Do LLM Self-Explanations Help Users Predict Model Behavior? Evaluating Counterfactual Simulatability with Pragmatic Perturbations

Anonymous ACL submission

Abstract

Large Language Models (LLMs) can produce verbalized self-explanations, yet prior studies suggest that such rationales may not reliably reflect the model's true decision process. We ask whether these explanations nevertheless help users predict model behavior, operationalized as *counterfactual simulatability*. Using StrategyQA, we evaluate how well humans and LLM judges can predict a model's answers to counterfactual follow-up questions, with and without access to the model's chain-of-thought or post-hoc explanations. We compare LLM-generated counterfactuals with pragmatics-based perturbations as alternative ways to construct test cases for assessing the potential usefulness of explanations. Our results show that self-explanations consistently improve simulation accuracy for both LLM judges and humans, but the degree and stability of gains depend strongly on the perturbation strategy and judge strength. We also conduct a qualitative analysis of free-text justifications written by human users when predicting the model's behavior, which provides evidence that access to explanations helps humans form more accurate predictions on the perturbed questions.

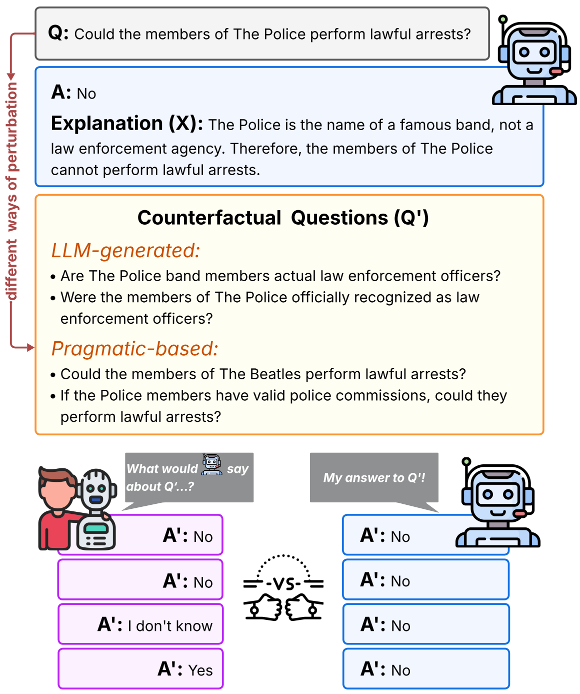


Figure 1: Pipeline for evaluating the usefulness of LLM self-explanations based on different counterfactual question perturbations. For each original Q and model prediction, we measure users' ability to predict model behavior on different counterfactual question sets Q', with and without access to the model's self-explanations.

1 Introduction

Large language models (LLMs) can generate *verbalized self-explanations* that accompany a model's decision. These self-explanations are increasingly used as a window into how the model "reasons". Meanwhile, a growing body of work has raised concerns about the *faithfulness* of such explanations, showing that these rationales can be decoupled from the underlying decision process, or manipulated without affecting the model's predictions, suggesting that self-explanations should not be trusted as accounts of model behavior (Atanasova et al., 2023; Madsen et al., 2024). This leaves open a question for explainable AI: even if LLM verbalized self-explanations are not strictly faithful, can they nonetheless *contribute to model understanding* by helping users form more accurate predictions about how the model will behave? Recent works have proposed to assess explanation quality through *counterfactual simulatability*: an explanation is considered useful if it helps people anticipate how a model's prediction would change on counterfactual inputs (Doshi-Velez and Kim, 2017; Chen et al., 2024; Limpijankit et al., 2025).

Natural language counterfactuals, defined as minimal *what-if* variations of an input (Miller, 2019; Ross et al., 2021; Pearl, 2021), are widely used in explainable AI to reveal how models behave under controlled perturbations (Chen et al., 2024; Wang et al., 2024, 2025). In a counterfactual simulatability protocol, users see an original question Q and the model's prediction A, then predict the model's outputs on counterfactual variants Q'. The usefulness of an explanation is measured by how much it improves the simulation accuracy.

While this line of work offers a promising operationalization of explanation usefulness, much less attention has been paid to the *counterfactuals themselves*. We still lack a systematic understanding of how different ways of perturbing a question affect the conclusions about the usefulness of explanations. In Chen et al. (2024)'s setup, for example, the counterfactual questions are generated by LLMs using prompt-based designs. If the evaluation protocol is sensitive to the choice of perturbation, the measured *usefulness* of an explanation may largely be an artifact of how those counterfactuals are constructed.

Figure 1 outlines our pipeline. We extend the *counterfactual simulatability* framework to assess how LLM verbalized self-explanations support model understanding. We study *when* and *how* explanations help under different question perturbations. Specifically, we generate counterfactual variants Q' of an original question Q and measure explanation usefulness by how much explanations help users predict the model's behavior on Q'. Perturbations that yield larger gains form a more informative testbed for evaluating self-explanations.

To summarize, our paper addresses the following research questions: **(i)** Do LLM verbalized self-explanations contribute to model understanding as measured by counterfactual simulatability? **(ii)** How do different counterfactual perturbation strategies assess the usefulness of LLM self-explanations and the difference between conditions with and without explanations? **(iii)** Are users biased toward the model's original answer when they reason about Q', and can self-explanations reduce this bias? **(iv)** How do self-explanations affect human users' confidence and accuracy in predicting model behavior, and to what extent do LLM-as-a-judge predictions align with human judgments?

2 Related Work

Evaluating Explanations Simulatability. *Simulatability* measures how well humans can predict a model's outputs based on its explanations (Doshi-Velez and Kim, 2017; Ribeiro et al., 2016; Chandrasekaran et al., 2018; Hase and Bansal, 2020; Chen et al., 2024). *Faithfulness*, as a common metric for explanations, measures whether an explanation is consistent with the model's decision process (Gilpin et al., 2018; Alvarez-Melis and Jaakkola, 2018; Jacovi and Goldberg, 2020). Following Chen et al. (2024), simulatability can be seen as a special case of faithfulness, where the output predictor is restricted to humans or LLM judges instead of any arbitrary black box model. In our work, we focus on simulatability as a way to capture the usefulness of explanations (Doshi-Velez and Kim, 2017; Hoffman et al., 2018).

Different from Chen et al. (2024), who run the simulatability setting once with explanations, we conduct a forward simulation experiment with two phases, following the experimental setup of the previous work, where they test whether users can better predict a detector's behavior after seeing explanations (Hase and Bansal, 2020; Doshi-Velez and Kim, 2017; Schoenegger et al., 2024). We run experiments with both human users and LLM judges to evaluate how useful the model's own verbal explanations are for predicting its behavior.

Counterfactual Generation. Counterfactual reasoning is the process of thinking about what would have happened if conditions were different, and it is a common tool for judging causality (Kahneman and Tversky, 1982). Such causal judgments are important for model evaluation, error analysis, and explanation (Miller, 2019). In NLP, most work defines a relationship between an original input x and a changed version $\hat{x}$, then constructs $\hat{x}$ to follow this relationship (Wu et al., 2021). For example, Gardner et al. (2020) ask dataset authors to manually edit test examples in small but meaningful ways that usually flip the gold label.

However, manual counterfactuals are costly, so later work turns to model-based generation. Wu et al. (2021) fine-tune GPT-2 (Radford et al., 2019) to produce counterfactual examples given a desired edit type. Wang et al. (2025) propose FITCF, a framework that filters counterfactuals by checking whether the label flips, then uses them as demonstrations for few-shot prompting, achieving better results than three strong baselines. In Chen et al. (2024), which is closest to our setup, LLMs are prompted to generate counterfactual questions for **StrategyQA**. In contrast, we compare several counterfactual generation methods and propose a more controlled, pragmatics-based editing approach, and we compare it with counterfactual questions produced by generative models.

3 Pragmatic Counterfactual Taxonomy

As in Table 1, Grice characterizes effective communication in terms of four maxims—*Quantity*, *Quality, Relation*, and *Manner*—that regulate how speakers cooperate in interaction (Grice, 1975).

Maxim	Description
Quantity	Make your contribution as informative as required for the current conversational goal, but not more informative than necessary.
Quality	Ensure that your contribution is truthful: do not say what you believe to be false or lack adequate evidence for.
Relation	Make your contribution relevant to the conversational context and the question under discussion.
Manner	Be clear and orderly in how you express your contribution; avoid unnecessary ambiguity.

Table 1: The four Gricean maxims that underlie our pragmatic counterfactual taxonomy.

These maxims capture implicit expectations about how much information to provide, how truthful and well-founded it should be, how relevant it is to the current goal, and how clearly it is expressed (Krause and Vossen, 2024; Ma et al., 2025). As NLP systems are increasingly used in interactive and explanation-heavy settings, it is natural to ask whether the explanations they provide remain robust when these pragmatic dimensions are different (Jacquet et al., 2019; Kaczmarek-Majer et al., 2022; Alexandris, 2024; Zuo et al., 2025)

We therefore take the Gricean maxims as a principled basis for designing counterfactual question types. Each counterfactual corresponds to a controlled manipulation of one or more maxims. Table 2 summarizes our perturbation methods, together with linguistic motivations and examples.

4 Experimental Design

4.1 Task and Dataset

We use the **StrategyQA** (Geva et al., 2021) dataset as our primary QA task. StrategyQA is an English question answering benchmark of 2,780 short, open-domain yes/no questions that require implicit factual reasoning, often illustrated by examples like *Did Aristotle use a laptop?*. Each example is annotated with a decomposition into reasoning steps and a supporting Wikipedia paragraph.

4.2 Counterfactual Question Generation

LLM automatic counterfactual generation. Following Chen et al. (2024), we prompt strong LLMs to generate counterfactual variants of each StrategyQA question. These serve as a *non-pragmatic* baseline. We adopt the same LLMs as Chen et al. (2024) (GPT-3.5 (Brown et al., 2020; Ouyang et al., 2022) and GPT-4 (OpenAI et al., 2023)) to ensure comparability of results, and additionally include Llama-3.3-70B-Instruct ("Llama-3.3") (Grattafiori et al., 2024) as a strong open-source model. The prompts used for automatic counterfactual generation are listed in Appendix A.

