Fine-Tuning on Diverse Reasoning Chains Drives Within-Inference CoT Refinement in LLMs

Haritz Puerto[1], Tilek Chubakov[1]
Xiaodan Zhu[2], Harish Tayyar Madabushi[3], Iryna Gurevych[1]
[1]Ubiquitous Knowledge Processing Lab (UKP Lab), TU Darmstadt and ATHENE National Research Center for Applied Cybersecurity, Germany
[2]Dept. of ECE & Ingenuity Labs Research Institute, Queen's University, Canada
[3]University of Bath, UK
https://www.ukp.tu-darmstadt.de

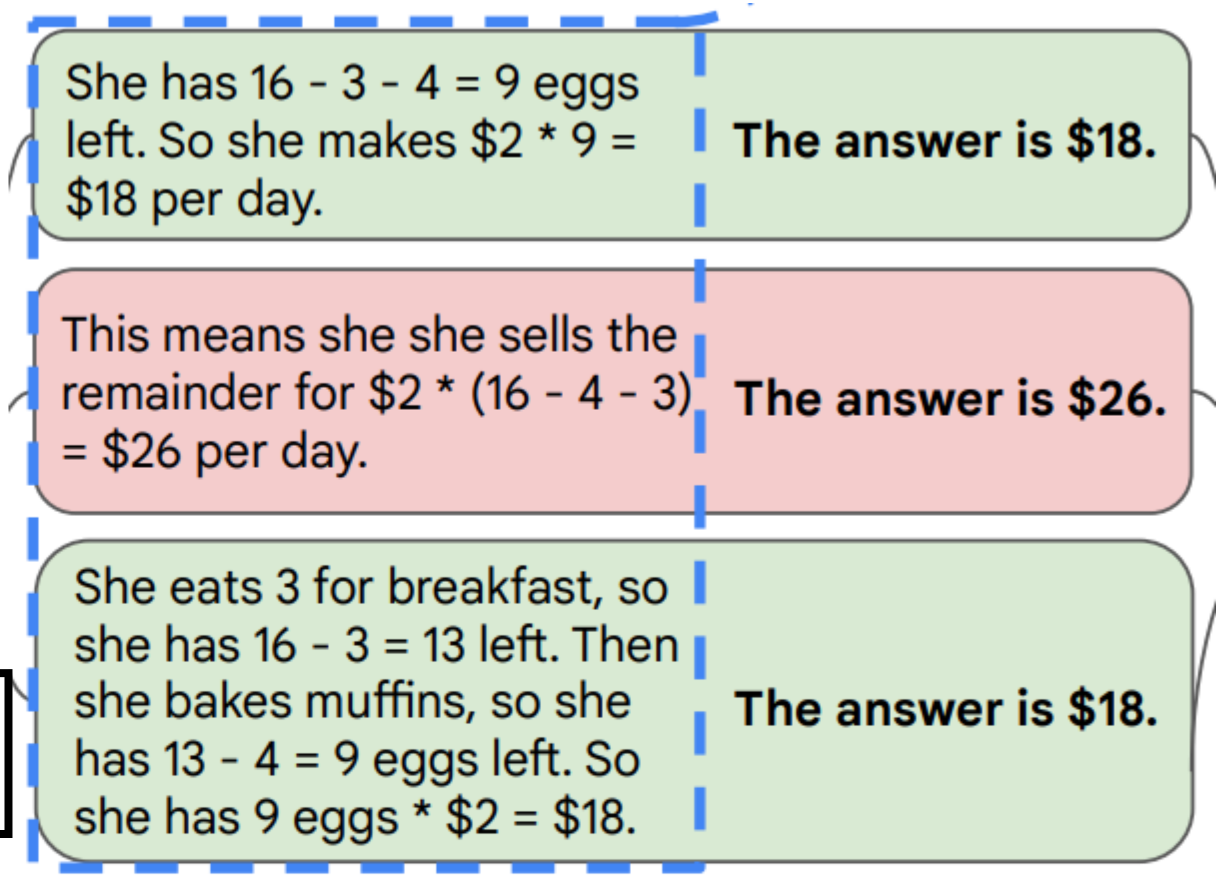
Project Website

CoT Can Be Wrong!

- Methods like self-consistency are effective but
  - CoTs can be **repetitive**
  - Most **common answer** can be **wrong**
  - **CoTs are not used to create better CoTs**

How to leverage CoTs while creating new ones?

