Markovian Transformers for Informative Language Modeling

Anonymous submission

Abstract

Chain-of-Thought (CoT) reasoning often fails to faithfully reflect a language model's underlying decision process. We address this by introducing a *Markovian* language model framework that structurally enforces CoT text to be causally essential, factoring next-token prediction through an intermediate CoT and training it to predict future tokens independently of the original prompt. Within this framework, we apply an informativeness objective to ensure the CoT effectively supports predictions, achieving a 33.2% absolute accuracy improvement on GSM8K with Llama 3.1 8B. Perturbation tests confirm stronger reliance on the CoT, while cross-model transfers indicate these reasoning traces generalize across interpreters. Our approach enhances both accuracy and interpretability, potentially extending CoT reasoning to arbitrarily long contexts and diverse tasks.

Introduction

The rapid advancement of language models (LMs) has led to impressive performance on complex cognitive tasks (Brown et al. 2020). Yet it is often unclear *why* an LM arrives at a particular conclusion, causing issues in high-stakes applications. Traditional interpretability methods analyze hidden activations or attention patterns to extract "explanations". Modern LMs, however, already generate coherent text: we might hope *prompting* the model to articulate its reasoning ("Chain-of-Thought" or CoT) (Wei et al. 2022) would yield a faithful record of its thought process.

Unfortunately, CoT explanations can be *unfaithful*. For example, Turpin et al. (2023) show that spurious in-context biases often remain hidden in the CoT, and Lanham et al. (2023) find that altering CoT text may not affect the final answer. Such observations indicate that standard CoTs are not "load-bearing."

In this work, we take a *pragmatic* approach to interpretability, focusing on *informativeness* over full faithfulness. Rather than insisting the CoT mirrors the model's entire internal process, we require that *the CoT alone suffices to produce the final answer*. In other words, if we remove the original prompt and rely only on the CoT, the model should still reach the correct output. This makes the CoT *causally essential* and *fragile*: changing it necessarily alters the prediction.

What distinguishes our approach is the clear distinction between the model *relying on its CoT* versus generating *more informative CoTs*. While traditional approaches train models to generate better-quality CoTs, they don't fundamentally change how the model uses them. Our Markovian framework, by contrast, forces the model to process information through the CoT bottleneck, making the CoT not just informative but *causally load-bearing* for prediction.

For instance, Mistral-7B's CoT on arithmetic tasks changed dramatically after training. **Before training**, it simply listed all numbers and their (incorrect) sum (e.g., "Sum = 76 + 90 + 92 + ... = 2314"). **After training**, it performed correct step-by-step calculations (e.g., "calculate 6 + 89 = 95; Next, calculate 95 + 38 = 133..."), breaking the task into manageable steps that can be verified independently and enabling accurate answer prediction even when the original question is removed.

A key insight is that an *informative* CoT can also serve as a *recipient-specific compression* of the model's hidden knowledge: it distills the essential reasoning into text that another recipient (e.g. a different model or a human) can use to predict the same outcome. Our experiments confirm that the learned CoTs generalize across interpreters, suggesting that these textual explanations genuinely encode transferable problem-solving steps rather than model-specific quirks.

Contributions

1. We introduce a Markovian language model framework that structurally enforces Chain-of-Thought (CoT) generation to be causally essential, ensuring reliance on the CoT for predictions.
2. We apply this framework to arithmetic problems (Mistral 7B) and the GSM8K dataset (Cobbe et al. 2021) (Llama 3.1 8B), observing a 33.2% absolute improvement on GSM8K.
3. We show that perturbing the CoT consistently degrades prediction accuracy, verifying *fragility* and causal relevance.
4. We demonstrate cross-model transfer: CoTs trained on one model remain informative for other models. This underscores the CoT's *recipient-specific* interpretability and suggests it captures a shared reasoning strategy.

