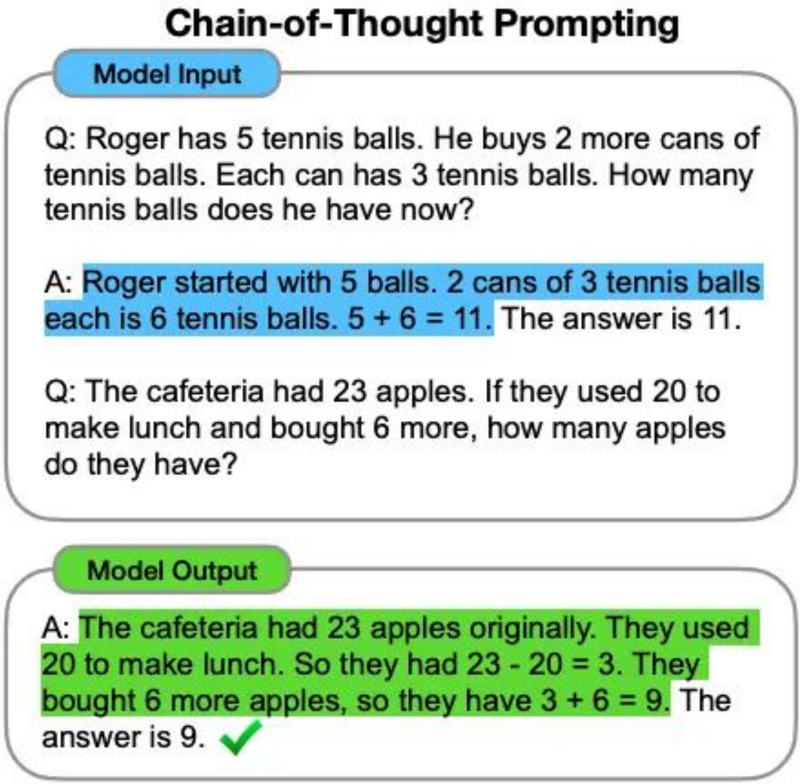
ETH zürich

Training LLM to Reason in a Continuous Latent Space

Menelik Nouvellon

some background & motivation: Why Continuous (Latent) Reasoning?

Chain-of-Thought (CoT) Reasoning: What & Why



[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. (2022)
https://doi.org/10.48550/arXiv.2201.11903

Recent examples of CoT (extension of CoT)

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

research@deepseek.com

Abstract

We introduce our first-generation reasoning models, DeepSeek-R1-Zero and DeepSeek-R1. DeepSeek-R1-Zero, a model trained via large-scale reinforcement learning (RL) without supervised fine-tuning (SFT) as a preliminary step, demonstrates remarkable reasoning capabilities. Through RL, DeepSeek-R1-Zero naturally emerges with numerous powerful and intriguing reasoning behaviors. However, it encounters challenges such as poor readability, and language mixing. To address these issues and further enhance reasoning performance, we introduce DeepSeek-R1, which incorporates multi-stage training and cold-start data before RL. DeepSeek-R1 achieves performance comparable to OpenAI-o1-1217 on reasoning tasks. To support the research community, we open-source DeepSeek-R1-Zero, DeepSeek-R1, and six dense models (1.5B, 7B, 8B, 14B, 32B, 70B) distilled from DeepSeek-R1 based on Qwen and Llama.

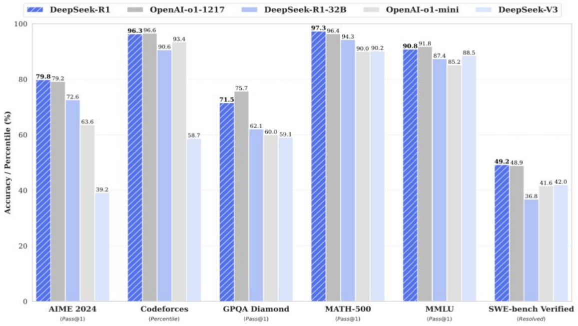
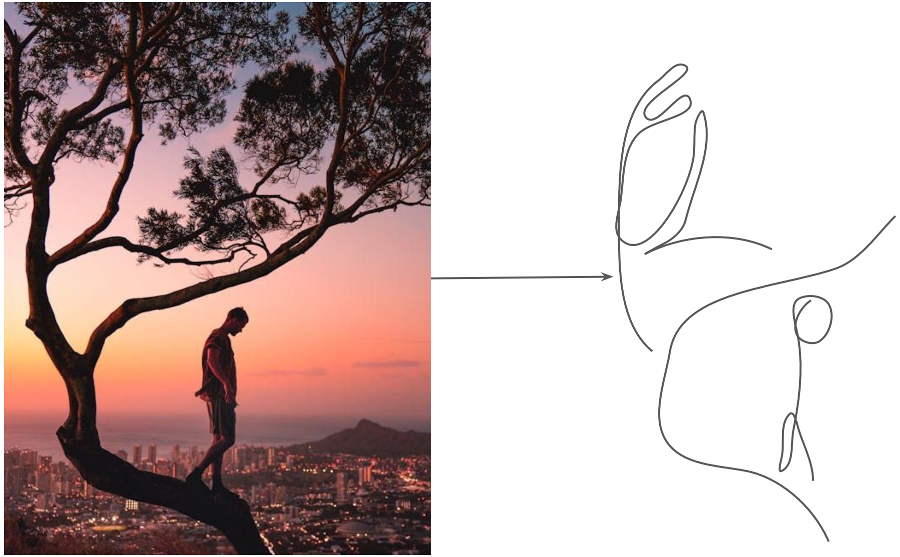


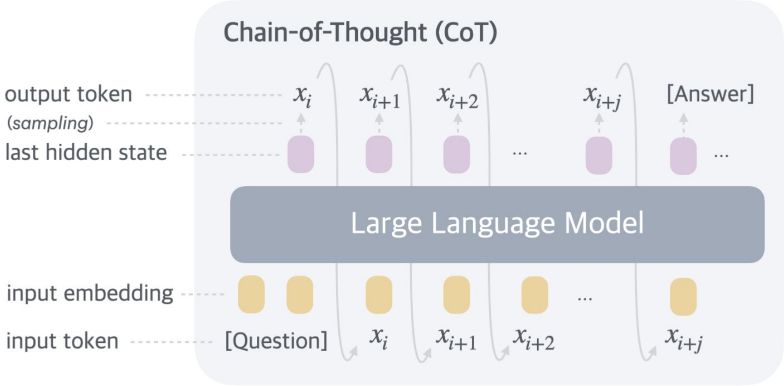
Figure 1 | Benchmark performance of DeepSeek-R1.

