

Unpacking Reasoning in LLMs

Input Formats, Generating CoTs, and Fair Evaluation

Haritz Puerto

April 2025

Why Reasoning in LLMs Matters?



DATA SCIENCE

7	10	17	41	55	24
21	14	26	48	41	20
46	45	62	84	18	13
30	39	22	64	35	33
27	84	37	42	96	23
21	42	33	16	87	12
25	37	68	25	37	33
26	44	57	45	63	25
12	18	22	45	31	57

Ai2 Scholar QA

A research tool for literature synthesis, with broad and deep coverage via a corpus of 8M+ full text papers and 108M+ abstracts. Ai2 Scholar QA can make mistakes. Check source documents by following citations. A project from [Ai2](#).

Enter a literature review question...



How Can We Improve Reasoning?



1. Can we map the **input** to another **space** that better suits for reasoning?
2. Can we improve the LLM's **processing** of **reasoning traces**?

Code Prompting Elicits Conditional Reasoning Abilities in Text+Code LLMs

Haritz Puerto¹, Martin Tutek^{2*}, Somak Aditya³, Xiaodan Zhu^{1,4}, Iryna Gurevych¹

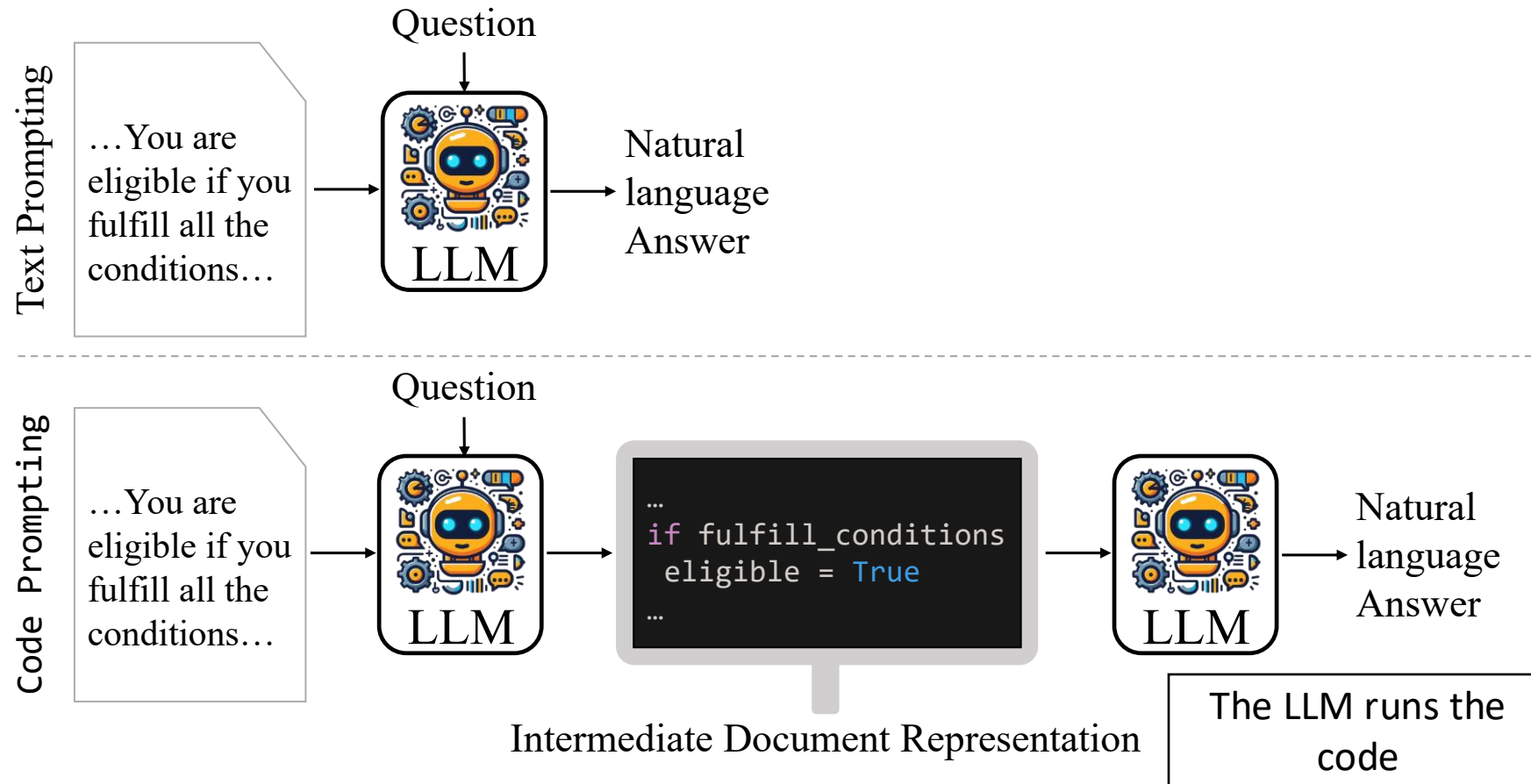
¹Ubiquitous Knowledge Processing Lab (UKP Lab),
TU Darmstadt and Hessian Center for AI (hessian.AI)

²Technion – IIT, ³IIT Kharagpur, ⁴Queen's University

<https://www.ukp.tu-darmstadt.de>

Can we elicit reasoning abilities in LLMs by merely changing the input format (text \rightarrow code) ?

Code Prompting





Text Prompt

Question: Ann's husband passed away. She needs help for the burial in the UK. Can she be eligible for funeral expenses payment?

Doc: You can get a Funeral Expense Payment if all of the following apply:

- You meet the rules on your relationship with the deceased
- You're arranging a funeral in the UK



Answer: CoT + Ans



Code Prompt

```
# Question: Ann's husband passed away. She needs help for the burial in the UK. Can she be eligible for funeral expenses payment?
husband_pass_away = True
needs_help_for_burial_in_UK = True
eligible_funeral_expenses_payment = None # This is the question
```

```
# Doc:
# You can get a Funeral Expense Payment...
if (meet_rules_relationship and
    funeral_in_UK):
    eligible_funeral_expenses_payment = True
```



Answer: CoT + Ans

Methodology & Scope

- Text+Code LLMs: GPT 3.5, Mixtral, Mistral
- Task: **Conditional Question Answering**
- Datasets:
 - ConditionalQA
 - ShARC
 - BoardgameQA

[ConditionalQA: A Complex Reading Comprehension Dataset with Conditional Answers](#) (Sun et al., ACL 2022)

Document:

Section 1: Overview

You can get a Funeral Expense Payment if all of the following apply:

- You meet the rules on your relationship with the deceased
- you're arranging a funeral in the UK

....

Section 2: What you will get

Section 3: Your relationship

You must be one of the following:

- the partner of the deceased when they died
- a close relative or close friend
-

You might not get a Funeral Expenses Payment if another close relative of the deceased (such as a sibling or parent) is in work.

