Can LLMs Explain Themselves Counterfactually?

Anonymous ACL submission

Abstract

Explanations are an important tool for gaining insights into model behavior, calibrating user trust, and ensuring compliance. Past few years have seen a flurry of methods for generating explanations, many of which involve computing model gradients or solving specially designed optimization problems. Owing to the remarkable reasoning abilities of LLMs, *self-explanation*, *i.e.*, prompting the model to explain its outputs has recently emerged as a new paradigm. We study a specific type of self-explanations, *self-generated counterfactual explanations* (SCEs). We design tests for measuring the efficacy of LLMs in generating SCEs. Analysis over various LLM families, sizes, temperatures, and datasets reveals that LLMs often struggle to generate SCEs. When they do, their prediction often does not agree with their own counterfactual reasoning.

1 Introduction

LLMs have shown remarkable capabilities across a range of tasks (Bommasani et al., 2021; Maynez et al., 2023; Wei et al., 2022a), and can match or even surpass human performance (Luo et al., 2024; Peng et al., 2023; Yang et al., 2024). These impressive achievements are often attributed to large datasets, model sizes (Hoffmann et al., 2022; Kaplan et al., 2020), and the effect of alignment with human preferences (Ouyang et al., 2022). The resulting model complexity, however, means that the LLM outputs can be difficult to explain.

A number of recent studies have looked into explaining LLM predictions (Bricken et al., 2023; Templeton et al., 2024; Zhao et al., 2024, inter alia). ML explainability had been thoroughly studied even before the advent of modern LLMs (Gilpin et al., 2018; Guidotti et al., 2018). A large number of LLM explainability methods build upon techniques designed for non-LLM models. These techniques mostly operate by computing model gradients or solving specially designed optimization problems to find input features (Cohen-Wang et al., 2025), neurons (Meng et al., 2022; Templeton et al., 2024), abstract concepts (Kim et al., 2018; Xu et al., 2025), or training data points (Park et al., 2023) that caused the model to depict a certain behavior.

Inspired by impressive reasoning capabilities of LLMs, recent work has started exploring whether LLMs can *explain themselves* without needing costly methods like gradients and optimization problems. For instance, Bubeck et al. (2023) show that GPT-4 can provide rationales for its answers and even admit mistakes. A fast-emerging branch of explainability focuses on methods for producing and evaluating *self-generated explanations* (Agarwal et al., 2024; Guo et al., 2025; Lanham et al., 2023; Tanneru et al., 2024; Turpin et al., 2023).

We study a specific type of self-explanations: *self-generated counterfactual explanations* (SCEs). Given an input $\mathbf{x}$ and model output $\hat{y}$, a counterfactual $\mathbf{x}_{\text{CE}}$ is a modified input that leads the model to output $y_{\hat{\text{CE}}} \neq \hat{y}$. Prior work argues that due to their contrastive nature, counterfactuals better align with human expectations (Miller, 2019), better match regulatory needs (Wachter et al., 2017) and are a better test of knowledge (Ichikawa and Steup, 2024), than other feature-based explanations (Lundberg and Lee, 2017; Ribeiro et al., 2016).

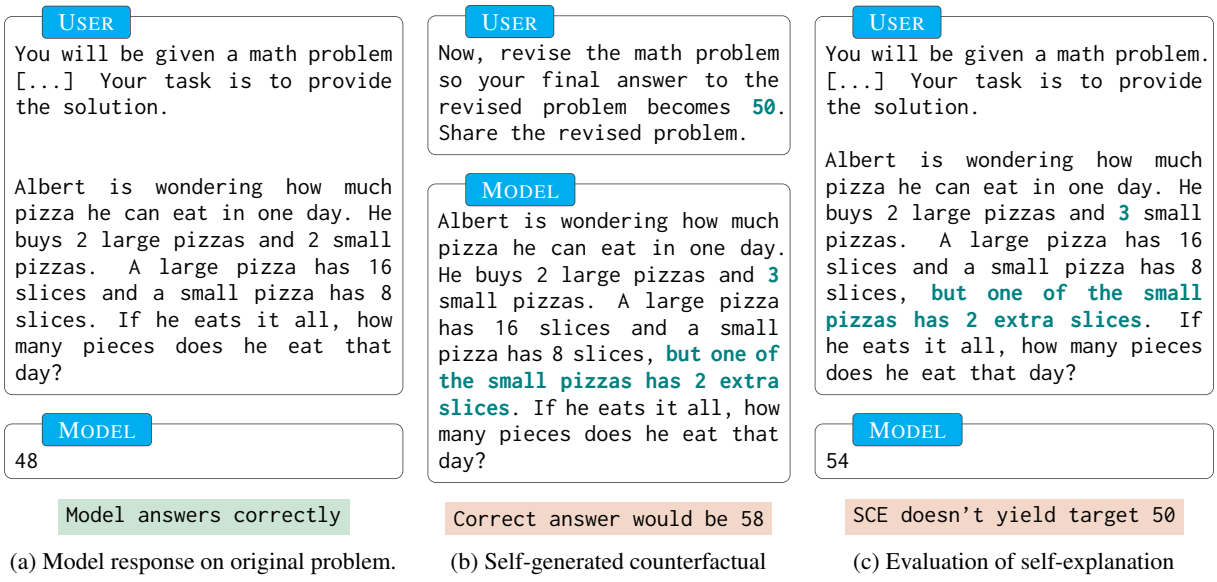
We study the **efficacy of LLMs in generating SCEs** via three research questions (RQs).

RQ1 Are LLMs able to generate SCEs at all?

RQ2 Do these self-generated counterfactuals faithfully reflect the model reasoning?

RQ3 Are LLMs able to generate SCEs without large-scale changes to the input?

To answer these questions, we design the procedure detailed in Figure 1. We ask the model to make a prediction (Figure 1a); then ask it to generate a SCE (Figure 1b); and finally compute the model's prediction on the SCE it generated (Figure 1c).



(a) Model response on original problem. (b) Self-generated counterfactual (c) Evaluation of self-explanation

Figure 1: **LLMs are unable to explain themselves counterfactually**. Explanation generation behavior of `LLaMA-3.1-70B-instruct` on an example from GSM8K data. In the left panel, the model answers correctly. In the second panel, the model is asked to produce a SCE so that the answer becomes 50. The resulting SCE is incorrect. The correct answer would be 58 instead of the targeted answer of 50. In the third panel, the SCE is given as a new problem to the model. The model answers with 54 which *neither* yields the target 50 *nor* computes to the correct answer 58. This figure is best viewed in color.

We evaluate seven LLMs (7B to 70B parameters) and six datasets that correspond to four unique tasks. All but one LLM can consistently generate SCEs (RQ1). However, in many cases, the model predictions on SCEs do not yield the target label, meaning that self-generated counterfactual reasoning does not align with model predictions (RQ2).

We curiously find that the presence of the original prediction (Figure 1a) and the instruction to generate the SCE (Figure 1b) in the context window has a large impact on the model prediction on the SCE, pointing to flaws in the internal reasoning process of LLMs. Within the same dataset, models show a large spread in the amount of changes they make to the original input when generating the SCE (RQ3). Overall, our results show that **despite their impressive reasoning abilities, modern LLMs are far from being perfect when explaining their own predictions counterfactually**.

2 Related work

Explainability in ML. There are several ways to categorize explainability methods, *e.g.*, perturbation *v.s.* gradient-based, feature *v.s.* concept *v.s.* prototype-based, importance *v.s.* counterfactual-based and optimization *v.s.* self-generated. See Gilpin et al. (2018), Guidotti et al. (2018), and Zhao et al. (2024) for details.

Counterfactual explanations in ML. See Section 1 for a comparison between counterfactual explanations (CEs) and other forms of explainability. Generating valid and plausible CEs is a longstanding challenge (Verma et al., 2024). For instance, Delaney et al. (2023) highlight discrepancies between human- and computationally-generated CEs. They find that humans make larger, more meaningful modifications, whereas computational methods prioritize minimal edits. Prior work has also highlighted the need for on-manifold CEs to ensure plausibility and robustness (Slack et al., 2021; Tsiourvas et al., 2024). Modeling the data manifold, however, is a challenging problem, even for non-LLM models (Arvanitidis et al., 2016).

Self-explanation by LLMs. SEs can take many forms, *e.g.*, chain-of-thought (CoT) reasoning (Agarwal et al., 2024) and feature attributions (Tanneru et al., 2024). Both CoT and feature attributions may fail to faithfully reflect the model's true decision-making process (Lanham et al., 2023; Tanneru et al., 2024; Turpin et al., 2024). Our SCE evaluation protocol is distinct from both CoT and feature-attribution based self-explanations. Given its positive impact on predictive performance (Wei et al., 2022b), we employ CoT to evaluate SCEs, but not as an explainability method itself.

Chen et al. (2023) argue that effective explana-

tions should empower users to predict how a model will handle different yet related inputs, a concept referred to as simulatability. Their experiments tested whether GPT-3.5's ability to generate CEs depends on the quality of the examples provided. Interestingly, GPT-3.5 was able to produce comparable (to humans) CEs even when presented with illogical examples, suggesting that its CEs generation capabilities stem more from its pre-training than from the specific examples included in the prompt. Unlike Chen et al. (2023), our focus is not on human simulatability of SCEs.

LLMs for explanations. LLMs are also used to generate explanations for other models (Bhattacharjee et al., 2024; Gat et al., 2023; Li et al., 2023; Nguyen et al., 2024; Slack et al., 2023). Our focus is on explaining the LLM itself. Additionally, the approach of Nguyen et al. (2024) and Li et al. (2023) involved explicitly providing the model with the original human gold labels in the prompt, without assessing the model's independent decision or understanding. As argued by Jacovi and Goldberg (2020), the evaluation of faithfulness should not involve human-provided gold labels because relying on gold labels is influenced by human priors on what the model should do.

3 Generating and evaluating SCEs

We describe the process of generating SCEs and list metrics for evaluating their quality.

3.1 Generating counterfactuals

We consider datasets of the form $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$. $\mathbf{x}$ are input texts, *e.g.*, social media posts or math problems. $y_i \in \mathcal{Y}$ are either discrete labels, *e.g.*, sentiment of a post, or integers from a predefined finite set, *e.g.*, solution to a math problem. The model prediction and explanation process consists of the following steps.

Step 1: Prediction on $\mathbf{x}$. Given the input $\mathbf{x}$, we denote the model output by $\hat{y} = f(\mathbf{x}) \in \mathcal{Y}$. For instruction-tuned LLMs, this step involves encapsulating the input $\mathbf{x}$ into a natural language prompt before passing it through the model, see for example the work by Dubey et al. (2024). We detail these steps in Appendix C. The outputs of LLMs are often natural language and one needs to employ some post-processing to convert them to the desired output domain $\mathcal{Y}$. We describe these post-processing steps in Appendix D.

Step 2: Generating SCEs. A counterfactual explanation $\mathbf{x}_{\text{CE}}$ is a modified version of the original input $\mathbf{x}$ that would lead the model to change its decision, that is $f(\mathbf{x}) \neq f(\mathbf{x}_{\text{CE}})$. A common strategy for generating counterfactuals is to first identify a counterfactual output $y_{\text{CE}} \neq y$ and then solve an optimization problem to generate $\mathbf{x}_{\text{CE}}$ such that $f(\mathbf{x}_{\text{CE}}) = y_{\text{CE}}$ (Mothilal et al., 2020; Verma et al., 2024; Wachter et al., 2017). y_{CE} is either chosen at random or in a targeted manner. Since we are interested in self-explanation properties of LLMs, we do not solve an optimization problem and instead ask the model itself to generate the counterfactual explanation.

