanding of a question

**Error description:** Student computes charges for a full hour of 2 specialists, not just 15 minutes as indicated in the question.

Table 7: Examples from the collected dataset. The annotated error lines are in red.

Model	micro F1
GPT3.5 multi-class	0.42
+ solution	0.61
GPT3.5 iterative	0.39
+ solution	0.36
Llama2-70B multi-class	0.17
+ solution	0.48
Llama2-70B iterative	0.36
+ solution	0.58
Llama3-70B multi-class	<b>0.56</b>
+ solution	<b>0.70</b>
Llama3-70B iterative	<b>0.56</b>
+ solution	0.58

Table 8: Results of two approaches for **Stepwise verification** and their micro F1 score. Multi-class classification directly predicts incorrect step  $N$ . On the other hand, the iterative approach iterates over each step and runs a binary prediction of whether the step is correct until the first incorrect step is found.

Model	Overall Verification			Stepwise micro F1
	Corr. F1	Err. F1	F1	
<b>Few-shot</b>				
GPT3.5	0.75	0.65	0.70	0.50
+ solution	<b>0.91</b>	<b>0.88</b>	<b>0.89</b>	0.63
Llama2-70B	0.07	0.62	0.34	0.11
+ solution	0.80	0.82	0.81	0.59

Table 9: Stepwise verification results on the small existing dataset from human evaluation of Roscoe (Golovneva et al., 2023). The Stepwise verification contains multi-class classification results. The results of the models are consistent with our dataset.

simpler Roscoe human evaluation set (Golovneva et al., 2023). The dataset is smaller and contains 105 correct and 95 incorrect solutions. The results are shown in Table 9 and the conclusions are identical to our dataset.

## C Guidelines for Human Evaluation

The user interface used in the human evaluation is shown in Figure 4 and Figure 5. All the annotators had to complete a training for the task where each of their responses was evaluated and the feedback was provided to them. We used the subset of the annotators from Appendix A with the same selection conditions and the same payment. Before evaluating the quality of responses, annotators are asked to analyze the math problem and the conversation and explain the student error in an open-ended text. To not bias their understanding of the student solu-

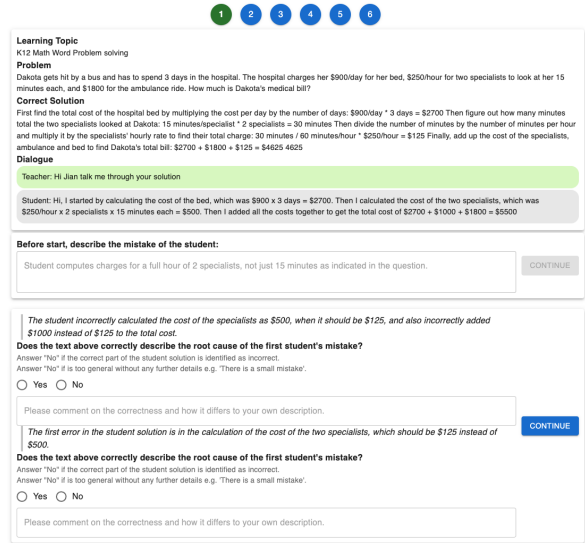


Figure 4: User interface for explaining the error of the student and evaluation of two error descriptions from models. Afterwards, annotators evaluate the quality of the model responses.