....

Question:

Scenario: Ann lives in the UK. Her husband has succumbed to cancer. She needs help to give her late husband a decent burial.

Question: Can she be eligible for funeral expenses payments?

Answer:

Answer: Yes

Conditions: ["you're arranging a funeral in the UK"]

Answer: No

Conditions: ["You might not get a Funeral Expenses Payment if another close relative of the deceased ..."]

Code Prompts Outperforms Text Prompts

Model	Prompt	CondQA	ShARC	BGQA-1	BGQA-2	BGQA-3	ΔCP
GPT 3.5	Text	58.70	62.95	51.15	37.42	27.77	8.42
	Code	60.60	54.98	58.67	55.56	50.29	
Mixtral	Text	48.17	53.77	56.38	39.64	30.15	4.22
	Code	44.73	59.06	53.33	47.39	44.72	
Mistral	Text	35.74	43.60	47.40	48.78	47.86	2.74
	Code	33.28	49.92	53.80	51.27	48.79	

- Code prompts performs best on most datasets (11/15)
- All models achieve best results with code prompts on most datasets

Code Prompts Outperforms Text Prompts

Model	Prompt	CondQA	ShARC	BGQA-1	BGQA-2	BGQA-3	Δ CP
GPT 3.5	Text	58.70	62.95	51.15	37.42	27.77	8.42
	Code	60.60	54.98	58.67	55.56	50.29	
Mixtral	Text	48.17	53.77	56.38	39.64	30.15	4.22
	Code	44.73	59.06	53.33	47.39	44.72	
Mistral	Text	35.74	43.60	47.40	48.78	47.86	2.74
	Code	33.28	49.92	53.80	51.27	48.79	

- BGQA-2, 3 are the most reasoning-intensive tasks. The gains are the largest here

Ablation: Implicit Text Simplification?

- Are the performance gains due to the implicit text simplification obtained from the code format? 🤔
 - Key entities (variables) and conditions are explicitly stated 👉
- 2 ablations:
 - Atomic Statements
 - Back-translation

Original Sentence

Applying for the legal right to deal with someone's property, money, and possessions (their estate) when they die is called applying for probate.

Atomic Statements

Applying for the legal right is a process.

The someone is a person who has died.

The process is called 'applying for probate'.

The legal right is to deal with someone's property, money, and possessions.

The property, money, and possessions are collectively called the 'estate'.

Original Transformation

Question: Ann's husband passed away. She needs help for the burial in the UK. Can she be eligible for funeral expenses payment?

Doc: You can get a Funeral Expense Payment if all of the following apply:

- You meet the rules on your relationship with the deceased
- You're arranging a funeral in the UK

```
# Question: Ann's husband passed away. She needs help for the burial in the UK. Can she be eligible for funeral expenses payment?
husband_pass_away = True
needs_help_for_burial_in_UK = True
eligible_funeral_expenses_payment = None # This is the question
```

```
# Doc:
# You can get a Funeral Expense Payment...
if (meet_rules_relationship and funeral_in_UK):
    eligible_funeral_expenses_payment = True
```

Back-translation

Question: Ann's husband passed away. She needs help for the burial in the UK. Can she be eligible for funeral expenses payment?

Doc: if you meet the rules on your relationship with the deceased and you are arranging a funeral in the UK,
you can get a Funeral Expense Payment

Ablation: Implicit Text Simplification?

Dataset	Δ Atomic St.	Δ Code \rightarrow NL
CondQA	-2.66	-4.72
BGQA-1	-4.37	-1.43
BGQA-2	-8.72	-5.39
BGQA-3	-19.26	-3.68

- Natural Language Text resembling code does not yield performance boosts
- Code prompts enhance LLM performance beyond mere text simplification 😲

Ablation: Code Semantics are Important

- So it is important to have code to elicit LLM's reasoning abilities
- But, just showing any code is important? 🤔
- Do we need to keep the original text as code comment too? 🤔

Ablations:

- Anonymized code: instead of `if` employed → `if var1`
- Random code: code with no relation to the original instance
- Remove text comments

Ablation: Code Semantics are Important

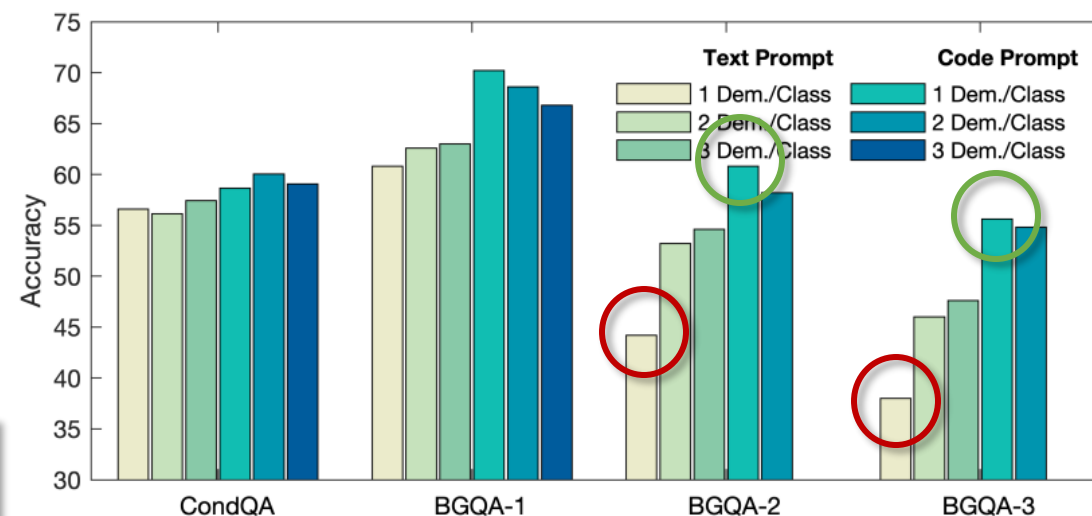
Prompt	CQA	CQA-YN	BG ₁	BG ₂	BG ₃
Anonym.	-1.62	-2.90	-6.60	-4.80	-4.00
Random	-3.40	-2.67	-7.40	-9.20	-9.80
- Comments	N.A.	-14.02	-16.70	-16.20	-5.20

- Removing the original text instance (as comments) drops performance a lot → LLM uses the text instance to answer!
- Anonymized code drops performance
- Random code drops performance even more
 - Performance is close to text prompts → LLM ignores the code

Code Prompting is More Efficient

- We prompt LLMs with 1-3 demonstrations per class (Yes/No/Span)
- Gap is largest when using 1 demonstration
- Text Prompts needs > 1

Code Prompts are more sample efficient than text prompts



Code Prompting Improves Variable State Tracking

- Pretraining on code might improve entity tracking in LLMs
- After each sentence in the answer output, we check whether the model remembers the initial context

Question: Ann's husband passed away. She needs help for the burial in the UK. Can she be eligible for funeral expenses payment?