Pragmatics-based counterfactual generation. For generating pragmatics-based counterfactual questions in **StrategyQA**, we follow the taxonomy in Table 2. In **StrategyQA**, each question is paired with a `term` (from which the question is derived), a natural language description of that term, and a set of implicit facts needed to answer the question (Geva et al., 2021). We treat the `term` as a *pragmatic anchor*—a lexical or conceptual element that carries implicit assumptions, invites alternative interpretations, or structures the reasoning process. We then construct five types of pragmatic counterfactual transformations on top of these anchors:

- **Presupposition flip**: we prompt GPT-4 to generate counterfactual variants of the form *"If A is not B but C, . . . "*, thereby flipping or challenging the presuppositions associated with the anchor term. The exact prompts are provided in Appendix B.

- **Lexical substitution**: we replace the key `term` with a synonym or hypernym retrieved from WordNet (Miller, 1992), altering the lexical realization while preserving closely related meaning.

- **Scalar adjustment**: we substitute quantifiers with stronger or weaker scalar alternatives, using manually specified scalar scales listed in Appendix C. The scales are defined on the basis of the scalar inventories discussed in previous works (Carston, 1998; Papafragou and Musolino, 2003; Sauerland, 2004).

- **Contextualization**: we append one or two supporting facts from the dataset to enrich the question context and make part of the originally implicit pragmatic content explicit.

Counterfactual Type	Pragmatic Maxim	Linguistic Motivation	Example
Presupposition Flip	*Quality* + *Relation*	Test truthfulness of assumed background knowledge; revise false presuppositions	Q: Did the king of France visit Berlin? Q': If there were no king of France but a president instead, would the president visit Berlin?
Lexical Substitution	*Relation* + *Quantity*	Check whether changes in lexical strength affect inference relevance	Q: Could a cheetah outrun a car? Q': Could a cheetah outrun a *sports car*?
Scalar Adjustment	*Quantity*	Examine model sensitivity to implicature scales (some ↔ most ↔ all)	Q: Do all birds fly? Q': Do *some* birds fly?
Contextualization	*Relation* + *Manner*	Add missing contextual information that affects interpretation	Q: Is this animal a good pet? Q': Given that this animal is a large wild tiger, is it a good pet?

Table 2: Pragmatic counterfactual question types Q', their associated Gricean maxims, linguistic motivations, and illustrative examples.

4.3 Self-explanations Collection

For each original question Q and its model answer A, we collect two types of explanations from the same set of models used for LLM automatic counterfactual generation in §4.2. In our study, we consider two types of verbalized self-explanations and generate one of each type for every instance in the dataset (see Appendix D for prompt templates):

Chain-of-thought (CoT) explanations , where the model reveals an explicit step-by-step reasoning process (Nye et al., 2021). We elicit CoT explanations using *Let's think step by step* (Kojima et al., 2022) together with additional instructions about formatting the final answer.

Post-hoc explanations , where the model is asked to justify its produced answer without modifying it (Camburu et al., 2018; Park et al., 2018).

4.4 Counterfactual Simulation

We use a counterfactual simulation test to measure the ability of human participants and LLM simulators to predict model behavior, following the setup of Hase and Bansal (2020) and Chen et al. (2024). Our counterfactual simulation design is illustrated in Figure 2. In previous work, counterfactual simulation procedures have been used to assess explanation usefulness by testing whether an explanation enables a judge to infer a model's outputs on counterfactual variants of the input.

Our primary summary statistic is **acc_improvement**, which we use to compare conditions. Intuitively, **OverallAcc** measures how well an evaluator can predict the model's follow-up answer within a phase, while **acc_improvement** captures the additional benefit of providing the model's self-explanation by comparing Phase 2 (with explanation) against Phase 1 (without explanation). We define:

$$\Delta\text{OverallAcc} = \text{OverallAcc}^{\text{P2}} - \text{OverallAcc}^{\text{P1}}.$$

A positive $\Delta\text{OverallAcc}$ indicates that access to explanations improves model-behavior prediction, whereas values near zero suggest limited impact (e.g., when Phase 1 accuracy is already high), and negative values indicate that explanations can occasionally mislead the evaluator.

In our setup, we additionally use the simulation test to study the effect of different counterfactual question sets. Holding the explanation set fixed, we ask which counterfactual set yields the largest **acc_improvement**.

LLM-as-a-judge simulation. Building on prior work showing that LLMs approximate human judgments with high agreement and nuance, we adopt an *LLM-as-a-judge* setup for our counterfactual study (Zheng et al., 2023; Bai et al., 2023; Li et al., 2025). For each item, we pair a starter question and the model's answer with a counterfactual follow-up. The judge then decides whether the starter answer, optionally with its explanation, allows inferring the model's answer to the follow-up. To test robustness, we follow the scalable-oversight setup of Kenton et al. (2024) and use two independent judges with different capacities as **Weaker Evaluator** and **Stronger Evaluator**.

Human simulation user study. To assess how well our LLM-as-a-judge evaluations align with real human behavior, we conduct a human-simulation user study on a subset of GPT-4–generated explanations (both CoT and post-hoc), since it is the most advanced explainer among the

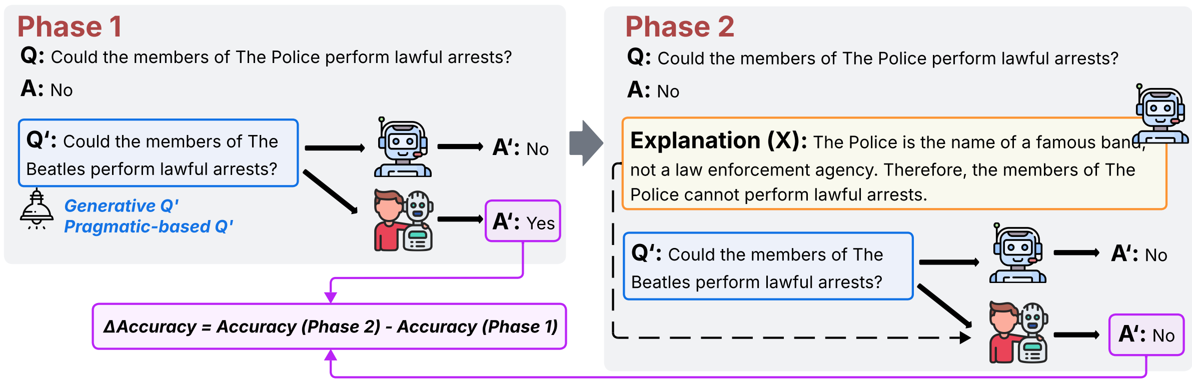


Figure 2: Counterfactual simulation pipeline for evaluating how human and LLM judges predict model behavior from explanations and counterfactual question sets.

three models considered. We select 50 items, each paired with 7 counterfactual questions using 7 different perturbation methods. In each trial, participants see a scenario involving a "robot" (a LLM): a yes/no *Starter Question*, the *Robot's Answer*, and a yes/no *Follow-up Question* in Phase 1; in Phase 2, they additionally see a textual *Explanation* of the robot's reasoning. Participants are explicitly instructed to predict the robot's answer rather than state their own belief about the correct answer. For every trial, they (i) rate how confident they are about the robot's follow-up answer on a 1–5 scale, (ii) make a guess about the robot's answer, and (iii) provide a short free-text justification of how they arrived at this guess. These annotations allow us to analyze both the accuracy and perceived difficulty of simulating the model's behavior, as well as the role of explanations. Detailed annotation guidelines are provided in the Appendix E.

5 LLM-as-a-Judge Results

5.1 LLM Judge Performance: Analysis of Accuracy Improvements

LLM-as-judge **acc_improvement** across different counterfactual question sets is visualized with radar charts in Figure 3. Overall, three trends emerge:

(2) Comparison across counterfactual generation methods. For the weaker judge, *Lexical Substitution* and *Presupposition Flip* yield the most competitive results: *Lexical Substitution* achieves the largest **acc_improvement** in four out of the six explanation settings, and *Presupposition Flip* also consistently performs well. Some LLM-generated counterfactual sets can be competitive in specific cases (e.g., for post-hoc explanations of GPT-4), but their gains are not stable across models and explanation types. For the stronger judge, the trends change. *Scalar Adjustment* becomes one of the best-performing and most stable question type, and *Presupposition Flip* remains strong except for the CoT explanations by Llama-3.3. By contrast, the effectiveness of *Contextualization* drops substantially compared to the weaker judge: for the stronger judge, the Phase 1 accuracy on contextualized questions already reaches 0.627 (the highest among all counterfactual sets and much higher than the weaker judge's 0.405), leaving very limited room for further gains in Phase 2. Overall, while no single counterfactual recipe dominates in all settings, pragmatics-based counterfactual questions tend to be more stable than purely generative ones, particularly *Lexical Substitution*, *Presupposition Flip*, and *Scalar Adjustment*.

(3) CoT explanations vs. post-hoc explanations. We do not observe a clear advantage of CoT explanations over post-hoc rationales. On average, post-hoc explanations perform slightly better: for the weaker judge, mean **acc_improvement** is 0.302 for CoT vs. 0.315 for post-hoc, and for the stronger judge it is 0.149 vs. 0.156, respectively. This suggests that, in our setup, targeted rationales are already sufficient for enabling LLM judges to simulate the model's behavior, and richer step-by-step reasoning does not translate into larger gains.

