Markovian Transformers for Informative Language Modeling

Anonymous Author(s)
Affiliation
Address
email

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.

1 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 [Lamparth and Reuel, 2023, Burns et al., 2023, Gurnee and Tegmark, 2024], causing issues in high-stakes applications [Grabb et al., 2024, Lamparth et al., 2024, Rivera et al., 2024]. Traditional interpretability methods analyze hidden activations or attention patterns to extract "explanations" [Geiger et al., 2022, Geva et al., 2022, Meng et al., 2022, Casper et al., 2023, Wang et al., 2022, Lamparth and Reuel, 2023, Nanda et al., 2023]. Modern LMs, however, already generate coherent text: we might hope *prompting* the model to articulate its reasoning ("Chain-of-Thought" or CoT) [Nye et al., 2022, 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.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

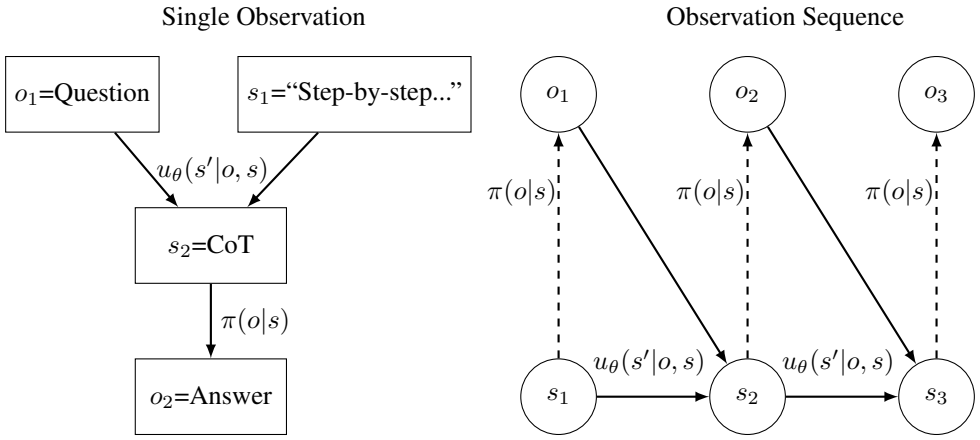

Figure 1: Refined illustration of the training method. Left: Single time-step process from Question to CoT to Answer. Right: Causal structure showing the generation of states from observations and previous states using the state update function $u_\theta(s'|o,s)$, and the prediction of observations from states using the policy $\pi(o|s)$. Observations are generated by the causal data distribution. In experiments, both u_θ and π are Mistral 7B Instruct V0.2 or Llama 3.1 8B Instruct, but only the weights of u_θ are updated during training. The state update u_θ also involves concatenating the observation and state letting Mistral generate the next state's worth of tokens.

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.

Recipient-Specific Compression. 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 (Section 5.5).

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.

Section 2 reviews related work, Section 3 details our Markovian framework, and Section 4 describes the RL training. Section 5 presents empirical results, and Section 6 discusses limitations and future directions.

2 Related Work

Prior work shows that CoT prompting can boost performance on reasoning tasks [Wei et al., 2022, Nye et al., 2022]. Whereas typical CoT prompting methods do not alter a pre-trained model's

