DETECTING UNFAITHFUL CHAIN-OF-THOUGHT VIA CIRCUIT-GUIDED INTERNAL-EXTERNAL DISCREPANCY

Anonymous authors
Paper under double-blind review

ABSTRACT

Chain-of-thought (CoT) reasoning improves the problem-solving ability of large language models (LLMs), but generated reasoning traces may not faithfully reflect the model's actual decision process. Existing CoT unfaithfulness detectors mainly rely on external signals from generated rationales, such as textual plausibility or answer consistency, while overlooking evidence from the model's internal computation. Although recent circuit tracing methods provide a way to obtain model-internal evidence by tracing how information flows through model components during reasoning, constructing full reasoning circuits for long CoTs is costly and difficult to scale. To address these challenges, we propose **C**ircuit-guided **I**nternal-**E**xternal Discrepancy Scorer (CIE-SCORER), a framework for instance-level CoT unfaithfulness detection. The key idea is that faithful reasoning traces should align with the model's computational process, whereas unfaithful traces may diverge from it. CIE-SCORER efficiently traces compact sentence-level circuits from informative reasoning tokens, constructs internal and external reasoning graphs, and measures their discrepancy using Fused Gromov–Wasserstein distance. Experiments on four datasets from FaithCoT-Bench show that CIE-SCORER achieves state-of-the-art performance while reducing the cost of circuit construction, demonstrating the effectiveness of combining mechanistic interpretability signals with external reasoning traces for CoT unfaithfulness detection.

1 INTRODUCTION

Chain-of-thought (CoT) reasoning has become a central mechanism for improving the reasoning ability of large language models (LLMs), by enabling them to produce intermediate textual steps before committing to a final answer (Wei et al., 2022; Kojima et al., 2022). Such reasoning traces have led to substantial performance gains in mathematics, programming, logic, and commonsense reasoning (Imani et al., 2023; Jiang et al., 2024; Huang & Chang, 2023; Zhao et al., 2023), and are treated as an intuitive window into the model's reasoning process (Yao et al., 2023; Besta et al., 2024; Yeo et al., 2025). However, recent studies have shown that this *apparent transparency* can be misleading: although CoT traces resemble step-by-step reasoning in natural language, they may not correspond to the internal mechanisms that underlie the model's predictions (Feng et al., 2023; Zhao et al., 2025a; Barez et al., 2025; Matton et al., 2025). Existing works have examined surface-level reliability concerns, including hallucination, factual correctness, and rationale quality (Wang et al., 2025a; Qi et al., 2024; Huang et al., 2025; Park et al., 2025). These concerns are related to the notion of ***faithfulness***, which asks whether an explanation reflects the model's internal decision process (Shen et al., 2025b; Lyu et al., 2023). An ***unfaithful*** explanation may be coherent and persuasive while providing misleading accounts of model behavior, limiting its usefulness for understanding or auditing model behavior (Turpin et al., 2023; Agarwal et al., 2024). Therefore, determining whether CoT reasoning is faithful is essential for evaluating its role as an explanation of LLM behavior.

Recent studies examine CoT faithfulness from two main perspectives. One line of work evaluates the causal strength between CoT traces and final predictions through counterfactual interventions, where parts of the CoT are perturbed to examine their effect on the final prediction (Lanham et al., 2023; Turpin et al., 2023; Yang et al., 2025; Xiong et al., 2025). While effective for revealing whether CoT functions as a faithful explanation, such analysis does not directly determine the faithfulness

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

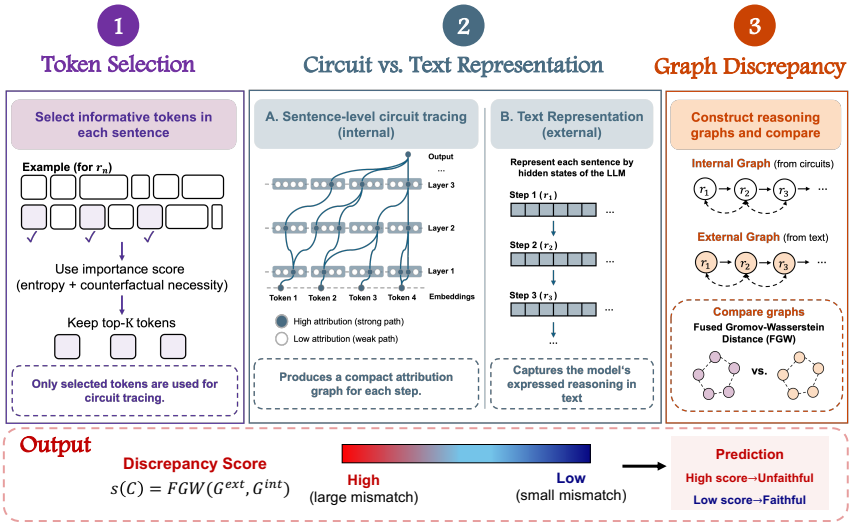


Figure 1: Overview of the CIE-SCORER for CoT unfaithfulness detection. The framework selects informative reasoning tokens and traces sentence-level internal attribution circuits from model activations. These internal circuit representations are compared against external text-based reasoning representations. The discrepancy between the two reasoning graphs is measured using FGW distance.

of a specific generated trace. A second line evaluates reasoning traces through LLM-as-Judge or process reward models, treating erroneous reasoning with correct answers or plausible reasoning with incorrect answers as indicators of unfaithfulness (Arcuschin et al., 2025; Wen et al., 2025).However, reasoning correctness and faithfulness are not equivalent: a reasoning trace may arrive at the correct answer while failing to reflect the model's actual decision process. Since faithfulness needs to be assessed for individual reasoning traces, Shen et al. (2025b) introduces the FINE-CoT dataset for training and evaluating CoT unfaithfulness detectors, which we use as the primary benchmark in our study. Taken together, current detection methods for CoT unfaithfulness still rely largely on the external content of generated reasoning traces, leaving the model's internal computation underexplored. This raises ***Challenge 1:*** *how can we detect CoT unfaithfulness using evidence from the model's internal decision process rather than only surface-level reasoning quality?*

White-box interpretability offers a way to move beyond surface-level reasoning quality by examining the internal mechanisms (Zhao et al., 2024; Du et al., 2019). In particular, circuit tracing methods analyze how information flows through internal model components during reasoning, allowing us to identify which internal features and interactions contribute to the prediction (Gao et al., 2024; Hanna et al., 2024; Ameisen et al., 2025). These internal traces offer a source of evidence for detecting CoT unfaithfulness. However, existing mechanistic interpretability pipelines remain computationally demanding and difficult to scale, especially for long reasoning traces. Constructing attribution graphs across many reasoning steps requires tracing large numbers of internal interactions, making full circuit construction costly. Although recent work uses step-level circuit topology to train classifiers for reasoning correctness (Zhao et al., 2025b), it is limited to short reasoning traces. This gives rise to ***Challenge 2:*** *how can we exploit internal interpretability signals for CoT unfaithfulness detection while avoiding the prohibitive cost of full circuit construction?*

Based on the insights above, we propose **C**ircuit-guided **I**nternal-**E**xternal Discrepancy Scorer (CIE-SCORER) for CoT unfaithfulness detection. To address ***Challenge 1***, CIE-SCORER leverages circuit-level evidence to characterize the model's internal computational trajectory, allowing unfaithfulness to be detected from a mismatch between internal computation and external reasoning traces. To address ***Challenge 2***, we select informative tokens in each reasoning sentence before circuit tracing, enabling analysis without constructing full circuits over CoTs. The resulting sentence-level computational graphs form a circuit sequence, which is compressed into internal reasoning embeddings. We then compare this internal trajectory with the external reasoning trajectory by constructing two sentence graphs under a shared graph-building rule: an internal graph constructed from the compressed circuit embeddings and an external graph constructed from sentence representations encoded by a frozen

BERT encoder. The key intuition is that faithful reasoning traces should align with the model's internal computational process, whereas unfaithful traces may diverge from it. After aligning the two latent spaces with a lightweight adaptor, we compute the Fused Gromov–Wasserstein (FGW) distance between the graphs and use it as the unfaithfulness score. An overview of CIE-SCORER is shown in Fig. 1. To the best of our knowledge, CIE-SCORER is the first approach to explicitly combine the model's internal computational process with its externally displayed reasoning trace. Experiments on four datasets from FaithCoT-Bench demonstrate that CIE-SCORER achieves state-of-the-art performance. Our main contributions are summarized as follows:

- We formulate CoT unfaithfulness detection from a **circuit-guided internal-external discrepancy** perspective, to our knowledge, the first to integrate model-internal computation with external reasoning traces for instance-level faithfulness analysis.
- We propose CIE-SCORER, which performs token-selected circuit tracing, compresses sentence-level circuit graphs into internal reasoning embeddings, and compares them with external reasoning representations using Fused Gromov–Wasserstein distance.
- Experiments on four FaithCoT-Bench datasets show that CIE-SCORER achieves state-of-the-art performance against representative baselines, validating the effectiveness of efficient circuit-guided detection.

2 PRELIMINARIES

2.1 COT UNFAITHFULNESS DETECTION

CoT reasoning generates intermediate reasoning steps before producing a final answer. Given a query q and a prompting instruction p, a large language model $\mathcal{M}$ produces a reasoning trajectory $\mathcal{C} = (r_1, r_2, \ldots, r_T)$, where r_t denotes the t-th reasoning step and T is the trajectory length, followed by a final answer a. Although $\mathcal{C}$ is often treated as an explanation for a, it may not faithfully reflect the model's actual decision process. Prior studies have mainly examined CoT unfaithfulness at the population level, using counterfactual interventions to test whether CoT explanations are causally linked to final answers. In contrast, Shen et al. (2025b) formulates CoT faithfulness as an instance-level detection problem, where each query–CoT instance is assigned a binary label based on observable signals of unfaithfulness.