Doc: You can get a Funeral Expense Payment if all of the following apply:

- You meet the rules on your relationship with the deceased
- You're arranging a funeral in the UK


Answer: ...

Is it true that Ann's husband passed away?
Is it true she needs help for the burial in the UK?

Code Prompting Improves Variable State Tracking

Dataset	Correct Ans.		Incorrect Ans.	
	Text	Code	Text	Code
CondQA	71.08	4.39	60.79	11.39
BGQA-1	39.33	8.84	51.65	22.12
BGQA-2	44.79	15.04	52.54	24.75
BGQA-3	54.01	14.21	52.13	16.98

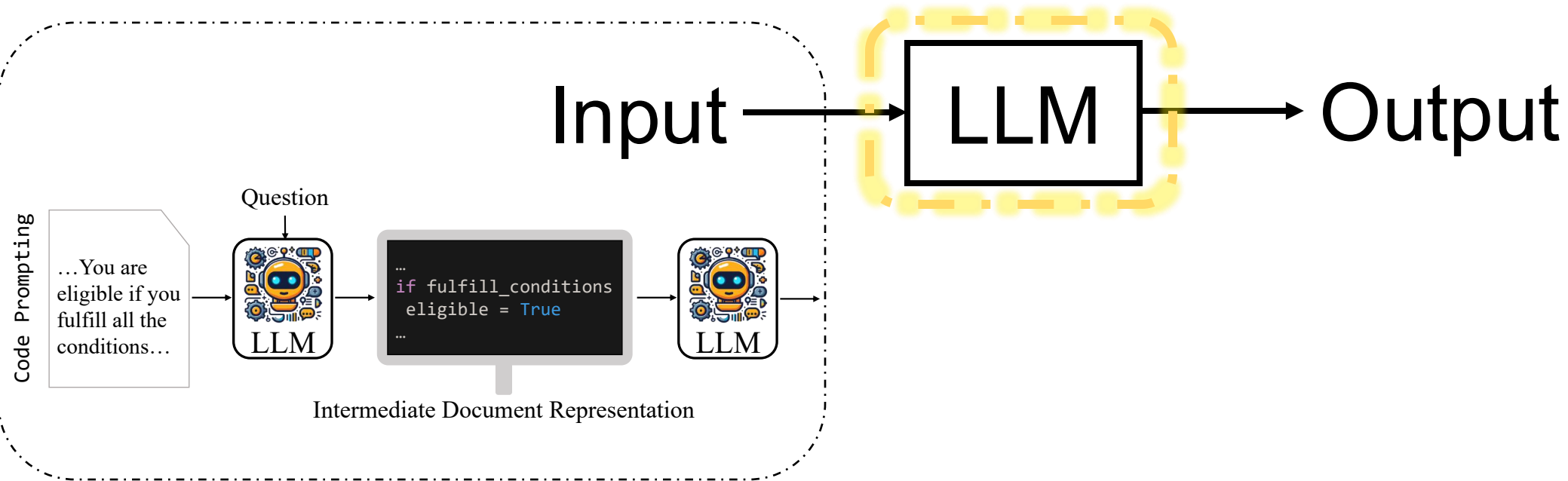
Memory errors on the questions about the context of the question (lower is better)

Code Prompts yield much better results at remembering the initial facts of the context! 

Takeaways

- The input representation of an LLM affect its reasoning abilities
- Code Prompts can elicit conditional reasoning abilities in text+code LLMs
 - Improves variable state tracking 🔍
 - It's more efficient (in terms of #demonstrations required) 🙌

Improving Reasoning in LLMs



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

Haritz Puerto¹, Tilek Chubakov¹

Xiaodan Zhu², Harish Tayyar Madabushi³, Iryna Gurevych¹

¹Ubiquitous Knowledge Processing Lab (UKP Lab),

TU Darmstadt and Hessian Center for AI (hessian.AI), Germany

²Dept. of ECE & Ingenuity Labs Research Institute, Queen's University, Canada

³University of Bath, UK

<https://www.ukp.tu-darmstadt.de>

CoT Can Be Wrong!

- These CoTs are independent of each other, so the LLM can make the same mistake many times
- How to pick the right answer?

How can the LLM generate many reasoning traces so that they can refine its answer?

She has $16 - 3 - 4 = 9$ eggs left. So she makes $\$2 * 9 = \18 per day.

The answer is \$18.

This means she she sells the remainder for $\$2 * (16 - 4 - 3) = \26 per day.

The answer is \$26.

She eats 3 for breakfast, so she has $16 - 3 = 13$ left. Then she bakes muffins, so she has $13 - 4 = 9$ eggs left. So she has $9 \text{ eggs} * \$2 = \18 .

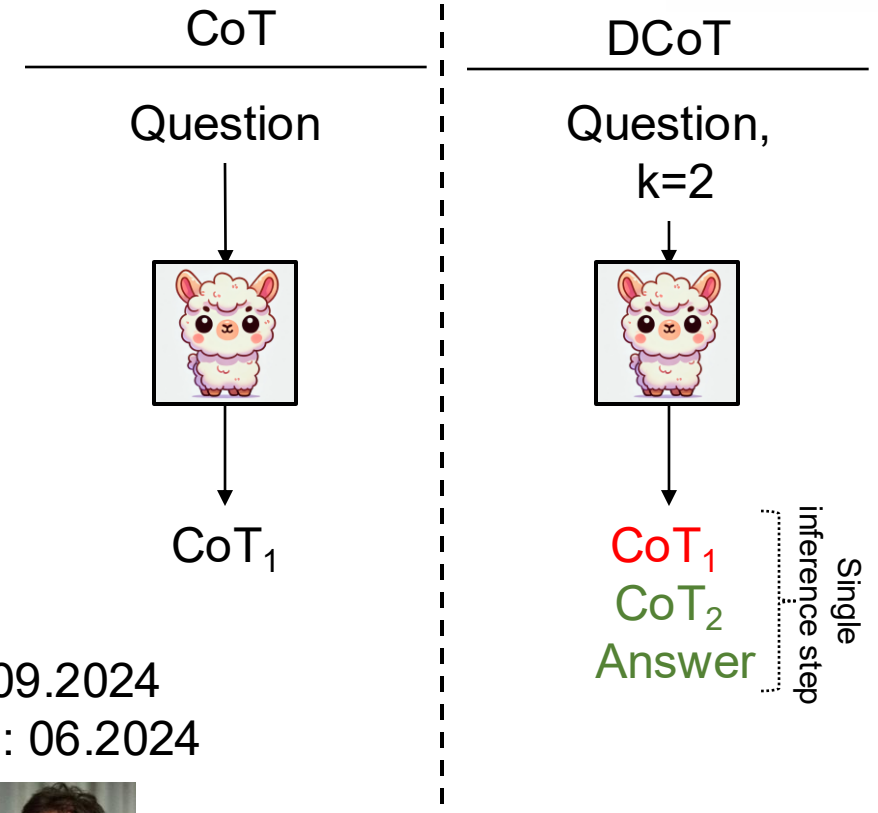
The answer is \$18.

