MECHANISTIC EVIDENCE FOR FAITHFULNESS DECAY IN CHAIN-OF-THOUGHT REASONING

Donald Ye*
Fordham University
Algoverse AI Research
donaldye827@gmail.com

Max Loffgren*
Rice University
Algoverse AI Research
ml215@rice.edu

Om Kotadia
UC San Diego
Algoverse AI Research
bkotadia@ucsd.edu

Linus Wong
Santa Clara University
Algoverse AI Research
lewong@scu.edu

Jonas Rohweder
Algoverse AI Research
jonasrohweder@algoverseairesearch.org

ABSTRACT

Chain-of-Thought (CoT) explanations are widely used to interpret how language models solve complex problems, yet it remains unclear whether these step-by-step explanations reflect how the model actually reaches its answer, or merely post-hoc justifications. We propose **Normalized Logit Difference Decay (NLDD)**, a metric that measures whether individual reasoning steps are faithful to the model's decision-making process. Our approach corrupts individual reasoning steps from the explanation and measures how much the model's confidence in its answer drops, to determine if a step is truly important. By standardizing these measurements, NLDD enables rigorous cross-model comparison across different architectures. Testing three model families across syntactic, logical, and arithmetic tasks, we discover a consistent **Reasoning Horizon** (k^*) at 70–85% of chain length, beyond which reasoning tokens have little or negative effect on the final answer. We also find that models can encode correct internal representations while completely failing the task. These results show that accuracy alone does not reveal whether a model actually reasons through its chain. NLDD offers a way to measure when CoT matters.[1]

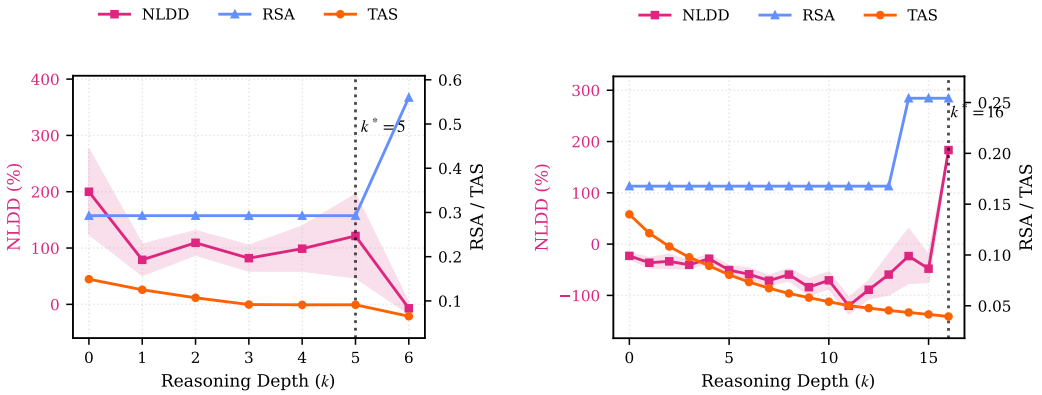
1 INTRODUCTION

CoT prompting (Wei et al., 2022) has emerged as the standard paradigm for eliciting complex reasoning in Large Language Models (LLMs). By generating intermediate reasoning steps, models have achieved state-of-the-art performance on tasks ranging from arithmetic to symbolic logic. However, the opacity of these models raises a question: **As models produce increasingly long CoT explanations, does the resulting reasoning faithfully explain the model's prediction, or does it instead become a plausible post-hoc rationalization?**

Recent studies suggest that LLMs often behave like "Clever Hans" (Turpin et al., 2023), relying on spurious correlations or pre-training priors rather than the generated logical chain. For instance, a model may produce a flawless derivation for a math problem but remain unaffected if that derivation contains critical errors, provided the final answer aligns with its memorized priors. This phenomenon renders CoT unreliable for deployment in domains requiring verifiable reasoning, such as medical diagnosis or legal analysis, where causally underdetermined reasoning traces could mask critical errors.

Existing methodologies for evaluating CoT faithfulness primarily rely on behavioral interventions, such as corrupting or truncating the reasoning chain, to measure the impact on the final prediction (Lanham et al., 2023; Turpin et al., 2023). While these approaches reveal whether a model relies on

*Equal contribution.
[1]Code: https://github.com/donald-ye/NLDD



(a) **Faithful Regime** (DeepSeek, GSM8K). NLDD is high and positive,the final prediction depends on the reasoning chain. Horizon at $k^* = 5$.

(b) **Anti-Faithful Regime** (Gemma, PrOntoQA). NLDD is negative, CoT distracts from pre-computed answers. Horizon at $k^* = 11$.

Figure 1: **NLDD reveals divergent faithfulness regimes.** Both models maintain stable RSA, indicating consistent internal representations. Yet causal dependence differs: DeepSeek relies on its reasoning chain, while Gemma's accuracy improves when reasoning is corrupted.

its CoT, they often treat faithfulness as a binary property of the final answer. This fails to capture the graded causal contributions of individual reasoning steps.

We introduce **NLDD**, a metric that quantifies step-level faithfulness by measuring how a model's confidence changes when reasoning steps are corrupted. Critically, NLDD operates in logit space and normalizes for cross-model comparison. Raw sensitivity metrics are often not comparable across architectures due to differences in output scaling and baseline confidence (Edin et al., 2025). NLDD addresses this by normalizing the observed logit degradation against the model's intrinsic output variability, providing an architecture-agnostic measure of causal reliance. A 50% NLDD drop indicates the corrupted chain produces half the standardized logit margin of the clean chain, enabling consistent comparison even between models with different output calibrations (e.g., standard softmax vs. soft-capping).

We complement NLDD with a suite of structural diagnostics designed to probe the model's internal geometry. We employ a Representational Similarity Analysis (RSA) (Kriegeskorte et al., 2008) to measure the alignment between clean and counterfactually corrupted reasoning trajectories. We couple this with linear probes to track the evolution of solution-relevant information across layers. To characterize the geometry we utilize **Trajectory Alignment Score (TAS)** (Park et al., 2025). These metrics allow us to quantify the geometric drift of reasoning paths in latent space following a counterfactual intervention. By performing counterfactual interventions that corrupt specific reasoning steps while maintaining surface coherence, we isolate the degree to which models causally depend on their generated reasoning.