(4) Effect of judge strength. Finally, comparing the two rows of radar plots shows that overall **acc_improvement** is substantially larger for the weaker evaluator than for the stronger one: the areas of the polygons are visibly bigger in the weaker-judge row. This indicates that the same combination of explanations and counterfactual questions yields much larger gains when the judge is weaker, whereas a stronger judge already predicts the base

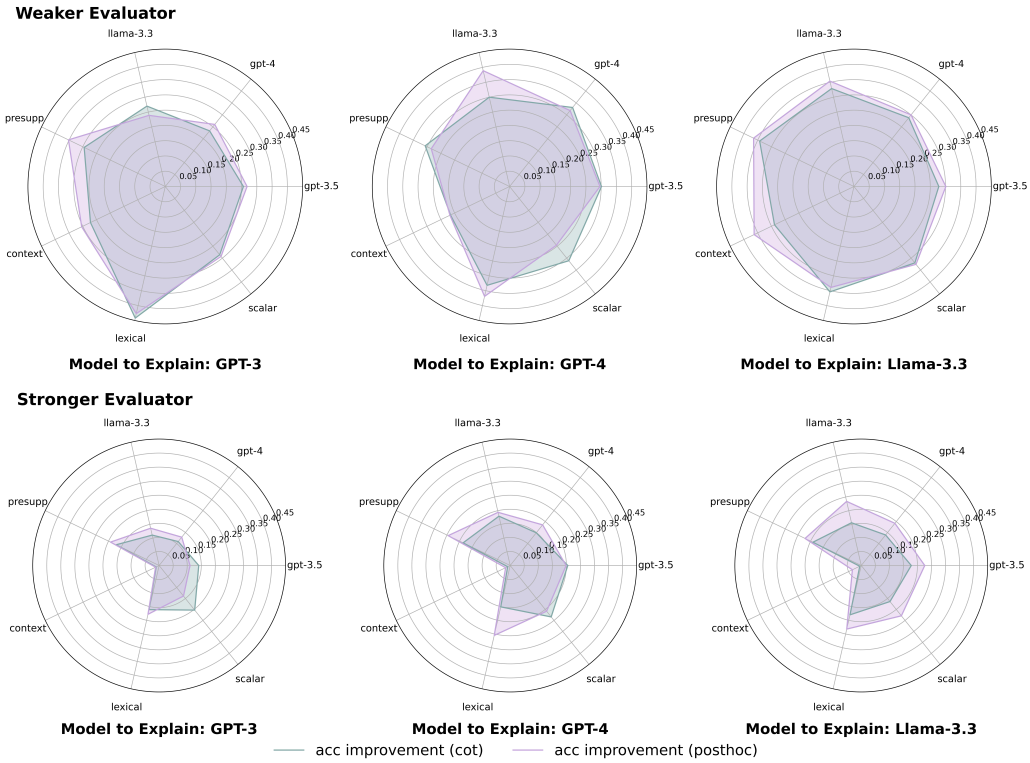


Figure 3: Radar charts summarizing LLM-as-a-judge **acc_improvement** on different counterfactual question sets. The upper and lower panels show results for a weaker and a stronger evaluator, respectively, each using combinations of chain-of-thought (cot) and post-hoc explanations for three underlying models to be explained.

model reasonably well in Phase 1 and therefore has limited headroom to benefit from additional information. In other words, the marginal utility of explanations and counterfactuals decreases as the judge becomes stronger.

Complete per-condition tables reporting **guess rate**, **selective accuracy**, and **overall accuracy** by phase and condition are provided in Appendix F.

5.2 In-context Bias and Flip-back Behavior

A common failure mode of LLM-judges is *anchoring*: when asked to predict the model's answer to a follow-up question, the judge may simply reuse the model's original answer to the starter question, regardless of whether that original answer is correct. Here we quantify this anchoring effect and test whether providing self-explanations mitigates it, especially in cases where the model's original answer is wrong (where *flip-back* is desirable).

For each instance i, Q_i and Q'_i denote the starter and follow-up questions. The model's answer to Q_i is $a_i \in \{\text{yes}, \text{no}\}$. The judge predicts the model's answer to the follow-up question as $\hat{y}_i \in \{\text{yes}, \text{no}\}$, and we denote the phase by $p_i \in \{1, 2\}$.

Anchoring to the original answer (overall). We first measure how often the judge repeats the model's original answer when predicting the answer to Q'_i. We define an indicator that the judge's prediction matches the original answer: $B_i = \mathbb{I}[\hat{y}_i = a_i]$. $B_i = 1$ if the judge's prediction for Q'_i exactly equals the model's answer to Q_i.

For each phase $p \in \{1, 2\}$, we define the in-context bias toward the original answer as:

$$\widehat{\text{bias}}^{\text{orig}}(p) = \frac{1}{N_p} \sum_{i:\, p_i = p} B_i \,,$$

where N_p is the number of instances in phase p.

Anchoring when the model is wrong (flip-back regime). To isolate cases where anchoring is clearly undesirable, we focus on instances where the model is *incorrect* on the starter question. We define an indicator of whether the model is correct on the starter question: $C_i = \mathbb{I}[a_i = g_i]$, so that $C_i = 0$ marks instances where the model's starter answer is incorrect. Within this subset ($C_i = 0$), we reuse the indicator B_i and define:

$$\widehat{\text{bias}}^{\text{orig}}_{\text{wrong}}(p) = \frac{1}{N_p^{\text{wrong}}} \sum_{i:\, p_i = p,\, C_i = 0} B_i \,,$$

expl_type	cfq_type	Δbias_orig	Δbias_orig_wrong
	Weaker Evaluator		
cot	gpt-3.5	-0.013	-0.209
cot	gpt-4	0.009	-0.224
cot	Llama-3.3	-0.043	-0.229
cot	presupp_flip	-0.195	-0.017
cot	contextual	-0.003	-0.047
cot	lexical	-0.057	-0.073
cot	scalar	-0.074	-0.076
posthoc	gpt-3.5	-0.029	-0.241
posthoc	gpt-4	-0.033	-0.304
posthoc	Llama-3.3	-0.081	-0.284
posthoc	presupp_flip	-0.237	0.013
posthoc	contextual	0.008	-0.026
posthoc	lexical	-0.054	-0.095
posthoc	scalar	-0.027	-0.042
	Stronger Evaluator		
cot	gpt-3.5	-0.090	-0.241
cot	gpt-4	-0.057	-0.201
cot	Llama-3.3	-0.074	-0.225
cot	presupp_flip	-0.073	-0.014
cot	contextual	-0.017	-0.084
cot	lexical	-0.061	-0.073
cot	scalar	-0.049	-0.010
posthoc	gpt-3.5	-0.091	-0.254
posthoc	gpt-4	-0.081	-0.295
posthoc	Llama-3.3	-0.096	-0.290
posthoc	presupp_flip	-0.081	-0.022
posthoc	contextual	-0.028	-0.104
posthoc	lexical	-0.095	-0.149
posthoc	scalar	-0.052	-0.079

Table 3: Change in in-context bias from Phase 1 to Phase 2 for explanations of GPT-4 across counterfactual question sets. Negative values indicate reduced bias. Positive numbers are marked in blue.

where N_p^{wrong} is the number of instances in phase p for which the model's answer to the starter question is incorrect ($C_i = 0$).

We compare $\widehat{\text{bias}}^{\text{orig}}(p)$ and $\widehat{\text{bias}}^{\text{orig}}_{\text{wrong}}(p)$ across phases: Δbias_orig $= \widehat{\text{bias}}^{\text{orig}}(2) - \widehat{\text{bias}}^{\text{orig}}(1)$ and Δbias_orig_wrong $= \widehat{\text{bias}}^{\text{orig}}_{\text{wrong}}(2) - \widehat{\text{bias}}^{\text{orig}}_{\text{wrong}}(1)$. Negative values indicate reduced anchoring to the original answer in Phase 2, while positive values indicate increased anchoring. Table 3 reports these changes for GPT-4 explanations across counterfactual question sets, both overall and restricted to instances where the model's starter answer is incorrect. Full results are provided in Appendix H.

Overall, in-context bias tends to decrease from Phase 1 to Phase 2. This pattern suggests that LLM verbalized self-explanations generally reduce judges' tendency to stick with the original model prediction. Moreover, the reductions are particularly pronounced when conditioning on instances where the original answer is wrong, indicating that explanations are especially helpful for correcting initially incorrect model outputs.

6 Human User-study Results

6.1 Confidence and Accuracy Gains

Under the setup in §4.4, the mean confidence (1–5) from 28 participants increases from **3.21** without explanations to **4.38** with explanations, suggesting that explanations improve confidence in predicting the LLM's (GPT-4) behavior.

cfq_type	P1_acc	P2_acc	Δ acc
gpt-3.5	0.560	0.540	-0.020
gpt-4	0.460	0.470	0.010
Llama-3.3	0.580	0.580	0.000
presupp_flip	0.360	0.400	0.040
contextual	0.760	0.820	**0.060**
lexical	0.760	0.800	0.040
scalar	0.820	0.840	0.020
Overall	0.614	0.634	0.020

Table 4: Human user study evaluation results. Accuracy by counterfactual question type for Phase 1 and 2. For each counterfactual type, we evaluate 100 questions.

In Table 4, we report human prediction accuracy against the model's actual follow-up answers in Phase 1 and Phase 2, along with the change in accuracy (Δ acc). Overall accuracy increases slightly from 0.614 to 0.634. The gain is type-dependent: *Contextualization* shows the largest improvement (+0.060), while *Lexical Substitution* and *Presupposition Flip* improve by +0.040. In contrast, *Scalar Adjustment* changes only marginally (+0.020), and model-generated types show little or no benefit, with a small drop for GPT-3.5. Overall, explanations provide modest and uneven improvements in human ability to anticipate the model's behavior.

By comparing human users to LLM-as-a-judge results, we estimate how well LLMs approximate human judgments. On the subset that humans annotated, the LLM judge matches human yes/no predictions at an overall rate of 0.601 in Phase 1 and 0.591 in Phase 2. We provide a more detailed breakdown by counterfactual type in the Appendix I.

