UNIVERSITÄT DES SAARLANDES

On the Ability of Transformers to Learn Complex Reasoning

Michael Hahn

RLeap Symposium 2026

SIC Saarland Informatics Campus

Emmy Noether-Programm DFG Deutsche Forschungsgemeinschaft

ZUSE SCHOOL ELIZA

ellis

SCI AM

AI may have just solved a million-dollar math problem. The field will never be the same

A mathematician compared the feat to the history-making chess competition in which IBM's Deep Blue computer beat Garry Kasparov in 1997

OpenAI makes breakthrough on 80-year-old maths problem

Company says work on Paul Erdős planar unit distance problem shows advance in AI reasoning

The Guardian

Quanta magazine

ARTIFICIAL IN

AI Has Solved One of Math's $1 Million Millennium Prize Problems

19 |

Mathematicians at OpenAI showed that the Navier–Stokes equations, which describe how fluids flow, can sometimes "blow up." But the massive result is not without controversy.

Can LLMs Track States?

cf. Kim & Schuster, ACL 2023; Liu et al, NeurIPS 2023; Merrill et al, ICML 2025; Grazzi et al, ICLR 2025; and others

Can LLMs Track States?

Mary

Josh

Kim

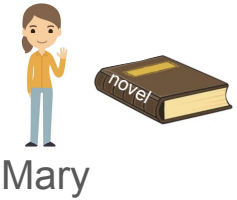
Paul

Ruth

cf. Kim & Schuster, ACL 2023; Liu et al, NeurIPS 2023; Merrill et al, ICML 2025; Grazzi et al, ICLR 2025; and others

Can LLMs Track States?

Paul had an encyclopedia.
Mary had a novel.



Mary

Josh

Kim

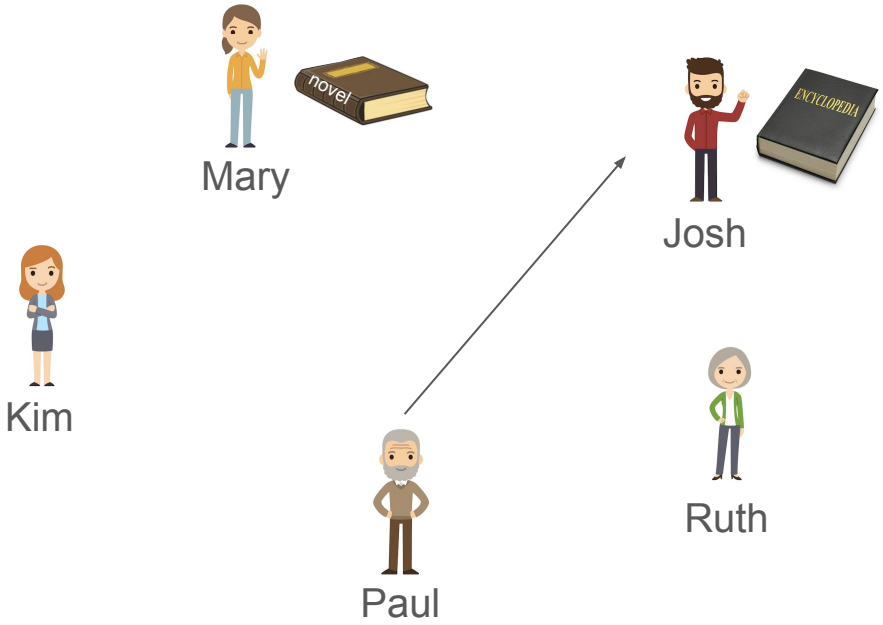
Paul

Ruth

cf. Kim & Schuster, ACL 2023; Liu et al, NeurIPS 2023; Merrill et al, ICML 2025; Grazzi et al, ICLR 2025; and others

Can LLMs Track States?

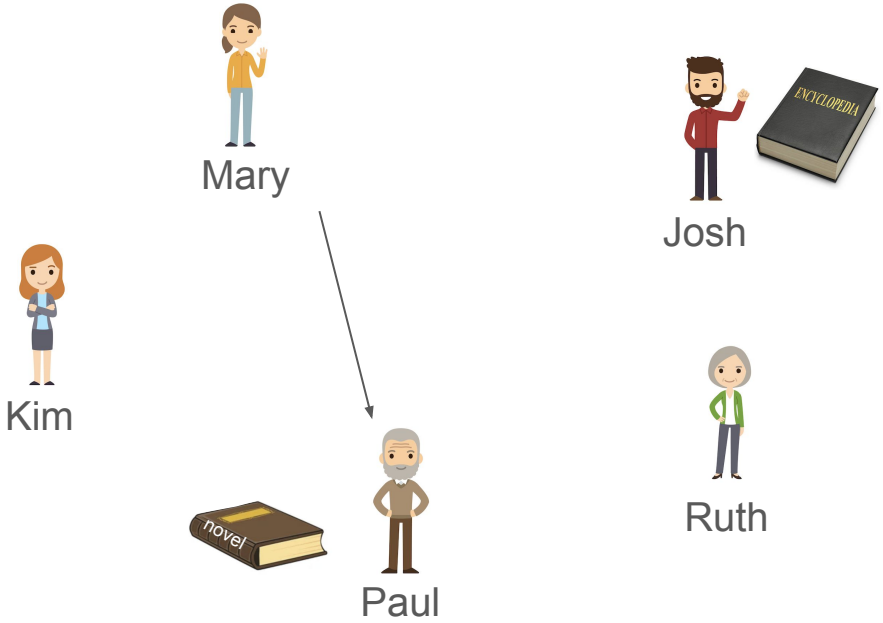
Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.



cf. Kim & Schuster, ACL 2023; Liu et al, NeurIPS 2023; Merrill et al, ICML 2025; Grazzi et al, ICLR 2025; and others

Can LLMs Track States?

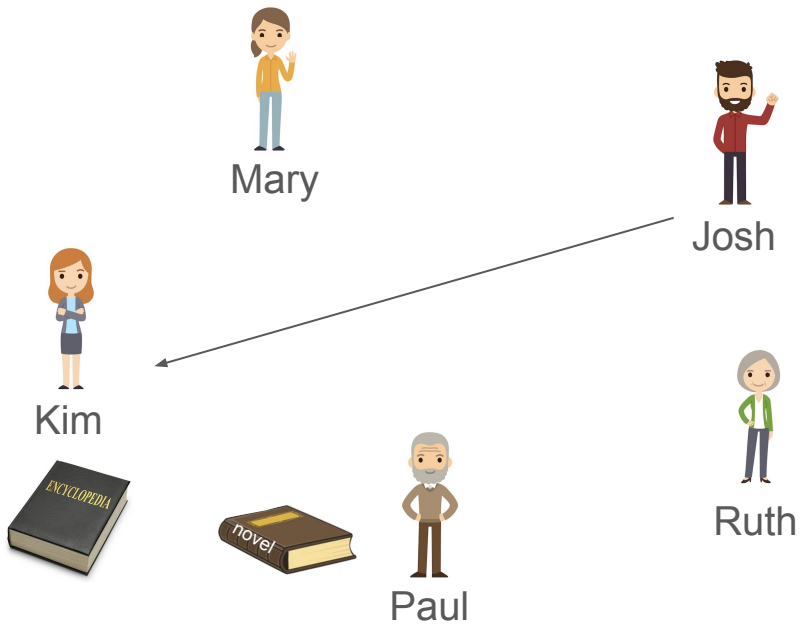
Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.



cf. Kim & Schuster, ACL 2023; Liu et al, NeurIPS 2023; Merrill et al, ICML 2025; Grazzi et al, ICLR 2025; and others

Can LLMs Track States?

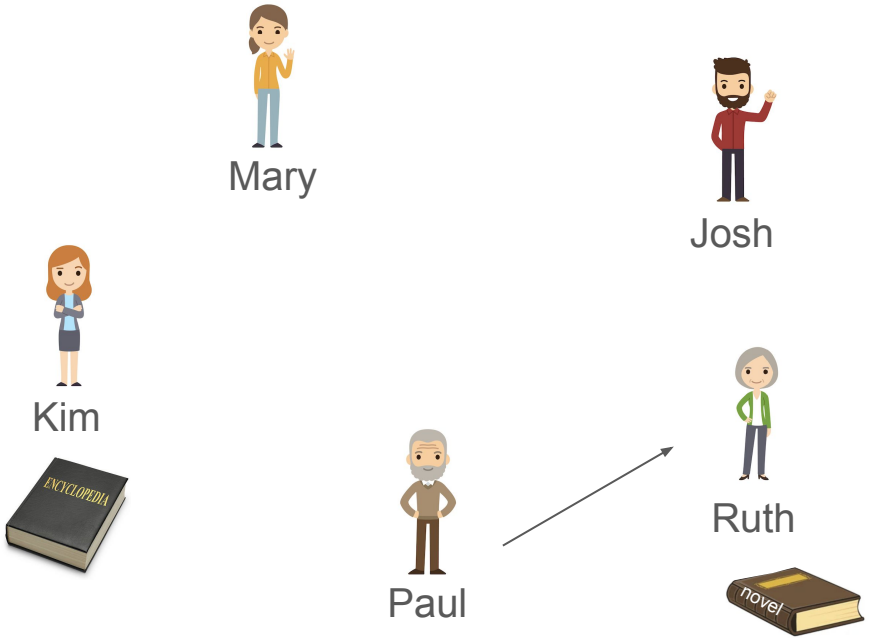
Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.
Josh gave his book to Kim.



cf. Kim & Schuster, ACL 2023; Liu et al, NeurIPS 2023; Merrill et al, ICML 2025; Grazzi et al, ICLR 2025; and others

Can LLMs Track States?

Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.
Josh gave his book to Kim.
Paul gave his book to Ruth.



cf. Kim & Schuster, ACL 2023; Liu et al, NeurIPS 2023; Merrill et al, ICML 2025; Grazzi et al, ICLR 2025; and others

Can LLMs Track States?

Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.
Josh gave his book to Kim.
Paul gave his book to Ruth.
Who has the encyclopedia?


Mary


Josh


Kim


Paul


Ruth

cf. Kim & Schuster, ACL 2023; Liu et al, NeurIPS 2023; Merrill et al, ICML 2025; Grazzi et al, ICLR 2025; and others

Can LLMs Track States?

Mary

Josh

Kim

Paul

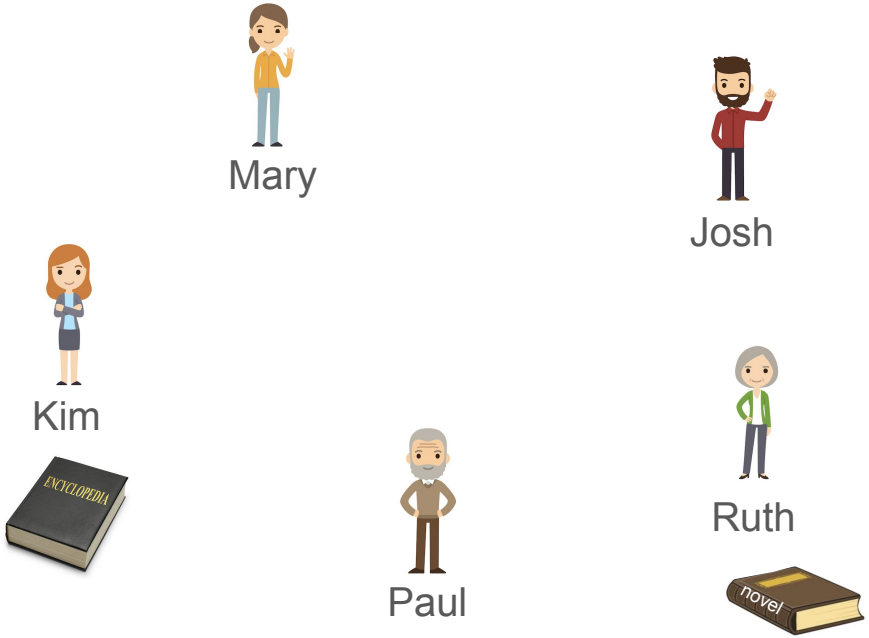
Ruth

Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.
Josh gave his book to Kim.
Paul gave his book to Ruth.
Who has the encyclopedia?

Answer: Kim has the encyclopedia.

cf. Kim & Schuster, ACL 2023; Liu et al, NeurIPS 2023; Merrill et al, ICML 2025; Grazzi et al, ICLR 2025; and others

Can LLMs Track States?



Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.
Josh gave his book to Kim.
Paul gave his book to Ruth.
Who has the encyclopedia?

Answer: Kim has the encyclopedia.

Ruth has the encyclopedia.

cf. Kim & Schuster, ACL 2023; Liu et al, NeurIPS 2023; Merrill et al, ICML 2025; Grazzi et al, ICLR 2025; and others

Can LLMs Track States?

