UC San Diego

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Wei, Wang, Schuurmans, Bosma, Ichter, Xia, Chi, Le, Zhou
(Google Research, Brain Team)

Presenter : Sankalpa Hota (PID : A69041322)
Date : April 13 2026
Course : CSE 240D

Problem : LLMs fail at complex reasoning

The Gap

Scaling model size alone
does not solve tasks requiring:

→ Arithmetic reasoning

→ Commonsense reasoning

→ Symbolic manipulation

Even 540B parameter models fail on multi-step math word problems with standard prompting.

Limitation: Rationale Training

Generating natural language rationales requires costly labeling of large training datasets but not scalable for every new task.

Limitation: Standard Few Shot Prompting

Works for simple QA, but performance on reasoning tasks does not improve substantially with increasing model scale.

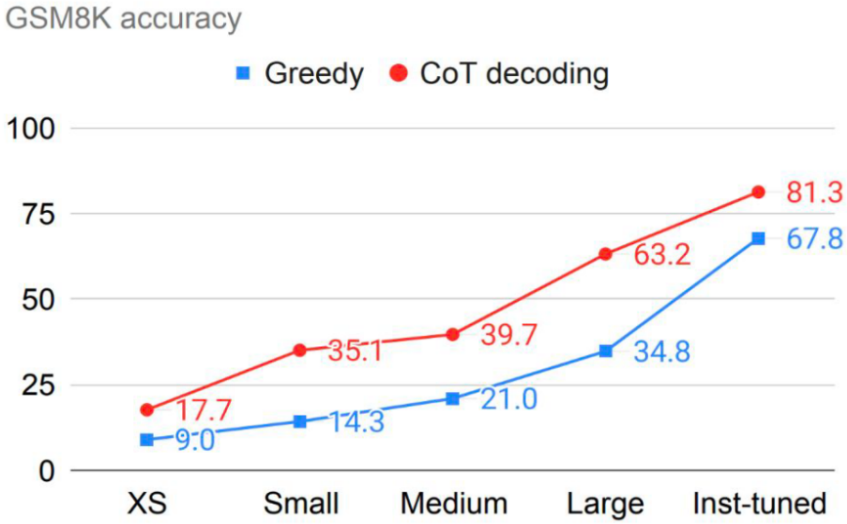
What was happening before Chain of Thought?

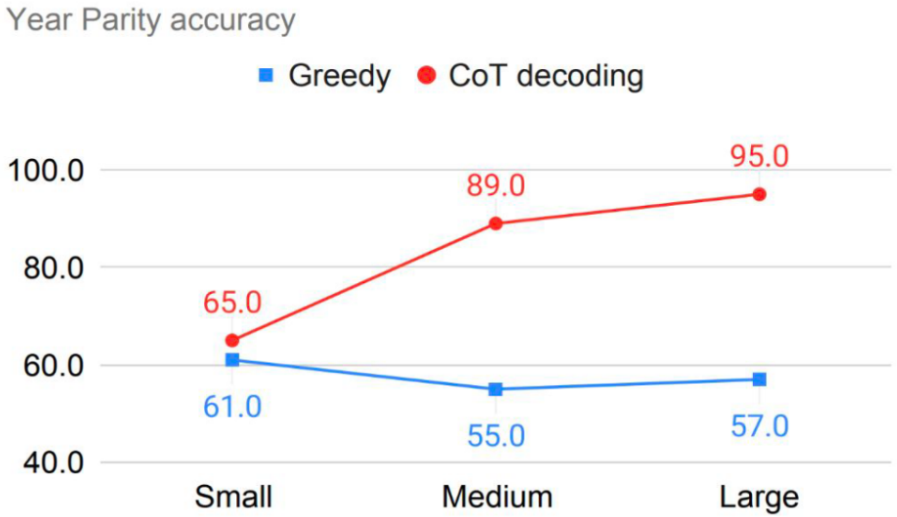
What was happening before Chain of Thought?