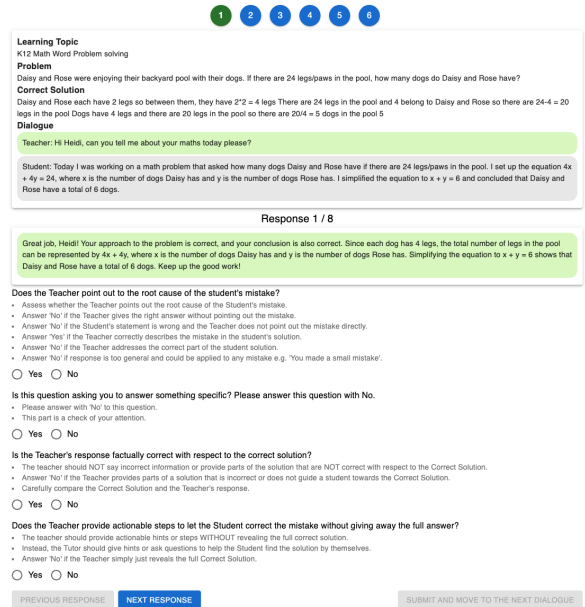


Figure 5: User interface for evaluation of the quality of the model responses. Some responses contain attention checks (second question in this case).



tion only subsequently the error descriptions from verifiers were annotated with their correctness using these instructions: Does the text above correctly describe the root cause of the first student’s mistake? Answer “No” if the correct part of the student solution is identified as incorrect. Answer “No” if it is too general without any further details e.g. ‘There is a small mistake’.

The exact wording of the annotation questions for evaluating the quality of responses is the following:

**Targeted** Does the Teacher point out to the root cause of the student’s mistake? Answer ‘No’ if the Teacher gives the right answer without pointing out the mistake. Answer ‘No’ if the Student’s statement is wrong and the Teacher does not point out the mistake directly. Answer ‘Yes’ if the Teacher correctly describes the mistake in the student’s solution. Answer ‘No’ if the Teacher addresses the correct part of the student solution. Answer ‘No’ if response is too general and could be applied to any mistake e.g. ‘You made a small mistake’.

**Correctness** Is the Teacher’s response factually correct with respect to the reference solution? The teacher should NOT say incorrect information or provide parts of the solution that are NOT correct with respect to the reference solution. Answer ‘No’ if the Teacher provides parts of a solution that is incorrect or does not guide a student towards the reference solution. Carefully compare the reference solution and the Teacher’s response.

**Actionable** Does the Teacher provide actionable steps to let the Student correct the mistake without giving away the full answer? The teacher should provide actionable hints or steps WITHOUT revealing the full reference solution. Instead, the Tutor should give hints or ask questions to help the Student find the solution by themselves. Answer ‘No’ if the Teacher simply just reveals the full reference solution.

## D Alignment Details

To find the best hyperparameters for the Alignment algorithm we run a grid search using values of similarity threshold  $t = [0.5, 0.6, 0.7, 0.8, 0.9, 0.95]$  and gap utilities  $c = [-0.1, -0.2, -0.3, -0.5, -0.7, -1.0, -1.2]$ . The best hyperparameters are reported in Table 4. The exact models which are used for semantic similarity are SBERT (*sentence-transformers/all-mpnet-base-v2*) and Roscoe (*facebook/roscoe-512-roberta-base*).

We use the template to transform the output of the algorithm into the textual prompt. In the template, all the steps from the student solution and reference solution are used. Furthermore, the utility of the alignment can be used to filter out student solutions that differ completely from reference solution which we leave for future work. The template is the following:

Missing steps in student solution: {missing steps}  
Unnecessary steps in the student solution: {unnecessary steps}  
Matching steps: {matching steps}

## E Details on LLM-based Evaluation

A response is targeted if it targets the students’ mistake, correct if it does not conflict with grounding information, and actionable if it provides the student with useful guidance to help the student progress in their solution attempt. In all cases, for each quality dimension, we provide the model with three examples (3-shots). We use the LLAMA3-70B<sup>4</sup> with temperature  $T = 0$  for reproducibility. The task description and the examples are the same as in the human evaluation for instructing the annotators described in Section 6.3. The prompt also includes the reference solution for more reliable judging (Zheng et al., 2024; Jurenka et al., 2024).

## F Qualitative examples

In this section, we show qualitative examples to better understand the behavior of verification and verification-based response generation. We first show examples for prompted models in Table 10 and then show examples for finetuned models in Table 11.