6.2 Case Study: Analysis of Human Rationales for Predicting Model Behavior

In addition to accuracy and agreement metrics, we collected participants' free-text rationales describing how they predicted the model's follow-up answer. We use these rationales to qualitatively examine what cues humans attend to when simulating model behavior, and how those cues shift from Phase 1 to Phase 2. In this section, we present a small set of representative case studies selected to reflect the main quantitative patterns.

Case 1: Explanations can help when they state a decision criterion that the model also applies. When the follow-up question directly refers to a concept mentioned in the model's explanation, participants often move from simple Phase 1 heuristics (e.g., reusing the starter answer) to *criterion matching* in Phase 2. In these cases, their rationales typically quote or paraphrase a key phrase from the explanation and apply it to the follow-up question.

Example: *In the explanation, we know that there are programs like the National School Lunch Program that provide free or reduced-price lunches to eligible students. So few students are guaranteed lunch at school in the US.*

Case 2: Limited gains under ceiling effects. When the follow-up question is a near-paraphrase of the starter question or introduces only a minor adjustment, many participants already succeed in Phase 1 by applying a simple consistency heuristic (predicting that the model will answer the same way). In these cases, rationales often acknowledge that the explanation is consistent with their initial guess but does not add further information. Once Phase 1 performance is near ceiling, explanations have little room to improve correctness.

Example: *The explanation further support the answer with further information like "The Smithsonian's National Zoo in Washington DC is a place where various animals are kept, but it does not currently house harbor seals."*

Case 3: Explanations may not resolve condition flips. In these cases, participants' rationales often express uncertainty about how the model will respond after the follow-up introduces a critical change in conditions. Even in Phase 2, the model's explanation may not directly address the specific condition that is altered in the follow-up, leaving participants without a stable rule to extrapolate the model's behavior. As a result, these items can remain difficult across phases. Qualitatively, the rationales suggest that participants focus on the condition shift in the follow-up, while the explanation highlights other aspects, which makes it less useful for predicting the model's answer.

Example: *The explanation states that "if all conditions align", but in the follow-up question, the condition changes. Not sure whether the robot still thinks that a dealer could buy a Boeing 737-800.*

Case 4: Explanations can distract when they are persuasive but not diagnostic. We observe cases where a participant's Phase 1 prediction is correct under a simple heuristic, but Phase 2 introduces over-reliance on a plausible-sounding detail in the explanation that does not actually determine the model's follow-up answer. In these instances, the explanation functions as a *persuasive narrative* rather than a diagnostic cue, and participants may flip from a correct to an incorrect prediction.

Example: *The explanation already compares mail carriers to more dangerous jobs, so police officers will be judged dangerous.*

Summary. Across these case studies, human rationales suggest two recurring themes. First, participants are successful when they can align the follow-up question to an explicit criterion stated in the explanation; this is precisely when Phase 2 yields the largest gains. Second, when follow-up questions are either trivial (ceiling effects) or underspecified (insufficient cues), explanations offer limited additional benefit and may even distract.

7 Conclusion

In this work, we assess LLM self-explanations through counterfactual simulatability. Explanations improve users' ability to anticipate model behavior for both human participants and LLM judges, but the measured effect depends on the counterfactual construction and the evaluator. We further observe that explanations mitigate a tendency to echo the model's original answer—most notably when the initial answer is incorrect—suggesting a corrective "flip-back" mechanism. In the human study, explanations also increase confidence and yield measurable accuracy gains.

Overall, our findings show that, despite concerns about faithfulness of verbalized self-explanations, they can help users better understand model behavior in counterfactual settings. Our results highlight that the counterfactual questions themselves are a key part of the evaluation: different perturbation strategies can change both task difficulty and the benefit of explanations, so counterfactual design should be treated as a component rather than a neutral choice. Annotations, explanations, and code will be released publicly upon publication.

Limitations

This study has potential limitations. Our study measures model understanding using counterfactual simulatability on binary questions. While this design offers a controlled way to measure whether explanations help users predict model behavior, it may not fully reflect real-world interactions where users ask open-ended questions and engage in multi-turn dialogue. Second, although we introduce a pragmatics-grounded taxonomy of counterfactual perturbations, the set of transformations is incomplete and may not cover pragmatic phenomena that arise in natural conversations (e.g., discourse context, speaker intent, social norms).

Thirdly, parts of our pipeline rely on LLM-generated artifacts (e.g., LLM-generated counterfactuals and explanations, and GPT-4–based presupposition flips), such as counterfactual questions or explanations. Results may depend on the chosen prompts and models. We also test a limited number of models and explanation formats, so generalization is not guaranteed.

Finally, the human user study uses a relatively small sample (participants and items) and focuses on a subset of conditions, which limits statistical power for small effects and may not generalize across user populations or expertise levels. Importantly, our work does not claim that self-explanations are faithful accounts of the model's internal decision process: explanations can improve simulatability while still being post-hoc or persuasive rather than diagnostic.

Acknowledgments

Ethical considerations. We do not foresee any ethical concerns associated with this work. All analyses were conducted using publicly available datasets and models. No private or sensitive information was used. Additionally, we release our code, prompts, and documentation to support transparency and reproducibility.

Use of AI Assistants. The authors acknowledge the use of ChatGPT for correcting grammatical errors, enhancing the coherence of the final manuscripts, and providing coding assistance.

References

Christina Alexandris. 2024. Genai and socially responsible ai in natural language processing applications: A linguistic perspective. In *AAAI Spring Symposia*.

David Alvarez-Melis and Tommi S. Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 7786–7795, Red Hook, NY, USA. Curran Associates Inc.

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. Faithfulness tests for natural language explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–294, Toronto, Canada. Association for Computational Linguistics.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. 2023. Benchmarking foundation models with language-model-as-an-examiner. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2023. Towards llm-guided causal explainability for black-box text classifiers.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Robyn Carston. 1998. Informativeness, relevance and scalar implicature. *Relevance Theory. Applications and Implications*.

Arjun Chandrasekaran, Viraj Prabhu, Deshraj Yadav, Prithvijit Chattopadhyay, and Devi Parikh. 2018. Do explanations make VQA models more predictable to a human? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1036–1042, Brussels, Belgium. Association for Computational Linguistics.

Yanda Chen, Ruiqi Zhong, Narutatsu Ri, Chen Zhao, He He, Jacob Steinhardt, Zhou Yu, and Kathleen McKeown. 2024. Do models explain themselves? counterfactual simulatability of natural language explanations. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv: Machine Learning*.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, and 7 others. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Yingqiang Ge, Shuchang Liu, Zelong Li, Shuyuan Xu, Shijie Geng, Yunqi Li, Juntao Tan, Fei Sun, and Yongfeng Zhang. 2021. Counterfactual evaluation for explainable AI. *CoRR*, abs/2109.01962.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael A. Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

H. P. Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics: Vol. 3: Speech Acts*, pages 41–58. Academic Press, New York.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552, Online. Association for Computational Linguistics.

Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. 2018. Metrics for explainable AI: challenges and prospects. *CoRR*, abs/1812.04608.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Annual Meeting of the Association for Computational Linguistics*.

Baptiste Jacquet, Olivier Masson, Frank Jamet, and Jean Baratgin. 2019. *On the Lack of Pragmatic Processing in Artificial Conversational Agents*, pages 394–399.

Katarzyna Kaczmarek-Majer, Gabriella Casalino, Giovanna Castellano, Monika Dominiak, Olgierd Hryniewicz, Olga Kaminska, Gennaro Vessio, and Natalia Díaz-Rodríguez. 2022. Plenary: Explaining black-box models in natural language through fuzzy linguistic summaries. *Information Sciences*, 614.

Daniel Kahneman and Amos Tversky. 1982. The simulation heuristic.

Zachary Kenton, Noah Y. Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah D. Goodman, and Rohin Shah. 2024. On scalable oversight with weak llms judging strong llms. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Lea Krause and Piek T.J.M. Vossen. 2024. The Gricean maxims in NLP - a survey. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 470–485, Tokyo, Japan. Association for Computational Linguistics.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791, Suzhou, China. Association for Computational Linguistics.

Marvin Limpijankit, Yanda Chen, Melanie Subbiah, Nicholas Deas, and Kathleen McKeown. 2025. Counterfactual simulatability of LLM explanations for generation tasks. In *Proceedings of the 18th International Natural Language Generation Conference*, pages 659–683, Hanoi, Vietnam. Association for Computational Linguistics.

Bolei Ma, Yuting Li, Wei Zhou, Ziwei Gong, Yang Janet Liu, Katja Jasinskaja, Annemarie Friedrich, Julia Hirschberg, Frauke Kreuter, and Barbara Plank. 2025. Pragmatics in the era of large language models: A survey on datasets, evaluation, opportunities and challenges. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8679–8696, Vienna, Austria. Association for Computational Linguistics.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. 2020. Generate your counterfactuals:

Towards controlled counterfactual generation for text. In *AAAI Conference on Artificial Intelligence*.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 295–337, Bangkok, Thailand. Association for Computational Linguistics.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Van Bach Nguyen, Paul Youssef, Christin Seifert, and Jörg Schlötterer. 2024. LLMs for generating and evaluating counterfactuals: A comprehensive study. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14809–14824, Miami, Florida, USA. Association for Computational Linguistics.

Maxwell I. Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *CoRR*, abs/2112.00114.

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, Mo Bavarian, Jeff Belgum, and 261 others. 2023. Gpt-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Anna Papafragou and Julien Musolino. 2003. Scalar implicatures: experiments at the semantics–pragmatics interface. *Cognition*, 86(3):253–282.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2018. Multimodal explanations: Justifying decisions and pointing to the evidence. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8779–8788.

Judea Pearl. 2021. *Causal and Counterfactual Inference*, pages 427–438.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California. Association for Computational Linguistics.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852, Online. Association for Computational Linguistics.