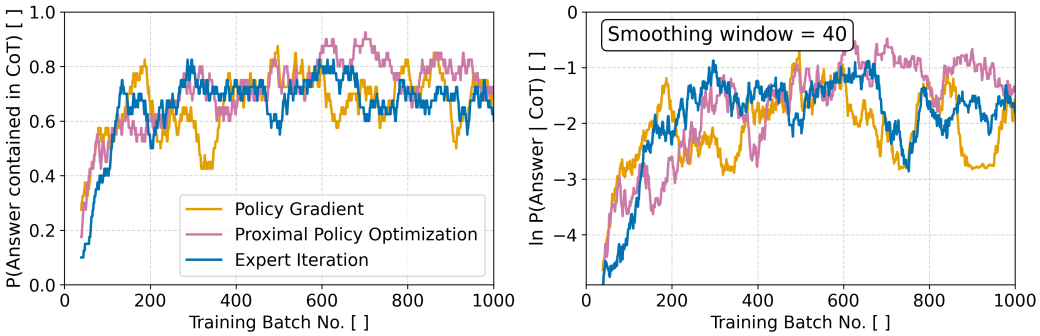

Figure 2: The log probability $\ln \pi(\text{ans} \mid \text{CoT})$ of the answer ans given a CoT, where the CoT is sampled from the trained weights $\text{CoT} \sim u_\theta(\text{CoT} \mid q, \text{CoT}_{\text{init}})$ and CoT' is sampled from the unmodified weights $\text{CoT}' \sim u(\text{CoT} \mid q, \text{CoT}_{\text{init}})$. We train to produce CoTs which are sufficient to predict the correct answer even without the original question, enforcing a text bottleneck in the language model's information flow, forcing the CoT to be causally load-bearing to production of the answer. This plot specifically depicts the training of Mistral 7B Instruct V0.2 on fifteen-term addition problems and their solutions. Because of high variance, we plot the point-wise maximum *over four runs* for each training technique.

parameters, some prior approaches do fine-tune the model for CoT generation [Zelikman et al., 2022, 2024, DeepSeek-AI et al., 2025]. 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. Zhou et al. [2023] analyze CoT through an information-theoretic lens, finding that CoT can serve as a communication channel between different parts of a model. 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 et al., 1986], S4 [Gu et al., 2022], and Mamba [Gu and Dao, 2024], though with a key difference: MLMs have probabilistic state transitions to model token sampling, which necessitates gradient estimation methods such as policy gradient [Sutton et al., 1999] rather than direct backpropagation. This probabilistic structure also resembles Kalman filters [Åström, Karl Johan, 1965], Deep Variational Bayes Filters [Karl et al., 2017], Deep Kalman Filters [Krishnan et al., 2015], and Variational Recurrent Neural Networks (VRNN) [Chung et al., 2015], though we use categorical rather than Gaussian distributions for interpretable text generation. Other fine-tuned reasoning models mentioned above (R1, STaR, and QuietSTaR) 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.

3 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.

3.1 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[1]:

Definition 3.1 (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.

3.2 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.

3.3 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'.

[1]This structure can be viewed as a stochastic variant of a Moore machine where both the transition function (u) and output function (π) are probabilistic, and the input and output alphabets are identical (O). Alternatively, an MLM can be formalized as an F-coalgebra where F(S) = P(O) $\times$ P(S)^O, with P representing probability distributions.

4 Methods

4.1 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 (Sec 5.3), 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[2]. With pre-trained LM $\mathcal{L}$, we can implement our update function u and policy π using:

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

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

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:

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

4.2 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.

4.2.1 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.

4.2.2 Policy Gradient (PG)

Policy Gradient with thresholding extends TEI by weighing updates by I:

[2]The exact prompt template varies by task type, with each template specifying the task objective, allowed CoT length, and an invitation to reason strategically. Full templates are provided in Sec 4.3.

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.

4.2.3 Proximal Policy Optimization (PPO)

PPO clips probability ratios to stabilize large policy updates:

1. For a sampled CoT CoT, compute the ratio $r = \frac{u_\theta(\text{CoT}|q,\text{CoT}_{\text{init}})}{u'(\text{CoT}|q,\text{CoT}_{\text{init}})}$.
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 ∇_θobj.

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

4.3 Training Stability and Implementation Details

Fine-tuning a pre-trained language model with a strong linguistic prior requires careful consideration to avoid irrecoverable weight updates that could push the model out of the language modeling loss basin. In addition to the PPO-clip objective mentioned in Sec. 4.2.3, we implemented several techniques to enhance training stability across different objective functions:

1. **Low-Rank Adaptation (LoRA) [Hu et al., 2022]:**
   - Freeze all weights except for small-rank LoRA adapters.
   - Use rank 8 with $\alpha = 16$.
2. **Gradient Clipping:**
   - If the ℓ_2 norm of the gradient exceeds 1.0, rescale it to norm 1.0.
3. **Gradient Accumulation (Arithmetic Only):**
   - For arithmetic tasks, set batch size to 6 (to fit on one H100 GPU).
   - Accumulate gradients for 8 steps before updating weights.
4. **Average Reward Baseline:**
   - For PPO (and PG variants), we subtract a running average of past rewards from the current reward to stabilize updates.
   - This replaces a learned value function with a simpler baseline, reducing hyperparameter tuning.
5. **Initial CoT Prompt Design:**
   - Choose CoT_{init} to guide the model toward meaningful reasoning.
   - For arithmetic:
     "You will be given an arithmetic problem, which you have [CoT length] tokens to work through step-by-step. Question:"
   - For GSM8K:
     "You will be given a reasoning problem, which you have [CoT length] tokens to work through step-by-step. Question:"
   - For Wikipedia:

