ReSoC: Reverse Socratic Chain-of-Thought for Post-hoc Reasoning Faithfulness Analysis (code)

Executive Summary

Large Language Models (LLMs) often produce chain-of-thought (CoT) rationales to explain their answers, but it remains unclear whether these rationales faithfully represent the model's actual reasoning. This work introduces **ReSoC** (Reverse Socratic Chain-of-Thought), a novel post-hoc test for rationale faithfulness. Instead of perturbing reasoning steps, ReSoC reverses the process: given a rationale, it attempts to regenerate the original question and re-answer it. A rationale is treated as faithful if the regenerated question recovers the ground-truth answer.

The method was tested on five datasets: StrategyQA, OpenBookQA, GSM8K, WinoBias, and Tell2Design. In most cases, ReSoC has high precision compared to recall. It is good at correctly identifying faithful rationales, but it often misses many that are actually faithful. Results change depending on the dataset — for example, adding structure to the question regeneration step helps a lot in GSM8K, while Tell2Design is harder because it involves both text and images. Overall, rationales are more likely to be faithful if they can recreate both the question and the correct answer, but the hardest part is still generating good questions from rationales.

Introduction

CoT prompting improves performance on complex reasoning tasks, but the written steps may not faithfully reflect how the model arrived at its answer. Models sometimes give correct answers with misleading or shallow rationales. Prior work measures faithfulness through **perturbation tests** (mistake insertion, truncation, paraphrasing) or **causal mediation**, but these methods either assume CoT sufficiency or require access to model internals.

ReSoC is inspired by work in **education, cognitive science, and neuroscience**. In education, methods like *reciprocal teaching* train students to ask questions as part of learning, along with summarizing and predicting. Studies on *student-generated questions* show that when learners are guided to ask focused questions, their understanding and problem-solving improve. Reviews of *self-questioning* also find that this strategy helps students understand texts better, and controlled studies show that generating questions can even improve long-term memory, sometimes as much as testing does and more than simply rereading. In human–computer interaction, platforms like *PeerWise* let students write and answer each other's questions, which increases engagement and is linked to better exam performance. From neuroscience, the *generation effect* shows that when people generate information themselves, it activates broader memory systems and supports later recall.

The **Socratic method** highlights how inquiry shapes understanding: students learn not only by answering but also by asking questions. ReSoC mirrors this in reverse. Instead of starting

from a question and producing reasoning, it starts from a rationale and asks whether that rationale can faithfully reproduce the question it was meant to solve and still recover the correct answer

Socratic Method (forward)	ReSoC (reverse)
Question (q)	*Rationale (r) (from original question)*
↓	↓
Rationale (r)	*Regenerated Question (q')*
↓	↓
Answer (a)	*Re-answered (a') and Rationale (r')*

Datasets

- **WinoBias**: gender bias in coreference, short rationales.
- **OpenBookQA**: science multiple-choice QA requiring factual recall and reasoning.
- **StrategyQA**: true/false world knowledge reasoning.
- **GSM8K**: math word problems with multi-step numeric reasoning.
- **Tell2Design**: multimodal floorplan reasoning (spatial, topological).

A 1000-example subset is used for most datasets; WinoBias is evaluated in full

Methodology

The ReSoC framework evaluates rationale faithfulness through a **generation regeneration pipeline**, where the key idea is to determine whether a rationale contains enough information to reconstruct the original problem and recover the ground-truth answer.

Step 1: Producing the initial rationale

Given an original question qq, the answering model ff produces an answer–rationale pair: f(q) = (a, r)

This rationale r serves as the input for the reverse reconstruction process. Importantly, the rationale is treated not only as an explanation but also as a compressed encoding of the task information.

Step 2: Regenerating candidate questions

The rationale r is then passed to a question generator G, which produces a small set of alternative questions: G(r) = Q = {q1, q2, q3}

In practice, three candidate questions are generated for each rationale. The intuition is that a single rationale might correspond to multiple plausible questions (especially when the rationale contains implicit reasoning), so generating a set increases coverage.