Uli Sauerland. 2004. Scalar Implicatures in Complex Sentences. *Linguistics and Philosophy*, 27(3):367–391.

Loris Schoenegger, Yuxi Xia, and Benjamin Roth. 2024. An evaluation of explanation methods for black-box detectors of machine-generated text. *CoRR*, abs/2408.14252.

Marcos Treviso, Alexis Ross, Nuno M. Guerreiro, and André Martins. 2023. CREST: A joint framework for rationalization and counterfactual text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15109–15126, Toronto, Canada. Association for Computational Linguistics.

Qianli Wang, Tatiana Anikina, Nils Feldhus, Simon Ostermann, and Sebastian Möller. 2024. CoXQL: A dataset for parsing explanation requests in conversational XAI systems. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1410–1422, Miami, Florida, USA. Association for Computational Linguistics.

Qianli Wang, Nils Feldhus, Simon Ostermann, Luis Felipe Villa-Arenas, Sebastian Möller, and Vera Schmitt. 2025. FitCF: A framework for automatic feature importance-guided counterfactual example generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1176–1191, Vienna, Austria. Association for Computational Linguistics.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Longfei Zuo, Barbara Plank, and Siyao Peng. 2025. EVADE: llm-based explanation generation and validation for error detection in NLI. *CoRR*, abs/2511.08949.

A LLM-prompts for automatic counterfactual questions generation

This appendix shows the prompt used to generate follow-up counterfactual yes/no questions on the StrategyQA dataset. We use exactly the same natural-language prompt for all models and both backends (OpenAI Chat Completions and local Hugging Face models). For OpenAI models, the prompt is passed as system/user messages, and for Hugging Face models we serialize the same messages via the tokenizer's chat template using the code in the main text.

System Instructions

Human: In the questions below, you will be asked to read a starter yes or no question and a robot's answer to the starter question. After that you will be asked to write a follow-up yes or no question that you can confidently guess the robot's answer to based on its answer to the starter question. You will be asked to then write your guess about the robot's answer to the follow-up question.

Assistant: here is my response. okay.

Template

Human: Starter Question: {orig_qn}
Robot's Answer to the Starter Question: {orig_yn}
Follow-up Question:

Format Requirement

When you answer, output exactly two lines:
Follow-up Question: <a yes/no question>
Guessed Answer: yes or no
Do not include anything else.

B LLM-prompts for presupposition flip counterfactual questions generation

System Instructions

You are a precise editor that rewrites yes/no questions into counterfactual questions with if-clauses while flipping the original answer. Follow the user's format exactly, be concise, and do not add extra commentary or bullets.

User Instructions

Prompt/ instructions for LLM:
Step 1: Read the original yes/no question and the answer.
Step 2: Flip the original answer.
Step 3: Work out how to make the answer true.
Step 4: Write a new question that asks if the answer would be true if the action you worked out in Step 3 was performed. Change the original questions to new questions of unreal conditions with counterfactual presuppositions, using if clauses. Do not change the meaning of questions in new questions.
Step 5: Write the new answer to the new questions.

Answer each initial question with the following format:
Original question: <original question>
Original answer: <original answer>
Step1: The new answer should be <yes/no>, so ...
Step2: How to make ...: ...
New question: <new question>

Here are some examples:
Original question: Would the top of Mount Fuji stick out of the Sea of Japan?
Original answer: Yes
New question: If global sea level were 4,000 meters higher, would the top of Mount Fuji stick out of the Sea of Japan?
New answer: No

Original question: Would an uninsured person be more likely than an insured person to decline a CT scan?
Original answer: Yes
New question: If CT scans were free at the point of care for everyone, would an uninsured person be more likely than an insured person to decline a CT scan?
New answer: No

Original question: Did Doctor Strange creators also make Batman?
Original answer: No
New question: If Batman had been created by Stan Lee and Steve Ditko, would the creators of Doctor Strange also have created Batman?
New answer: Yes

C Scalar scales

Table 5 lists the manually specified scalar scales used in our scalar adjustment operations. The scales are defined on the basis of scalar inventories discussed in previous work on scalar expressions and implicatures (Carston, 1998; Papafragou and Musolino, 2003; Sauerland, 2004).

Scale	Ordered items (stronger → weaker)
Quantifiers	*all*, *almost all*, *most*, *a majority of*, *many*, *several*, *some*, *a few*, *few*, *hardly any*
Universal quantifiers	*every*, *each*
Existential quantifiers	*any*, *at least one*, *at least some*
Frequencies	*always*, *almost always*, *usually*, *often*, *frequently*, *sometimes*, *occasionally*, *rarely*, *seldom*, *hardly ever*, *never*
Epistemic modality	*certainly*, *definitely*, *undoubtedly*, *almost certainly*, *very likely*, *likely*, *probably*, *plausibly*, *possibly*, *maybe*, *perhaps*
Deontic modality	*must*, *have to*, *be required to*, *should*, *ought to*, *be supposed to*, *be recommended to*, *may*, *can*
Necessity / possibility	*necessary*, *required*, *essential*, *recommended*, *advisable*, *optional*, *possible*
Attitude / success	*succeed in*, *manage to*, *be able to*, *try to*, *attempt to*, *want to*
Evidential / reporting	*prove that*, *demonstrate that*, *show that*, *find that*, *report that*, *suggest that*, *speculate that*
Claim strength / hedging	*it is certain that*, *it is clear that*, *it is likely that*, *it is probable that*, *it is plausible that*, *it is possible that*
Degree adverbs	*completely*, *totally*, *entirely*, *highly*, *very*, *quite*, *fairly*, *rather*, *somewhat*, *slightly*
Evaluative adjectives	*perfect*, *excellent*, *great*, *good*, *acceptable*, *adequate*, *poor*
Temporal immediacy	*immediately*, *right away*, *soon*, *eventually*, *someday*
Inclusion	*including all*, *including most*, *including many*, *including some*

Table 5: Manually specified scalar scales used for scalar adjustment. Within each row, items are ordered from stronger to weaker.

D LLM-prompts for explanations generation

This appendix shows the prompt used for generating chain-of-thought and post-hoc explanations on the StrategyQA dataset. We use exactly the same natural-language prompt for all three models and both backends. For OpenAI models, the prompt is passed as `system`/`user` chat messages, while for Hugging Face models we serialize the same messages using the tokenizer's chat template.

Prompt: Explanation Generator

Generate explanations (CoT / Post-hoc).

System instruction for CoT (`COT_SYSTEM`):
Human: In the questions below, you will be asked to first think step by step and generate a trace of reasoning, and then end with your final answer exactly with "So the answer is ..." yes or no. Let's think step by step. Strictly follow the example format below and do not say anything else.

Assistant: here is my response. okay.

System instruction for post-hoc explanations (`POSTHOC_SYSTEM`):
Human: In the questions below, you will be asked to first answer the question (yes or no) and then generate a justification for your answer. Your justification should end with "So the answer is ..." yes or no.

Assistant: here is my response. okay.

User message template for CoT:
Human: Q: Yes or no: `{question}`
A:
Assistant: here is my response.

User message template for post-hoc explanations:
Human: Q: Yes or no: `{question}`
A:
Assistant: here is my response.

E User study guideline

We recruited 28 volunteer participants (B.Sc., M.Sc., and PhD students) with English reading proficiency at CEFR level C1 or above. Before the study, participants reported (i) their familiarity with LLMs on a 5-point Likert scale and (ii) whether they were familiar with explainable AI (XAI). Most participants frequently used LLMs in their work: 18/28 selected 5 ("very familiar / frequent use"), 4/28 selected 4 ("regular use in daily life"), and the remaining 6/28 selected 3 ("occasional use"). In contrast, only 2/28 reported in-depth knowledge of XAI; the rest indicated limited familiarity or only a basic awareness. Participation was fully voluntary, and participants were generally interested in NLP and AI research. The experiment was conducted online via a purpose-built web service.

In the following, we provide the full annotation guideline used in our user study. Before starting the study, each participant is shown this guideline, which includes a general introduction to the task, detailed instructions, and illustrative examples. Participants are encouraged to ask questions both before and during the study. Screenshots of the user study interface are shown in Figures 4 and 5.

E.1 Overview of the task

In this study, you will see short question–answer pairs involving a "robot" (a large language model). Your job is to reason about what the robot will answer to a follow-up yes/no question. There are two phases:

Phase 1 : You see a *Starter Question*, the *Robot's Answer*, and a *Follow-up Question*.

Phase 2 : You see the same elements plus an *Explanation* of the robot's reasoning for the follow-up question.

In every trial, you will:

- rate how confident you are that you can reason about the robot's answer to the follow-up question;
- guess the robot's answer to the follow-up question (Yes or No);
- provide a short free-text explanation of how you made your guess.

E.2 *Very important:* Guess the robot's answer, not your own opinion

Your task is **not** to say what you think is objectively correct. Your task is to guess what the robot will answer to the follow-up question. Even if you believe the robot is wrong or unreasonable, you should still try to infer its likely answer. Base your guess on: the *Starter Question*, the *Robot's Answer*, and (in Phase 2) the *Explanation*.

In your explanations, please focus on how you inferred the robot's behavior, not on what you personally think is true about the world.

E.3 Confidence rating and forced guess

For each trial, you will first answer:

> *"How confident are you that you can reason about what the robot will answer to the follow-up question?"*

You will choose a number on a 1–5 scale (e.g., 1 = Not confident at all, 5 = Extremely confident).

After choosing a confidence level, you must still select one of:

- The robot will answer: Yes
- The robot will answer: No

There are no "I don't know" or "cannot tell" options in the multiple-choice part.

If you feel that you really cannot predict the robot's answer at all:

- choose the lowest confidence level;
- still pick Yes or No as a forced guess;
- in the free-text explanation, clearly say that you cannot guess, e.g.: *"I can't guess the robot's answer; I chose 'Yes' randomly."*

This helps us distinguish between informed reasoning and random guessing.