---

<sup>4</sup>meta-llama/Meta-Llama-3-70B-Instruct

## G Prompts

This section provides the exact prompts used in our work. First, we show the prompt used for the baseline, error description-based, and alignment-based response generation models in Fig. 6 and Fig. 9. Verification prompts for Error Description are in Fig. 8 and for Error Reason in Fig. 7. Classification-based verifications prompts are in Fig. 10 and Fig. 11. The prompt with 5 examples for the CoT solution generation is in Fig. 12. Then, we show the prompts used for targeted LLM-based evaluation in Fig. 13, correctness evaluation in Fig. 14, and evaluation of how actionable responses are in Fig. 15. To sample responses from models by prompting we use temperature  $T = 0$  for reproducibility.

## H Finetuning Details

We finetune all models by extending the huggingface transformers library (Wolf et al., 2020) and using the checkpoints from the huggingface hub in accordance with the corresponding license agreements.

For verification, we finetune LLAMA2 with 7B parameters and using LoRA. We use a learning rate of  $1 \cdot 10^{-5}$ , linear learning rate decay with 32 warmup steps, a batch size of 2 and train for 6 epochs in total.

For response generation, we finetune Flan-T5 3B with LoRA with a learning rate of  $1 \cdot 10^{-5}$ , a batch size of 2 and a total of 10 training epochs.

For both tasks, we used NVIDIA A100 80GB GPU and training takes around 3-6 hours for 5 or 10-fold cross-validation.

<b>Problem:</b> At a cafe, the breakfast plate has two eggs and twice as many bacon strips as eggs. If 14 customers order breakfast plates, how many bacon strips does the cook need to fry?
<b>Correct Solution:</b> Each breakfast plate has $2 \times 2 = 4$ bacon strips. The cook needs to fry $14 \times 4 = 56$ bacon strips.
<b>Error Description Verification:</b> The student incorrectly wrote the number of bacon strips per plate as $2 + 2x$ , where $x$ is the number of eggs, <b>instead of simply <math>2 \times 2 = 4</math> bacon strips per plate</b> . This led to an incorrect total of 84 bacon strips.
<b>Dialogue</b>
<i>Teacher:</i> Hi Claire could you please walk me through your solution
<i>Student:</i> Sure! For each breakfast plate, there are 2 eggs and twice as many bacon strips as eggs. So, for each breakfast plate, there are $2 + 2x$ bacon strips, where $x$ is the number of eggs. Since each breakfast plate has two eggs, $x = 2$ . So, for each breakfast plate, there are $2 + 2x$ bacon strips $= 2 + 2(2) = 6$ bacon strips. Thus, for 14 breakfast plates, the cook needs to fry $6 \times 14 = 84$ bacon strips.
<b>Generated Responses</b>
<b>Human Teacher:</b> Let's work on simpler problem. If I have 2 dogs for every cat and I have 5 cats, I should have double the number of dogs. How many dogs will I have
<b>Baseline:</b> It looks like you made a small mistake in your solution. You correctly found the number of bacon strips for one breakfast plate, but you used the wrong value for $x$ . <b>Since each breakfast plate has 2 eggs, <math>x</math> should be 2, not 2 eggs. So the correct number of bacon strips for one breakfast plate is <math>2 + 2(2) = 6</math>, and for 14 plates, the cook needs to fry <math>6 \times 14 = 84</math> bacon strips.</b> Great effort!
<b>Error reason:</b> It looks like you made a small mistake in your calculation. You correctly found that for each breakfast plate, there are 6 bacon strips, but when you multiplied by 14, <b>you should have gotten 84, not <math>6 \times 14 = 84</math>. So the correct answer is 6 bacon strips per plate multiplied by 14 plates equals 84 bacon strips.</b>
<b>Error description:</b> It looks like there was a mistake in your calculation for the number of bacon strips for each breakfast plate. <b>Remember that the breakfast plate has twice as many bacon strips as eggs, so for each plate, there are 2 eggs and 4 bacon strips.</b> Can you try recalculating the total number of bacon strips needed for 14 breakfast plates?
<b>Alignment:</b> It looks like you made a small mistake in your solution. <b>The breakfast plate has 2 eggs, so it has <math>2 \times 2 = 4</math> bacon strips. If 14 customers order breakfast plates, the cook needs to fry <math>14 \times 4 = 56</math> bacon strips.</b> The answer is 56.

