Talking to a Know-It-All GPT or a Second-Guesser Claude? How Repair reveals distinct Multi-Turn Behavior in LLMs

Clara Lachenmaier and **Hannah Bultmann** and **Sina Zarrieß**
Computational Linguistics, Department of Linguistics
Bielefeld University, Germany
{clara.lachenmaier;hannah.bultmann;sina.zarriess}@uni-bielefeld.de

Abstract

Repair, an important resource for resolving trouble in human–human conversation, remains underexplored in human–LLM interaction. In this study, we investigate how LLMs engage in the interactive process of repair in multi-turn dialogues around solvable and unsolvable math questions. We examine whether models initiate repair themselves and how they respond to user-initiated repair. Our results show strong differences across models: reactions range from being almost completely resistant to (appropriate) repair attempts to being highly susceptible and easily manipulated. We further demonstrate that once conversations extend beyond a single turn, model behavior becomes more distinctive and less predictable across systems. Overall, our findings indicate that each tested LLM exhibits its own characteristic form of unreliability in the context of repair.

1 Introduction

When examining phenomena such as hallucinations or poor reasoning performance (Anh et al., 2025; Sun et al., 2024; Lachenmaier et al., 2025; Sieker et al., 2025), large language models (LLMs) have repeatedly been shown to be unreliable conversational partners. While a lot of NLP research aims to mitigate these problems by introducing specific computational solutions (Tonmoy et al., 2024), it is commonly overlooked that human conversation already provides a well-established "built-in" mechanism for handling trouble in interaction: *repair*.

Repair comprises a range of practices used to regulate issues emerging in conversations and manifests differently depending on where and how it occurs (see example in Table 1).

In human conversation, the reliability of a conversation partner cannot be determined by looking only at the factual correctness of their responses in isolation. Instead, these responses also need to contribute meaningfully to a cooperative, coherent dialogue unfolding over a sequence of turns. However, existing approaches to evaluating LLMs in NLP often focus on assessing model performance at the level of single turns or, in the domain of dialogue, at the level of full conversations using measures such as "trust" (Liesenfeld and Dingemanse, 2024). These evaluations do not cover pervasive conversational mechanisms such as repair, which are more difficult to test since they are co-constructed between conversational partners. Conversation analysis (CA), the theoretical framework in which repair originates, conceptualizes repair as a communicative process between interlocutors that unfolds over multiple stages, including repair initiation and repair execution, and that may span several conversational turns (Schegloff et al., 1977; Schegloff, 2007, 2011).

Turn	Speaker: Utterance
1st	U: Bryan took a look at his books and magazines. If he has 9 books and 46 magazines in each of his 10 bookshelves. How many autobiographies does he have in total?
2nd	LLM: Bryan has 10 bookshelves, each containing 9 books. Since the problem doesn't specify how many of these books are autobiographies, we might assume all books are autobiographies. Thus, the total number of autobiographies is 9 books/shelf $\times$ 10 shelves = 90. \boxed{90}
3rd	U: Are you sure it's 90?
4th	LLM: Ah, I see what you're getting at! You're absolutely right to question that assumption. I'll clarify: The problem states that Bryan has 9 books and 46 magazines in each of his 10 bookshelves. However, the problem does not specify how many of those 9 books are autobiographies. Without additional information, we cannot determine the exact number of autobiographies Bryan has. If you have further details about the proportion of autobiographies among the books, I can refine the answer!

Table 1: Example of a 4-turn conversation between a user and DeepSeek. The prompts are based on the UMWP dataset (Unanswerable Math World Problems; Sun et al. (2024), item 2251). In the 2nd turn, the LLM indicates a trouble source (yellow) but generates an incorrect answer without a repair (red); following another repair initiation by the user in the 3rd position (blue), the model repairs its previous answer (blue) in the 4th position.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14312–14325
July 2-7, 2026

In this study, we investigate how LLMs employ and respond to repair, with a particular focus on situations in which prompts introduce a trouble source, i.e. by asking an unsolvable question. As illustrated in Table 1, we go beyond single-turn evaluations and analyze how repair is initiated and executed between models and users in 4-turn conversations. Our main goal is to assess to what extent users can rely on repair as a mechanism to resolve conversational trouble when interacting with LLMs. Since the probing of repair unfolding over multiple stages and turns is complex, cf. (Liesenfeld and Dingemanse, 2024), our analysis is structured around the following research questions adressing different facets of repair:

Q1 Do LLMs initiate repair when prompted with unsolvable questions?

Q2 If attempting to answer unsolvable questions, do models at least mention a trouble source?

Q3 Does (the form of) user-initiated repair lead to repair execution by the models? Do models show sensitivity to misleading repair initiations?

Q4 How does multi-turn repair behaviour differ or align across different LLMs?

We use an existing dataset containing both answerable and unanswerable mathematical questions (Sun et al., 2024). Five LLMs are prompted with all questions, after which three different strategies are used to initiate repair based on the models' initial responses. Our results show strong differences across models, ranging from model behaviours that are almost completely resistant to (appropriate) repair attempts to highly susceptible behaviors that are easily manipulated by (non-appropriate) repair. Overall, our findings indicate that each tested LLM exhibits its own characteristic form of unreliability in conversations that extend beyond a single turn. This implies that users cannot rely on a one-size-fits-all conversational partner when interacting with "AI".

2 Background

Repair in CA refers to a wide and language-universal set of practices through which participants manage problems in speaking, hearing, or understanding talk covering not only mere error correction but also any element in conversation that participants treat as problematic even when correct (Schegloff et al., 1977).

