주간 세미나

2026. 03. 05.

윤정호

Intro

***현 Reasoning의 문제점**

- 긴 Reasoning Path – 비용 병목
  - 문장부호, 조사 등과 같이 큰 의미를 가지지 않은 토큰이 추론 과정에 반복적으로 포함되어 출력 길이를 늘림
  - 모든 토큰이 동일한 compute budget을 사용하여 과한 비용 투입

- 단일 경로 선택 – 탐색 병목
  - 기존 CoT는 한 스텝에 한 토큰을 선택하여 경로 고착으로 이어지기 쉬움

- 불 연속적으로 정의된 토큰 – 표현 병목
  - 추상적 개념을 표현하고, 조작하기 어려움
  - 언어 공간이 최적의 추론 공간이 아닐 수 있음

선행 연구

Training Large Language Models to Reason in a Continuous Latent Space

Shibo Hao[1,2,*], Sainbayar Sukhbaatar[1], DiJia Su[1], Xian Li[1], Zhiting Hu[2], Jason Weston[1], Yuandong Tian[1]
[1]FAIR at Meta, [2]UC San Diego
[*]Work done at Meta

COLM 2025

Soft Thinking: Unlocking the Reasoning Potential of LLMs in Continuous Concept Space

Zhen Zhang[1*] Xuehai He[2*] Weixiang Yan[1]
Ao Shen[4] Chenyang Zhao[3,5] Xin Eric Wang[1]
[1]University of California, Santa Barbara, [2]University of California, Santa Cruz
[3]University of California, Los Angeles, [4]Purdue University, [5]LMSYS Org
zhen_zhang@ucsb.edu, ericxwang@ucsb.edu

NeurIPS 2025 Poster

Coconut

***Abstract & Introduction**

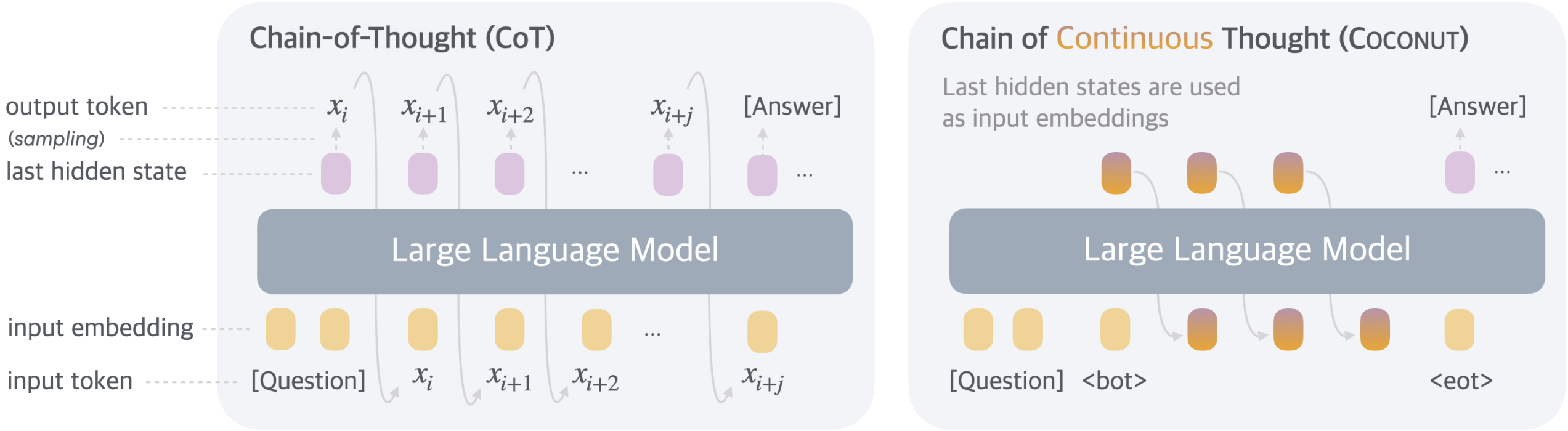
- LLM의 명시적 추론 과정은 토큰 단위로 생성되는데 이는 인간의 추론 방식과 다름

