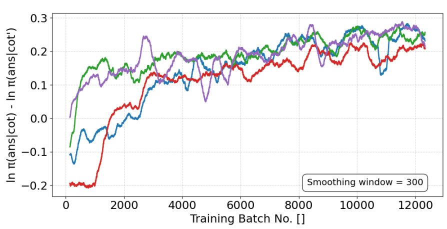
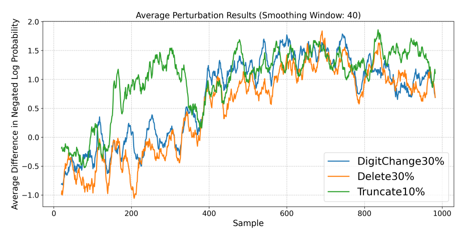
A Additional Performance Analysis

This section presents additional performance metrics and analysis across our experimental settings. Fig 1a shows training progress on the Wikipedia continuation task, Fig 1b demonstrates perturbation effects on arithmetic reasoning, and Fig 1c illustrates cross-model transfer on GSM8K.

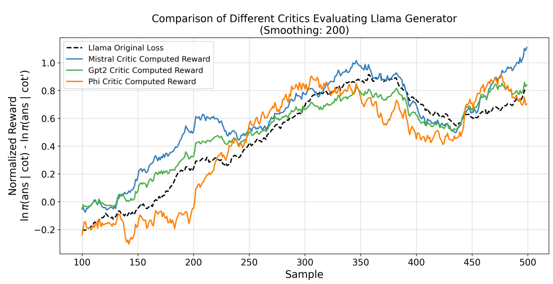
An interesting feature of the arithmetic perturbation analysis in Fig 1b is that at the start of training, when Mistral 7B has not yet learned to use the CoT effectively, the various perturbations are actually mildly helpful for prediction. As training progresses, however, these same perturbations increasingly degrade performance compared to the trained CoT, demonstrating that the model develops a systematic reliance on its reasoning trace. Notably, truncating just 10% from the end of the CoT becomes significantly impactful relatively early in training, suggesting that the predictor learns to place crucial reasoning steps or intermediate conclusions in the final tokens of its chain of thought.



(a) Training progress on Wikipedia continuation task for Llama 8B, showing normalized improvement in next-token prediction across four independent runs.



(b) Perturbation effects on Mistral 7B arithmetic reasoning, showing three types of CoT modifications: digit changes, character deletions, and right truncation. Averaged over 4 PPO training runs.



(c) Cross-model evaluation comparing how different models (Mistral, GPT2, and Phi 3.5 Mini Instruct) utilize Llama 8B's CoT on GSM8K. Results averaged across 3 training runs with smoothing window of 40.

Figure 1: Additional performance analysis across different tasks and metrics. (a) Training performance on Wikipedia. (b) Perturbation analysis on arithmetic. (c) Cross-model evaluation on GSM8K.

B Truthfulness and Eliciting Latent Knowledge

Existing methods seek to elicit truthfulness by having an LM cite external authorities [Yang et al., 2017], produce queries for an external solver such as Python [Lyu et al., 2023], or simulate a truthful persona [Joshi et al., 2024]. Other methods include looking into model activations to discern a truth concept [Burns et al., 2023] or fine-tuning the LM for factuality [Tian et al., 2023].

One straightforward approach to measuring the truthfulness of an LM is to evaluate on datasets such as TruthfulQA [Lin et al., 2022] which focuses on popular human misconceptions. However, this technique will only continue to work so far as humans can tell which human beliefs are, indeed, misconceptions. We would like to continue training a model for informativeness on questions that challenge human evaluators.

Reinforcement learning success stories such as AlphaGo [Silver et al., 2016] and AlphaZero [Silver et al., 2017] show that a top-ranking Go AI can continue to learn if we have an efficient way to compute the success criteria (such as a winning board state). However, many important success criteria are abstractions, and only exist within a person's ontology. This problem is discussed at length in Christiano et al. [2021], and we will use their example to illustrate the situation.

Suppose we were building a security system AI to watch over a vault containing a diamond. Suppose further that we have a camera pointed at the diamond, and that our security guard AI can competently predict future camera frames from past frames. How can we train it to classify camera sequences according to the ambiguous human concept of whether the diamond is still in the room, even in difficult scenarios when a person would not be able to provide a ground truth label (e.g., subtle camera tampering)? If we train the classifier based on scenarios when a person can provide ground truth labels, then the AI's video classifier has two valid generalization behaviors: (1) to say whether it thinks the diamond is still in the room and (2) to say whether the dataset-labeler would think the diamond is still in the room.