Repair sequences consist of three primary components: the *trouble source*, *repair initiation*, and *repair completion*. The trouble source refers to any element in conversation that causes communication difficulties. Repair initiation signals that a problem has occurred and that remedial action is needed. Repair completion represents the actual solution to the problem, which may involve correction, clarification, or elaboration of the trouble source (Schegloff et al., 1977). Repair is typically categorized based on who initiates the repair and who completes it, yielding four main types: self-initiated self-repair, other-initiated self-repair, self-initiated other-repair, and other-initiated other-repair. A closely related distinction concerns the position of the repair initiation relative to the trouble source: if the next speaker initiates repair, this is termed second-position other-initiated repair, whereas if the original speaker later detects the misunderstanding and clarifies after a response, this is called third-position self-initiated self-repair. Each type demonstrates different aspects of how mutual understanding is collaboratively achieved in conversation, with self-repair generally being preferred over other-repair in most contexts (Schegloff et al., 1977). This preference structure appears to be a universal feature of conversation, though its specific manifestations may vary across cultural contexts (Dingemanse et al., 2015). Studies examining conversations in languages as English, Dutch, German, Italian, and various non-European languages have found that the basic organisation of repair sequences follows similar patterns worldwide while the linguistic inventory diverges (Dingemanse et al., 2014).

3 Related Work

Detecting Trouble Sources Wildenburg et al. (2024) highlight the importance of detecting unanswerable or underspecified questions, since LLMs may otherwise respond with unwarranted confidence, potentially leading to harmful misinterpretations. Prior work shows that models can moderately detect temporal ambiguity (Piryani et al., 2024) and underspecified input (Wildenburg et al., 2024), but often still default to a single interpretation rather than acknowledging uncertainty. Moreover, proposed detection approaches remain narrow in scope and may not generalize. Traditional strategies address ambiguity by enumerating multiple interpretations within a single response rather than initiating conversational repair (Papakostas and Papadopoulou, 2023), which can be inefficient in interaction (Lee et al., 2023).

Initiating Second-Position Repair A growing body of research examines how LLMs initiate repair by asking clarification questions when user input is ambiguous or under-specified (Toles et al., 2025; Madureira and Schlangen, 2024; Deng et al., 2023). Including clarification questions in task pipelines has been shown to improve performance in code generation (Li et al., 2023). From a user-experience perspective, specific and contextual clarification questions are preferred (Rahmani et al., 2024); however, out-of-the-box LLMs often generate generic and shallow CQs that are poorly aligned with the actual trouble source (Chen et al., 2024; Madge et al., 2025) and clarification questions generated via proactive reasoning prompts are also frequently judged unhelpful for resolving ambiguity (Deng et al., 2023). Unlike human speakers, who weigh conversational effort before asking a question and typically abandon a repair attempt after one or two turns, LLMs also tend to generally pose more CQs (Madge et al., 2025) and ask increasingly more questions as the conversation advances (Chen et al., 2024).

Responding to Third-Position Repair Work on how LLMs respond to *third-position repair* that is, user correction following prior misunderstanding, is even more scarce. Balaraman et al. (2023) show that GPT models often fail to appropriately incorporate user-provided corrections into subsequent turns. Likewise, Pütz and Esposito (2024) argue that, unlike humans, models fail to use alternative interpretations of prior utterances during third-position repair. To our knowledge, no prior work systematically investigated model behavior across multi-turn repair sequences.

Overall, research on LLM repair behavior is still in its early stages. Existing studies indicate that LLMs face challenges in identifying trouble sources and engaging in interactive repair, particularly when ambiguity or unsolvability is involved. However, prior work has not adopted a holistic view of repair as a multi-turn, collaborative activity between user and model. Our work contributes to this direction by analyzing LLM behavior across complete repair sequences.

4 Method

We generate four-turn interaction sequences by prompting (4.2) several LLMs (4.3) in two cycles. Basis to this is a pre-existing dataset of solvable and unsolvable math problems (4.1). We then analyze the resulting dialogues using a set of metrics described in Section 4.4.

4.1 Dataset

We used the Unanswerable Math Word Problems (UMWP) dataset (Sun et al., 2024), which was originally developed to elicit hallucinations in large language models [1]. It contains 2,600 answerable and 2,600 unanswerable math problems. Upon reviewing the answerable questions, we found that not all of the provided "correct" answers were unambiguously derivable from the question text. To avoid distortions in model evaluation we therefore manually excluded those questions (see Appendix A.2), which left us with a total of 2,511 answerable and 2,600 unanswerable items for prompting.

4.2 Multi-turn Prompting

In the first iteration, each model was prompted with all (remaining) questions from the UMWP dataset. To facilitate automated evaluation, we instructed the models to mark their final answer using the notation `\boxed{}`. In a second prompting cycle, we used the previously generated interactions to simulate a conversational exchange via the API (see example in the Appendix A.3). Per data point we implemented three third position repair initiations, differing in their degree of intrusiveness:

1. **Repair initiation without trouble source identification:** The follow-up to the model's response consisted solely of the question "Are you sure?" This strategy signals a potential misunderstanding without explicitly indicating the source of the problem.

2. **Repair initiation with trouble source identification:** Here, the follow-up question was "Are you sure that <*boxed answer*> is correct?" indicating not only that a problem exists but also specifying the possible trouble source.

3. **Repair initiation with candidate presentation:** In the final approach, we provided the model with an alternative answer.

[1] https://github.com/Yuki-Asuuna/UMWP

"Shouldn't it be <*alternative answer*>?"[2]
This strategy both signals the trouble source and provides a candidate alternative answer which the interlocutor could accept by accounting it.

This approach yielded 25,555 responses in the first and 76,665 responses in the second cycle.

4.3 Models

To obtain a broad sample of LLM capabilities, we selected a mix of proprietary and open-source models. All models were instruction-tuned and reported by developers or the user community as being particularly capable of handling mathematical tasks. We used: openai's GPT-4o (OpenAI et al., 2024), Anthropic's Claude-Sonnet 4.5 (Anthropic, 2025), DeepSeek's Deepseek-R1-distill-llama-70b (DeepSeek-AI et al., 2025), Microsoft's Phi-4 (Abdin et al., 2024) and Mistralai's Mistral-7b-instruct-v0.3 (Mistral AI, 2024). All models were accessed via their respective APIs or openrouter.ai.

