Objectives Implicit in AI Reasoning Work

Ravi B. Sojitra
Stanford University

First draft: May 4, 2025
This draft: May 4, 2025

Summary

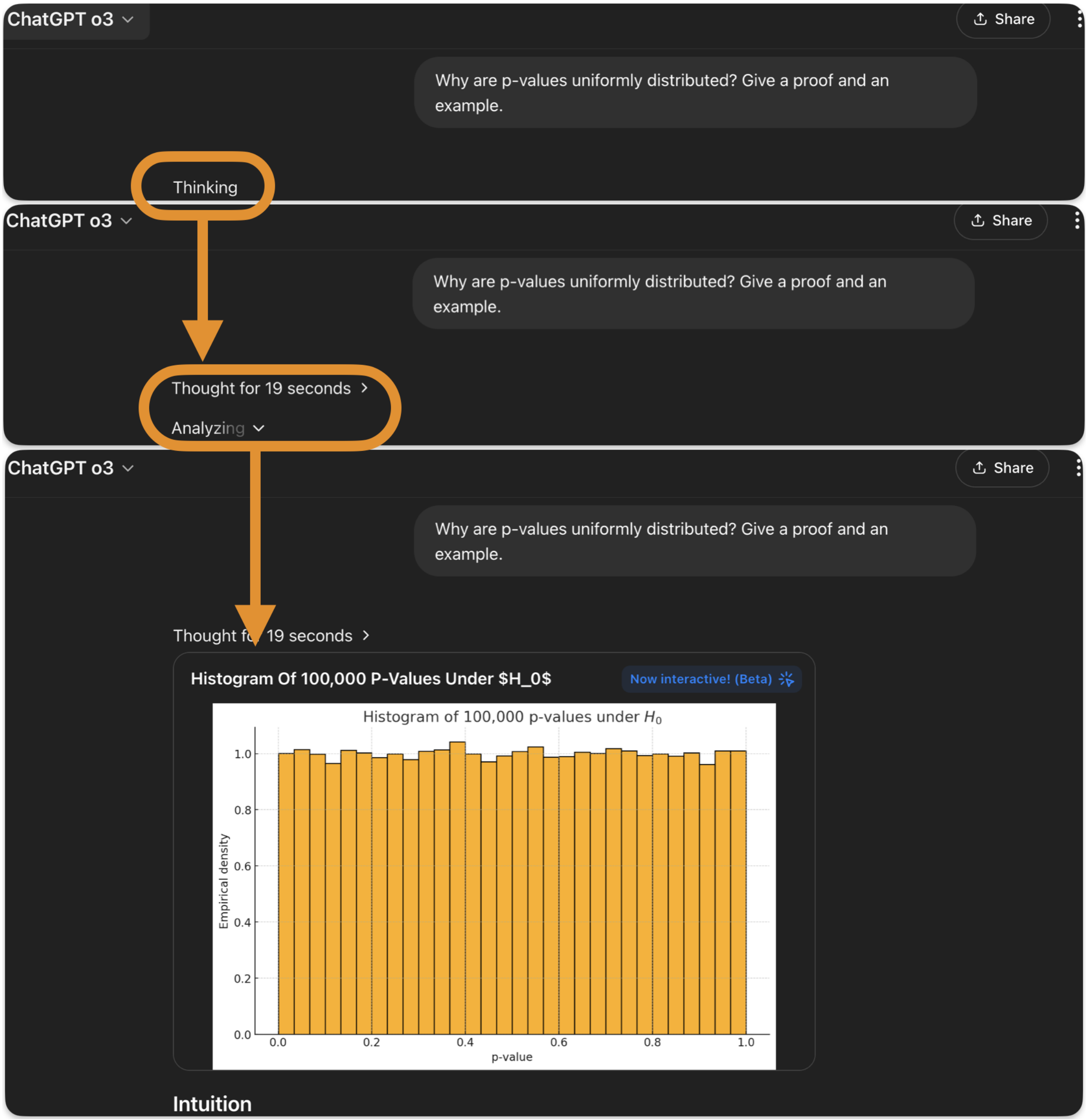
- Artificial Intelligence (AI) companies and scientists define “AI reasoning capabilities” loosely, so it is unclear what it means for an approach to be better than another.
- For example, when training AI to reason before answering math questions, some methods reward correct intermediate steps while others reward final answers only. How should we compare methods? Does it suffice to compare impact on accuracy?
- I have noticed at least four distinct objectives implicit in AI reasoning work: (1) increase accuracy; (2) increase trust; (3) increase generalization; and (4) satiate curiosity. Each prima facie has different technical requirements and evaluation criteria.
- Submit feedback by creating a GitHub Issue at the `public-notes` repository. Thanks!