Diverse CoT

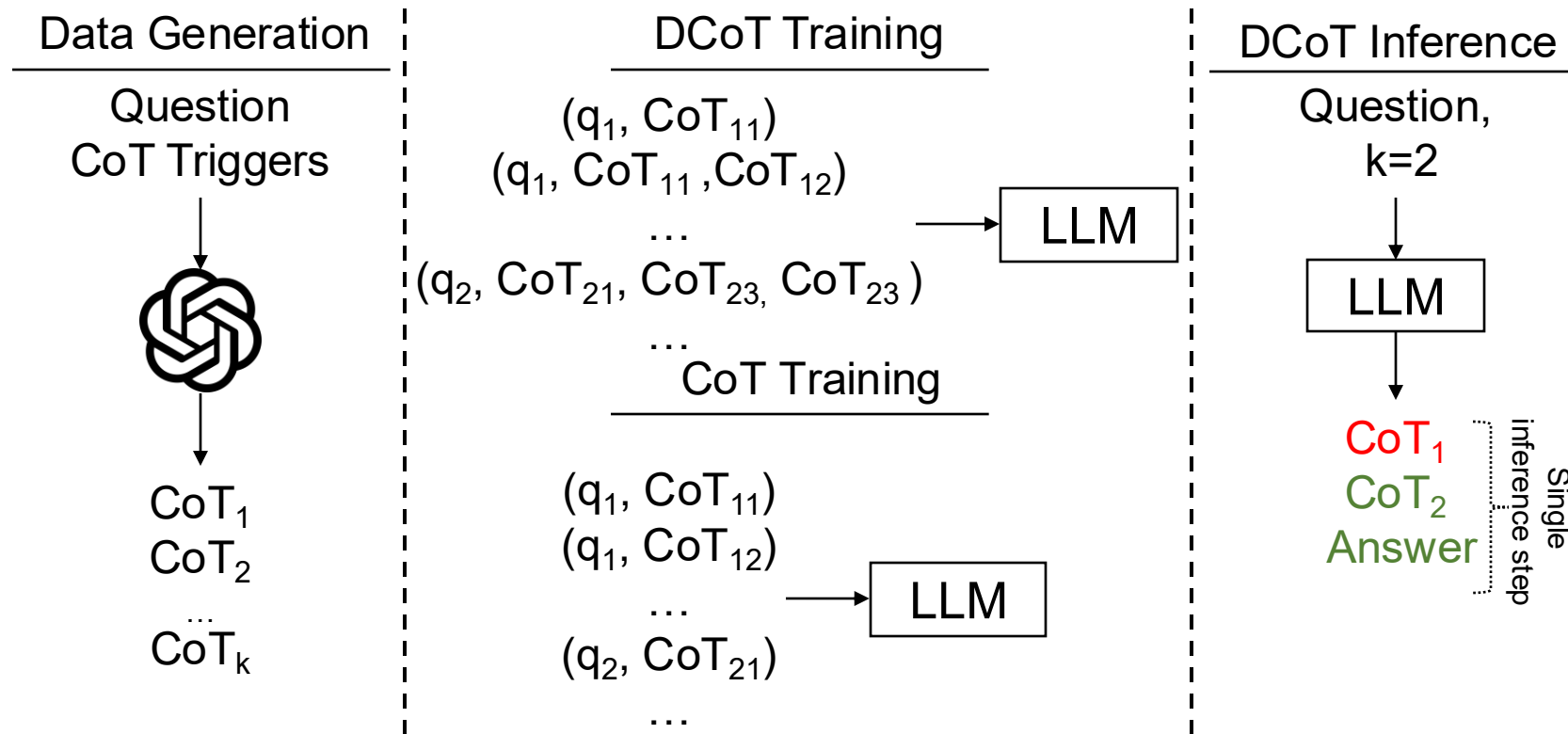
- Train LLMs to:
 1. Generate multiple CoTs in their output
 2. Final answer
- Why?
 - CoTs are aware of prior ones
 - Can refine answers
 - Can force diversity



O1: 09.2024
Ours: 06.2024



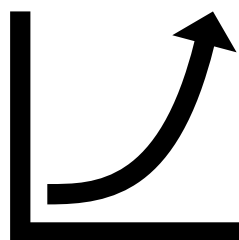
Training Schema



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

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

DCoT Can Refine Answers at Inference Time

- CoT = DCoT@1
- Just a second CoT always improves!
 - Increase cost per call is negligible
- DCoT benefits from CoT extensions
 - Eg: self-consistency

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@2	48.63±.67↑	65.33±2.80↑	62.46±.45↑	67.30±.49↑

LLM	Method	Avg.
Phi 1.5 (1.3B)	CoT	47.2
	DCoT (Ours)	49.39
	CoT + SC	46.48
Phi 2 (2.7B)	DCoT + SC	49.01
	CoT	60.85
	DCoT	62.6
LLaMA2 7B	CoT + SC	61.5
	DCoT + SC	65.12
	CoT	58.97
LLaMA2 13B	DCoT	60.8
	CoT + SC	62.9
	DCoT + SC	61.09
LLaMA2 70B*	CoT	64.39
	DCoT	66.18
	CoT + SC	66.82
LLaMA2 70B*	DCoT + SC	68.12
	CoT	66.96
LLaMA2 70B*	DCoT	68.63

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: "earl".

7. The result word is "earl".

[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".