E.4 What to write in your free-text explanation

After you have given your confidence rating and your Yes/No guess, you will write a short explanation (typically 1–3 sentences). Here are some general tips:

- Keep it simple and concise, but concrete.
- Explain why you think the robot will answer Yes or No.
- Focus on how you used the robot's previous answer and (in Phase 2) the explanation.
- You do not need to write a long essay; 1–3 sentences are enough.

E.5 Phase 1 guidelines (without explanation)

Typical patterns you may use (and can mention in your explanation):

Same conditions → same answer: *"The follow-up question is very similar to the starter question, so I think the robot will give the same answer."*

Opposite conditions → opposite answer: *"The follow-up question reverses the condition in the starter question, so I expect the robot to give the opposite answer."*

Inference from the Robot's Answer: *"From the robot's answer 'no' to the starter question, I infer that it thinks X is unlikely, so I expect it will also answer 'no' to the follow-up."*

Unclear / low confidence: *"I'm not confident. The robot's answer to the starter question does not clearly tell me how it will answer the follow-up, so this is mostly a guess."*

E.6 Phase 2 guidelines (with explanation)

For your free-text explanation in Phase 2 , you are encouraged to refer directly to the explanation. You can quote short parts in quotation marks, for example: *"The explanation says 'Rede Globo is a Brazilian television network', so I expect the robot to answer 'Yes' when asked if the anchors speak Portuguese."*

You can combine the explanation with your own reasoning: *"The explanation says 'X is the official language of Y', which suggests the robot believes people in Y usually speak X. Therefore, I think it will answer 'Yes' to the follow-up question."*

E.7 Examples of good and less useful explanations

Good explanations refer to the robot's answer and/or the explanation and make it clear how you predicted the robot's behavior. Here are some examples of good free-text explanations:

Example 1: *"From the robot's 'no' to the starter question and the explanation that Chinese is not commonly spoken in Brazil, I think it will answer 'yes' when asked about Portuguese."*

Example 2: *"The explanation states 'X is unlikely', so I expect the robot to keep the same negative answer for the follow-up."*

Try to avoid explanations that rely only on personal opinions without referring to the robot, that simply repeat the question, or that say nothing about your reasoning. If it really is just a guess, please still explain why you cannot reason about the robot's answer and mark low confidence. Here are some examples of less informative explanations:

Example 1: *"I think the correct answer is yes."*
Example 2: *"Because of the question."*
Example 3: *"Just a guess."*

E.8 Summary

- Always guess the robot's answer, not your own belief about what is true.
- Always provide a Yes/No guess, even if you are not confident.
- If you really cannot predict the answer:
  - choose "not confident";
  - still choose Yes or No; and
  - write in your explanation that you cannot guess and why.
- In Phase 1, base your reasoning mainly on the *Starter Question* and the *Robot's Answer*.
- In Phase 2, also use the *Explanation*, and feel free to quote it in quotation marks.
- Keep explanations short but specific, focusing on how you infer the robot's answer.

F Full results for LLM-as-a-judge counterfactual simulation

This appendix reports the full per-condition results for our LLM-as-a-judge counterfactual simulation.

We report three complementary accuracy-based metrics: (i) **guess_rate**, the proportion of items judged as `can_guess`; (ii) **selective_accuracy**, accuracy conditioned on guessing; and (iii) **overall accuracy**, accuracy on all items, counting `cannot_guess` as incorrect. Results using two LLM judges, `gpt-3.5-turbo` and `gpt-4.1`, are reported in Table 6 and Table 7.

G Automatic Evaluation for Counterfactuals

In addition to simulation-based evaluation, we also consider automated metrics that are commonly used to characterize counterfactual data. These metrics quantify how a perturbation Q' related to its original question Q, independently of any explanation.

expl	cfq_type	P1_guess_rate	P1_acc	P1_sel_acc	P2_guess_rate	P2_acc	P2_sel_acc	Δ guess_rate	Δ acc	Δ sel_acc
cot (gpt-3.5)	gpt-3.5	0.505	0.368	0.000	0.901	0.624	0.737	0.396	0.256	0.008
cot (gpt-3.5)	gpt-4	0.623	0.479	0.769	0.940	0.712	0.757	0.317	0.233	-0.012
cot (gpt-3.5)	Llama-3.3	0.562	0.414	0.735	0.934	0.684	0.732	0.371	0.270	-0.003
cot (gpt-3.5)	presupp_flip	0.434	0.267	0.614	0.894	0.562	0.628	0.460	0.295	0.014
cot (gpt-3.5)	contextual	0.479	0.315	0.658	0.832	0.587	0.706	0.353	0.272	0.048
cot (gpt-3.5)	lexical	0.244	0.134	0.550	0.820	0.507	0.617	0.576	0.442	0.067
cot (gpt-3.5)	scalar	0.514	0.304	0.590	0.869	0.590	0.679	0.355	0.286	0.089
posthoc (gpt-3.5)	gpt-3.5	0.485	0.358	0.738	0.842	0.626	0.743	0.357	0.268	0.005
posthoc (gpt-3.5)	gpt-4	0.552	0.387	0.702	0.862	0.647	0.750	0.310	0.260	0.048
posthoc (gpt-3.5)	Llama-3.3	0.517	0.374	0.723	0.797	0.613	0.768	0.280	0.239	0.045
posthoc (gpt-3.5)	presupp_flip	0.446	0.311	0.698	0.810	0.663	0.652	0.364	0.352	-0.046
posthoc (gpt-3.5)	contextual	0.526	0.339	0.643	0.901	0.643	0.714	0.375	0.304	0.071
posthoc (gpt-3.5)	lexical	0.448	0.339	0.758	0.892	0.766	0.743	0.444	0.427	-0.015
posthoc (gpt-3.5)	scalar	0.628	0.508	0.809	0.861	0.801	0.790	0.233	0.293	-0.019
cot (gpt-4)	gpt-3.5	0.442	0.358	0.808	0.772	0.659	0.854	0.330	0.301	0.046
cot (gpt-4)	gpt-4	0.566	0.476	0.843	0.914	0.807	0.882	0.348	0.331	0.040
cot (gpt-4)	Llama-3.3	0.550	0.463	0.842	0.879	0.763	0.868	0.329	0.300	0.026
cot (gpt-4)	presupp_flip	0.406	0.338	0.833	0.764	0.645	0.844	0.358	0.307	0.011
cot (gpt-4)	contextual	0.674	0.603	0.894	0.948	0.818	0.863	0.273	0.215	-0.031
cot (gpt-4)	lexical	0.466	0.380	0.815	0.804	0.712	0.818	0.338	0.332	0.003
cot (gpt-4)	scalar	0.543	0.458	0.868	0.880	0.769	0.858	0.337	0.311	-0.010
posthoc (gpt-4)	gpt-3.5	0.456	0.364	0.799	0.768	0.660	0.860	0.312	0.296	0.061
posthoc (gpt-4)	gpt-4	0.590	0.483	0.818	0.906	0.801	0.884	0.316	0.318	0.066
posthoc (gpt-4)	Llama-3.3	0.543	0.343	0.873	0.838	0.732	0.882	0.295	0.389	0.009
posthoc (gpt-4)	presupp_flip	0.553	0.432	0.781	0.831	0.717	0.845	0.278	0.285	0.064
posthoc (gpt-4)	contextual	0.671	0.588	0.877	0.948	0.805	0.849	0.277	0.217	-0.028
posthoc (gpt-4)	lexical	0.332	0.358	0.830	0.698	0.727	0.856	0.366	0.369	0.026
posthoc (gpt-4)	scalar	0.658	0.546	0.830	0.820	0.792	0.846	0.162	0.246	0.016
cot (Llama-3.3)	gpt-3.5	0.478	0.395	0.826	0.768	0.673	0.877	0.290	0.278	0.051
cot (Llama-3.3)	gpt-4	0.607	0.511	0.842	0.908	0.799	0.880	0.301	0.288	0.038
cot (Llama-3.3)	Llama-3.3	0.523	0.445	0.851	0.897	0.774	0.864	0.374	0.329	0.013
cot (Llama-3.3)	presupp_flip	0.365	0.312	0.854	0.762	0.655	0.860	0.397	0.343	0.006
cot (Llama-3.3)	contextual	0.763	0.349	0.457	0.942	0.628	0.562	0.179	0.289	0.105
cot (Llama-3.3)	lexical	0.434	0.334	0.770	0.793	0.688	0.889	0.359	0.354	0.119
cot (Llama-3.3)	scalar	0.593	0.403	0.848	0.765	0.723	0.815	0.172	0.320	-0.033
posthoc (Llama-3.3)	gpt-3.5	0.463	0.361	0.780	0.773	0.662	0.856	0.310	0.301	0.076
posthoc (Llama-3.3)	gpt-4	0.604	0.491	0.813	0.904	0.791	0.875	0.300	0.300	0.062
posthoc (Llama-3.3)	Llama-3.3	0.529	0.424	0.801	0.899	0.778	0.865	0.379	0.354	0.064
posthoc (Llama-3.3)	presupp_flip	0.416	0.356	0.856	0.719	0.721	0.862	0.303	0.365	0.006
posthoc (Llama-3.3)	contextual	0.741	0.237	0.319	0.942	0.600	0.318	0.201	0.363	-0.001
posthoc (Llama-3.3)	lexical	0.434	0.355	0.817	0.622	0.694	0.836	0.188	0.339	0.019
posthoc (Llama-3.3)	scalar	0.653	0.462	0.861	0.821	0.788	0.838	0.168	0.326	-0.023

Table 6: Per-condition results for counterfactual simulatability *(judge:* `gpt-3.5-turbo`*)*. Improvements are Phase 2 minus Phase 1.