Background

Frontier AI companies increasingly advertise reasoning capabilities of publicly facing AI systems. For example, Large Language Model (LLM) chatbots ChatGPT o3, Claude 3.7 Sonnet, and DeepSeek R2. When you submit a query or prompt, these AI systems will probably take more time to respond compared to those that do not leverage AI reasoning techniques. However, the responses are expected to be of higher quality, where quality is defined in terms of impact on benchmark task performance. The impact is unambiguously impressive if you are interested in those benchmark tasks (see references in the section titled “Examples of techniques that enable AI reasoning”).

Figure 1 uses three screenshots to illustrate a real interaction I had with ChatGPT o3. Currently, AI chatbot users can explicitly choose to leverage such capabilities or not by selecting the appropriate model. Individual models are described as having reasoning capabilities or not because the costs and latency vary across models (for now).

Figure 1: **An example of ChatGPT o3 "reasoning" before responding to my question**. First (top screenshot), I provide ChatGPT o3 a question and extra information about what I want in an answer. Specifically, a proof and an example. Then (middle screenshot), I wait about nineteen seconds for the model to respond, which is a latency that one does not have in prior versions of AI systems lacking reasoning capabilities. Finally (bottom screenshot), the AI system leads with a visual example before providing an intuitive and formal answer.



Why are p-values uniformly distributed? Give a proof and an example.

Why are p-values uniformly distributed? Give a proof and an example.

Why are p-values uniformly distributed? Give a proof and an example.