Greedy decoding. At every step, the model picks the single highest probability next token and commits to it .No backtracking, no alternatives considered.

argmax $P(tn \mid t_1, t_2, ..., tn\text{-}1)$ (Its greedy so it picks the argmax and moves on :P)

CoT-decoding works reliably across model scales





How would you solve this?

Think of a number

Multiply it by **4**

Add **20**

Divide by **2**

Subtract your original number **twice**

Add **5**

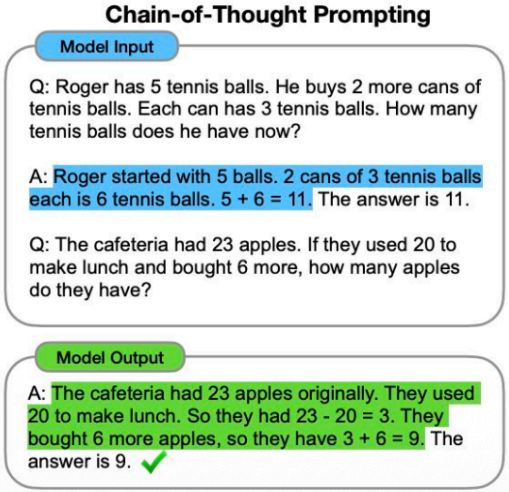
What is your answer?

Answer is always 15!

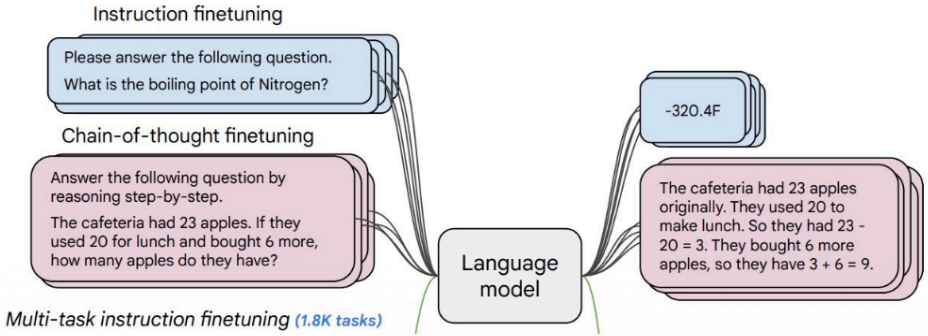
But the way you deduced it step by step and solved it one after another is exactly what Chain of Thought is ,which we will deep dive in next few slides.

Chain of thought reasoning

- Chain of thought has been proven super useful in many reasoning tasks
- How to elicit chain of thought reasoning from LLMs?
  - Chain of thought prompting: few-shot, zero-shot, and many many follow-up works
  - Fine-tuning with a lot of CoT data



Wei et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS 2022.



Chung et al. Scaling Instruction-Finetuned Language Models. JMLR 2024.

Chain of thought reasoning

- Chain of thought has been proven super useful in many reasoning tasks
- How to elicit chain of thought reasoning from LLMs?
  - Chain of thought prompting: few-shot, zero-shot, and many many follow-up works
    - How to disentangle the effect of "human teaching" in the prompt vs. the model's own ability to reason?
  - Fine-tuning with a lot of CoT data
    - Requires collecting a large amount of CoT data

CoT-decoding: Beyond Greedy Decoding Paths

Standard QA format

Q: *I have 3 apples, my dad has 2 more apples than me, how many apples do we have in total?*
A:

Language model

Decoding step 0

top-1: 5
top-2: I
top-3: We
top-4: You
top-5: The

Continue greedy decoding

5 apples ✗

I have 3 apples, my dad has 2 more apples than me, so he has 5 apples. 3+5=8. We have 8 apples in total. ✓

We have 5 apples in total. ✗

You have 3 apples, your dad has 2 more apples than you, so he has 5 apples. 3+5=8. You have 8 apples in total. ✓

The answer is 5. ✗

Ablation Study

Method	How it works	Does it help?	Why / Why not
Standard Prompting	Direct answer, no reasoning	Baseline	
Equation Only	Outputs math equation before answer	Minimal	Helps only for math not for common sense or complex problems
Variable Compute	Spends more compute on complex problems.	No	Extra tokens alone don't help natural language steps matter
Chain of Thought After Answer	Reasoning shown after the final answer	No	Sequential reasoning must come before the answer to be useful
Chain of Thought Prompting	Step by step reasoning before answer	**Yes**	Natural language reasoning guides the model to correct answers. Good performance against common sense and reasoning as well.