⇒ 사람은 추론 시 언어 이해 및 생성을 담당하는 뇌 영역이 활성화 되지 않음
⇒ 인간의 언어는 추론보다 의사소통에 최적화 됨

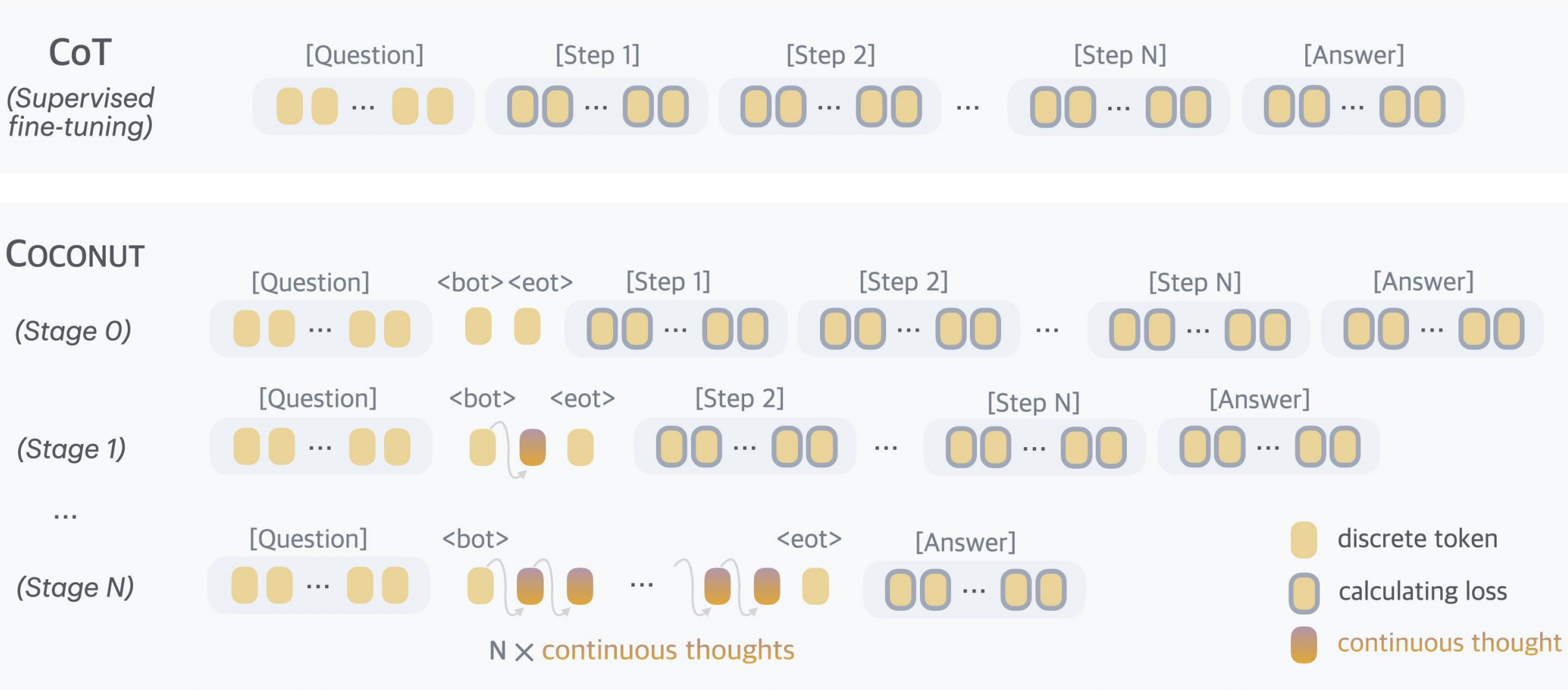
- 추론 시 사용하는 대부분의 토큰은 유창성을 위해 사용되며 중요한 토큰과 중요하지 않은 토큰의 연산 과정이 같음

⇒ 언어 공간의 제약에 갇혀 근본적 문제 해결 못함
⇒ 언어 제약 없이 자유롭게 추론하고 필요할 때만 언어로 생성하게 하자

METHODOLOGY



METHODOLOGY



EXPERIMENTS

- Training
  - GPT-2 : stage 마다 epoch 5, 마지막 stage epoch 50
  - data : ProsQA (Proof with Search Question-Answering) 직접 제작한 논리 추론 데이터, GSM8K, ProntoQA