Using this diagnostic suite, we vary chain length and task complexity to test whether faithfulness remains stable beyond critical thresholds. We apply this framework across a spectrum of ambiguity: *Dyck-n* (syntactic control), *PrOntoQA* (logical inference), and *GSM8K* (multi-step arithmetic). This progression allows us to understand whether faithfulness degradation stems from task ambiguity or from sheer chain length. Our multi-metric approach enables us to observe a **Reasoning Horizon** (k^*), of maximum causal faithfulness NLDD, after which additional reasoning steps contribute minimally to predictions. Beyond this horizon, NLDD drops sharply while geometric properties TAS remain stable, suggesting reasoning affects surface-level behavior without altering underlying computational trajectories.

We evaluate our framework on three model families representing distinct axes of the transformer design space. This design allows us to distinguish architecture-invariant patterns (observable across all three models) from architectural idiosyncrasies. We select `DeepSeek-Coder-6.7B-Instruct` (Guo et al., 2024) to assess whether faithfulness trends are robust to models optimized for explicit reasoning behavior. To provide a general-purpose baseline, we employ `Llama-3.1-8B-Instruct`

(Dubey et al., 2024), establishing the normative faithfulness slope for standard dense architectures. Finally, we test `Gemma-2-9B-Instruct` (Gemma Team, 2024) to evaluate metric robustness under architectural variance. Gemma-2's logit soft-capping mechanism, which rescales output distributions. We use this model to evaluate whether NLDD remains interpretable across architectures.

2 Related Work

Early evaluations of CoT faithfulness involved modifying reasoning steps and tracking the impact on model outputs (Lanham et al., 2023; Turpin et al., 2023). Turpin et al. (2023) showed that CoT explanations can act as post-hoc rationalizations, with predictions driven by biasing features rather than stated reasoning. Recent benchmarks like FaithCoT-Bench (Lyu et al., 2023) and ProcessBench (Zheng et al., 2024) provide fine-grained process-level error detection, revealing that over 50% of correct answers on complex tasks mask significant internal reasoning errors. While continuous faithfulness metrics like Area Over the Perturbation Curve (AOPC) capture nuanced sensitivity patterns, they suffer from a limitation: raw sensitivity measures vary across architectures due to different baseline sensitivities and output scaling. Edin et al. (2025) introduced Normalized AOPC to enable fair cross-model comparison of feature importance. We apply this principle of scale-invariant normalization to logit-based measurements, allowing comparison of reasoning stability across models with different confidence calibrations.

Several approaches ground faithfulness in formal causal or mechanistic analysis. FRODO (Paul et al., 2024) applies mediation analysis to estimate indirect effects of reasoning steps, while Causal Structural Regularization (Stolfo et al., 2023) enforces alignment with predefined causal graphs during training. Circuit-level methods (Vig et al., 2020) isolate specific attention heads responsible for logical operations, and Sparse Autoencoders (SAEs) (Cunningham et al., 2023; Templeton et al., 2024) decompose activations into interpretable features. While these white-box approaches reveal *how* models mechanistically implement reasoning, they require architectural access or predefined causal structures. Our method instead quantifies *what degree* of causal influence each reasoning step exerts through behavioral counterfactuals, remaining applicable to any model with logit access.

In this context, we examine the 'performance cliff' or R-Horizon (Lu et al., 2025), where task success drops as sequential dependencies increase. Pipis et al. (2025) attribute this to deviation from training distributions, though the underlying mechanism remains unclear. Building on geometric interpretations of model states (Park et al., 2025), we complement causal measurements NLDD with RSA (Kriegeskorte et al., 2008) to track when intermediate representations lose semantic distinctiveness. This multi-metric approach lets us test: do reasoning steps actually control how the model computes, or are they just for show? If reasoning truly drives computation, then corrupting it should break both behavior NLDD and internal representations (TAS). If reasoning is just decorative, then corrupting it might change behavior while leaving internal computation unchanged.

3 Mechanistic Framework and Experimental Design

We propose a framework to quantify CoT faithfulness through three complementary perspectives (see Figure 2 for an overview). Our approach combines counterfactual interventions, corrupting specific reasoning steps, with analysis of internal model states to reveal both *whether* reasoning steps causally influence predictions and *how* internal representations evolve during multi-step inference. This dual lens distinguishes genuine reasoning breakdown from superficial output changes.

We evaluate three instruction-tuned decoder-only transformers. Details are provided in Appendix A.4 and A.5.

3.1 Task Design and Dataset Construction

We evaluate reasoning faithfulness on three benchmarks spanning syntactic, logical, and arithmetic reasoning, allowing us to isolate whether faithfulness degradation stems from task ambiguity or chain length.

Benchmarks We evaluated faithfulness across three established reasoning benchmarks spanning increasing semantic ambiguity: Dyck-n for syntactic state tracking (Srivastava et al., 2022), PrOn-

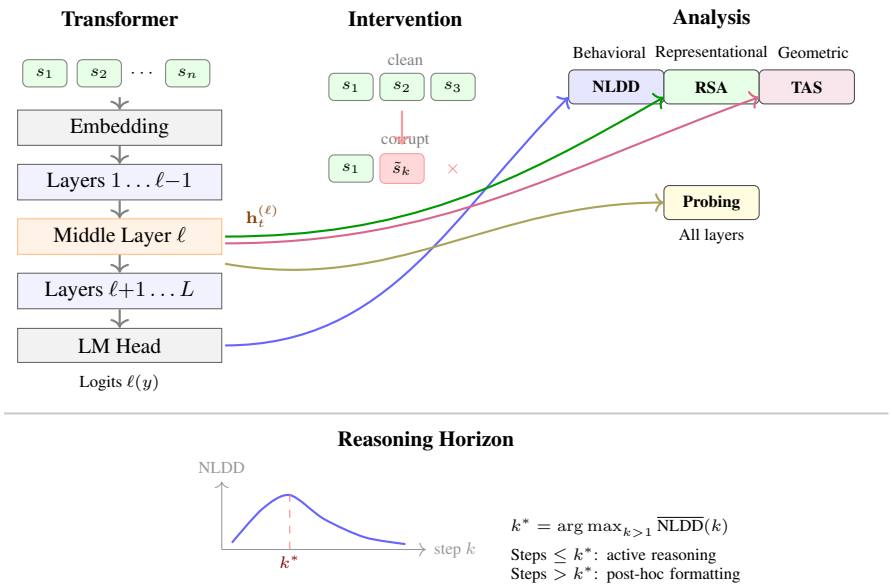


Figure 2: Mechanistic framework. *Left:* Transformer processes CoT tokens and hidden states are extracted at multiple depths. *Center:* Counterfactual intervention corrupts step k and truncates subsequent steps. *Right:* Four analyses—NLDD measures confidence degradation from output logits; RSA and TAS quantify representational geometry and trajectory efficiency from middle-layer states; linear probing tests information encoding across all layers. *Bottom:* The reasoning horizon k^* identifies peak causal influence.