Mary

Josh

Kim

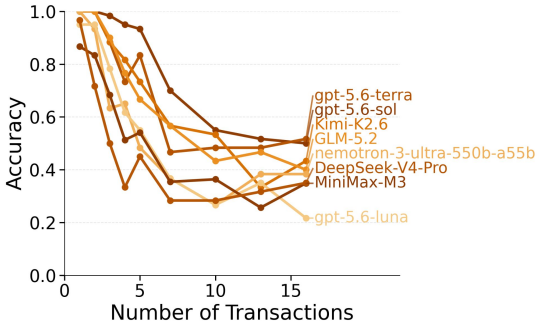
Paul

Ruth

Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.
Josh gave his book to Kim.
Paul gave his book to Ruth.
Who has the encyclopedia?

Answer: Kim has the encyclopedia.

Ruth has the encyclopedia.



straight answer, no CoT

Can LLMs Track States?

Mary

Josh

Kim

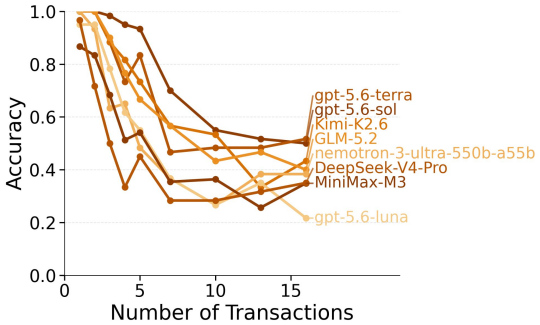
Paul

Ruth

Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.
Josh gave his book to Kim.
Paul gave his book to Ruth.
Who has the encyclopedia?

Answer: Kim has the encyclopedia.

Ruth has the encyclopedia.



straight answer, no CoT

Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.
Josh gave his book to Kim.
Paul gave his book to Ruth.
Who has the encyclopedia?
Please think step by step.

Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.
Josh gave his book to Kim.
Paul gave his book to Ruth.
Who has the encyclopedia?
Please think step by step.

Got it — let's carefully trace the movements step by step:

1. **Initial ownership:**
   - Paul → *encyclopedia*
   - Mary → *novel*
2. **Paul gave his book (encyclopedia) to Josh:**
   - Josh → *encyclopedia*
   - Paul → nothing
3. **Mary gave her book (novel) to Paul:**
   - Paul → *novel*
   - Mary → nothing
4. **Josh gave his book (encyclopedia) to Kim:**
   - Kim → *encyclopedia*
   - Josh → nothing
5. **Paul gave his book (novel) to Ruth:**
   - Ruth → *novel*
   - Paul → nothing

✅ **Final ownership:**

- Kim → *encyclopedia*
- Ruth → *novel*
- Paul, Mary, Josh → nothing

correct!

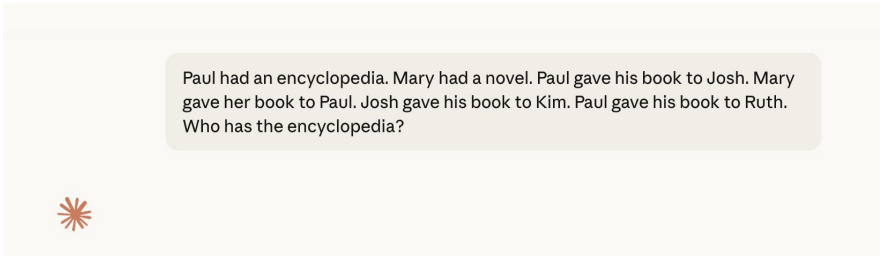
Answer: Kim has the encyclopedia.

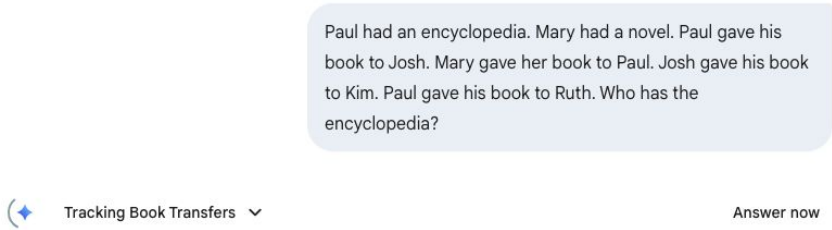
ChatGPT-5 Instant, Aug 31, 2025.

Paul had an encyclopedia. Mary had a novel. Paul gave his book to Josh. Mary gave her book to Paul. Josh gave his book to Kim. Paul gave his book to Ruth. Who has the encyclopedia?

Thinking ›

Answer now





Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.
Josh gave his book to Kim.
Paul gave his book to Ruth.
Who has the encyclopedia?
Please think step by step.

ChatGPT-5 Instant, Aug 31, 2025.

Got it — let's carefully trace the movements step by step:

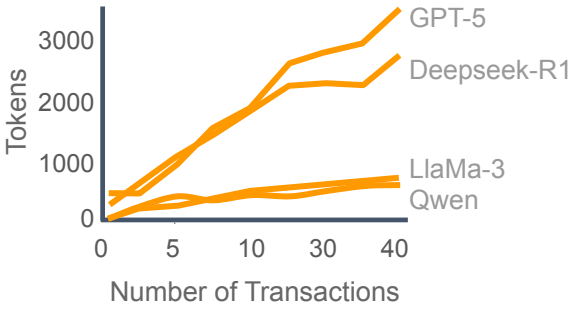
1. **Initial ownership:**
   - Paul → *encyclopedia*
   - Mary → *novel*
2. **Paul gave his book (encyclopedia) to Josh:**
   - Josh → *encyclopedia*
   - Paul → nothing
3. **Mary gave her book (novel) to Paul:**
   - Paul → *novel*
   - Mary → nothing
4. **Josh gave his book (encyclopedia) to Kim:**
   - Kim → *encyclopedia*
   - Josh → nothing
5. **Paul gave his book (novel) to Ruth:**
   - Ruth → *novel*
   - Paul → nothing

✅ **Final ownership:**

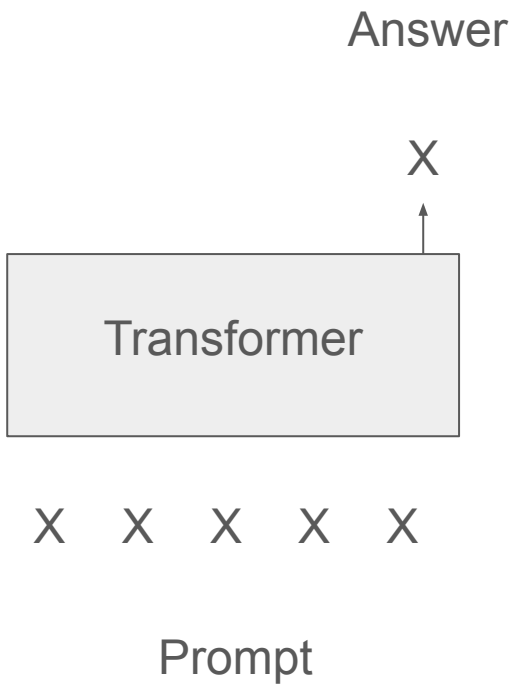
- Kim → *encyclopedia*
- Ruth → *novel*
- Paul, Mary, Josh → nothing

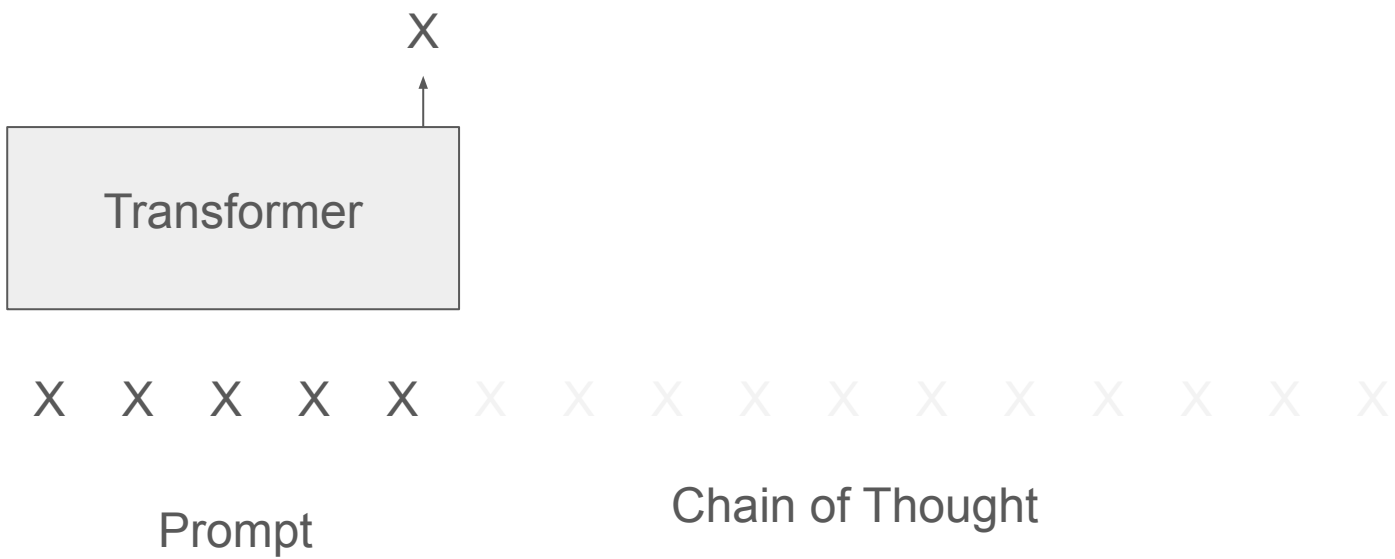
correct!

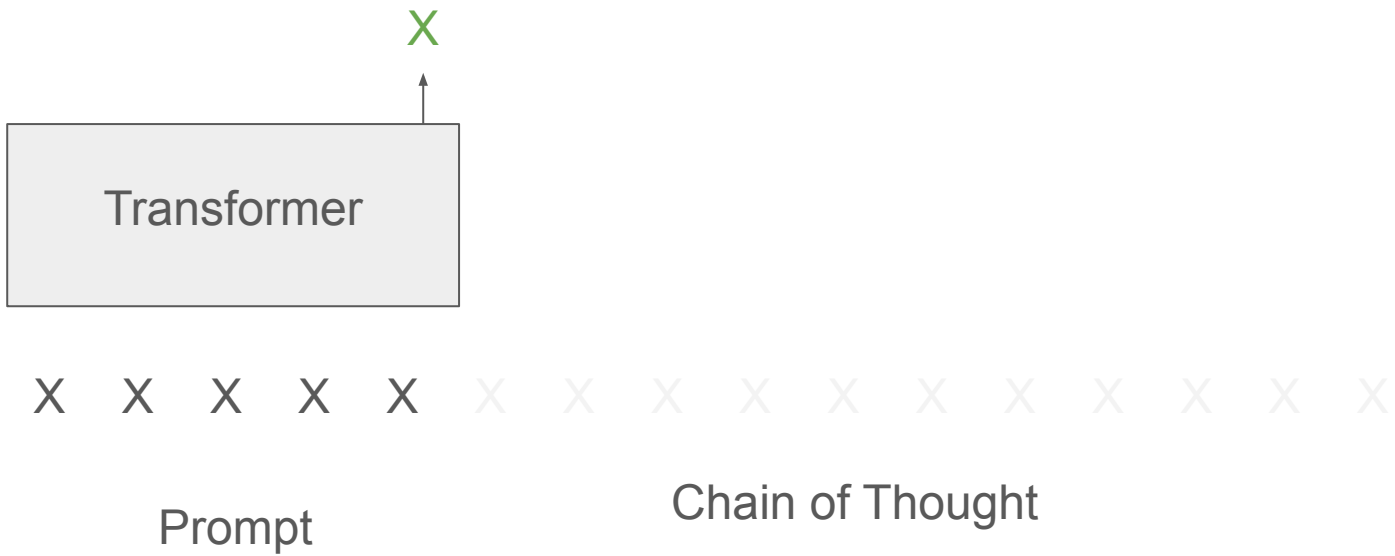
Answer: Kim has the encyclopedia.