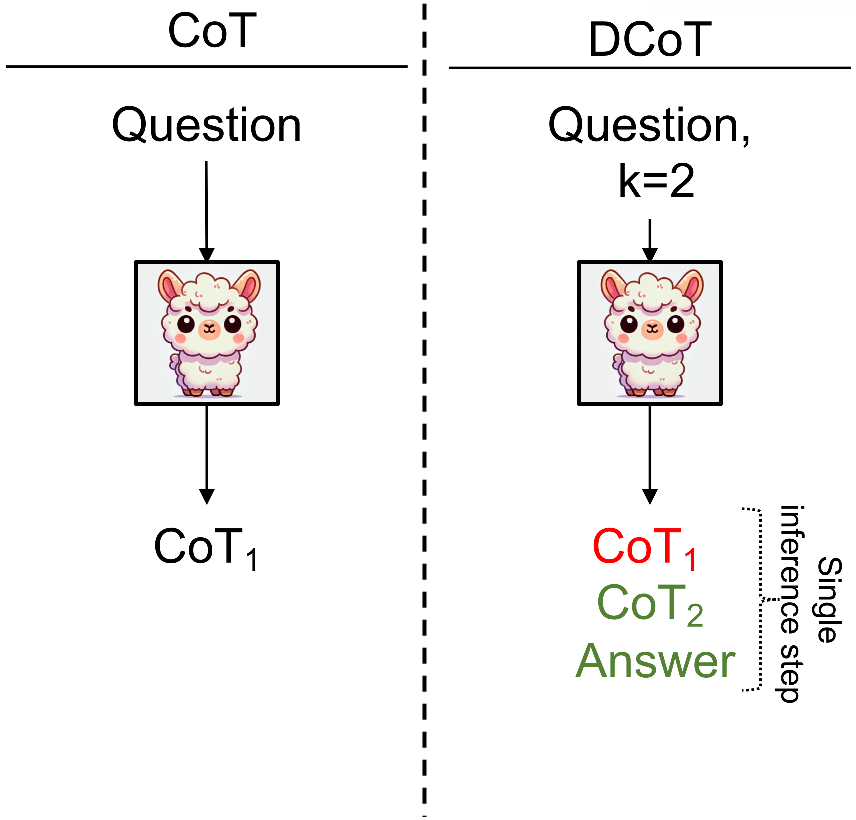
Can language models refine their reasoning *on the go*, like humans do?



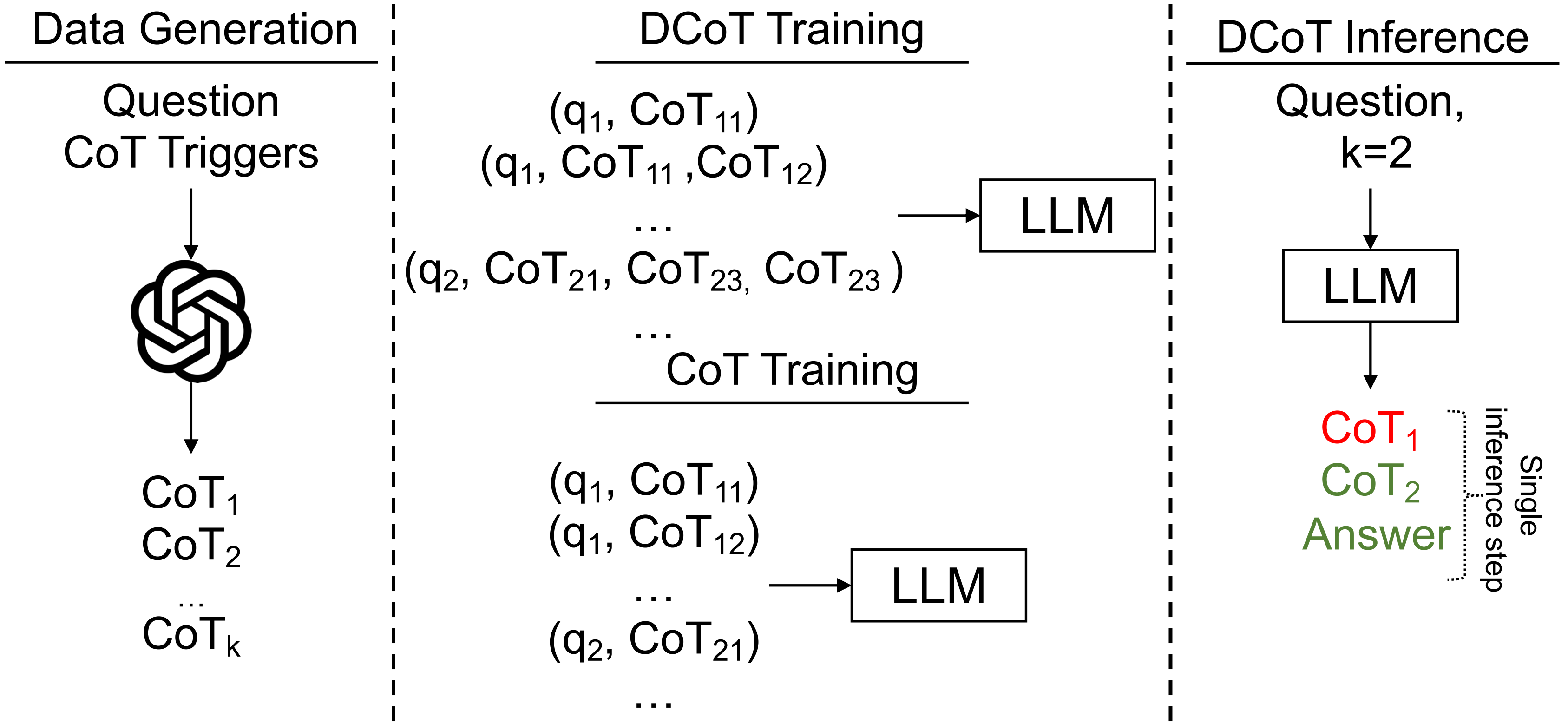
Wang et al., (2023)