Definition 1 (Instance-level CoT Unfaithfulness Detection) *Given a query q and a generated CoT trajectory $\mathcal{C} = (r_1, r_2, \ldots, r_T)$ produced by a large language model $\mathcal{M}$, CoT unfaithfulness detection aims to decide whether $\mathcal{C}$ faithfully reflects the model's internal reasoning process, denoted as $\mathcal{R}_{\mathcal{M}}(q)$. Formally, this task is defined as a binary classification function:*

$$f : (q, \mathcal{C}) \mapsto \{0, 1\}, \tag{1}$$

where $f(q, \mathcal{C}) = 1$ indicates that $\mathcal{C}$ is unfaithful to $\mathcal{R}_{\mathcal{M}}(q)$, and $f(q, \mathcal{C}) = 0$ indicates that $\mathcal{C}$ is faithful. Different detection methods instantiate f using different observable evidence.

2.2 CIRCUIT TRACING

To reveal how the model *internally* produces its reasoning, we adopt circuit tracing (Ameisen et al., 2025) as an interpretable, white-box approximation of the computation behind each generated token. It runs on a transcoder-based replacement model that decomposes dense MLP activations into sparse, overcomplete features, enabling feature-level attribution. Given a prefix p (e.g. a prompt, or a prompt followed by a partial CoT), tracing attributes the model's prediction of the next token (the start of the continued reasoning) and returns a prefix-specific attribution graph $\mathrm{Trace}(p) = \mathcal{G}_{\mathrm{attr}} = (\mathcal{V}, \mathcal{E})$. The node set $\mathcal{V}$ contains *input* nodes (the prefix tokens), *feature* nodes (sparse transcoder features that fire at specific layers and positions), and *output* nodes (candidate next-token logits, i.e. the possible first tokens of the next reasoning step). The directed edges $\mathcal{E}$ carry causal attribution weights measuring how strongly one feature drives another feature or an output. This graph decomposes a next-token prediction into sparse feature activations and their directed contributions, exposing the internal information flow that underlies the displayed reasoning.

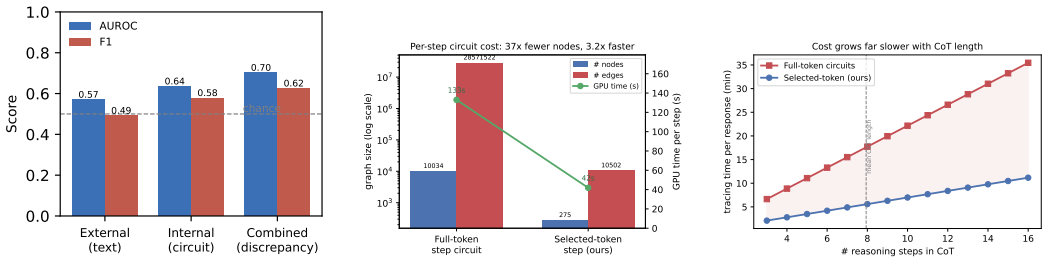


Figure 2: **Effectiveness and efficiency motivation for circuits. Left:** circuit evidence is discriminative. **Middle:** per reasoning-step circuit cost. **Right:** scaling of circuit-tracing cost with CoT length.

2.3 Why Circuit Tracing for CoT Unfaithfulness Detection?

Circuit evidence separates faithful from unfaithful reasoning. We first verify that circuit-based evidence is discriminative for unfaithful detection. On step-level FaithCoT-Bench instances (detailed setup in App. B), we compare three probes under an identical setting (Fig. 2, left): an *external-only* probe (a frozen BERT encoder over the reasoning step), an *internal-only* probe (a GIN over the step's circuit graph), and a *combined* probe that scores the cosine discrepancy between the two representations. The external-only probe performs close to random (AUROC=0.57), whereas the internal circuit probe is clearly higher (0.64), and explicitly modeling the internal-external discrepancy performs best (0.70). Two conclusions follow: the emitted text alone carries little faithfulness signal while the model's internal circuit does; and the discrepancy between internal computation and displayed reasoning is more informative. This directly motivates a circuit-guided, discrepancy-based detector, consistent with our Definition 1 of unfaithfulness as a mismatch with $\mathcal{R}_{\mathcal{M}}(q)$.

Full-token circuits do not scale. However, tracing a *full* attribution graph over every token of a reasoning step is expensive. On Qwen2.5-7B-Instruct, a single-step full-token circuit averages 1.0×10^4 nodes and 2.9×10^7 edges and costs 133s of GPU time (Fig. 2, middle); a typical CoT contains ~ 8 such steps, so exhaustively tracing one response costs on the order of 10^3s. This cost is dominated by the number of traced token positions and therefore grows with CoT length (Fig. 2, right), making token-level tracing impractical at scale. Selecting only a few informative tokens per sentence before tracing (Sec. 3.1) largely alleviates this, keeping long-CoT analysis tractable.

Motivation for our method. Together, these findings reveal the central trade-off: circuit tracing provides effective internal evidence for detecting CoT unfaithfulness, but dense circuit construction is too expensive to scale. We therefore need an *effective and efficient circuit-guided unfaithfulness detector*: one that exploits model-internal circuit evidence, addressing *Challenge 1* raised in Sec.1, while avoiding exhaustive full-position tracing, addressing *Challenge 2*. This motivates the design of CIE-Scorer, introduced in the next section.

3 Method

We introduce the **C**ircuit-guided **I**nternal-**E**xternal Discrepancy Scorer (CIE-Scorer) in two stages. Sec. 3.1 selects informative tokens and constructs a compact circuit for each reasoning sentence. Sec. 3.2 encodes these circuits and the displayed reasoning as internal and external sentence graph, and measures their discrepancy using Fused Gromov–Wasserstein (FGW) distance.

3.1 Token-Selected Sentence-Level Circuit Construction

We use reasoning sentence as the basic unit of analysis, matching how the displayed CoT is organized. Recall from Sec. 2.3 that circuit evidence is discriminative yet tracing every token is costly; we therefore trace only a compact set of informative tokens per sentence. Let the t-th sentence be $r_t = (o_{t,1}, o_{t,2}, \ldots, o_{t,L_t})$, where $o_{t,\mu}$ is the μ-th token and L_t its length. To identify the most informative tokens, we combine two complementary signals: a simple entropy-based filter that reduces the number of candidates while retaining potentially important tokens, and a counterfactual

necessity score that ranks how much each token contributes to the model's decision. We then trace the circuit around the highest-ranked tokens.

Entropy-based candidate selection. We first use token-level entropy as a *lightweight* signal to identify positions likely to participate in active reasoning: high-entropy positions mark where the model is actively deciding, and entropy is read directly from the forward pass at no extra cost. For token $o_{t,\mu}$ with predictive distribution $p_{t,\mu} = \mathrm{Softmax}(z_{t,\mu}/\tau)$ over the vocabulary ($z_{t,\mu} \in \mathbb{R}^V$ the logits, τ the temperature), we compute its entropy

$$H_{t,\mu} = -\sum_{j=1}^{V} p_{t,\mu,j} \log p_{t,\mu,j}. \tag{2}$$

Since low-entropy tokens contribute little to reasoning (Wang et al., 2025b), we keep the top-B highest-entropy positions as a candidate set $\mathcal{C}_t$ and prune the sentence before necessity scoring.

Counterfactual necessity score. Entropy flags where the model is uncertain, but uncertainty alone does not imply causal importance; we therefore rank the candidates by how much each one actually affects the reasoning. For a candidate $o_{t,\mu} \in \mathcal{C}_t$, let $\tilde{r}_{t,\mu}$ be the sentence with that token removed (one additional forward pass per candidate). We measure two effects of this ablation: a *semantic change* of the sentence representation $h(\cdot)$ (the model's mean-pooled final-layer hidden state), and a *predictive change* of the next-token distribution $p(\cdot)$ at the sentence's final position. The first captures how much the sentence's content changes, the second how much the model's subsequent decision changes. We combine them into a necessity score:

$$N_{t,\mu} = \lambda\big[\,1 - \cos\big(h(r_t),\, h(\tilde{r}_{t,\mu})\big)\big] + (1-\lambda)\, D_{\mathrm{KL}}\big(p(\cdot \mid r_t) \,\|\, p(\cdot \mid \tilde{r}_{t,\mu})\big), \tag{3}$$

where $\lambda \in [0,1]$ balances the two effects. We take the K tokens with the highest score as $\mathcal{S}_t \subseteq r_t$.

Selected-token circuit construction. Given the selected positions $\mathcal{S}_t$ within sentence r_t, we trace a compact sentence-level circuit $G_t^{\mathrm{circ}} = \mathrm{Trace}(r_t; \mathcal{S}_t)$. Here, r_t alone provides the activation context, and attribution is restricted to $\mathcal{S}_t$ and anchored at the rightmost selected token. We process each sentence independently, avoiding repeated tracing of the question and an expanding prefix. This sentence-local approximation reduces tracing cost, while dependencies across reasoning sentences are modeled subsequently in the trace-level sentence graph. Under the same sentence-level context, selecting only 6 of the $\sim$55 tokens per sentence reduces the node and edge counts by factors of 37 and 2,700, respectively, and yields a 3.2$\times$ tracing speedup (42 s versus 133 s per step; Fig. 2).

3.2 Circuit-Guided Internal–External Discrepancy Scorer

Motivated by Sec. 2.3, where the external-internal discrepancy emerges as the most informative signal, CIE-Scorer uses this discrepancy to measure CoT unfaithfulness. Since directly comparing sentence-level circuits across an entire trace is computationally expensive, we use a graph neural network (GNN) to compress each sentence-level circuit into a comparable representation. The external and internal views are organized as *trace-level sentence graphs*, with reasoning sentences as nodes and dependencies as edges. Larger discrepancy is treated as stronger evidence of unfaithfulness.

Internal sentence representations. We order the sentence circuits by CoT position into $\mathcal{G}^{\mathrm{circ}} = (G_1^{\mathrm{circ}}, \ldots, G_T^{\mathrm{circ}})$, a step-wise approximation of the model's internal computation along the trace. Since these circuits are sparse, heterogeneous, and variable in size, direct comparison is inefficient; we therefore encode each with a graph neural network into a compact latent vector $x_t^{\mathrm{int}} = f_{\mathrm{GNN}}(G_t^{\mathrm{circ}}) \in \mathbb{R}^d$ and stack them into the internal node-feature matrix $X^{\mathrm{int}} = (x_1^{\mathrm{int}}, \ldots, x_T^{\mathrm{int}})$.