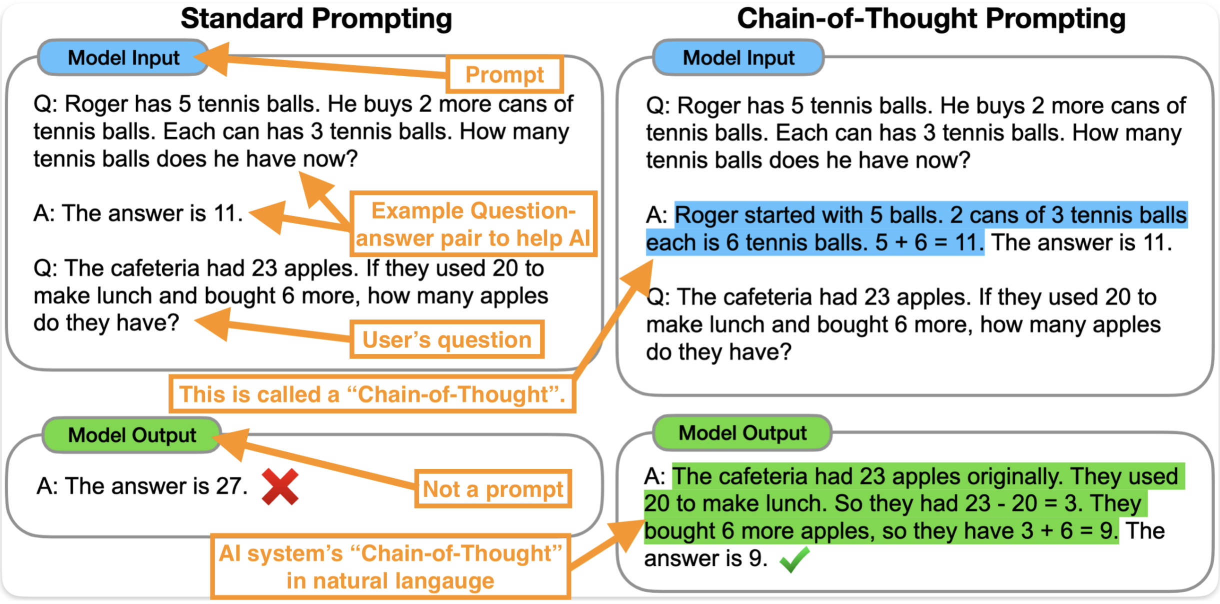
Intuition

Examples of techniques that enable AI reasoning

There is substantial heterogeneity in approaches to enabling and evaluating AI reasoning. However, one concept that appears across all of the LLM work that I have seen is "Chain-of-Thought" (COT) or "reasoning trace". A COT is a sequence of conceptual steps that a human or AI system uses to arrive at an answer. Importantly, COTs may not be written in natural language, may be AI generated, and may not even be correct.

The right side of Figure 2 provides an example of an AI system's response to a **prompt** containing (1) an example question-answer pair, (2) an **example natural language COT**, and (3) a question for the AI chatbot to answer. It illustrates the impact of adding an example COT to a prompt on the correctness of AI's answers to math word problems. Enabling AI reasoning leveraging this observation requires addressing many technical challenges. Based on the work I have seen so far, different methods explicitly focus on questions like the following though it may be unclear why they do.

First, how can one collect high quality COTs? These COTs may be used augment


Standard Prompting
Model Input
Prompt
Example Question-answer pair to help AI
User's question
This is called a "Chain-of-Thought".
Model Output
Not a prompt
AI system's "Chain-of-Thought" in natural langauge
Chain-of-Thought Prompting
Model Input
Model Output

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The answer is 27. ✗

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Figure 2: **Chain-of-Thought example from Wei et al. (2022) that I annotated in orange.** This figure illustrates how a user can improve an AI chatbot's accuracy on math problems by adding a "Chain of Thought" (COT) to a prompt's example question-answer pair. "Prompts" are defined as all of the input users supply to AI chatbots. This includes the main question, contextual information, and examples such as question-answer pairs to help the AI understand what is being asked. Prompts may also include COT examples to illustrate how to reason about similar questions.

prompts as example formats for the AI system to respond with or used to finetune [1] the AI system. Muennighoff and Yang et al. (2025) perform supervised finetuning using high quality, human generated COTs for a range of difficult formal mathematics problems. An important limitation is that these questions, answers, and COTs are difficult to source (e.g. Stanford Statistics Department's doctoral qualifying exam). My understanding is that AI companies spend serious money to source such high quality question-COT-answer math datasets. Wei et al. (2022) achieve impressive results by simply augmenting prompts with a few manually constructed COTs as illustrated in Figure 2. In a recent research paper describing DeepSeek-Prover-V2, Ren and Shao et al. (2025) strive to collect highly correct COT data for theorem proving by leveraging formal verification. A key step was to assess which problems had subproblems that the AI system could solve correctly. Then, they stitched together responses to subproblems to create COTs for the larger problem.

Second, how can one scale COT data collection to many tasks? The approach by Ren and Shao et al. (2025) scales COT data collection while prioritizing correctness. Such approaches are scalable in settings where we have a way to automatically check correctness (e.g. code generation and math), but not in general. Zelikman and Wu et al. (2022) scale COT generation for prompts by iterating between asking the AI system to generate COTs in responses to questions and finetuning on the ones that result in the AI system answering correctly. This requires having answers to the questions a priori. Importantly, the COTs are AI generated and not necessarily correct. Huang et al. (2023) take this a step further by showing that even having the AI generate and use its own COT tends to improve accuracy. This means one can improve performance on sets of questions that lack answers. Shao et al. (2023) go even further by proposing a method for automatically generating novel questions, COTs, and answers. Again, there is no guarantee on correctness of COTs or answers, but the impact on benchmark performance is impressive.