[2] DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning (2025)
https://doi.org/10.48550/arXiv.2501.12948

So why do we need latent continuous reasoning ?

Motivation #1: Information Loss in Text Decoding





Motivation #2: Avoiding Filler & Overhead

Question:

"The cafeteria had 23 apples. They used 20 to make lunch, then bought 6 more. How many apples do they have now?"

Reasoning:

1) The cafeteria started with 23 apples.
2) They used 20 apples for lunch, leaving 3 apples.
3) Next, they bought 6 more apples, so 3 + 6 = 9.

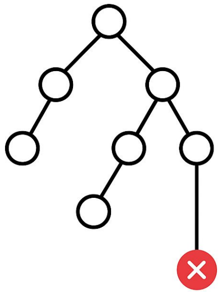
Hence, the final answer is 9.

filler phrases

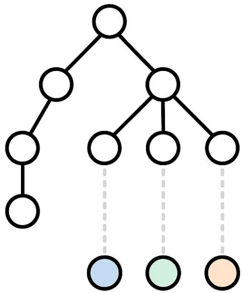
Core math: All that's *really* needed is: "23 - 20 = 3; 3 + 6 = 9."

Motivation #3: Parallel Exploration & Backprop

Text: Early Commitment

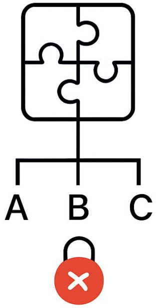


Continuous: Parallel Exploration

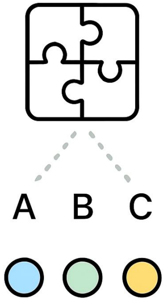


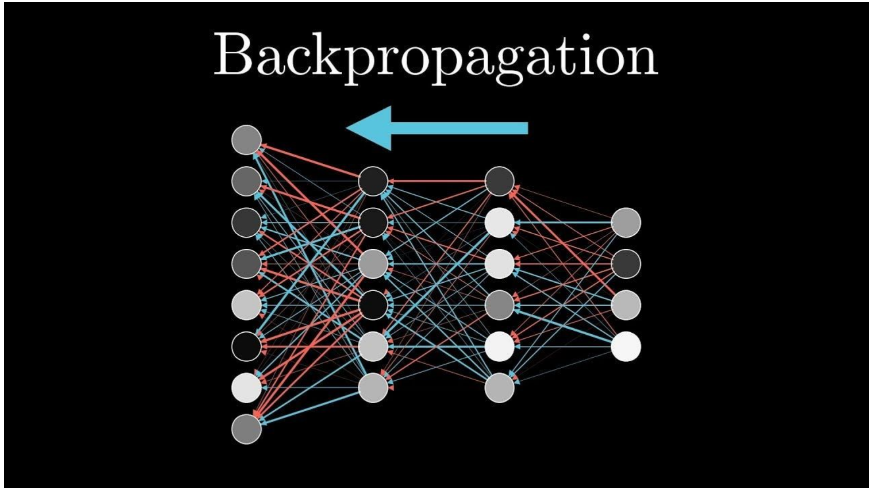
Example :