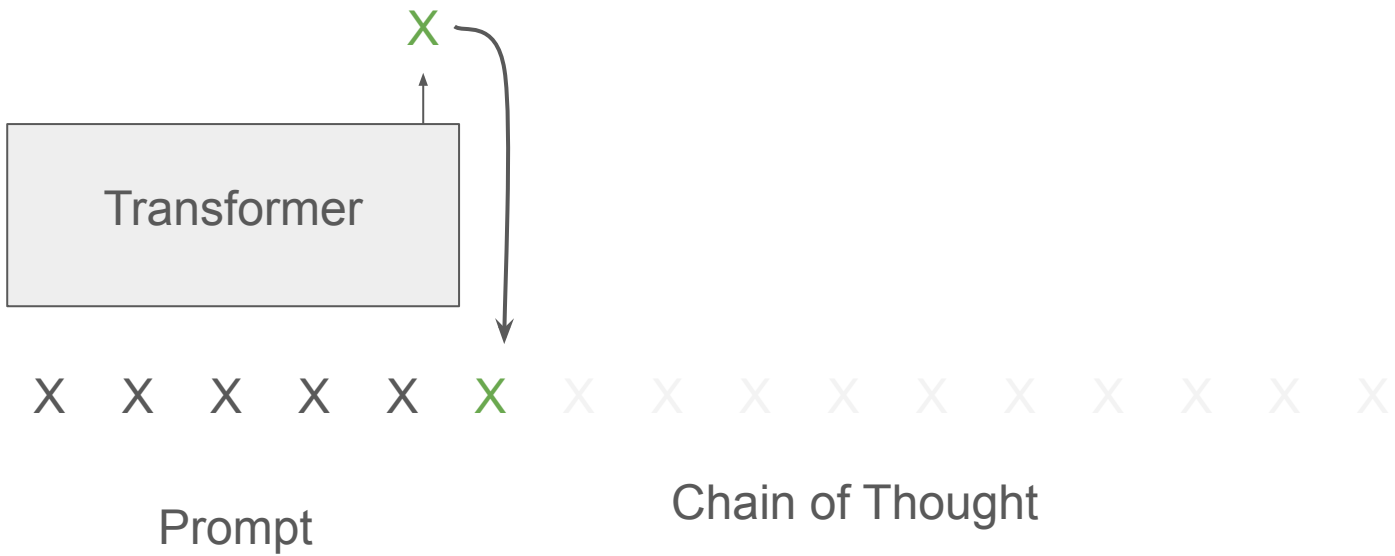


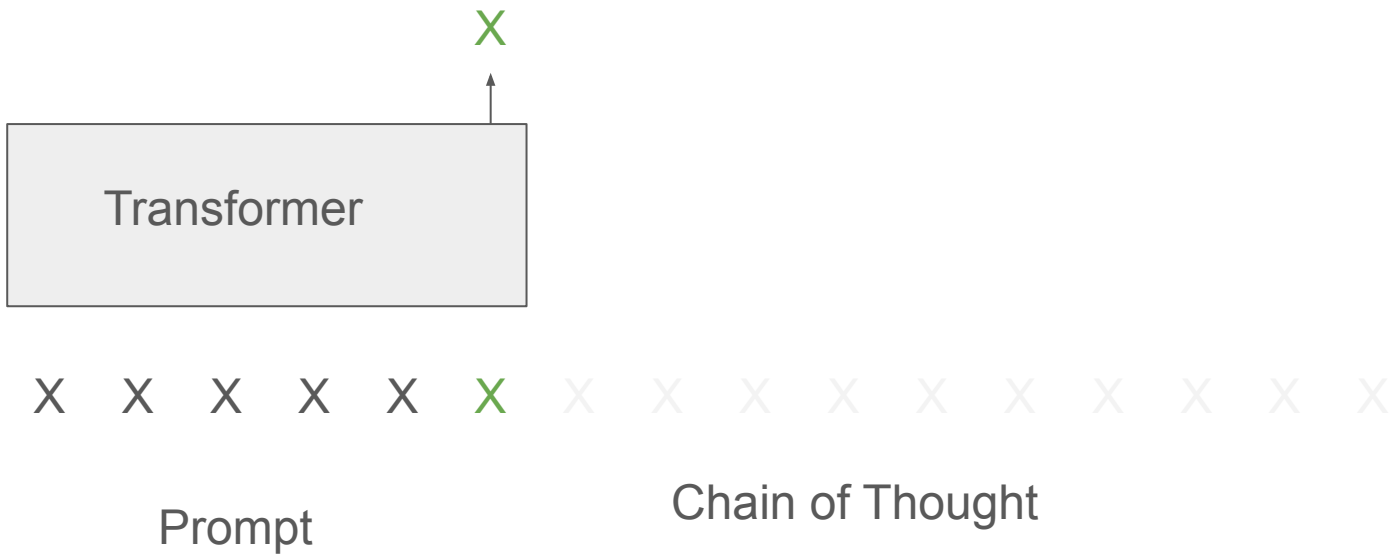
Models:
gpt-5-mini-2025-08-07,
meta-llama/Llama-3.3-70B-Instruct-Turbo,
deepseek-ai/DeepSeek-R1-Distill-Llama-70B,
Qwen/Qwen2.5-72B-Instruct-Turbo