Third, how can one optimize COT formats? Some approaches simply augment prompts with natural language (e.g. append prompts with COT examples padded with "thought" delimiters or tokens like "`<thought>`" and "`</thought>`") while others finetune or modify the underlying LLM powering the AI system. For example, Wei et al. (2022) augment prompts with manually constructed examples of COTs. Muennighoff and Yang et al. (2025) go a step further at test time by suppressing the end-of-thinking delimeter (e.g. "`</thought>`") and augment prompts with "`wait`" to encourage the AI system to generate more text/reason before responding. Zelikman and Wu et al. (2022) go further by finetuning on COTs that resulted in correct answers to generate presumably higher quality COTs. Hao et al. (2023) use "continuous COTs", which are test-time representations (i.e. embeddings) the AI system creates when generating text, instead of forcing the system to use natural language tokens only. Zeliman et al. (2024) introduce more significant changes to the underlying LLM architecture.

[1] Finetuning is a form of model training that is initialized with a model trained on data collected from a relevant setting or task.

Fourth, when training (e.g. finetuning), what should the AI systems objective function be? For example, Muennighoff and Yang et al. (2025) perform supervised finetuning using their COT dataset. Ren and Shao et al. (2025) use both supervised finetuning and Group Relative Policy Optimization. While these approaches reward based on whether the AI's final answer is correct, others reward valid intermediate steps.

Question: Why are there diverging technical design choices?

Taking a step back, the vision for AI reasoning work is unclear. At first, I was surprised that efforts to improve AI reasoning often ignore correctness of logic in COTs used for prompt augmentation and supervised finetuning. I asked whether the goal is to enable deductive and industic logic capabilities or to improve accuracy on well defined tasks.

Answer: There are at least four distinct, implicit goals.

My impression is that unless one is interested in building machines that perform deductive and inductive logic for its own sake, the correctness of COTs used to train or help models is not of primary interest. However, depending on the goal, there are other COT qualities that may be of interest. For example, verifiability and scalability.

During group discussions on Rethinking Foundations of Real-world ML (REFORM), it occurred to me that there are multiple objectives for this research, and there is heterogeneity in how scientists value progress towards each. In practice, this means prioritizing different performance metrics. In hindsight, this is obvious, but it begs the question about what this should mean for technical choices in techniques.

1. Increase accuracy by creating test-time sufficient statistics or features.

It is unsurprising that improving accuracy on well defined tasks is one motivation. After all, the most common objective in statistics and machine learning research is to maximize accuracy of estimators and to do so efficiently. What this means in practice is that, scientists may only care about the qualities of COTs that have an impact on accuracy in benchmark tasks (e.g. code generation, information retrieval, and labeling) and costs related to training data collection.

From this perspective, recent AI reasoning work can be viewed as "test time computing" efforts to improve AI performance. In particular, the idea is to leverage computing resources that are unused when AI systems generate predictions. For example, one may use many Graphics Processing Units (GPUs) to train large neural network or (generalized) linear regression models ("'training time"), but after the models are estimated, a relatively small amount of compute is necessary for prediction ("test time"). Is it possible to use idle machines to further improve performance when models are deployed to production?

Moving forward, this suggests that more scientists who value correctness above other objectives will explore ways to free AI systems from reasoning with natural language (tokens). For example, the Hao et al. (2023) argue that "reasoning in a continuous latent space" leads to more performant AI systems. Concretely, this means that instead of using natural language COTs as illustrated in Figure 2, the AI should perhaps use the embeddings ("representations") it already generates for prediction. I assume their intuition is that there is a loss of information when mapping the AI's embeddings to the natural language token embeddings, so conditioning on natural language tokens not as ideal as conditioning on the raw COT embeddings the model generates to begin with.