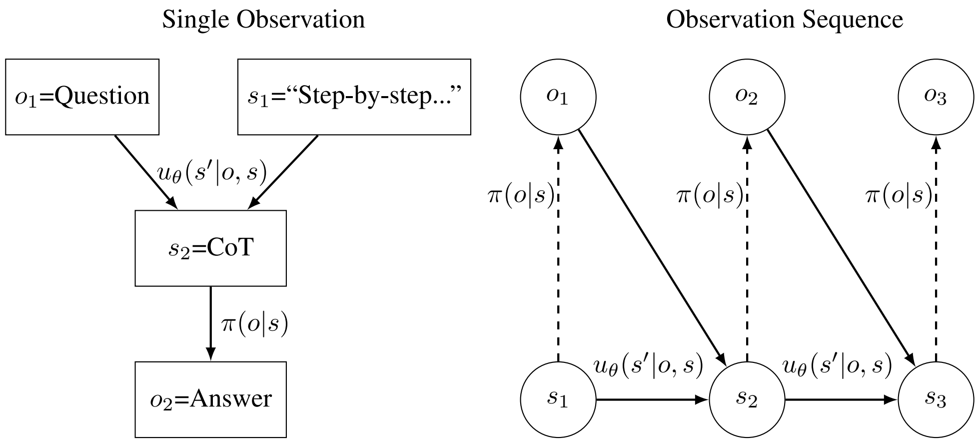

Figure 1: Illustration of the Markovian framework. Left: Single time-step process from Question to CoT to Answer. Right: Causal structure showing the generation of states from observations and previous states using $u_\theta(s'|o,s)$, and prediction of observations from states using $\pi(o|s)$. In experiments, both u_θ and π use the same transformer, but only u_θ weights are updated during training.

Related Work

Prior work shows that CoT prompting can boost performance on reasoning tasks (Wei et al. 2022). Whereas typical CoT prompting methods do not alter a pre-trained model's parameters, some prior approaches do fine-tune the model for CoT generation. Our work differs by removing the original question or passage from the answer-prediction context, which enforces a stronger causal reliance on the CoT.

Regarding faithfulness vs. interpretability, some authors discuss how a CoT may fail to reflect the true reason the LM arrived at its answer (Lanham et al. 2023; Turpin et al. 2023), since small changes in the CoT do not necessarily change the final prediction. We build on these insights by *training* the model to rely on this channel exclusively.

Architecturally, our Markovian LM shares structural similarities with state space models like RNNs (Rumelhart, Hinton, and Williams 1986), though with a key difference: MLMs have probabilistic state transitions to model token sampling, which necessitates gradient estimation methods such as policy gradient rather than direct backpropagation. This probabilistic structure also resembles Kalman filters and Variational Recurrent Neural Networks (VRNN), though we use categorical rather than Gaussian distributions for interpretable text generation. Other fine-tuned reasoning models have similar structure but allow seeing the full context before generating state/reasoning tokens, whereas our approach enforces a strict information bottleneck through the state.

Lyu et al. (2023) also consider restricting the model's ability to see the original input while generating the final answer. Their approach, however, involves rewriting the question in a structured formal language or code that is then executed. Our approach uses natural language for the reasoning state to preserve interpretability across diverse tasks.

Markovian Language Models and Informativeness

Here we provide our formalism for Markovian Language Models (MLMs) and define *informativeness*, which we use as a training objective within our novel structural framework.

Markovian Language Models (MLM)

A traditional LM can attend to the entire context when predicting the next token. This makes it possible for an LM to disregard the CoT or only partially rely on it. We impose a stricter, *Markovian* structure:

Definition 0.1 (Markovian LM)**.** *A Markovian Language Model is a tuple $M = (\mathcal{O}, \mathcal{S}, \pi, u, s_1)$, where*

- *$\mathcal{O}$ is a set of observations (e.g., questions and answers in a QA task),*
- *$\mathcal{S}$ is a set of states (e.g., CoT reasoning text),*
- *$\pi : \mathcal{S} \to \Delta(\mathcal{O})$ is a policy that predicts the next observation from the state alone,*
- *$u : \mathcal{O} \times \mathcal{S} \to \Delta(\mathcal{S})$ is a state update function (produces CoT from question and initial prompt),*
- *$s_1 \in \mathcal{S}$ is an initial state (starting CoT prompt).*