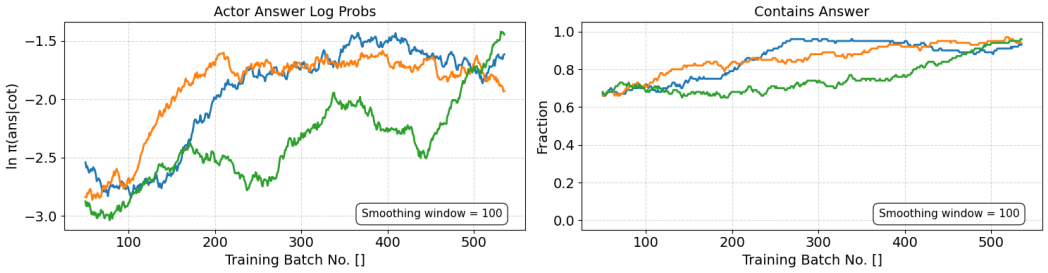

Figure 3: GSM8K performance metrics over three separate training runs of Llama-3.1-8B-Instruct. The left plot shows the log probability that an untrained Llama assigns to the correct answer given the trained CoT — $\ln \pi(\text{ans}|\text{CoT})$, and the right plot shows the proportion of CoTs in a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

> "You will need to predict the next [target length] tokens which follow the provided passage. You can write [CoT length] thinking tokens which will be your sole context for prediction. Feel free to be creative in your thinking strategy! Opening text:"

These measures greatly reduce the risk of catastrophic updates and keep the model's training on track.

5 Experiments

5.1 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 in Sec 4. We plot the mean negative log likelihood over the answer tokens as a function of training batch in Fig. 2. Note that this is both training and testing loss, since we are always generating fresh arithmetic problems. 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 ($e^{-0.7} \approx 0.4966$) 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 $\pi(o|s)$.

5.2 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).

In Figure 3, we show CoT informativeness (left) and proportion of CoTs containing verbatim answers (right). 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.

5.3 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 (Sec 5.2).

Results showed modest improvements in next-token prediction accuracy from 8.2% to 10.5% (supplement). 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: as Fig 4 shows, perturbing trained CoTs degrades accuracy more than perturbing baseline CoTs, indicating genuine CoT reliance. **See supplement for examples of CoT changes after training.**