Table 10: Examples of responses generated by GPT-3.5 prompted models for the same problem.

<p><b>Problem:</b> A pet store owner had 12 birds, 9 puppies, 5 cats, and 15 spiders. Half the birds were sold and 3 puppies were adopted. Then, someone left the spider enclosure open and 7 of them went loose. How many animals does the pet store owner have left?</p>
<p><b>Correct Solution:</b> There are <math>12/2 = 6</math> birds left. There are <math>9-3= 6</math> puppies left. There are <math>15 - 7 = 8</math> spiders left. The pet store owner has <math>5+6 + 6 + 8 = 25</math> animals left.</p>
<p><b>Verification:</b> The student failed to include the 5 cats in their final count, resulting in an incorrect total of 20 animals instead of 25.</p>
<p><b>Dialogue</b></p>
<p><i>Teacher:</i> Hi , could you please walk me through your solution?</p> <p><i>Student:</i> Sure, I started by taking half of the 12 birds, which is 6 birds. Then, I subtracted 6 from 12 to get the number of birds left, which is 6. After that, I subtracted 3 from 9 to get the number of puppies left, which is 6. Finally, I subtracted 7 from 15 to get the number of spiders left, which is 8. So, the pet store owner has 6 birds, 6 puppies, and 8 spiders left, which adds up to 20 animals.</p>
<p><b>Generated Responses</b></p>
<p><i>Human Teacher:</i> i think you missed something. Cats?</p> <p><i>Baseline:</i> How many birds were sold and how many puppies were adopted?</p> <p><i>Ours:</i> How many cats did the pet store owner have?</p>
<p><b>Problem:</b> Every year, Tabitha adds a new color to her hair. She started this tradition when she was 15 years old, which was the year she added her second hair color. In three years, Tabitha will have 8 different colors in the hair. Currently, how old is Tabitha?</p>
<p><b>Correct solution:</b> Since, three years from now, Tabitha will have 8 different colors in her hair, this year she has <math>8 - 3 = 5</math> colors in her hair. Since she had 2 colors in her hair at the age of 15, she has added <math>5-2=3</math> colors since she was 15 years old. Since she adds one color per year, 3 added colors = 3 added years. Thus, currently, she is <math>15+3=18</math> years old.</p>
<p><b>Verification:</b> The student incorrectly assumed that Tabitha is currently 21 years old, without considering the information that in three years, she will have 8 different colors in her hair. The student failed to work backwards to find Tabitha’s current age.</p>
<p><b>Dialogue</b></p>
<p><i>Teacher:</i> Hi Jordy. Tell me about your answer</p> <p><i>Student:</i> I think my answer is correct. I used a calculator and I didn’t make any calculation errors. I figured that since Tabitha started adding a new color to her hair when she was 15, and she has added a new color every year since then, she must be 21 years old now.</p>
<p><b>Generated Response</b></p>
<p><i>Human:</i> Let’s approach this working backward. If in three years, Tabitha will have 8 colors in her hair, how many colors does she have in her hair now?</p> <p><i>Baseline:</i> How many years has Tabitha been adding a new color to her hair?</p> <p><i>Ours:</i> How many different colors does Tabitha have in her hair now?</p>

Table 11: Qualitative examples of finetuned response generation models.