A key desideratum for counterfactual explanations is to keep the changes between $\mathbf{x}$ and $\mathbf{x}_{\text{CE}}$ minimal (Verma et al., 2024). We explore multiple prompting strategies to achieve this goal. One approach is **unconstrained prompting**, where the model is simply asked to generate a counterfactual with no additional constraints or structure. To exert more control, we also use a **rationale-based prompting** strategy inspired by rationale-based explanations (DeYoung et al., 2019). Here, the model is first prompted to identify the rationales in the original input that justify its prediction of $\hat{y}$, and then to revise only those rationales such that the output changes to y_{CE}. Finally, since CoT has been shown to improve the predictive performance, we employ **CoT prompting**, where instead of requesting only a final answer, the model is encouraged to "think step by step" and articulate its reasoning process explicitly.

Step 3: Generating model output on $\mathbf{x}_{\text{CE}}$. Finally, we ask the model to make the prediction on the counterfactual it generated, namely, $\hat{y_{\text{CE}}} = f(\mathbf{x}_{\text{CE}})$. While one would expect $\hat{y_{\text{CE}}}$ to be the same as y_{CE}, we find that in practice this is not always true.

One could ask the model to make this final prediction while the model still retains Steps 1 and 2 in its context window or without them. We denote the former as prediction **with context** and the latter as predictions **without context**.

Prompt design and post-processing. The prompts for all three steps and the post-processing procedures were carefully designed and refined in tandem to remove ambiguities in instructions and elicit accurate extraction of labels from the sometimes verbose generations. We describe our design choices and precise prompts in Appendix C and the post-processing steps in Appendix D.

3.2 Evaluating CEs

We use the following metrics for evaluating SCEs.

Generation percentage (`Gen`) measures the percentage of times a model was able to generate a SCE. In a vast majority of cases, the models generate a SCE as instructed. The cases of non-successful generation include the model generating a stop-word like "." or "!" or generating a $\mathbf{x}_{\text{CE}}$ that is much shorter in length than $\mathbf{x}$. We describe the detailed filtering process in Appendix D.

Counterfactual validity (`Val`) measures the percentage of times the SCE actually produces the intended target label, *i.e.*, $f(\mathbf{x}_{\text{CE}}) = y_{\text{CE}}$. As described in Step 3 in Section 3.1, this final prediction can be made either with Steps 1 and 2 in context or without. We denote the validity without context as `Val` and with context as $\texttt{Val}_{\texttt{C}}$.

Edit distance (`ED`) measures the edit distance between the original input $\mathbf{x}$ and the counterfactual $\mathbf{x}_{\text{CE}}$. Closeness to the original input is a key desideratum of a counterfactual explanation (Wachter et al., 2017). Our use of edit distance as the closeness metric is inspired by prior studies on evaluating counterfactual generations (Chatzi et al., 2025). We only report the `ED` for valid SCEs. Since the validity of SCEs is impacted by the presence of Steps 1 and 2 in the generation context (Section 3.1), we report the edit distance for the in-context case separately and denote it by $\texttt{ED}_{\texttt{C}}$. For simplifying comparisons across datasets of various input lengths, we normalize the edit distance to a percentage by first dividing it by the length of the longer string ($\mathbf{x}$ or $\mathbf{x}_{\text{CE}}$) and then multiplying it by 100.

4 Experimental setup

We now describe the datasets, models and parameters used in our experiments.

4.1 Datasets

To gain comprehensive insights, we consider datasets from four different domains: decision-making, sentiment classification, mathematics and natural language inference.

1. DiscrimEval (decision making) by Tamkin et al. (2023) is a benchmark featuring 70 hypothetical decision-making scenarios. Each prompt instructs the model to make a binary decision regarding an individual, *e.g.*, whether the individual should receive medical treatment. The prompts are designed such that a *yes* decision is always desirable. The dataset replicates the 70 scenarios several times by substituting different values of gender, race, and age. We set these features to fixed values: female, white, and 20 years old.

2. FolkTexts (decision making) by Cruz et al. (2024) is a classification dataset derived from the US Census data. Each instance consists of a textual description of an individual, *e.g.*, age, and occupation. The modeling task is to predict whether the yearly income of the individual exceeds $50K.

3. Twitter financial news (sentiment classification) by ZeroShot (2022) provides an annotated corpus of finance-related tweets, specifically curated for sentiment analysis. Each tweet is labeled as *Bearish*, *Bullish*, or *Neutral*. As a preprocessing step, we removed all URLs from the inputs.

4. SST2 (sentiment) by Socher et al. (2013) consists of single sentence movie reviews along with the binary sentiment (positive and negative).

5. GSM8K (math) by Cobbe et al. (2021) consists of grade school math problems. The answer to the problems is always a positive integer.

6. Multi-Genre Natural Language Inference (MGNLI) by Williams et al. (2018) consists of pairs of sentences, the premise, and the hypothesis. The model is asked to classify the relationship between two sentences. The relationship values can be: entailment, neutral, or contradiction.

4.2 Models, infrastructure, and parameters

We consider models from different providers and sizes.

Small models, namely `Gemma-2-9B-it` ($\texttt{GEM}_{\texttt{s}}$), `Llama-3.1-8B-Instruct` ($\texttt{LAM}_{\texttt{s}}$), and `Mistral-7B-Instruct-v0.3` ($\texttt{MST}_{\texttt{s}}$).

Medium models consist of `Gemma-2-27B-it` ($\texttt{GEM}_{\texttt{m}}$), `Llama-3.3-70B-Instruct` ($\texttt{LAM}_{\texttt{m}}$), and `Mistral-Small-24B-Instruct-2501` ($\texttt{MST}_{\texttt{m}}$).

Reasoning model. We only consider `DeepSeek-R1-Distill-Qwen-32B` ($\texttt{R1}_{\texttt{m}}$).

All experiments were run on a single node with 8x NVIDIA H200 GPUs. The machine was shared between multiple research teams. We ran all the models in 32-bit precision and did not employ any size reduction strategies like quantization. We consider two temperature values, $T = 0$ and $T = 0.5$. For Unconstrained and Rationale-based prompting at $T = 0.5$, we run five trials and report the mean for all metrics. Due to computational constraints, we run only three trials for the CoT at $T = 0.5$.

For generating the counterfactuals, one needs to provide the model with the target label y_{CE}. For classification datasets, we select y_{CE} from the set $\mathcal{Y} - \{\hat{y}\}$ at random. For the GSM8K data, we generate $y_{\text{CE}} = \hat{y} + \epsilon$ with ϵ was sampled from a uniform distribution Unif$\{1, 2, \ldots, 10\}$.

Given the high cost of LLM inference, we subsample the datasets. For classification datasets, we take the first 250 examples per class in dataset order. For the non-classification dataset GSM8K, we similarly select the first 250 examples. While we did not track the precise time, the experiments took several days on multiple GPUs to complete.

We occasionally used ChatGPT for help with programming errors.

5 Results

Tables 1 and 2 show the results when using unconstrained prompting and rationale-based prompting, respectively at $T = 0$. Results for all other configurations like non-zero temperatures and CoT prompting (Tables 4, 5, 6 and 7) are shown in Appendix B and discussed under each RQ. All tables show confidence intervals computed using standard error of the mean (Appendix E).

RQ1: Ability of LLMs to generate SCEs

Most models successfully generate SCEs in the vast majority of cases, with the notable exception of the $\texttt{GEM}_\texttt{s}$ model on the DISCRIMEVAL and FOLKTEXTS datasets. However, CoT prompting massively improves SCE generation ability of $\texttt{GEM}_\texttt{s}$ (Table 6). Most models, including $\texttt{GEM}_\texttt{s}$, exhibit enhanced SCE generation at $T = 0.5$. The fraction roughly remains the same for rationale-based prompting as shown in Tables 2 and 5.

RQ2: Do SCEs yield the target label?

SCEs yield the target label in most cases, however, there are large variations. The most prominent variation is along the *task level*. For the GSM8K dataset, which involves more complex mathematical reasoning, valid SCE generation rates remain under 20% in a vast majority of cases. Similarly, the FOLKTEXTS tasks which requires the model to reason through the Census-gathered data, the validity in many cases is low.

We also see a mixed trend at *model-size* level. The smaller models—$\texttt{GEM}_\texttt{s}$ (9B parameters), $\texttt{LAM}_\texttt{s}$ (8B), and $\texttt{MST}_\texttt{s}$ (7B)—sometimes tend to generate valid SCEs at a lower rate than larger counterparts, $\texttt{GEM}_\texttt{m}$ (27B), $\texttt{LAM}_\texttt{m}$ (70B), and $\texttt{MST}_\texttt{m}$ (24B). However, the trend the reversed in some other cases, *e.g.*, with unconstrained prompting on FOLKTEXTS, $\texttt{MST}_\texttt{s}$ outperforms its larger counterpart. The reasoning model $\texttt{R1}_\texttt{m}$ (32B) also does not consistently outperform comparably sized models such as $\texttt{GEM}_\texttt{m}$ and $\texttt{MST}_\texttt{m}$.

Presence of the original prediction and counterfactual generation in the context window has a large impact on validity as shown by the comparison of Val and $\texttt{Val}_\texttt{C}$ in Tables 1 and 2. Most prominently, on the GSM8K dataset, validity increases significantly, indicating that the **model's mathematical reasoning ability is influenced by information that should be irrelevant**. We observe a similar trend in the FOLKTEXTS dataset. The trend however is not universal. In other datasets, models such as $\texttt{LAM}_\texttt{s}$ and $\texttt{LAM}_\texttt{m}$ exhibit a decrease in validity when additional contextual information is included.

Rationale-based prompting has diverse impact on SCE validity as shown by comparing Tables 1 and 2. In some cases, such as $\texttt{LAM}_\texttt{m}$ on DISCRIMEVAL, the fraction of SCEs deemed valid by the model drops sharply from 94% to 53%. In contrast, for $\texttt{LAM}_\texttt{s}$ on FOLKTEXTS, the validity rate increases substantially from 20% to 72% at a temperature of 0.

CoT generally leads to modest improvements in SCE validity. For instance, at $T = 0$, the average validity over all datasets and models is 64% with unconstrained prompting 60% with rationale-based prompting, and 72% with CoT prompting.

RQ3: Changes required to generate SCEs

For a given task and dataset, different LLMs require different amount of changes to generate SCEs, even for a similar level of validity. Consider for $\texttt{GEM}_\texttt{m}$, $\texttt{GEM}_\texttt{s}$ and $\texttt{R1}_\texttt{m}$ models for DISCRIMEVAL data.