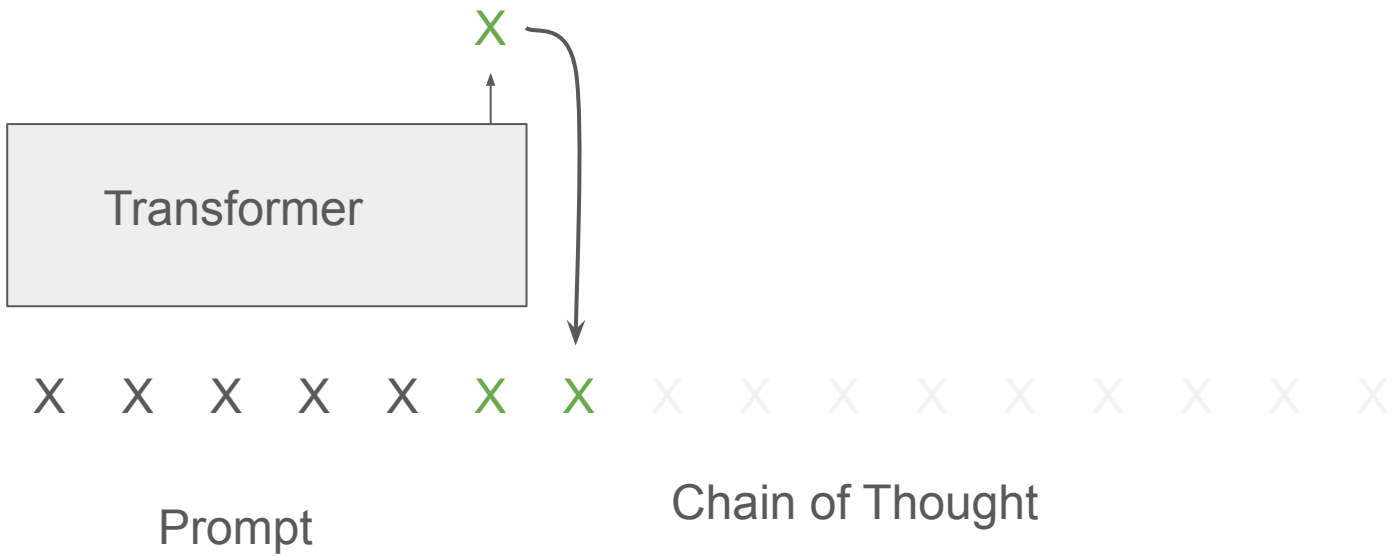

Answer
X
Transformer
X X X X X
Prompt

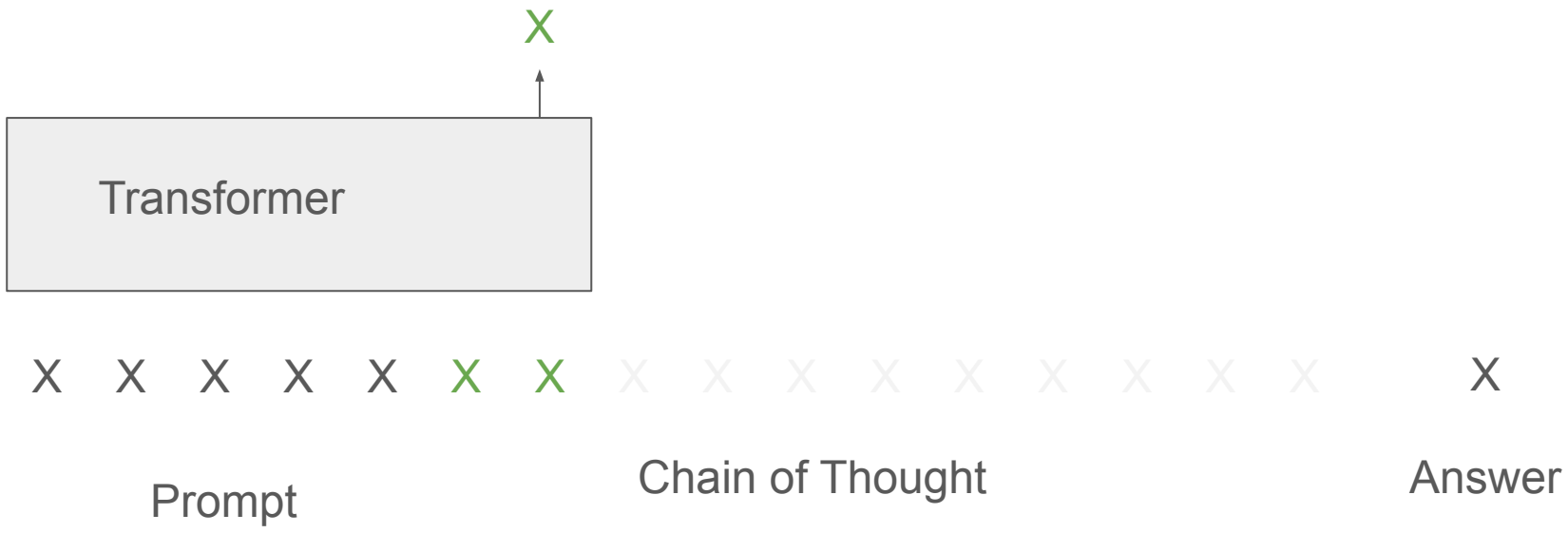

X
Transformer
X X X X X X X X X X X X X X X X
Prompt
Chain of Thought

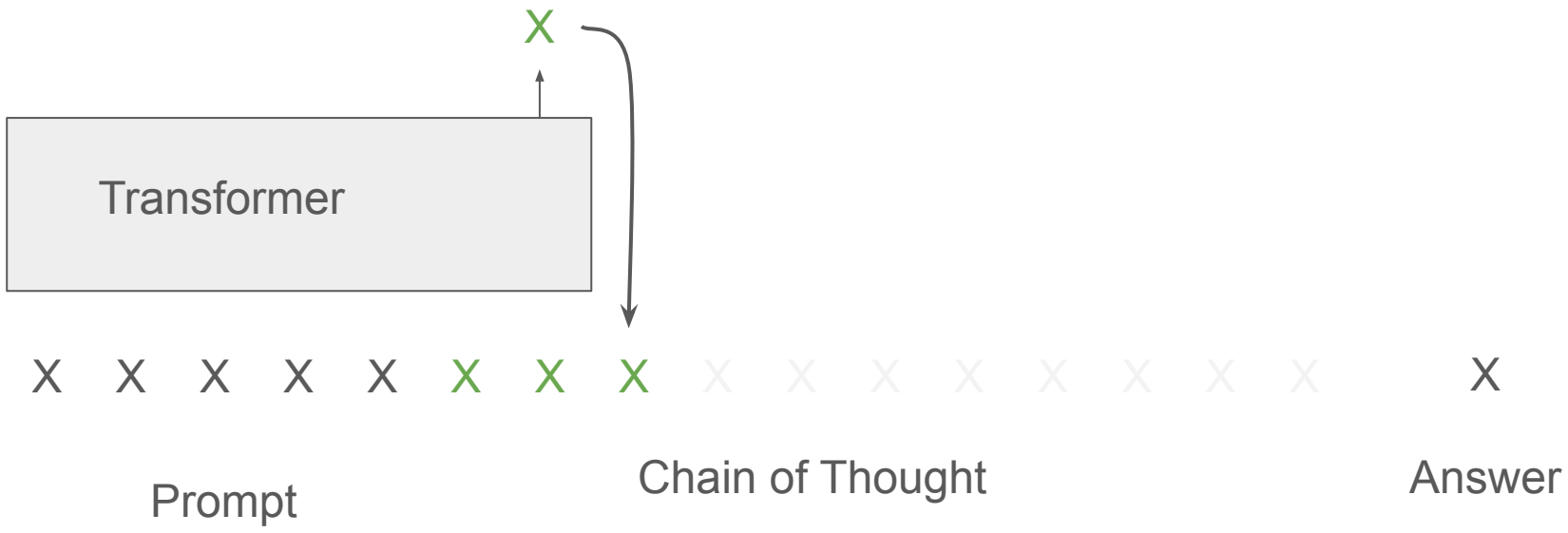

X
Transformer
X X X X X X X X X X X X X X X X
Prompt
Chain of Thought

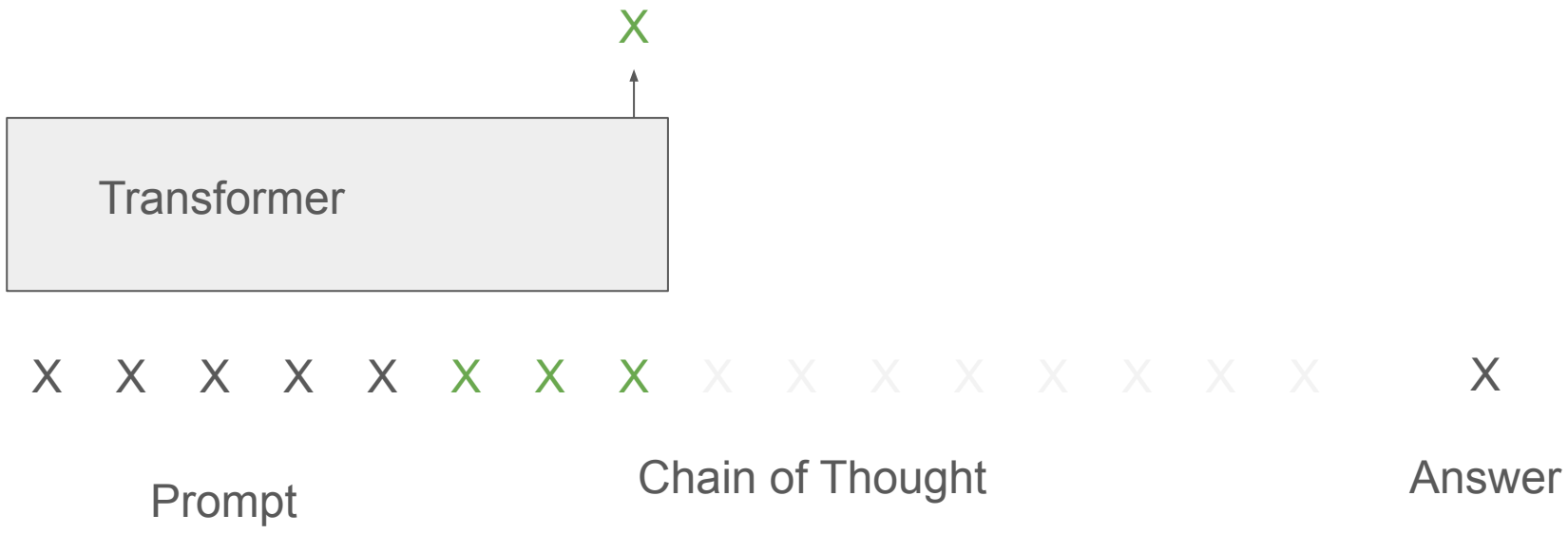

X
Transformer
X X X X X X
Prompt
Chain of Thought

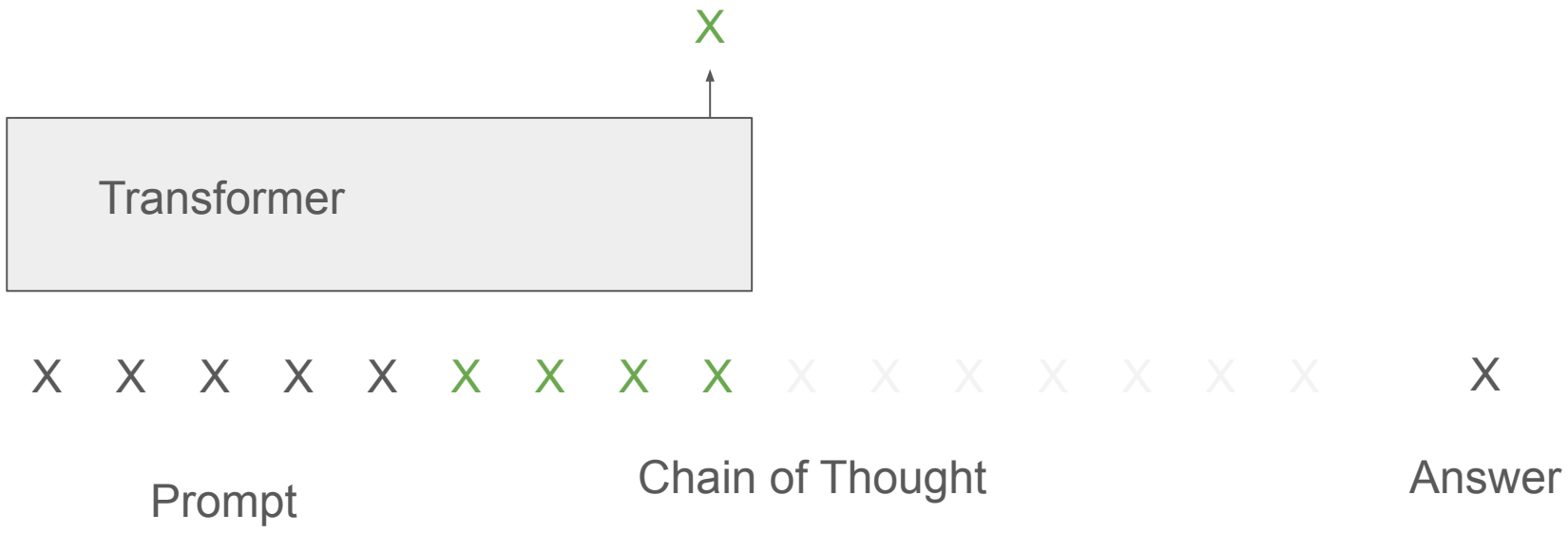

X
Transformer
X X X X X X X X X X X X X X X X
Prompt
Chain of Thought

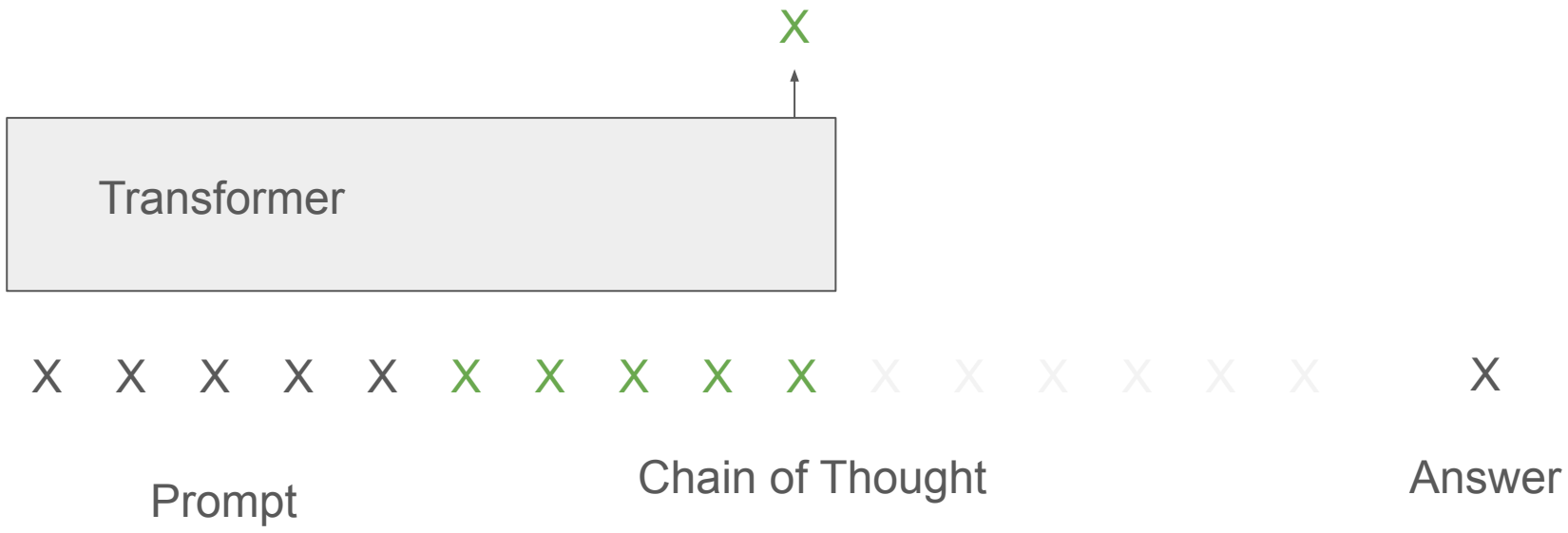

X
Transformer
X X X X X X X
Prompt
Chain of Thought

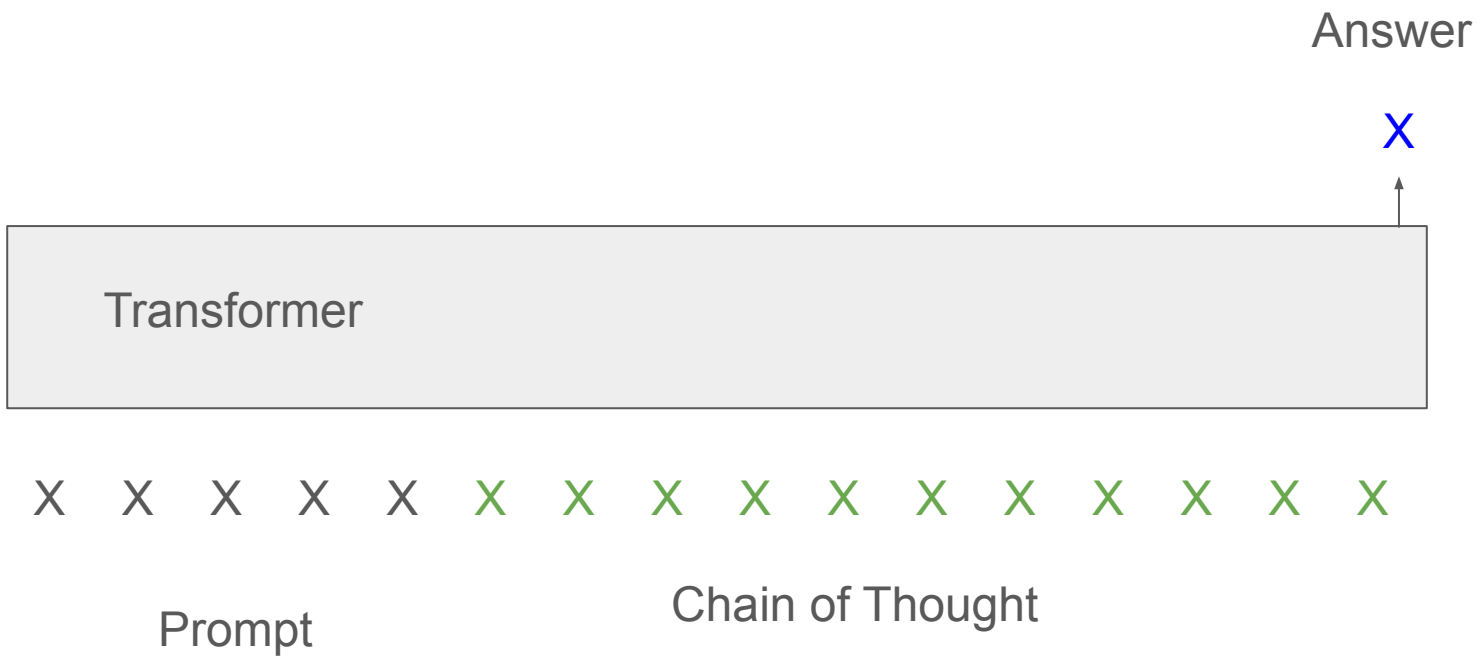

X
Transformer
X X X X X X X X X X X X X X X X
Prompt
Chain of Thought
Answer

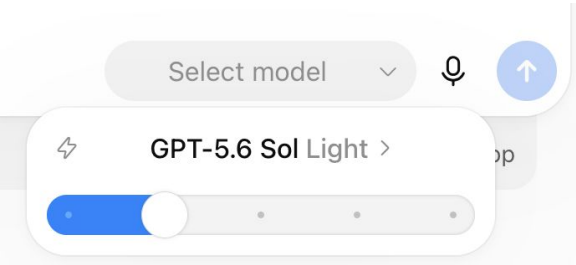

X
Transformer
X X X X X X X X X X X X X X X X X
Prompt
Chain of Thought
Answer

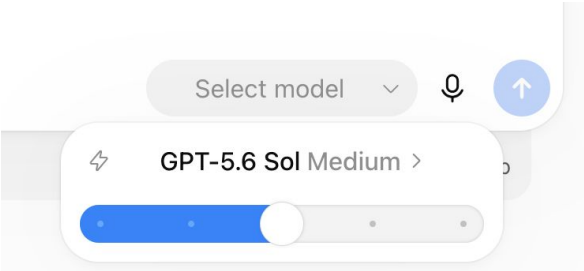

X
Transformer
X X X X X X X X X X X X X X X X X
Prompt
Chain of Thought
Answer


X
Transformer
X X X X X X X X X X X X X X X X
Prompt
Chain of Thought
Answer


X
Transformer
X X X X X X X X X X X X X X X X
Prompt
Chain of Thought
Answer


Answer
X
Transformer
X X X X X X X X X X X X X X X X
Prompt
Chain of Thought


Select model
GPT-5.6 Sol Light


Select model
GPT-5.6 Sol Medium

Please answer very briefly, and without external resources (e.g. Python interpreter). What is 61157*4555?

The result of 61157×4555 is **278657635**.

Please answer very briefly, and without external resources (e.g. Python interpreter). What is 61157*4555?