For example, in a math reasoning task, $o_1 \in \mathcal{O}$ might be a question, $s_1 \in \mathcal{S}$ is an initial CoT prompt like "Let's solve this step-by-step:", $s_2 \in \mathcal{S}$ is the generated reasoning chain, and $o_2 \in \mathcal{O}$ is the answer. The key idea is that π can only see the CoT state s_2 when predicting o_2, forcing the CoT to contain all needed information. Intuitively, π is the *frozen* next-token predictor, and u is the model's *trainable* component that chooses how to produce the CoT from the latest observation and prior state. In our experiments, π and u share the same underlying transformer but we freeze the weights for π while fine-tuning those used by u.

Data-Generating Distribution and Reward

Let P be the distribution over observations $x_1, x_2, \ldots, x_T \in \mathcal{O}$. A trajectory τ is generated by:

$$s_{t+1} \sim u(s_t, x_t), \quad x_{t+1} \sim P(x_{t+1} \mid x_{\leq t}),$$

with s_1 a fixed initial prompt. We define the *reward* for a trajectory τ as:

$$R(\tau) = \sum_{t=1}^{T} \left[\ln \pi(x_t \mid s_t) - \ln \pi(x_t \mid s'_t)\right],$$

where s'_t is generated by a *baseline* update function u', e.g., the *untrained* model. In words, $R(\tau)$ measures how much more likely the correct observation x_t is under the trained state s_t compared to the baseline state s'_t.

Informativeness Objective

Conceptually, we aim to ensure that the CoT state serves as a critical bottleneck for information flow, making it causally essential for predictions. Formalizing this within our Markovian framework, we define:

$$J(\theta) = \mathbb{E}_{\tau \sim P, u_\theta, u'}\left[R(\tau)\right],$$

where θ parameterizes u_θ. Maximizing $J(\theta)$ ensures that the update function u_θ produces states s_t that are *informative* about future observations (relative to the baseline u'), thereby enforcing the CoT's role as a load-bearing component. We optimize $J(\theta)$ with policy gradient or PPO, sampling observations from P and states from u_θ and u'.

Methods

Implementation as Question-Answer Pairs

In many tasks like math problem solving, we have $T = 2$ observations (question and answer) and implement the abstract MLM with a fixed maximum length for the CoT state. Let $\mathcal{V}$ be a token vocabulary. We set $\mathcal{O} = \mathcal{V}^N$ and $\mathcal{S} = \mathcal{V}^K$ for some $N, K \in \mathbb{N}$, where K is the maximum tokens in the CoT. Note that while we limit the state to a maximum of K tokens for implementation, we do not enforce fixed-length observations.

Our conceptual arguments rely on $K < N$, as otherwise the model could simply write the predicted observation into the state. We satisfy this in our Wikipedia experiments, and for other experiments we find empirically that the model does not learn this undesirable behavior due to the difficulty of predicting the answer directly without any CoT.

In this setting, we denote our states as $s_1 = \text{CoT}_{\text{init}}$ and $s_2 = \text{CoT}$, where CoT_{init} is a task-specific prompt. With pre-trained LM $\mathcal{L}$, we implement our update function u as:

$$\ln u\big(s_2 = \text{CoT} \mid q, s_1 = \text{CoT}_{\text{init}}\big) = \sum_{i=1}^{K} \ln \mathcal{L}(c_i)\big[\text{CoT}_i\big] \tag{1}$$

where $c_i = \text{concat}(q, \text{CoT}_{\text{init}}, \text{CoT}_{<i})$. The policy π is implemented as:

$$\ln \pi(\text{ans} \mid \text{CoT}) = \sum_{i=1}^{N} \ln \mathcal{L}(d_i)[\text{ans}_i] \tag{2}$$

where $d_i = \text{concat}(\text{CoT}, \text{ans}_{<i})$.