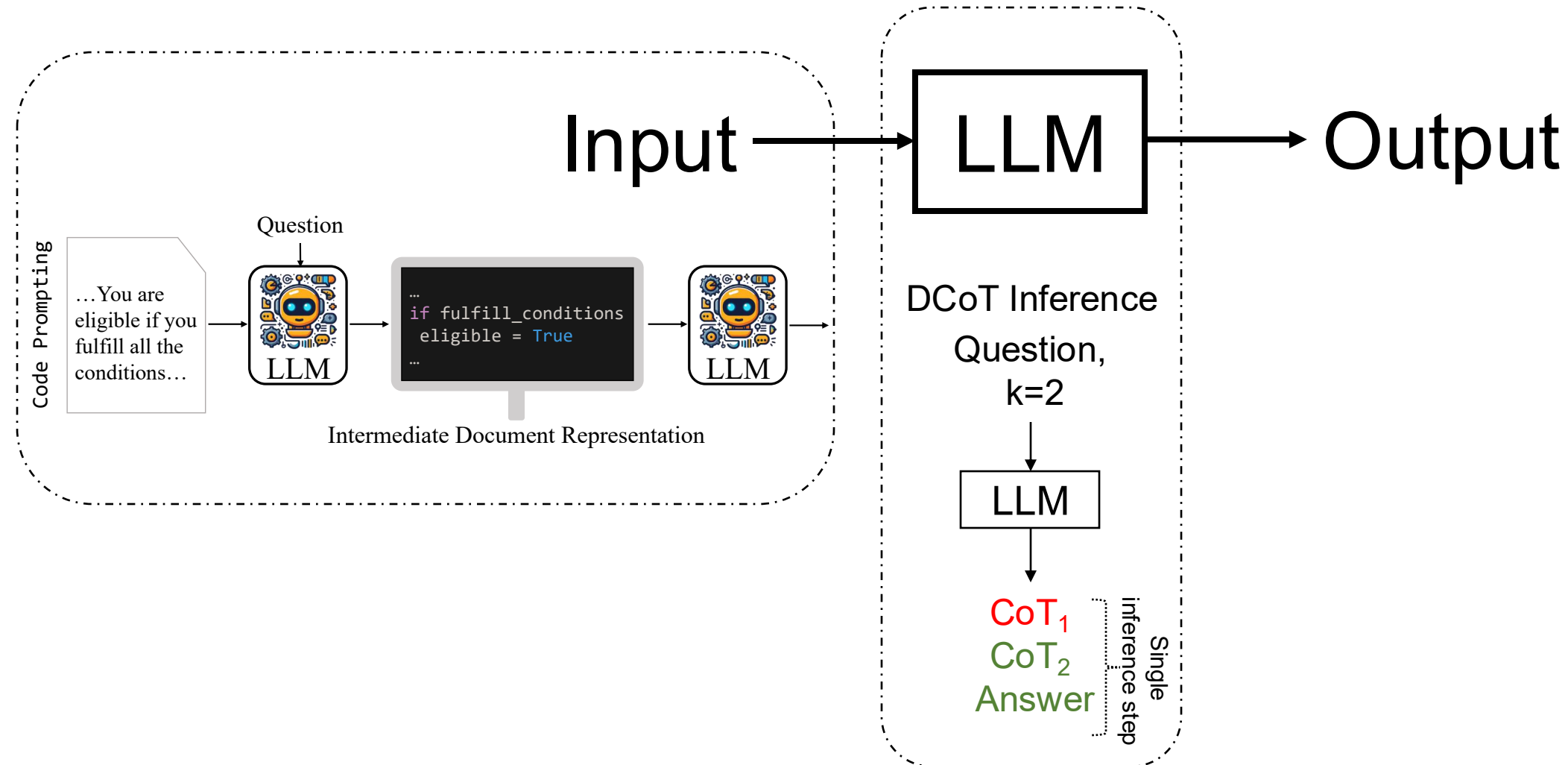
Now, if we concatenate these letters, we get the word "earl". Therefore, the answer is "earl".

[Final answer] earl

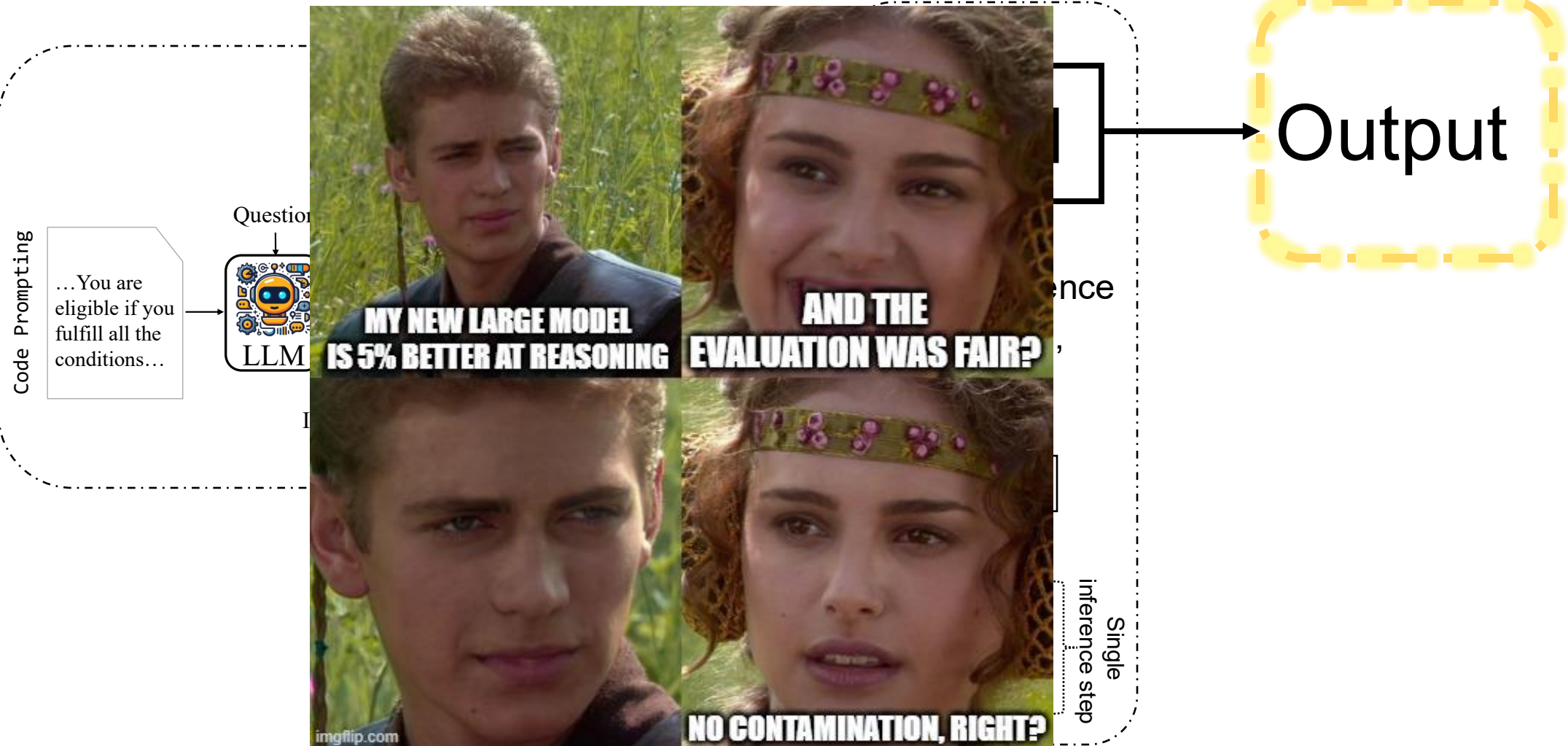
Takeaways

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

Improving Reasoning in LLMs



How to Ensure Fair Evaluations?