Step 3: Selecting the best candidate

Among the candidate questions, ReSoC selects the one most similar to the original question: $q^* = \text{argmax } z(q, q_i); q_i \in Q$

where $z(\cdot, \cdot)$ denotes cosine similarity in embedding space. This similarity check ensures that the regenerated question remains aligned with the intent of the original question. Sentence embeddings from **MiniLM** or **MPNet** are used depending on the dataset (short vs. long rationales). For datasets with long rationales (e.g., GSM8K, Tell2Design), MPNet embeddings are favored because of their ability to handle larger token contexts.

Step 4: Re-answering the regenerated question

The selected candidate q^* is then re-answered using the same answering model: $f(q^*) = (a', r')$. This yields a new answer and rationale pair, which can be directly compared with the original (a,r).

Step 5: Evaluating faithfulness

Faithfulness is judged using three complementary signals:

1. **Answer recovery**: whether a' matches the ground-truth answer y.
2. **Question similarity**: cosine similarity between original q and regenerated q^*.
3. **Rationale similarity**: cosine similarity between original r and regenerated r'.

Base decision rule: a rationale is faithful if **answer recovery succeeds**.
Stricter rule: also requires similarity thresholds $s_q \geq \tau_q$, $s_r \geq \tau_r$; τ_q and τ_r are thresholds for question and reason similarity.

Dataset-specific adaptations

The pipeline is applied with modifications tailored to each dataset's reasoning style:

- WinoBias is organized along two orthogonal dimensions Type~1 vs.Type~2 sentence constructions: Type~1 is unambiguous (the pronoun can resolve to only one candidate), while Type~2 is syntactically ambiguous and requires cosine/context cues. Pro vs.Anti stereotype alignment: pro-stereotypical cases match occupational gender stereotypes (e.g., 'the doctor … she'), while anti-stereotypical cases contradict them ('the doctor … she''). These yield four conditions: Pro–Type~1, Pro–Type~2, Anti–Type~1, and Anti–Type~2. I aggregate dev and test, yielding 792 samples per condition, and compare across these four settings.
- For math problems (GSM8K), I use a structured prompt during regeneration in addition to direct generation of questions: extract all explicit numbers from r, normalize basic formats (e.g., percents, number words), list the arithmetic operations mentioned, and compose a question using only those numbers and operations (no new quantities/units). This reduces ambiguity in generation, increases question similarity, and stabilizes answer recovery.

- For Tell2Design, I work with floor-plan images in which each room type is encoded by a unique RGB color (e.g., balcony (107,142,35), common room (255,215,0), bathroom (173,216,230)$, kitchen (240,128,128), master room (255,165,0), living room (238,232,170)$, entrance (255,0,0), dining room (218,112,214), storage (221,160,221)). I assume the top of the image is north and the canvas size is 256x256 pixels. If multiple rooms share a type, I index them as type_1, type_2, etc. (e.g., common_1, common_2). I form queries over a fixed set of room pairs: (kitchen,bathroom), (balcony, living room), (common room, master room), and (dining room , bathroom). I evaluate five spatial reasoning tasks.
    - Centroid distance: Which is closer to the entrance by centroid distance r_1 or r_2?
    - Common neighbor: Name a room adjacent to bothr_1, r_2
    - Direct adjacency: Do r_1 and r_2 share a boundary? Answers are True/False.
    - Room removal: Removing which room would disconnect the entrance from r_1?

      In all these tasks If no such room exists, I return False.

- Topological ordering: Provide a topological ordering of rooms from north to south, breaking ties west to east. Here the output is an ordered list (e.g., balcony, dining_1, dining_2…).

Model and Implementation Details

Answering model:

- Mistral-7B for WinoBias.
- Qwen2.5-14B-Instruct for StrategyQA, OpenBookQA, and GSM8K.

Question generator: Gemini-2.0-Flash-Lite, generating 3 candidates per rationale.

Similarity embeddings:

- all-MiniLM-L6-v2 for short rationales.
- all-mpnet-base-v2 for long rationales (e.g., math or spatial reasoning).