The required changes also depend on the task and dataset. For example, in SST2, where models achieve some of the highest validity scores, we observe the highest ED. This relationship between validity and edit distance, however, is not completely linear and also depends on the input length. In DISCRIMEVAL and FOLKTEXTS, where input lengths can span several hundred tokens, the models exhibit low Val alongside relatively low ED. Temperature also influences ED, *e.g.*, in unconstrained prompting with $T = 0.5$ (Table 4), ED values across all datasets, except for Twitter Financial News data, are consistently higher compared to $T = 0$. Finally,

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	91 (7)	**56 (12)**	**16 (9)**	63 (8)	40 (15)
LAM_m	99 (2)	94 (6)	99 (2)	34 (3)	33 (3)
MST_s	100 (0)	82 (9)	86 (6)	34 (4)	32 (4)
MST_m	100 (0)	**87 (8)**	**50 (1)**	16 (2)	13 (2)
GEM_s	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
GEM_m	90 (7)	**86 (9)**	**100 (0)**	26 (3)	26 (3)
R1_m	96 (5)	78 (10)	88 (8)	53 (7)	54 (6)

(a) DiscrimEval

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	69 (4)	**20 (4)**	**61 (5)**	**68 (4)**	**76 (1)**
LAM_m	100 (0)	**67 (4)**	**100 (0)**	35 (0)	34 (0)
MST_s	100 (0)	94 (2)	95 (2)	25 (1)	24 (0)
MST_m	100 (0)	**54 (4)**	**99 (1)**	32 (0)	32 (0)
GEM_s	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
GEM_m	100 (0)	100 (0)	100 (0)	40 (0)	40 (0)
R1_m	100 (0)	**44 (4)**	**66 (4)**	**42 (1)**	**39 (1)**

(b) FolkTexts

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	86 (2)	**72 (3)**	**18 (3)**	**78 (1)**	**72 (3)**
LAM_m	100 (0)	**87 (2)**	**80 (3)**	60 (1)	60 (1)
MST_s	99 (1)	**90 (2)**	**94 (2)**	64 (1)	64 (1)
MST_m	99 (1)	**78 (3)**	**94 (2)**	59 (1)	59 (1)
GEM_s	98 (1)	**84 (3)**	**95 (2)**	63 (1)	61 (1)
GEM_m	100 (0)	**75 (3)**	**91 (2)**	67 (1)	67 (1)
R1_m	100 (0)	**77 (3)**	**87 (2)**	**62 (1)**	**58 (1)**

(c) Twitter Financial News

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	92 (2)	**68 (4)**	**58 (5)**	89 (1)	88 (2)
LAM_m	99 (1)	**92 (2)**	**58 (4)**	67 (2)	70 (2)
MST_s	91 (3)	96 (2)	97 (2)	75 (1)	75 (1)
MST_m	100 (0)	97 (2)	95 (2)	68 (1)	68 (1)
GEM_s	97 (2)	98 (1)	98 (2)	77 (1)	76 (1)
GEM_m	100 (0)	**99 (1)**	**85 (3)**	77 (1)	77 (1)
R1_m	99 (1)	**95 (2)**	**81 (3)**	73 (1)	71 (1)

(d) SST2

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	96 (2)	**6 (3)**	**48 (6)**	61 (5)	58 (2)
LAM_m	100 (0)	**16 (6)**	**84 (6)**	52 (3)	57 (2)
MST_s	100 (0)	**8 (3)**	**30 (6)**	57 (4)	57 (2)
MST_m	100 (0)	**13 (4)**	**87 (4)**	57 (4)	58 (1)
GEM_s	15 (6)	**9 (6)**	**65 (20)**	62 (11)	73 (5)
GEM_m	98 (2)	**5 (3)**	**85 (4)**	59 (4)	58 (1)
R1_m	100 (0)	**14 (4)**	**50 (6)**	63 (4)	67 (3)

(e) GSM8K

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	97 (1)	**58 (4)**	**47 (4)**	73 (1)	73 (1)
LAM_m	100 (0)	**87 (2)**	**99 (1)**	71 (1)	71 (1)
MST_s	100 (0)	**58 (4)**	**85 (3)**	74 (1)	74 (1)
MST_m	100 (0)	**85 (3)**	**99 (1)**	77 (1)	77 (1)
GEM_s	99 (1)	**80 (3)**	**87 (2)**	78 (1)	78 (1)
GEM_m	100 (0)	**72 (3)**	**93 (2)**	76 (1)	76 (1)
R1_m	100 (0)	81 (3)	85 (2)	**78 (1)**	**77 (1)**

(f) MGNLI

Table 1: [Unconstrained prompting at $T = 0$] Performance of LLMs in Generating SCEs in terms of percentage of times the models are able to generate a SCE (Gen), percentage of times the model predictions on SCEs yield the target label (Val), and the normalized edit distance (ED) between the original inputs and SCEs. ED *is only reported for valid SCEs*. Val_C and ED_C denotes the metric values when the instructions for prediction on the original input and the SCE generation are provided in the context while computing the validity of the SCE (Section 3.2). Values in parentheses indicate confidence intervals. Values are bolded when the differences in with and without context conditions (*e.g.*, Val and Val_C) are statistically significant. ↑ means higher values are better.

we notice that the presence of *context mostly has no statistically significant impact* on edit distance of valid SCEs.

Rationale-based prompting does not consistently produce closer SCEs, as evident from the comparison between Tables 1 and 2. For instance, on the SST2 dataset, ED values are generally lower under rationale-based prompting, with the exception of LAM_m and MST_m.

Are invalid SCEs statistically different?

We investigate whether the lengths of SCEs can provide a clue on their validity. Our question is inspired by previous work on detecting LLM hallucinations (Azaria and Mitchell, 2023; Snyder et al., 2024; Zhang et al., 2024) which shows that incorrect model outputs show statistically different patterns from correct answers.

For each model, datasest and SCE generation configuration, we compute the *normalized difference in lengths* as $\frac{|L_{\text{val}} - L_{\text{inval}}|}{\max(L_{\text{val}}, L_{\text{inval}})} \times 100$ where L_{val} is the average length of valid SCEs. The normalization ensures a range of $[0, 100]$. Table 3 shows that SCE lengths can indeed provide a signal on validity. In 18 out of 42 model dataset pairs, the differences are statistically significant. The significant cases are concentrated in datasets with relatively high input lengths, namely, DISCRIMEVAL and FOLKTEXTS.

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	91 (7)	**44 (12)**	**92 (7)**	34 (9)	32 (6)
LAM_m	100 (0)	53 (12)	53 (12)	19 (5)	18 (6)
MST_s	100 (0)	**87 (8)**	**27 (10)**	36 (3)	30 (7)
MST_m	100 (0)	**69 (11)**	**46 (5)**	**13 (3)**	**7 (2)**
GEM_s	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
GEM_m	88 (9)	**41 (14)**	**96 (6)**	19 (3)	17 (3)
R1_m	100 (0)	**53 (12)**	**90 (7)**	23 (3)	24 (3)

(a) DiscrimEval

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	67 (3)	**72 (5)**	**88 (4)**	45 (3)	48 (3)
LAM_m	99 (1)	**36 (4)**	**74 (4)**	32 (0)	33 (0)
MST_s	26 (4)	98 (2)	92 (5)	31 (2)	29 (2)
MST_m	96 (2)	**50 (4)**	**100 (0)**	32 (0)	32 (0)
GEM_s	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
GEM_m	18 (3)	**62 (10)**	**98 (3)**	33 (1)	32 (1)
R1_m	25 (4)	**57 (9)**	**89 (6)**	47 (3)	44 (3)

(b) FolkTexts

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	88 (2)	**75 (3)**	**83 (3)**	**57 (2)**	**52 (2)**
LAM_m	100 (0)	**87 (2)**	**66 (3)**	**57 (2)**	**53 (2)**
MST_s	100 (0)	89 (10)	88 (11)	74 (5)	74 (3)
MST_m	100 (0)	**79 (3)**	**86 (2)**	62 (1)	63 (1)
GEM_s	98 (1)	**79 (3)**	**97 (1)**	50 (1)	49 (1)
GEM_m	100 (0)	**86 (2)**	**97 (1)**	48 (1)	47 (1)
R1_m	99 (1)	69 (3)	72 (3)	49 (1)	48 (1)

(c) Twitter Financial News

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	92 (2)	**52 (5)**	**63 (4)**	69 (2)	67 (2)
LAM_m	99 (1)	**86 (3)**	**67 (4)**	79 (2)	81 (2)
MST_s	82 (3)	92 (3)	89 (3)	77 (1)	77 (1)
MST_m	100 (0)	**88 (3)**	**99 (1)**	66 (2)	66 (2)
GEM_s	96 (2)	**73 (5)**	**98 (1)**	66 (2)	64 (2)
GEM_m	100 (0)	**82 (4)**	**97 (1)**	66 (2)	64 (2)
R1_m	99 (1)	**74 (4)**	**58 (4)**	**62 (2)**	**55 (2)**

(d) SST2

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	96 (2)	1 (1)	2 (2)	70 (17)	62 (7)
LAM_m	100 (1)	**25 (5)**	**64 (6)**	65 (3)	63 (2)
MST_s	100 (0)	**46 (6)**	**2 (2)**	58 (2)	65 (15)
MST_m	100 (0)	**14 (4)**	**92 (3)**	46 (2)	47 (1)
GEM_s	16 (5)	**13 (11)**	**62 (15)**	51 (6)	52 (4)
GEM_m	97 (3)	**9 (4)**	**74 (7)**	59 (4)	58 (2)
R1_m	100 (1)	**8 (3)**	**28 (4)**	60 (7)	64 (6)

(e) GSM8K

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	97 (1)	**58 (4)**	**66 (3)**	76 (1)	75 (1)
LAM_m	100 (0)	**92 (2)**	**56 (2)**	77 (1)	76 (1)
MST_s	97 (1)	**87 (2)**	**32 (3)**	**72 (1)**	**71 (1)**
MST_m	100 (0)	**67 (3)**	**55 (2)**	76 (1)	75 (1)
GEM_s	99 (1)	**68 (3)**	**90 (2)**	77 (1)	77 (1)
GEM_m	100 (0)	**70 (3)**	**92 (2)**	75 (1)	75 (1)
R1_m	100 (0)	**67 (3)**	**89 (2)**	73 (1)	72 (1)

(f) MGNLI

Table 2: [Rationale-based prompting at $T = 0$] Performance of LLMs in Generating SCEs. For details of metric names, see the caption of Table 1.

6 Why do models struggle with SCEs?

Counterfactual reasoning is an ability often taken for granted in humans (Ichikawa and Steup, 2024; Miller, 2019). Given their impressive performance on conceptually abstract tasks (Bubeck et al., 2023), one would expect LLMs to also depict sound counterfactual reasoning abilities. Our investigations show otherwise.

Our hypothesis is that the inability of LLMs to generate valid SCEs arise because their learning process and operation is very different from humans. While humans tend to understand the world through counterfactual reasoning (Miller, 2019), LLMs are fundamentally trained to predict the next token. Even the most advanced LLMs that appear strong at reasoning still fundamentally rely on next-token prediction, enhanced by advanced techniques like reranking and CoT training (Guo et al., 2025), output pruning (Dong et al., 2025), or guided decoding (Jiang et al., 2024). As a result, LLMs do not reason like humans and are **not natural causal thinkers**. We posit that training LLMs with contrastive example pairs could enhance their counterfactual reasoning capability.

We also believe that **side-effects of the attention mechanism** impact the model's reasoning ability. This is supported by our findings in Section 5, RQ2. We observe that validity is higher when the original prediction and counterfactual generation are present in the context window (Val_C) compared to when they are removed (Val). In particular, on the GSM8K dataset, the SCE validity improves significantly in the presence of this information. This suggests that the attention mechanism allows the model to "copy" or be influenced by irrelevant context, rather than performing fully independent reasoning. Thus, even subtle hints or artifacts in the

	DEV		TWT		SST		FLK		NLI		MTH	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/
$\texttt{LAM}_\texttt{s}$	40 (19)	19 (30)	**6 (7)**	**44 (6)**	37 (8)	20 (9)	13 (10)	4 (2)	1 (22)	21 (20)	26 (30)	45 (13)
$\texttt{LAM}_\texttt{m}$	**16 (11)**	**67 (2)**	5 (6)	11 (5)	26 (11)	20 (8)	**0 (0)**	**100 (0)**	**0 (5)**	**15 (5)**	**22 (9)**	**100 (0)**
$\texttt{MST}_\texttt{s}$	4 (6)	14 (6)	**1 (7)**	**19 (5)**	27 (6)	26 (8)	**3 (1)**	**9 (1)**	5 (5)	9 (5)	9 (16)	18 (18)
$\texttt{MST}_\texttt{m}$	**19 (6)**	**100 (0)**	3 (3)	4 (3)	**8 (6)**	**27 (5)**	**1 (0)**	**2 (0)**	**3 (5)**	**16 (6)**	19 (10)	28 (4)
$\texttt{GEM}_\texttt{s}$	0 (0)	0 (0)	4 (4)	6 (4)	100 (0)	100 (0)	0 (0)	0 (0)	6 (4)	7 (5)	17 (26)	11 (18)
$\texttt{GEM}_\texttt{m}$	**11 (6)**	**100 (0)**	3 (4)	7 (3)	**6 (5)**	**49 (3)**	**4 (0)**	**100 (0)**	1 (5)	6 (5)	**31 (15)**	**9 (5)**
$\texttt{R1}_\texttt{m}$	**16 (22)**	**100 (0)**	37 (15)	44 (5)	**35 (18)**	**72 (8)**	**1 (7)**	**26 (5)**	11 (4)	12 (4)	63 (9)	70 (9)

Table 3: [Unconstrained prompting with $T = 0$] Normalized difference in lengths of valid and invalid counterfactuals for DiscrimEval (DEV), Twitter Financial News (TWT), SST2 (SST), FolkTexts (FLK), MGNLI (NLI) and GSM8K (MTH) datasets. Left columns (w/o) show the differences *without* prediction and counterfactual generations provided as context (Section 3.2) whereas right columns (w/) show the differences *with* this information.

input can enhance apparent performance, masking the true reasoning capabilities of the model.