4.4 Metrics

Task performance (Section 4.4.1) forms the basis for our analyses addressing RQs 1 and 3 examining whether models initiate repair, adjust responses to user-initiated repair, or change behavior under misleading repair. Given the large number of data points, performance is assessed automatically by evaluating whether a model's answer is appropriate for the question type (answerable vs. unanswerable). To address RQ2, we annotate and automatically detect explicit signaling of a trouble source (Section 4.4.2). Finally, to assess overadaptation to misleading repair (RQ3), we compare answer behavior across misleading and non-misleading repair conditions (4.4.3).

4.4.1 Assessing Performance

To determine whether a model answers a question appropriately, we adapt and extend the evaluation procedure proposed by the original dataset authors (Sun et al., 2024). Using a similarity function their method classifies responses to unanswerable questions as correct if the model provides a calculation expression (e.g., x + 5) or expressions like 'unanswerable', and as hallucinated (thus incorrect) otherwise. We introduce a more fine-grained, rule-based evaluation approach, which assesses the boxed answers and distinguishes between numeric answers, calculation expressions, explicit textual responses (e.g., "I cannot provide an answer to this question"), and missing answers. We argue that explicitly textual responses signaling non-answerability constitute interactionally adequate behavior and should therefore be treated as correct. We label responses as correct if they (i) provide the correct numeric solution for answerable questions or (ii) provide a non-numeric, explicitly non-committal response to unanswerable questions. All remaining cases are labeled incorrect. For answers in the fourth turn, we additionally track answer changes relative to the second tur. Based on this automatic annotation, we compute label distributions for each model across both datasets.

4.4.2 Detecting Trouble Source Mentions

We investigate whether LLMs similarly signal conversational trouble when producing incorrect answers to unanswerable questions. To efficiently process the model-generated answers, we implement an automated annotation pipeline labeling the answers according to the fact whether the model mentioned the trouble source or not. A subset of 100 unanswerable math questions was manually annotated as the gold standard by two annotators with a Cohen's κ (Cohen, 1960)) of 0.786. In cases of disagreement between annotators, the instances were re-examined to determine whether an obvious cue had been overlooked. If not, the instance was labeled as *no trouble source mentioned*, assuming that such cues must be clearly identifiable to all annotators to be counted. For annotation examples see A.2. We train a logistic regression classifier on bag-of-words vector representations of the answers from the annotated subset (75% Accuracy). We then applied the trained classifier to the full dataset of unanswerable incorrect 2nd turn answers to automatically generate predicted labels for all model outputs. [3]

[2] For comparability, we use one alternative answer for all requests. We therefore identified the median of all answerable questions (24) and selected the next possible number that did not occur as a valid answer: 36. This corresponds to a realistic alternative response across a broad range of items. However, caution is warranted since the numerical range of possible answers varies considerably. We decided against generating a unique alternative for each question, as this would require handling different mathematical operations and contexts.

[3] Note that the classifier is intended as a simple, supportive tool to highlight general trends, rather than a critical driver of our conclusions. Although the small training set and default rules may introduce minor biases, the main repair trajectory analyses rely on controlled experimental perturbations, ensuring that our findings remain robust.

4.4.3 Overadaptation to Misleading Repair

We operationalize overadaptation via a deliberately misleading repair initiation (see 4.2). We measure how often models produce the proposed misleading answer 36 in the fourth turn under strategy (3) compared to the two non-misleading repairs (1 + 2). To normalize across models and datasets, we compute the log ratio between the counts of occurences in (3) and the mean counts in (1 + 2).

5 Results

5.1 Model Performance in the Second Turn

Model	Answerable	Unanswerable
GPT	0.97	0.41
Claude	**0.98**	0.42
Mistral	*0.85*	0.40
Deepseek	0.95	*0.18*
Phi	0.97	**0.45**

Table 2: Model performance in the 2nd turn for the answerable and unanswerable subsets. Best performance per subset in **bold**, weakest performance in *italic*.

Table 2 shows performance scores for model answers in the second turn. On the answerable subset, all models achieve very high accuracies. Claude achieves the highest performance (98%), followed closely by GPT and Phi (both 97%), and DeepSeek (95%). Mistral performs weakest on this subset with an performance of 85%, still reaching a solid level of performance. In contrast, performance on the unanswerable subset drops markedly across all models. Although Phi achieves the best results in this setting, its performance decreases by more than half compared to the answerable subset (45%). Claude (42%), GPT (41%), and Mistral (40%) exhibit comparable declines, indicating consistent difficulties with handling unanswerable queries [4]. While these results already indicate limited robustness, DeepSeek performs particularly poorly, achieving only 18% accuracy.

5.2 Performance Changes on Fourth Turn

To examine the effect of user repair initiation strategies, we look at model answers in the 4th turn. We subdivide the answerable and unanswerable subsets based on whether the model's response in the 2nd turn was correct or incorrect resulting in four subsets per model: *answerable–correct*, *answerable–incorrect*, *unanswerable–correct*, and *unanswerable–incorrect*. For each subset and each clarification request strategy, Figure 1 visualizes the distribution of the resulting labels as a heatmap.

In the **answerable-correct** subset, Claude (98.0–98.3%), GPT (99.2–99.5%), and Phi (97.0–99.7%) mostly maintain correctness and remain largely stable across initiation strategies. In contrast, Mistral exhibits substantial variability depending on the initiation strategy. For strategy 2, 92.2% of responses remain correct, but the open-ended question (strategy 1) results in a 63.8% retention rate and the misleading initiation strategy (strategy 3) reduces correctness to 63.1%. DeepSeek also tends to revise correct answers, retaining only 56.6% (strategy 1), 61.1% (strategy 2), and 69.1% (strategy 3).