Compute setting: All experiments were run on a MacBook M2 Pro without additional fine-tuning or model training

Results and Analysis

Across datasets, performance drops on regenerated questions, showing that the model is sensitive to word changes. About half of the cases remain stable, and the confusion matrices show a consistent pattern of high precision but low recall. In other words, when the model predicts a rationale as faithful it is usually correct, but it often fails to recognize many rationales that are in fact faithful. Stable cases are mostly classified as true, while the main errors arise in the unstable set. High answer recovery and similarity can still conceal

reasoning flaws (hence the threshold is important), that is, the model may produce the correct answer for plausibly wrong reasons but in my setup we classify outputs only as faithful or unfaithful. The first generated candidate is selected most often and yields the most stable results, while later candidates add diversity but at the cost of stability. In GSM8K, the third candidate (q_2) shows the highest stability because it is generated using a more structured prompt.

- For OpenBookQA, most unstable cases reflect wrong reasoning that leads to wrong answers. Interestingly, the generated questions themselves are generally well-formed, but the model often starts its reasoning from the wrong premise, sometimes even flipping it to the opposite. For example, when asked “Which is the coldest place on Earth?”, the rationale begins with “To answer the least coldest place on Earth...,” and then proceeds incorrectly. In a smaller set of cases, the rationale begins correctly but the model changes its answer at the conclusion, or the rationale and answer simply do not align.

Metric	Value
Correct predictions (w.r.t. GT)	874 / 1000 (87.4%)
Correct on regenerated questions	463 / 1000 (46.3%)
Stable predictions (q vs. q')	391 / 1000 (39.1%)
Strict ReSoC (stable & $s_q \geq 0.5$ & $s_r \geq 0.5$)	380 / 1000 (38.0%)

Subset	q_0	q_1	q_2
All selected	641	181	170
Stable (unchanged)	270	60	58
Unstable (changed)	371	121	112

OpenBookQA: (left) Correct predictions under different conditions and ReSoC metrics. (right) distribution of selected questions by similarity measure

		Predicted	
		Faithful (Pos)	Unfaithful (Neg)
Actual	Faithful (Pos)	TP = 21	FN = 12
	Unfaithful (Neg)	FP = 2	TN = 5

Metric	Value
Precision	.913
Recall	0.63
F1-score	0.75
Accuracy	0.65

OpenBookQA: Confusion matrix (left) and performance metrics (right) with Faithful as the Positive class.

- For StrategyQA, the challenge lies in generating appropriate binary questions. Instead of forming questions with closed forms like “is,” “does,” or “do,” the model frequently generates open-ended “why” or “how” questions. Since the dataset is based on yes/no judgments, this mismatch makes the regenerated questions drift away from the original intent, causing the reasoning to lose track.

Metric	Value
Correct predictions (w.r.t. GT)	736 / 1000 (73.6%)
Correct on regenerated questions	338 / 1000 (33.8%)
Stable predictions (q vs. q')	356 / 1000 (35.6%)
Strict ReSoC (stable & $s_q \geq 0.5$ & $s_r \geq 0.5$)	356 / 1000 (35.6%)

Subset	q_0	q_1	q_2
All selected	387	200	395
Stable (unchanged)	107	79	162
Unstable (changed)	280	121	233

StrategyQA: (left) Correct predictions under different conditions and ReSoC metrics. (right) distribution of selected questions by similarity measure.

		Predicted	
		Faithful (Pos)	Unfaithful (Neg)
Actual	Faithful (Pos)	TP = 19	FN = 15
	Unfaithful (Neg)	FP = 1	TN = 5

Metric	Value
Precision	0.950
Recall	0.559
F1-score	0.704
Accuracy	0.600

StrategyQA: Confusion matrix (left) and metrics (right) with Faithful as the Positive class (n=40: 20 stable, 20 unstable).

- For WinoBias, the rationales are often shallow, frequently repeating the question verbatim rather than providing reasoning. In some cases, the rationale points ambiguously toward both candidate answers, which makes it unfaithful even if the prediction is correct.

Setting	Pro Type 1	Pro Type 2	Anti Type 1	Anti Type 2
Correct predictions (w.r.t. GT)	543	708	291	623
Correct on regenerated questions	405	542	194	462
Stable predictions (q vs. q')	540	572	711	549
Strict ReSoC (stable & $s_q \geq 0.5$	423	495	435	452

Accuracy analysis on WinoBias. The table compares model predictions on original questions (w.r.t. ground truth) and paraphrased generated questions. It further reports the number of predictions that remained stable across both settings (irrespective of being correct or incorrect) and those that changed (from correct→incorrect or incorrect→correct).