External sentence representations. The displayed reasoning is external text, so we encode each sentence r_t with a frozen BERT sentence encoder, $x_t^{\mathrm{ext}} = g(r_t)$, rather than the model's hidden states, keeping the external view independent of internal computation. A lightweight MLP adaptor P aligns it with the internal space, $\tilde{x}_t^{\mathrm{ext}} = P(x_t^{\mathrm{ext}})$, giving the external node-feature matrix $\tilde{X}^{\mathrm{ext}} = (\tilde{x}_1^{\mathrm{ext}}, \ldots, \tilde{x}_T^{\mathrm{ext}})$.

Shared sentence-graph construction. Given sentence representations $X = (x_1, x_2, \ldots, x_T)$, we construct a trace-level graph by treating X as the node feature matrix. The same construction rule is applied to both adapted external representations $\tilde{X}^{\text{ext}}$ and internal representations X^{int}, yielding $G^{\text{ext}} = (A^{\text{ext}}, \tilde{X}^{\text{ext}})$ and $G^{\text{int}} = (A^{\text{int}}, X^{\text{int}})$. Applying the same construction rule to both views reduces discrepancies attributable solely to graph construction and makes the two graphs directly comparable. The adjacency matrix captures two complementary aspects of a reasoning trace: **local temporal progression** and **long-range semantic dependency**. Each sentence is connected to its immediate successor to preserve the sequential flow of the CoT, while additional forward edges are weighted by the positive semantic similarity between sentence representations:

$$A_{ij} = \begin{cases} \mathbb{I}[j = i+1] + \gamma \max(0, \cos(x_i, x_j)), & i < j, \\ 0, & j \le i, \end{cases} \tag{4}$$

where γ controls the strength of semantic dependency edges. We row-normalize A to stabilize the scale of outgoing dependencies, and set the node measure ν to be uniform over reasoning sentences. This construction provides a unified graph abstraction for both displayed and internal reasoning: node features encode sentence-level reasoning states, while edges encode temporal and semantic dependencies across the trace.

FGW-based unfaithfulness score. We compare the two graphs using **FGW distance**, which jointly captures discrepancies in sentence-level representations and graph structure while allowing soft correspondence between nodes. Unlike node-wise distances, FGW jointly compares node features and relational structure, making it suitable for our internal–external graph comparison. For $G_1 = (A_1, X_1, \nu_1)$ and $G_2 = (A_2, X_2, \nu_2)$, it is defined as

$$\text{FGW}_\alpha(G_1, G_2) = \min_{\pi \in \Pi(\nu_1, \nu_2)} \sum_{i,j,k,l} \Big[\alpha\big(A_1(i,j) - A_2(k,l)\big)^2 + (1-\alpha)\|X_1(i) - X_2(k)\|_2^2\Big]\pi_{ik}\pi_{jl}, \tag{5}$$

where π is the soft node coupling and $\alpha \in [0,1]$ controls the trade-off between structural and feature discrepancy. We use uniform node measures and define the instance-level unfaithfulness score as $s(\mathcal{C}) = \text{FGW}_\alpha(G^{\text{ext}}, G^{\text{int}})$, with larger $s(\mathcal{C})$ indicating stronger internal–external discrepancy.

Training objective. Let $y_n \in \{0, 1\}$ denote the trace-level label of the n-th sample, where $y_n = 1$ indicates an unfaithful trace and $y_n = 0$ indicates a faithful trace. Following our internal–external alignment hypothesis, we encourage faithful traces to have smaller graph discrepancy and unfaithful traces to exceed a margin m. We train the model with a margin-based distance objective:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \left[(1 - y_n)\, s(\mathcal{C}_n) + y_n \max\big(0, m - s(\mathcal{C}_n)\big)\right], \tag{6}$$

where $m > 0$ is a margin specifying the minimum desired discrepancy for unfaithful traces. The first term minimizes FGW distance for faithful samples, encouraging consistency between external and internal graphs, while the second term pushes unfaithful samples to have a discrepancy larger than m. During inference, a trace is predicted as unfaithful when its score exceeds a validation-set threshold. The pseudo-code algorithm of CIE-SCORER is provided in Appendix E

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Dataset. We evaluate our method on `FaithCoT-Bench` proposed by Shen et al. (2025b), a benchmark designed for instance-level CoT unfaithfulness detection. The benchmark consists of CoT trajectories collected from four domain-specific datasets, with human annotations assigned according to predefined unfaithfulness criteria. Specifically, it includes `LogicQA` (Liu et al., 2020), `TruthfulQA` (Lin et al., 2021), `AQuA` (Ling et al., 2017), and the biomedical subset of `HLE` (Phan et al., 2025). Detailed statistics are provided in Appendix C. Because FaithCoT-Bench labels are human annotations of *observable* unfaithfulness, we additionally validate CIE-SCORER *without* relying on such labels using the CAUSAL DIAGNOSTICITY testbed (Zaman & Srivastava, 2025), which supplies controlled faithful/unfaithful explanation pairs. We compare against six families of detectors that capture complementary signals: *simple baselines* (Random, Perplexity) (Cui et al.,

Table 1: F1 (%) comparison of CoT faithfulness detection methods. The best result for each dataset and backbone is highlighted in **bold**. Complete results with both accuracy and F1 are reported in the appendix. "–" indicates that the result is unavailable.

Paradigm	Method	Logic-QA		Truthful-QA		AQuA		HLE-Bio	
		LLaMA	Qwen	LLaMA	Qwen	LLaMA	Qwen	LLaMA	Qwen
Baselines	Random	35.4	37.0	42.7	34.8	37.4	40.7	43.8	46.8
	Perplexity	19.2	40.8	40.5	5.7	36.1	45.2	52.4	55.8
Counterfactual-based	Adding Mistakes	47.9	38.3	60.7	38.5	66.7	59.2	51.6	53.8
	Option Shuffling	52.6	49.1	59.3	34.7	16.7	28.6	14.3	48.3
	Removing Steps	27.6	45.2	50.4	47.5	46.2	22.7	37.0	20.0
	Early Answering	48.6	40.4	52.6	43.2	53.3	28.6	48.3	48.0
	Paraphrasing	47.9	29.8	49.1	41.5	42.9	21.3	40.0	40.1
Logits-based	Answer Tracing	45.9	37.2	50.5	45.5	30.8	36.4	76.2	51.2
	Information Gain	51.2	42.0	40.5	57.8	20.2	39.3	9.5	41.2
LLM-as-judge	BiGGen	59.4	51.2	67.3	59.6	70.3	72.3	69.2	62.9
Attribution-based	CC-SHAP	42.6	56.3	60.1	59.5	62.5	45.2	64.8	63.2
Circuit-based	CRV	–	–	66.7	58.9	62.8	68.7	–	–
	CIE-SCORER	**60.8**	**58.6**	**71.5**	**60.4**	**72.8**	**78.5**	**79.7**	**71.9**

2025), *counterfactual-based methods* (Lanham et al., 2023), *logit-based methods* such as Answer Tracing (Huang et al., 2025) or Information Gain (Li et al., 2024), *LLM-as-Judge methods* using BiGGen (Kim et al., 2025), *attribution-based methods* represented by CC-SHAP (Parcalabescu & Frank, 2024), and *circuit-based methods* represented by CRV (Zhao et al., 2025b).

Implementation. For baselines, we follow the implementation and evaluation protocol of FaithCoT-Bench, using the Gradient Boosting Classifier for the CRV-based baseline. We implement circuit tracing with the `crv-8b-instruct-transcoders` released by CRV, matched to `Llama-3.1-8B-Instruct`. We additionally train a matched Top-K transcoder for `Qwen2.5-7B-Instruct` and rerun the pipeline on it. External representations use a frozen BERT sentence encoder over each reasoning sentence. All runs use NVIDIA A800 80GB GPU. Detailed hyperparameters are provided in Appendix F.

4.2 PERFORMANCE COMPARISON

CIE-SCORER achieves the best F1 across both backbones and all datasets. As shown in Table 1, CIE-SCORER consistently outperforms the strongest baseline. On LLaMA-3.1-8B-Instruct, it improves F1 by 1.4, 4.2, 2.5, and 3.5 points on Logic-QA, Truthful-QA, AQuA, and HLE-Bio, respectively. On Qwen2.5-7B, the corresponding improvements are 2.3, 0.8, 6.2, and 8.7 points. These gains across two backbone models demonstrate the effectiveness and robustness of CIE-SCORER for CoT faithfulness detection. Complete accuracy and F1 results are reported in the Appendix H.

Existing baselines provide useful but incomplete faithfulness signals. Simple, counterfactual-based, and logit-based methods exhibit inconsistent performance across tasks, indicating that surface perturbations and token-level probability changes provide only partial faithfulness signals. CC-SHAP compares the input attribution profiles of predictions and explanations; although it is the strongest baseline on Qwen Logic-QA and HLE-Bio, it does not perform consistently across settings. BiGGen performs strongly but relies primarily on the generated text rather than internal computation. CRV accesses internal circuits but fails on Logic-QA and HLE-Bio because of the cost of dense full-circuit tracing, whereas CIE-SCORER handles all datasets using compressed circuits.

CIE-SCORER is diagnostic of faithfulness under a causal testbed. To test whether CIE-SCORER measures faithfulness itself rather than plausibility, we additionally evaluate it under the CAUSAL DIAGNOSTICITY framework (Zaman & Srivastava, 2025), which builds faithful/unfaithful explanation pairs via in-context knowledge editing and reports *diagnosticity*: the fraction of pairs for which the metric assigns a lower unfaithfulness score to the faithful explanation . Following this

Table 2: Causal diagnosticity across different unfaithfulness categories.

Metric	Fact	Ana.	Obj.	Multi	Avg
Early Answering	**0.756**	0.534	0.566	0.468	0.581
Adding Mistakes	0.534	0.590	0.614	0.542	0.570
Paraphrasing	0.556	0.535	0.425	0.448	0.491
CC-SHAP	0.559	0.318	0.539	0.442	0.465
CIE-SCORER	0.564	**0.667**	**0.692**	**0.624**	**0.637**

protocol on `Qwen2.5-7B-Instruct` with a *frozen* CIE-SCORER (trained on FaithCoT and never tuned on these pairs), Table 2 shows that CIE-SCORER attains the highest average diagnosticity (0.637) and is the best-performing metric on three of the four tasks. Moreover, CIE-SCORER is the only metric that remains clearly above random (0.5) on *all four* tasks (0.564–0.692), whereas existing CoT metrics drop to or below random on at least one task. This indicates that CIE-SCORER tracks causal faithfulness rather than task-specific surface cues, complementing the human-labeled FaithCoT-Bench evaluation.