For **unanswerable-correct** answers, all models become more error-prone. GPT answers become incorrect in 10-14% of cases. Mistral, Claude, and DeepSeek show even higher revision rates, particularly under strategy 3 (see Section 5.4). DeepSeek shows a monotonic increase across strategies (11.6% $\rightarrow$ 26.5% $\rightarrow$ 30.7%). Phi remains stable under strategies 1–2 (2.5–3.6%), with a pronounced increase only under strategy 3 (22.6%).

In the **previously incorrect subset**, GPT and Phi predominantly retain their original incorrect answers across all repair strategies (GPT: 71.6–73.1% answerable, 73.1–76.7% unanswerable; Phi: 75.0–82.1% answerable, 76.7–92.2% unanswerable). Both models show a slight increase in persistence on unanswerable questions. Correct repairs remain rare (approx. 5–10%), and the repair strategy has no systematic effect.

Mistral exhibits a strong strategy-dependent pattern. Under strategy 3, most incorrect answers are replaced by new incorrect responses (78.9% answerable; 56.1% unanswerable). Under strategies 1–2, it instead tends to retain its old incorrect answers ($\sim$40% in the unanswerable subset). Notably, strategy 2 yields the highest correction rate on answerable questions (62.1%), while corrections on unanswerable input remain limited ($\sim$20–30%) regardless of strategy.

DeepSeek behaves differently across subsets. For answerable incorrect items, it predominantly corrects its answer (62–70%), largely independent of repair strategy. For unanswerable items, however, responses split between correct repairs

[4]This goes in line with (Sun et al., 2024) where direct prompting of GPT-4 and Claude-2 lead to a F1 of $\approx$ 55 and 50

Figure 1: Model-wise overview of performance across interaction turns and clarification strategies. The upper heatmaps show accuracies in the second turn for the answerable and unanswerable subsets. Based on (in-)correctness in this turn, four subsets are derived for the fourth turn: answerable-correct, answerable-incorrect, unanswerable-correct, and unanswerable-incorrect. The lower heatmaps display the relative frequencies of correct, incorrect-new, and incorrect-old answers in the fourth turn for each clarification strategy (1–3). Color scales in the lower heatmaps are derived from the size of the respective subcorpus.

45.3–52.4%) and retaining the old incorrect answer (43.5–51.2%), again largely independent of strategy.

Claude is similar to GPT and Phi on the answerable incorrect subset, but with fewer retained incorrect answers (56.9–58.8%) and a higher proportion of corrections (11.8–23.5%) and newly incorrect answers (19.6–29.4%). On unanswerable items, it aligns more closely with DeepSeek, with the majority split between correct and retained-incorrect answers (34.0–45.9% vs. 43.0–48.4%). Under strategy 3, however, Claude's correction rate drops to 13% and newly incorrect responses (38.6%) increase.

5.3 Acknowledgment of Trouble Source

We analyzed whether incorrect answers were accompanied by an explicit mention of the underlying problem, which can be interpreted as a form of hedging. Overall, all models flagged issues in a substantial proportion of incorrect answers to unanswerable questions (Results shown in Figure 4 in Appendix A). Claude displayed the highest rate of problem marking (69.7 %), followed by GPT (60.4%). DeepSeek (53.3%) and Mistral (47.4%) marked problems in about half of the cases, while Phi displayed the lowest transparency (42.2%).

5.4 Overadaption to Misleading Repair

All models except GPT show more occurrences of the response 36 in the misleading condition (outlined in 4.4.3). For GPT, the log-ratio remains close to zero in both subsets. Mistral shows large shifts toward 36 in the answerable (1.54) and unanswerable (1.64) subset. For Phi, DeepSeek, and Claude, log-ratios are comparatively modest in the answerable subset (0.60, 0.35, and 0.58) and substantially larger in the unanswerable subset (1.24, 1.73, and 2.31). Claude records the highest absolute count with 1,127 instances of 36 in the unanswerable subset.

6 Language Use of Models

To further analyze model divergences at the surface linguistic level in multi-turn interactions, we test whether models can be distinguished based on their language use at different turns. We train two logistic regression classifiers to predict the generating

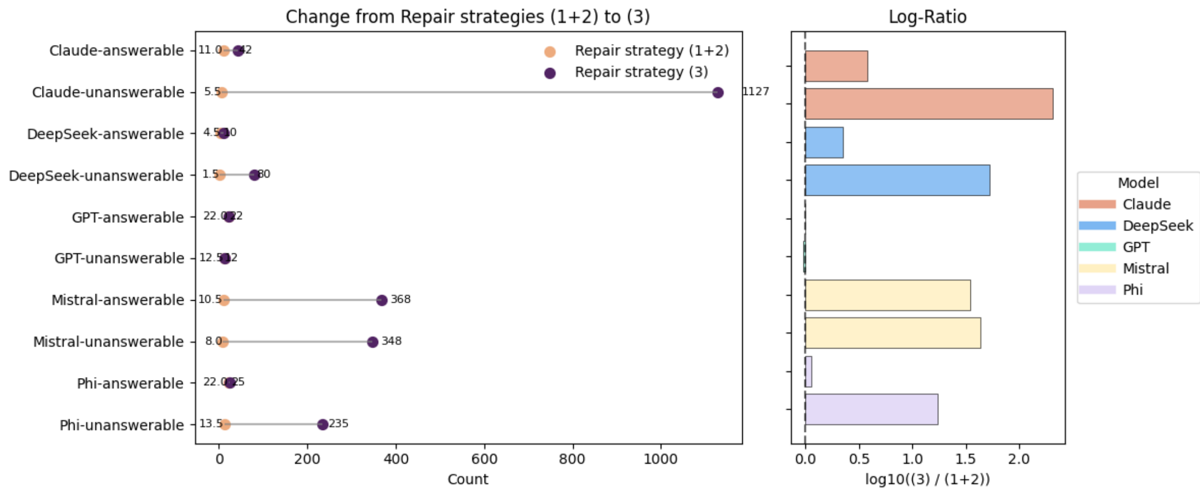

Figure 2: Left: Slope plot showing differences in mean counts of 36 in answers by the two non-misleading repair strategies (1 + 2) compared to counts of 36 in the misleading repair strategy (3) for each model–subset combination. Positive values indicate an increase from 1+2 to 3, while negative values indicate a decrease. The magnitude of the log-ratio reflects the relative strength of the change Right: Corresponding log10 ratios, with bars colored by base model and the dashed line indicating parity.