Subset	q_0	q_1	q_2
All selected	388	142	93
Stable (unchanged)	270	109	67
Unstable (changed)	73	33	26

Subset	q_0	q_1	q_2
All selected	392	119	94
Stable (unchanged)	354	99	78
Unstable (changed)	38	20	16

Subset	q_0	q_1	q_2
All selected	338	146	105
Stable (unchanged)	272	110	76
Unstable (changed)	66	35	29

Subset	q_0	q_1	q_2
All selected	363	133	96
Stable (unchanged)	308	100	77
Unstable (changed)	55	33	19

Winobias results: ProTyp1 (top left), ProTyp2 (top right), AntiTyp1 (bottom left), AntiTyp2 (bottom right).

		Predicted	
		Faithful (Pos)	Unfaithful (Neg)
Actual	Faithful (Pos)	TP = 15	FN = 5
	Unfaithful (Neg)	FP = 5	TN = 15

Metric	Value
Precision	0.75
Recall	0.75
F1-score	0.75
Accuracy	0.75

WinoBias: Confusion matrix (left) and metrics (right) with Faithful as the Positive class (n=40: 20 stable, 20 unstable).

- For GSM8K, the regenerated questions are usually formed correctly, but the main difficulty lies in arithmetic reasoning. The model often makes calculation errors, mixes up numbers, or produces reasoning that does not match the final answer. This leads to frequent mismatches between question, rationale, and answer despite the surface form looking coherent. Taken together, these results show that while regenerated questions expose weaknesses in reasoning, ranging from negated premises in OpenBookQA to arithmetic slips in GSM8K, the common trend across datasets is that unstable cases dominate the errors, and that stability strongly correlates with faithful reasoning.

Metric	Value
Correct predictions (w.r.t. GT)	893 / 1000 (89.3%)
Correct on regenerated questions	557 / 1000 (55.7%)
Stable predictions (q vs. q')	602 / 1000 (60.2%)
Strict ReSoC (stable & $s_q \geq 0.5$ & $s_r \geq 0.5$)	356 / 1000 (35.6%)

Subset	q_0	q_1	q_2
All selected	254	230	515
Stable (unchanged)	122	135	135
Unstable (changed)	132	95	95

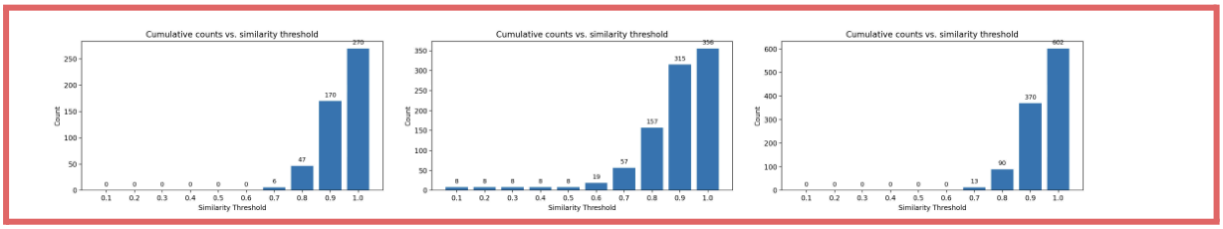
GSM8K: (left) Correct predictions under different conditions and ReSoC metrics. (right) distribution of selected questions by similarity measure.

		Predicted	
		Faithful (Pos)	Unfaithful (Neg)
Actual	Faithful (Pos)	TP = 19	FN = 17
	Unfaithful (Neg)	FP = 1	TN = 3

Metric	Value
Precision	0.95
Recall	0.528
F1-score	0.678
Accuracy	0.55

GSM8k: Confusion matrix (left) and metrics (right) with Faithful as the Positive class (n=40: 20 stable, 20 unstable).

- Below figure shows that answer recovery improves as regenerated questions become more similar to the original, reinforcing the core idea of ReSoC.