4.3 ABLATION STUDY

Token selection is critical for identifying internal reasoning signals. When replacing our token selection strategy with single variants, performance consistently drops across all datasets, as shown in Table 3. Compared with the full model, the entropy-only variant leads to an average decrease of 10.0 Acc points and 13.2 F1 points, with particularly clear degradation on HLE-Bio. The counterfactual-only variant yields an average decrease of 6.5 Acc points and 9.0 F1 points. These results indicate that either signal alone is insufficient to reliably preserve the internal evidence relevant to faithfulness detection. **By integrating complementary token-level signals, our token selection reduces circuit-tracing cost while producing more informative internal reasoning graphs.**

Table 3: Ablation study on four datasets using LLaMA-3.1-8B-Instruct. Results are reported in Acc and F1 (%). The best result for each metric under each dataset is highlighted in **bold**.

Variant	Logic-QA		Truthful-QA		AQuA		HLE-Bio	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
CIE-SCORER	**69.0**	**60.8**	**78.0**	**71.5**	**77.0**	**72.8**	**78.0**	**79.7**
Entropy only	58.0	59.8	70.0	57.3	75.0	62.2	59.0	52.9
Counterfactual only	61.0	58.7	68.0	62.5	74.0	70.7	73.0	57.1
w/o sequential edges	63.0	51.8	75.0	66.7	69.0	65.4	75.0	64.2
w/o similarity edges	65.0	53.7	74.0	68.1	73.0	65.0	69.0	70.3
w/o internal GNN	60.0	58.3	67.0	63.1	64.0	65.7	71.0	58.6

Graph structure and internal graph encoding are both necessary for detection. As shown in Table 3, removing sequential or similarity edges consistently degrades performance. Specifically, removing sequential edges causes an average drop of 5.0 Acc points and 9.2 F1 points, while removing similarity edges results in an average decrease of 5.3 Acc points and 6.9 F1 points. This shows that reasoning order and semantic proximity provide complementary information for modeling circuits. Removing the internal GNN further leads to substantial degradation, with an average drop of 10.0 Acc points and 9.8 F1 points. **These results suggest that our method benefits not only from selected circuit features, but also from explicitly modeling their graph structure.**

Case study: token selection concentrates on informative, high-entropy tokens. Beyond the quantitative ablation, we analyze *which* tokens are selected. We measure the following statistics over all 2,699 step-circuits traced in our experiments. On average, each sentence contains 52.4 tokens, of which CIE-SCORER retains only about 6 (i.e., 11.5%); yet the selected positions have a mean predictive entropy that is 1.83 higher than the sentence average, confirming that selection favors high-uncertainty positions. Qualitatively, the selector focuses on decision-bearing tokens such as step markers, key concepts, relations, negations, and terminal punctuation, rather than function words, as illustrated below.

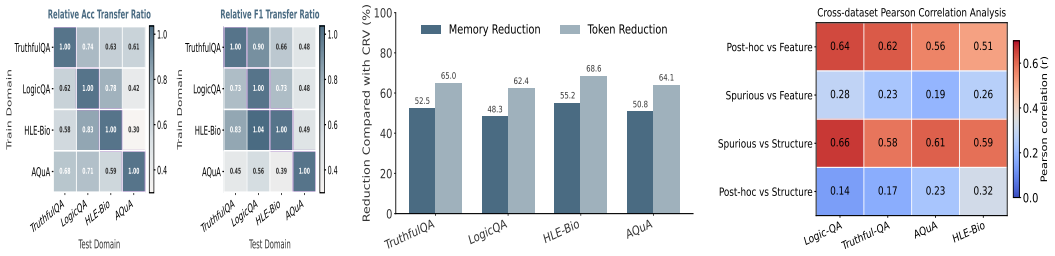
Case study: selected tokens on two reasoning steps

Step: "Step 1: Understanding the definition of a chemical reaction, which involves a change in chemical composition, properties, and characteristics of substances."
Selected: {`Step`, `Understanding`, `definition`, `reaction`, `change`, `and`}

Step: "Step 3: Identifying the option that does not involve a chemical reaction."
Selected: {`Step`, `:`, `ifying`, `option`, `not`, `.`}

4.4 IN-DEPTH ANALYSIS

Different unfaithfulness types exhibit distinct internal–external mismatch patterns. Following the (Shen et al., 2025b) taxonomy, we divide unfaithful CoTs into post-hoc reasoning and spurious reasoning chains. Based on the optimal FGW coupling $\pi^\star$, we decompose the discrepancy score into the feature-level contribution $s_{\text{feat}} = (1-\alpha)\sum_{i,k}\|\widetilde{X}^{\text{ext}}(i) - X^{\text{int}}(k)\|_2^2\pi^\star_{ik}$ and the structure-level



(a) Cross-domain transfer. (b) Efficiency analysis. (c) Cross-dataset Pearson correlation.

Figure 3: Additional analysis of CIE-SCORER. (a) Cross-domain generalization measured by relative transfer ratio. (b) Memory and token reduction compared with CRV. (c) Cross-dataset Pearson correlation between unfaithfulness types and the feature-/structure-level discrepancy components.

contribution $s_{\text{struct}} = \alpha \sum_{i,j,k,l} (A^{\text{ext}}(i,j) - A^{\text{int}}(k,l))^2 \pi^\star_{ik} \pi^\star_{jl}$, and compute Pearson correlations between each type indicator and the two components. As shown in Figure 3c, post-hoc reasoning correlates more strongly with feature-level discrepancy (0.58 vs. 0.22), suggesting that its displayed content may lack support from internal circuit evidence. In contrast, spurious reasoning correlates more strongly with structure-level discrepancy (0.61 vs. 0.24), suggesting disrupted dependencies between otherwise locally coherent steps. Further analyses are provided in Appendices G.2 and G.3.

4.5 CROSS-DOMAIN PERFORMANCE

CIE-SCORER demonstrates meaningful cross-domain transfer across several reasoning domains. To evaluate cross-domain generalization, we train the detector on one dataset and directly test it on the remaining datasets, as shown in Figure 3a. We report the relative transfer ratio (RTR), defined as $\text{RTR}_{i \to j} = \text{Perf}(\text{train} = i, \text{test} = j)/\text{Perf}(\text{train} = j, \text{test} = j)$, where Perf denotes either accuracy or F1. Thus, diagonal entries are normalized to 1.0, and higher off-diagonal values indicate stronger transfer to the target domain. The results show meaningful transfer across several domain pairs. In terms of F1, training on TruthfulQA yields an RTR of 0.90 on Logic-QA, while training on HLE-Bio yields RTRs of 1.04 on Logic-QA and 0.83 on TruthfulQA. In contrast, transfer involving AQuA is weaker, with most off-diagonal F1 RTR values below 0.5, indicating a larger domain gap for mathematical reasoning traces.

4.6 EFFICIENCY ANALYSIS

CIE-SCORER achieves better scalability than CRV by reducing both tracing cost and computational resources. As shown in Figure 3b, CIE-SCORER consistently reduces peak memory usage by 48.3%–55.2% and traced tokens by 62.4%–68.6% across the four datasets. This indicates that the proposed token selection strategy can effectively filter out redundant tracing targets before circuit construction, thereby reducing both the size of extracted circuits and the memory required for attribution computation. Moreover, on the datasets where CRV can successfully run, CIE-SCORER achieves clear runtime reductions of 67.6% on TruthfulQA and 46.7% on AQuA, while also improving F1 by 7.2% and 15.9%, respectively, as shown in Figure 4a. These results suggest that the efficiency gain does not come from simply discarding useful evidence; instead, by focusing circuit tracing on more informative tokens, CIE-SCORER obtains a better efficiency–effectiveness trade-off than CRV.

5 CONCLUSION

In this paper, we propose CIE-SCORER, a circuit-guided framework for instance-level CoT unfaithfulness detection that measures the discrepancy between circuit-derived evidence of internal computation and displayed reasoning trajectories. CIE-SCORER selects informative tokens to construct compact sentence-level circuit graphs, encodes them with a GNN into internal reasoning embeddings, and aligns them with external reasoning graphs via FGW distance to produce an unfaithfulness score. Experiments on four FaithCoT-Bench datasets show that CIE-SCORER achieves state-of-the-art performance while substantially reducing circuit-tracing costs. Further analysis reveals distinct discrepancy patterns across different types of unfaithfulness, highlighting the value of combining internal computational evidence with displayed reasoning for CoT unfaithfulness detection.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting and editing portions of the manuscript for clarity and readability, as well as with formatting the manuscript and supporting materials. We did not use generative AI tools to generate synthetic datasets, develop the conceptual framework, formulate mathematical claims, produce proofs, propose or refine research hypotheses, design the methodology or experiments, implement the proposed method, clean or analyze data, or interpret the experimental results; the remaining required-disclosure tasks are not applicable to this work. All AI-assisted content was carefully reviewed and revised by the authors. The authors independently verified the technical claims, citations, implementation, and experimental results. We take full responsibility for the final content of this work, including any text or artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, the collection or use of personally identifiable or sensitive information, or the release of a new dataset. The experiments are conducted using existing research benchmarks and pretrained language models.

REPRODUCIBILITY STATEMENT

To support reproducibility, we release an anonymous implementation of CIE-SCORER at `https://anonymous.4open.science/r/CIE-Scorer-7E75/`. The repository provides the environment and package configurations, installation instructions, documented command-line interfaces for training and evaluation.

REFERENCES

Chirag Agarwal, Sree Harsha Tanneru, and Himabindu Lakkaraju. Faithfulness vs. plausibility: On the (un)reliability of explanations from large language models. *ArXiv*, abs/2402.04614, 2024. URL `https://api.semanticscholar.org/CorpusID:267523276`.

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025. URL `https://transformer-circuits.pub/2025/attribution-graphs/methods.html`.

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. Chain-of-thought reasoning in the wild is not always faithful. *arXiv preprint arXiv:2503.08679*, 2025.