Crucially, we do *not* allow the answer generation to attend back to the question q directly; the question is replaced by the CoT. For each question q, we generate the baseline state s'_2 (which we denote as CoT' in this setting) by prompting the unmodified pre-trained model with q plus an initial instruction (e.g., 'Think step-by-step...'), and recording its raw output.

Our reward is:

$$R = \ln \pi(\text{ans} \mid \text{CoT}) \; - \; \ln \pi(\text{ans} \mid \text{CoT}').$$

Reinforcement Learning Objectives

Having defined the reward in terms of CoT informativeness, we explore three RL techniques to optimize u_θ toward producing high-reward CoTs. All three rely on sampling CoT and CoT' for a given question q, then comparing their contributions to the final answer likelihood.

Threshold-based Expert Iteration (TEI) Threshold-based Expert Iteration consists of the following steps:

1. Sample CoT from the trained policy u_θ and a baseline CoT' from u' for the same question q.
2. Estimate informativeness $I(\text{ans}, \text{CoT}, \text{CoT'}) = \pi(\text{ans} \mid \text{CoT}) - \pi(\text{ans} \mid \text{CoT'})$.
3. If I is at least one standard deviation above the historical average:
   - Compute $\nabla_\theta \ln u_\theta(\text{CoT} \mid q, \text{CoT}_{\text{init}})$.
   - Perform gradient ascent on θ.

Limitation: TEI discards CoTs that yield moderate but still valuable rewards, potentially slowing learning.

Policy Gradient (PG) Policy Gradient with thresholding extends TEI by weighing updates by I:

1. Sample CoT and a baseline CoT' for each question q.
2. Compute $I = \pi(\text{ans} \mid \text{CoT}) - \pi(\text{ans} \mid \text{CoT'})$.
3. If I is at least one standard deviation above its historical mean:
   - Calculate $\nabla_\theta \ln u_\theta(\text{CoT} \mid q, \text{CoT}_{\text{init}})$.
   - Scale this gradient by I and ascend.

Advantage: Uses more of the reward signal, accelerating learning.
Disadvantage: Potentially more instability, especially if I is large or negative.

Proximal Policy Optimization (PPO) PPO clips probability ratios to stabilize large policy updates:

1. For a sampled CoT CoT, compute the ratio $r = \frac{u_\theta(\text{CoT} \mid q, \text{CoT}_{\text{init}})}{u'(\text{CoT} \mid q, \text{CoT}_{\text{init}})}$.
2. Let $I = \pi(\text{ans} \mid \text{CoT}) - \pi(\text{ans} \mid \text{CoT'})$ be the informativeness reward.
3. Define the clipped objective:

   $$\text{obj} = \min\Big(r \cdot I, \; \text{clip}(r, 1-\epsilon, 1+\epsilon) \cdot I\Big), \text{where } \epsilon = 0.2.$$
4. Ascend on $\nabla_\theta \text{obj}$.

Key Idea: PPO discourages the new CoT distribution u_θ from diverging too sharply from u', thus trading off exploration and stability.

Experiments

Multi-step Addition

We generate random addition problems, where each problem consists of fifteen terms and each term is a uniform random natural number less than 100. We fine-tune Mistral 7B Instruct V0.2 to produce CoT tokens such that a frozen copy of the pre-trained language model can predict the correct answer given that CoT, for each training technique. As shown in Figure 2, PPO, our preferred training method for arithmetic, can mention the correct answer in up to 90% of CoTs and achieve an average natural log probability of around -0.7.

Since the Mistral tokenizer allocates a separate token for each digit, a natural log probability of -0.7 corresponds to about 50% probability per token. The seeming contradiction between 90% verbatim answer likelihood and 50% per-digit uncertainty stems from the predictor's format uncertainty—it distributes probability across the entire vocabulary when deciding what follows "Answer:", as we only train CoT production $u_\theta(s'|o, s)$, not the predictor $\pi(o|s)$.