Let’s understand the Math

A) Comparison between standard prompting v/s CoT
Standard Prompting :
$y = [a_1, a_2, ..., aT]$ answer tokens only

Chain of thought prompting:
$y = [r_1, r_2, ……., rK, \ a_1, a_2, ……., aT]$
Reasoning tokens Answer tokens

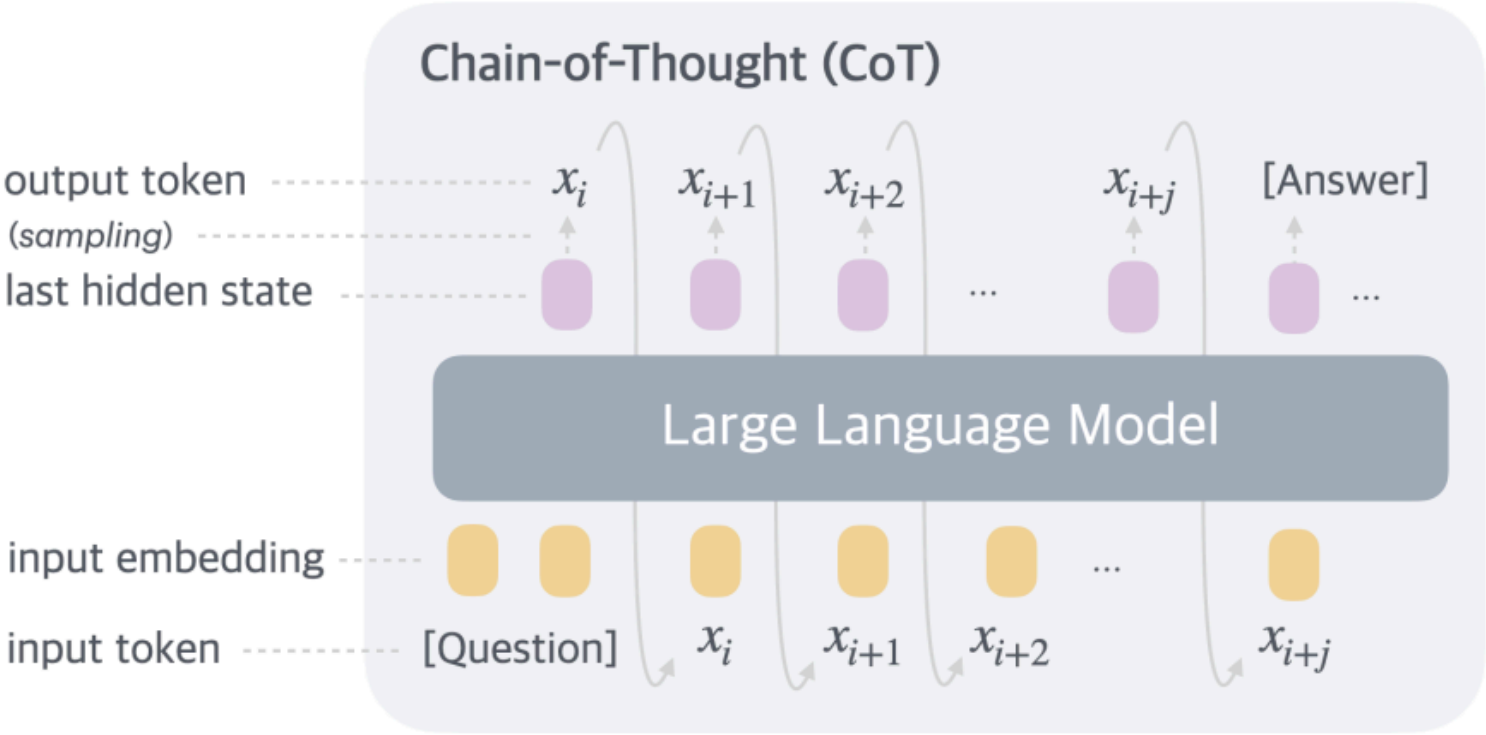
B) The COT model now **predicts reasoning tokens r before answer tokens a**. The answer token probabilities become:
$P(a \mid x) = \prod_{t=1 \text{ to } T} P(aT \mid x, r_1, ..., rK, a<t)$