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

Fazl Barez, Tung-Yu Wu, Iván Arcuschin, Michael Lan, Vincent Wang, Noah Siegel, Nicolas Collignon, Clement Neo, Isabelle Lee, Alasdair Paren, et al. Chain-of-thought is not explainability. *Preprint, alphaXiv*, pp. v1, 2025.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025.

Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, et al. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. *arXiv preprint arXiv:2502.13260*, 2025.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4443–4458, 2020.

Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36:70757–70798, 2023.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, and Ilya Sutskever Jan Leike Jeffrey Wu. Scaling and evaluating sparse autoencoders. 2024.

Yoav Gur-Arieh, Ana Marasović, and Mor Geva. Faithfulness metrics don't measure faithfulness: A meta-evaluation with ground truth. *arXiv preprint arXiv:2605.25052*, 2026.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*, 2024.

Pingjun Hong and Benjamin Roth. Do llm self-explanations help users predict model behavior? evaluating counterfactual simulatability with pragmatic perturbations, 2026. URL https://arxiv.org/abs/2601.03775.

Pingjun Hong, Beiduo Chen, Siyao Peng, Marie-Catherine de Marneffe, and Barbara Plank. LiTEx: A linguistic taxonomy of explanations for understanding within-label variation in natural language inference. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 34065–34085, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1728. URL https://aclanthology.org/2025.emnlp-main.1728/.

Pingjun Hong, Beiduo Chen, Siyao Peng, Marie-Catherine de Marneffe, Benjamin Roth, and Barbara Plank. Agree, disagree, explain: Decomposing human label variation in nli through the lens of explanations, 2026. URL https://arxiv.org/abs/2510.16458.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

Jennifer Hsia, Danish Pruthi, Aarti Singh, and Zachary C Lipton. Goodhart's law applies to nlp's explanation benchmarks. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 1322–1335, 2024.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In *61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*, pp. 1049–1065. Association for Computational Linguistics (ACL), 2023.

Yue Huang, Chujie Gao, Siyuan Wu, Haoran Wang, Xiangqi Wang, Yujun Zhou, Yanbo Wang, Jiayi Ye, Jiawen Shi, Qihui Zhang, et al. On the trustworthiness of generative foundation models: Guideline, assessment, and perspective. *arXiv preprint arXiv:2502.14296*, 2025.

Shima Imani, Liang Du, and Harsh Shrivastava. Mathprompter: Mathematical reasoning using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pp. 37–42, 2023.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4198–4205, 2020.

Xue Jiang, Yihong Dong, Lecheng Wang, Zheng Fang, Qiwei Shang, Ge Li, Zhi Jin, and Wenpin Jiao. Self-planning code generation with large language models. *ACM Transactions on Software Engineering and Methodology*, 33(7):1–30, 2024.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee, Namgyu Ho, Se June Joo, Miyoung Ko, Yoonjoo Lee, Hyungjoo Chae, Jamin Shin, Joel Jang, Seonghyeon Ye, Bill Yuchen Lin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. The biggen bench: A principled benchmark for fine-grained evaluation of language models with language models, 2025. URL https://arxiv.org/abs/2406.05761.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

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

Jiachun Li, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Towards faithful chain-of-thought: Large language models are bridging reasoners. *arXiv preprint arXiv:2405.18915*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*, 2017.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*, 2020.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 305–329, 2023.

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

Katie Matton, Robert Osazuwa Ness, John Guttag, and Emre Kıcıman. Walk the talk? measuring the faithfulness of large language model explanations. *arXiv preprint arXiv:2504.14150*, 2025.

Letitia Parcalabescu and Anette Frank. On measuring faithfulness or self-consistency of natural language explanations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6048–6089, 2024.

Seongheon Park, Xuefeng Du, Min-Hsuan Yeh, Haobo Wang, and Yixuan Li. Steer llm latents for hallucination detection. In *International Conference on Machine Learning*, pp. 47971–47990. PMLR, 2025.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. *arXiv preprint arXiv:2402.13950*, 2024.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. *arXiv preprint arXiv:2406.05946*, 2024.

Benjamin Roth, Nicholas Edwards, Pingjun Hong, Loris Schoenegger, and Sebastian Schuster. From models to systems: A survey of explainability for tool-augmented language models and ai agents. Discussion paper, University of Vienna, January 2026. URL http://eprints.cs.univie.ac.at/8619/.

Xu Shen, Yixin Liu, Yiwei Dai, Yili Wang, Rui Miao, Yue Tan, Shirui Pan, and Xin Wang. Understanding the information propagation effects of communication topologies in llm-based multi-agent systems. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12358–12372, 2025a.

Xu Shen, Song Wang, Zhen Tan, Laura Yao, Xinyu Zhao, Kaidi Xu, Xin Wang, and Tianlong Chen. Faithcot-bench: Benchmarking instance-level faithfulness of chain-of-thought reasoning. *arXiv preprint arXiv:2510.04040*, 2025b.

Nicholas Sofroniew, Isaac Kauvar, William Saunders, Runjin Chen, Tom Henighan, Sasha Hydrie, Craig Citro, Adam Pearce, Julius Tarng, Wes Gurnee, et al. Emotion concepts and their function in a large language model. *arXiv preprint arXiv:2604.07729*, 2026.

Shriyank Somvanshi, Md Monzurul Islam, Amir Rafe, Anannya Ghosh Tusti, Arka Chakraborty, Anika Baitullah, Tausif Islam Chowdhury, Nawaf Alnawmasi, Anandi Dutta, and Subasish Das. Bridging the black box: a survey on mechanistic interpretability in ai. *ACM Computing Surveys*, 58(8):1–35, 2026.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting, 2023. URL https://arxiv.org/abs/2305.04388.

Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, Miao Yu, Shiqian Zhao, Chenlong Yin, Jinhu Fu, Yibo Yan, Hanjun Luo, et al. A comprehensive survey in llm (-agent) full stack safety: Data, training and deployment. *arXiv preprint arXiv:2504.15585*, 2025a.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025b.

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

Xumeng Wen, Zihan Liu, Shun Zheng, Zhijian Xu, Shengyu Ye, Zhirong Wu, Xiao Liang, Yang Wang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

Adam Wojciechowski, Mateusz Lango, and Ondřej Dušek. Faithful and plausible natural language explanations for image classification: A pipeline approach. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2340–2351, 2024.

Zidi Xiong, Shan Chen, Zhenting Qi, and Himabindu Lakkaraju. Measuring the faithfulness of thinking drafts in large reasoning models, 2025. URL https://arxiv.org/abs/2505.13774.

Chuanpeng Yang, Yao Zhu, Wang Lu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. Survey on knowledge distillation for large language models: methods, evaluation, and application. *ACM Transactions on Intelligent Systems and Technology*, 2024.

Sohee Yang, Sang-Woo Lee, Nora Kassner, Daniela Gottesman, Sebastian Riedel, and Mor Geva. How well can reasoning models identify and recover from unhelpful thoughts?, 2025. URL https://arxiv.org/abs/2506.10979.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Evelyn Yee, Alice Li, Chenyu Tang, Yeon Ho Jung, Ramamohan Paturi, and Leon Bergen. Dissociation of faithful and unfaithful reasoning in llms. *arXiv preprint arXiv:2405.15092*, 2024.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

Kerem Zaman and Shashank Srivastava. A causal lens for evaluating faithfulness metrics. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 29413–29437, 2025.

Chengshuai Zhao, Zhen Tan, Pingchuan Ma, Dawei Li, Bohan Jiang, Yancheng Wang, Yingzhen Yang, and Huan Liu. Is chain-of-thought reasoning of llms a mirage? a data distribution lens. *arXiv preprint arXiv:2508.01191*, 2025a.

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

Zheng Zhao, Yeskendir Koishekenov, Xianjun Yang, Naila Murray, and Nicola Cancedda. Verifying chain-of-thought reasoning via its computational graph. *arXiv preprint arXiv:2510.09312*, 2025b.

Zirui Zhao, Wee Sun Lee, and David Hsu. Large language models as commonsense knowledge for large-scale task planning. *Advances in neural information processing systems*, 36:31967–31987, 2023.

A Related Work

A.1 Understanding Chain-of-Thought Reasoning

Chain-of-thought (CoT) reasoning has become a widely adopted mechanism for enhancing the reasoning ability of large language models (LLMs) (Wei et al., 2022; Kojima et al., 2022; Yao et al., 2023; Besta et al., 2024; Hong & Roth, 2026; Roth et al., 2026; Hong et al., 2025; 2026). By eliciting intermediate steps before the final answer, CoT improves performance on complex tasks such as mathematics, programming, logic, commonsense, and scientific reasoning (Imani et al., 2023; Jiang et al., 2024; Huang & Chang, 2023; Zhao et al., 2023). Beyond its original use as a prompting strategy, CoT traces have increasingly been used as supervision signals for reasoning-oriented training, reinforcement learning, distillation, and compression, enabling models to acquire or transfer multi-step reasoning abilities more effectively (Yeo et al., 2025; Yang et al., 2024; Chen et al., 2025; Shen et al., 2025a; Hou et al., 2025). Thus, CoT has evolved from a prompting technique into a central paradigm for both improving reasoning performance and exposing intermediate model behavior. However, generated reasoning traces may not faithfully reflect the model's actual decision process, motivating the need to examine CoT as an object of faithfulness analysis.

A.2 Faithfulness Evaluation for Chain-of-Thought

CoT traces are often treated as natural-language explanations of model decisions, but such apparent transparency can be misleading (Feng et al., 2023; Zhao et al., 2025a; Barez et al., 2025; Matton et al., 2025), and faithfulness should be distinguished from plausibility: an explanation may appear convincing to humans without reflecting the computation that supports the prediction (Jacovi & Goldberg, 2020; Wojciechowski et al., 2024). Existing evaluation approaches include interventions on reasoning traces (Lanham et al., 2023; Turpin et al., 2023; Yang et al., 2025; Xiong et al., 2025), sufficiency and comprehensiveness metrics (DeYoung et al., 2020), faithfulness tests for natural-language explanations (Atanasova et al., 2023), self-consistency analysis (Madsen et al., 2024), attribution-based comparison such as CC-SHAP (Parcalabescu & Frank, 2024), and LLM-based judgments of reasoning quality (Arcuschin et al., 2025; Wen et al., 2025). However, these methods may capture prediction dependence, textual consistency, or plausibility rather than internal computational faithfulness. This distinction also creates a construct-validity challenge: explanation metrics may be optimized without genuinely improving faithfulness (Hsia et al., 2024), while recent causal and ground-truth meta-evaluations show that many existing metrics do not reliably distinguish faithful from unfaithful explanations (Zaman & Srivastava, 2025; Gur-Arieh et al., 2026). CIE-SCORER therefore compares the emitted CoT representation with a circuit-attribution-based representation of the computation associated with its reasoning steps. We treat circuit evidence as an interpretable white-box proxy rather than ground-truth access to the model's reasoning.