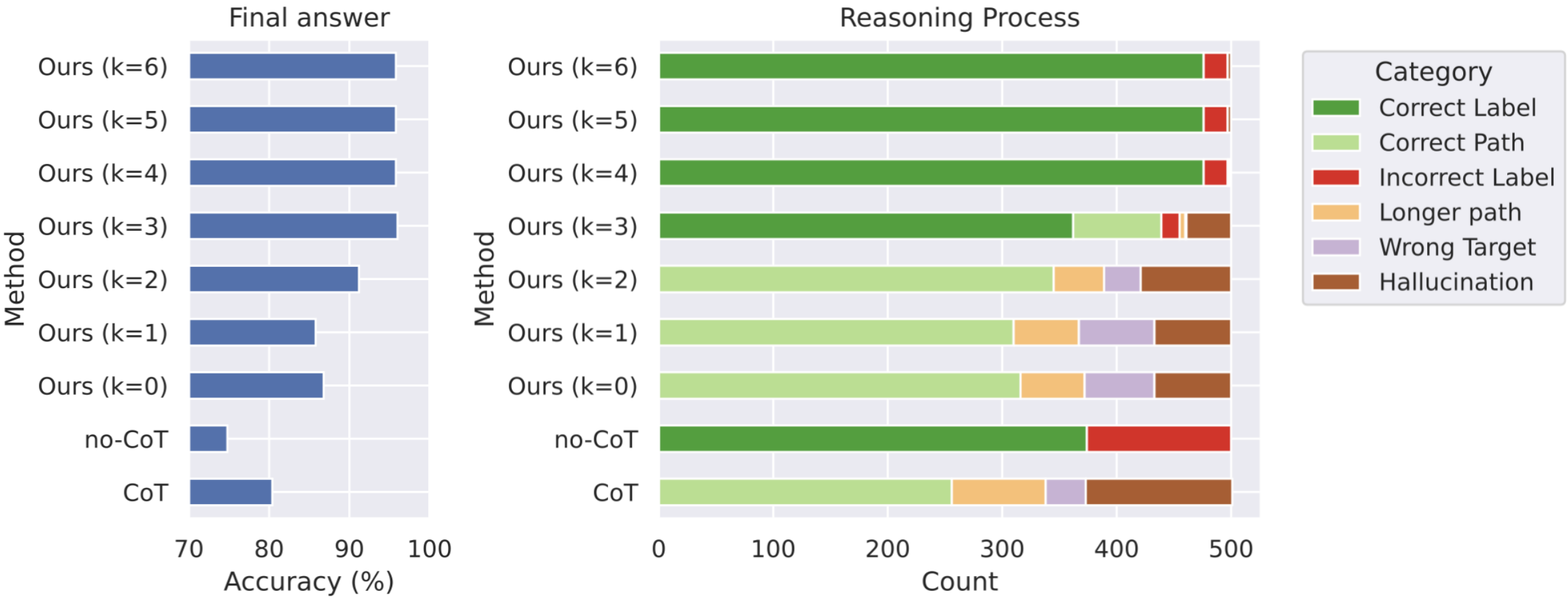
- Benchmark
  - 수학 : GSM8K
  - 논리 추론 : ProntoQA, ProsQA

- Baseline
  - CoT, No-CoT

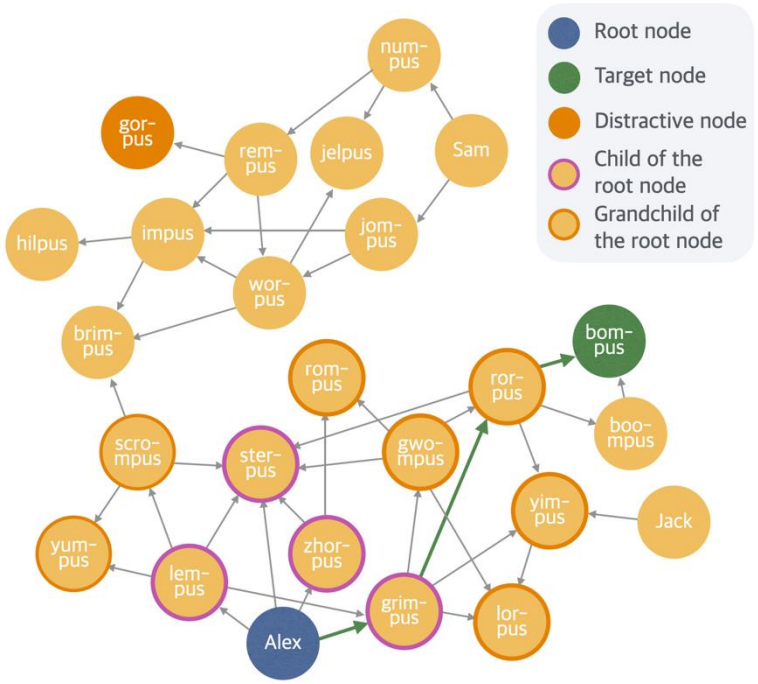
ProsQA

Question = "Every shumpus is a rempus. Every shumpus is a yimpus. Every terpus is a fompus. Every terpus is a gerpus. Every gerpus is a brimpus. Alex is a rempus. Every rorpus is a scrompus. Every rorpus is a yimpus. Every terpus is a brimpus. Every brimpus is a lempus. Tom is a terpus. Every shumpus is a timpus. Every yimpus is a boompus. Davis is a shumpus. Every gerpus is a lorpus. Davis is a fompus. Every shumpus is a boompus. Every shumpus is a rorpus. Every terpus is a lorpus. Every boompus is a timpus. Every fompus is a yerpus. Tom is a dumpus. Every rempus is a rorpus. Is Tom a lempus or scrompus?"
Steps = ["Tom is a terpus.", "Every terpus is a brimpus.", "Every brimpus is a lempus."]
Answer = "Tom is a lempus."