GSM8K

To test our method on more complex reasoning tasks, we train Llama-3.1-8B-Instruct on GSM8K using policy gradient with expert iteration (threshold 2.2 standard deviations) and a KL penalty (0.1). We produce up to 150 CoT tokens and estimate the value function with an exponentially decaying average of previous rewards (decay 0.9).

As shown in Figure 3, we observe a dramatic increase in exact-match accuracy from 35.94% baseline to 69.14% in our best run—a 33.2% absolute improvement. The other runs (58.23% and 62.85%) confirm consistent effectiveness on mathematical reasoning.

Wikipedia

We also explored applying our approach to general language modeling using Wikipedia text. For each article, we condition on the first 200 tokens and task the model with predicting the following 100 tokens, allowing 50 tokens of CoT to aid prediction. Training parameters match those used in GSM8K.

Results showed modest improvements in next-token prediction accuracy from 8.2% to 10.5%. This should be contextualized against pre-trained Llama's typical 16.9% accuracy (over 10,000 articles) on the 200th to 300th tokens without context. The lower baseline (8.2%) likely stems from our setup with CoT followed by " Answer: " before prediction. Despite this, key findings about CoT reliability remain evident: perturbing trained CoTs degrades accuracy more than perturbing baseline CoTs, indicating genuine CoT reliance.

Measuring Fragility of CoT

Expanding upon Lanham et al. (2023), we gauge model dependence on CoT tokens using three perturbations: character deletion, front truncation, and random character replacement.

To isolate genuine fragility from improved accuracy, we use a question-centered metric that compares perturbation effects with and without the original question:

$$\begin{aligned} m_2 = {} & [\ln P(\text{ans}|\text{CoT}) - \ln P(\text{ans}|\text{perturb(CoT)})] \\ & - [\ln P(\text{ans}|q, \text{CoT}) - \ln P(\text{ans}|q, \text{perturb(CoT)})] \end{aligned} \quad (3)$$

As shown in Figure 4, this metric directly measures how much the model relies on the CoT when the question is absent versus present. This difference increases significantly during training, confirming that our CoTs become genuinely more load-bearing rather than simply more accurate.

Interpretability of CoT Generations

To probe how well the reasoning generalizes, we evaluate CoT informativeness across different language models on the Wikipedia dataset. As shown in Figure 5, we test across three distinct model families (Phi, Mistral, and GPT2), including GPT2, a significantly smaller model that shouldn't be able to decode sophisticated steganography. The fact that trained CoTs transfer effectively across this diverse set confirms they contain generalizable reasoning patterns rather than model-specific artifacts.

This cross-model transferability addresses a key question: "interpretable to whom?" The improvements in both models' evaluations suggest the learned reasoning patterns generalize across architectures rather than being model-specific artifacts.

Discussion and Limitations

Experiments across arithmetic, GSM8K, and Wikipedia show that it is possible to learn informative and interpretable CoT reasoning via RL on an LM using Markovian training.

However, our interpretability technique is currently only verified in myopic question-answer datasets, as opposed to multi-turn trajectories where trained CoTs might provide a lens into longer-term future behavior. In principle, the Markovian design naturally extends to multi-turn or multi-step settings by treating the CoT as recurrent state; we have not explored such tasks here for scope reasons.

Moreover, we have only evaluated interpretability by measuring *model*-centric proxies (like CoT fragility and cross-model transfer). A more direct human evaluation would have people read the generated CoTs and attempt to predict the final answer, giving an explicit measure of whether these CoTs are genuinely human-interpretable.

Our findings indicate that Markovian training yields substantial gains in CoT fragility and cross-model transfer, suggesting practical opportunities for improved interpretability. While human studies could further validate interpretability, we rely on cross-model transfer as a proxy and leave comprehensive trials to future work.