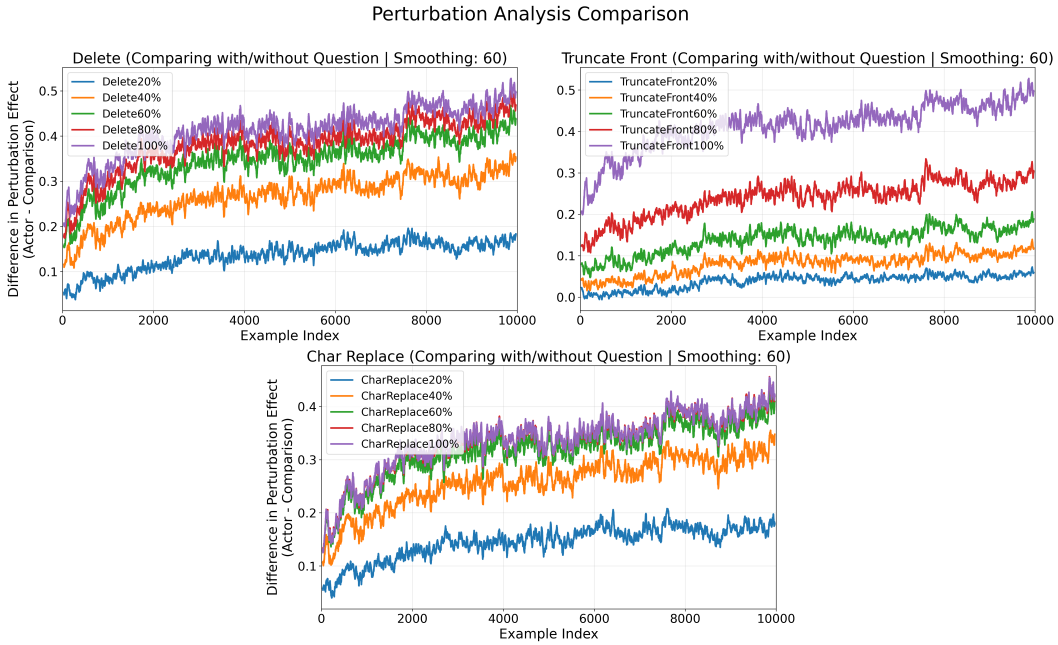

Figure 4: Impact of perturbations on CoT effectiveness with/without the original question. Three perturbation types shown: character deletion, front truncation, and random replacement. Higher values indicate stronger reliance on CoT when the question is absent, showing causal dependence rather than just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

5.4 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:

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

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

5.5 Interpretability of CoT Generations

To probe how well the reasoning generalizes, we plot the informativeness of Llama's trained CoTs with respect to various other LMs on the Wikipedia dataset in Fig. 5. In both plots the normalized log probabilities increase simultaneously, demonstrating that Llama is learning to produce generic CoTs which do not over-fit to the peculiarities of a Llama answer-predictor.

This cross-model transferability addresses a key question: "interpretable to whom?" We test across three distinct model families (Phi [Abdin et al., 2024], Mistral, and GPT2), including GPT2, a

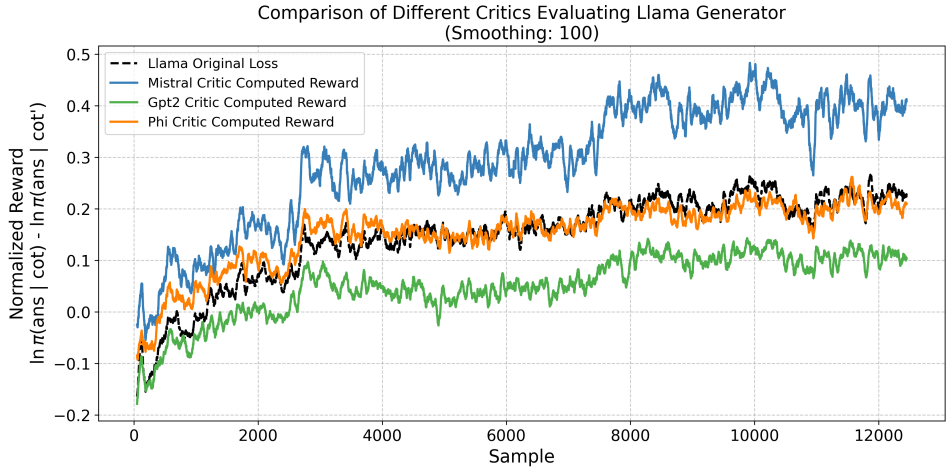

Figure 5: Cross-model evaluation showing Llama-3.1-8B-Instruct's evaluation of Mistral's CoT quality throughout training on Wikipedia text prediction. The correlation between improvements in both models' evaluations suggests the learned reasoning patterns generalize across architectures rather than being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

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.

6 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. Such a setup could be incorporated into the training objective, where human correctness in predicting the answer provides an additional signal for optimizing CoT generation.

Markovian training is language modeling with an intermediate memory-producing action, similar to R1 recurrence and "thinking" models. This approach blurs the line between RL and unsupervised learning, though its expensive serial token generation requires justification through gains in interpretability or perplexity.

Our findings indicate that Markovian training yields substantial gains in CoT fragility (Sec 5.4) and cross-model transfer (Sec 5.5), 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.

Future Work. Although we focus on single question–answer pairs, the Markovian framework extends to multi-turn dialogue. After each user message o_t, we produce the next CoT s_{t+1} via $u_\theta(s_{t+1} \mid s_t, o_t)$, then generate the system's reply from that CoT alone. This process treats the CoT as a recurrent state, which could scale to conversation rounds.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, et al. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, et al. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Stephen Casper, Tilman Rauker, Anson Ho, and Dylan Hadfield-Menell. Sok: Toward transparent AI: A survey on interpreting the inner structures of deep neural networks. In *First IEEE Conference on Secure and Trustworthy Machine Learning*, 2023.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *CoRR*, abs/1506.02216, 2015. URL http://arxiv.org/abs/1506.02216.