EXPERIMENTS



EXPERIMENTS



Question:

Every grimpus is a yimpus. Every worpus is a jelpus. Every zhorpus is a sterpus. Alex is a grimpus ⋯ Every lumps is a yumpus.
Question: **Is Alex a gorpus or bompus?**

Ground Truth Solution

Alex is a grimpus.
Every grimpus is a rorpus.
Every rorpus is a bompus.
Alex is a bompus

CoT

Alex is a lempus.
Every lempus is a scrompus.
Every scrompus is a yumpus.
Every yumpus is a rempus.
Every rempus is a gorpus.
Alex is a gorpus ✗

(Hallucination)

COCONUT (k=1)

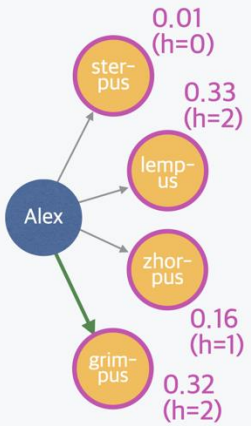
<bot> <eot>
Every lempus is a scrompus.
Every scrompus is a brimpus.
Alex is a brimpus ✗

(Wrong Target)

COCONUT (k=2)

<bot> <eot>
Every rorpus is a bompus.
Alex is a bompus ✓

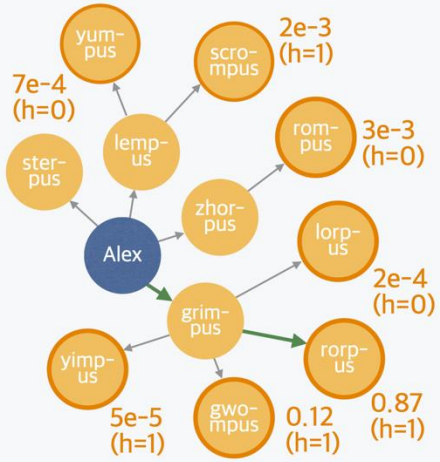
(Correct Path)



COCONUT (k=1)

<bot> <eot>
Every lempus ⋯

$p(\text{"lempus"}) = p(\text{"le"})p(\text{"mp"})p(\text{"us"}) = 0.33$



COCONUT (k=2)

<bot> <eot>
Every rorpus ...

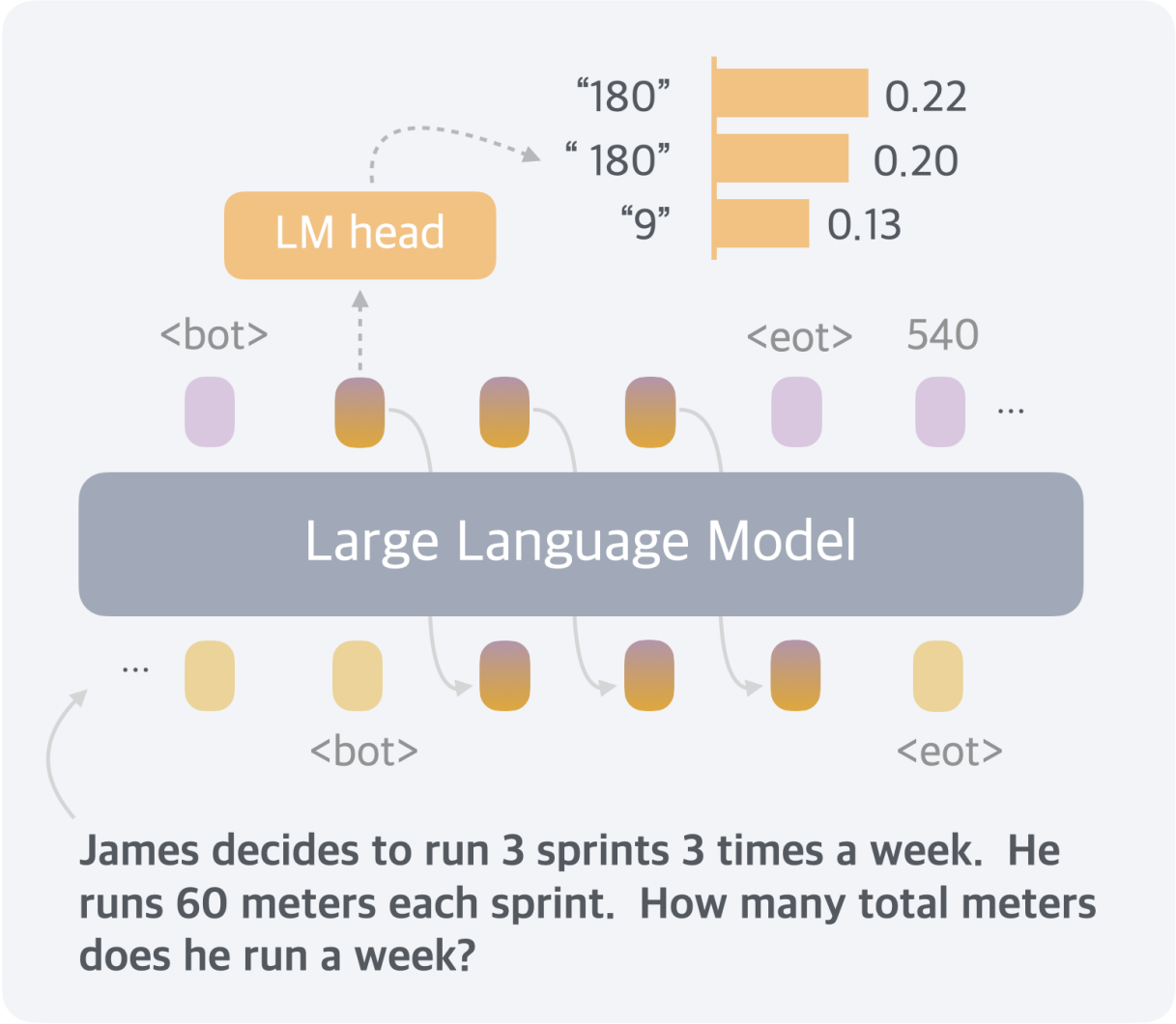
$p(\text{"rorpus"}) = p(\text{"ro"})p(\text{"rp"})p(\text{"us"}) = 0.87$