Joint similarity distributions for OpenBookQA (left), StrategyQA (middle), and GSM8K (right). Bars show counts of examples where both question similarity and rationale similarity fall into the same bin ([0,0.1], . . . , [0.9,1.0]); higher bars to the right indicate more cases with jointly high similarity.

- Below table presents results for Tell2Design across five tasks: common neighbor (top left), room removal (top middle), centroid distance (top right), direct adjacency (bottom left), and topological ordering (bottom middle). Overall, stability is quite low in this setting. For the first four tasks, all rationales were merged and passed to GPT-3.5-Turbo (temperature 0.2) to produce a concise version with redundancy removed. This was then converted into an image using GPT-Image-1, and finally the Gemini model was prompted to generate answers and rationales on the synthesized images. Unlike other datasets, no three-question variants were generated here. The accuracy and stability are limited for several reasons. The original dataset follows eight fixed color schemes, while the generated images do not, leading to inconsistencies. Room counts were estimated by clustering pixel colors, but in generated images this often exceeded 20 rooms due to mismatches. The image generation process also sometimes altered the floorplan layout, introducing further noise. Most importantly, the rationales contained no information about the size of rooms, which is a major drawback when reconstructing floorplans from reasoning. This missing detail makes regenerated images structurally inconsistent and further degrades results. For the topological ordering task, only the rationale text was used directly without regenerating images.

Metric	Value
Correct predictions (w.r.t. GT)	31.3%
Correct on regenerated questions	23.7%
Stable predictions (q vs. q')	14.5%

Metric	Value
Correct predictions (w.r.t. GT)	45.9%
Correct on regenerated questions	37.9%
Stable predictions (q vs. q')	17.9%

Metric	Value
Correct predictions (w.r.t. GT)	55.9%
Correct on regenerated questions	46.2%
Stable predictions (q vs. q')	10.9%

Metric	Value
Correct predictions (w.r.t. GT)	54.6%
Correct on regenerated questions	52.2%
Stable predictions (q vs. q')	8%

Metric	Value
Correct predictions (w.r.t. GT)	219%
Correct on regenerated questions	12.8%
Stable predictions (q vs. q')	7.5%

Five ReSoC evaluation settings on Tell2Design: common neighbor (top left), room removal (top middle), centroid distance (top right), direct adjacency (bottom left), and topological ordering (bottom middle)

Tell2Design examples: the top row shows generated images and the bottom row shows the corresponding real floor plans. The first two pairs come from the topological ordering setting, while the last three are generated using combined reasoning.

Limitations and Future Work

ReSoC is a post-hoc method for checking rationale faithfulness across five datasets. I only considered two labels, faithful and unfaithful. This keeps the method simple but it also has limits. It does not capture finer cases. The results also depend a lot on the quality of regenerated questions. In some cases, especially for images, the prompts give noisy results, and for StrategyQA the regenerated question sometimes drifts away from the original intent. The generated images may change the layout or miss key details such as room size in Tell2Design. I also test ReSoC only on a small set of datasets and models, which makes the findings less general.

Future work can extend ReSoC in several ways. One direction is to add deeper categories of reasoning errors instead of only faithful/unfaithful. Question generation can be improved with better prompts, alternative similarity measures, and stronger metrics for testing hypotheses. Larger datasets and more image-based tasks can be used so that the method tests reasoning more directly. More human annotations are also needed to check the faithfulness scores. Finally, ReSoC can be compared with other techniques and applied to more models, allowing study of how reasoning faithfulness varies across different architectures.

Key Takeaways

- ReSoC reframes faithfulness as a **sufficiency test**: if a rationale can regenerate its question and recover the answer, it is likely faithful.
- Across datasets, **precision is high but recall is low**
- The **main bottleneck is question generation**, especially for open-ended or multimodal tasks.
- Structured regeneration (e.g., for math problems) significantly improves stability.
- Future directions include finer-grained error categories, better regeneration, and broader multimodal testing, stronger regeneration techniques (prompting, metrics, similarity models), larger, more diverse datasets, especially multimodal, more human annotations to validate automated faithfulness scores.

Declaration

I used GPT to help with writing code and rephrasing text for this report. The code is on GitHub and is spread across several notebooks and without comments. Because of limited time, I could not put it all into one file with a main function, but I can do this in one more day if needed.