Diverse CoT

- Train LLMs to:
  1. Generate multiple CoTs in their output in a *single inference step*
  2. Final answer
- Why?
  - CoTs are aware of prior ones
    - Can refine answers
    - Can force diversity



Training Schema

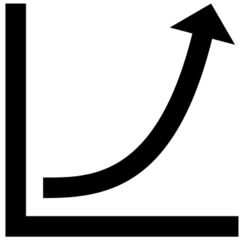


Prompt:[Question] Question [Options] Answer Options [Number of answers] k

Training Details

Models

- Phi 1.5 (1.3B)
- Phi 2 (2.7B)
- LLaMA 2 7B
- LLaMA 2 13B
- LLaMA 2 70B



Data

- 9k questions from 9 QA datasets
- <= 4 CoTs/Question
- Total: 32032 CoTs

Dataset	Task
ARC	Multiple choice
BGQA	Multiple choice
Coin Flip	Multiple choice
CQA	Span extraction
GSM8K	Num. Generation
HQA	Span extraction
LLC	Generation
Quartz	Multiple choice
StrQA	Boolean QA

Results

DCoT Can Refine Answers at Inference Time

LLM	Phi 1.5 (1.3B)	Phi 2 (2.7B)	LLaMA2 7B	LL. 13B	LL. 70B
CoT	47.2	60.85	58.97	64.39	66.96
DCoT	**49.39**	**62.6**	**60.8**	**66.18**	**68.63**

- DCoT (on average) outperforms CoT <u>despite being trained on the same CoTs</u>
- DCoT benefits from CoT extensions
  - Eg: self-consistency

DCoT Can Refine Answers at Inference Time

- CoT = DCoT@1
- Just a second CoT always improves!
    - Increase cost per call is negligible

Method	Phi 1.5	Phi 2	LL. 7B	LL. 13B
CoT	47.51±1.77	63.51±.71	59.30±.54	65.41±.91
DCoT@1	47.87±1.71	63.91±2.58	61.28±.50	65.80±.44

DCoT remains Robust in OOD

- Generalization to OOD requires thousands of tasks (Kim et al., 2023) 🤓
- Since little training in only 9 tasks, could generating more than 1 CoT be detrimental in OOD 🤔 No! 🙅‍♂️

LLM	Method	CSQA
Phi 1.5	CoT	33.88
	DCoT@1	32.26
	DCoT@2	34.23
	DCoT@3	33.81
	DCoT@4	34.73
Phi 2	CoT	44.29
	DCoT@1	44.15
	DCoT@2	44.13
	DCoT@3	45.99
	DCoT@4	45.43

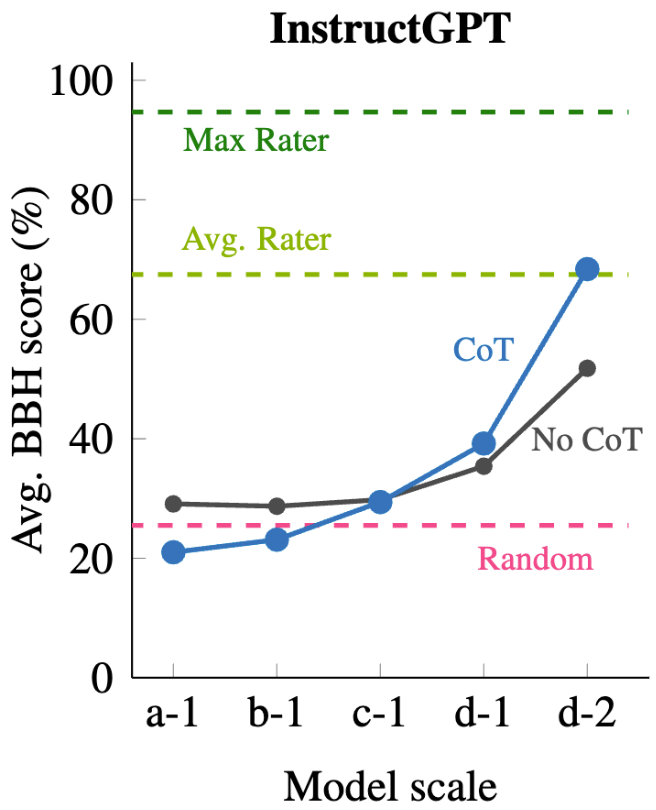
LLM	Method	CSQA
LLaMA2 7B	CoT	38.41
	DCoT@1	36.94
	DCoT@2	40.79
	DCoT@3	40.67
	DCoT@4	40.43
LLaMA2 13B	CoT	46.55
	DCoT@1	44.62
	DCoT@2	45.48
	DCoT@3	47.42
	DCoT@4	46.45