Scaling Up Membership Inference: When and How Attacks Succeed on Large Language Models

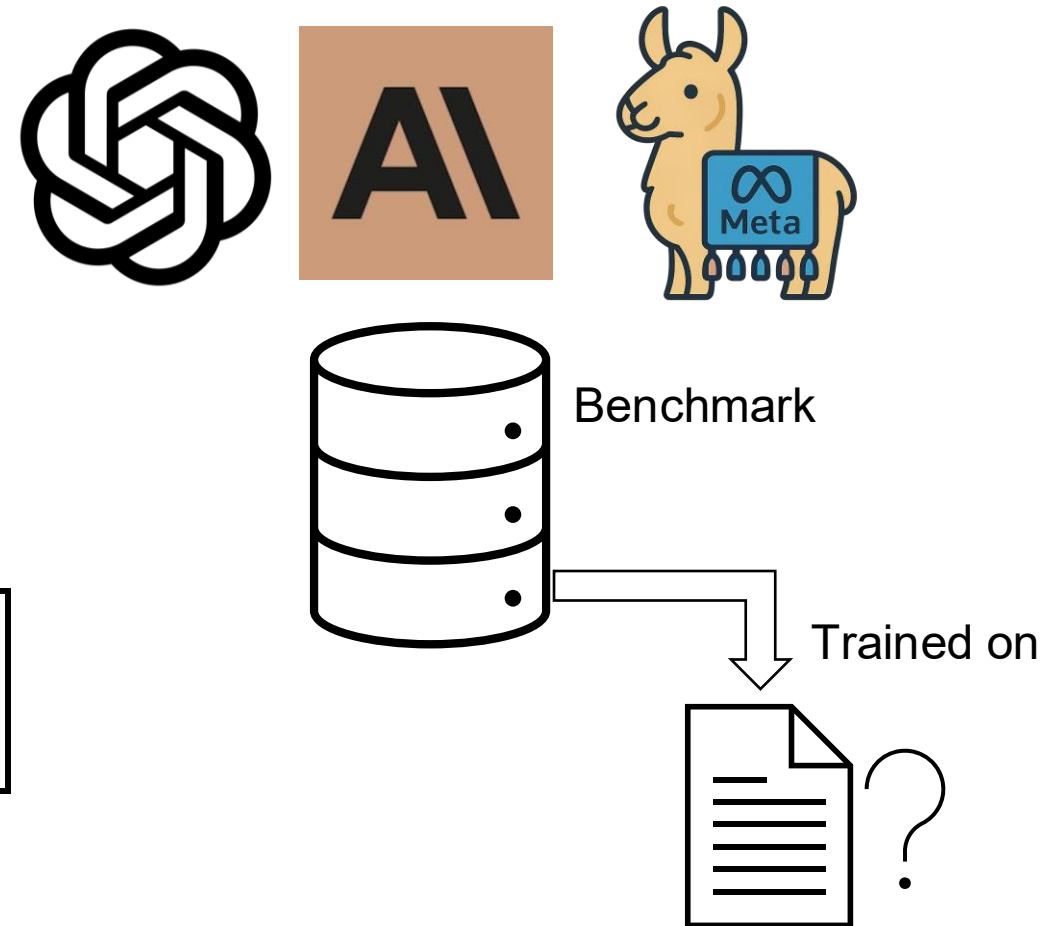
Haritz Puerto^{1,2,*}, Martin Gubri¹, Sangdoon Yun³, Seong Joon Oh^{1,4,5},

¹Parameter Lab, ²Ubiquitous Knowledge Processing Lab, Technical University of Darmstadt,
³NAVER AI Lab, ⁴University of Tübingen, ⁵Tübingen AI Center



Are Data Points from the Benchmark in the Training Set of the LLM?

- LLMs are trained vast amounts of **secret** data
- There is commercial interest in beating benchmarks
- A **fair evaluation** should guarantee **no test-set contamination**



How can we prove whether an LLM was trained on a data point?



LLM Devs. May be training on Copyrighted Data

Authors sue Anthropic for copyright infringement over AI training

The New York Times

Andrea Bartz, Charles Graeber and Kirk Wallace Johnson
allege company misused work to teach chatbot Claude

Major Canadian news outlets sue OpenAI

29 November 2024

Holly Honderich
BBC News

The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.

The Intercept Media, Inc. v. OpenAI, Inc.
District Court, S.D. New York

Written by
Perplexity Team

Published on
Oct 24, 2024

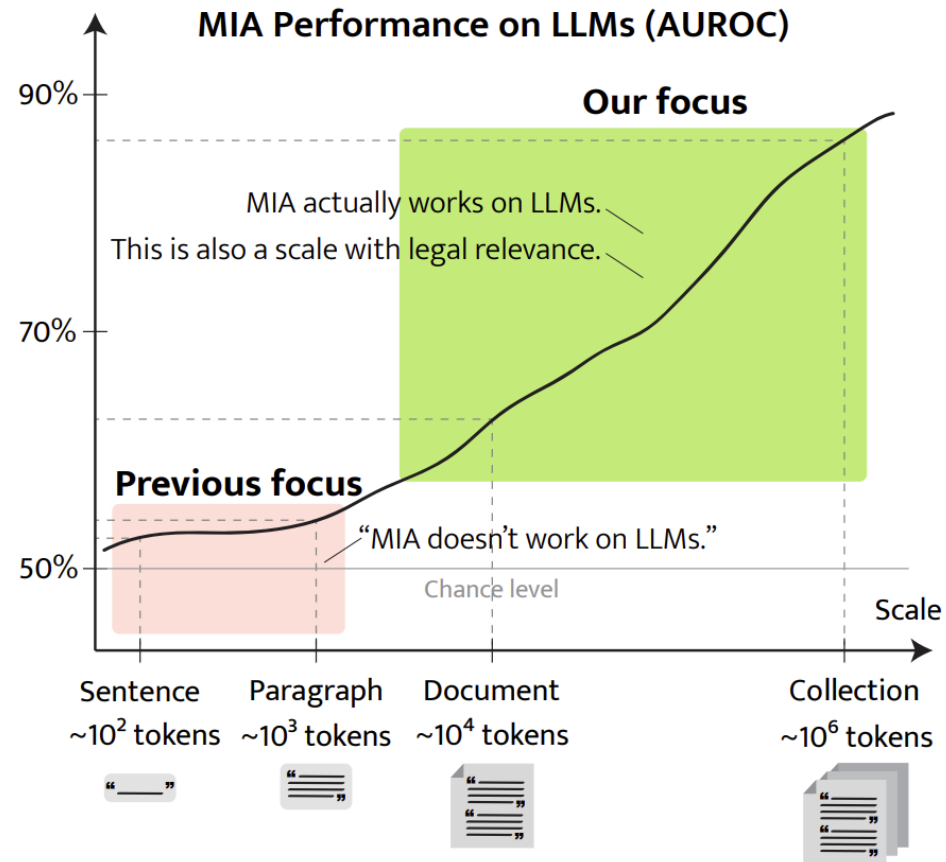
About the Dow Jones lawsuit

On Monday, we got sued by the Wall Street Journal and the New York Post. We were disappointed and surprised to see this.