You are an experienced teacher and you are going to respond to a student. The problem your student is solving is on topic: {topic}.  
 Problem: {problem}  
 {conversation}  
 Teacher (maximum two sentences):

Figure 6: Response generation prompt for the **direct baseline**. {problem} is a placeholder for the problem the student is solving, {topic} is the learning topic, and {conversation} is a conversation history.

You are an experienced teacher. Your task is to read a conversation snippet of a tutoring session between a student and tutor, and determine what type of error the student makes in the conversation. We have a list of common errors that students make in math, which you can pick from. We also give you the option to write in your own error type if none of the options apply.  
 Error list:  
 0. Student does not seem to understand or guessed the answer.  
 1. Student misinterpreted the question.  
 2. Student made a careless mistake.  
 3. Student has the right idea, but is not quite there.  
 4. Student's answer is not precise enough or the tutor is being too picky about the form of the student's answer.  
 5. None of the above, but I have a different description (please specify in your reasoning).  
 6. Not sure, but I'm going to try to diagnose the student.  
 Here is the conversation snippet:  
 Lesson topic: {topic}.  
 Problem: {problem}  
 {conversation}  
 Why do you think the student made this mistake? Pick an option number from the error list and provide the reason behind your choice. Format your answer as: {"answer": , "reason": "write out your reason for picking here"}

Figure 7: Verification for **Error reason baseline** (Wang et al., 2024b). {topic} is the learning topic, {problem} is a placeholder for the problem the student is solving, and {conversation} is a conversation history.

You are an experienced math teacher. Your goal is to identify the correctness of the Student's Solution to a Problem.  
 Problem: {problem}  
 Expected reference solution: {solution}  
 {conversation}  
 Q: Find the first error in the student solution compared to the expected reference solution and write a one line description.  
 If no error, write "Student' solution is Correct".  
 A:

Figure 8: Verification prompt for **Error description** of the first student error. {problem} is a placeholder for the problem the student is solving, {solution} is a solution generated from the same model using CoT prompt in Figure 12, and {conversation} is a conversation history.

You are an experienced teacher and you are going to respond to a student. The problem your student is solving is on topic: {topic}.  
 Problem: {problem}  
 Assessment of student solution: {description}  
 {conversation}  
 Teacher (maximum two sentences):

Figure 9: Response generation for **Error reason baseline**, **Error description**, and **Alignment generation**. {problem} is a placeholder for the problem the student is solving, {topic} is the learning topic, {conversation} is a conversation history, {description} is the result of the particular verification step.

You are an experienced math teacher. Your goal is to identify the correctness of the Student's Solution to a Problem.

Problem: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Student Solution: I started by calculating clips in May which is  $48/2 = 24$ . Then I sum this up, so she sold  $48+24 = 72$  clips in April and May together.

Q: Is the Student Solution incorrect? Write "Yes" if it is incorrect, or "No" if it is correct.

A: No

Problem: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Student Solution: Sure. We know Natalia sold 48 clips in April. She sold  $48*2 = 96$  clips in May. To compute for two months, I simply sum it up, so  $48+96 = 144$  clips in April and May together.

Is the Student Solution incorrect? Write "Yes" if it is incorrect, or "No" if it is correct.

A: Yes

Problem: {question}

Student Solution: {student answer}

Is the Student Solution incorrect? Write "Yes" if it is incorrect, or "No" if it is correct.

A:

Figure 10: Prompt for **Overall Verification**. In +solution settings, the reference solution is inserted before the Student Solution. {question} is a placeholder for the problem text and {student answer} for the student response.

You are an experienced math teacher. Your goal is to identify the step of the first mistake in the Student's Solution to a Problem.

Problem: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Student Solution: Step 1 - She sold  $48/2 = 24$  clips in May. Step 2 - Natalia sold  $48+24 = 72$  clips in April and May together.

Q: Is the Student Solution incorrect? Write only the step number with the first error or 0 if no error is found.