C) Compared to standard prompting now instead of computing directly like
P(correct answer | hard question),

Now, the model first generates reasoning r, then conditions on it:
$P(a \mid x, r)$ (answer given you have reasoning)
$P(r \mid x)$ (generate reasoning first)

Here, each reasoning step is an easier prediction than jumping straight to the answer.
$P(r_i \mid x, r<i)$

Visualising Chain of Thought Algorithm



Is Chain of Thought compute friendly ?

Is Chain of Thought compute friendly ? **No**
But why ?

It becomes O((n+k)^2) compared to O(n^2) for Attention matrix (QK).

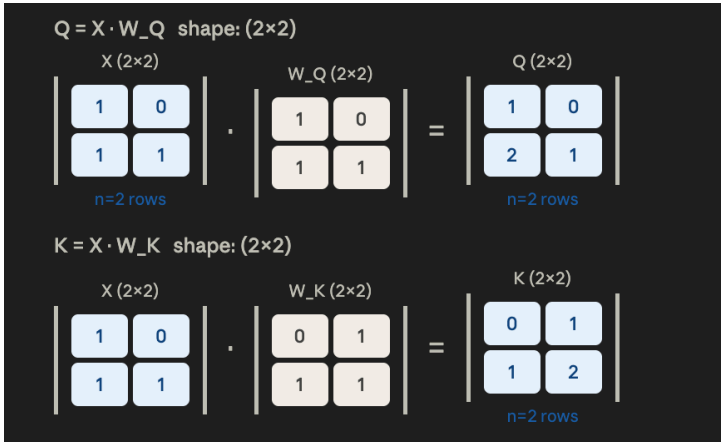
Here , k represents the reasoning tokens.

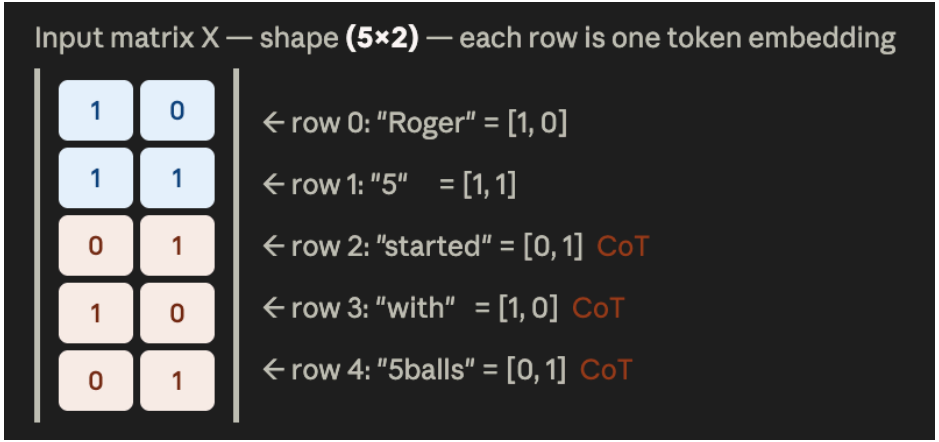
- Understanding QKV? - **Attention is all you need!**
  Q, K, and V are all projections of the same input X. Same data, three different weight matrices pointing it in three different directions in space.