A.3 Mechanistic Interpretability

Mechanistic interpretability analyzes the internal computations contributing to model predictions (Zhao et al., 2024; Du et al., 2019). Sparse feature representations and circuit tracing provide attribution graphs describing feature interactions and their contributions to model outputs. Recent work has further studied higher-level concepts, including emotion-related concepts and their functional roles in language models (Sofroniew et al., 2026), while broader surveys organize mechanistic interpretability methods through behavioral, counterfactual, and causal perspectives (Somvanshi et al., 2026). Such evidence is relevant to CoT faithfulness because it enables comparison between the displayed reasoning trace and the computation associated with its generation (Ameisen et al., 2025; Elhage et al., 2021). However, dense circuit tracing over long CoTs is computationally expensive (Zhao et al., 2025b). CIE-SCORER addresses this limitation by selecting informative tokens, constructing compact sentence-level circuits, and organizing them into a trace-level representation.

B MOTIVATION-PROBE AND EFFICIENCY-BENCHMARK DETAILS

B.1 THREE-PROBE SETUP

Data. We use FaithCoT-Bench responses generated by QWEN2.5-7B-INSTRUCT on the two datasets that provide step-level unfaithfulness annotations, TRUTHFULQA and LOGIQA. We form a step-level classification set: *unfaithful* steps are the indices listed in each unfaithful response's `unfaithful_step` field, and *faithful* steps are drawn from fully faithful responses (`unfaithfulness`=0), capped at 2 steps per response to limit class imbalance. This yields 174 unfaithful and 266 faithful steps (440 in total).

Split. We use a step-level stratified train/test split (each (response, step) pair is one sample) with test fraction 0.3 and random seed 0, giving 308 training and 132 test steps, preserving the label ratio in both folds.

Probes. The *internal-only* probe converts each step's compact circuit (Sec. 3.1) to a graph and encodes it with a two-layer GIN (hidden size 128, mean pooling) followed by a two-layer MLP. The *external-only* probe encodes the step text with a frozen BERT sentence encoder (mean-pooled over non-padding tokens, max length 256), followed by a two-layer MLP. The *combined* probe L2-normalizes linear projections of the internal and external embeddings into a shared 128-d space and scores their cosine similarity; a single-parameter head maps the similarity to $P(\text{unfaithful})$ so that higher similarity indicates a more faithful step.

Training and metrics. All heads are trained for 60 epochs with Adam (learning rate 10^{-3}, weight decay 10^{-4}), batch size 16, and binary cross-entropy. We report AUROC and F1; the F1 threshold is fixed at the median predicted score to keep both classes represented.

B.2 EFFICIENCY BENCHMARK

Full vs. compact cost. We sample 20 reasoning-step prompts (seed 0) from the traced set. For each, we build a full-token circuit (`max_feature_nodes`= 8192, all token positions) and our compact circuit (6 selected tokens, `max_feature_nodes`= 128), recording node/edge counts and per-step wall-clock time. Timing is GPU-synchronized on a single NVIDIA A800-80GB with no other jobs; one warm-up step is excluded, and we report means over the sampled steps.

Scaling. To evaluate response-level scaling, we cumulatively trace the first k reasoning steps of each CoT, for $k = 1, \ldots, T$, where T is the total number of reasoning steps. We compare full-token circuit construction, which traces all token positions within each included step, with our selected-token construction. For each k, we report the total GPU tracing time accumulated over the first k steps on the same hardware. Tracing is terminated if a configuration exceeds the available GPU memory. This setup measures how the end-to-end circuit-construction cost grows with the number of reasoning steps in a CoT, matching the horizontal axis of Figure 2.

C DATASETS

We use FaithCoT-Bench (Shen et al., 2025b), a benchmark for instance-level CoT faithfulness evaluation. The benchmark is constructed from four source datasets spanning different reasoning domains:

- `LogicQA` (Liu et al., 2020) originates from the Chinese National Civil Servants Examination, later translated into English, and is specifically designed to test *logical deductive reasoning*. Each question is paired with a short passage and multiple candidate answers, requiring the model to select the option that logically follows. Unlike general reading comprehension tasks, `LogicQA` emphasizes multi-step deduction and resistance to spurious correlations, making it a strong testbed for evaluating the causal structure of reasoning traces.
- `TruthfulQA` (Lin et al., 2021) consists of factual questions drawn from diverse domains, deliberately crafted to elicit common misconceptions or human-like falsehoods. Presented in multiple-choice format, the benchmark requires models to reject plausible-sounding but false

answers and instead provide factually correct responses. This adversarial setup directly tests whether reasoning faithfully distinguishes truth from widely held but misleading beliefs.

- `AQuA` (Ling et al., 2017) evaluates *step-by-step numerical reasoning* through algebraic word problems. Each instance requires decomposing the problem into intermediate reasoning steps that lead to a numerical solution. Because arithmetic reasoning depends on tightly coupled causal chains, this dataset provides a natural setting for examining whether CoTs maintain faithful intermediate justifications rather than skipping or fabricating steps.
- The biomedical portion of `HLE` (Phan et al., 2025), referred to as `HLE-Bio`, represents *knowledge-intensive biomedical reasoning*. Questions are drawn from technical biomedical texts and require the integration of domain-specific knowledge with logical inference. Compared to the other benchmarks, `HLE-Bio` poses heightened challenges due to its specialized terminology and dense factual grounding, offering a critical stress test for the faithfulness of reasoning traces.

Together, these source datasets allow FaithCoT-Bench to cover logical, factual, mathematical, and biomedical reasoning, making it suitable for evaluating CoT faithfulness across diverse task settings.

Construct-validity testbed (no human faithfulness labels). FaithCoT-Bench provides human annotations of *observable* unfaithfulness (post-hoc and spurious reasoning). To additionally verify that CIE-SCORER measures faithfulness itself rather than plausibility, we evaluate it under the CAUSAL DIAGNOSTICITY framework (Zaman & Srivastava, 2025), which does *not* use human faithfulness labels. Instead, for each of four controlled tasks: fact checking, analogy, object counting, and multi-hop reasoning, which edits the model with in-context editing (ICE) and generates a *faithful* and an *unfaithful* synthetic explanation for the edited behavior, yielding ground-truth explanation pairs. A metric is scored by *diagnosticity*: the fraction of pairs in which it assigns a lower unfaithfulness score to the faithful explanation (chance $= 0.5$). We follow the released protocol on `Qwen2.5-7B-Instruct` (following the official reference); the CIE-SCORER detector is trained on FaithCoT and kept *frozen*, never tuned on these pairs, so the evaluation is fully held out.

D BASELINES

To comprehensively evaluate CoT unfaithfulness detection, we compare CIE-SCORER with representative baselines from five categories: counterfactual-based methods, logit-based methods, LLM-as-Judge methods, attribution-based methods, and circuit-based methods.

① **Counterfactual-based methods.** Counterfactual-based methods assess CoT faithfulness by perturbing the generated reasoning trajectory and measuring whether the final answer changes. Given an original CoT $\mathcal{C}$ and its answer a, a counterfactual operator Δ produces a perturbed trajectory $\mathcal{C}' = \Delta(\mathcal{C})$, from which the model generates a new answer $a' = \mathrm{M}_{\mathrm{Ans}}(q, \mathcal{C}')$. If $a' \neq a$, the perturbed reasoning content is considered causally influential to the prediction. Conversely, if the answer remains unchanged after substantial perturbation, the modified steps may not be part of the causal reasoning path, suggesting potential unfaithfulness.

We evaluate the following representative perturbation strategies:

- *Adding Mistakes*: Inserts misleading or incorrect intermediate reasoning steps and tests whether the final answer is affected.
- *Option Shuffling*: Randomizes the order of multiple-choice options to examine whether the reasoning and prediction remain stable under changed option contexts.
- *Removing Steps*: Deletes selected reasoning steps to evaluate whether the removed content is necessary for deriving the answer.
- *Early Answering*: Forces the model to answer before completing the full reasoning trajectory, testing whether later steps are causally needed.
- *Paraphrasing*: Rewrites reasoning steps while preserving or slightly altering their semantics, measuring whether the model's prediction is robust to surface-level or semantic variations.

These methods are widely used to probe whether a CoT causally contributes to the final prediction (Lanham et al., 2023; Yang et al., 2025; Xiong et al., 2025; Paul et al., 2024; Yee et al., 2024).

② **Logit-based methods.** Logit-based methods analyze the model's probability dynamics during reasoning. The key assumption is that a faithful reasoning trace should provide evidence that gradually supports the final answer. At each generation step, the model outputs a logit vector $\mathbf{z}_i$ over the vocabulary, which is converted into a token probability by

$$p(c_i \mid \cdot) = \mathrm{softmax}(\mathbf{z}_i)[c_i] = \frac{\exp(z_{i,c_i})}{\sum_k \exp(z_{i,k})}. \tag{7}$$

Answer Tracing. Answer Tracing measures how the probability of the final answer changes as more reasoning tokens are generated. Let a denote the final answer token and $C_{1:i} = (c_1, \ldots, c_i)$ denote the prefix of the reasoning trace. The step-wise confidence change is computed as

$$\Delta_i = p(a \mid C_{1:i}) - p(a \mid C_{1:i-1}), \quad i = 1, \ldots, T. \tag{8}$$

A faithful CoT is expected to provide increasing support for the final answer as reasoning unfolds, whereas inconsistent or weakly supportive probability trends may indicate unfaithfulness.