Text: Early Commitment

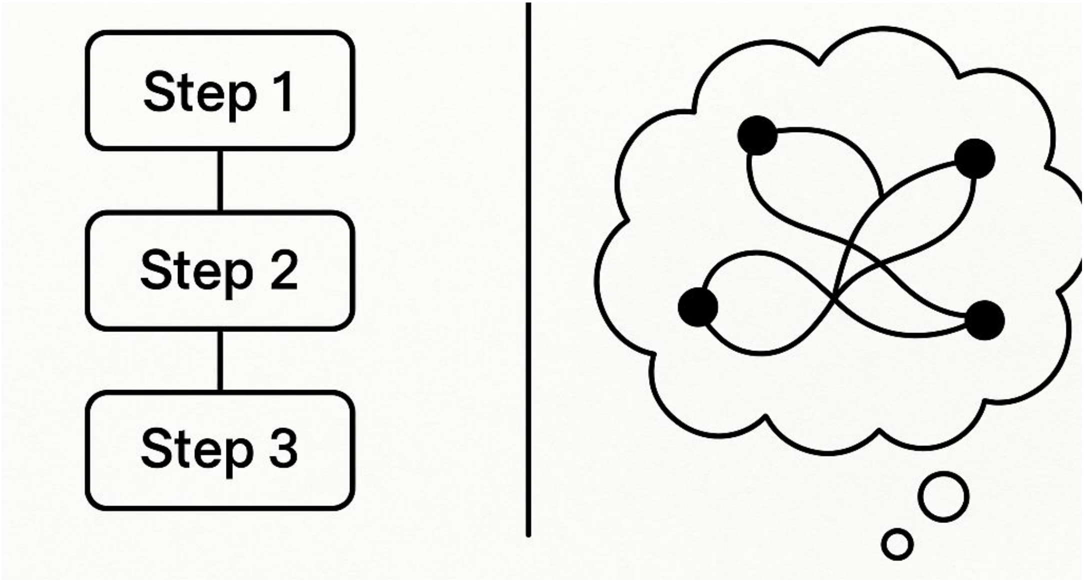


Continuous: Parallel





Motivation #4: Better Generalization



Textual reasoning **Continuous reasoning**

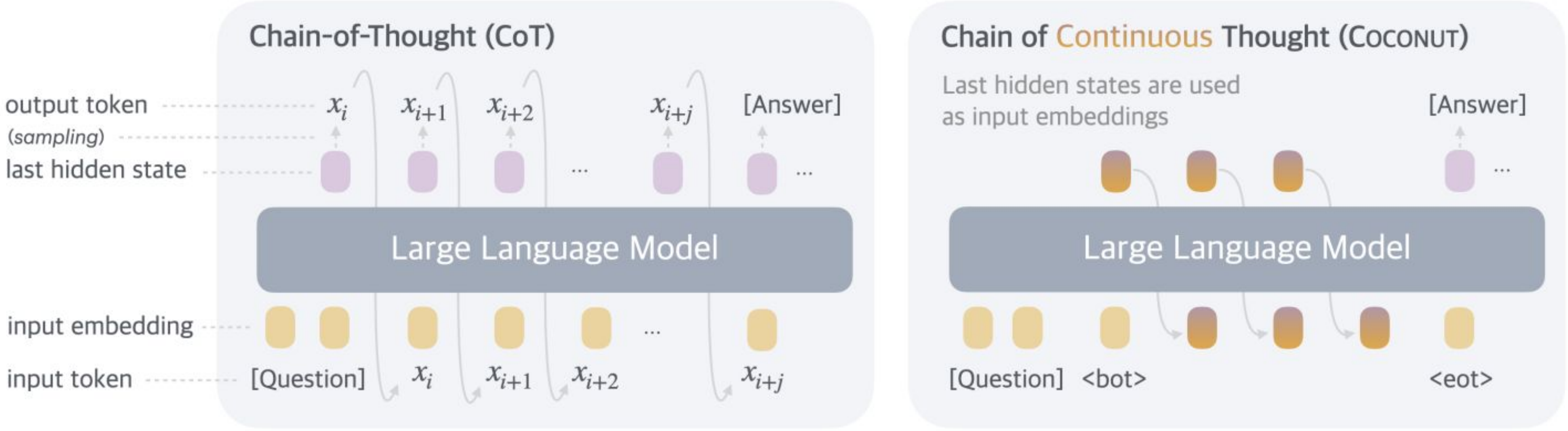
1. **Text CoT** is good but has overhead, can be rigid.

2. **Latent Reasoning** might preserve richer signals, allow branching, skip filler.

3. **End-to-End** optimization could push better multi-step performance.

Now let’s dive into the paper

Coconut: “**C**hain **o**f **Con**tin**u**ous **T**hought”



Inference in Coconut: Big Picture

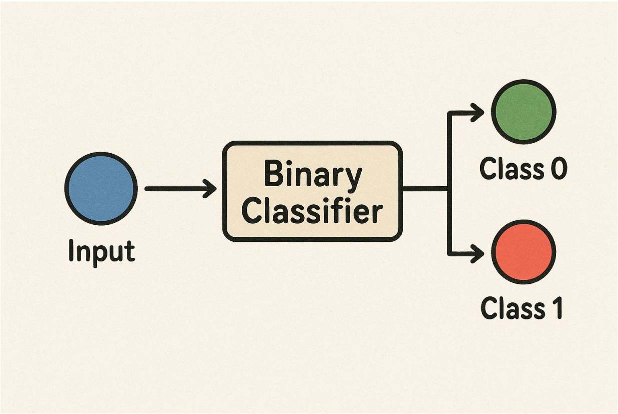
signal to jump to latent mode

[Latent Steps]

signal to output answer

Where do we put the <eot> ?

aka ***Where should the latent reasoning stop ?***



option 1

⇔

option 2

Step-by-Step Example

Biff the Bear buys 3 honey pots.
Each honey pot costs 5 honey coins.

Question: “How many honey coins does Biff pay in total?”

Normal CoT inference

Reasoning:
1) Biff buys 3 pots, each costs 5 honey coins.
2) Multiply 3 by 5, that equals 15.
Therefore, Biff must pay 15 honey coins total.

Coconut Inference

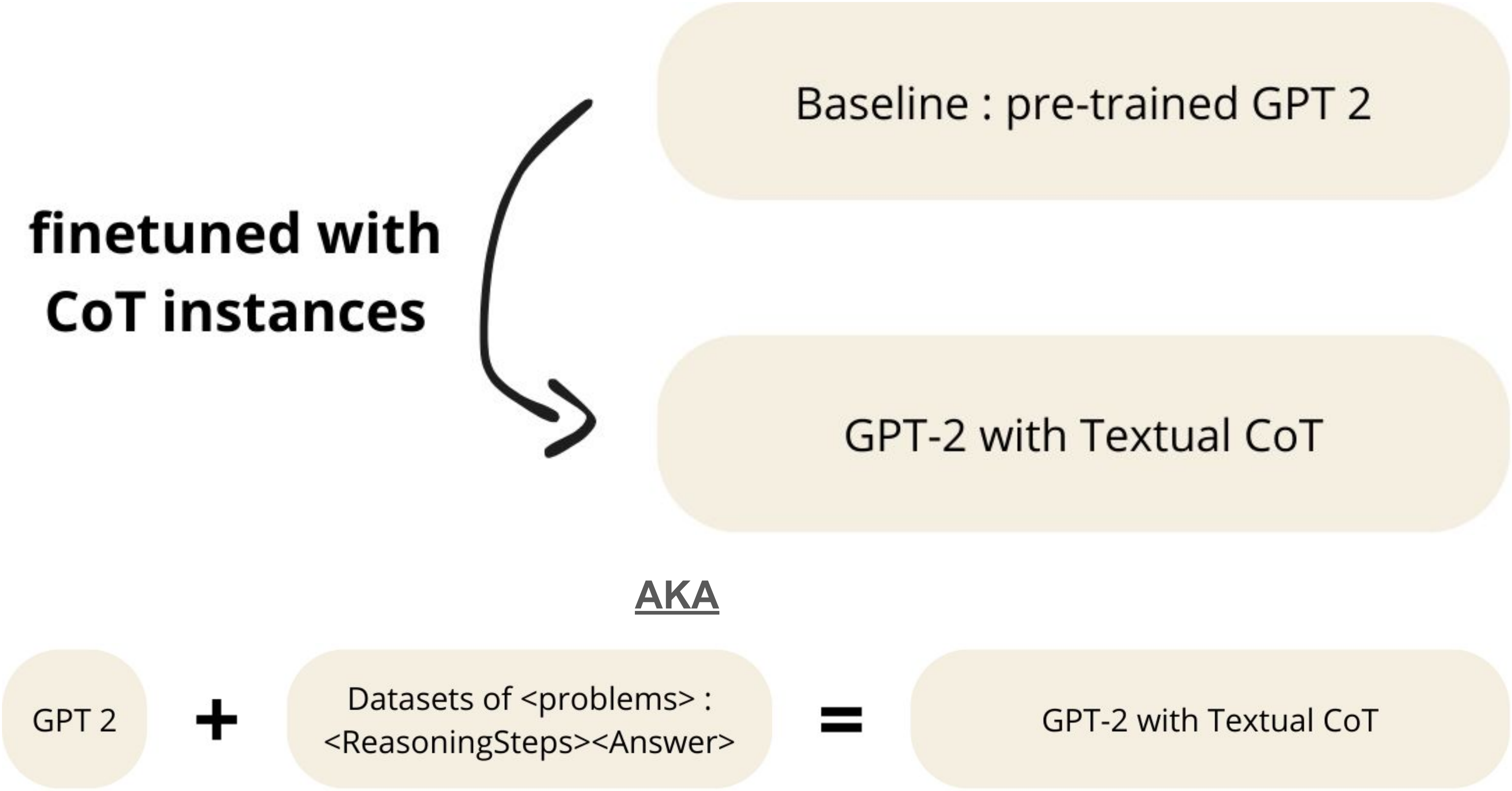
<bot> [... latent steps ...] <eot>
Final Answer: 15 honey coins