The result of 61157×4555 is **278657635**.

incorrect

actual: 278570135

See also:

for GPT4 zero shot: Dziri et al, NeurIPS 2023.

for: o3-mini: https://x.com/yuntiandeng/status/1889704768135905332

for DeepSeek R1: https://x.com/nouhadziri/status/1886423897567805835

Model
GPT-5.6 Sol
Claude Opus 5†
Grok 4.3
DeepSeek V4 Pro
Kimi K3
Qwen 3.7 Max

https://ood.dev/posts/calculator/

Model	3×3
GPT-5.6 Sol	99.2%
Claude Opus 5†	95.4%
Grok 4.3	78.4%
DeepSeek V4 Pro	99.6%
Kimi K3	97.4%
Qwen 3.7 Max	100.0%

https://ood.dev/posts/calculator/

Model	3×3	4×4
GPT-5.6 Sol	99.2%	69.4%
Claude Opus 5†	95.4%	78.0%
Grok 4.3	78.4%	25.2%
DeepSeek V4 Pro	99.6%	55.2%
Kimi K3	97.4%	64.8%
Qwen 3.7 Max	100.0%	99.0%

https://ood.dev/posts/calculator/

Model	3×3	4×4	5×5
GPT-5.6 Sol	99.2%	69.4%	21.4%
Claude Opus 5†	95.4%	78.0%	13.0%
Grok 4.3	78.4%	25.2%	2.4%
DeepSeek V4 Pro	99.6%	55.2%	26.4%
Kimi K3	97.4%	64.8%	6.4%
Qwen 3.7 Max	100.0%	99.0%	87.4%

https://ood.dev/posts/calculator/

Model	3×3	4×4	5×5	6×6
GPT-5.6 Sol	99.2%	69.4%	21.4%	5.2%
Claude Opus 5†	95.4%	78.0%	13.0%	1.0%
Grok 4.3	78.4%	25.2%	2.4%	0.2%
DeepSeek V4 Pro	99.6%	55.2%	26.4%	2.2%
Kimi K3	97.4%	64.8%	6.4%	0.2%
Qwen 3.7 Max	100.0%	99.0%	87.4%	55.2%

https://ood.dev/posts/calculator/

Model	3×3	4×4	5×5	6×6	7×7
GPT-5.6 Sol	99.2%	69.4%	21.4%	5.2%	0.0%
Claude Opus 5†	95.4%	78.0%	13.0%	1.0%	0.0%
Grok 4.3	78.4%	25.2%	2.4%	0.2%	0.0%
DeepSeek V4 Pro	99.6%	55.2%	26.4%	2.2%	0.0%
Kimi K3	97.4%	64.8%	6.4%	0.2%	0.0%
Qwen 3.7 Max	100.0%	99.0%	87.4%	55.2%	7.4%

https://ood.dev/posts/calculator/

$$61157 \times 4555 = 61157 \times (4000 + 500 + 50 + 5)$$

$$= 61157 \cdot 4000 + 61157 \cdot 500 + 61157 \cdot 50 + 61157 \cdot 5$$

$$= 244{,}628{,}000 + 30{,}578{,}500 + 3{,}057{,}850 + 305{,}785$$

$$= 278{,}570{,}135$$

So the exact result is **278,570,135**.

61157 × 4555

Let me work through this step by step:

61157 × 4555

= 61157 × (4000 + 500 + 50 + 5)

61157 × 4000 = 244,628,000

61157 × 500 = 30,578,500

61157 × 50 = 3,057,850

61157 × 5 = 305,785

Now adding them up:

244,628,000 + 30,578,500 = 275,206,500 275,206,535 + 3,057,850 = 278,264,350 278,264,350 + 305,785 = **278,570,135**

To calculate $61,157 \times 4,555$ without a calculator, we can use the long multiplication method by breaking down $4,555$ into its component parts: $4,000 + 500 + 50 + 5$.

Step 1: Calculate the Partial Products

1. **Multiply by 5:**
   $61,157 \times 5 = 305,785$
   (Calculation: $60,000 \times 5 = 300,000$; $1,000 \times 5 = 5,000$; $157 \times 5 = 785$)
2. **Multiply by 50:**
   $61,157 \times 50 = 3,057,850$
   (This is simply the first result shifted one decimal place to the left)
3. **Multiply by 500:**
   $61,157 \times 500 = 30,578,500$
   (This is the first result shifted two decimal places to the left)
4. **Multiply by 4,000:**
   $61,157 \times 4,000 = 244,628,000$
   ($61,157 \times 4 = 244,628$, then add three zeros)

Step 2: Sum the Partial Products

Now, we add all the results together:

\begin{array}{r@{\quad}l}
244,628,000 & (61,157 \times 4,000) \\
30,578,500 & (61,157 \times 500) \\
3,057,850 & (61,157 \times 50) \\
+ \quad 305,785 & (61,157 \times 5) \\
\hline
278,570,135 &
\end{array}

Broken down addition:

- $244,628,000 + 30,578,500 = 275,206,500$
- $275,206,500 + 3,057,850 = 278,264,350$
- $278,264,350 + 305,785 = 278,570,135$

Final Answer:
$61,157 \times 4,555 = 278,570,135$

Transformer Expressivity

Transformer Expressivity

Individual forward pass

inside TC^0 or AC^0

(depending on precision assumptions)

e.g., Hahn 2020; Hao et al 2022; Merrill and Sabharwal 2022

Transformer Expressivity

Individual forward pass

inside TC^0 or AC^0

(depending on precision assumptions)

e.g., Hahn 2020; Hao et al 2022; Merrill and Sabharwal 2022

Chain of Thought

Turing-complete

chain length = O(TM run time)

e.g., Perez et al 2019

Transformer Expressivity

Individual forward pass

inside TC^0 or AC^0

(depending on precision assumptions)

e.g., Hahn 2020; Hao et al 2022; Merrill and Sabharwal 2022

Chain of Thought

Turing-complete

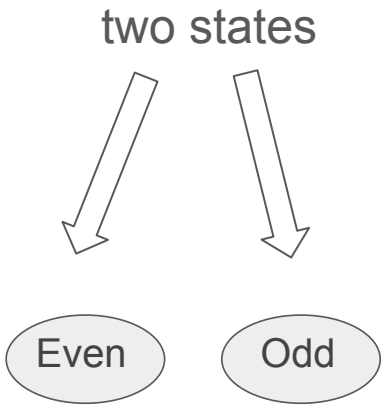
chain length = O(TM run time)

e.g., Perez et al 2019

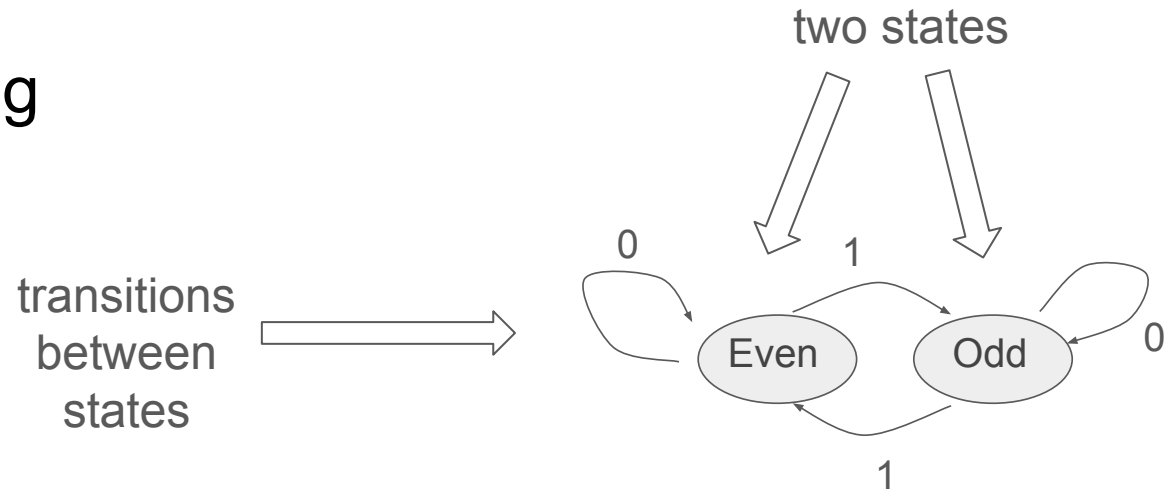
...but what about Learnability?

State Tracking

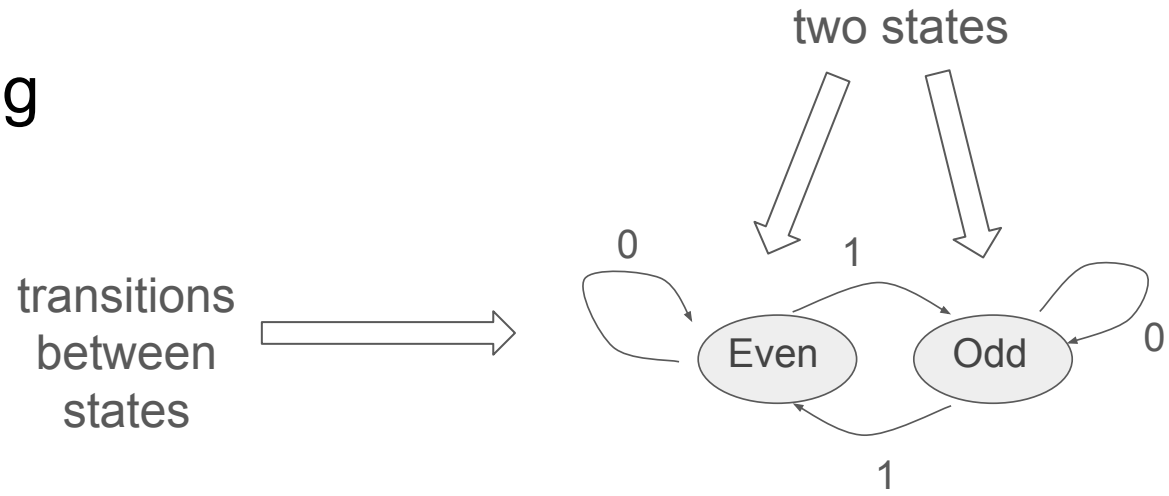
State Tracking



State Tracking

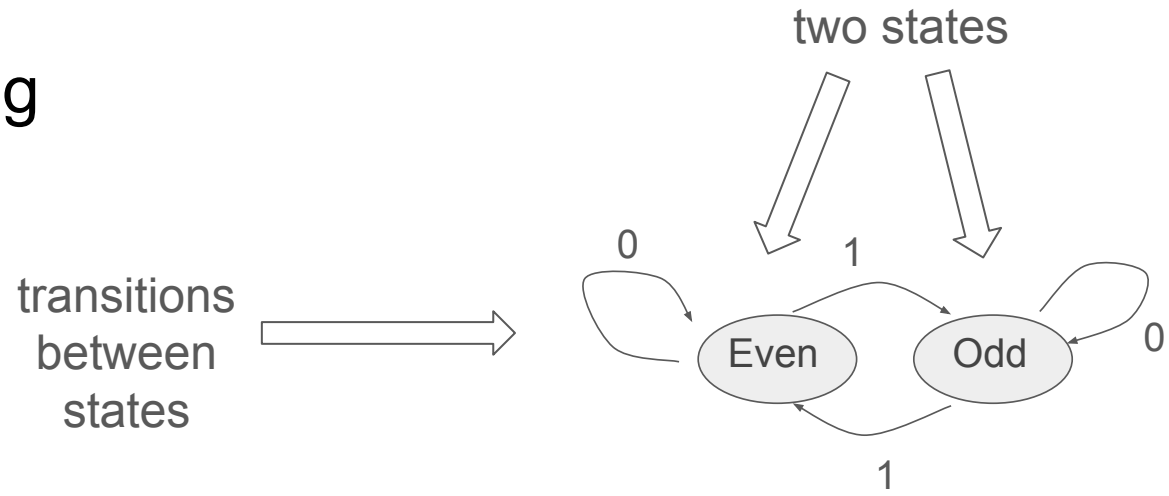


State Tracking



PARITY: Is the number of 1s even?

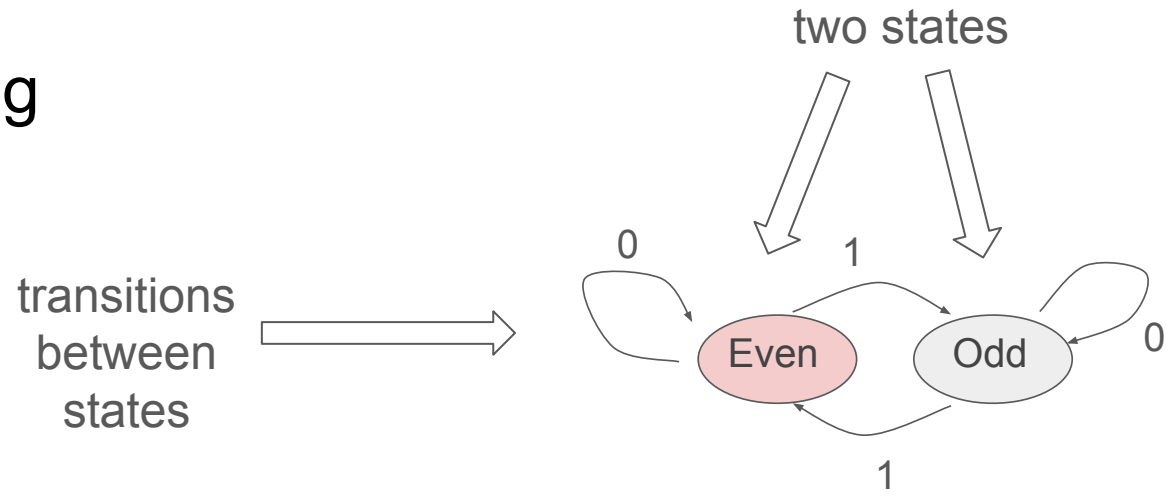
State Tracking



0 1 0 1 1

PARITY: Is the number of 1s even?