Information Gain. Information Gain measures how much the question reduces uncertainty over the generated reasoning trace. Given a question Q and a CoT $C = (c_1, \ldots, c_n)$, it is defined as

$$\begin{aligned} IG(C, Q) &= H(C) - H(C \mid Q) \\ &= -\sum_{i=1}^{n} p(c_i \mid C_{1:i-1}) \log p(c_i \mid C_{1:i-1}) + \sum_{i=1}^{n} p(c_i \mid C_{1:i-1}, Q) \log p(c_i \mid C_{1:i-1}, Q). \end{aligned} \tag{9}$$

A higher information gain suggests that the question provides stronger guidance for generating the reasoning trace, which is treated as a proxy signal for faithfulness. Overall, logit-based methods use internal probability signals, but they do not explicitly model the circuit-level computational structure behind the prediction.

③ **LLM-as-Judge methods.** LLM-as-Judge methods use a stronger language model to evaluate whether a generated CoT is faithful. Following the fine-grained rubric-based evaluation protocol of BiGGen (Kim et al., 2025), we provide the judge model with the question, generated CoT, final answer, and detailed scoring criteria, and ask it to produce a faithfulness judgment. Compared with direct prompting, this rubric-guided setting encourages the judge to assess the reasoning trace in a more structured manner, considering aspects such as logical consistency, support for the final answer, and potential post-hoc rationalization. Although such methods can capture fine-grained textual patterns, they still mainly rely on the external content of generated rationales and may conflate reasoning correctness, plausibility, and faithfulness.

④ **Attribution-based method.** CC-SHAP measures the self-consistency between a model's answer and its explanation by comparing their input-attribution distributions, ϕ^{ans} and ϕ^{expl}. Using cosine distance as the divergence measure, it is defined as

$$\text{CC-SHAP} = 1 - D_{\cos}\big(\phi^{\mathrm{ans}}, \phi^{\mathrm{expl}}\big) = \cos\big(\phi^{\mathrm{ans}}, \phi^{\mathrm{expl}}\big), \tag{10}$$

where $D_{\cos}(x, y) = 1 - \cos(x, y)$. A higher value indicates greater agreement between the attribution profiles of the answer and explanation, and thus stronger self-consistency, whereas a lower value indicates a larger attribution mismatch and potential post-hoc rationalization.

⑤ **Circuit-based method.** We also compare with CRV (Zhao et al., 2025b), a circuit-based method that uses mechanistic interpretability signals for reasoning analysis. For each CoT instance or reasoning step, CRV constructs a circuit graph to represent the model's internal computation. The graph is then converted into a fixed-dimensional feature vector using a set of handcrafted graph statistics, which is finally fed into a classifier F for prediction:

$$\mathbf{x}_{\mathrm{CRV}} = \phi(\mathcal{G}_{\mathrm{circuit}}), \quad \hat{y} = F(\mathbf{x}_{\mathrm{CRV}}), \tag{11}$$

where $\mathcal{G}_{\mathrm{circuit}}$ denotes the constructed circuit graph, $\phi(\cdot)$ denotes the graph feature extraction function, and F is the downstream classifier.

CRV extracts handcrafted features from each circuit graph and feeds them into a classifier. These features summarize the graph from three aspects: global statistics such as active node count and output uncertainty, node-level activation and influence statistics, and structural properties such as density, connectivity, and centrality. Although these features provide useful circuit-level signals, CRV mainly treats each circuit as an individual graph with summary statistics. In contrast, CIE-SCORER models the discrepancy between internal computational trajectories and external reasoning trajectories, while using token selection to reduce circuit-tracing cost.

Algorithm 1: Training Procedure of CIE-SCORER

Input: Training set $\mathcal{D} = \{(\mathcal{C}_n, y_n)\}_{n=1}^{N}$, where $\mathcal{C}_n = (r_{n,1}, \ldots, r_{n,T_n})$; tracing budget K; hyperparameters B, λ, α, m.
Output: Trained graph encoder f_{GNN} and adaptor P.

Initialize f_{GNN} and adaptor P;
foreach *epoch* **do**
  foreach *mini-batch* $\mathcal{B} \subset \mathcal{D}$ **do**
    $\mathcal{L}_{\mathcal{B}} \leftarrow 0$;
    foreach $(\mathcal{C}, y) \in \mathcal{B}$ **do**
      foreach *reasoning sentence* $r_t \in \mathcal{C}$ **do**
        Select informative tokens $\mathcal{S}_t = \mathrm{TokenSelect}(r_t; B, \lambda, K)$;
        Construct compressed circuit graph $G_t^{\mathrm{circ}} = \mathrm{Trace}(r_t; \mathcal{S}_t)$;
        Encode internal representation $x_t^{\mathrm{int}} = f_{\mathrm{GNN}}(G_t^{\mathrm{circ}})$;
        Extract external representation $x_t^{\mathrm{ext}} = \mathrm{BERT}(r_t)$;
      Construct internal sentence graph $G^{\mathrm{int}} = \mathrm{BuildGraph}\left((x_t^{\mathrm{int}})_{t=1}^{T}\right)$;
      Construct external sentence graph $G^{\mathrm{ext}} = \mathrm{BuildGraph}\left((P(x_t^{\mathrm{ext}}))_{t=1}^{T}\right)$;
      Compute discrepancy score $s(\mathcal{C}) = \mathrm{FGW}_{\alpha}(G^{\mathrm{ext}}, G^{\mathrm{int}})$;
      Compute training loss $\mathcal{L} = (1-y)s(\mathcal{C}) + y\max(0, m - s(\mathcal{C}))$;
      $\mathcal{L}_{\mathcal{B}} \leftarrow \mathcal{L}_{\mathcal{B}} + \mathcal{L}$;
    Update f_{GNN} and P by minimizing $\mathcal{L}_{\mathcal{B}}/|\mathcal{B}|$;
return f_{GNN}, P;

Table 4: Hyperparameter configuration on four datasets.

Dataset	λ	α	GNN layers	GNN dim	Batch size	Learning rate
Logic-QA	0.5	0.4	2	256	8	1e-4
Truthful-QA	0.5	0.5	2	256	10	1e-4
AQuA	0.5	0.5	2	256	10	1e-4
HLE-Bio	0.5	0.5	1	128	10	5e-5

E PSEUDO-CODE FOR CIE-SCORER

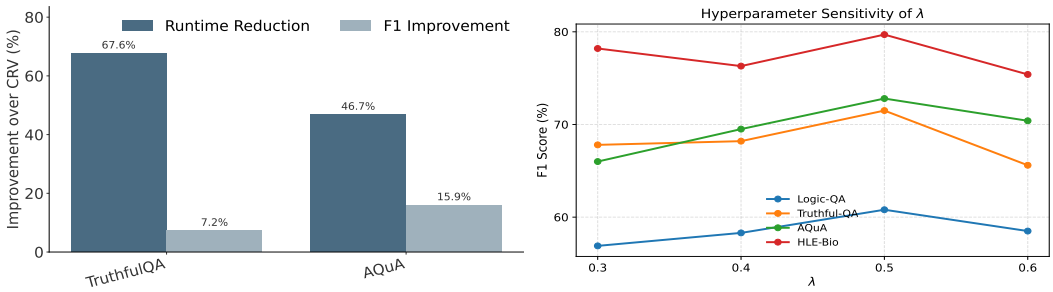
Algorithm 1 summarizes the training procedure of CIE-SCORER. Given a CoT trace, CIE-SCORER first constructs compact sentence-level circuits by selecting informative tokens and tracing their internal attribution structures. It then encodes these circuits into internal sentence representations, extracts external sentence representations using BERT, and builds two trace-level sentence graphs under the same graph construction rule. Finally, the model is trained by minimizing a margin-based objective over the FGW distance between the internal and external graphs, encouraging faithful traces to have small discrepancies and unfaithful traces to have large discrepancies.

F HYPERPARAMETERS

We report the main hyperparameter settings used in our experiments in Table 4. These hyperparameters are selected on the validation set and kept fixed during testing for each dataset.

F.1 CROSS-BACKBONE VALIDATION ON QWEN2.5-7B-INSTRUCT

To test whether CIE-SCORER depends on a specific backbone, we replicate the full pipeline on `Qwen2.5-7B-Instruct`. As no public Top-K transcoder is released for this model, we train a matched set ourselves following the CRV recipe: one Top-K transcoder per MLP layer (hookpoints `layers.*.mlp`, 28 layers) with TopK activation ($k = 128$), trained on a RedPajama-style web corpus and exported to the circuit-tracer format. Using this transcoder, we build sentence-level



(a) Efficiency-F1 (b) λ Sensitivity

Figure 4: (a) CIE-SCORER reduces runtime while improving F1 on the datasets where CRV can be successfully executed. (b) Sensitivity analysis of the token-selection hyperparameters λ.

circuits for the `Qwen2.5-7B-Instruct` CoTs on all four FaithCoT-Bench datasets, extract external sentence representations using a frozen BERT, and train/evaluate the FGW discrepancy detector with the same 2-layer GIN and two-layer MLP adaptor (dimension 256) as in the main experiments; all other hyperparameters follow Table 4. Crucially, the base model, the transcoder, and the trained detector are all matched to `Qwen2.5-7B-Instruct`, with no components shared from the Llama backbone.