2. Increase trust in AI answers with easily verifiable, step-by-step logic.

One aspiration in this space is to develop AI systems that can correctly answer open questions. That is, questions without answers known by humans today. For example, some aspire to build machines that prove novel or outstanding conjectures, and others aspire to build systems that discover new materials. This seems challenging yet possible because of the pace of performance improvement. AI models are increasingly providing accurate responses to difficult questions (e.g. AIME for a mathematics competition, MCAT for medical school admission, LSAT for law school admission).

One challenge that accompanies this aspiration, but not benchmark accuracy improvement, is that users need to be able to easily verify correctness without knowing the truth a priori. For example, if an AI model solves a difficult math homework problem or a proves an outstanding conjecture, the solution has little value unless there is a cheap and fast way to verify correctness of the intermediate steps. That is, users need to be able distinguish between scenarios where the machine is surprisingly correct versus unsurprisingly wrong. For some tasks this is straightforward. For example, for programming tasks, one can write a test suite to verify that the AI generated software behaves as desired.

However, for many tasks, there do not yet exist scalable verification strategies. For example, even manually assessing the quality of substantive claims based on inductive logic and data analysis is exceptionally difficult. If it were easy, the academic peer review process for empirical domains would be more straightforward. So, another approach is to build AI systems that produce valid, logical, intermediate, easy to verify steps that humans can check.

Moving forward, even for this objective, it is unclear to me that AI systems need to train and learn from highly correct and interpretable COTs since a verifiable rationale can be generated post hoc. It seems possible to build AI systems that write natural language COTs to defend answers proposed to questions (e.g. a question-answer pair in a prompt). If a generated COT and proposed answer are correct, human users should be more likely to accept the answer than when the COT is wrong.

3. Increase generalization by scaling up (reasoning trace) data collection.

Another aspiration is for AI systems to perform well on increasing numbers of tasks. A brute force approach would be to enumerate as many tasks as possible and collect high quality examples of task-solution pairs to train on. However, this is financially infeasible as the number of tasks grows. A less data intensive approach could be to collect data only for a subset of tasks that require the same reasoning capabilities necessary as the remaining tasks. However, this subset of tasks is probably still too large. A common strategy for LLMs is to trade quality of COT collection for breadth of tasks for which COTs are collected.

It is unclear what is best, so a combination of approaches is used. For example, for mathematical reasoning and code generation, one could generate many COTs and use only the ones that result in correct final answers. For settings where verification is challenging, one could manually construct a high quality COTs to append AI prompts before automatically generating COTs for a large dataset of questions without COTs or answers.

Moving forward, curriculum learning sounds like a promising direction for improving/reducing the tradeoff between COT quality and task coverage. For example, for deductive reasoning tasks (e.g. code generation, mathematics tasks), it seems more efficient to train AI systems to perform some conceptual subroutines that can be applied across tasks before training it to provide solutions to difficult tasks. During a recent REFORM group discussion, we agreed that this is plausible because the current LLM paradigm is already a form of curriculum learning (pretraining on language, instruction finetuning, Reinforcement Learning from Human Feedback, and other task-specific finetuning), and it has unlocked state of the art AI systems. I believe Ren and Shao et al. (2025) also share this perspective.

4. Satiate curiosity by accomplishing an impressive engineering feat.

It would be impressive for a team to develop and prove that a machine can think on its own and forms its own beliefs. There is debate is about whether investing resources in such an effort is a responsible decision, but to my knowledge there is no debate about it being a curiosity.

From a more pragmatic standpoint, I have noticed that this attitude fosters more creative ideas since there are no a priori commitments to improving specific metrics (e.g. accuracy, validity of logical steps). For example, I admire the scientists who were willing to explore finetuning on synthetic COTs knowing that correctness in the synthetic COTs is going to be compromised. The empirics are impressive, and the idea was counterintuitive for me. Their exploration and results have unlocked new questions for me about machine learning, and this is not something I would have explored if I had been forced to optimize for impact on the AI system's accuracy.