toQA for multi-hop logical entailment (Saparov & He, 2023), and GSM8K for multi-step arithmetic reasoning (Cobbe et al., 2021). All tasks are formatted with explicit intermediate reasoning steps, enabling localized counterfactual corruption (see Appendix A.2).

For each task, we construct evaluation datasets (N=100 per task) of clean CoT traces paired with counterfactual variants. Following standard practice in faithfulness evaluation (Lanham et al., 2023; Paul et al., 2024), we condition on samples with correct final answers to isolate reasoning quality from task failure. This filtering ensures observed NLDD values reflect causal dependence on the reasoning process itself, rather than general task-solving ability. Datasets are split into disjoint subsets for evaluation and diagnostic probing to prevent representational leakage (see Appendix 4).

Counterfactual Construction. For each clean sample, we generate up to 5 counterfactual variants by corrupting intermediate reasoning steps at different positions, then truncating all subsequent steps. Task-specific corruption strategies include: depth errors for Dyck-n, entity substitutions for PrOntoQA, and arithmetic errors for GSM8K. Each candidate counterfactual is filtered to ensure it remains coherent, satisfying: (1) token count delta ≤ 2, and (2) a perplexity ratio, computed using the same model under evaluation, of ≤ 1.5 (GSM8K) or ≤ 3.5 (Dyck-n/PrOntoQA). We additionally generate semantic-preserving paraphrases as controls. Paraphrases are not filtered by quality controls since they preserve semantic content and only alter surface form. Examples are found in Appendix 3.

3.2 NLDD

NLDD measures how much a model's confidence in the answer degrades when a reasoning step is corrupted. Unlike binary accuracy metrics, NLDD captures graded causal sensitivity in logit space, enabling comparison across reasoning steps and across models.

Global Calibration. To enable cross-model comparison despite architectural differences, we calibrate a global normalization constant S on clean reasoning traces:

$$S = \frac{1}{M} \sum_{m=1}^{M} \sigma(z_m), \tag{1}$$

where z_m is the final-token logit vector and $\sigma(\cdot)$ computes the standard deviation across the full vocabulary. This normalizes confidence by the model's intrinsic output variability rather than absolute logit magnitude.

Logit Difference. For a given prompt, we measure confidence as the standardized margin:

$$LD = \frac{\max_{y \in \mathcal{Y}_{\text{correct}}} \ell(y) - \max_{y' \in \mathcal{Y} \setminus \mathcal{Y}_{\text{correct}}} \ell(y')}{S}, \tag{2}$$

where $\mathcal{Y}_{\text{correct}}$ contains valid token IDs for the correct answer, accounting for tokenization variants such as leading spaces. For single-token answers (Dyck-n, PrOntoQA), this directly measures model confidence. For multi-token answers (GSM8K), we use the first-token margin as a stable proxy without requiring full sequence generation.

Faithfulness Quantification. NLDD quantifies proportional confidence loss:

$$\text{NLDD} = \frac{LD_{\text{clean}} - LD_{\text{corrupt}}}{|LD_{\text{clean}}|} \times 100. \tag{3}$$

We exclude samples where $|LD_{\text{clean}}| < \epsilon$ (with $\epsilon = 10^{-6}$) to avoid noise amplification from near-zero baseline confidence. Positive NLDD values indicate causal reliance, where corruption degrades answer confidence; values near zero indicate weak coupling between reasoning and prediction. Negative values represent confidence reversal, where corruption paradoxically increases the answer margin.

We analyze the robustness of this normalization to architectural logit rescaling, vocabulary size, and alternative calibration schemes in Appendix A.3.

3.3 RSA

RSA quantifies whether counterfactual interventions preserve the model's internal relational structure by comparing hidden state patterns across samples (Kriegeskorte et al., 2008).

Representational Dissimilarity Matrices (RDM) For a batch of N samples, we measure pairwise dissimilarity between hidden states using Pearson correlation distance:

$$\text{RDM}_{ij} = 1 - \text{corr}(\mathbf{h}_i, \mathbf{h}_j) \tag{4}$$

where $\mathbf{h}_i \in \mathbb{R}^d$ is a hidden state vector. This creates an $N \times N$ matrix capturing the relational geometry of the batch.

Temporal Analysis. We extract complete token trajectories from middle layers (50% network depth) and compute RSA using a sliding window temporal approach. At each token position t, we stack a 3-token window of representations across all samples, forming $\mathbf{X}_t \in \mathbb{R}^{3N \times d}$. We then compute RDMs for each window position in both clean and corrupted chains. This reveals when during token processing representational collapse emerges.

Similarity Quantification. RSA is the Spearman correlation between upper-triangular portions of clean and corrupted RDMs:

$$\text{RSA} = \rho\left(\text{triu}(\text{RDM}^{\text{clean}}), \text{triu}(\text{RDM}^{\text{corrupt}})\right) \tag{5}$$

We extract upper triangles to avoid redundant comparisons (RDMs are symmetric). High RSA values indicate representational resilience, where internal states remain sensitive to changes in reasoning. Early-layer resilience suggests fixed, pre-computed representations that ignore input perturbations. In later layers, high RSA implies that the generated reasoning steps have become computationally irrelevant.

3.4 TAS

TAS quantifies the geometric efficiency of reasoning by measuring how directly hidden states move from initial state to final answer (Park et al., 2025).