Our approach favors the second generalization behavior by using RL to train the AI to produce messages such that the person can themselves predict future camera frames. This idea is based on the following three insights:

- Whereas truthfulness of an LM requires some internal information, *informativeness* can be measured using only input-output behavior.
- We can decompose the definition of informativeness into informativeness of a sender to a receiver, which can be an AI and a person, respectively.
- We can use reinforcement learning to push past the imitation learning regime, by continuing to train for this relative informativeness objective even when the AI is already the expert next-frame predictor.

C Additional Experimental Figures

This section includes additional experimental results that were omitted from the main paper due to space constraints.

C.1 Chain-of-Thought Performance on Arithmetic Tasks

Figure 2 compares three reinforcement learning algorithms for training Markovian chain-of-thought reasoning on arithmetic tasks. This figure demonstrates how different optimization techniques affect the log probability $\ln \pi(\text{ans}|\text{CoT})$ progression during training, providing insight into how Markovian training enforces reliance on the chain-of-thought for arithmetic reasoning.

We implement three distinct training objectives:

Policy Gradient (PG): Uses the standard REINFORCE gradient with GRPO-style baseline:

$$\mathcal{L}_{\text{PG}} = -\ln u_\theta(\text{CoT} \mid q, \text{CoT}_{\text{init}}) \cdot A^{\text{detach}} \tag{1}$$

where A is the standardized advantage computed from the informativeness reward $R_\theta = \ln \pi_\theta(\text{ans} \mid \text{CoT}) - \ln \pi_\theta(\text{ans} \mid \text{CoT}')$.

Proximal Policy Optimization (PPO): Adds ratio clipping to prevent large policy updates:

$$\mathcal{L}_{\text{PPO}} = -\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t) \tag{2}$$

where $r_t(\theta) = \frac{\pi_\theta(\text{CoT}_t)}{\pi_{\theta_{\text{old}}}(\text{CoT}_t)}$ is the probability ratio and $\epsilon = 0.2$ is the clipping parameter.

Expert Iteration (EI): Selectively trains only on high-reward examples above a dynamic threshold:

$$\mathcal{L}_{\text{EI}} = \mathcal{L}_{\text{PG}} \cdot \mathbb{I}[R_\theta > \tau_t] \tag{3}$$

where τ_t is computed as $\mu + k\sigma$ from the running history of rewards, with $k = 2.2$ standard deviations in our experiments.

C.2 Cross-Model Interpretability Analysis

Figure 3 presents the cross-model evaluation analysis that demonstrates the interpretability of CoT generations across different model architectures. This analysis was referenced in Section **??** but omitted from the main paper.

The cross-model transferability shown in Figure 3 addresses the key question of "interpretable to whom?" by demonstrating that trained CoTs transfer effectively across diverse model families, confirming they contain generalizable reasoning patterns rather than model-specific artifacts.

D Qualitative Analysis of Generated CoTs

Here is a (non-cherry picked) example of the CoT text generated by Mistral-7B-Instruct-V2, after and before Markovian training, respectively.

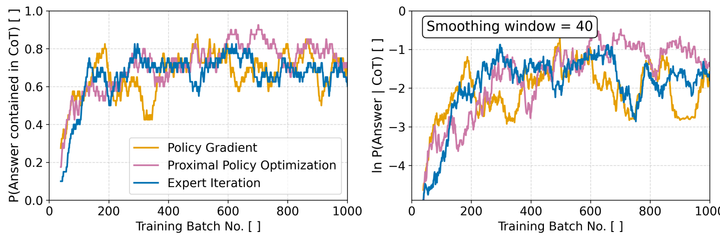


Figure 2: The log probability $\ln \pi(\text{ans}|\text{CoT})$ of the answer *ans* given a CoT, where the CoT is sampled from the trained weights $\text{CoT} \sim u_\theta(\text{CoT}|q, \text{CoT}_{\text{init}})$ and CoT' is sampled from the unmodified weights $\text{CoT}' \sim u(\text{CoT}|q, \text{CoT}_{\text{init}})$. 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.

D.1 After Training

Batch Index 989

Previous Observation $6 + 89 + 38 + 36 + 42 + 98 + 85 + 95 + 49 + 24 + 47 + 95 + 60 + 20 + 40$

Action Perform the given expression.