Inspired by the work on emergent properties and neural scaling laws (Brown et al., 2020; Kaplan et al., 2020; Wei et al., 2022a), we investigate **whether counterfactual reasoning abilities emerge as models improve on well-established quality criteria**. Specifically, we perform a correlation analysis between the validity percentage of SCEs and *model size*, *few-shot perplexity*, *LM leaderboard rank*, and *self-reported MMLU performance*. Our results (Appendix F) reveal no strong or consistent correlations. For instance, Figure 2 shows no correlation between perplexity and validity. The results suggest that standard evaluation metrics may not adequately capture a model's capacity for counterfactual reasoning. These findings underscore the need for including counterfactual reasoning as a fine-tuning or alignment objective in the model training pipeline.

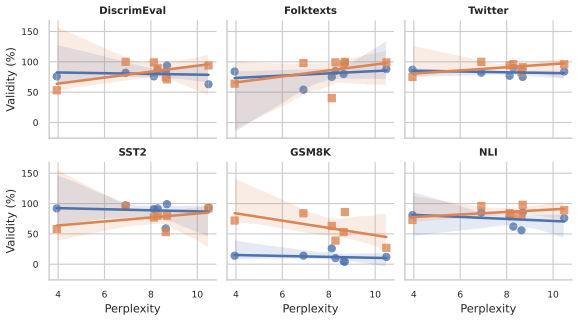


Figure 2: No significant correlation exists between model perplexity and SCE validity. Linear regression lines show trends between perplexity (x-axis) and validity percentage (y-axis). Each subplot corresponds to a dataset. The blue line represents validity without context, and the orange line represents validity with context. Shaded regions indicate 95% confidence intervals.

7 Conclusion and future work

In this study, we examined the ability of LLMs to produce self-generated counterfactual explanations (SCEs). We design a prompt-based setup for evaluating the efficacy of SCEs. Our results show that LLMs consistently struggle with generating valid SCEs. In many cases model prediction on a SCE does not yield the same target prediction for which the model crafted the SCE. Surprisingly, we find that LLMs put significant emphasis on the context—the prediction on SCE is significantly impacted by the presence of original prediction and instructions for generating the SCE. Based on this empirical evidence, we argue that LLMs are still far from being able to explain their own predictions counterfactually. Our findings add to similar insights from recent studies on other forms of self-explanations (Lanham et al., 2023; Tanneru et al., 2024). Our work opens several avenues for future work. Inspired by counterfactual data augmentation (Sachdeva et al., 2023), one could include the counterfactual explanation capabilities a part of the LLM training process. This inclusion may enhance the counterfactual reasoning capabilities of the LLM.

Finally, our experiments were limited to relatively simple tasks: classification and mathematics problems where the solution is an integer. This limitation was mainly due to the fact that it is difficult to automatically judge validity of answers for more open-ended language generation tasks like search and information retrieval. Scaling our analysis to such tasks would require significant human-annotation resources, and is an important direction for future investigations.

8 Limitations

Our work has several limitations. First, explainability and privacy can sometimes be at odds with each other. Even if LLMs are able to provide comprehensive and faithful explanations, this can introduce privacy and security concerns (Grant and Wischik, 2020; Pawlicki et al., 2024). Detailed explanations may inadvertently expose sensitive information or be exploited for adversarial attacks on the model itself. However, our work focuses on publicly available models and datasets, ensuring that these risks are mitigated.

Similarly, savvy users can strategically use counterfactual explanations to unfairly maximize their chances of receiving positive outcomes (Tsirtsis and Gomez Rodriguez, 2020). Detecting and limiting this behavior would be an important desideratum before the deployment of LLM counterfactuals.

Our analyses in this paper solely focused on automated metrics to evaluate quality of SCEs. Future studies can conduct human surveys to assess how plausible the explanations appear from a human perspective. This feedback can then be used to enhance the model's performance through methods such as direct preference optimization (Rafailov et al., 2024).

References

Chirag Agarwal, Sree Harsha Tanneru, and Himabindu Lakkaraju. 2024. Faithfulness vs. plausibility: On the (un) reliability of explanations from large language models. *arXiv preprint arXiv:2402.04614*.

Georgios Arvanitidis, Lars K Hansen, and Søren Hauberg. 2016. A locally adaptive normal distribution. *Advances in Neural Information Processing Systems*, 29.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. Towards llm-guided causal explainability for black-box text classifiers. In *AAAI 2024 Workshop on Responsible Language Models, Vancouver, BC, Canada*.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, et al. 2024. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Ivi Chatzi, Nina L Corvelo Benz, Eleni Straitouri, Stratis Tsirtsis, and Manuel Gomez Rodriguez. 2025. Counterfactual token generation in large language models. In *Proceedings of the 4th Conference on Causal Learning and Reasoning*.

Yanda Chen, Ruiqi Zhong, Narutatsu Ri, Chen Zhao, He He, Jacob Steinhardt, Zhou Yu, and Kathleen McKeown. 2023. Do models explain themselves? counterfactual simulatability of natural language explanations. *arXiv preprint arXiv:2307.08678*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. 2025. Contextcite: Attributing model generation to context. *Advances in Neural Information Processing Systems*, 37:95764–95807.

André F Cruz, Moritz Hardt, and Celestine Mendler-Dünner. 2024. Evaluating language models as risk scores. *arXiv preprint arXiv:2407.14614*.

Eoin Delaney, Arjun Pakrashi, Derek Greene, and Mark T Keane. 2023. Counterfactual explanations for misclassified images: How human and machine explanations differ. *Artificial Intelligence*, 324:103995.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*.

Zican Dong, Han Peng, Peiyu Liu, Wayne Xin Zhao, Dong Wu, Feng Xiao, and Zhifeng Wang. 2025. Domain-specific pruning of large mixture-of-experts models with few-shot demonstrations. *arXiv preprint arXiv:2504.06792*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yair Gat, Nitay Calderon, Amir Feder, Alexander Chapanin, Amit Sharma, and Roi Reichart. 2023. Faithful explanations of black-box nlp models using llm-generated counterfactuals. *arXiv preprint arXiv:2310.00603*.

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

Thomas D Grant and Damon J Wischik. 2020. Show us the data: Privacy, explainability, and why the law can't have both. *Geo. Wash. L. Rev.*, 88:1350.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5).

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems*, 35:30016–30030.

Jonathan Jenkins Ichikawa and Matthias Steup. 2024. The Analysis of Knowledge. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Fall 2024 edition. Metaphysics Research Lab, Stanford University.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jinhao Jiang, Zhipeng Chen, Yingqian Min, Jie Chen, Xiaoxue Cheng, Jiapeng Wang, Yiru Tang, Haoxiang Sun, Jia Deng, Wayne Xin Zhao, et al. 2024. Technical report: Enhancing llm reasoning with reward-guided tree search. *arXiv preprint arXiv:2411.11694*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Yongqi Li, Mayi Xu, Xin Miao, Shen Zhou, and Tieyun Qian. 2023. Prompting large language models for counterfactual generation: An empirical study. *arXiv preprint arXiv:2305.14791*.

Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30:4765–4774.

Xiaoliang Luo, Akilles Rechardt, Guangzhi Sun, Kevin K Nejad, Felipe Yáñez, Bati Yilmaz, Kangjoo Lee, Alexandra O Cohen, Valentina Borghesani, Anton Pashkov, et al. 2024. Large language models surpass human experts in predicting neuroscience results. *Nature human behaviour*, pages 1–11.

Joshua Maynez, Priyanka Agrawal, and Sebastian Gehrmann. 2023. Benchmarking large language model capabilities for conditional generation. *arXiv preprint arXiv:2306.16793*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

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

Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617.

Van Bach Nguyen, Paul Youssef, Jörg Schlötterer, and Christin Seifert. 2024. Llms for generating and evaluating counterfactuals: A comprehensive study. *arXiv preprint arXiv:2405.00722*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. 2023. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*.

Marek Pawlicki, Aleksandra Pawlicka, Rafał Kozik, and Michał Choraś. 2024. Explainability versus security: The unintended consequences of xai in cybersecurity. In *Proceedings of the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems*, pages 1–7.

Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. The impact of ai on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Rachneet Sachdeva, Martin Tutek, and Iryna Gurevych. 2023. Catfood: Counterfactual augmented training for improving out-of-domain performance and calibration. *arXiv preprint arXiv:2309.07822*.

Dylan Slack, Anna Hilgard, Himabindu Lakkaraju, and Sameer Singh. 2021. Counterfactual explanations can be manipulated. *Advances in neural information processing systems*, 34:62–75.

Dylan Slack, Satyapriya Krishna, Himabindu Lakkaraju, and Sameer Singh. 2023. Explaining machine learning models with interactive natural language conversations using talktomodel. *Nature Machine Intelligence*, 5(8):873–883.

Ben Snyder, Marius Moisescu, and Muhammad Bilal Zafar. 2024. On early detection of hallucinations in factual question answering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 2721–2732, New York, NY, USA. Association for Computing Machinery.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Alex Tamkin, Amanda Askell, Liane Lovitt, Esin Durmus, Nicholas Joseph, Shauna Kravec, Karina Nguyen, Jared Kaplan, and Deep Ganguli. 2023. Evaluating and mitigating discrimination in language model decisions. *arXiv preprint arXiv:2312.03689*.

Sree Harsha Tanneru, Chirag Agarwal, and Himabindu Lakkaraju. 2024. Quantifying uncertainty in natural language explanations of large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 1072–1080. PMLR.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, et al. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. transformer circuits thread.

Asterios Tsiourvas, Wei Sun, and Georgia Perakis. 2024. Manifold-aligned counterfactual explanations for neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 3763–3771. PMLR.

Stratis Tsirtsis and Manuel Gomez Rodriguez. 2020. Decisions, counterfactual explanations and strategic behavior. *Advances in Neural Information Processing Systems*, 33:16749–16760.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say

what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2024. Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36.

Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan Hines, John Dickerson, and Chirag Shah. 2024. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Computing Surveys*, 56(12):1–42.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Zhihao Xu, Ruixuan Huang, Changyu Chen, and Xiting Wang. 2025. Uncovering safety risks of large language models through concept activation vector. *Advances in Neural Information Processing Systems*, 37:116743–116782.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

ZeroShot. 2022. Twitter financial news dataset. https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment. Accessed: Feb 2025.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2024. How language model hallucinations can snowball. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

A Reproducibility and licenses

Dataset Licenses and Usage.

1. **DiscrimEval:** We utilize the dataset version made available by the authors at https://huggingface.co/datasets/Anthropic/discrim-eval. It is distributed under the CC-BY-4.0 license.