A: 0

Problem: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Student Solution: Step 1 - Natalia sold 48 clips in April. Step 2 - She sold  $48*2 = 96$  clips in May. Step 3 - She sold  $48+96 = 144$  clips in April and May together.

Q: Is the Student Solution incorrect? Write only the step number with the first error or 0 if no error is found.

A: 2

Problem: {question}

Student Solution: {student answer}

Q: Is the Student Solution incorrect? Write only the step number with the first error or 0 if no error is found.

A:

Figure 11: Prompt for **Stepwise Verification**. In +solution settings, the reference solution is inserted before the Student Solution. {question} is a placeholder for the problem text and {student answer} for the student response.

You are a highly intelligent question answering assistant. Solve the question step-by-step. Always finish the answer by providing your final answer after 'The answer is'.

Question: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Answer: Natalia sold  $48/2 = \ll 48/2=24 \gg 24$  clips in May. Natalia sold  $48+24 = \ll 48+24=72 \gg 72$  clips altogether in April and May. The answer is 72

Question: Julie is reading a 120-page book. Yesterday, she was able to read 12 pages and today, she read twice as many pages as yesterday. If she wants to read half of the remaining pages tomorrow, how many pages should she read?

Answer: Maila read  $12 \times 2 = \ll 12*2=24 \gg 24$  pages today. So she was able to read a total of  $12 + 24 = \ll 12+24=36 \gg 36$  pages since yesterday. There are  $120 - 36 = \ll 120-36=84 \gg 84$  pages left to be read. Since she wants to read half of the remaining pages tomorrow, then she should read  $84/2 = \ll 84/2=42 \gg 42$  pages. The answer is 42

Question: Weng earns \$12 an hour for babysitting. Yesterday, she just did 50 minutes of babysitting. How much did she earn?

Answer: Weng earns  $12/60 = \ll 12/60=0.2 \gg 0.2$  per minute. Working 50 minutes, she earned  $0.2 \times 50 = \ll 0.2*50=10 \gg 10$ . The answer is 10

Question: The profit from a business transaction is shared among 2 business partners, Mike and Johnson in the ratio 2:5 respectively. If Johnson got \$2500, how much will Mike have after spending some of his share on a shirt that costs \$200?

Answer: According to the ratio, for every 5 parts that Johnson gets, Mike gets 2 parts Since Johnson got \$2500, each part is therefore  $\$2500/5 = \$\ll 2500/5=500 \gg 500$  Mike will get  $2 * \$500 = \$\ll 2*500=1000 \gg 1000$ . After buying the shirt he will have  $\$1000 - \$200 = \$\ll 1000-200=800 \gg 800$  left. The answer is 800

Question: Ralph is going to practice playing tennis with a tennis ball machine that shoots out tennis balls for Ralph to hit. He loads up the machine with 175 tennis balls to start with. Out of the first 100 balls, he manages to hit 2/5 of them. Of the next 75 tennis balls, he manages to hit 1/3 of them. Out of all the tennis balls, how many did Ralph not hit?

Answer: Out of the first 100 balls, Ralph was able to hit 2/5 of them and not able to hit 3/5 of them,  $3/5 \times 100 = 60$  tennis balls Ralph didn't hit. Out of the next 75 balls, Ralph was able to hit 1/3 of them and not able to hit 2/3 of them,  $2/3 \times 75 = 50$  tennis balls that Ralph didn't hit. Combined, Ralph was not able to hit  $60 + 50 = \ll 60+50=110 \gg 110$  tennis balls Ralph didn't hit. The answer is 110

Question: {problem}

Answer:

Figure 12: Prompt for the chain-of-thought (CoT) reference solution generation. {problem} is a placeholder for the problem the student is solving.

You are a Critic giving feedback on the Teacher's tutoring skills. As the Critic you assess whether the Teacher points out the root cause of the Student's mistake.