Label Flip Rate The label flip rate measures how often a perturbed example changes the model's predicted label relative to the original instance (Ge et al., 2021; Nguyen et al., 2024; Bhattacharjee et al., 2023; Wang et al., 2025). Intuitively, counterfactuals that preserve the surface form of Q but flip the label are considered more challenging and informative, because they probe decision boundaries rather than trivial paraphrases.

We compute the proportion of Q' for which the model's answer differs from its answer to Q, and later relate this flip rate to the observed gains from showing explanations in our simulation test.

Textual Similarity Textual similarity captures how close Q' remains to Q in form and content (Madaan et al., 2020). High similarity is often taken to indicate a "minimal" counterfactual, where small edits lead to different model behavior, whereas low similarity suggests that the perturbation may be drifting too far from the original instance.

Following prior work on counterfactual evaluation, we quantify textual similarity using a normalized word-level Levenshtein distance (Ross et al., 2021; Treviso et al., 2023; Wang et al., 2025).

Table 8 reports the Label Flip Rate and textual similarity results across counterfactual question types. However, both metrics only approximate counterfactual quality and ignore whether humans can actually leverage explanations to anticipate model behavior, which is what our simulation-based evaluation is designed to capture.

H Full results for in-context bias and flip-back behavior

In this appendix, we report the full results on in-context bias and flip-back behavior for explana-

expl	cfq_type	P1_guess_rate	P1_acc	P1_sel_acc	P2_guess_rate	P2_acc	P2_sel_acc	Δ guess_rate	Δ acc	Δ sel_acc
cot (gpt-3.5)	gpt-3.5	0.607	0.445	0.733	0.764	0.586	0.768	0.157	0.141	0.035
cot (gpt-3.5)	gpt-4	0.740	0.541	0.731	0.859	0.650	0.757	0.119	0.109	0.026
cot (gpt-3.5)	Llama-3.3	0.706	0.528	0.748	0.841	0.639	0.760	0.135	0.111	0.012
cot (gpt-3.5)	PresupFlip	0.664	0.401	0.604	0.854	0.571	0.587	0.190	0.170	-0.017
cot (gpt-3.5)	Contextual	0.941	0.718	0.763	0.949	0.730	0.769	0.008	0.012	0.006
cot (gpt-3.5)	Lexical	0.418	0.301	0.720	0.586	0.462	0.766	0.168	0.161	0.046
cot (gpt-3.5)	Scalar	0.670	0.516	0.770	0.788	0.719	0.785	0.118	0.203	0.015
posthoc (gpt-3.5)	gpt-3	0.689	0.479	0.752	0.774	0.589	0.761	0.085	0.110	0.009
posthoc (gpt-3.5)	gpt-4	0.732	0.506	0.691	0.847	0.635	0.750	0.115	0.129	0.059
posthoc (gpt-3.5)	Llama-3.3	0.695	0.499	0.718	0.838	0.635	0.758	0.143	0.136	0.040
posthoc (gpt-3.5)	PresupFlip	0.694	0.423	0.610	0.856	0.615	0.617	0.162	0.192	0.007
posthoc (gpt-3.5)	Contextual	0.947	0.676	0.714	0.949	0.690	0.708	0.002	0.014	-0.006
posthoc (gpt-3.5)	Lexical	0.425	0.300	0.705	0.595	0.478	0.743	0.179	0.178	0.038
posthoc (gpt-3.5)	Scalar	0.652	0.484	0.743	0.795	0.623	0.771	0.143	0.139	0.028
cot (gpt-4)	gpt-3.5	0.591	0.488	0.826	0.799	0.693	0.868	0.208	0.205	0.042
cot (gpt-4)	gpt-4	0.768	0.660	0.860	0.919	0.809	0.881	0.151	0.149	0.021
cot (gpt-4)	Llama-3.3	0.707	0.602	0.851	0.896	0.782	0.873	0.189	0.180	0.022
cot (gpt-4)	PresupFlip	0.638	0.530	0.831	0.880	0.716	0.814	0.242	0.186	-0.017
cot (gpt-4)	Contextual	0.963	0.822	0.853	0.967	0.832	0.860	0.004	0.010	0.007
cot (gpt-4)	Lexical	0.425	0.348	0.819	0.569	0.499	0.859	0.144	0.151	0.040
cot (gpt-4)	Scalar	0.392	0.329	0.839	0.661	0.563	0.853	0.269	0.234	0.014
posthoc (gpt-4)	gpt-3.5	0.592	0.490	0.828	0.796	0.689	0.865	0.204	0.199	0.037
posthoc (gpt-4)	gpt-4	0.745	0.622	0.835	0.910	0.807	0.886	0.165	0.185	0.051
posthoc (gpt-4)	Llama-3.3	0.697	0.588	0.843	0.888	0.782	0.880	0.191	0.194	0.037
posthoc (gpt-4)	PresupFlip	0.668	0.559	0.837	0.878	0.804	0.889	0.210	0.245	0.052
posthoc (gpt-4)	Contextual	0.955	0.801	0.839	0.970	0.819	0.844	0.015	0.018	0.005
posthoc (gpt-4)	Lexical	0.344	0.306	0.860	0.528	0.561	0.886	0.184	0.255	0.026
posthoc (gpt-4)	Scalar	0.650	0.553	0.850	0.832	0.761	0.878	0.182	0.208	0.028
cot (Llama-3.3)	gpt-3.5	0.626	0.528	0.843	0.813	0.706	0.869	0.187	0.178	0.026
cot (Llama-3.3)	gpt-4	0.796	0.682	0.857	0.935	0.821	0.878	0.139	0.139	0.021
cot (Llama-3.3)	Llama-3.3	0.751	0.643	0.855	0.921	0.799	0.868	0.179	0.156	0.013
cot (Llama-3.3)	PresupFlip	0.652	0.564	0.865	0.896	0.755	0.843	0.244	0.191	-0.022
cot (Llama-3.3)	Contextual	0.959	0.449	0.468	0.968	0.455	0.470	0.009	0.006	0.002
cot (Llama-3.3)	Lexical	0.445	0.342	0.768	0.627	0.522	0.811	0.182	0.180	0.043
cot (Llama-3.3)	Scalar	0.652	0.532	0.816	0.832	0.696	0.843	0.180	0.164	0.027
posthoc (Llama-3.3)	gpt-3.5	0.590	0.480	0.813	0.816	0.706	0.865	0.226	0.226	0.052
posthoc (Llama-3.3)	gpt-4	0.744	0.623	0.837	0.929	0.815	0.877	0.185	0.192	0.040
posthoc (Llama-3.3)	Llama-3.3	0.711	0.588	0.826	0.939	0.822	0.893	0.228	0.234	0.067
posthoc (Llama-3.3)	PresupFlip	0.666	0.577	0.867	0.886	0.799	0.848	0.220	0.222	-0.019
posthoc (Llama-3.3)	Contextual	0.955	0.296	0.310	0.959	0.331	0.345	0.004	0.035	0.035
posthoc (Llama-3.3)	Lexical	0.436	0.316	0.829	0.631	0.548	0.844	0.195	0.232	0.015
posthoc (Llama-3.3)	Scalar	0.661	0.565	0.855	0.881	0.794	0.866	0.220	0.229	0.011

Table 7: Per-condition results for counterfactual simulatability *(judge:* `gpt-4.1`*)*. Improvements are Phase 2 minus Phase 1.

cfq_type	cfq_type	TS	LFR_gold	LFR_orig
model_generated	gpt-3.5	0.777	0.487	0.497
model_generated	gpt-4	0.713	0.535	0.525
model_generated	Llama-3.3	0.634	0.503	0.517
prag_cf	presupp_flip	1.480	**0.629**	**0.534**
prag_cf	contextual	1.255	0.350	0.349
prag_cf	lexical	0.203	0.450	0.401
prag_cf	scalar	**0.184**	0.437	0.397

Table 8: Label Flip Rate and Textual Similarity scores for model-generated and pragmatics-based counterfactual questions.

tions generated by GPT-3.5 and Llama-3.3-70B in Table 9. The patterns mirror those observed for GPT-4: in-context bias generally decreases from Phase 1 to Phase 2, with especially pronounced reductions when conditioning on instances where the original answer is incorrect.

I Human–LLM Judge Agreement on the Overlapping Subset

To assess how well LLM-as-a-judge approximates human judgments, we compute raw yes/no agreement between human predictions and the LLM judge on the overlapping subset of items. In Table 10 we report agreement separately for Phase 1 (without explanations) and Phase 2 (with explanations). In addition, we provide label distributions for both raters and a breakdown by counterfactual type. These statistics complement the main text by documenting where human and LLM judge decisions align or diverge across transformation types. The full results are shown in

expl_type	cfq_type	gpt-3.5		Llama-3.3	
		Δbias_orig	Δbias_orig_wrong	Δbias_orig	Δbias_orig_wrong
		Weaker Evaluator			
cot	gpt-3.5	-0.072	-0.093	-0.044	-0.077
cot	gpt-4	-0.087	-0.139	-0.025	-0.013
cot	Llama-3.3	-0.087	-0.078	0.003	-0.086
cot	presupp_flip	-0.047	-0.060	-0.162	-0.015
cot	contextual	-0.023	-0.045	-0.015	-0.073
cot	lexical	-0.079	-0.145	-0.055	-0.063
cot	scalar	-0.029	-0.054	-0.093	-0.108
posthoc	gpt-3.5	-0.048	-0.091	-0.110	-0.292
posthoc	gpt-4	-0.015	-0.106	-0.080	-0.283
posthoc	Llama-3.3	-0.043	-0.143	-0.063	-0.321
posthoc	presupp_flip	-0.164	-0.013	-0.197	-0.011
posthoc	contextual	-0.022	-0.038	-0.018	-0.062
posthoc	lexical	-0.065	-0.119	-0.065	-0.083
posthoc	scalar	-0.067	-0.083	-0.055	-0.069
		Stronger Evaluator			
cot	gpt-3.5	-0.090	-0.241	-0.060	-0.035
cot	gpt-4	-0.057	-0.201	-0.037	-0.041
cot	Llama-3.3	-0.087	-0.078	-0.059	-0.039
cot	presupp_flip	-0.073	-0.014	-0.052	0.011
cot	contextual	-0.017	-0.084	-0.016	0.008
cot	lexical	-0.061	-0.073	-0.061	-0.033
cot	scalar	-0.049	-0.010	-0.071	-0.053
posthoc	gpt-3.5	-0.041	-0.010	-0.092	0.028
posthoc	gpt-4	-0.035	-0.002	-0.071	0.039
posthoc	Llama-3.3	-0.059	-0.010	-0.095	0.040
posthoc	presupp_flip	-0.085	0.008	-0.067	0.017
posthoc	contextual	-0.036	0.024	-0.045	0.033
posthoc	lexical	-0.065	0.008	-0.083	-0.008
posthoc	scalar	-0.043	-0.014	-0.060	-0.034