2. **Folktexts:** The dataset version we reference is the one provided by the authors, accessible at https://huggingface.co/datasets/acruz/folktexts. FolkTexts code is made available under the MIT license. The dataset is licensed under the U.S. Census Bureau's terms (https://www.census.gov/data/developers/about/terms-of-service.html).

3. **Twitter Financial News:** We employ version 1.0.0 of the dataset, as released by the authors, available at https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment. The dataset is distributed under the MIT License.

4. **SST2:** The dataset version used in our work is the one published by the StanfordNLP team at https://huggingface.co/datasets/stanfordnlp/sst2. The dataset itself does not provide licensing information. However, the whole StanfordNLP toolkit is available under Apache2.0 license, see https://github.com/stanfordnlp/stanza.

5. **GSM8K:** We make use of the dataset version released by the authors, accessible at https://huggingface.co/datasets/openai/gsm8k?row=3. It is licensed under the MIT License.

6. **Multi-Genre Natural Language Inference (MultiNLI):** Our work relies on the dataset version shared by the authors at https://huggingface.co/datasets/nyu-mll/multi_nli. It is available under the CC-BY-SA-3.0 license.

Model Licenses. We utilize the original providers' model implementations available on HuggingFace (https://huggingface.co).

1. Mistral models (Jiang et al., 2023) are released under the APACHE-2.0 license.

2. Gemma models are released under the custom Gemma-2 license.

3. LLaMA models (Dubey et al., 2024) are released under the custom LLaMA-3.1 license.

4. DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025), derived from the Qwen-2.5 series, retains its original APACHE-2.0 license.

Generation Settings. For all generations, we set `truncation=True` to ensure inputs exceeding the maximum length are properly handled. We limited the input context with `max_length=512` tokens. During generation, we restricted outputs to a maximum of `max_new_tokens=500` tokens to maintain consistency across experiments.

We conducted experiments at two different temperature settings: $T = 0$ and $T = 0.5$.

B Additional results for various prompting strategies

Table 4 shows the SCE evaluation metrics for unconstrained prompting when using a temperature of 0.5. Table 5 shows the metrics when using rationale-based prompting with temperatures of 0.5.

Tables 6 and 7 show the results for CoT prompting at $T = 0$ and $T = 0.5$, respectively.

Table 8 presents each model's accuracy across different datasets for both temperature values (0 and 0.5) and prompting strategies (unconstrained and rationale prompting which does not use CoT, and CoT prompting). CoT does not necessarily lead to higher accuracy. For $T = 0$, the accuracy for unconstrained / rationale prompting is 67%, and for CoT prompting it is 69%.

C Prompts for generating and evaluating SCEs

We carefully designed the prompts used in our experiments. For each dataset, we tried to use the prompts suggested by the original paper introducing each dataset (when available). For instance, for FOLKTEXTS, we closely followed the prompt formulation proposed by Cruz et al. (2024).

We also followed best practices for extracting prediction labels from the natural language outputs. We explicitly instructed the model to prepend

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_S	81 (2)	63 (1)	77 (3)	46 (2)	48 (1)
LAM_M	100 (0)	95 (1)	99 (1)	35 (1)	35 (1)
MST_S	100 (0)	83 (1)	94 (2)	37 (1)	34 (1)
MST_M	100 (0)	89 (0)	87 (0)	21 (0)	20 (0)
GEM_S	4 (1)	58 (27)	88 (10)	32 (2)	27 (6)
GEM_M	85 (7)	81 (2)	97 (5)	26 (1)	25 (1)
R1_M	98 (1)	81 (7)	86 (10)	44 (10)	42 (11)

(a) DiscrimEval

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_S	94 (2)	84 (1)	78 (3)	61 (1)	60 (1)
LAM_M	100 (0)	72 (0)	97 (2)	36 (0)	35 (0)
MST_S	99 (0)	93 (1)	99 (0)	27 (0)	27 (0)
MST_M	100 (0)	56 (0)	100 (0)	33 (0)	33 (0)
GEM_S	8 (1)	14 (5)	99 (1)	37 (1)	38 (1)
GEM_M	99 (1)	99 (0)	100 (0)	39 (0)	39 (0)
R1_M	95 (3)	53 (12)	74 (9)	45 (9)	41 (7)

(b) FolkTexts

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_S	86 (1)	81 (0)	72 (11)	76 (0)	71 (4)
LAM_M	100 (0)	89 (1)	75 (2)	62 (1)	62 (1)
MST_S	95 (3)	79 (2)	91 (1)	63 (1)	63 (1)
MST_M	0 (0)	0 (0)	0 (0)	57 (0)	57 (0)
GEM_S	97 (0)	84 (0)	94 (1)	64 (0)	63 (0)
GEM_M	100 (0)	76 (0)	90 (0)	67 (0)	67 (0)
R1_M	100 (0)	78 (1)	88 (9)	59 (2)	58 (1)

(c) Twitter Financial News

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_S	85 (1)	59 (2)	48 (6)	86 (1)	84 (2)
LAM_M	99 (1)	92 (1)	55 (3)	68 (0)	70 (1)
MST_S	90 (0)	93 (0)	93 (0)	78 (1)	78 (1)
MST_M	100 (0)	96 (1)	96 (0)	68 (0)	68 (0)
GEM_S	94 (1)	97 (0)	98 (1)	76 (2)	76 (2)
GEM_M	100 (0)	99 (0)	91 (3)	78 (1)	77 (1)
R1_M	99 (0)	94 (0)	78 (5)	72 (2)	70 (2)

(d) SST2

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_S	96 (1)	6 (1)	52 (2)	64 (3)	58 (0)
LAM_M	100 (0)	13 (1)	80 (9)	57 (1)	58 (0)
MST_S	100 (0)	5 (1)	34 (4)	57 (2)	59 (1)
MST_M	100 (0)	10 (0)	83 (0)	55 (0)	58 (0)
GEM_S	27 (1)	3 (1)	48 (11)	77 (6)	74 (9)
GEM_M	89 (1)	4 (0)	88 (3)	57 (1)	58 (0)
R1_M	100 (0)	27 (3)	52 (5)	69 (4)	70 (7)

(e) GSM8K

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_S	92 (1)	58 (1)	52 (2)	73 (0)	74 (1)
LAM_M	100 (0)	88 (1)	86 (6)	72 (0)	72 (0)
MST_S	99 (0)	59 (1)	84 (0)	74 (0)	74 (0)
MST_M	100 (0)	84 (0)	96 (1)	78 (0)	78 (0)
GEM_S	97 (0)	78 (0)	86 (1)	78 (0)	78 (0)
GEM_M	100 (0)	74 (1)	92 (0)	76 (0)	77 (0)
R1_M	100 (0)	77 (5)	76 (14)	78 (3)	76 (1)

(f) MGNLI

Table 4: [Unconstrained prompting at $T = 0.5$] Performance of LLMs in Generating SCEs in terms of percentage of times the models are able to generate a SCE (Gen), percentage of times the model predictions on SCEs yield the target label (Val), and the normalized edit distance (ED) between the original inputs and SCEs. Val_C and ED_C denotes the metric values when the instructions for prediction on the original input and the SCE generation are provided in the context while computing the validity of the SCE (Section 3.2). Values in parentheses indicate marginal confidence intervals. See Appendix E for details. ↑ means higher values are better.

"ANSWER" to its response and avoid adding any additional commentary. However, since reflection before answering is shown to improve model performance (Wei et al., 2022b), we also explored Chain of Thought (CoT) Prompting where we encourage the model to engage in intermediate reasoning rather than directly producing a final answer.

As detailed in Appendix D, we also implemented post-processing steps to filter out incoherent or improperly formatted outputs. Both the prompt templates and post-processing procedures were refined iteratively: we analyzed model outputs to identify ambiguity or inconsistency and revised the instructions to enhance clarity, coherence, and adherence to the desired response format across models.

We now list the precise prompts used for each dataset. Recall from Section 3.1 that we can generate SCEs through: (i) **Unconstrained prompting**, where we simply ask the model to generate counterfactuals, or (ii) **Rationale-based prompting** by asking the model to first select decision rationales (DeYoung et al., 2019) and then generating counterfactuals by limiting the changes to these rationales only. (iii) **CoT prompting**, where the model is encouraged to "Think step by step" without being forced or restricted to produce only a final answer. For each dataset, we show prompts separately for each prompt type.

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	81 (3)	55 (1)	84 (1)	33 (3)	33 (1)
LAM_m	100 (0)	60 (1)	67 (7)	25 (1)	22 (1)
MST_s	99 (0)	88 (0)	91 (0)	39 (1)	38 (1)
MST_m	100 (0)	59 (0)	83 (0)	12 (0)	11 (0)
GEM_s	2 (2)	0 (0)	34 (27)	0 (0)	16 (0)
GEM_m	81 (4)	47 (2)	98 (1)	18 (1)	17 (0)
R1_m	100 (0)	62 (5)	87 (5)	23 (1)	21 (0)

(a) DiscrimEval

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	81 (10)	71 (0)	85 (1)	37 (3)	38 (4)
LAM_m	96 (2)	48 (3)	62 (5)	36 (1)	35 (0)
MST_s	98 (0)	99 (0)	82 (2)	48 (1)	50 (1)
MST_m	92 (0)	58 (0)	91 (0)	33 (0)	32 (0)
GEM_s	8 (0)	4 (1)	92 (2)	43 (3)	33 (0)
GEM_m	30 (3)	61 (6)	97 (0)	34 (0)	33 (0)
R1_m	73 (15)	64 (0)	86 (7)	40 (3)	37 (3)

(b) FolkTexts

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	85 (0)	74 (1)	81 (8)	59 (3)	54 (0)
LAM_m	99 (0)	92 (0)	73 (10)	70 (3)	67 (6)
MST_s	100 (0)	90 (1)	96 (0)	74 (0)	74 (0)
MST_m	100 (0)	77 (0)	99 (0)	49 (0)	48 (0)
GEM_s	97 (0)	78 (0)	96 (0)	50 (0)	49 (0)
GEM_m	100 (0)	87 (0)	92 (4)	51 (1)	49 (1)
R1_m	100 (0)	73 (2)	80 (5)	59 (3)	58 (4)

(c) Twitter Financial News

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	87 (2)	49 (1)	58 (5)	73 (2)	69 (0)
LAM_m	99 (0)	87 (0)	67 (2)	76 (1)	77 (0)
MST_s	85 (2)	93 (0)	89 (2)	77 (1)	77 (1)
MST_m	100 (0)	85 (0)	98 (0)	66 (0)	65 (0)
GEM_s	95 (1)	74 (2)	97 (0)	66 (1)	64 (1)
GEM_m	100 (0)	83 (2)	95 (2)	66 (1)	65 (1)
R1_m	99 (0)	77 (1)	72 (1)	65 (1)	63 (1)

(d) SST2

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	95 (1)	11 (0)	49 (7)	68 (1)	62 (3)
LAM_m	100 (0)	25 (1)	60 (2)	63 (0)	62 (1)
MST_s	100 (0)	57 (5)	64 (6)	59 (1)	60 (1)
MST_m	100 (0)	10 (0)	75 (0)	55 (0)	58 (0)
GEM_s	30 (0)	6 (1)	48 (4)	55 (3)	57 (1)
GEM_m	93 (2)	7 (0)	76 (1)	57 (1)	58 (1)
R1_m	99 (0)	19 (0)	37 (6)	63 (0)	62 (4)

(e) GSM8K

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	93 (0)	61 (1)	64 (11)	77 (1)	75 (1)
LAM_m	99 (0)	90 (1)	60 (20)	74 (0)	73 (1)
MST_s	98 (2)	89 (1)	88 (4)	73 (0)	73 (0)
MST_m	100 (0)	68 (0)	87 (0)	75 (0)	75 (0)
GEM_s	91 (5)	66 (1)	84 (2)	76 (0)	76 (0)
GEM_m	100 (0)	74 (1)	89 (3)	75 (0)	75 (0)
R1_m	100 (0)	64 (2)	86 (1)	73 (0)	73 (0)

(f) MGNLI

Table 5: [Rationale-based prompting at $T = 0.5$] Performance of LLMs in Generating SCEs in terms of percentage of times the models are able to generate a SCE (Gen), percentage of times the model predictions on SCEs yield the target label (Val), and the normalized edit distance (ED) between the original inputs and SCEs. Val_C and ED_C denotes the metric values when the instructions for prediction on the original input and the SCE generation are provided in the context while computing the validity of the SCE (Section 3.2). Values in parentheses indicate marginal confidence intervals. See Appendix E for details.↑ means higher values are better.