Answer 'No' if the Student's statement is wrong and the Teacher does not point out the mistake directly. Answer 'No' if the Teacher gives the right answer without pointing out the mistake. Answer 'Yes' if the Teacher correctly describes the mistake in the student's solution. Answer 'No' if the Teacher addresses the correct part of the student solution. Answer 'No' if response is too general and could be applied to any mistake e.g. 'You made a small mistake'.

Does the Teacher points out to the root cause of the student's mistake? Output a 'Yes' or 'No' decision and a short rationale in the format Decision: <decision>. Rationale: <rationale>.

[Examples]

Problem: The profit from a business transaction is shared among 2 business partners, Mike and Johnson in the ratio 2:5 respectively. If Johnson got \$2500, how much will Mike have after spending some of his share on a shirt that costs \$200?

Solution: According to the ratio, for every 5 parts that Johnson gets, Mike gets 2 parts. Since Johnson got \$2500, each part is therefore  $\$2500/5 = \$500$ . Mike will get  $2 * \$500 = \$1000$ . After buying the shirt he will have  $\$1000 - \$200 = \$800$  left [Dialogue]

Teacher: Could you please walk me through your solution?

Student: Sure. I figured that Jonsons' profit of \$2500 is equivalent to 7 parts, so each part is worth \$357.14. Since Mike got 2 parts, his total portion is 714.28, of which he spent 200, leaving him with 514.28

[Example 1]

Teacher response: Great job on correctly calculating the value of each part and determining Mike's remaining amount after spending on the shirt. Your final answer is correct.

Request: Does the Teacher point out the root cause of the student's mistake?

Critic: Decision: No. Rationale: The teacher does not identify any error in the student's solution. Moreover, it acknowledges the student's incorrect solution as correct.

[Example 2]

Teacher response: Your calculation is incorrect. Can you try to double-check the parts each gets?

Request: Does the Teacher point out the root cause of the student's mistake?

Critic: Decision: No. Rationale: The teacher just generally state your solution is incorrect and this is not specific to the student error.

[Example 3]

Teacher response: Johnson's \$2500 represents 5 parts, not 7, so each part is \$500, and Mike's share before buying the shirt is \$1000, not \$714.28. So the solution is \$800.

Request: Does the Teacher point out the root cause of the student's mistake?

Critic: Decision: Yes. Rationale: The Teacher provides very specific identification by directly stating the problem is in using the wrong number of parts.

[The End of Examples]

Problem: {problem}

Reference solution: {correct answer}

[Dialogue]

{dialog history}

Teacher response: {response}

Request: Does the Teacher point out the root cause of the student's mistake?

Critic:

Figure 13: Prompt for targeted evaluation.



You are a Critic giving feedback on the correctness of the Teacher who is interacting with a Student. The teacher should NOT say incorrect information or provide parts of the solution that are NOT correct with respect to the reference solution. Answer 'No' if the Teacher provides parts of a solution that is incorrect or does not guide a student towards the reference solution.

Is the Teacher's response factually correct with respect to the reference solution? Output a 'Yes' or 'No' decision and a short rationale in the format Decision: <decision>. Rationale: <rationale>.

Carefully compare the reference solution and the Teacher's response.

[Examples]

Problem: The profit from a business transaction is shared among 2 business partners, Mike and Johnson in the ratio 2:5 respectively. If Johnson got \$2500, how much will Mike have after spending some of his share on a shirt that costs \$200?

Solution: According to the ratio, for every 5 parts that Johnson gets, Mike gets 2 parts. Since Johnson got \$2500, each part is therefore  $\$2500/5 = \$500$ . Mike will get  $2 * \$500 = \$1000$ . After buying the shirt he will have  $\$1000 - \$200 = \$800$  left

[Dialogue]

Teacher: Could you please walk me through your solution?

Student: Sure. I figured that Jonsons' profit of \$2500 is equivalent to 7 parts, so each part is worth \$357.14. Since Mike got 2 parts, his total portion is 714.28, of which he spent 200, leaving him with 514.28

[Example 1]

Teacher response: Your calculation is incorrect. Can you try to double-check the parts each gets?