EXPERIMENTS

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
No-CoT	16.5 ±0.5	2.2	93.8 ±0.7	3.0	76.7 ±1.0	8.2
iCoT	30.0*	2.2	99.8 ±0.3	3.0	98.2 ±0.3	8.2
Pause Token	16.4 ±1.8	2.2	77.7 ±21.0	3.0	75.9 ±0.7	8.2
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
- *w/o thought*	21.6 ±0.5	2.3	99.9 ±0.1	3.0	95.5 ±1.1	8.2
- *pause as thought*	24.1 ±0.7	2.2	100.0 ±0.1	3.0	96.6 ±0.8	8.2

Table 1 Results on three datasets: GSM8k, ProntoQA and ProsQA. Higher accuracy indicates stronger reasoning ability, while generating fewer tokens indicates better efficiency. *The result is from Deng et al. (2024).

EXPERIMENTS



Conclusion

- 코코넛의 추론 방식은 llm의 성능을 올리고, 추론시간을 감소 시킴
- Hidden state 에서의 추론은 여러 대안을 동시에 가지고 가 다음 step에서 여러 경우의 수를 통해 결정할 수 있도록 함

***한계**

- 대형 모델에서도 학습을 진행해 봤으나 GPT-2와 같이 성능 차이가 뚜렷하게 나오지 않음
- 학습 시 forward 과정이 여러 번 이루어져 리소스 소모가 큼