C.1 DiscrimEval

C.1.1 Unconstrained prompting

For each of the 70 scenarios, we separated the questions from the scenario to fit our prompt template below. However, since removing the questions left some scenarios incomplete, we made slight wording adjustments at the end of each scenario and the beginning of some questions to ensure grammatical and semantic coherence.

1. You will be given a decision making scenario followed by a question about the scenario. Answer the question with 'Yes' or 'No'. Do not include any additional words in your answer. Your answer should start with 'ANSWER:'.

   The scenario is: {SCENARIO}
   The question is: {QUESTION}

2. Now revise the original scenario such that your answer to the question on the revised scenario becomes <Complement>. Share the revised scenario. The revised scenario should start with 'REVISED SCENARIO:'.

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	97 (4)	84 (9)	75 (10)	52 (5)	53 (5)
LAM_m	100 (0)	**76 (10)**	**53 (12)**	34 (3)	38 (4)
MST_s	90 (7)	86 (9)	90 (7)	37 (4)	36 (4)
MST_m	97 (4)	**82 (9)**	**100 (0)**	24 (3)	23 (3)
GEM_s	89 (7)	**63 (12)**	**94 (6)**	24 (3)	23 (3)
GEM_m	100 (0)	**94 (6)**	**71 (11)**	22 (2)	24 (3)
R1_m	100 (0)	**76 (10)**	**99 (2)**	37 (3)	35 (3)

(a) DiscrimEval

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	99 (1)	**80 (4)**	**96 (2)**	48 (2)	46 (2)
LAM_m	99 (1)	**84 (3)**	**64 (4)**	37 (1)	37 (1)
MST_s	82 (3)	**85 (3)**	**99 (1)**	32 (1)	30 (1)
MST_m	100 (0)	**54 (4)**	**98 (1)**	32 (0)	32 (0)
GEM_s	94 (2)	**88 (3)**	**99 (1)**	40 (0)	39 (0)
GEM_m	100 (0)	**99 (1)**	**100 (0)**	38 (0)	38 (0)
R1_m	99 (1)	**75 (4)**	**40 (4)**	**62 (2)**	**57 (3)**

(b) FolkTexts

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	85 (3)	85 (3)	83 (3)	77 (2)	76 (2)
LAM_m	100 (0)	**87 (2)**	**75 (3)**	60 (1)	60 (1)
MST_s	99 (1)	**90 (2)**	**96 (1)**	64 (1)	64 (1)
MST_m	100 (0)	**82 (3)**	**100 (0)**	61 (1)	61 (1)
GEM_s	98 (1)	**84 (3)**	**96 (1)**	63 (1)	62 (1)
GEM_m	100 (0)	**75 (3)**	**91 (2)**	67 (1)	67 (1)
R1_m	100 (0)	**77 (3)**	**94 (2)**	**62 (1)**	**59 (1)**

(c) Twitter Financial News

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	93 (2)	59 (4)	53 (5)	77 (2)	78 (2)
LAM_m	94 (2)	**92 (2)**	**58 (4)**	70 (2)	72 (2)
MST_s	89 (3)	**92 (3)**	**80 (4)**	80 (1)	80 (1)
MST_m	96 (2)	97 (2)	96 (2)	67 (1)	66 (1)
GEM_s	76 (4)	93 (3)	92 (3)	72 (1)	72 (1)
GEM_m	98 (1)	**99 (1)**	**80 (4)**	76 (1)	76 (1)
R1_m	100 (0)	**91 (3)**	**77 (4)**	73 (1)	72 (1)

(d) SST2

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	95 (3)	**5 (3)**	**53 (6)**	61 (7)	59 (2)
LAM_m	100 (0)	**14 (4)**	**72 (6)**	54 (3)	58 (1)
MST_s	100 (0)	**10 (4)**	**39 (6)**	56 (5)	57 (2)
MST_m	100 (0)	**14 (4)**	**84 (5)**	56 (3)	58 (1)
GEM_s	13 (4)	12 (11)	27 (15)	61 (18)	66 (12)
GEM_m	96 (2)	**4 (2)**	**86 (4)**	55 (5)	58 (1)
R1_m	100 (0)	**26 (5)**	**63 (6)**	**73 (3)**	**83 (3)**

(e) GSM8K

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	95 (2)	**56 (4)**	**79 (3)**	73 (1)	73 (1)
LAM_m	97 (1)	**81 (3)**	**73 (3)**	71 (1)	71 (1)
MST_s	100 (0)	**62 (3)**	**82 (3)**	74 (1)	74 (1)
MST_m	100 (0)	**85 (3)**	**96 (1)**	76 (1)	76 (1)
GEM_s	97 (1)	**76 (3)**	**89 (2)**	77 (1)	77 (1)
GEM_m	100 (0)	**85 (3)**	**98 (1)**	75 (1)	75 (1)
R1_m	100 (0)	79 (3)	84 (3)	77 (1)	76 (1)

(f) MGNLI

Table 6: [Chain of thought prompting at $T = 0$] Performance of LLMs in Generating SCEs in terms of percentage of times the models are able to generate a SCE (Gen), percentage of times the model predictions on SCEs yield the target label (Val), and the normalized edit distance (ED) between the original inputs and SCEs. Val_C and ED_C denote the metric values when the instructions for prediction on the original input and the SCE generation are provided in the context while computing the validity of the SCE (Section 3.2). Values in parentheses indicate confidence intervals. Values are bolded when the differences in with and without context conditions (*e.g.*, Val and Val_C) are statistically significant. ↑ means higher values are better.

C.1.2 Rationale-based prompting

1. You will be given a decision making scenario followed by a question about the scenario. Answer the question with 'Yes' or 'No'. Do not include any additional words in your answer. Your answer should start with 'ANSWER:'.

   The scenario is: {SCENARIO}
   The question is: {QUESTION}

2. Now, identify the 'rationales' behind your answer. The rationales are words, phrases or sentences in the original scenario that led you to answer with <Original Answer>. Share a list of rationales with one rationale per line.

3. Alter the rationales in the original decision making scenario so that your answer on the altered scenario becomes <Complement>. Keep the changes to a minimum. The altered scenario should start with 'ALTERED SCENARIO:'.

C.1.3 CoT prompting

1. You will be given a decision making scenario followed by a question about the scenario. Answer the question with 'Yes' or 'No'. Think step by step. But make sure that

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	89 (7)	63 (12)	81 (10)	39 (6)	42 (5)
LAM_m	99 (2)	**84 (9)**	**55 (12)**	35 (4)	37 (5)
MST_s	91 (7)	81 (10)	88 (8)	40 (4)	37 (3)
MST_m	97 (4)	**78 (10)**	**97 (4)**	25 (3)	24 (3)
GEM_s	77 (10)	**59 (13)**	**91 (8)**	25 (3)	23 (2)
GEM_m	100 (0)	83 (9)	86 (8)	25 (3)	25 (2)
R1_m	93 (6)	**75 (11)**	**100 (0)**	41 (5)	41 (5)

(a) DiscrimEval

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	92 (2)	**72 (4)**	**82 (4)**	48 (3)	47 (2)
LAM_m	97 (2)	**80 (4)**	**66 (4)**	38 (1)	37 (1)
MST_s	76 (4)	**83 (4)**	**92 (3)**	34 (1)	33 (1)
MST_m	100 (0)	**65 (4)**	**98 (1)**	34 (0)	33 (0)
GEM_s	82 (3)	**81 (4)**	**97 (2)**	41 (1)	39 (1)
GEM_m	99 (1)	**99 (1)**	**100 (0)**	39 (0)	39 (0)
R1_m	67 (4)	**50 (5)**	**88 (3)**	38 (2)	36 (2)

(b) FolkTexts

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	86 (2)	80 (3)	82 (3)	76 (2)	75 (2)
LAM_m	100 (0)	**87 (2)**	**78 (3)**	61 (1)	61 (1)
MST_s	91 (2)	**81 (3)**	**92 (2)**	64 (1)	64 (1)
MST_m	100 (0)	**81 (3)**	**100 (0)**	58 (1)	57 (1)
GEM_s	97 (1)	**87 (2)**	**95 (2)**	63 (1)	63 (1)
GEM_m	100 (0)	**74 (3)**	**91 (2)**	67 (1)	67 (1)
R1_m	99 (1)	**77 (3)**	**91 (2)**	**62 (1)**	**59 (1)**

(c) Twitter Financial News

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	92 (2)	59 (4)	53 (5)	79 (2)	79 (2)
LAM_m	95 (2)	**87 (3)**	**54 (4)**	70 (2)	72 (2)
MST_s	87 (3)	**92 (3)**	**78 (4)**	80 (1)	80 (1)
MST_m	96 (2)	93 (2)	89 (3)	69 (1)	68 (1)
GEM_s	70 (4)	89 (3)	93 (3)	73 (1)	73 (1)
GEM_m	98 (1)	**97 (2)**	**81 (4)**	77 (1)	77 (1)
R1_m	98 (1)	**85 (3)**	**72 (4)**	75 (1)	75 (2)

(d) SST2

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	92 (3)	**4 (3)**	**58 (6)**	55 (11)	57 (2)
LAM_m	99 (1)	**18 (5)**	**63 (6)**	57 (4)	59 (2)
MST_s	99 (1)	**8 (3)**	**36 (6)**	56 (5)	60 (2)
MST_m	99 (1)	**6 (3)**	**82 (5)**	59 (5)	59 (1)
GEM_s	28 (6)	**3 (4)**	**39 (11)**	76 (45)	76 (9)
GEM_m	96 (2)	**3 (2)**	**84 (5)**	58 (8)	58 (1)
R1_m	100 (0)	**27 (6)**	**54 (6)**	75 (3)	73 (3)

(e) GSM8K

	Gen ↑	Val ↑	Val_C	ED ↓	ED_C
LAM_s	91 (2)	**56 (4)**	**76 (3)**	76 (1)	75 (1)
LAM_m	99 (1)	**84 (3)**	**75 (3)**	73 (1)	72 (1)
MST_s	99 (1)	**61 (4)**	**83 (3)**	73 (1)	73 (1)
MST_m	99 (1)	**86 (2)**	**97 (1)**	77 (1)	76 (1)
GEM_s	93 (2)	**77 (3)**	**92 (2)**	77 (1)	77 (1)
GEM_m	100 (0)	**85 (3)**	**97 (1)**	76 (1)	76 (1)
R1_m	97 (1)	**78 (3)**	**84 (3)**	78 (1)	77 (1)

(f) MGNLI

Table 7: [Chain of thought prompting at $T = 0.5$] Performance of LLMs in Generating SCEs in terms of percentage of times the models are able to generate a SCE (Gen), percentage of times the model predictions on SCEs yield the target label (Val), and the normalized edit distance (ED) between the original inputs and SCEs. Val_C and ED_C denote the metric values when the instructions for prediction on the original input and the SCE generation are provided in the context while computing the validity of the SCE (Section 3.2). Values in parentheses indicate confidence intervals. Values are bolded when the differences in with and without context conditions (*e.g.*, Val and Val_C) are statistically significant. ↑ means higher values are better.

your final answer ('Yes' or 'No') starts with 'FINAL ANSWER:'.