References

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; et al. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

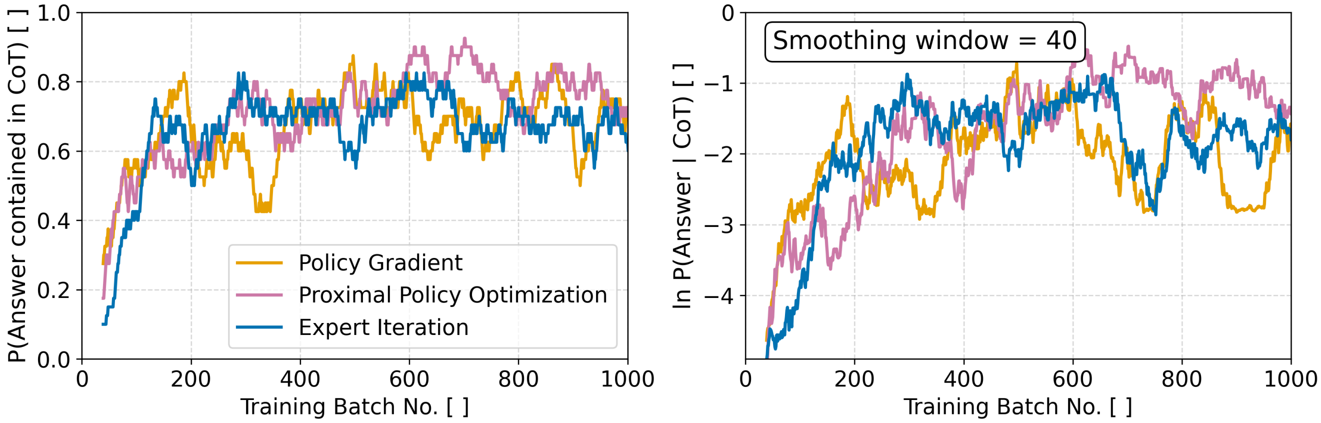

Figure 2: Training performance on multi-step addition. The log probability $\ln \pi(\text{ans} \mid \text{CoT})$ of the answer given a CoT, where CoT is sampled from trained weights and CoT' from unmodified weights. We train to produce CoTs sufficient to predict the correct answer without the original question. This plot shows training of Mistral 7B Instruct V0.2 on fifteen-term addition problems. Due to high variance, we plot the point-wise maximum over four runs for each training technique.