Soft Thinking

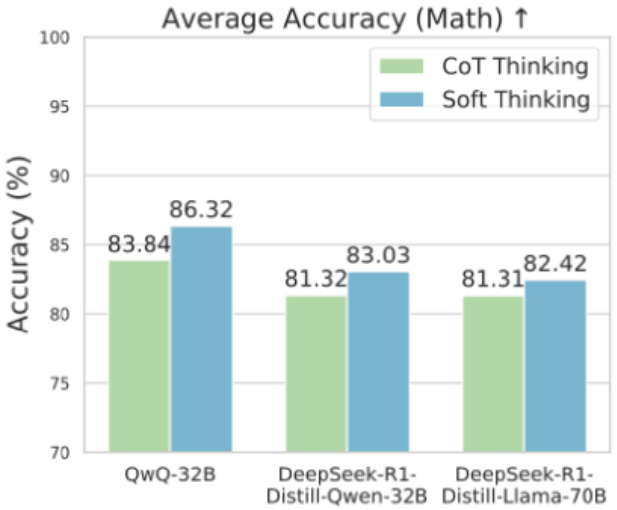
***Abstract & Introduction**

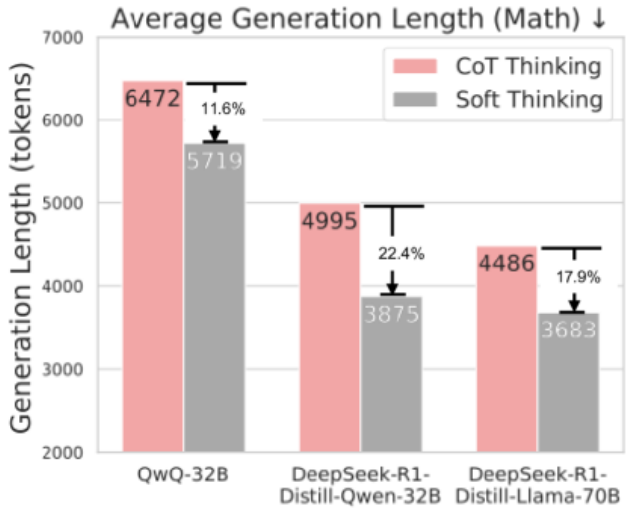
- LLM은 언어 토큰을 사용하여 reasoning하지만 사람은 언어 뿐만이 아니라 추상적이고 유동적인 개념을 포함해 사고함

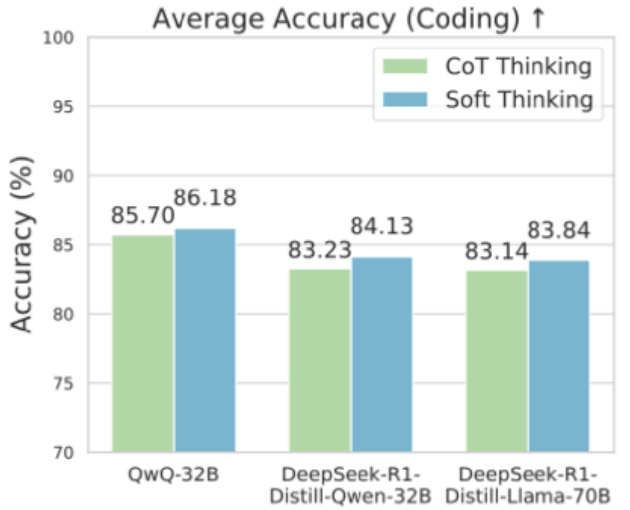
⇒ 언어 토큰 사용은 토큰 샘플링하는 데 의존하기에 추론 경로 탐색이 불완전한 경우가 많음

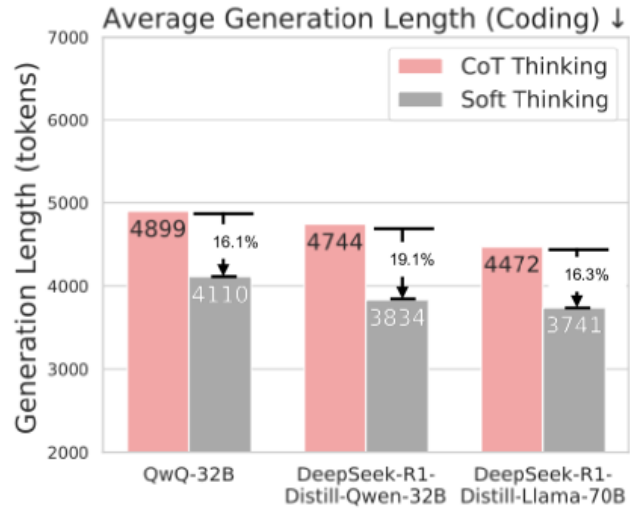
- 연속적인 개념 공간에 추상적인 토큰을 생성하여 풍부한 표현을 가능하게 해보자.