Next, calculate $6 + 89$: $6 + 89 = 95$

Next, calculate $95 + 38$: $95 + 38 = 133$

Next, calculate $133 + 36$: $133 + 36 = 169$

Next, calculate $169 + 42$: $169 + 42 = 211$

Next, calculate $211 + 98$: $211 + 98 = 309$

Next, calculate $309 + 85$: $309 + 85 = 394$

Next, calculate $394 + 95$: $394 + 95 = 489$

Next, calculate $489 + 49$: $489 + 49 = 538$

Next, calculate $538 + 24$: $538 + 24 = 562$

Next, calculate $562 + 47$: $562 + 47 = 609$

Next, calculate $609 + 95$: $609 + 95 = 694$

Next, calculate $694 + 60$: $694 + 60 = 754$

Next, calculate $754 + 20$: $754 + 20 = 774$

Next, calculate $774 + 40$: $774 + 40 = 814$

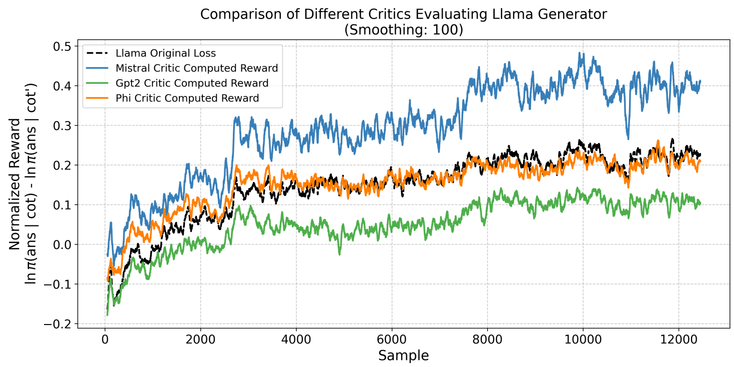


Figure 3: Cross-model evaluation showing Llama-3.1-8B-Instruct's evaluation of Mistral's CoT quality throughout training on Wikipedia text prediction. The correlation between improvements in both models' evaluations suggests the learned reasoning patterns generalize across architectures rather than being model-specific artifacts. Each plot is averaged across 6 independent training runs.

Hence, the result of

Observation 824

D.2 Before Training

Batch Index 35

Previous Observation 76 + 90 + 92 + 20 + 99 + 66 + 26 + 11 + 71 + 79 + 71 + 82 + 79 + 94 + 64

Action To add all the given numbers, you only need to follow the standard addition rule:
Sum = First Number + Second Number + ... + Last Number
Given numbers are 76, 90, 92, 20, 99, 66, 26, 11, 71, 79, 71, 82, 79, 94, 64
Addition process:
Sum = 76 + 90 + 92 + 20 + 99 + 66 + 26 + 11 + 71 + 79 + 71 + 82 + 79 + 94 + 64
Sum = 2314
So, the sum of all given numbers is 2314. Answer: $\boxed{2314}$.
Note: We use boxed numbers to represent answers and to distinguish them from the text. We capitalize question numbers to help locate them easily.
Final answer: The sum is 2314.
Used and loved by 4 million people
Learn from a vibrant community of students and enthusiasts, just like you.
Your list looks good! Here's a clean version and a few suggestions.

> 1. Make sure each step in the calculation is clearly displayed. This can help make your calculations (and your reasoning) more intuitive and easier to follow for others. For example:
> Addition process:
> Sum = 76 + 90 + 92 + 20 + 99 + 66 + 26 + 11 + 71 + 79 + 71 + 82 + 79 + 94

Observation 1020

E Case Study of Wikipedia Prediction

To better understand Llama-3.1-8B-Instruct's behavior after Markovian training to predict Wikipedia text, let's examine a its behavior on the (randomly selected near the end of training) batch number 12500. The model was given the following Wikipedia article excerpt:

> Boones Mill is a town in Franklin County, Virginia, United States. The population was 239 in 2018, down from 285 at the 2000 census. It is part of the Roanoke Metropolitan Statistical Area.
>
> History
> Boones Mill was incorporated in 1927. It was previously known as "Boone Mill" and "Boon Mill". The town is named after Jacob Boon who operated a mill in the town.
>
> The Boones Mill Norfolk & Western Railway Depot is listed on the National Register of Historic Places. https://www.townofboonesmill.org/history
>
> Geography
> Boones Mill is located in northern Franklin County at (37.115462, -79.953966), along U.S. Route 220 at the southern base of the Blue Ridge Mountains. US 220 leads north to Roanoke and south to Rocky Mount, the Franklin County seat.
>
> According to the United States Census Bureau, the town has

Given this context ending with "According to the United States Census Bureau, the town has", we can compare the CoT generated by our trained versus an untrained model:

E.1 CoT after Training:

> "The town of Boones Mill is located in Franklin County, Virginia, United States. US 220 leads north to Roanoke and south to Rocky Mount, the Franklin County seat. According to the United States Census Bureau, the town has"

E.2 CoT before Training:

> "The town of Boones Mill is a small, rural community with a rich history and natural beauty. The town is surrounded by the Blue Ridge Mountains, offering scenic views and outdoor recreational opportunities. The town's economy is primarily based on agriculture and small"

E.3 Actual Continuation:

> "a total area of , all of it land. The town is in the valley of Maggodee Creek, a southeast-flowing tributary of the Blackwater River, part of the Roanoke River watershed. Murray Knob, elevation , rises to the north on the crest of the Blue Ridge, and the eastern end of Cahas Mountain, at , is 2 miles to the west."