Now let’s look at the training process

Baseline : pre-trained GPT 2

finetuned with CoT instances

GPT-2 with Textual CoT


finetuned with
CoT instances
Baseline : pre-trained GPT 2
GPT-2 with Textual CoT
AKA
GPT 2
+
Datasets of <problems> :
<ReasoningSteps><Answer>
=
GPT-2 with Textual CoT

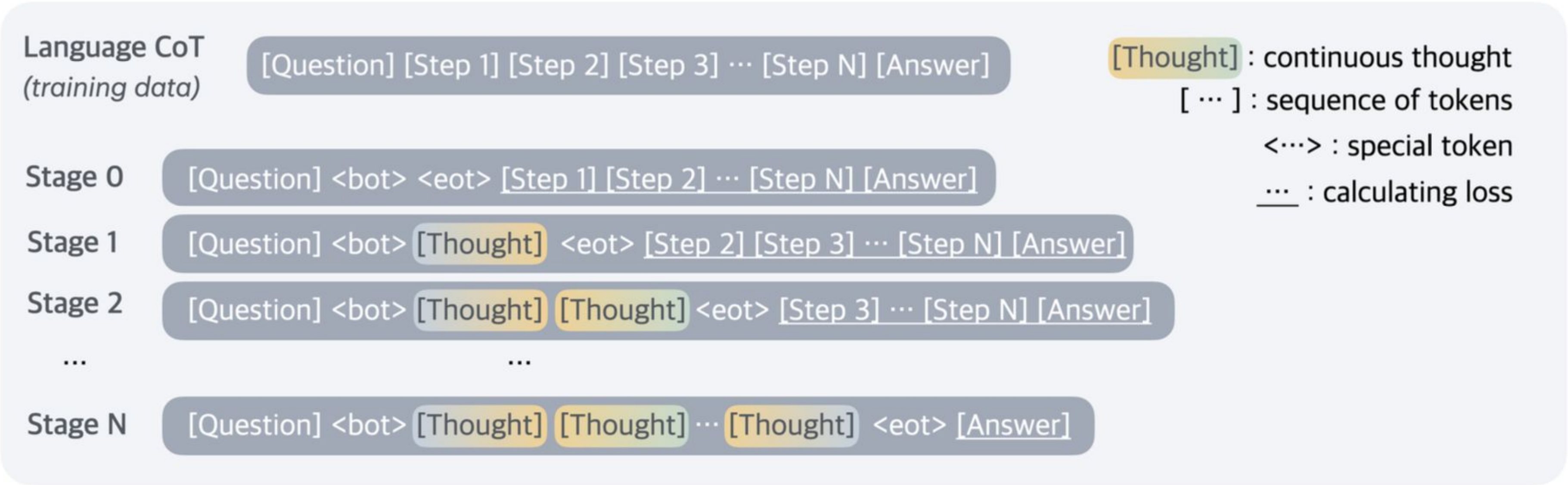
Baseline : pre-trained GPT 2

finetuned with CoT instances

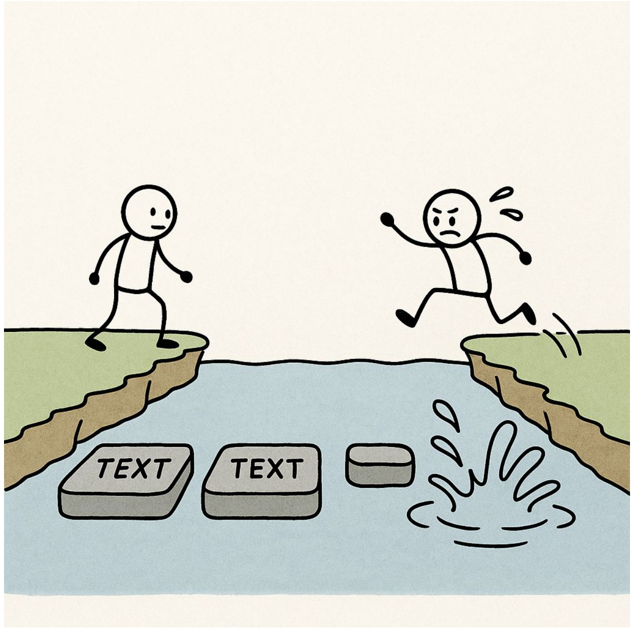
GPT-2 with Textual CoT

?

GPT2 - Coconut


Language CoT
(training data)
[Question] [Step 1] [Step 2] [Step 3] ··· [Step N] [Answer]
[Thought] : continuous thought
[···] : sequence of tokens
<···> : special token
··· : calculating loss
Stage 0
[Question] <bot> <eot> [Step 1] [Step 2] ··· [Step N] [Answer]
Stage 1
[Question] <bot> [Thought] <eot> [Step 2] [Step 3] ··· [Step N] [Answer]
Stage 2
[Question] <bot> [Thought] [Thought] <eot> [Step 3] ··· [Step N] [Answer]
...
...
Stage N
[Question] <bot> [Thought] [Thought] ··· [Thought] <eot> [Answer]

Why Not Jump Immediately to All Latent Steps?