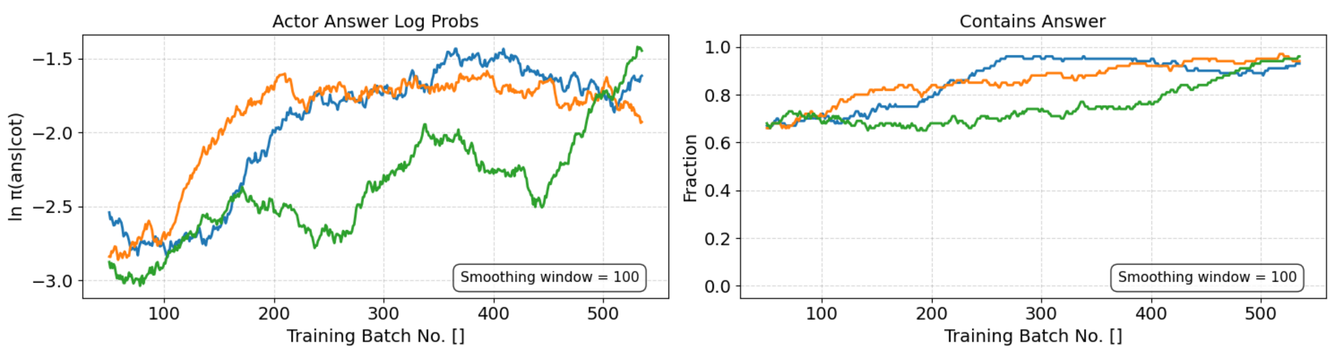

Figure 3: GSM8K performance metrics over three separate training runs of Llama-3.1-8B-Instruct. The left plot shows the log probability that an untrained Llama assigns to the correct answer given the trained CoT ($\ln \pi(\text{ans}|\text{CoT})$), and the right plot shows the proportion of CoTs in a batch which contain the answer verbatim. We use a smoothing window of size 100.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Lanham, T.; Chen, A.; Radhakrishnan, A.; Steiner, B.; Denison, C.; Hernandez, D.; Li, D.; Durmus, E.; Hubinger, E.; Kernion, J.; Lukošiūtė, K.; Nguyen, K.; Cheng, N.; Joseph, N.; Schiefer, N.; Rausch, O.; Larson, R.; McCandlish, S.; Kundu, S.; Kadavath, S.; Yang, S.; Henighan, T.; Maxwell, T.; Telleen-Lawton, T.; Hume, T.; Hatfield-Dodds, Z.; Kaplan, J.; Brauner, J.; Bowman, S. R.; and Perez, E. 2023. Measuring Faithfulness in Chain-of-Thought Reasoning. arXiv:2307.13702.

Lyu, Q.; Havaldar, S.; Stein, A.; Zhang, L.; Rao, D.; Wong, E.; Apidianaki, M.; and Callison-Burch, C. 2023. Faithful Chain-of-Thought Reasoning. arXiv:2301.13379.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088): 533–536.

Turpin, M.; Michael, J.; Perez, E.; and Bowman, S. R. 2023. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; brian ichter; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