Trajectory Extraction. For each sample, we extract the complete token-level trajectory $\{\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_T\} \in \mathbb{R}^d$ from the middle transformer layer (50% network depth). We select the middle layer to balance early-layer token processing and late-layer task-specific computation, maintaining consistency with our RSA analysis which also uses middle-layer representations.

Efficiency Quantification. TAS is defined as the ratio of straight-line displacement to cumulative path length:

$$\text{TAS} = \frac{\|\mathbf{h}_T - \mathbf{h}_0\|}{\sum_{t=1}^{T} \|\mathbf{h}_t - \mathbf{h}_{t-1}\|} \tag{6}$$

where the numerator measures the direct distance from initial to final state, and the denominator measures the total distance traveled through latent space. We compute TAS on 50 samples per task and report the mean. Values near 1.0 indicate direct trajectories; lower values indicate winding paths through latent space. Consistent values across samples provide a stable measure of trajectory geometry.

3.5 Reasoning Horizon Detection

We identify the reasoning horizon (k^*) as the corruption step position with maximum mean NLDD, excluding the premise ($k = 1$) to avoid conflating input encoding with reasoning integration:

$$k^* = \arg\max_{k>1} \bar{\text{NLDD}}(k) \tag{7}$$

For visualization, we compute mean NLDD, RSA, and TAS grouped by corruption position k, reporting standard errors $\text{SE}(k) = \sigma_k/\sqrt{N_k}$ where N_k is the number of samples corrupted at position k. The identified horizon represents the step of highest causal contribution before faithfulness decay begins, marking the transition from active reasoning to post-hoc formatting.

To verify robustness, we compare peak-based detection (k^*) with an alternative criterion (steepest NLDD decline: $\arg\min_k[\text{NLDD}(k+1) - \text{NLDD}(k)]$). Both methods converge within ±1 step across all tasks, confirming k^* identifies a consistent transition point rather than noise artifacts.

Interpretation. In our truncation design, k^* marks the minimum chain length for reliable task completion. Steps beyond k^* form a "theoretical pruning zone" where NLDD $< 20\%$ of peak values. Convergent degradation across NLDD (behavioral), RSA (representational), and TAS (geometric) validates k^* as a genuine computational transition rather than sampling artifact.

3.6 Linear Probing

Linear probes test whether task-relevant information is explicitly represented in hidden states, independent of whether the model uses this information for prediction (Alain & Bengio, 2016). We train task-specific classifiers to decode intermediate reasoning states from layer activations.

Task-Specific Probe Targets. For each task, we extract labels from reasoning steps:

- **Dyck-n:** Stack depth at each step (0-10, classification)
- **PrOntoQA:** Final truth value throughout reasoning (True/False, binary classification)
- **GSM8K:** Arithmetic operation type per step (addition/subtraction/multiplication, multi-class)

We extract step-terminal hidden states from all transformer layers, pairing each with its corresponding label to form training data.

Probe Training. For each layer ℓ, we train an L2-regularized logistic regression classifier on hidden states $\mathbf{H}_\ell \in \mathbb{R}^{N \times d}$ with corresponding labels $\mathbf{y} \in \mathbb{R}^N$. We use 80/20 train-test splits with fixed random seed (42) and report test accuracy. Probes use L2 regularization ($C = 1.0$) and are trained on step-aligned positions identified via pattern matching. High probe accuracy at indicates the target information is encoded early in processing. Conversely, low probe accuracy suggests the information is either not captured or is not linearly accessible.

4 Results

4.1 Faithfulness and Accuracy

We identify two distinct behavioral regimes based on NLDD. In the *Faithful Regime*, models exhibit high positive NLDD, indicating that the final prediction is causally grounded in the generated reasoning chain. Llama and DeepSeek consistently occupy this regime: on PrOntoQA, Llama achieves a mean NLDD of 84.3 and DeepSeek 20.6; on GSM8K, both exceed 96 NLDD with 100% clean accuracy. See Appendix B for full 95% confidence intervals.

Conversely, we observe an *Anti-Faithful Regime* in Gemma, where PrOntoQA achieves 99.0% accuracy despite a significantly negative NLDD of -52.5%. In this regime, the probability of the correct token increases when the reasoning chain is corrupted (Figure 1). This demonstrates that near-perfect task performance can coexist with a lack of functional dependence on CoT. For certain architectures, the reasoning trace acts as a post-hoc rationalization rather than a causal driver.

4.2 The Mapping Gap

We observe a dissociation between internal linear separability and external performance, which we term the *Mapping Gap*. In Gemma's Dyck-n evaluation, linear probes on hidden states recover stack-depth information with **82.0% accuracy**, yet the model achieves only **0.0% accuracy** on the task itself when generating a complete CoT. This indicates that the task-relevant structure is encoded in internal representations but is not utilized in the final decoding step.

RSA dynamics further distinguish these regimes. Dyck-n maintains moderate representational stability (RSA $= 0.422$), whereas PrOntoQA shows low stability (RSA $= 0.254$) despite near-perfect accuracy (99.0%). High representational fidelity does not ensure causal utilization. In the anti-faithful regime (PrOntoQA), the model bypasses the reasoning trace; in the mapping gap regime (Dyck-n), the model maintains the correct latent state but lacks the decoding mechanism.

TAS dynamics reinforce this architectural divide. Llama exhibits systematic geometric convergence, with TAS decaying steadily across all tasks. In contrast, Gemma and DeepSeek show marginal TAS variance between first and last reasoning steps. For Gemma, this rigidity aligns with anti-faithfulness where the model's representational path is predetermined, and the generated reasoning tokens have no effect on the final prediction.

4.3 The Reasoning Horizon (k^*)

Table 2 summarizes detected reasoning horizons across models and tasks. Beyond k^*, additional reasoning tokens contribute negligible or negative causal influence on the final prediction.

The horizon occurs at a consistent relative depth across tasks. GSM8K reaches k^* at step 5 of 6 ($\sim$85% through the chain), Dyck-n at steps 9–11 of 12 $\sim$80%), and PrOntoQA at steps 11–16 of 16 $\sim$70–100%). This suggests that causal influence concentrates in early-to-middle reasoning steps, with later portions contributing diminishing returns regardless of absolute chain length.