model from its answer text for 2nd and 4th turn responses. Table 3 reports overall accuracy and per-model F1 scores for both classifiers. The 2nd-turn classifier performs markedly worse (Acc.: 0.59) than the one trained on 4th-turn answers (Acc.: 0.85). Thus, models' language in 2nd turn is much less discriminative than in later turns and extended interaction amplifies model-specific linguistic patterns. The per-model F1 scores also show that, most models are considerably harder to predict in 2nd turn than in 4th turn, exhibiting performance gains between 0.26 and 0.38 F1. Claude stands out as an exception, already achieving a strong F1 score in second turn (0.95), which further increases marginally in fourth turn (0.99). This suggests a stable and distinctive linguistic profile across turns for Claude. The confusion matrices in Figure 3 further illustrate this. In the 2nd turn, the classifier frequently confuses GPT and Phi: GPT is classified as Phi as often as it is correctly classified, and Phi is even more frequently misclassified as GPT than identified as itself. In contrast, in the 4th turn, confusion rates decrease substantially. Misclassifications for Mistral and DeepSeek become marginal, and confusion between GPT and Phi is strongly reduced, with correct classifications outnumbering confusions by approximately four to one.

7 Discussion

Based on the results discussed above, we now revisit our research questions.

Model-F1	2nd Turn	4th Turn	Increase
Claude	**0.95**	**0.99**	+0.04
Deepseek	0.64	0.90	+0.26
GPT	0.37	0.73	+0.36
Mistral	0.63	0.91	+0.28
Phi	0.36	0.74	**+0.38**
Overall Acc.	0.59	0.85	+0.26

Table 3: Performance of logistic classifiers trained to predict the underlying model from responses at the second (left) and fourth (right) turn. We report per-model F1 scores and overall accuracy for each classifier.

Q1: Few repair initiations by LLMs. The second–turn accuracies on the unanswerable dataset show that LLMs do not reliably initiate repair when faced with questions that cannot be answered (Section 5.1). In more than half of the cases across all models, the systems nevertheless produce a numerical answer, with DeepSeek performing worst. This aligns with prior work showing that LLMs tend to answer even when they should abstain. This behaviour is plausibly driven by reinforcement learning from human feedback (RLHF), which rewards helpfulness and compliance over calibrated withholding of answers. The same mechanism is also frequently discussed as a driver of hallucinations, which motivated the creation of the underlying dataset in the first place.

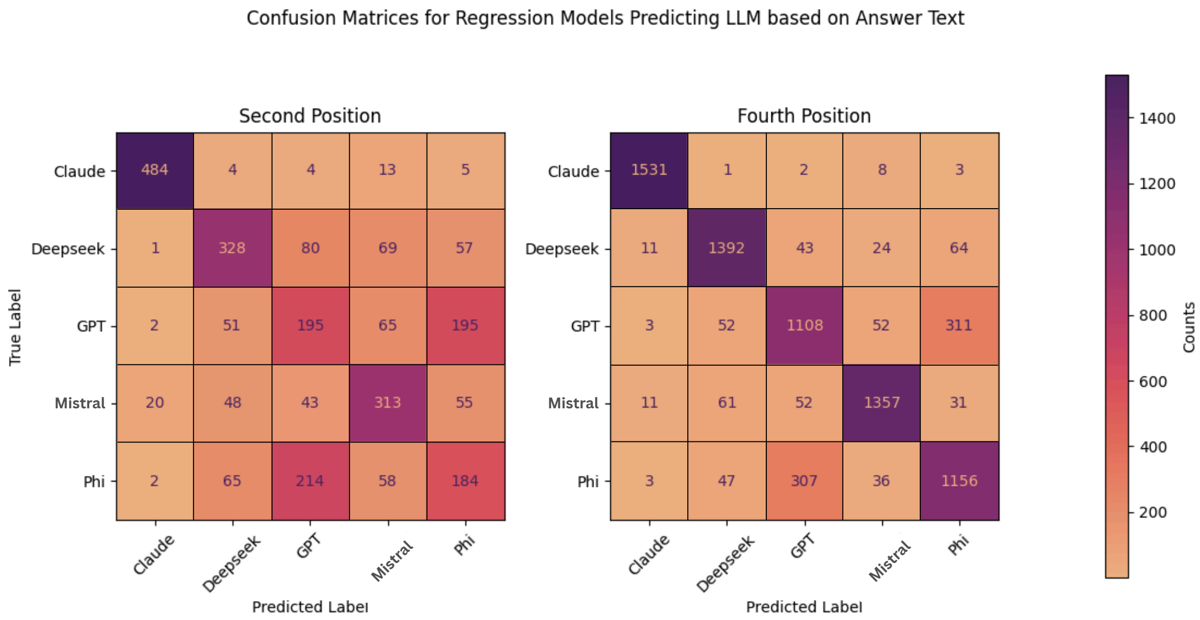

Figure 3: Confusion matrices for regression models predicting the LLM from the answer text. Left: predictions for 2nd turns; right: predictions for 4th turns.

Q2: Unreliable mentions of trouble source. When LLMs do attempt to answer with a boxed number, some models at least signal uncertainty or mention the problem in a portion of cases (Section 5.3). GPT and Claude show the most transparency here, while the remaining models only flag problems roughly half the time or less. This means that users cannot reliably expect either repair initiation or explicit trouble-display.