Why do use multi-stage curriculum ?

Method	GSM8k		ProntoQA		ProsQA	
	Acc. (%)	# Tokens	Acc. (%)	# Tokens	Acc. (%)	# Tokens
COCONUT (Ours)	34.1 ±1.5	8.2	99.8 ±0.2	9.0	97.0 ±0.3	14.2
- *w/o curriculum*	14.4 ±0.8	8.2	52.4 ±0.4	9.0	76.1 ±0.2	14.2

The LLM still needs guidance to learn latent reasoning

1. **Inference:** hidden steps, all in continuous latent space

2. **Training:** multi-stage replacement of textual steps -- *this is called multi-stage curriculum*

3. **Architecture:** special tokens <bot> and <eot> mark the boundaries of latent reasoning,

What about their result & experiment ?

Experimental Setup

Math

GSM8k : dataset of 8.5K grade school math word problems created by human problem writers

<problem> : <ReasoningSteps><finalAnswer>

Experimental Setup

Math

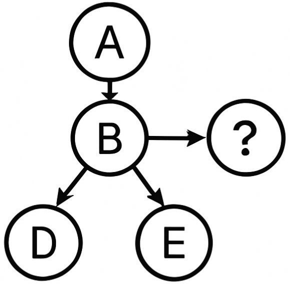
<problem> : <ReasoningSteps><finalAnswer>

Logical Reasoning

ProntoQA

simple chain

ProsQA



logic graph

ProntoQA : "Stella is a zumpus. Zumpuses are gorpuses... Is Stella floral?"
ProsQA : "Tom is a terpus. Every terpus is a brimpus..."

Now the match that everyone have been waiting for…

🥊 CoT vs Coconut 🥊

Method	GSM8k		ProntoQA		ProsQA	
	Acc. (%)	# Tokens	Acc. (%)	# Tokens	Acc. (%)	# Tokens
CoT	42.9 ±0.2	25.0	98.8 ±0.8	92.5	77.5 ±1.9	49.4
COCONUT (Ours)	34.1 ±1.5	8.2	99.8 ±0.2	9.0	97.0 ±0.3	14.2

Math

Method	GSM8k		ProntoQA		ProsQA	
	Acc. (%)	# Tokens	Acc. (%)	# Tokens	Acc. (%)	# Tokens
CoT	42.9 $\pm$0.2	25.0	98.8 $\pm$0.8	92.5	77.5 $\pm$1.9	49.4
Coconut (Ours)	34.1 $\pm$1.5	8.2	99.8 $\pm$0.2	9.0	97.0 $\pm$0.3	14.2

CoT leads on accuracy

But Coconut seems promising
with 4x less tokens

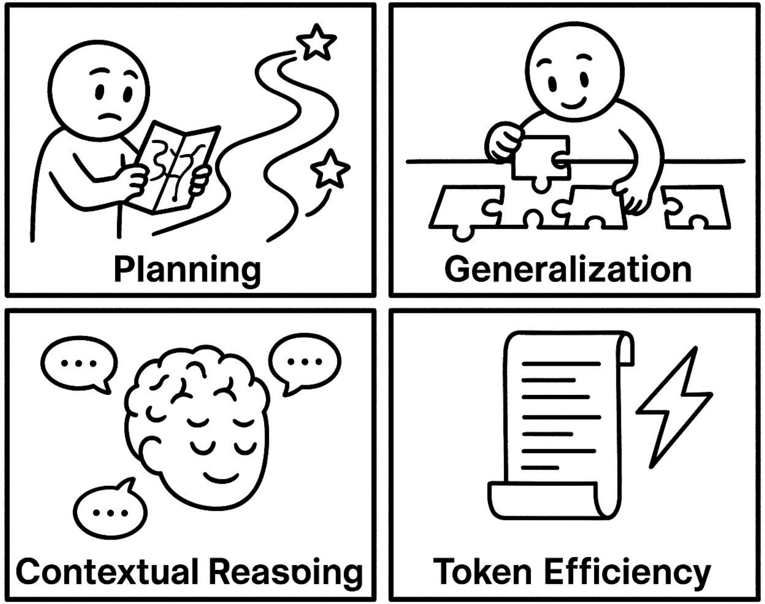
Logical Reasoning

Method	GSM8k		ProntoQA		ProsQA	
	Acc. (%)	# Tokens	Acc. (%)	# Tokens	Acc. (%)	# Tokens
CoT	42.9 ±0.2	25.0	98.8 ±0.8	92.5	77.5 ±1.9	49.4
COCONUT (Ours)	34.1 ±1.5	8.2	99.8 ±0.2	9.0	97.0 ±0.3	14.2

CoT

Coconut matches & outperform CoT
19.5% accuracy improvement while using 71% fewer tokens

Method	GSM8k		ProntoQA		ProsQA	
	Acc. (%)	# Tokens	Acc. (%)	# Tokens	Acc. (%)	# Tokens
CoT	42.9 $\pm$0.2	25.0	98.8 $\pm$0.8	92.5	77.5 $\pm$1.9	49.4
COCONUT (Ours)	34.1 $\pm$1.5	8.2	99.8 $\pm$0.2	9.0	97.0 $\pm$0.3	14.2



Now let's look at how Coconut compares to other reasoning approaches beyond just CoT...

iCoT

		Input	CoT	Output
Explicit CoT	Stage 0:	2 1 × 4 3 =	8 4 + 0 6 3	= 8 0 4
	Stage 1:	2 1 × 4 3 =	4 + 0 6 3	= 8 0 4
	Stage 2:	2 1 × 4 3 =	+ 0 6 3	= 8 0 4
	Stage 3:	2 1 × 4 3 =	0 6 3	= 8 0 4
	Stage 4:	2 1 × 4 3 =	6 3	= 8 0 4
	Stage 5:	2 1 × 4 3 =	3	= 8 0 4
Implicit CoT	Stage 6:	2 1 × 4 3 =		= 8 0 4