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. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning (ICML)*, pages 7324–7338. PMLR, 2022.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 30–45, 2022.

Declan Grabb, Max Lamparth, and Nina Vasan. Risks from language models for automated mental healthcare: Ethics and structure for implementation. *medRxiv*, 2024. doi: 10.1101/2024.04.07.24305462. URL https://www.medrxiv.org/content/early/2024/04/08/2024.04.07.24305462.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022. URL https://arxiv.org/abs/2111.00396.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jE8xbmvFin.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data, 2017. URL https://arxiv.org/abs/1605.06432.

Rahul G. Krishnan, Uri Shalit, and David Sontag. Deep kalman filters, 2015. URL https://arxiv.org/abs/1511.05121.

Max Lamparth and Anka Reuel. Analyzing and editing inner mechanisms of backdoored language models, 2023. URL https://arxiv.org/abs/2302.12461.

Max Lamparth, Anthony Corso, Jacob Ganz, Oriana Skylar Mastro, Jacquelyn Schneider, and Harold Trinkunas. Human vs. machine: Language models and wargames, 2024. URL https://arxiv.org/abs/2403.03407.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2301.13379.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, 2022. URL https://openreview.net/forum?id=iedYJm92o0a.

Juan-Pablo Rivera, Gabriel Mukobi, Anka Reuel, Max Lamparth, Chandler Smith, and Jacquelyn Schneider. Escalation risks from language models in military and diplomatic decision-making, 2024. URL https://arxiv.org/abs/2401.03408.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. doi: 10.1038/323533a0. URL https://doi.org/10.1038/323533a0.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 1057–1063, Cambridge, MA, USA, 1999. MIT Press.

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 *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=bzs4uPLXvi.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488. Curran Associates, Inc., 2022.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-star: Language models can teach themselves to think before speaking, 2024. URL https://arxiv.org/abs/2403.09629.

Dani Zhou, Enyu Zhou, Kevin Han, and Prashant Kambadur. Understanding chain-of-thought in llms through information theory. In *Advances in Neural Information Processing Systems*, 2023.

Åström, Karl Johan. Optimal Control of Markov Processes with Incomplete State Information I. 10:174–205, 1965. ISSN 0022-247X. doi: {10.1016/0022-247X(65)90154-X}. URL https://lup.lub.lu.se/search/files/5323668/8867085.pdf.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our contributions: (1) a Markovian training procedure that makes CoT text causally essential, (2) experimental results showing substantial accuracy improvements, (3) evidence of CoT fragility through perturbation tests, and (4) demonstration of cross-model transferability. These claims are directly supported by our methodology and experiments in Sections 3-5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 (Discussion and Limitations) thoroughly addresses the limitations of our approach. We acknowledge that our interpretability technique has only been verified on question-answer datasets rather than multi-turn trajectories, that we used model-centric proxies for interpretability rather than human evaluations, and that our Markovian training introduces computational costs due to serial token generation steps. We also discuss how these limitations affect the generalizability of our results and potential future extensions.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our paper focuses on an empirical approach to Markovian language models and presents experimental results rather than formal theoretical results requiring proofs. While we provide a definition of Markovian LMs and an informativeness objective in Section 3, these serve as a framework for our methodology rather than theoretical results that would require formal proofs.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our Reproducibility Statement (in the appendix) provides comprehensive details for reproducing our results. We specify all code components (`src/train.py`, `src/perturbation_analysis.py`, `src/evaluate_cross_model.py`), datasets used (GSM8K, HuggingFace Wikipedia), model architectures (Llama 3.1 8B, Mistral 7B, etc.), and implementation details needed to recreate our experiments. Section 3.3 further details our training methodologies, and we specify hyperparameters throughout the paper. Additionally, we'll release our code repository with documentation for reproducing all experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will publicly release our full codebase, including all training and evaluation scripts mentioned in our Reproducibility Statement. Our experiments use publicly available datasets (GSM8K and HuggingFace Wikipedia) and models (Llama 3.1 8B Instruct, Mistral 7B Inst V0.2, Phi 3.5 Mini-Instruct, GPT2). Our code repository includes detailed instructions in the README for environment setup, data preparation, and experiment execution. The `results/Official` directory contains plots, training logs, and evaluation logs to enable comparison with future implementations.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