- Effective in CSQA
- Limited effectiveness in OOD numerical reasoning
  - Expected; LLMs struggle in numerical OOD (Qian et al., 2023)

Robust where CoT is Decremental

- CoT is detrimental on BBH unless the model is large enough (Suzgun et al., 2023) 🤓
- Would generating more CoTs (i.e., DCoT) be worse? 🤔 No! 🙅‍♂️

Method	Phi 1.5	Phi 2	LL. 7B	LL. 13B
CoT	28.37	46.7	31.08	36.38
DCoT@1	28.31	44.56	31.23	34.59
DCoT@2	28.07	45.81	31.11	35.94
DCoT@3	28.35	45.92	31.00	36.90
DCoT@4	28.21	46.71	31.13	36.45



Suzgun et al., (2023)

DCoT is not ensembling CoTs

- Ensembling CoTs require at least 3 CoTs to be beneficial → We only need 2
- Our manual analysis shows that:
  - If the 1st CoT is correct, subsequent CoTs are the same
  - If the 1st CoT is wrong, subsequent CoTs are different
  - If the 1st CoT is **slightly** wrong, 2nd CoT is just a **small** change
  - If the 1st CoT is **completely** wrong, 2nd CoT is **completely** different
  - If the answer from the last CoT is wrong, the model may select a prior (better) answer

DCoT refines the reasoning on the go

Examples

Question: Take the last letters of each words in “Billie Paloma Tanner Raul” and concatenate them.

[Answer 1] 1. Write down the words: Billie Paloma Tanner Raul
2. Take the last letter of the first word "Billie", which is "e".
3. Take the last letter of the second word "Paloma", which is "a".
4. Take the last letter of the third word "Tanner", which is "r".
5. Take the last letter of the fourth word "Raul", which is "l".
6. Concatenate these letters together: "ealr".
7. The result word is "ealr".

[Answer 2] Sure, let’s break it down step by step:
1. The last letter of "Billie" is "e".
2. The last letter of "Paloma" is "a".
3. The last letter of "Tanner" is "r".
4. The last letter of "Raul" is "l".

Now, if we concatenate these letters, we get the word "earl". Therefore, the answer is "earl".

[Final answer] earl

CHE
ITÄT
ADT
OUS
DGE
ING

Question: Cornelia likes to travel. She visited already 42 different countries. 20 of them were in Europe and 10 in South America. From the rest of the countries, only half of them were in Asia. How many Asian countries has Cornelia visited?

[Answer 1] 1. We know that Cornelia visited 42 countries in total.
2. We also know that 20 of those countries were in Europe and 10 were in South America.
3. Therefore, the remaining countries must be half of the total number of countries Cornelia visited, which is 42/2 = 21.

Remaining countries = 42-20-10

4. We know that Cornelia visited only half of the Asian countries, which means there are 21 - 10 = 11 Asian countries left to be determined.
5. Therefore, Cornelia visited 11 Asian countries in total.

[Answer 2] Step 1: Let's start by finding out how many countries Cornelia visited in Europe and South America. We know that she visited 20 countries in Europe and 10 countries in South America, so the total number of countries she visited in these two regions is: 20 + 10 = 30

Step 2: We know that Cornelia visited 42 countries in total. We also know that 30 of these countries were in Europe and South America. Therefore, the number of countries she visited outside of these regions is: 42 - 30 = 12

42-30 are the remaining countries!

Correct

Step 3: We are told that only half of the countries Cornelia visited outside of Europe and South America were in Asia. This means that the number of countries she visited in Asia is: 12 / 2 = 6

Therefore, Cornelia visited 6 Asian countries.

[Final answer] 6

TECHNISCHE UNIVERSITÄT DARMSTADT

UBIQUITOUS KNOWLEDGE PROCESSING

Takeaways

- First analysis of the potential behind the ideas of reasoning models (preprint from June 2024)
- Just generating **a second CoT consistently improves** performance
  - Despite being trained on exactly the same CoTs
- Making the **CoTs aware of each other** allows the model to **refine** its **answer**

haritz.puerto@tu-darmstadt.de