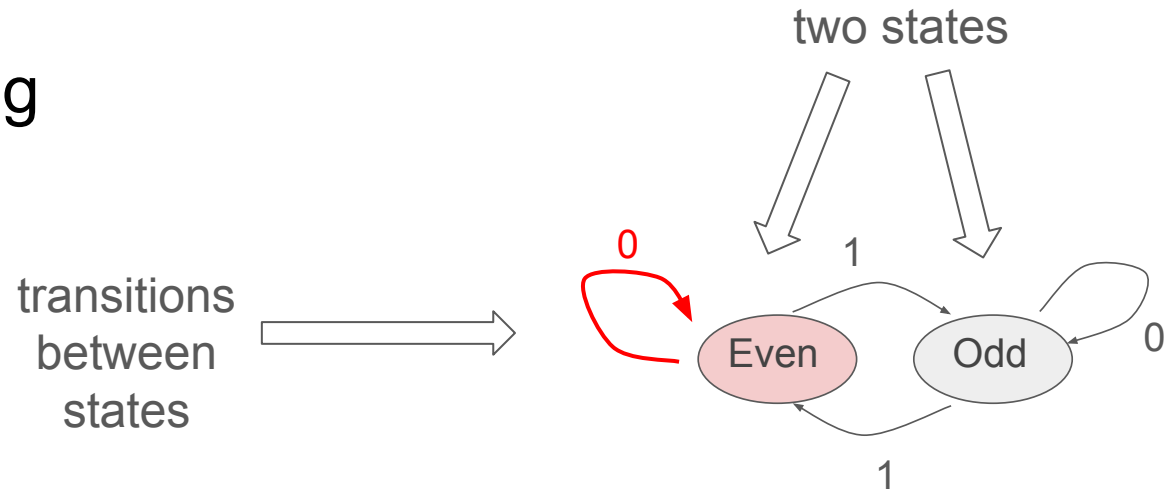
State Tracking



0 1 0 1 1

PARITY: Is the number of 1s even?

State Tracking

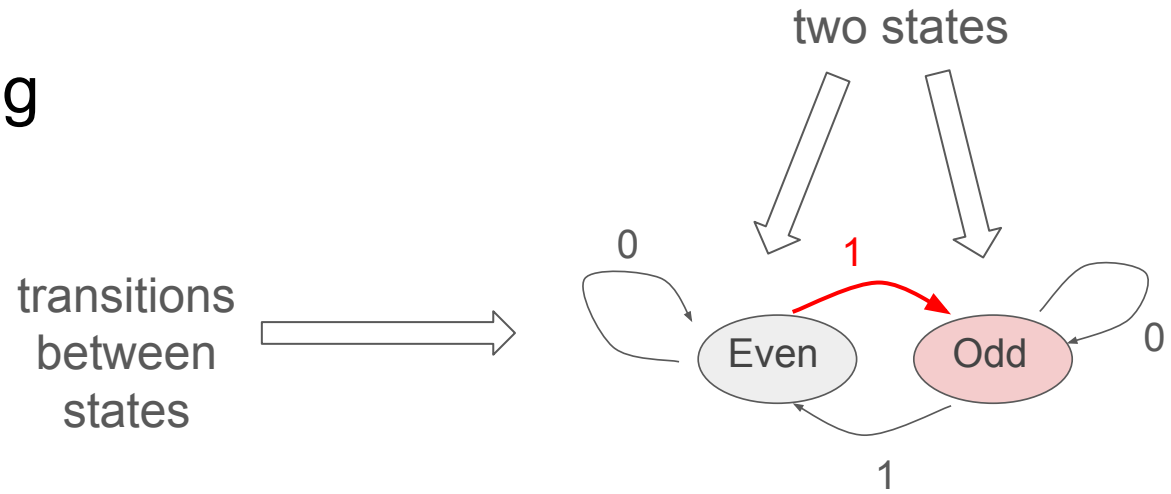


Even

0 1 0 1 1

PARITY: Is the number of 1s even?

State Tracking

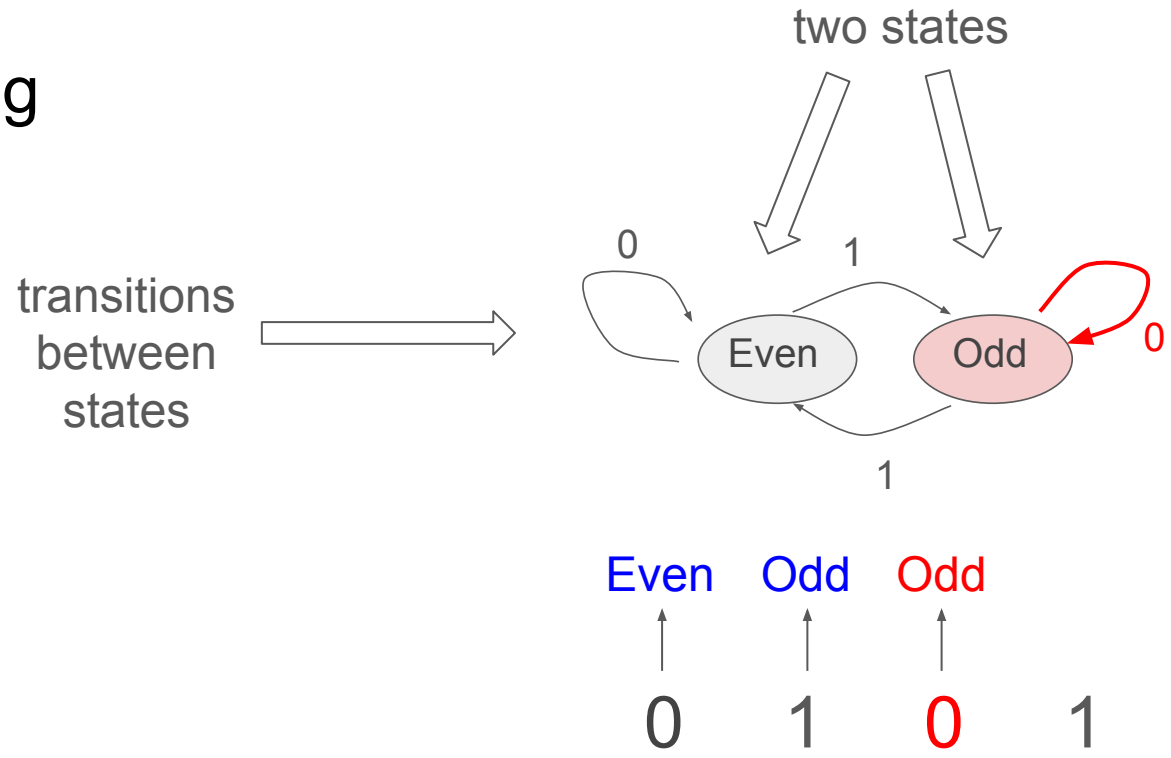


Even Odd

0 1 0 1 1

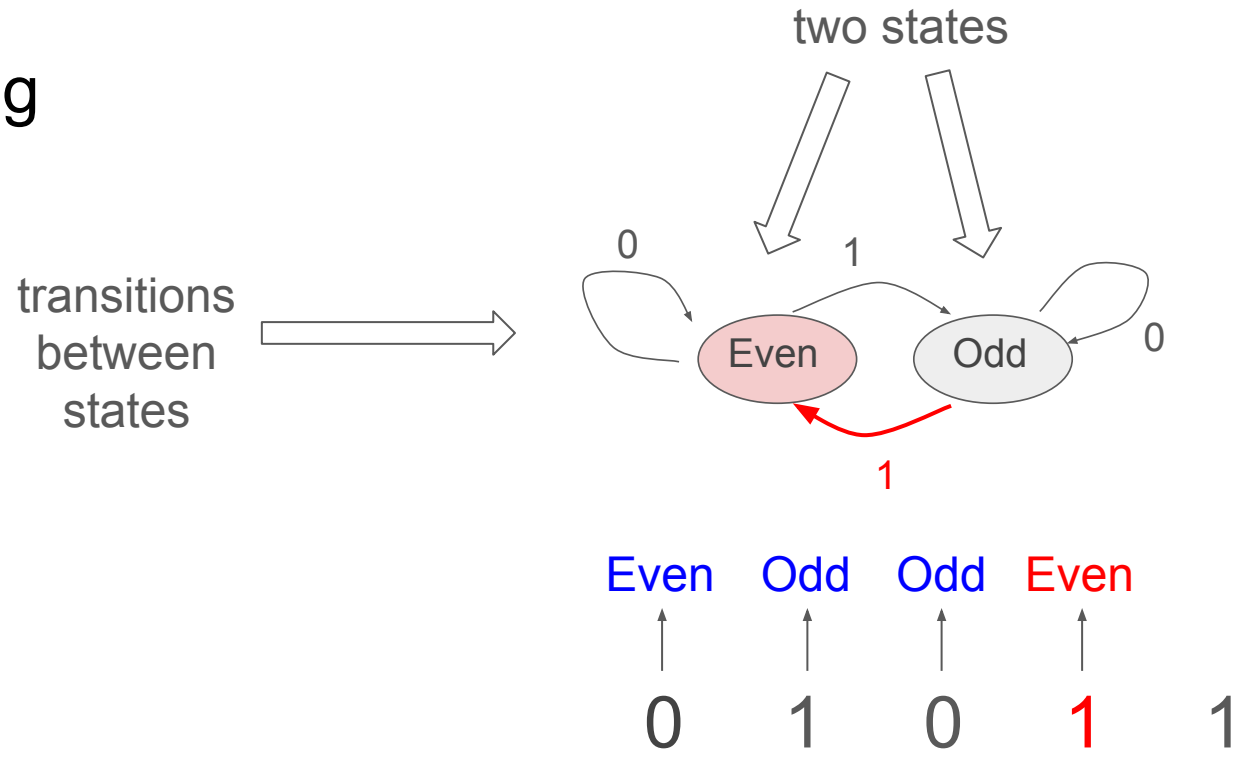
PARITY: Is the number of 1s even?

State Tracking



PARITY: Is the number of 1s even?

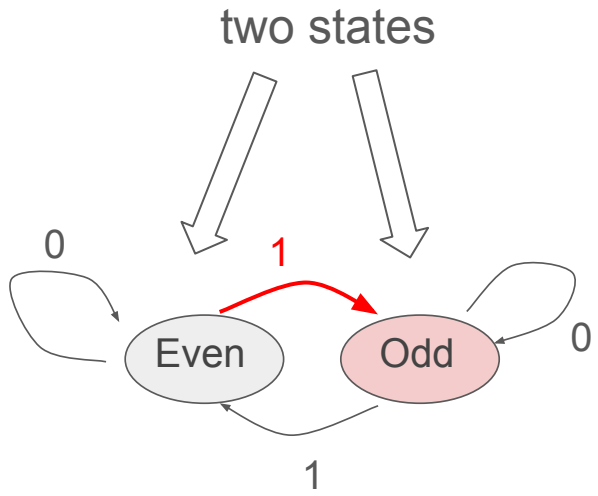
State Tracking



PARITY: Is the number of 1s even?

State Tracking

transitions between states



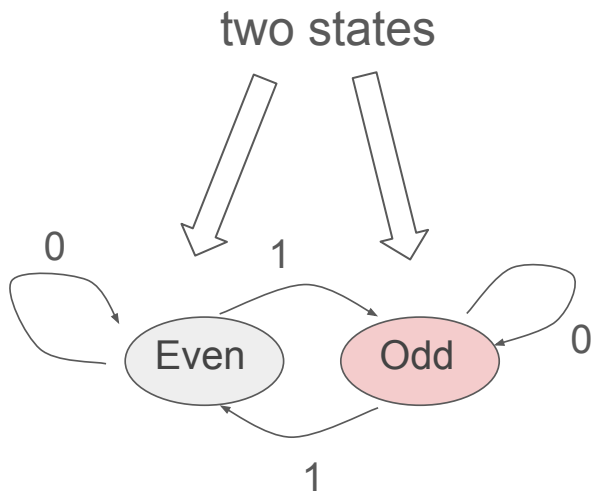
Even Odd Odd Even Odd

0 1 0 1 1

PARITY: Is the number of 1s even?