Mark Zuckerberg gave Meta's Llama team the OK to train on copyrighted works, filing claims

Kyle Wiggers · 10:10 AM PST · January 9, 2025

Introduction to MIA on LLMs



What is MIA?

Membership Inference Attacks (MIAs) determine if a data sample was used to train a model.

Does MIAs work on LLMs?

- Initial works claim some success
- They build classifiers based on the perplexity
 - High perplexity → Non-member data
 - Low perplexity → Member data
- They create benchmarks using members and non-members based on the knowledge cut-off of the model

Does MIAs work on LLMs?

- Recent works show that evaluation methods based on cut-offs are flawed!
 - Cut-offs introduce temporal biases that even bag of words can classify
 - We can't evaluate MIA using cut-offs!
- These works propose to use LLMs trained on datasets with a train-test split
 - The Pile dataset and Pythia models
- In this setup, MIA do not work for LLMs
- Maini et al. (2024) show that aggregating MIA scores across multiple docs can yield successful dataset-level MIA!


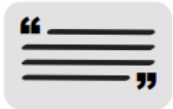


Objectives and Contributions

When and How MIA works on LLMs? → How many tokens do we need?

Extend and adapt dataset inference attack to any data scale

Analysis of MIA performance for various fine-tuning scenarios

Defining Data Scales for MIA

Scale	Definition	#Tokens
 Sentence	Natural definition	43 on avg.
 Paragraph	LLM context size	512, 1024, 2048, ...
 Document	Natural definition	14K on avg.
 Collection	Multiple docs	14K × #docs

LLM Training Scenarios

Pretraining

- Base models

Continual Learning

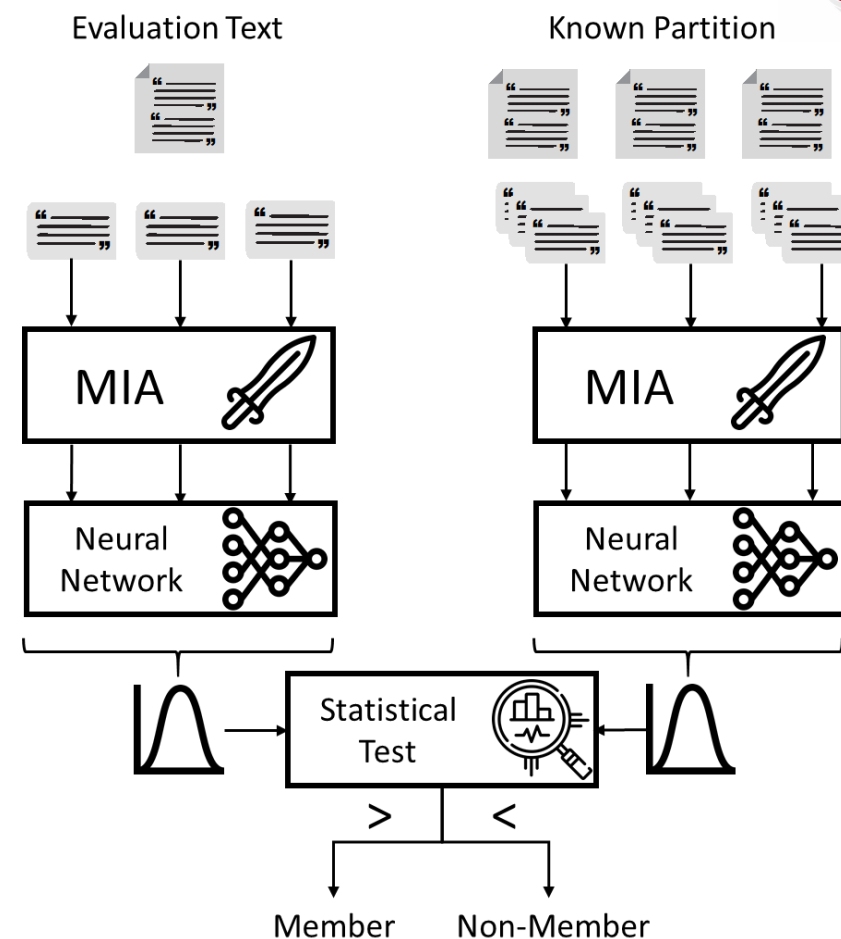
- Eg: Training base models on new Wikipedia dumps

Fine-tuning

- LLM training on end-tasks (e.g. CoT answering)

Method

- We extend Dataset Inference to any data scale
- For a document:
 - Collect paragraphs
 - Get their membership scores using current perplexity-based methods
 - Aggregate those scores to get a document score

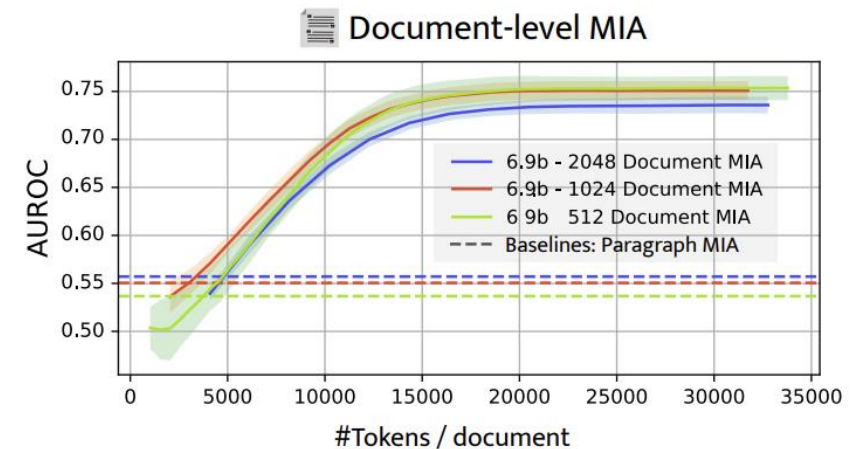
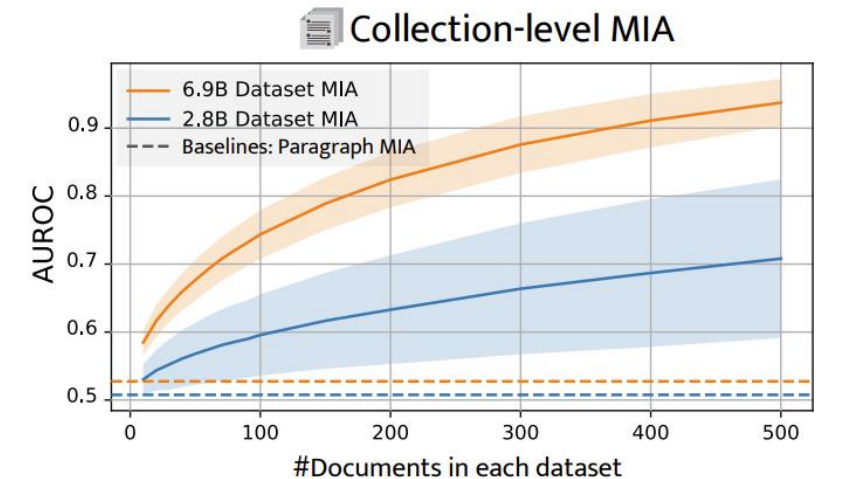