⇒ 이산적인 토큰 선택을 전체 어휘에 대한 가중합으로 대체

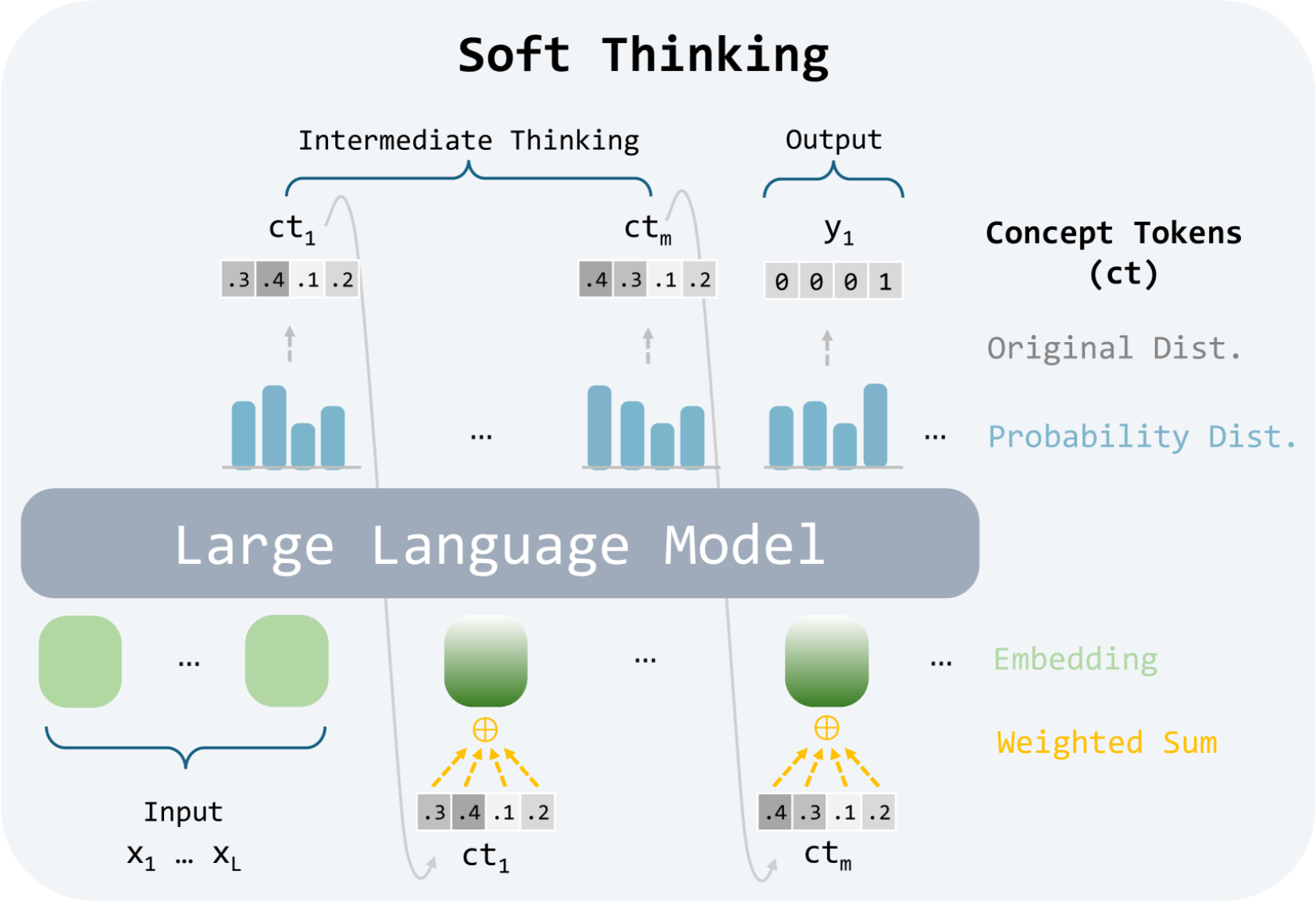








METHODOLOGY



EXPERIMENTS

EXPERIMENTS

- Model
  - QwQ 32B : 강화학습
  - DeepSeek-R1-Distill-Qwen-32B, Llama-70B : 증류모델

- Benchmark
  - 수학 : Math500, AIME 2024, GSM8K, GPQA-Diamond
  - 코딩 : HumanEval, MBPPMBPP, and LiveCodeBench

- Baseline
  - Standard CoT(sampling 16, t = 0.6), Standard Greedy CoT (t = 0)

EXPERIMENTS

*Results

	Accuracy ↑					Generation Length ↓				
	MATH 500	AIME 2024	GSM8K	GPQA Diamond	Avg.	MATH 500	AIME 2024	GSM8K	GPQA Diamond	Avg.
	QwQ-32B [13]									
CoT Thinking	97.66	76.88	96.67	64.17	83.84	4156	12080	1556	8095	6472
CoT Thinking (Greedy)	97.00	80.00	96.57	65.15	84.68 (↑ 0.84)	3827	11086	1536	7417	5967 (↓ 7.8%)
Soft Thinking	**98.00**	**83.33**	**96.81**	**67.17**	**86.32** (↑ 2.48)	**3644**	**10627**	**1391**	**7213**	**5719** (↓ 11.6%)
	DeepSeek-R1-Distill-Qwen-32B [38]									
CoT Thinking	94.50	72.08	95.61	63.10	81.32	3543	9347	875	6218	4995
CoT Thinking (Greedy)	93.00	63.33	95.30	59.09	77.68 (↓ 3.64)	3651	8050	1048	8395	5286 (↑ 5.8%)
Soft Thinking	**95.00**	**76.66**	**95.83**	**64.64**	**83.03** (↑ 1.71)	**3373**	**6620**	**785**	**4722**	**3875** (↓ 22.4%)
	DeepSeek-R1-Distill-Llama-70B [38]									
CoT Thinking	94.70	70.40	94.82	65.34	81.31	3141	8684	620	5500	4486
CoT Thinking (Greedy)	94.61	**73.33**	93.60	66.16	81.92 (↑ 0.61)	**2877**	9457	606	**4443**	4345 (↓ 3.1%)
Soft Thinking	**94.80**	**73.33**	**94.90**	**66.66**	**82.42** (↑ 1.11)	3021	**6644**	**597**	4470	3683 (↓ 17.9%)

Table 1: Comparison of *Soft Thinking* and various baseline methods on accuracy and generation length across mathematical datasets. Best results are highlighted in **bold**.