State Tracking

transitions between states

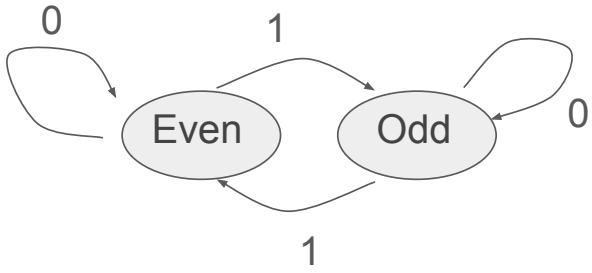


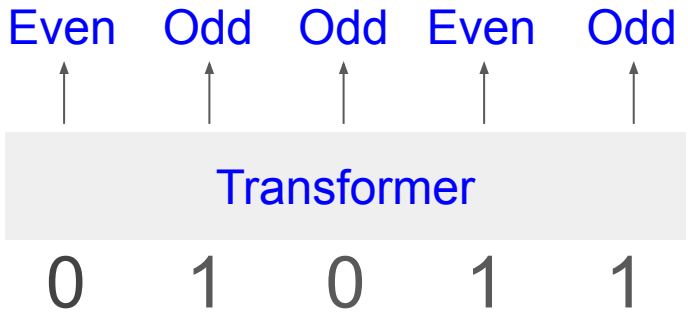
Even Odd Odd Even Odd

0 1 0 1 1

PARITY: Is the number of 1s even?

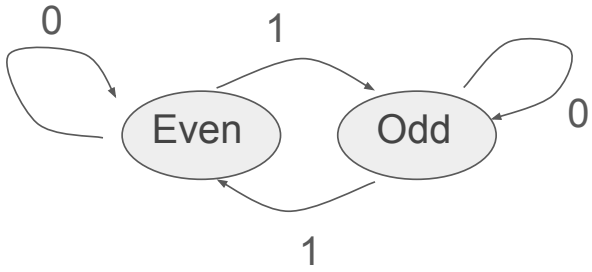
State Tracking





PARITY: Is the number of 1s even?

State Tracking

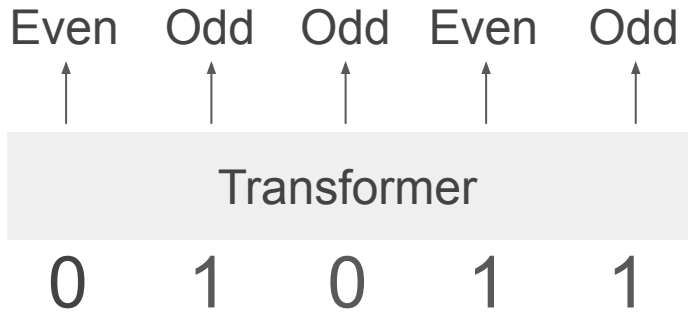


Theoretical Result:

Any transformer outputting PARITY must make errors as inputs get long.

Hahn, TACL 2020

Hahn and Rofin, ACL 2024
(Best Paper Award)

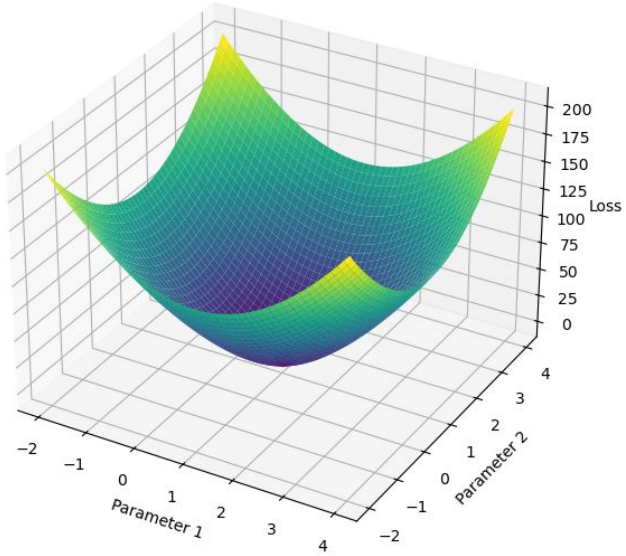


PARITY: Is the number of 1s even?

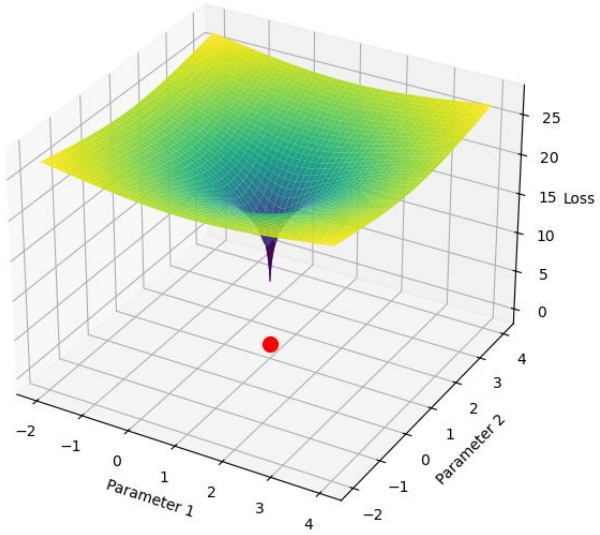
Theorem (Informal)

For transformers, learning state tracking requires a steep loss landscape.

few transactions

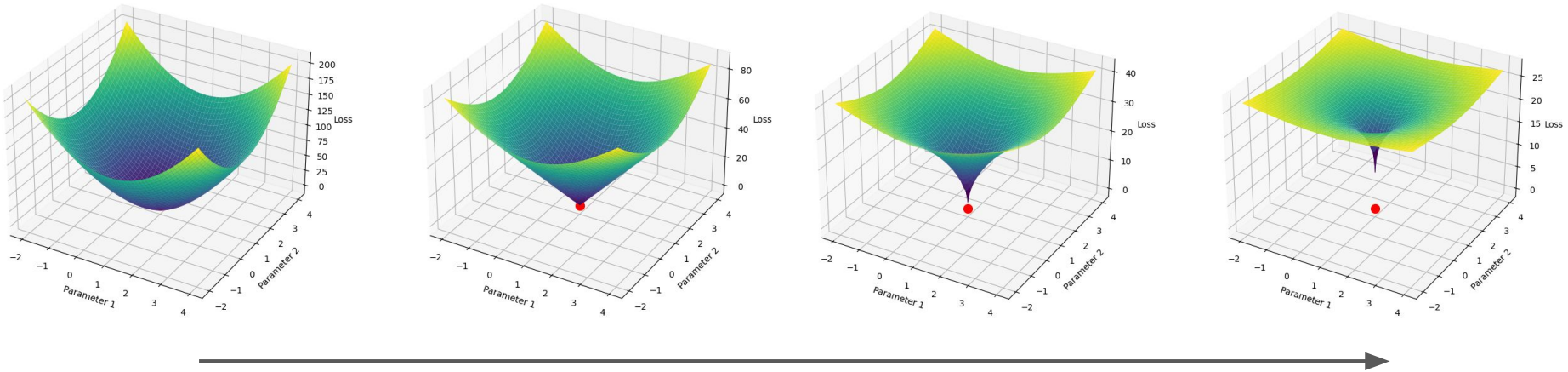


many transactions



Theorem (Informal)

For transformers, learning state tracking requires a steep loss landscape.



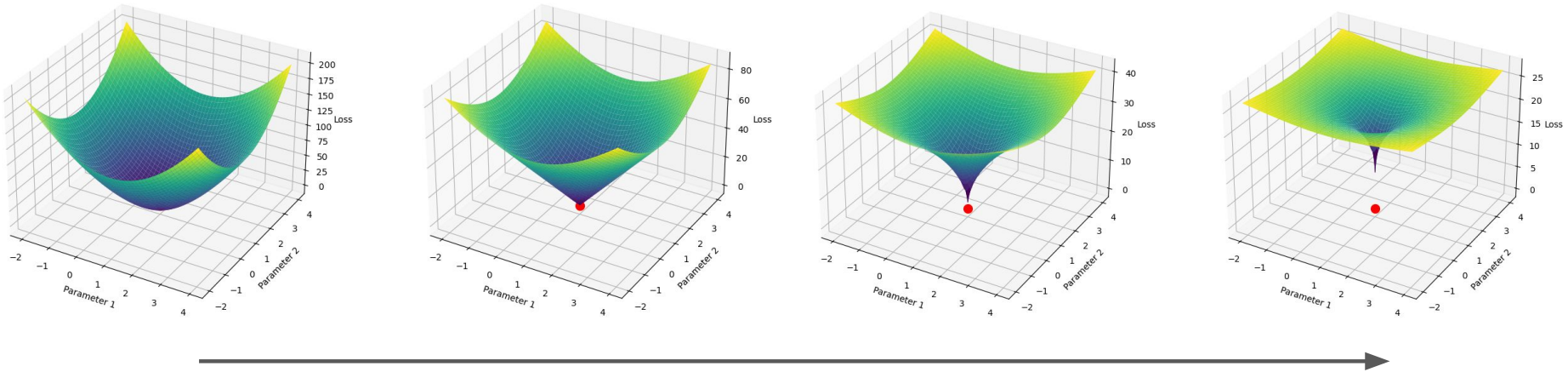
more transactions

sharper minima

Hahn & Rofin, ACL 2024

Theorem (Informal)

For transformers, learning state tracking requires a steep loss landscape.



more transactions

sharper minima

Hahn & Rofin, ACL 2024

Proof Idea

Proof Idea

Assume transformer can track states

Proof Idea

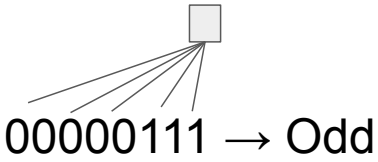
Assume transformer can track states

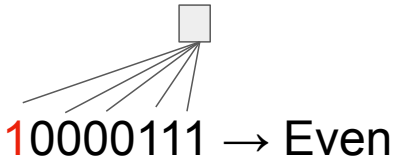
00000111 → Odd

10000111 → Even
01000111 → Even
00100111 → Even
00010111 → Even
00001111 → Even
00000011 → Even
00000101 → Even
00000110 → Even