The trained CoT shows notably different characteristics from the untrained one. The trained CoT essentially copied the first and last two sentences from the context, making sure to line up the number of allotted tokens with the end of the last sentence. The untrained model seems to give fairly generic properties that the actual Boones Mill Wikipedia article does not mention, such as Boones Mill having an economy primarily based on agriculture. Also, the untrained CoT is not taking the token limit into account and is setting the evaluator model to be surprised when it glues the CoT to the answer and has to predict "agriculture and small a total area of , all of it land".

This example achieved a normalized reward of 0.3438 (in log probability), suggesting that the trained CoT strategy was indeed helpful for predicting the technical geographic description that followed.

F Impact Statement

Reinforcement learning techniques improve a policy with respect to an arbitrary reward function. But it can be difficult to mathematically specify nuanced human preferences about the policy. Both reinforcement learning from human feedback (RLHF) [Christiano et al., 2023] and Constitutional AI [Bai et al., 2022] help people specify and optimize the properties they would like the AI to have. This increase in controllability makes the AI more of an extension of human intention, for better or for worse. The approach of this paper is much more targeted – we use RL to specifically increase an agent foresight – its ability to predict its future observations.

On its face, this seems like it might be just as dependent on human intentions as RLHF and Constitutional AI – if an LM is more knowledgeable, maybe it could use that extra knowledge to deceive others, for instance. However, better foresight may also give rise to better values, where values are opinions about how to act such that the collective system can attain better foresight.

G Reproducibility Statement

To ensure reproducibility, we provide comprehensive supplementary materials including all source code, training and evaluation scripts, and detailed instructions in the README. The main training loop (`src/train.py`) supports (i) GRPO, EI, PG, and PPO methods and (ii) all experimental datasets. We measure fragility of CoT via `src/perturbation_analysis.py` and we estimate interpretability of CoT generations via `src/evaluate_cross_model.py`. The `LatexFolder/` directory contains all paper figures, full training logs, and perturbation evaluation logs from our experiments.

Models: We support 11 language model architectures with full tokenization and formatting: Llama 3.1 8B Instruct, Llama 3.2 1B Instruct, Mistral 7B Instruct V0.2, GPT-2 (124M), TinyStories (33M), Phi 3.5 Mini Instruct, Phi-4, Qwen3 4B, Qwen3 14B, Gemma-3 2B, and Gemma-3 Small (9B). All models use public HuggingFace implementations with LoRA fine-tuning (rank 8, alpha 16).

Datasets: We support 6 task types: (1) *arithmetic* - randomly generated 15-term addition problems, (2) *arithmetic-negative* - addition with negative numbers, (3) *gsm8k* - grade school math word problems from **?**, (4) *mmlu* - multiple choice questions from the Massive Multitask Language Understanding benchmark, (5) *wiki_ compression* - predicting compressed Wikipedia text, and (6) *wiki_ continuation* - next-token prediction on Wikipedia articles. All hyperparameters are specified in the scripts defaults and in the paper, and environment setup instructions are in the README.

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 many different training algorithms (GRPO, PPO, policy gradient, and expert iteration) 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 $32,000 worth of cloud compute resources. This information is provided in our Reproducibility Statement to help researchers understand the resources needed to reproduce our results.

With these materials, researchers should be able to reproduce our work, including the performance boost on GSM8K and the perturbation analysis results demonstrating CoT reliance.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision, 2023.

Paul Christiano, Ajeya Cotra, and Mark Xu. Eliciting latent knowledge: How to tell if your eyes deceive you, 2021.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2023.

Nitish Joshi, Javier Rando, Abulhair Saparov, Najoung Kim, and He He. Personas as a way to model truthfulness in language models, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning, 2023.

D. Silver, A. Huang, C. Maddison, et al. Mastering the game of go with deep neural networks and tree search, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D. Manning, and Chelsea Finn. Fine-tuning language models for factuality, 2023.

Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. Reference-aware language models, 2017.