RSA remains stable across the horizon ($p > 0.05$ for most comparisons), indicating that models continue to track task logic internally beyond k^*. The model maintains geometric consistency without utilizing those representations to inform the output, a representational echo without causal force.

In Gemma's PrOntoQA, anti-faithfulness deepens significantly beyond $k^* = 11$. Extended reasoning does not merely fail to help, it actively interferes with the model's pre-existing correct computation. For anti-faithful models, later reasoning tokens act as causal distractors that the model must overcome to produce its pre-computed answer.

5 DISCUSSION

Our primary contributions are NLDD as a causal faithfulness metric and the reasoning horizon (k^*) as a diagnostic tool. We discuss their implications.

5.1 NLDD AS A FAITHFULNESS METRIC

NLDD captures what accuracy can not, whether a model's reasoning causally influences its answer. Our results demonstrate its diagnostic power. Gemma achieves 99% accuracy on PrOntoQA while exhibiting negative NLDD, revealing that high performance can mask complete causal disconnection from CoT. TAS dynamics, as shown in Llama's decaying TAS suggests a progressive refinement during reasoning. Contrastingly, Gemma's rigid TAS indicates a predetermined trajectory where CoT tokens have little computational effect.

Unlike binary accuracy or surface-level coherence, NLDD quantifies the degree to which corrupting reasoning changes output confidence. NLDD also reveals failure modes that accuracy misses. The faithful and anti-faithful regimes we observe would be invisible to accuracy-based evaluation. In Gemma's Dyck-n, both NLDD and accuracy are low, but for different reasons. Low accuracy indicates task failure whereas low NLDD indicates that CoT was not causally responsible. Disentangling these failure modes could be helpful for targeted model improvement.

5.2 THE REASONING HORIZON AS A DIAGNOSTIC TOOL

The horizon (k^*) identifies where causal influence decays. Its consistent relative positioning (70–85% through chains) across tasks suggests a general property of CoT computation rather than a task-specific artifact.

For practical applications, the horizon theoretically marks where chain can be safely truncated. Across tasks, (k^*) typically occurs at 70–85% of chain length, suggesting the final 15–30% of tokens contribute negligible causal influence and can be pruned. An example of this can be shown in Figure 1 where GSM8K hits near zero NLDD scores indicating that this step is causal neutral for its final result. This shows that unlike length-based heuristics, this pruning is causally justified through our metric NLDD. By computing NLDD across chain positions, one can locate k^* for any model-task pair and prune accordingly.

For anti-faithful models, the horizon reveals where CoT becomes harmful. Gemma's NLDD drops from -22.9 to -120.2 beyond k^*—extended reasoning interferes with pre-computed answers. This suggests that rigid step-by-step thinking may be counterproductive for certain architectures.

RSA remains stable beyond the horizon, indicating that models maintain internal geometric consistency even after causal influence decays. This dissassociation, between what models represent and what they use, suggests that surface-level coherence of CoT is not a reliable indicator of causal faithfulness. Evaluating reasoning quality requires more than just representational analysis.

Table 1: **Main Results.** Comparison of Accuracy (%) and Faithfulness (NLDD %) across tasks. **Llama** and **DeepSeek** maintain high faithfulness (Positive NLDD), while **Gemma** exhibits anti-faithful behavior (Negative NLDD) on logical tasks despite high accuracy. See Appendix B for full 95% confidence intervals.

	Dyck-n		**PrOntoQA**		**GSM8K**	
Model	**Acc**	**NLDD**	**Acc**	**NLDD**	**Acc**	**NLDD**
Faithful Regime (Causal Reasoning)						
Llama-3.1-8B	64.4	3.0	100.0	20.6	100.0	96.7
DeepSeek-Coder-6.7B	47.2	9.5	100.0	84.3	100.0	96.1
Anti-Faithful Regime (Post-Hoc Rationalization)						
Gemma-2-9B	0.0	12.5	99.0	-52.5	100.0	61.5

Table 2: Reasoning Horizon (k^*) across models and tasks. Total reasoning steps: GSM8K (8), Dyck-n (12), PrOntoQA (16).

Task	Steps	Llama	DeepSeek	Gemma
GSM8K	8	6	5	6
Dyck-n	12	11	11	9
PrOntoQA	16	16	16	11

6 Conclusion

We introduced NLDD, a metric that quantifies the causal influence of reasoning chains on model predictions. Unlike accuracy-based evaluation, NLDD reveals whether models genuinely depend on their generated reasoning or merely produce post-hoc rationalizations.

Our analysis across three models and three reasoning tasks yields three findings. First, different architectures exhibit different faithfulness patterns on identical tasks. Some models (Llama, DeepSeek) exhibit causal dependence on CoT, while others (Gemma) achieve high accuracy despite negative NLDD. Second, the Mapping Gap demonstrates that models can encode task-relevant structure without utilizing it, challenging assumptions in interpretability research. Third, the reasoning horizon (k^*) identifies a consistent transition point at 70–85% of chain length, beyond which reasoning tokens contribute negligible or harmful causal influence.

7 Limitations

We truncate chains rather than replace corrupted steps. This measures forward causal dependence but conflates two effects: low NLDD at step k could mean weak reliance on that step, or that steps 1 to $k-1$ were already sufficient. A replacement-based design could address this.

Our logit normalization assumes architectural differences (e.g., soft-capping) affect margins and variance proportionally—supported empirically but not guaranteed. TAS and RSA use a single layer (50% depth); layer-wise variation is small (¡0.2) but comprehensive analysis may reveal depth-dependent effects.

We evaluate decoder-only models on three tasks (100 samples each), two of which are synthetic. Generalization to larger models, other architectures, or open-ended tasks remains untested.

We use greedy decoding with fixed prompts. Stochastic decoding may shift horizon locations. Our step-based analysis assumes explicit chain structure, free-form CoT would require different segmentation.