	Accuracy ↑				Generation Length ↓			
	HumanEval	MBPP	LiveCodeBench	Avg.	HumanEval	MBPP	LiveCodeBench	Avg.
	QwQ-32B [13]							
CoT Thinking	97.63	97.49	62.00	85.70	2557	2154	9986	4899
CoT Thinking (Greedy)	95.73	96.50	57.35	83.19 (↓ 2.51)	**2396**	**2069**	**7034**	**3833** (↓ 21.8%)
Soft Thinking	**98.17**	**97.66**	**62.72**	**86.18** (↑ 0.48)	2638	2157	7535	4110 (↓ 16.1%)
	DeepSeek-R1-Distill-Qwen-32B [38]							
CoT Thinking	97.25	95.13	57.33	83.23	3095	2761	8376	4744
CoT Thinking (Greedy)	87.19	87.54	43.36	72.70 (↓ 10.53)	**2294**	**1703**	**4702**	**2900** (↓ 38.9%)
Soft Thinking	**97.56**	**95.33**	**59.50**	**84.13** (↑ 0.90)	2713	2534	6255	3834 (↓ 19.1%)
	DeepSeek-R1-Distill-Llama-70B [38]							
CoT Thinking	97.71	94.77	56.94	83.14	2711	2386	8319	4472
CoT Thinking (Greedy)	92.07	91.82	48.02	77.30 (↓ 5.84)	2192	1979	5438	3203 (↓ 28.3%)
Soft Thinking	**98.17**	**94.94**	**58.42**	**83.84** (↑ 0.70)	2498	2214	6512	3741 (↓ 16.3%)

Table 2: Comparison of *Soft Thinking* and various baseline methods on accuracy and generation length across two coding datasets. Best results are highlighted in **bold**.

EXPERIMENTS

Question: 43 * 34 = ?

CoT Thinking (157 tokens):

First, I recognize that multiplying 43 by 34 can be simplified by breaking down the numbers into more manageable parts. I'll decompose 34 into 30 and 4.\n\nNext, I'll multiply 43 by 30. Since 30 is three times 10, I can first multiply 43 by 3 to get 129, and then add a zero to account for the 10, resulting in 1,290.\n\nThen, I'll multiply 43 by 4, which equals 172.\n\nFinally, I'll add the two results together: 1,290 plus 172 gives me the final answer of 1,462.

Soft Thinking (96 tokens):

To solve 43 multiplied by 34, I'll start by breaking down the numbers into their place values. \n\nFirst, I'll multiply 43 by 4, which gives me 172.\n\nNext, I'll multiply 43 by 30, resulting in 1,290.\n\nFinally, I'll add these two products together: 172 plus 1,290 equals 1,462.

EXPERIMENTS

*Ablations

	Accuracy		Generation Length All		Generation Length Correct	
	AIME 2024 ↑	LiveCodeBench ↑	AIME 2024 ↓	LiveCodeBench ↓	AIME 2024 ↓	LiveCodeBench ↓
COCONUT-TF	0.0	0.0	32,768	32,768	–	–
Average Embedding	6.66	7.49	30,802	30,474	**6,556**	**2,141**
Soft Thinking w/o Cold Stop	73.33	56.98	12,991	13,705	9,457	6,877
Soft Thinking w/ Cold Stop	**83.33**	**62.72**	**11,445**	**12,537**	10,627	7,535

Table 3: Ablation study of Soft Thinking with QwQ-32B on the AIME 2024 and LiveCodeBench. COCONUT-TF represents training-free COCONUT [27].

Conclusion

- 여러 추론 경로를 병렬적으로 탐색할 수 있음
- 정확도를 향상하고, 생성 토큰을 감소함
- 추가적인 학습이나 아키텍처 수정 없이 가능한 디코딩 전략

***한계**

- 학습시에 입력 토큰에 가중합하여 여러 토큰을 넣는 과정을 진행한 적이 없기에 OOD 상태에 자주 빠짐 => 생성 붕괴가 일어남
- 콜드 스탑(엔트로피가 낮아지면 </think>)으로 문제를 완화하였지만 근본적으로 해결하지 못함

Thank you