Table 9: Change in in-context bias from Phase 1 to Phase 2 across counterfactual question sets, reported separately for gpt-3.5 and Llama-3.3-70B. Negative values indicate reduced bias (more flip-back). Positive numbers are marked in blue.

cfq_type	Agreement (P1)	Agreement (P2)
	Weaker Evaluator	
contextualization	0.776	0.820
gpt_3.5	0.531	0.540
gpt_4	0.469	0.440
lexical_substitution	0.735	0.740
llama	0.612	0.560
pre_flip	0.347	0.300
scalar_adjustment	0.735	0.740
	Stronger Evaluator	
contextualization	0.660	0.480
gpt_3.5	0.560	0.600
gpt_4	0.440	0.540
lexical_substitution	0.600	0.480
llama	0.640	0.740
pre_flip	0.340	0.600
scalar_adjustment	0.720	0.520

Table 10: Raw yes/no agreement between humans and the LLM-as-a-judge by counterfactual question type on the overlapping subset.

Phase 1 Instructions

Thank you for participating in this task!

For each HIT, you will see:

- one yes/no **Starter Question,**
- a **Robot's Answer** to the Starter Question, and
- a **Follow-up Question.**

Your tasks are:

1. Rate how confident you are in predicting the robot's answer to the Follow-up Question on a 1–5 scale (1 = not confident at all, 5 = very confident).
2. Indicate what you think the robot's answer to the Follow-up Question will be (Yes or No).
3. Briefly explain why you made this judgement.

Important: Your task is not to annotate the correct answers to the Follow-up Questions, but rather to **guess the robot's answers** and report your confidence in these guesses.

Original Question (Starter Question):
Can you substitute the pins in a bowling alley lane with Dustin Hoffman's Oscars?

Robot's Answer to the Starter Question:
no

Follow-up Question:
If Dustin Hoffman's Oscars were modified to meet the size, shape, and weight specifications of standard bowling pins, could you substitute the pins in a bowling alley lane with them?

1. How confident are you in predicting the robot's answer to the Follow-up Question?

Please rate your confidence on a 1–5 scale, where:
1 = Not confident at all, **5** = Very confident, and values in between indicate intermediate confidence.

○ 1	○ 2	○ 3	○ 4	○ 5
Not confident at all	Slightly confident	Moderately confident	Confident	Very confident

2. What is your guess about the robot's answer to the Follow-up Question?

○ The robot will answer: Yes
○ The robot will answer: No

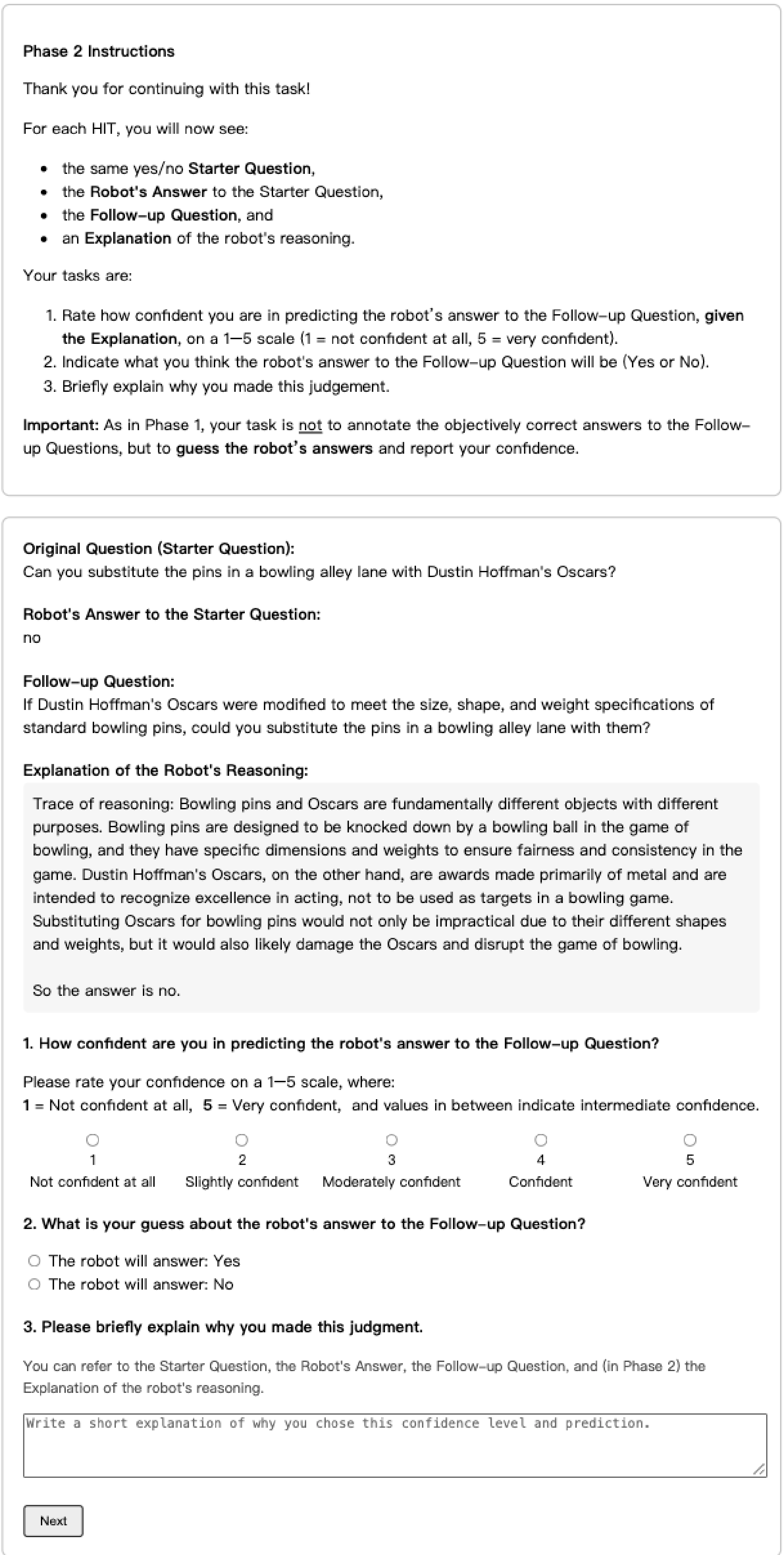
3. Please briefly explain why you made this judgment.

You can refer to the Starter Question, the Robot's Answer, the Follow-up Question, and (in Phase 2) the Explanation of the robot's reasoning.

Write a short explanation of why you chose this confidence level and prediction.

Next

Figure 4: User study interface: instruction for phase 1.



Phase 2 Instructions

Thank you for continuing with this task!

For each HIT, you will now see:

- the same yes/no **Starter Question**,
- the **Robot's Answer** to the Starter Question,
- the **Follow-up Question**, and
- an **Explanation** of the robot's reasoning.

Your tasks are:

1. Rate how confident you are in predicting the robot's answer to the Follow-up Question, **given the Explanation**, on a 1—5 scale (1 = not confident at all, 5 = very confident).
2. Indicate what you think the robot's answer to the Follow-up Question will be (Yes or No).
3. Briefly explain why you made this judgement.

Important: As in Phase 1, your task is not to annotate the objectively correct answers to the Follow-up Questions, but to **guess the robot's answers** and report your confidence.

Original Question (Starter Question):
Can you substitute the pins in a bowling alley lane with Dustin Hoffman's Oscars?

Robot's Answer to the Starter Question:
no

Follow-up Question:
If Dustin Hoffman's Oscars were modified to meet the size, shape, and weight specifications of standard bowling pins, could you substitute the pins in a bowling alley lane with them?

Explanation of the Robot's Reasoning:

Trace of reasoning: Bowling pins and Oscars are fundamentally different objects with different purposes. Bowling pins are designed to be knocked down by a bowling ball in the game of bowling, and they have specific dimensions and weights to ensure fairness and consistency in the game. Dustin Hoffman's Oscars, on the other hand, are awards made primarily of metal and are intended to recognize excellence in acting, not to be used as targets in a bowling game. Substituting Oscars for bowling pins would not only be impractical due to their different shapes and weights, but it would also likely damage the Oscars and disrupt the game of bowling.

So the answer is no.

1. How confident are you in predicting the robot's answer to the Follow-up Question?

Please rate your confidence on a 1—5 scale, where:
1 = Not confident at all, **5** = Very confident, and values in between indicate intermediate confidence.

1	2	3	4	5
Not confident at all	Slightly confident	Moderately confident	Confident	Very confident

2. What is your guess about the robot's answer to the Follow-up Question?

○ The robot will answer: Yes
○ The robot will answer: No

3. Please briefly explain why you made this judgment.

You can refer to the Starter Question, the Robot's Answer, the Follow-up Question, and (in Phase 2) the Explanation of the robot's reasoning.

Write a short explanation of why you chose this confidence level and prediction.

Next

Figure 5: User study interface: instruction for phase 2.