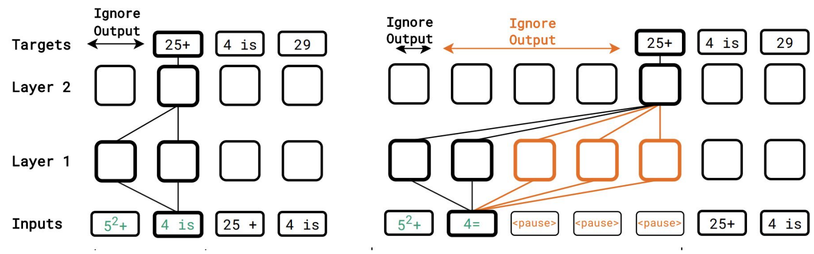
[3] From Explicit CoT to Implicit CoT: Learning to Internalize CoT Step by Step (2024) https://doi.org/10.48550/arXiv.2405.14838

iCoT

		Input	CoT	Output
Explicit CoT	Stage 0:	2 1 × 4 3 =	8 4 + 0 6 3	= 8 0 4
	Stage 1:	2 1 × 4 3 =	4 + 0 6 3	= 8 0 4
	Stage 2:	2 1 × 4 3 =	+ 0 6 3	= 8 0 4
	Stage 3:	2 1 × 4 3 =	0 6 3	= 8 0 4
	Stage 4:	2 1 × 4 3 =	6 3	= 8 0 4
	Stage 5:	2 1 × 4 3 =	3	= 8 0 4
Implicit CoT	Stage 6:	2 1 × 4 3 =		= 8 0 4

[3] From Explicit CoT to Implicit CoT: Learning to Internalize CoT Step by Step (2024) https://doi.org/10.48550/arXiv.2405.14838

Pause Token



(a) Standard inference and finetuning (b) Pause-inference and finetuning

[4] Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. (2023) https://doi.org/10.48550/arXiv.2310.02226

Method	GSM8k	
	Acc. (%)	# Tokens
iCoT	30.0*	2.2
Pause Token	16.4 ±1.8	2.2
Coconut (Ours)	34.1 ±1.5	8.2

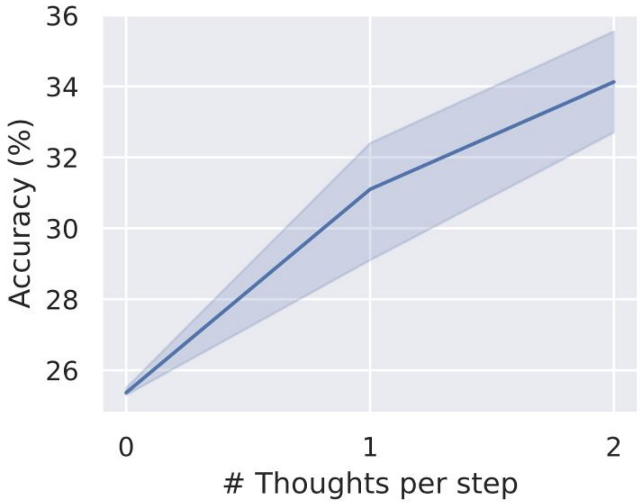


Figure 3 Accuracy on GSM8k with different number of continuous thoughts.

<u>“Chaining” continuous thoughts enhances reasoning</u>

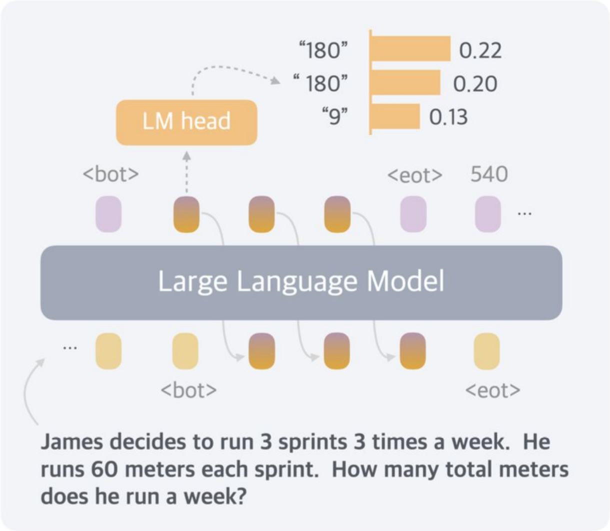


Figure 4 A case study where we decode the continuous thought into language tokens.

Continuous thoughts are efficient representations of reasoning

No-CoT

it's like CoT but no CoT

trained on GSM8k

~~<problem> → <ReasoningSteps><finalAnswer>~~

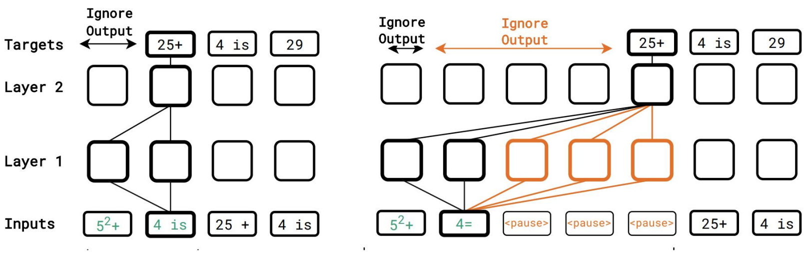
<problem> → <finalAnswer>

iCoT

		Input	CoT	Output
Explicit CoT	Stage 0:	2 1 × 4 3 =	8 4 + 0 6 3	= 8 0 4
	Stage 1:	2 1 × 4 3 =	4 + 0 6 3	= 8 0 4
	Stage 2:	2 1 × 4 3 =	+ 0 6 3	= 8 0 4
	Stage 3:	2 1 × 4 3 =	0 6 3	= 8 0 4
	Stage 4:	2 1 × 4 3 =	6 3	= 8 0 4
	Stage 5:	2 1 × 4 3 =	3	= 8 0 4
Implicit CoT	Stage 6:	2 1 × 4 3 =		= 8 0 4

[3] From Explicit CoT to Implicit CoT: Learning to Internalize CoT Step by Step (2024) https://doi.org/10.48550/arXiv.2405.14838