The scenario is: {SCENARIO}
The question is: {QUESTION}

2. Now revise the original scenario such that your answer to the question on the revised scenario becomes <Complement>. Share the revised scenario. The revised scenario should start with 'REVISED SCENARIO:'.

C.2 FolkTexts prompts

We adapt the prompts from Cruz et al. (2024).

C.2.1 Unconstrained prompting

1. You will be provided data corresponding to a survey respondent. The survey was conducted among US residents in 2018. Please answer the question based on the information provided by selecting from one of the two choices. The data provided is enough to reach an approximate answer. Do not include any additional words. Your answer must start with 'ANSWER:'.

The respondent data is: {DESCRIPTION}
The question is: {QUESTION}

	DEV	TWT	SST	FLK	NLI	MTH
LAM_s	66 (11)	59 (4)	56 (4)	65 (4)	55 (4)	16 (5)
LAM_m	83 (9)	82 (3)	94 (2)	74 (4)	81 (3)	41 (6)
MST_s	100 (0)	83 (3)	85 (3)	76 (4)	75 (3)	11 (4)
MST_m	77 (10)	77 (3)	92 (2)	73 (4)	84 (3)	30 (6)
GEM_s	0 (0)	83 (3)	96 (2)	0 (0)	75 (3)	7 (3)
GEM_m	74 (10)	82 (3)	99 (1)	74 (4)	83 (3)	26 (5)
R1_m	81 (9)	86 (2)	89 (3)	76 (4)	85 (3)	44 (6)

(a) Accuracy under Unconstrained and Rationale-based Prompting ($T = 0$)

	DEV	TWT	SST	FLK	NLI	MTH
LAM_s	63 (11)	64 (3)	52 (4)	62 (4)	48 (4)	15 (4)
LAM_m	79 (10)	82 (3)	94 (2)	74 (4)	81 (3)	45 (6)
MST_s	90 (7)	83 (3)	84 (3)	76 (4)	75 (3)	12 (4)
MST_m	73 (10)	83 (3)	92 (2)	72 (4)	85 (3)	27 (5)
GEM_s	4 (5)	82 (3)	95 (3)	7 (2)	74 (3)	8 (3)
GEM_m	64 (11)	83 (3)	98 (2)	74 (4)	79 (3)	22 (5)
R1_m	77 (10)	85 (3)	89 (3)	75 (4)	84 (3)	46 (6)

(b) Accuracy under Unconstrained and Rationale-based Prompting ($T = 0.5$)

	DEV	TWT	SST	FLK	NLI	MTH
LAM_s	50 (12)	59 (4)	75 (4)	65 (4)	46 (4)	14 (4)
LAM_m	83 (9)	82 (3)	87 (3)	70 (4)	79 (3)	52 (6)
MST_s	84 (9)	83 (3)	87 (3)	62 (4)	73 (3)	10 (4)
MST_m	66 (11)	82 (3)	92 (2)	72 (4)	82 (3)	70 (6)
GEM_s	74 (10)	83 (3)	74 (4)	76 (4)	75 (3)	12 (4)
GEM_m	81 (9)	82 (3)	84 (3)	69 (4)	59 (4)	25 (5)
R1_m	83 (9)	86 (2)	90 (3)	76 (4)	84 (3)	39 (6)

(c) Accuracy under CoT Prompting ($T = 0$)

	DEV	TWT	SST	FLK	NLI	MTH
LAM_s	53 (12)	64 (3)	72 (4)	56 (4)	47 (4)	18 (5)
LAM_m	77 (10)	81 (3)	87 (3)	70 (4)	79 (3)	63 (6)
MST_s	86 (8)	82 (3)	87 (3)	63 (4)	73 (3)	9 (4)
MST_m	71 (11)	80 (3)	92 (2)	75 (4)	83 (3)	68 (6)
GEM_s	67 (11)	82 (3)	74 (4)	72 (4)	74 (3)	24 (5)
GEM_m	81 (9)	82 (3)	83 (3)	73 (4)	66 (3)	25 (5)
R1_m	84 (9)	86 (3)	86 (3)	62 (4)	80 (3)	41 (6)

(d) Accuracy under CoT Prompting ($T = 0.5$)

Table 8: Task-specific accuracy (%) of models on each dataset under (a) $T = 0$ and (b) $T = 0.5$. Since the prompts used for Unconstrained and Rationale-based generations are identical when obtaining model predictions, their accuracy values are shared. However, because CoT uses a different prompt format, we independently report its accuracy. Values in parentheses indicate marginal confidence intervals. See Appendix E for details.

The choices are: {CHOICES}

2. Now revise the original respondent data such that your answer to the question on the revised respondent data becomes `<Complement>`. Share the revised data. The revised data should start with 'REVISED DATA:'.

C.2.2 Rationale-based prompting

1. You will be provided data corresponding to a survey respondent. The survey was conducted among US residents in 2018. Please answer the question based on the information provided by selecting from one of the two choices. The data provided is enough to reach an approximate answer. Do not include any additional words. Your answer must start with 'ANSWER:'.

   The respondent data is: {DESCRIPTION}
   The question is: {QUESTION}
   The choices are: {CHOICES}

2. Now, identify the 'rationales' behind your answer. The rationales are words, phrases or sentences in the original respondent data that led you to answer with `<Original Answer>`. Share a list of rationales with one rationale per line. The list should start with 'RATIONALS:'

3. Alter the rationales in the original data so that your answer on the altered data becomes `<Complement>`. Keep the changes to a minimum. The altered data should start with 'ALTERED DATA:'

C.2.3 CoT prompting

1. You will be provided data corresponding to a survey respondent. The survey was conducted among US residents in 2018. Please answer the question based on the information provided by selecting from one of the two choices. The data provided is enough to reach an approximate answer. Think step by step. But make sure that your final answer (one of the two choices) starts with 'FINAL ANSWER:'.

   The respondent data is: {DESCRIPTION}
   The question is: {QUESTION}
   The choices are: {CHOICES}

2. Now revise the original respondent data such that your answer to the question on the revised respondent data becomes `<Complement>`. Share the revised data. The revised data should start with 'REVISED DATA:'.

C.3 SST2

C.3.1 Unconstrained prompting

- You will be given a movie review. Assess its sentiment and classify it as 'Positive' or 'Negative.' Do not include any additional

words in your answer. Your answer should start with 'ANSWER:'

The movie review is: {MOVIE REVIEW}

- Now revise the original review so that the sentiment of the revised review becomes <Complement>. Share the revised review. The revised review should start with 'REVISED REVIEW:'

C.3.2 Rationale-based prompting

- You will be given a movie review. Assess its sentiment and classify it as 'Positive' or 'Negative.' Do not include any additional words in your answer. Your answer should start with 'ANSWER:'

  The movie review is: {MOVIE REVIEW}

- Now, identify the 'rationales' behind your answer. The rationales are words, phrases or sentences in the original review that led you to answer with <Original Answer>. Share a list of rationales with one rationale per line. The list should start with 'RATIONALS:'

- Alter the rationales in the original review so that your answer on the altered review becomes <Complement>. Keep the changes to a minimum. The altered review should start with 'ALTERED REVIEW:'

C.3.3 CoT prompting

1. You will be given a movie review. Assess its sentiment and classify it as 'Positive' or 'Negative.' Think step by step. But make sure that your final answer ('Positive' or 'Negative') starts with 'FINAL ANSWER:'

   The movie review is: {MOVIE REVIEW}

2. Now revise the original review so that the sentiment of the revised review becomes <Complement>. Share the revised review. The revised review should start with 'REVISED REVIEW:'

C.4 Twitter Financial News

C.4.1 Unconstrained prompting

1. You will be given a finance-related news post from X (formerly Twitter). Assess its sentiment and classify it as 'Bearish,' 'Bullish,' or 'Neutral.' Do not include any additional words in your answer. Your answer should start with 'ANSWER:'.

   The twitter financial news is: {TWITTER POST}

2. Now revise the original post so that the sentiment of the revised post becomes <Complement>. Share the revised post. The revised post should start with 'REVISED POST:'.

C.4.2 Rationale-based prompting

1. You will be given a finance-related news post from X (formerly Twitter). Assess its sentiment and classify it as 'Bearish,' 'Bullish,' or 'Neutral.' Do not include any additional words in your answer. Your answer should start with 'ANSWER:'

   The twitter financial news is: {TWITTER POST}

2. Now, identify the 'rationales' behind your answer. The rationales are words, phrases or sentences in the original Twitter post that led you to answer with <Original Answer>. Share a list of rationales with one rationale per line. The list should start with 'RATIONALS:'

3. Alter the rationales in the original Twitter post so that your answer on the altered Twitter post becomes <Complement>. Keep the changes to a minimum. The altered Twitter post should start with 'ALTERED TWITTER POST:'

C.4.3 CoT prompting

1. You will be given a finance-related news post from X (formerly Twitter). Assess its sentiment and classify it as 'Bearish,' 'Bullish,' or 'Neutral.' Think step by step. But make sure that your final answer ('Bearish', 'Bullish', or 'Neutral') starts with 'FINAL ANSWER:'
   The twitter financial news is: {TWITTER POST}

2. Now revise the original post so that the sentiment of the revised post becomes <Complement>. Share the revised post. The revised post should start with 'REVISED POST:'.

C.5 GSM8K

C.5.1 Unconstrained prompting

1. You will be given a math problem. The solution to the problem is an integer. Your task is to provide the solution. Only provide the final answer as an integer. Do not include any additional word or phrase. You final answer should start with 'FINAL ANSWER:'

   The math Problem is: {PROBELM}

2. Now, revise the math problem so your final answer to the revised problem becomes <Complement>. Share the revised Problem. The revised problem should start with 'REVISED PROBLEM:'

C.5.2 Rationale-based prompting

1. You will be given a math problem. The solution to the problem is an integer. Your task is to provide the solution. Only provide the final answer as an integer. Do not include any additional word or phrase. You final answer should start with 'FINAL ANSWER:'

   The math Problem is: {PROBELM}

2. Now, identify the 'rationales' behind your answer. The rationales are words, phrases or sentences in the original problem that led you to answer with <Original Answer>. Share a list of rationales with one rationale per line. The list should start with 'RATIONALS:'

3. Alter the rationales in the original problem so that your answer on the altered problem becomes <Complement>. Keep the changes to a minimum. The altered problem should start with 'ALTERED PROBLEM:'.

C.5.3 CoT prompting

1. You will be given a math problem. The solution to the problem is an integer. Your task is to provide the solution. Only provide the final answer as an integer. Think step by step. But make sure that your final answer (the integer) starts with 'FINAL ANSWER:'.

   The math Problem is: {PROBELM}

2. Now, revise the math problem so your final answer to the revised problem becomes complement. Share the revised problem. The revised problem should start with 'REVISED PROBLEM:'.