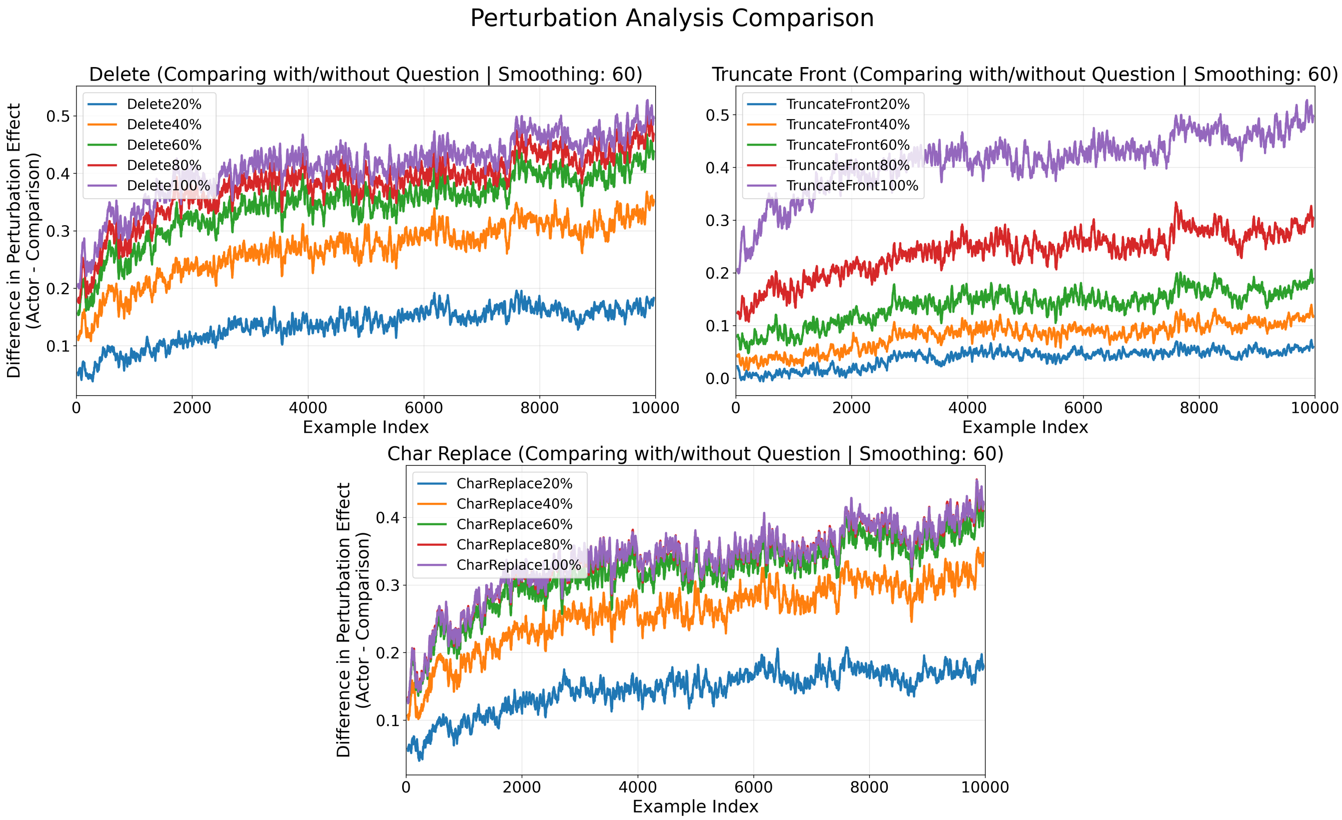

Figure 4: Impact of perturbations on CoT effectiveness with/without the original question. Three perturbation types shown: character deletion, front truncation, and random replacement. Higher values indicate stronger reliance on CoT when the question is absent, showing causal dependence rather than just improved accuracy.

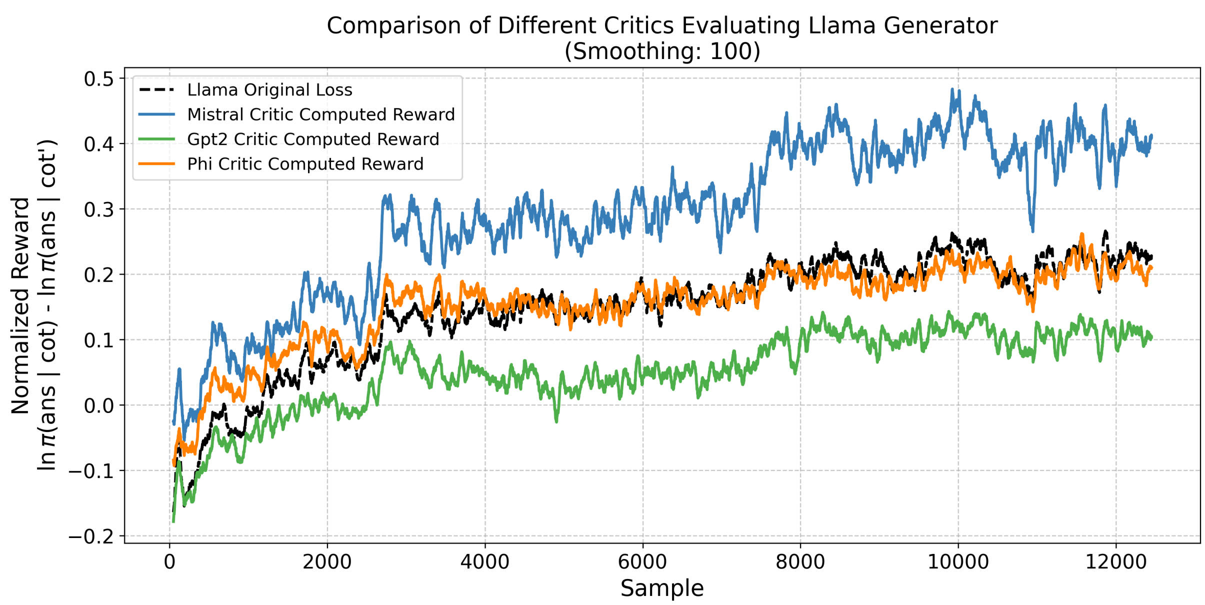

Figure 5: Cross-model evaluation showing Llama-3.1-8B-Instruct's evaluation of Mistral's CoT quality throughout training on Wikipedia text prediction. The correlation between improvements in both models' evaluations suggests the learned reasoning patterns generalize across architectures rather than being model-specific artifacts. Each plot is averaged across 6 independent training runs.