Method

- Pythia 2.8B, 6.9B, GPT-Neo 2.7B
- Members: training set of The Pile
- Non-members: test set of The Pile

Aggregating Text Subunits is Effective

- Collection-level MIA becomes more effective as we increase the number of docs in the collection
- Document-level MIA becomes more effective as the docs are longer
 - Doc MIA is effective from 10k tokens
 - First time Doc MIA is effective on LLMs!



Multi-Scale MIA

- As the data scale is smaller, the performance decreases
- MIA is effective at the right scale!

Data Scale	ArXiv	HackerNews	Wiki
Sentence	0.501 \pm 0.003	0.500 \pm 0.003	0.507 \pm 0.004
Paragraph	0.528 \pm 0.004	0.511 \pm 0.015	0.523 \pm 0.013
Document	0.697 \pm 0.060	0.513 \pm 0.040	0.560 \pm 0.011
Collection (500)	0.943 \pm 0.025	0.709 \pm 0.340	0.844 \pm 0.132

Recent Training Makes MIA More Effective

Dataset	Scale	Pretrained	Continual Learning
ArXiv	Paragraph	0.509 ± 0.006	0.587 ± 0.009
	Document	0.523 ± 0.01	0.582 ± 0.06
	Collection	0.718 ± 0.122	1.0 ± 0.0
GitHub	Paragraph	0.494 ± 0.009	0.559 ± 0.017
	Document	0.498 ± 0.01	0.579 ± 0.014
	Collection	0.479 ± 0.069	0.885 ± 0.064
Wikipedia	Paragraph	0.534 ± 0.015	0.577 ± 0.012
	Document	0.531 ± 0.019	0.590 ± 0.015
	Collection	0.665 ± 0.169	0.997 ± 0.007

Fine-Tuning

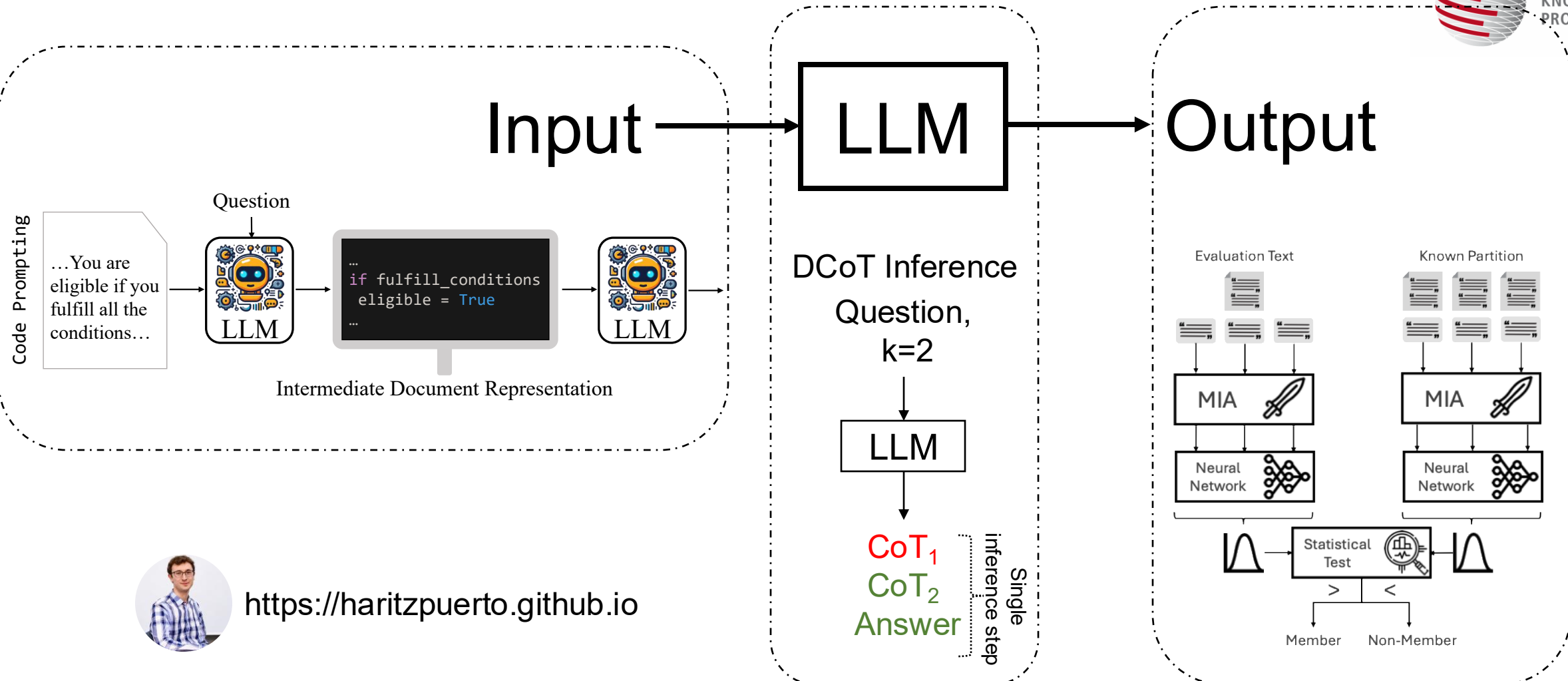
- Can we know if a fine-tuned model was trained on a specific **dataset/sentence**?
- We use DCoT-Phi-2 model
- MIA can be used as proof of data for fine-tuning LLMs, in contrast to claims by prior works

MIA	AUROC
Sentence	0.793 ± 0.024
Collection (20)	0.993 ± 0.012

Takeaways

- MIA can work on pretrained LLMs at the right scale (>10k tokens)
 - We achieve the first success on document-level MIA for pretrained LLMs
- Fine-tuning scenarios increase the effectiveness of MIA
- MIA is a suitable method to analyze test-set contamination

Conclusions



<https://haritzpuerto.github.io>