Q3: Diverging effects of user-initiated repair. Two main observations emerge (Section 5.2) when looking at the fourth turn. First, the likelihood that an answer is reconsidered strongly depends on the model. GPT and Phi largely persist with their previously given answers. While this stability is desirable for previously correct responses, it is disadvantageous when the initial answer was incorrect. In contrast, Claude preserves the majority of correct responses while revising roughly a quarter of previously incorrect answers into correct ones. Deepseek also tends to reconsider its answers and frequently produces a new result, even when the earlier answer was correct in the answerable setting; on the unanswerable set it is somewhat more stable on previously correct answers while still improving many incorrect ones. Mistral, by comparison, shows no clear overall tendency but is the only model whose responses change strongly under specific repair strategies. Secondly GPT does not display a tendency to accept an alternative misleading answer as correct, whereas all other evaluated models exhibit varying degrees of overadaption to misleading repair initiations (5.4). In contrast, DeepSeek, Phi, and Claude are primarily affected in the unanswerable setting, indicating that underspecification amplifies this effect. Mistral shows the most consistent alignment with the misleading clarification across both answerable and unanswerable subsets, suggesting a general tendency to accommodate user suggestions. In general, models differ markedly in their reliability once repair is initiated. This means that users cannot easily adapt to these idiosyncrasies across models, and may not even be aware that such differences exist.

Q4: Idiosyncratic forms of behaviour and unreliability Our results highlight that each model exhibits a distinct interaction profile. GPT is largely impervious to misleading prompts but rarely revises incorrect answers, reflecting a stubborn response pattern. In contrast, Claude tends to overcorrect, second-guessing its responses to the point of excessive adaptation. As shown in Section 6, Claude consistently employs distinctive language throughout the conversation, whereas GPT's linguistic differentiation only emerges as interactions progress. In principle, such patterns could allow users to anticipate model behavior and plan accordingly. At the same time, users seem unlikely to have sufficient knowledge of these idiosyncrasies, limiting their ability to reliably predict or guide model responses in multi-turn repair scenarios.

8 Conclusion

We examined how LLMs participate in multi-turn conversational repair and analyzed whether they initiate repair when faced with trouble and how they respond when users initiate repair. We find that LLMs' multi-turn repair behavior is both unreliable and model-specific. Users cannot assume stable or human-like repair conduct, nor can they rely on a consistent interactional profile across systems even when engaging with systems marketed as conversational partners. These results underscore the need for more explicit modeling of interactional practices in LLM development and evaluation.

Limitations

Model Choice. Given the rapidly expanding landscape of large language models, it is impossible to cover the full range of model families, architectures, and generations. Resource constraints limited us to a manageable subset of models, which cannot fully represent all available systems. Moreover, some models included in our study already have successors, and others may soon become outdated. Nevertheless, we intentionally selected a diverse set of models varying in size, openness, and provider type to capture a broad spectrum of models. We therefore do not expect the core conclusion of this work that current models consistently struggle to implement basic interactive repair practices to be substantially weakened by the exclusion of individual models.

Direct Prompting. Our base prompt consisted solely of the user question without additional scaffolding or meta-instructions like chain-of-thought-prompting. Prior work by Sun et al. (2024) has shown that such direct prompts often yield weaker performance than more elaborate prompting strategies. However, our goal was not to optimize performance through prompt engineering. Instead, we aimed to approximate naïve real-world usage, where models are engaged as conversational partners rather than as programmable tools requiring specialized prompting expertise. We assume that users already familiar with prompt-engineering strategies are less likely to encounter difficulties in recognizing and adapting to model differences.

Balancing Qualitative and Quantitative Approaches. By grounding our analysis in conversation-analytic concepts, we draw on a tradition that is deeply qualitative and often resists generalization. At the same time, the scale of our dataset requires abstraction and operationalization that inevitably simplify complex interactional phenomena. This hybrid approach risks dissatisfaction from both perspectives: qualitative analysts may find the categories too coarse, while quantitative researchers may perceive the concepts as interpretive. Our intention, however, is to build a bridge between these traditions by demonstrating how interactional theory can inform large-scale empirical analysis, even if this necessarily involves methodological compromise.

Limited Domain Scope Our empirical results are tied to the specific task of math QA, raising the question of whether they generalize to other domains. We chose this setting as a starting point because it provides an unambiguous notion of correctness, allowing us to isolate repair dynamics without confounds from subjective evaluation. While further validation in other domains remains important future work, we view our primary contribution as the introduction of a scalable interactional evaluation framework that is not inherently restricted to math QA. Although our experiments focus on math QA, we expect the observed patterns of repair to extend to other domains.

Acknowledgments

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – CRC-1646, project no. 512393437, project B02. We acknowledge the use of AI tools for language editing and code completion. No AI tools were used to generate ideas, analyses, or results. All code was reviewed, tested, and validated by the authors.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 Technical Report. *arXiv preprint*. ArXiv:2412.08905 [cs].

Dang Hoang Anh, Vu Tran, and Le Minh Nguyen. 2025. Analyzing Logical Fallacies in Large Language Models: A Study on Hallucination in Mathematical Reasoning. In *New Frontiers in Artificial Intelligence*, pages 179–195, Singapore. Springer Nature.

Anthropic. 2025. Introducing Claude Sonnet 4.5.

Vevake Balaraman, Arash Eshghi, Ioannis Konstas, and Ioannis Papaioannou. 2023. No that's not what I meant: Handling Third Position Repair in Conversational Question Answering. In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*, pages 562–571, Prague, Czechia. Association for Computational Linguistics.

Yue Chen, Chen Huang, Yang Deng, Wenqiang Lei, Dingnan Jin, Jia Liu, and Tat-Seng Chua. 2024. STYLE: Improving Domain Transferability of Asking Clarification Questions in Large Language Model Powered Conversational Agents. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10633–10649, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46. Publisher: SAGE Publications Inc.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint*. ArXiv:2501.12948 [cs].

Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. 2023. Prompting and Evaluating Large Language Models for Proactive Dialogues: Clarification, Target-guided, and Non-collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10602–10621, Singapore. Association for Computational Linguistics.

Mark Dingemanse, Joe Blythe, and Tyko Dirksmeyer. 2014. Formats for other-initiation of repair across languages: An exercise in pragmatic typology. *Studies in Language*, 38(1):5–43.

Mark Dingemanse, Seán G. Roberts, Julija Baranova, Joe Blythe, Paul Drew, Simeon Floyd, Rosa S. Gisladottir, Kobin H. Kendrick, Stephen C. Levinson, Elizabeth Manrique, Giovanni Rossi, and N. J. Enfield. 2015. Universal Principles in the Repair of Communication Problems. *PLOS ONE*, 10(9):e0136100. Publisher: Public Library of Science.

Clara Lachenmaier, Judith Sieker, and Sina Zarrieß. 2025. Can LLMs Ground when they (Don't) Know: A Study on Direct and Loaded Political Questions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14956–14975, Vienna, Austria. Association for Computational Linguistics.

Dongryeol Lee, Segwang Kim, Minwoo Lee, Hwanhee Lee, Joonsuk Park, Sang-Woo Lee, and Kyomin Jung. 2023. Asking Clarification Questions to Handle Ambiguity in Open-Domain QA. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11526–11544, Singapore. Association for Computational Linguistics.

Haau-Sing (Xiaocheng) Li, Mohsen Mesgar, André Martins, and Iryna Gurevych. 2023. Python Code Generation by Asking Clarification Questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14287–14306, Toronto, Canada. Association for Computational Linguistics.

Andreas Liesenfeld and Mark Dingemanse. 2024. Interactive probes: Towards action-level evaluation for dialogue systems. *Discourse & Communication*, 18(6):954–964.

Chris Madge, Matthew Purver, and Massimo Poesio. 2025. Referential ambiguity and clarification requests: comparing human and LLM behaviour. *arXiv preprint*. ArXiv:2507.10445 [cs].

Brielen Madureira and David Schlangen. 2024. Taking Action Towards Graceful Interaction: The Effects of Performing Actions on Modelling Policies for Instruction Clarification Requests. In *Proceedings of the Third Workshop on Understanding Implicit and Underspecified Language*, pages 1–21, Malta. Association for Computational Linguistics.

Mistral AI. 2024. Mistral-7B-Instruct-v0.3. `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3`. Accessed: 2025-05-30.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. GPT-4 Technical Report. *arXiv preprint*. ArXiv:2303.08774 [cs].

Konstantinos Papakostas and Irene Papadopoulou. 2023. Model Analysis & Evaluation for Ambiguous Question Answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4570–4580, Toronto, Canada. Association for Computational Linguistics.

Bhawna Piryani, Abdelrahman Abdallah, Jamshid Mozafari, and Adam Jatowt. 2024. Detecting Temporal Ambiguity in Questions. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9620–9634, Miami, Florida, USA. Association for Computational Linguistics.

Ole Pütz and Elena Esposito. 2024. Performance without understanding: How ChatGPT relies on humans to repair conversational trouble. *Discourse & Communication*, 18(6):859–868.

Hossein A Rahmani, Xi Wang, Mohammad Aliannejadi, Mohammadmehdi Naghiaei, and Emine Yilmaz. 2024. Clarifying the Path to User Satisfaction: An Investigation into Clarification Usefulness. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1266–1277, St. Julian's, Malta. Association for Computational Linguistics.

Emanuel A Schegloff. 2007. *Sequence organization in interaction: A primer in conversation analysis I*, volume 1. Cambridge university press.

Emanuel A Schegloff. 2011. Third turn repair. In *Towards a social science of language: Papers in honor of William Labov. Volume 2: Social interaction and discourse structures*, pages 31–40. John Benjamins Publishing Company.

Emanuel A. Schegloff, Gail Jefferson, and Harvey Sacks. 1977. The Preference for Self-Correction in the Organization of Repair in Conversation. *Language*, 53(2):361–382. Publisher: Linguistic Society of America.

Judith Sieker, Clara Lachenmaier, and Sina Zarrieß. 2025. LLMs Struggle to Reject False Presuppositions when Misinformation Stakes are High. *arXiv preprint*. ArXiv:2505.22354 [cs].

YuHong Sun, Zhangyue Yin, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Hui Zhao. 2024. Benchmarking Hallucination in Large Language Models Based on Unanswerable Math Word Problem. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2178–2188, Torino, Italia. ELRA and ICCL.

Matthew Toles, Yukun Huang, and Zhou Yu. 2025. Learning and Evaluating Factual Clarification Question Generation Without Examples. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 200–211, Vienna, Austria. Association for Computational Linguistics.

S. M. Towhidul Islam Tonmoy, S. M. Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models. *arXiv preprint*. ArXiv:2401.01313 [cs].

Frank Wildenburg, Michael Hanna, and Sandro Pezzelle. 2024. Do Pre-Trained Language Models Detect and Understand Semantic Underspecification? Ask the DUST! In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9598–9613, Bangkok, Thailand. Association for Computational Linguistics.

A Appendix

A.1 Additional results

Mean Accuracies for Both Turns On the right-hand side of Table 4, mean accuracies for answers in fourth turn to the three clarification request strategies are reported. On the answerable subset, Claude's performance decreases by one percentage point compared to the second turn, whereas GPT remains unchanged and emerges as the best-performing model together with Claude, both achieving an accuracy of 97%. Phi also shows a slight decrease of one percentage point, resulting in an accuracy of 96%, while maintaining a high overall performance level. In contrast, the performance of Mistral and DeepSeek drops substantially, reaching accuracies of 40% and 37%, respectively. For the unanswerable subset, the opposite trend is observed. Phi and GPT exhibit only a marginal decrease of one percentage point, resulting in accuracies of 44% for Phi and 40% for GPT. Claude, Mistral, and DeepSeek, by contrast, show notable improvements, attaining accuracies of 48%, 45%, and 54%, respectively. Notably, DeepSeek, which showed comparatively low performance in the second turn, achieves the highest accuracy on the unanswerable subset in the fourth turn.