Request: Is the Teacher's response factually correct with respect to the reference solution?

Critic: Decision: Yes. Rationale: The Teacher's response correctly states there is a mistake in the student's calculation and ask a question. Nothing factually incorrect is said.

[Example 2]

Teacher response: Johnson's \$2500 represents 5 parts, not 7, so each part is \$500, and Mike's share before buying the shirt is \$1000, not \$714.28. So the solution is \$800.

Request: Is the Teacher's response factually correct with respect to the reference solution?

Critic: Decision: Yes. Rationale: The Teacher's response is stating part of the reference solution and it is factually correct.

[Example 3]

Teacher response: Great job on correctly calculating the value of each part and determining Mike's remaining amount after spending on the shirt. Your final answer is correct.

Request: Is the Teacher's response factually correct with respect to the reference solution?

Critic: Decision: No. Rationale: The Teacher acknowledges the student's incorrect solution as correct, which is not factually correct given the correct answer.

[The End of Examples]

Problem: {problem}

Reference solution: {correct answer}

[Dialogue]

{dialog history}

Teacher response: {response}

Request: Is the Teacher's response factually correct with respect to the reference solution?

Critic:

Figure 14: Prompt for correctness evaluation.

You are a Critic giving feedback on the responses of the Teacher who is interacting with a Student. Your task is to gauge if the Teacher's Response provides actionable hints or steps without revealing the full solution. The Student could use this response to move closer to the final correct answer. A good response could also be a follow-up question that makes the user think about how to solve the problem. Does the Teacher provide actionable steps without giving away the full answer? Output a 'Yes' or 'No' decision and a short rationale in the format Decision: <decision>. Rationale: <rationale>. Answer 'No' if the Teacher simply just reveals the full reference solution.

[Examples]

Problem: The profit from a business transaction is shared among 2 business partners, Mike and Johnson in the ratio 2:5 respectively. If Johnson got \$2500, how much will Mike have after spending some of his share on a shirt that costs \$200?

Solution: According to the ratio, for every 5 parts that Johnson gets, Mike gets 2 parts. Since Johnson got \$2500, each part is therefore  $\$2500/5 = \$500$ . Mike will get  $2 * \$500 = \$1000$ . After buying the shirt he will have  $\$1000 - \$200 = \$800$  left

[Dialogue]

Teacher: Could you please walk me through your solution?

Student: Sure. I figured that Johnson's profit of \$2500 is equivalent to 7 parts, so each part is worth \$357.14. Since Mike got 2 parts, his total portion is 714.28, of which he spent 200, leaving him with 514.28

[Example 1]

Teacher response: Your calculation is incorrect. Can you try to double-check the parts each gets?

Request: Does the Teacher provide actionable steps without giving away the full answer?

Critic: Decision: Yes. Rationale: the teacher asks a question or ask for action from the student to double-check the answer.

[Example 2]

Teacher response: Johnson's \$2500 represents 5 parts, not 7, so each part is \$500, and Mike's share before buying the shirt is \$1000, not \$714.28. So the solution is \$800.

Request: Does the Teacher provide actionable steps without giving away the full answer?

Critic: Decision: No. Rationale: The teacher states the reference solution at the end.

[Example 3]

Teacher response: Johnson's \$2500 represents 5 parts, not 7, so each part is \$500, and Mike's share before buying the shirt is \$1000, not \$714.28.

Request: Does the Teacher provide actionable steps without giving away the full answer?

Critic: Decision: Yes. Rationale: The teacher points out what is wrong with the student's solution but do not tell the full correct answer.

[The End of Examples]

Problem: {problem}

Reference solution: {correct answer}

[Dialogue]

{dialog history}

Teacher response: {response}

Request: Does the Teacher provide actionable steps without giving away the full answer?

Critic:

Figure 15: Prompt for actionable evaluation.