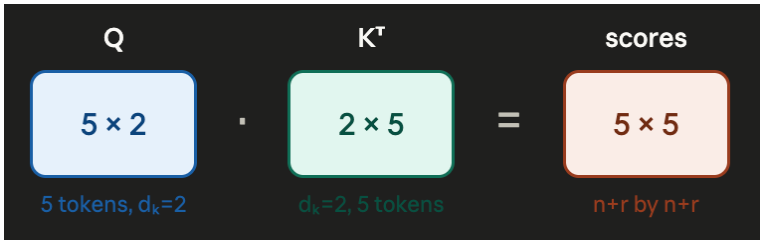
  Q: Each token produces a query vector. This is the ***search request***.
  K: Each token produces a key vector. This is the ***index card***: what it says it contains.
  V: What do I actually hand over once found? ***Retrieval***

  Note in COT: W_Q , W_K, W_V are fixed and the same as the non-COT architecture. Only X changes.

2x2 (Attention) to 5x5 (COT)







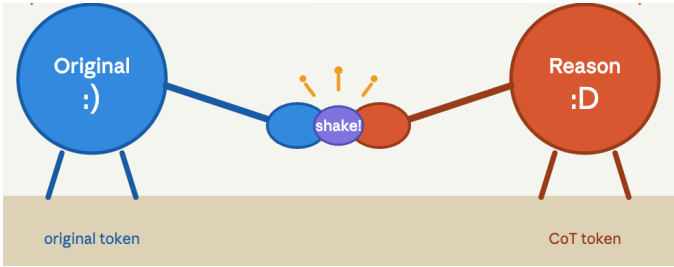
Crossblocks V/S Bottom right blocks

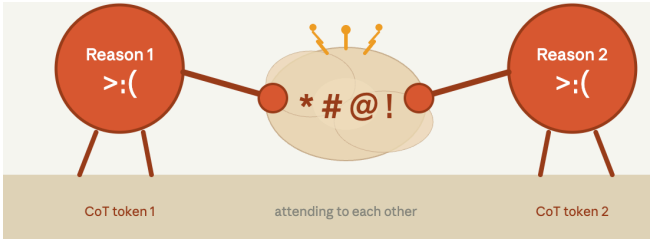
- What are these blocks?
  (n+k)^2 = n^2 (top left) + 2nk (cross) + k^2 (bottom)

- **Cross Blocks:**

  This is where COT applies the reasoning to the original tokens.
  Original tokens attending to CoT tokens AND CoT tokens attending to original. Here, reasoning steps can refer back to the question.
- **Bottom Blocks:**

  CoT tokens attending to each other. Reasoning steps build on each other.



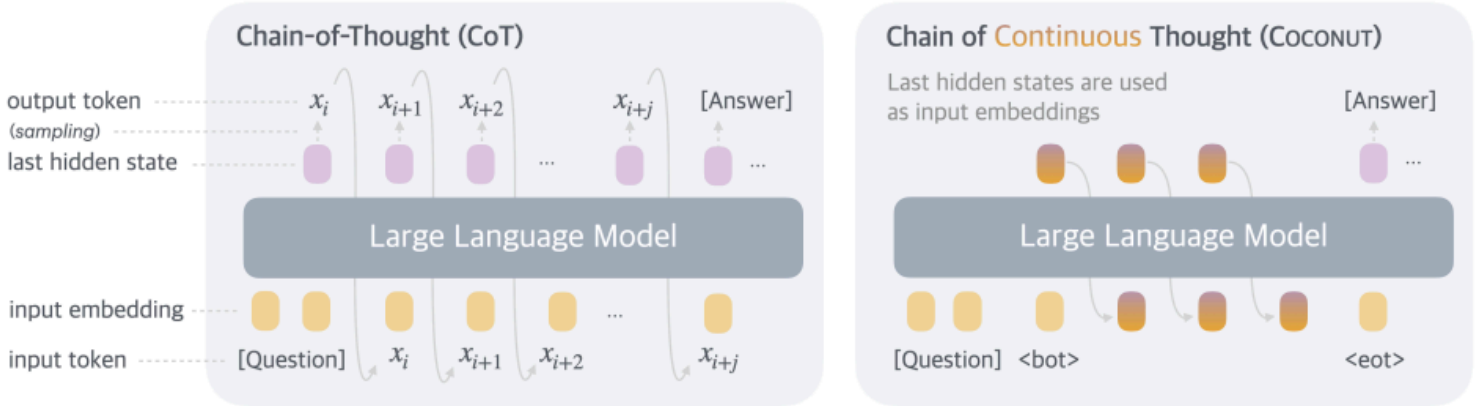


What if reasoning happened in hidden layers instead of output tokens?