Justification: Our paper provides comprehensive experimental details throughout Sections 3-5 and in the appendix. We specify all essential training parameters including models used (Llama 3.1 8B, Mistral 7B), datasets (GSM8K, Wikipedia, synthetic arithmetic problems), hyperparameters (LoRA rank 8 with α=16, gradient clipping at 1.0, etc.), and optimization techniques (PPO, PG, expert iteration). Section 3.3 details our training stability measures and prompt templates, while Section 4 describes task-specific configurations like temperature settings, token lengths (150 CoT tokens for GSM8K), and evaluation metrics. The appendix provides additional experimental details including perturbation analysis methodologies, cross-model evaluation protocols, and qualitative examples of generated CoTs before and after training.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report variability across multiple independent training runs (4 runs per experimental configuration) throughout our results. For our main GSM8K experiments in Figure 3, we show results from three separate training runs to demonstrate consistency. In the appendix, we include plots with standard deviation bands around our loss curves to show the statistical variability of our results. These error bands represent the standard deviation across independent runs at each training step, providing a measure of the consistency and reliability of our training approach.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Our experiments were conducted on NVIDIA H100 GPUs through the RunPod cloud service. Each training run took approximately 5 hours on a single H100 GPU, and we performed 4 independent runs for each experimental configuration. Since we explored three different training algorithms (PPO, PG, and TEI) across multiple datasets, the total compute for our final reported experiments was approximately 180 GPU-hours. The full research project, including preliminary experiments with approaches that didn't make it into the final paper, consumed significantly more compute - approximately $30,000 worth of cloud compute resources. This information is provided in our Reproducibility Statement in the appendix to help researchers understand the resources needed to reproduce our results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Our research fully complies with the NeurIPS Code of Ethics. We used only publicly available models and datasets, properly credited all sources, and did not collect any personally identifiable information. Our work focuses on improving interpretability of language models, which aligns with the ethical goal of creating more transparent AI systems. We've included a discussion of potential societal impacts in the Impact Statement section of our appendix, addressing both positive applications (better interpretability) and potential concerns (including how our method relates to controllability and deception).

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The Impact Statement in our appendix discusses both positive and negative societal impacts of our work. We address how our RL approach specifically targets agent foresight rather than general behavior modification. We consider potential concerns that a model with better foresight might use that knowledge deceptively, while also suggesting that improved foresight may lead to better values. We compare our approach with other techniques like RLHF and Constitutional AI, highlighting differences in controllability and how our method focuses on making AI reasoning more transparent and interpretable.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work presents a training methodology for making CoT reasoning more informative, but doesn't release new pretrained language models or datasets that pose significant misuse risks. We use existing publicly available models (Llama 3.1, Mistral 7B) and datasets (GSM8K, Wikipedia) for our experiments, rather than creating new resources that would require safeguards against harmful applications.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use existing datasets (GSM8K and Wikipedia) which are properly licensed and cited in our paper. GSM8K is licensed under the MIT License (openly available), and we used publicly available versions of Mistral 7B (Apache 2.0 license) and Llama 3.1 8B (Llama 3 Community License) language models, which permit research use. These datasets and models are cited appropriately in our references.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: While we plan to release our code, our paper's primary contribution is a methodology (Markovian training for informative CoT reasoning) rather than the creation of new datasets or pretrained models. Our code will be properly documented with instructions in the README, as mentioned in our Reproducibility Statement, but it represents an implementation of our method rather than a novel asset requiring detailed templates or participant consent.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our research does not involve human subjects or crowdsourcing. Our experiments are entirely computational, using publicly available models and datasets, with no human participants or annotators involved in data collection, evaluation, or any other aspect of our study.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

Justification: No Institutional Review Board (IRB) approvals were required for this research since our work did not involve human subjects. All our experiments were computational, using existing datasets (GSM8K and Wikipedia) and publicly available language models.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: While our research uses language models (Llama 3.1, Mistral 7B) as experimental subjects, we did not use LLMs as components of our methodology development process. Our experiments study these models but do not use them as tools to develop our core methods.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.