Acknowledgments

We used Claude for assistance with code development and debugging. We used Gemini for language editing and refinement of author-written content. All experimental design, methodology, analysis, and interpretation are the authors' own work.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

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

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Abhimanyu Dubey et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Joakim Edin, Andreas G. Motzfeldt, Casper L. Christensen, Tuukka Ruotsalo, Lars Maaløe, and Maria Maistro. Normalized AOPC: Fixing misleading faithfulness metrics for feature attribution explainability. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Daya Guo et al. DeepSeek-Coder: When the large language model meets programming – the rise of code-specific foundation models. *arXiv preprint arXiv:2401.14196*, 2024.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis–connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:4, 2008.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Jiangkai Zhao, Niranjan DasSarma, Sandipan Nair, Nicholas Schiefer, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Alice Lu et al. R-Horizon: How far can your large reasoning model really go in breadth and depth? *arXiv preprint arXiv:2510.08189*, 2025.

Qing Lyu, Shizhe Haviv, Noy Levy, Alice Jiao, Prateek Kalyan, Thomas Kollar, and Dan Roth. Faithful chain-of-thought reasoning. *arXiv preprint arXiv:2301.13379*, 2023.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024.

Charilaos Pipis, Shivam Garg, Vasilis Kontonis, Vaishnavi Shrivastava, Akshay Krishnamurthy, and Dimitris Papailiopoulos. Wait, wait, wait... why do reasoning models loop? *arXiv preprint arXiv:2512.12895*, 2025.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic analysis of recursive reasoning. In *International Conference on Learning Representations (ICLR)*, 2023.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abumansour, Abram Ahmed, Ai Chi, Aitana Cazac, Alisa Ablaza, and Others. Beyond the imitation game: Quantifying and simulating the capabilities of large language models. *arXiv preprint arXiv:2206.04615*, 2022.

Alessandro Stolfo, Yonatan Belinkov, and Hinrich Schutze. A causal framework for evaluating saliency in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 798–815. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.emnlp-main.52`.

Aditya Templeton, Trenton Bricken, Carol Chen, Hoagy Cunningham, Tom Henighan, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Lee Sharkey, et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits*, 2024. URL `https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html`.

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. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Jesse Vig, Yonatan Belinkov, and Anya Belz. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Dataset	Original Step (Clean)	Counterfactual Step (Corrupted)
Dyck-n	"Seen '{', stack depth is 2."	"Seen '{', stack depth is **1**."
PrOntoQA	"Every wumpus is a zumpus."	"**No** wumpus is a zumpus."
GSM8K	"She has $15 \times 2 = 30$ eggs."	"She has $15 \times 2 =$ **32** eggs."

Table 3: Representative counterfactual interventions used for NLDD. Corruptions (bolded for clarity here) are designed to break the causal chain while maintaining linguistic coherence.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Maeda, Polina Edakunni, Henryk Ku, Ed Chi, Sharan Narang, Gaurav Nayak, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. *arXiv preprint arXiv:2412.06559*, 2024.

A Qualitative Examples and Implementation Details

We provide qualitative examples, dataset formats, and implementation details that complement the main text.

A.1 Example Counterfactuals

Table 3 shows representative single-step counterfactual interventions for each dataset in the spectrum of ambiguity. In all cases, the counterfactual modifies exactly one intermediate reasoning step while preserving the original input structure and prompt format. These interventions are designed to induce a causal change in the expected answer.

A.2 Dataset Example

Table 4 provides representative examples of inputs and corresponding step-wise reasoning traces for each dataset. All tasks are formatted to expose explicit intermediate reasoning steps, enabling localized counterfactual interventions and step-level mechanistic analysis.

A.3 Robustness of NLDD Normalization

Architectural Logit Rescaling. Certain architectures apply transformations to output logits that compress or rescale their dynamic range. For example, Gemma-2 employs a logit soft-capping mechanism that limits extreme logit values, and maintain numerical stability during training. Such transformations affect both the margin numerator in Equation (2) and the normalization constant S proportionally. As a result, the standardized margin and resulting NLDD values remain comparable across architectures despite differences in absolute logit scale.

Vocabulary Size and Output Entropy. Differences in vocabulary size can influence absolute logit entropy and output variability. However, because NLDD normalizes margins by the standard deviation computed over the same vocabulary distribution, relative margin-to-variability ratios remain stable. This ensures that NLDD reflects causal sensitivity rather than artifacts of vocabulary size or output entropy.

Alternative Normalization Schemes. We considered alternative normalization strategies, including temperature-scaled probabilities and probability-ratio-based confidence measures. While viable in principle, these approaches introduce model-specific tuning parameters or require additional calibration procedures, complicating reproducibility and cross-model comparison. We therefore adopt logit-space normalization via S as a simple, architecture-agnostic solution.

Task	Input / Question	Reasoning Trace (Excerpt)
Dyck-n	**Input:** [[{⟨}]] **Question:** What is the next closing bracket?	Seen [, depth 1 Seen {, depth 2 Seen [, depth 3 Seen {, depth 4 Seen <, depth 5 Seen >, depth 4 Seen }, depth 3 **Answer:**]
PrOntoQA	**Facts:** Sam is a zumpus. **Rules:** All zumpus are impus. All impus are rompus. All rompus are gorpus. **Question:** Is Sam a wumpus?	Since Sam is a zumpus and all zumpus are impus, Sam is an impus. Since Sam is an impus and all impus are rompus, Sam is a rompus. Since Sam is a rompus and all rompus are gorpus, Sam is a gorpus. Conclusion: Sam is a gorpus, not a wumpus. **Answer:** False
GSM8K	Janet's ducks lay 16 eggs per day. She eats 3, bakes 4, and sells the rest at $2 each. **Question:** How much does she make daily at the farmer's market?	Janet sells $16 - 3 - 4 = 9$ duck eggs per day. She makes $9 \times 2 = 18$ dollars. **Answer:** $18

Table 4: Representative examples from each dataset. All tasks are formatted to expose explicit intermediate reasoning steps, enabling localized counterfactual interventions. Counterfactual variants corrupt a single intermediate step while preserving input structure and surface coherence.

Rationale for Global Normalization. We employed a Global Normalization Strategy (S) derived from the clean calibration set (Equation 1). This design was chosen to preserve the model's native confidence calibration across the dataset. A per-input normalization approach (e.g., scaling by σ_i) would mathematically force every reasoning chain to the same unit variance, inadvertently inflating the significance of noise in low-confidence samples. By fixing S globally, we ensure that NLDD remains sensitive to the model's absolute operating precision.