Could we train a model to do the same "chain of thought" internally, without generating any intermediate text at all?

What if reasoning happened in hidden layers instead of output tokens?
Could we train a model to do the same "chain of thought" internally, in latent space, without generating any intermediate text at all?

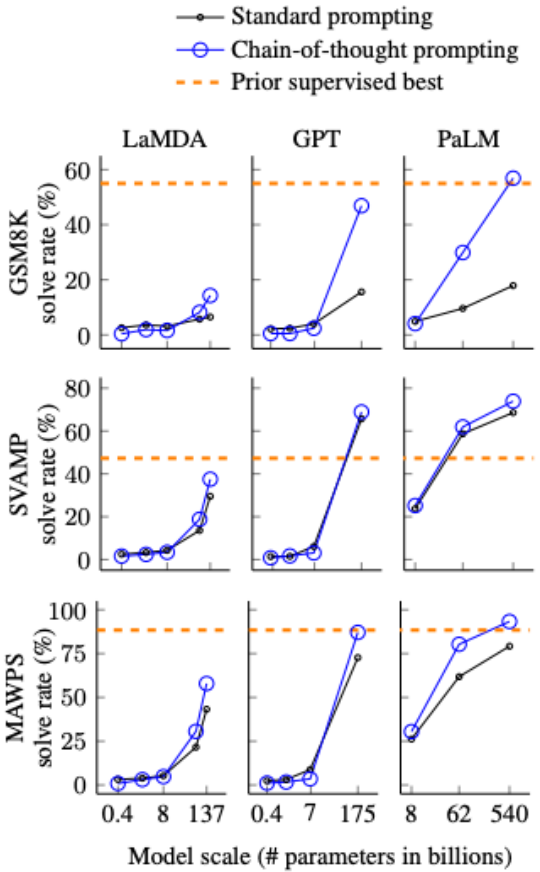
Yes ,and this exact idea has been built and tested. It's called **Coconut** (Chain of Continuous Thought).
Instead of forcing reasoning into the token stream as text, Coconut uses the last hidden state of the LLM as a representation of the reasoning state, called a "continuous thought." Rather than decoding this state into words, it feeds it directly back to the model as the next input embedding in the continuous space.



Results

A) Greater gains on harder problems :
The more complex the task, the more COT helps.
On GSM8K (the hardest benchmark), performance more than doubled for the largest GPT and PaLM models, while on simple single step problems, improvements were minimal or even negative.

B) Competitive with supervised fine tuning :
PaLM 540B with COT prompting achieved good results on GSM8K, SVAMP, and MAWPS . It matched or surpassed models that were specifically fine tuned on labeled training data, and coming within 2% of the best on AQuA and ASDiv.



Limitations

1.46% of incorrect answers had "almost correct" CoTs. 54% had major semantic errors. Factual generation remains an open problem.

2.Emergence only at ≥100B parameters makes CoT costly to deploy in production. How to elicit reasoning in smaller models is open.

3. Compute

Strengths

1.No fine tuning required .CoT works purely through prompting. A single model checkpoint can handle many tasks without any gradient updates or task specific training.

2. Works across arithmetic, commonsense, and symbolic reasoning. These are the areas where human use language for.

3.Interpretable reasoning : The chain of thought provides a visible window into model behavior, making it easier to debug where reasoning goes wrong.

UC San Diego

Q&A

UC San Diego

BACK UP

References

1. Wei et al. (2022) , Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
2. Brown et al. (2020) , Language Models are Few-Shot Learners (GPT-3)
3. Vaswani et al. (2017) , Attention Is All You Need https
4. Xie et al. (2022) , An Explanation of In-Context Learning as Implicit Bayesian Inference
5. Tishby & Zaslavsky (2015) , Deep Learning and the Information Bottleneck Principle
6. Hao et al. (2024) , Training Large Language Models to Reason in a Continuous Latent Space (Coconut)