Pause Token

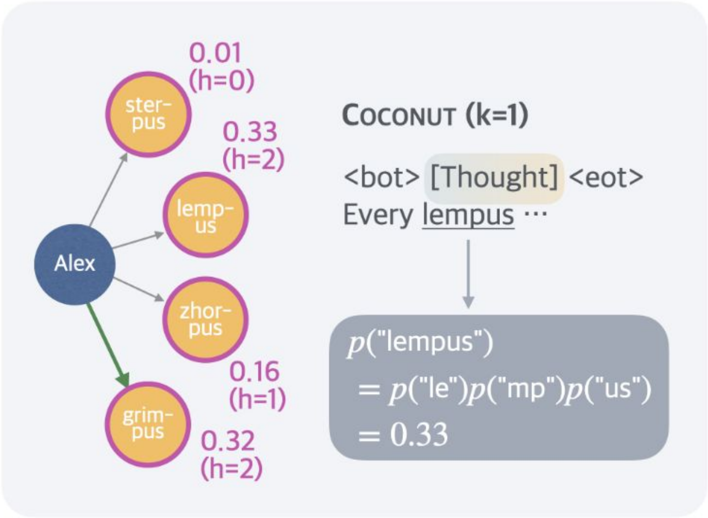


(a) Standard inference and finetuning (b) Pause-inference and finetuning

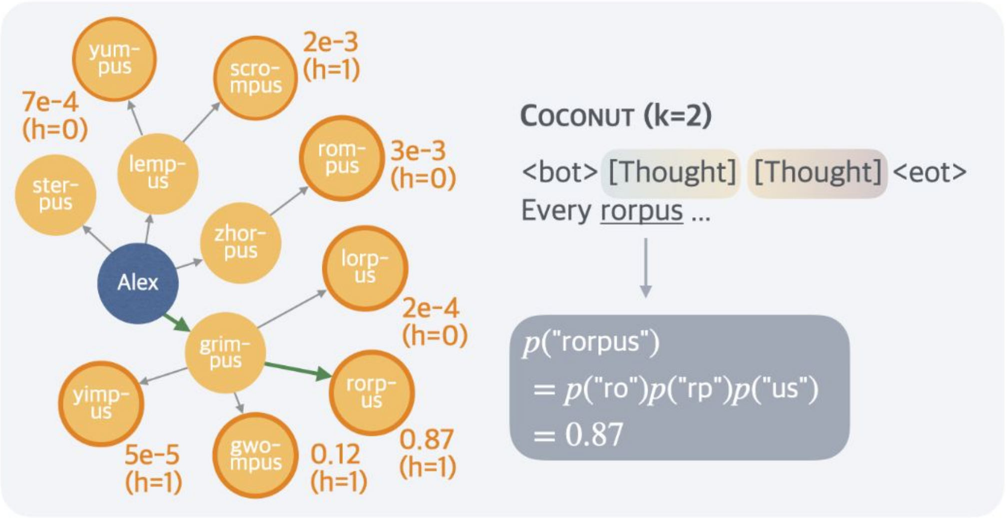
[4] Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. (2023) https://doi.org/10.48550/arXiv.2310.02226

Method	ProsQA	
	Acc. (%)	# Tokens
CoT	77.5 ±1.9	49.4
No-CoT	76.7 ±1.0	8.2
iCoT	98.2 ±0.3	8.2
Pause Token	75.9 ±0.7	8.2
COCONUT (Ours)	97.0 ±0.3	14.2
- *w/o curriculum*	76.1 ±0.2	14.2
- *w/o thought*	95.5 ±1.1	8.2
- *pause as thought*	96.6 ±0.8	8.2

Latent reasoning outperforms language reasoning in planning-intensive tasks



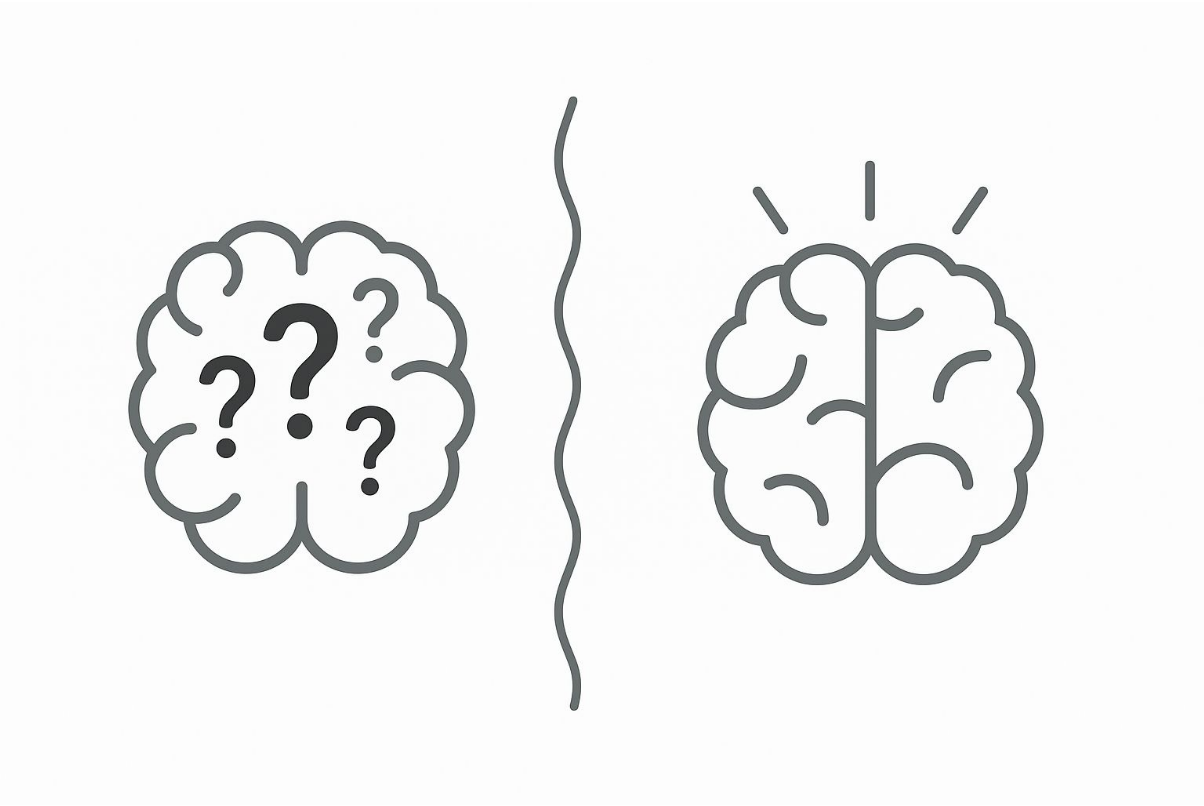
Multiple concepts have significant probability