G ADDITIONAL EXPERIMENTAL RESULTS

G.1 HYPERPARAMETER ANALYSIS

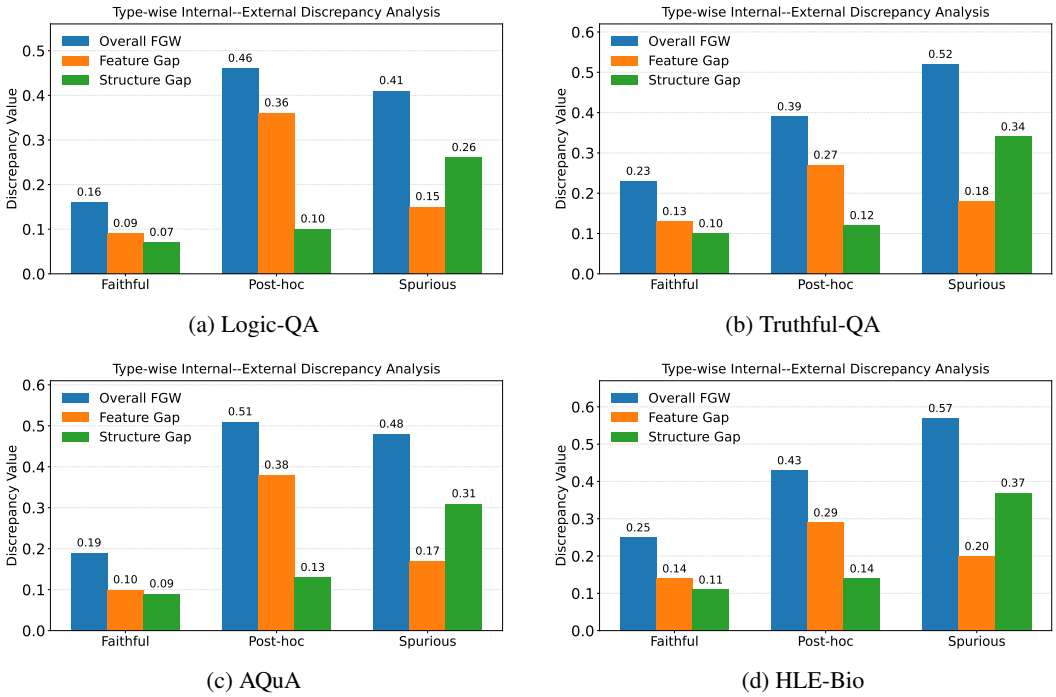
CIE-SCORER benefits from balanced token-selection signals. We analyze the sensitivity of the token-selection coefficient λ, which balances the local semantic change and downstream predictive change in the counterfactual necessity score. As shown in Figure 4b, $\lambda = 0.5$, the setting used in our main experiments, achieves the highest F1 score on all four datasets. Performance generally decreases as λ moves away from 0.5, indicating that overemphasizing either semantic change or predictive change leads to less effective token selection.

G.2 IN-DEPTH ANALYSIS

Different unfaithfulness types exhibit distinct discrepancy patterns across datasets. To further validate the motivation of CIE-SCORER, we conduct a type-wise analysis of the internal–external discrepancy. Following the benchmark taxonomy, we divide CoTs into three categories: faithful CoTs, post-hoc reasoning, and spurious reasoning chains. For each category, we report the overall FGW discrepancy together with its feature-level and structure-level components.

As shown in Figure 5, faithful CoTs consistently exhibit the lowest overall discrepancy across all four datasets, indicating closer alignment between the displayed reasoning trace and the internal circuit-derived computation. For post-hoc reasoning, the feature-level discrepancy is consistently larger than the structure-level discrepancy: the former ranges from 0.27 to 0.38, whereas the latter ranges from 0.10 to 0.14. For example, the feature-level and structure-level discrepancies are 0.36 and 0.10, respectively, on Logic-QA, and 0.27 and 0.12, respectively, on Truthful-QA. This pattern suggests that post-hoc CoTs primarily exhibit a mismatch in reasoning content: their displayed statements are not adequately supported by the corresponding internal circuit evidence.

In contrast, spurious reasoning chains exhibit the opposite pattern, with structure-level discrepancy consistently exceeding feature-level discrepancy. Specifically, the structure-level discrepancy ranges from 0.26 to 0.37, compared with 0.15 to 0.20 for the feature-level discrepancy. For example, the structure-level and feature-level discrepancies are 0.31 and 0.17, respectively, on AQuA, and 0.37 and 0.20, respectively, on HLE-Bio. This pattern suggests that spurious chains primarily disrupt dependencies between reasoning steps, even when individual steps remain locally coherent. Overall, these results indicate that post-hoc and spurious reasoning are associated with distinct forms of internal–external mismatch.



(a) Logic-QA (b) Truthful-QA

(c) AQuA (d) HLE-Bio

Figure 5: Type-wise internal–external discrepancy analysis across four datasets. Each subfigure reports the overall FGW discrepancy and its feature-level and structure-level components for faithful CoTs, post-hoc reasoning, and spurious reasoning chains.

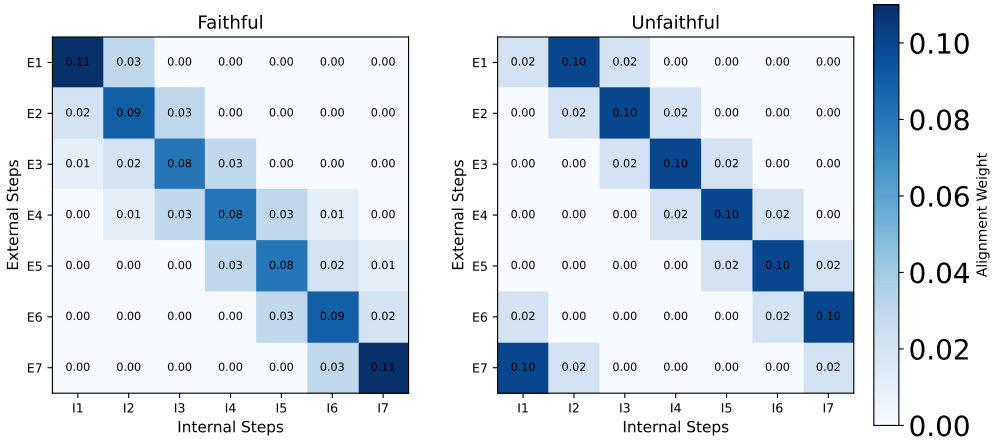


Figure 6: Case study of FGW coupling matrices. The faithful example shows near-diagonal alignment between external and internal steps, while the unfaithful example exhibits more off-diagonal alignment.

G.3 Case study of alignment patterns

We visualize the FGW coupling matrix π for representative faithful and unfaithful CoTs. This visualization helps illustrate how the external reasoning steps are matched to the internal circuit-derived steps. As shown in Figure 6, the faithful example exhibits a near-diagonal alignment pattern, indicating that each displayed reasoning step is mainly supported by its corresponding internal computation step. In contrast, the unfaithful example shows a more shifted and diffuse alignment pattern, suggesting weaker and less consistent correspondence between the displayed reasoning

Table 5: Comparison of CoT faithfulness detection methods with LLaMA-3.1-8B-Instruct and Qwen2.5-7B on four datasets. Results are reported in Acc and F1 (%). The best available result for each metric under each dataset and backbone is highlighted in **bold**. "–" indicates that the result is unavailable.

Paradigm	Method	Logic-QA		Truthful-QA		AQuA		HLE-Bio	
		Acc↑	F1↑	Acc↑	F1↑	Acc↑	F1↑	Acc↑	F1↑
LLaMA-3.1-8B-Instruct									
Baselines	Random	42.0	35.4	43.3	42.7	43.2	37.4	35.7	43.8
	Perplexity	52.3	19.2	44.4	40.5	49.4	36.1	47.4	52.4
Counterfactual-based	Adding Mistakes	57.5	47.9	51.1	60.7	71.6	66.7	46.4	51.6
	Option Shuffling	48.3	52.6	51.1	59.3	61.0	16.7	40.0	14.3
	Removing Steps	51.7	27.6	36.7	50.4	72.4	46.2	39.3	37.0
	Early Answering	35.2	48.6	40.0	52.6	72.4	53.3	46.4	48.3
	Paraphrasing	42.5	47.9	67.8	49.1	68.4	42.9	45.7	40.0
Logits-based	Answer Tracing	33.0	45.9	38.9	50.5	53.2	30.8	64.3	76.2
	Information Gain	51.7	51.2	47.8	40.5	41.6	20.2	52.5	9.5
LLM-as-judge	BiGGen	52.9	59.4	61.0	67.3	73.7	70.3	70.2	69.2
Attribution-based	CC-SHAP	57.0	42.6	64.0	60.1	72.0	62.5	69.0	64.8
Circuit-based	CRV	–	–	68.0	66.7	63.0	62.8	–	–
	CIE-SCORER	**69.0**	**60.8**	**78.0**	**71.5**	**77.0**	**72.8**	**78.0**	**79.7**
Qwen2.5-7B									
Baselines	Random	46.9	37.0	38.1	34.8	45.5	40.7	37.5	46.8
	Perplexity	39.6	40.8	66.0	5.7	47.7	45.2	50.0	55.8
Counterfactual-based	Adding Mistakes	**69.8**	38.3	67.0	38.5	60.9	59.2	65.7	53.8
	Option Shuffling	40.2	49.1	33.3	34.7	54.5	28.6	62.5	48.3
	Removing Steps	34.4	45.2	36.1	47.5	60.9	22.7	54.3	20.0
	Early Answering	29.2	40.4	35.1	43.2	59.8	28.6	62.9	48.0
	Paraphrasing	65.6	29.8	68.0	41.5	57.5	21.3	57.1	40.1
Logits-based	Answer Tracing	26.0	37.2	38.1	45.5	43.7	36.4	40.0	51.2
	Information Gain	52.1	42.0	44.3	57.8	58.0	39.3	50.0	41.2
LLM-as-judge	BiGGen	58.3	51.2	58.0	59.6	73.9	72.3	67.2	62.9
Attribution-based	CC-SHAP	69.0	56.3	65.7	59.5	61.4	45.2	61.1	63.2
Circuit-based	CRV	–	–	54.0	58.9	68.8	68.7	–	–
	CIE-SCORER	61.0	**58.6**	**79.0**	**60.4**	**82.0**	**78.5**	**82.9**	**71.9**

trajectory and the internal computation process. These examples provide an intuitive view of why internal–external discrepancy is informative for unfaithfulness detection.

H FULL RESULTS

Table 5 reports the complete per-dataset accuracy and F1 for all methods on two backbones, `LLaMA-3.1-8B-Instruct` and `Qwen2.5-7B`. CIE-SCORER achieves the best performance on `LLaMA-3.1-8B-Instruct` across all datasets and metrics, and on `Qwen2.5-7B` it obtains the best F1 on all four datasets (and the best accuracy on three of four).

I LIMITATIONS

While CIE-SCORER achieves promising results, it requires access to model internals and is therefore mainly applicable to white-box or open-source LLMs. In this work, we focus on textual CoT traces from FaithCoT-Bench, leaving broader evaluation on more diverse domains, model families, and multimodal reasoning settings for future work.