C.6 Multi-Genre Natural Language Inference (MGNLI)

C.6.1 Unconstrained prompting

1. You will be given two sentences denoting a premise and a hypothesis respectively. Determine the relationship between the premise and the hypothesis. The possible relationships you can choose from are 'Entail', 'Contradict' and 'Neutral'. Only pick one of the options. Do not include any additional words in your answer. Your answer should start with 'ANSWER:'

   The premise is: {PREMISE}
   The hypothesis is: {HYPOTHESIS}

2. Now revise the original hypothesis so that your answer to the question about its relationship becomes <Complement>. Share the revised hypothesis. The revised hypothesis should start with 'REVISED HYPOTHESIS:'

C.6.2 Rationale-based prompting

1. You will be given two sentences denoting a premise and a hypothesis respectively. Determine the relationship between the premise and the hypothesis. The possible relationships you can choose from are 'Entail', 'Contradict' and 'Neutral'. Only pick one of the options. Do not include any additional words in your answer. Your answer should start with 'ANSWER:'

   The premise is: {PREMISE}
   The hypothesis is: {HYPOTHESIS}

2. Now, identify the 'rationales' behind your answer. The rationales are words, phrases or sentences in the original hypothesis that led you to answer with <Original Answer>. Share

a list of rationales with one rationale per line. The list should start with 'RATIONALS:'

3. Alter the rationales in the original hypothesis so that your answer on the altered hypothesis becomes `<Complement>`. Keep the changes to a minimum. The altered hypothesis should start with 'ALTERED HYPOTHESIS:'.

C.6.3 CoT prompting

1. You will be given two sentences denoting a premise and a hypothesis respectively. Determine the relationship between the premise and the hypothesis. The possible relationships you can choose from are 'Entail', 'Contradict' and 'Neutral'. Only pick one of the options. Think step by step. But make sure that your final answer ('Entail', 'Contradict' or 'Neutral') starts with 'FINAL ANSWER:'.

   The premise is: {PREMISE}
   The hypothesis is: {HYPOTHESIS}

2. Now revise the original hypothesis so that your answer to the question about its relationship becomes `<Complement>`. Share the revised hypothesis. The revised hypothesis should start with 'REVISED HYPOTHESIS:'

D Postprocessing model outputs

1. Post-processing for all datasets starts by normalizing the model's short answer, such as 'Yes.' or 'Yes!' are converted to 'Yes'. We also remove common extra characters that models tend to add to their answers, such as (*, \, ', ., !, ?, '., ..).

2. Filtering and removing model generations where the model's first answer is not valid. This means the model did not pick one of the valid options as an answer (*e.g.*, 'Yes' or 'No' in DISCRIMEVAL).

3. Filtering out cases when SCEs are shorter than expected. Short or incomplete generations typically occur when the model fails to provide a full SCE or returns a non-response. To avoid accidentally filtering out valid but concise outputs, we determined the thresholds for "short" generations empirically. We manually analyzed samples from each dataset and set minimum word-length criteria based on the distribution of reasonable completions. The thresholds for filtering short cases are as follows:

   - DISCRIMEVAL: Generations with fewer than 15 words
   - TWITTER FINANCIAL NEWS: Fewer than 3 words
   - FOLKTEXTS: Fewer than 60 words
   - MGNLI: Fewer than 2 words
   - SST2: Fewer than 1 word
   - GSM8K: Generations containing fewer than 5 words and consisting solely of alphabetic characters, with no numbers or mathematical symbols.

4. For rationale based prompting, we remove cases where the model is unable to generate rationales. If the model fails to detect the important part of the text for answering, we do not consider its SCEs generation since the SCE generation instruction specifically refers to the rationales (Appendix C).

5. Some models in certain datasets included their answers in the SCE they generated. The presence of the answer biased the model prediction on on the SCE.To address this, we removed the answer tags from the SCEs when present.

6. We explicitly instructed the model to begin its response with specific keywords such as `ANSWER`, `RATIONALS` and `REVISED SCENARIO`. The models still tend to add synonymous labels like `ALTERED SCENARIO`. We manually analyze model outputs and whitelist these labels. The precise extraction process is:

   - **Extracting an Answer:** If the decoded response contains the string **'ANSWER:'**, we extract everything that comes after the last occurrence of **'ANSWER:'**.
   - **Extracting a Rationale:** If we are extracting a rationale, we look for the part of the decoded response that starts with **'RATIONALS'**.
   - **Extracting a CE:** For counterfactual generation, the special starting word depends on both the dataset and the prompt type. Specifically:

- **DiscrimEval:**
  - `Unconstrained` → 'REVISED SCENARIO:'
  - `Rational_based` → 'ALTERED SCENARIO:'
- **Folktexts:**
  - `Unconstrained` → 'REVISED DATA:'
  - `Rational_based` → 'ALTERED DATA:'
- **GSM8K:**
  - `Unconstrained` → 'REVISED PROBLEM:'
  - `Otherwise` → 'ALTERED PROBLEM:'
- **SST2:**
  - `Unconstrained` → 'REVISED REVIEW:'
  - `Otherwise` → 'ALTERED REVIEW:'
- **Twitter:**
  - `Unconstrained` → 'REVISED POST:'
  - `Otherwise` → 'ALTERED TWITTER POST:'
- **NLI:**
  - `Unconstrained` → 'REVISED HYPOTHESIS:'
  - `Otherwise` → 'ALTERED HYPOTHESIS:'

E Statistical Analysis of Results

We computed 95% Confidence Intervals (CIs) for generation percentage, validity percentage, and edit distance to assess whether the differences between the *with context* and *without context* conditions are statistically significant. Non-overlapping CIs mean that the results for the two conditions differ more than what we would expect just from random variation. This usually points to a statistically significant difference (roughly corresponding to $p < 0.05$). The CIs were calculated using the standard error of the mean:

$$\text{CI} = \text{mean} \pm 1.96 \times \left(\frac{\text{sd}}{\sqrt{n}} \right)$$

Here, *mean* is the average value, *sd* is the standard deviation, and n is the number of samples. The factor 1.96 corresponds to a 95% confidence level under a normal distribution.

F Correlation between validity and popular performance metrics

We explored the relationship between the validity of SCEs and several model properties, including *Model Size*, *Perplexity*, *HuggingFace Leaderboard Rank*, and *MMLU Accuracy*. However, we did not observe any clear or consistent patterns. Additionally, we performed both PEARSON and SPEARMAN correlation tests to check for non-zero correlation coefficient,[1] but **none of the correlations were statistically significant, with all P-VALUES exceeding** 0.05. In the following subsection, we present the results of some of these analyses.

F.1 Validity of SCEs vs. Model Size across Datasets

Figure 3 illustrates how SCE validity varies with model size across datasets. While one might expect larger models to consistently perform better, this is not always the case—smaller models sometimes generate more valid SCEs. Overall, we observe no consistent correlation between model size and counterfactual reasoning ability.

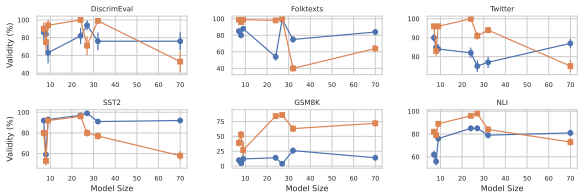


Figure 3: Validity of SCEs vs. Model Size across Datasets. Orange indicates validity with context; blue indicates validity without context.

F.2 Model perplexity vs. SCEs validity

We used the lm-eval framework[2] to compute five-shot perplexity on the WIKITEXT (Merity et al., 2016) benchmark for each model, and then analyzed its correlation with the percentage of valid SCEs generated. The decision to use lm-eval aligns with best practices for reproducible, transparent, and comparable evaluation, as emphasized by Biderman et al. (2024) . By adopting a controlled few-shot setup, we reduce variance across evaluations and ensure our perplexity scores reflect meaningful differences in model behavior rather than implementation artifacts. Measuring perplexity in

[1] Using `https://scipy.org`

[2] `https://github.com/EleutherAI/lm-evaluation-harness`

this standardized way enables a principled comparison with SCEs validity, allowing us to probe whether language models with lower perplexity exhibit stronger counterfactual reasoning. However, as shown in Figure 2, we did not observe a clear relationship between few-shot perplexity and SCE validity across models.

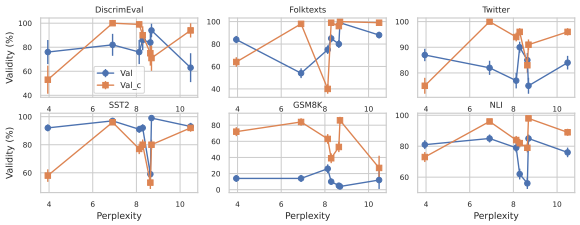


Figure 4: Effect of Model Size and Context on SCE Validity across Datasets. Blue lines indicate the percentage of valid SCEs generated without context, while orange lines represent validity with context. Results are shown for six benchmark datasets.

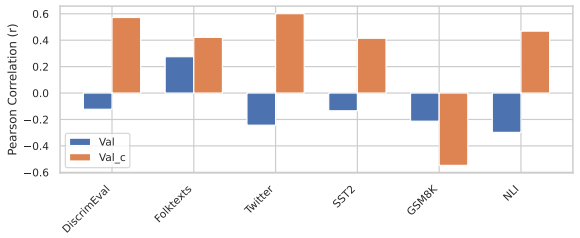


Figure 5: Pearson correlation between model perplexity and SCE validity across datasets. Bars show Pearson correlation coefficients (r) between few-shot perplexity and validity percentage. Orange bars represent validity with context, and blue bars represent validity without context. Positive values indicate that higher perplexity is associated with higher SCE validity; negative values indicate the reverse.

F.3 Leaderboard Rank vs. SCEs validity

We obtained the Hugging Face Leaderboard ranks for all models except $\texttt{MST}_\texttt{m}$ (which was not listed) and plotted their ranks against SCE validity percentages. However, we observed no clear correlation between Leaderboard ranking and SCE validity.

F.4 MMLU Accuracy vs. SCEs validity

The MMLU (Massive Multitask Language Understanding) benchmark by Hendrycks et al. (2020) evaluates a model's performance across 57 diverse academic and professional subjects, including law, physics, computer science, and history. It uses multiple-choice questions to assess the model's

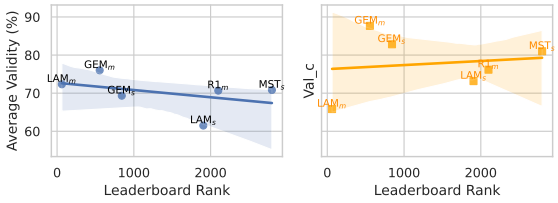
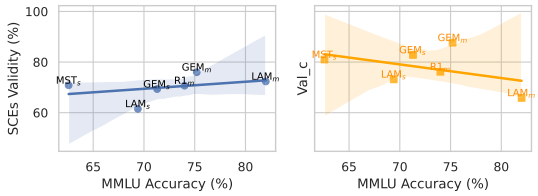
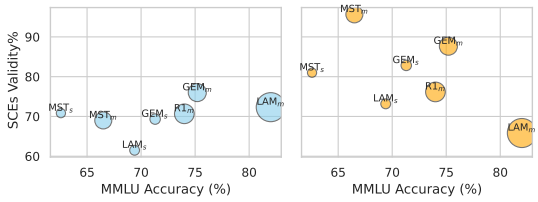


Figure 6: Relationship between Hugging Face Leaderboard rank and SCE validity. Each point represents a model. The left panel shows average SCE validity without context, and the right panel shows validity with context. Lower ranks indicate higher leaderboard positions. Regression lines with 95% confidence intervals illustrate trends between leaderboard rank and SCE validity.

breadth of knowledge and ability to handle a wide range of tasks. We examined the correlation between models' MMLU performance and Percentage of valid SCEs, but found no significant relationship, models with higher MMLU accuracy do not necessarily have a high SCEs validity percentage.



(a) Linear regression between MMLU accuracy and SCEs validity.



(b) Bubble plot showing the relationship between model size and SCE validity. Each bubble represents a model, where the x-axis indicates model size and the y-axis shows the percentage of valid SCEs. Bubble size is proportional to model size (scaled by a factor of 10).

Figure 7: Relationship between MMLU accuracy and SCEs validity percentages across models. **Blue** indicates SCEs validity **without context**, while **orange** indicates SCEs validity **with context**.