clear convergence

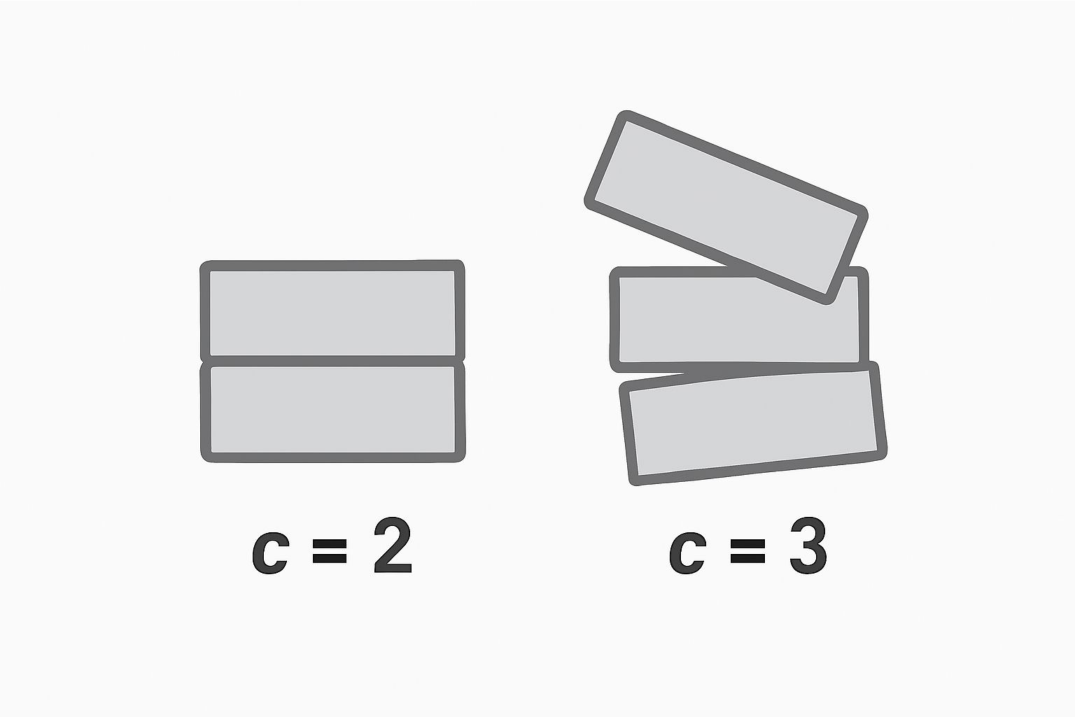
Latent reasoning outperforms language reasoning in planning-intensive tasks

Interpretability Trade-off



is there any other limitations ?

Training stability issue



Possible Extensions

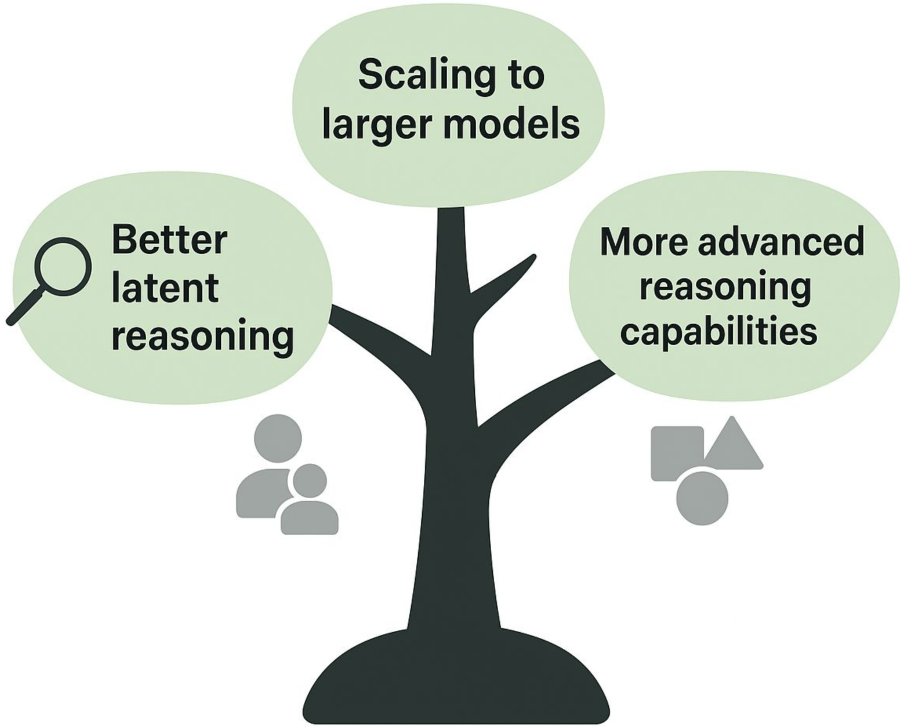
Pretraining with continuous thoughts

- Current approach relies on finetuning
- Could continuous thoughts be part of pretraining?
- Potential for more generalizable reasoning abilities

Hybrid approaches

- Combining language and latent reasoning
- "Generating the reasoning skeleton in language"
- "Completing the reasoning process in latent space"

Future Research Directions



Concluding Thoughts

- Key Takeaways
  - Coconut enables reasoning in continuous latent space
  - Shows emergent BFS-like search behavior
  - Improves efficiency (71% fewer tokens) while maintaining accuracy
- Critics
  - Still requires language supervision during training

Thank you

Q&A