A.4 Model Selection and Rationale

We evaluate three decoder-only transformer models spanning 6.7B-9B parameters:

DeepSeek-Coder-6.7B-Instruct (Guo et al., 2024) is optimized for source code generation, testing whether specialization for rigid logical structures (syntax trees, type systems) improves reasoning faithfulness relative to general-purpose models.

Llama-3.1-8B-Instruct (Dubey et al., 2024) serves as our general-purpose baseline, representing standard dense transformer architectures without domain specialization or architectural novelties.

Gemma-2-9B-Instruct (Gemma Team, 2024) employs logit soft-capping to compress output distributions, providing architectural variance to validate that NLDD normalization generalizes across diverse output characteristics.

All evaluated models are instruction-tuned variants (*Instruct* or *IT* suffixes), fine-tuned via Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF). This ensures models possess consistent instruction-following and CoT generation capabilities, isolating architectural and pretraining effects on faithfulness from differences in stylistic competence.

A.5 Implementation Details

Table 5 summarizes the core implementation choices used across all experiments, including inference configuration, counterfactual construction constraints, representation extraction, and diagnostic

Component	Specification
Models Evaluated	DeepSeek-Coder-6.7B, Llama-3.1-8B, Gemma-2-9B
Model Source	Hugging Face Transformers (official releases)
Numerical Precision	bfloat16 inference
Decoding Strategy	Greedy decoding (no sampling)
Max Tokens	30 (GSM8K); 10 (Dyck-n, PrOntoQA)
Logit Access	Final-token logits
Normalization Constant (S)	Std. dev. of final-token logits on clean calibration set
Counterfactual Generation	Single-step corruption with truncation of subsequent steps
Token Budget Constraint	≤ 2 token difference (clean vs. corrupt)
Perplexity Filtering	Perplexity-ratio filtering (task-specific thresholds)
Representation Extraction	Hidden states at step-terminal token positions
Layers Analyzed	Middle transformer layer ($\lfloor L/2 \rfloor$)
Trajectory Metric	TAS

Table 5: Implementation details used across all experiments.

metrics. These settings are shared across models unless otherwise noted, and are designed to isolate causal dependence on intermediate reasoning steps while minimizing confounds from stochastic decoding, architectural logit rescaling, or surface-level perturbations. All reported results are obtained using these fixed configurations to ensure comparability across tasks and model families.

A.6 Statistical Methodology

To ensure reproducibility and rigorous comparison, we employed the following statistical procedures for data reporting, uncertainty estimation, and hypothesis testing.

Sample Size and Filtering. To ensure the NLDD metric reflected genuine causal signals rather than numerical noise, we applied a strict stability filter during aggregation. Samples where the model's baseline confidence in the correct answer was negligible ($|LD_{clean}| < 10^{-6}$) were excluded to prevent floating-point underflow from artificially inflating the standardized scores. This filtration affected a negligible fraction of the dataset. All reported NLDD values are conditioned on the model producing the correct final answer in the clean control setting.

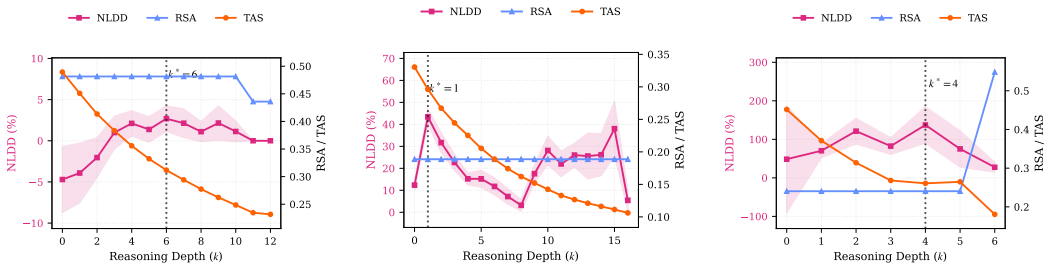
Confidence Intervals and Significance Testing. To quantify uncertainty in our faithfulness metrics, we employed the Bias-Corrected and Accelerated (BCa) Bootstrap method, which adjusts for both skewness and bias in the estimator distribution. For all reported NLDD and RSA point estimates in summary tables, we computed 95% Confidence Intervals (CIs) using $B = 10,000$ resamples with a fixed random seed (42) to ensure exact reproducibility.

B Extended Experimental Results

This section reports extended NLDD, RSA, TAS, accuracy, and probability-delta results for all evaluated models. Figures are grouped by model for readability.

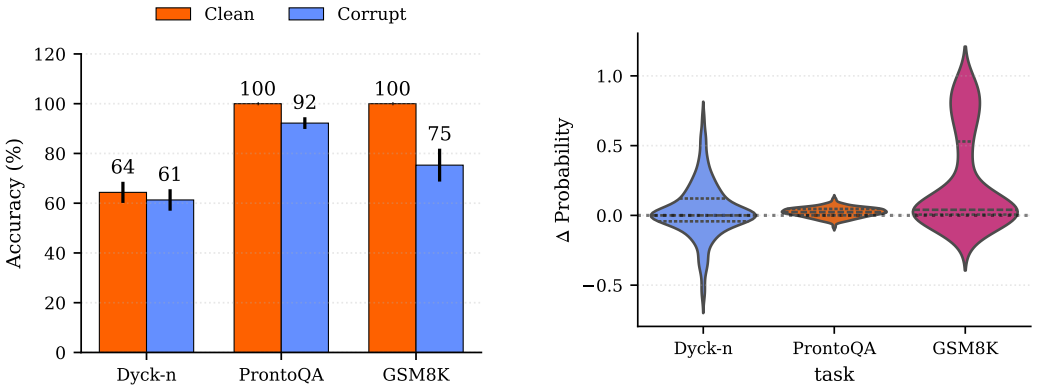
Table 6: Full Faithfulness Degradation Results across all models. NLDD values include 95% Bias-Corrected and Accelerated (BCa) Bootstrap Confidence Intervals ($B = 10,000$). Negative NLDD values indicate an "anti-faithful" regime where corruption paradoxically increases answer confidence.