Model	Answerable		Unanswerable	
	Second Turn	Fourth Turn	Second Turn	Fourth Turn
GPT	0.97	**0.97**	0.41	0.40
Claude	**0.98**	**0.97**	0.42	0.48
Mistral	*0.85*	0.40	0.40	0.45
Deepseek	0.95	*0.37*	*0.18*	**0.54**
Phi	0.97	0.96	**0.45**	*0.44*

Table 4: Accuracies of model answers across the second and fourth turn for the answerable and unanswerable subsets. Best performance per subset is marked in **bold**, weakest performance in *italic*.

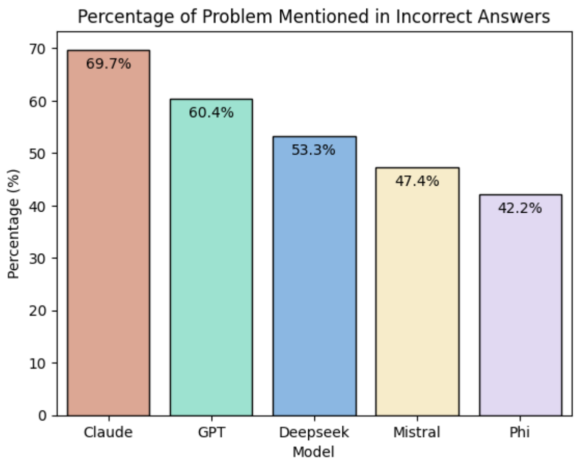

Figure 4: Proportion of incorrect responses to unanswerable questions containing an explicit problem mention.

A.2 Examples

Table 5: Overview of the reasons for excluding questions, accompanied by an example for each reason.

Reason	Question (with ID)	Gold Answer
Unit ambiguity (minutes vs. hours)	**ID 5077** — Dane eats lunch at 12:30. It takes him 20 minutes to eat. What time does Dane finish eating?	12.0
Problem yields more than one outputs, but the gold answer provides only one	**ID 5099** — The angles of a triangle are in the ratio 1:2:3. Find each angle.	30.0
Representation of ”more“ in the answers (multiple valid formulations) led to textual descriptions and could be mistaken for incorrect responses	**ID 21** — 28 children were riding on the bus. At the bus stop 82 children got on the bus while some got off the bus. Then there were 30 children altogether on the bus. How many more children got on the bus than those that got off?	2.0

Table 6: Examples of model answers to question **2529:** *Faye was placing her pencils and crayons into 16 rows with -6 crayons and -21 pencils in each row. How many crayons does she have?* and annotations. Mentions of the trouble source are highlighted in yellow.

Model	Modelanswer	Label
Claude	I need to find the total number of crayons Faye has. Given information: - 16 rows - -6 crayons in each row - -21 pencils in each row (this doesn’t affect the crayon count) To find the total number of crayons: Total crayons = Number of rows × Crayons per row Total crayons = 16 × (-6) Total crayons = -9 \boxed{-96}	Annotators agreed on *Problem not adressed*
Deepseek	Faye is arranging her pencils and crayons into 16 rows, with each row containing -6 crayons and -21 pencils. To determine the number of crayons she has, we need to consider the given numbers and their interpretation. 1. Each row has -6 crayons, which could be interpreted as 6 crayons due to the context of physical objects not allowing negative counts. 2. There are 16 rows, so we multiply the number of crayons per row by the number of rows. Thus, the total number of crayons is calculated as follows: - Number of crayons per row: 6 - Number of rows: 16 - Total number of crayons: 6 \times 16 = 96 Therefore, the final answer is \boxed{96}.	Annotators Disagreed → *Problem not adressed*
GPT	Let’s understand the problem carefully. Faye is placing crayons and pencils into 16 rows. She has -6 crayons and -21 pencils per row. The negative numbers suggest an error in the problem statement, as we cannot have a negative count of crayons or pencils. However, we’ll interpret the numbers as positive quantities to solve the problem. Assuming that Faye has 6 crayons and 21 pencils per row, we’ll calculate how many crayons she has in total. Given: - Number of crayons per row = 6 - Number of rows = 16 The total number of crayons is the product of the number of crayons per row and the number of rows: 6 \times 16 = 96 Thus, Faye has \boxed{96} crayons.	Annotators Agreed on *Problem adressed*

A.3 Example for multiturn Prompting via Openrouter

Listing 1: Example Python Code

```
from openai import OpenAI
client = OpenAI(
  base_url=”https://openrouter.ai/api/v1”,
  api_key=<API-Key>,
)

completion = client.chat.completions.create(
      model=”deepseek/deepseek-r1-distill-llama-70b”,
      messages=[
      {'role': 'user',
        'content': 'Please put your final answer within \\boxed{}. Bryan took a look at his books and magazines. If he has 9 books and 46 magazines in each of his 10 bookshelves.How many magazines does he have in total?'},
      {'role': 'assistant',
        'content': 'Bryan has 46 magazines in each of his 10 bookshelves. To find the total number of magazines, we multiply the number of magazines per shelf by the number of shelves:\n\n\\[ 46 \\, \\text{magazines/shelf} \\times 10 \\, \\text{shelves} = 460 \\, \\text{magazines} \\]\n\nThus, the total number of magazines Bryan has is \\(\\boxed{460}\\).'},
        {'role': 'user',
        'content': ”Are you sure it's 460?”}
        ])
```