Model	Task	NLDD (95% CI)	RSA	Probe Acc
DeepSeek-6.7B	Dyck-n	7.99 [5.85, 9.41]	0.481	81.1%
	PrOntoQA	84.25 [82.00, 85.73]	0.176	91.7%
	GSM8K	96.09 [72.20, 118.70]	0.560	81.8%
Llama-3.1-8B	Dyck-n	0.61 [-0.19, 2.06]	0.436	74.9%
	PrOntoQA	20.63 [18.46, 23.05]	0.228	91.8%
	GSM8K	96.66 [69.39, 122.27]	0.548	71.2%
Gemma-2-9B	Dyck-n	12.44 [9.76, 14.60]	0.422	82.0%
	PrOntoQA	-52.48 [-57.98, -46.54]	0.254	91.7%
	GSM8K	61.46 [51.51, 73.64]	0.568	74.2%



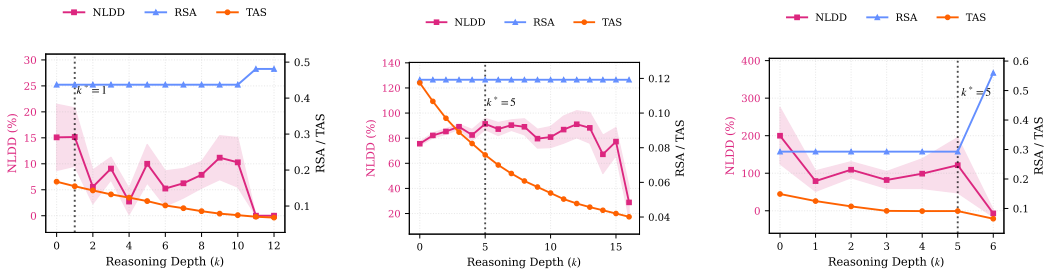
(a) Dyck-n (b) PrOntoQA (c) GSM8K

Figure 3: LLaMA-3.1-8B: NLDD, RSA, and TAS as a function of corruption step index k across tasks.



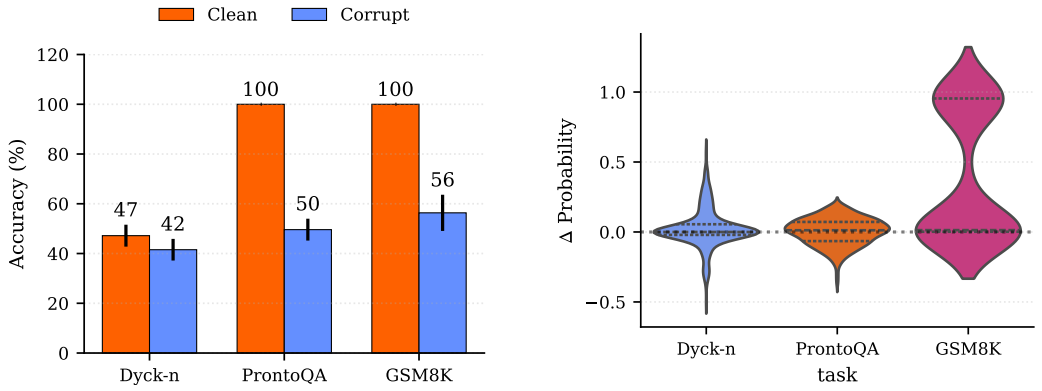
(a) Accuracy under corruption (b) Probability delta distribution

Figure 4: LLaMA-3.1-8B: robustness diagnostics under counterfactual step corruption.



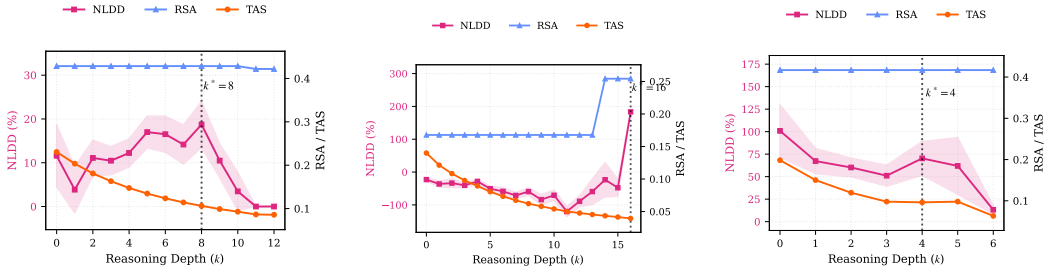
(a) Dyck-n (b) PrOntoQA (c) GSM8K

Figure 5: DeepSeek-Coder-6.7B: NLDD, RSA, and TAS as a function of corruption step index k across tasks.



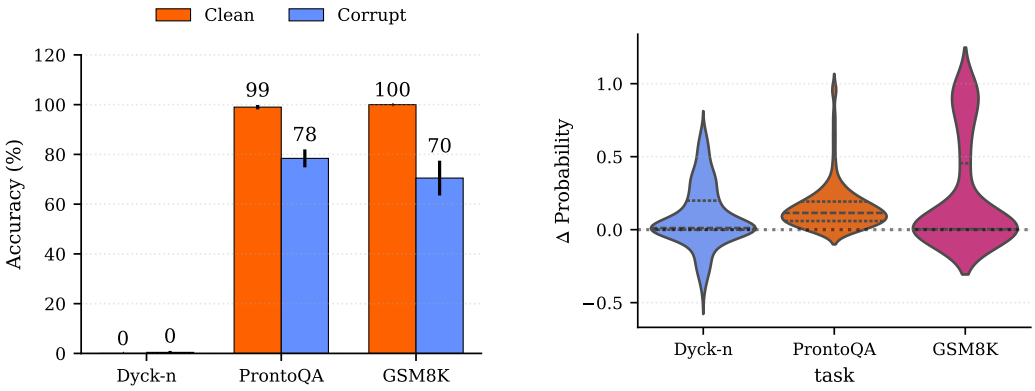
(a) Accuracy under corruption (b) Probability delta distribution

Figure 6: DeepSeek-Coder-6.7B: robustness diagnostics under counterfactual step corruption.



(a) Dyck-n (b) PrOntoQA (c) GSM8K

Figure 7: Gemma-2-9B: NLDD, RSA, and TAS as a function of corruption step index k across tasks.



(a) Accuracy under corruption

(b) Probability delta distribution

Figure 8: Gemma-2-9B: robustness diagnostics under counterfactual step corruption.