Proof Idea

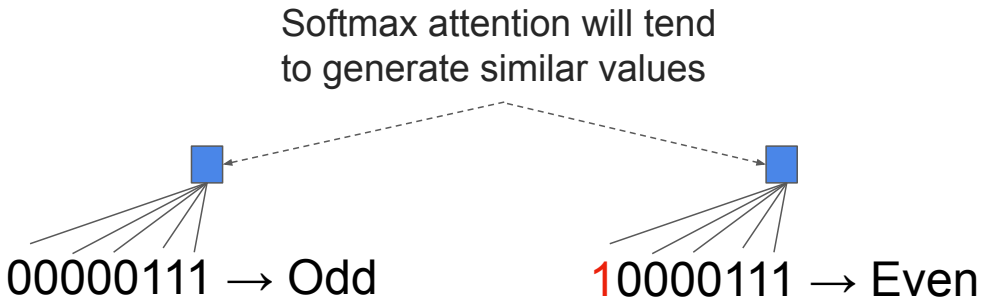
Assume transformer can track states





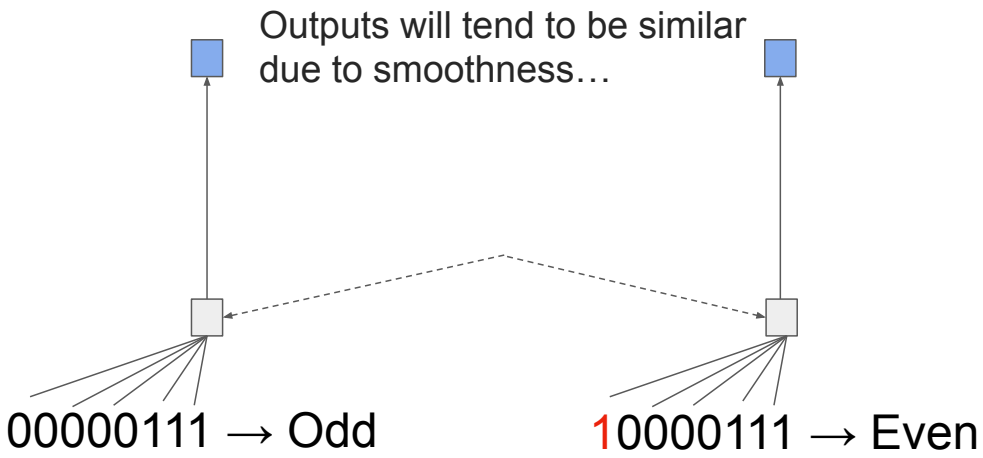
Proof Idea

Assume transformer can track states



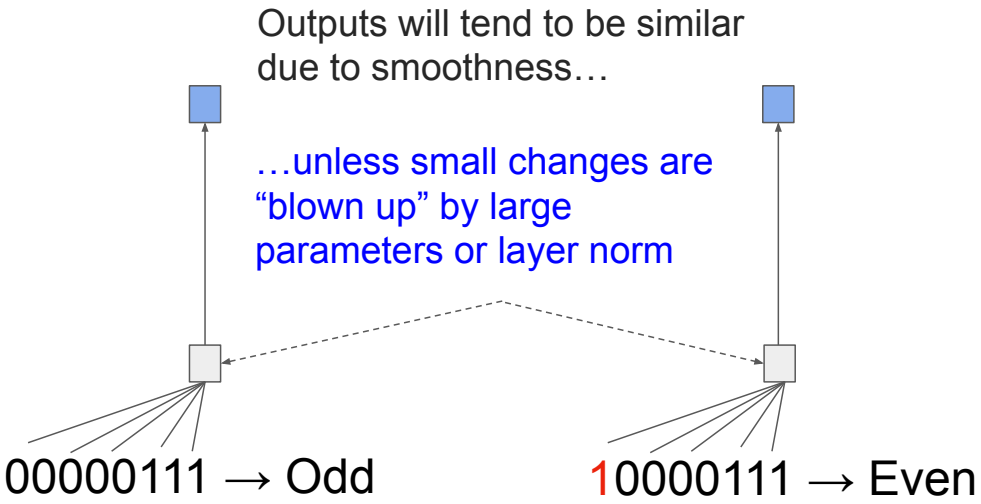
Proof Idea

Assume transformer can track states



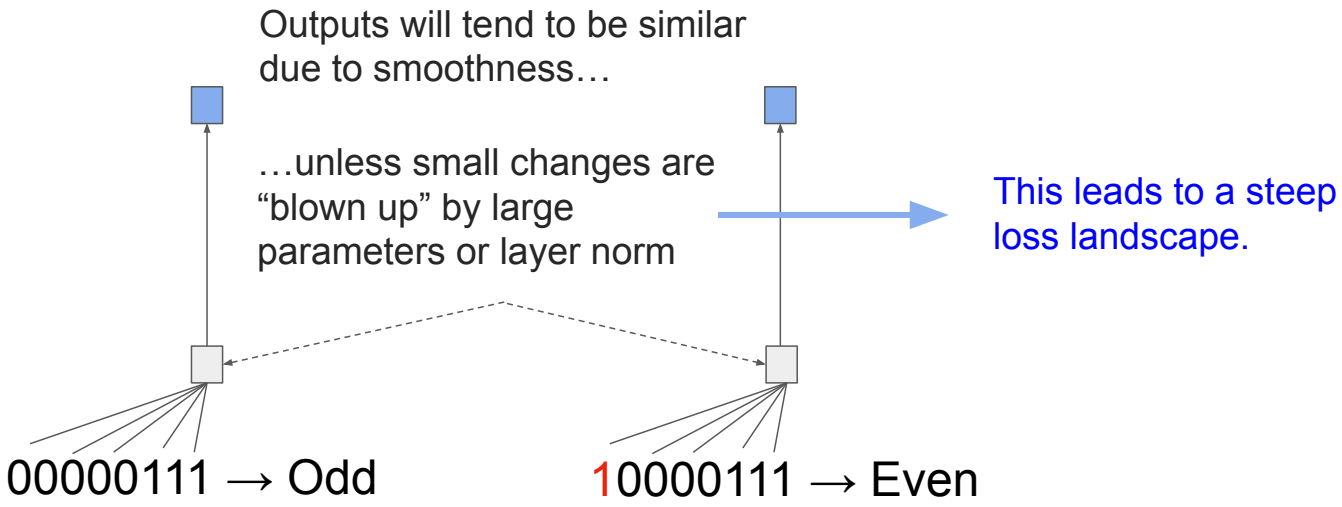
Proof Idea

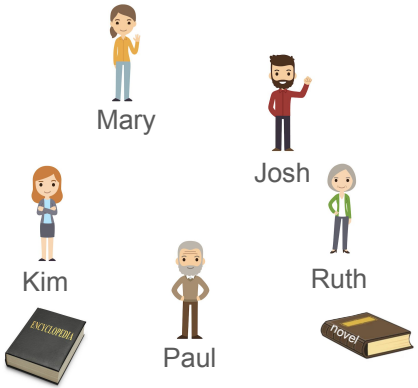
Assume transformer can track states

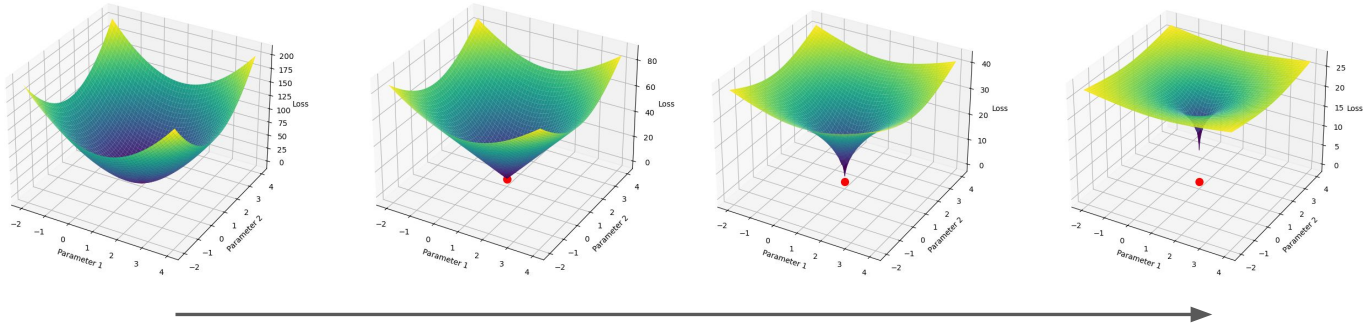


Proof Idea

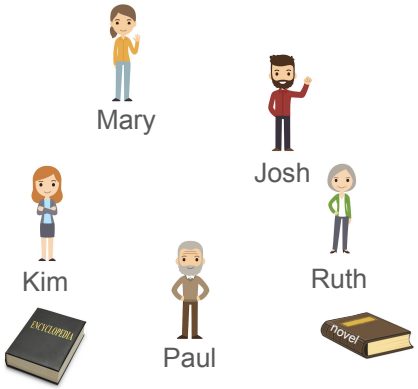
Assume transformer can track states

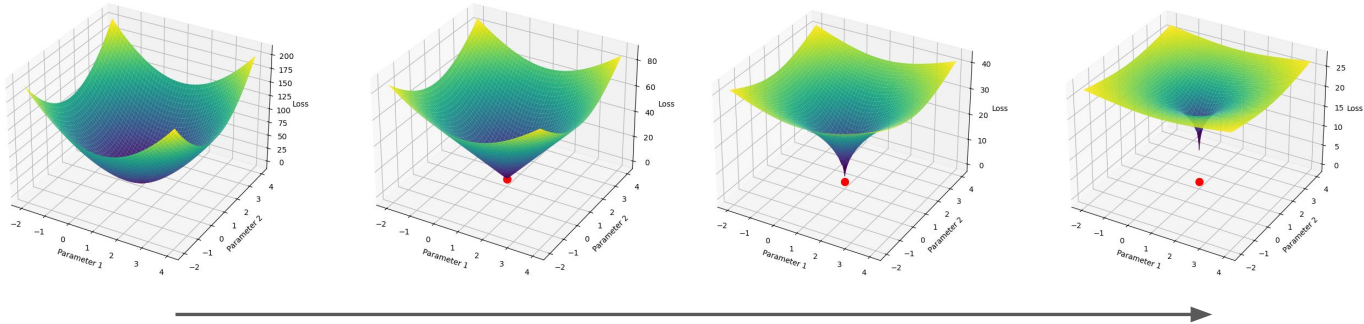






more transactions





more transactions

Got it — let's carefully trace the movements step by step:

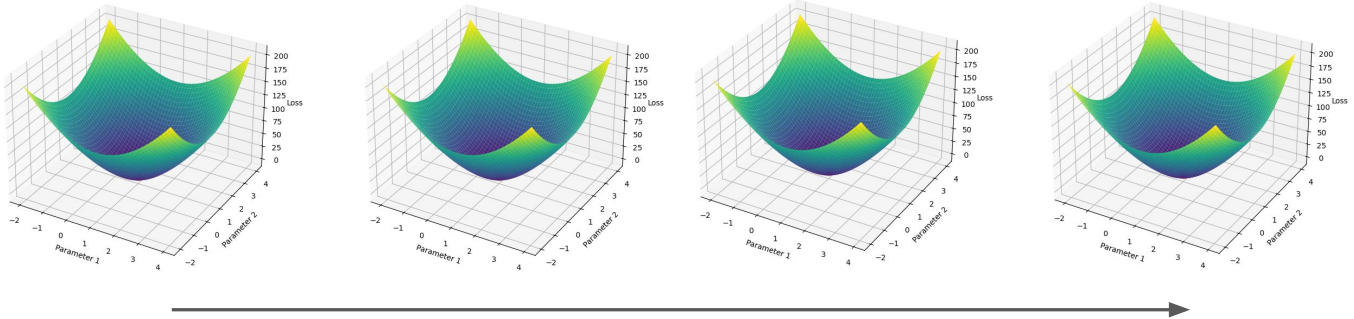
1. **Initial ownership:**
   - Paul → *encyclopedia*
   - Mary → *novel*
2. **Paul gave his book (encyclopedia) to Josh:**
   - Josh → *encyclopedia*
   - Paul → nothing
3. **Mary gave her book (novel) to Paul:**
   - Paul → *novel*
   - Mary → nothing
4. **Josh gave his book (encyclopedia) to Kim:**
   - Kim → *encyclopedia*
   - Josh → nothing
5. **Paul gave his book (novel) to Ruth:**
   - Ruth → *novel*
   - Paul → nothing

✅ **Final ownership:**

- Kim → *encyclopedia*
- Ruth → *novel*
- Paul, Mary, Josh → nothing

Answer: Kim has the encyclopedia.

with step-by-step reasoning



more transactions

Hahn & Rofin, ACL 2024

A Formal Framework for Understanding Length Generalization in Transformers

Xinting Huang[1*] **Andy Yang**[2*] **Satwik Bhattamishra**[3] **Yash Sarrof**[1]
Andreas Krebs[4] **Hattie Zhou**[5] **Preetum Nakkiran**[6] **Michael Hahn**[1†]
[1]Saarland University [2]University of Notre Dame [3]University of Oxford
[4]University of Tübingen [5]Mila, Université de Montréal [6]Apple

ICLR 2025

Born a Transformer – Always a Transformer? On the Effect of Pretraining on Architectural Abilities

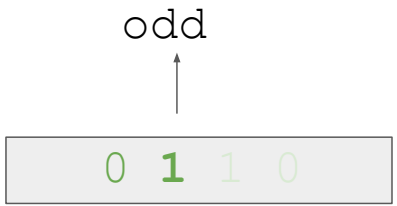
Mayank Jobanputra[1*], **Yana Veitsman**[1*], **Yash Sarrof**[1], **Aleksandra Bakalova**[1],
Vera Demberg[1], **Ellie Pavlick**[2], **Michael Hahn**[1]
[1]Saarland University [2]Brown University
{mayank,mhahn}@lst.uni-saarland.de

NeurIPS 2025

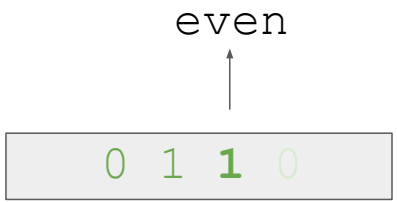
0 1 1 0

PARITY

PARITY

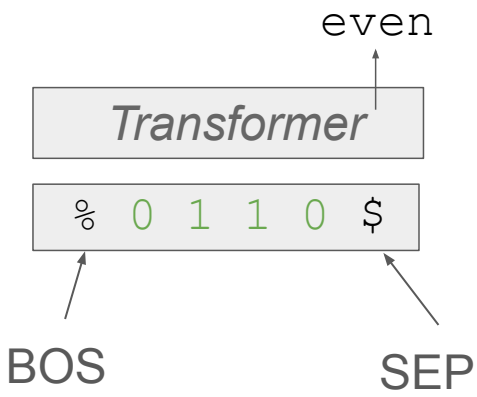


PARITY

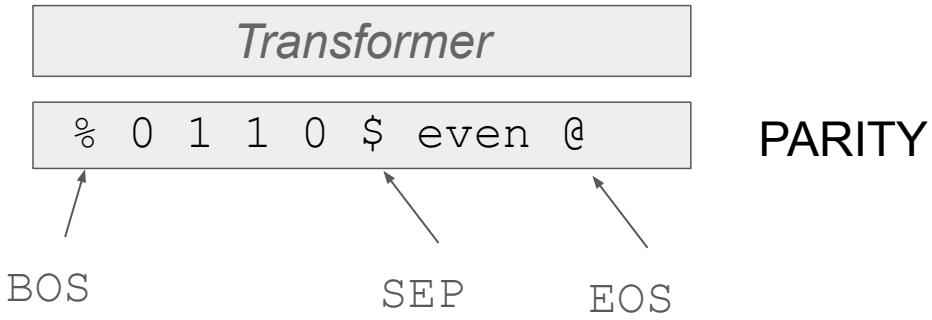


PARITY

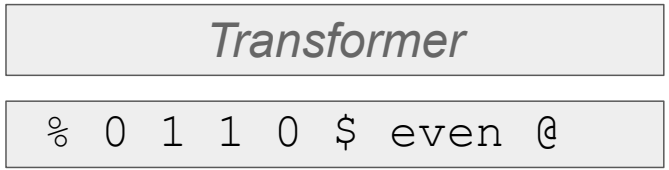
PARITY



PARITY



PARITY



PARITY

If we train transformer at lengths ≤ N, …

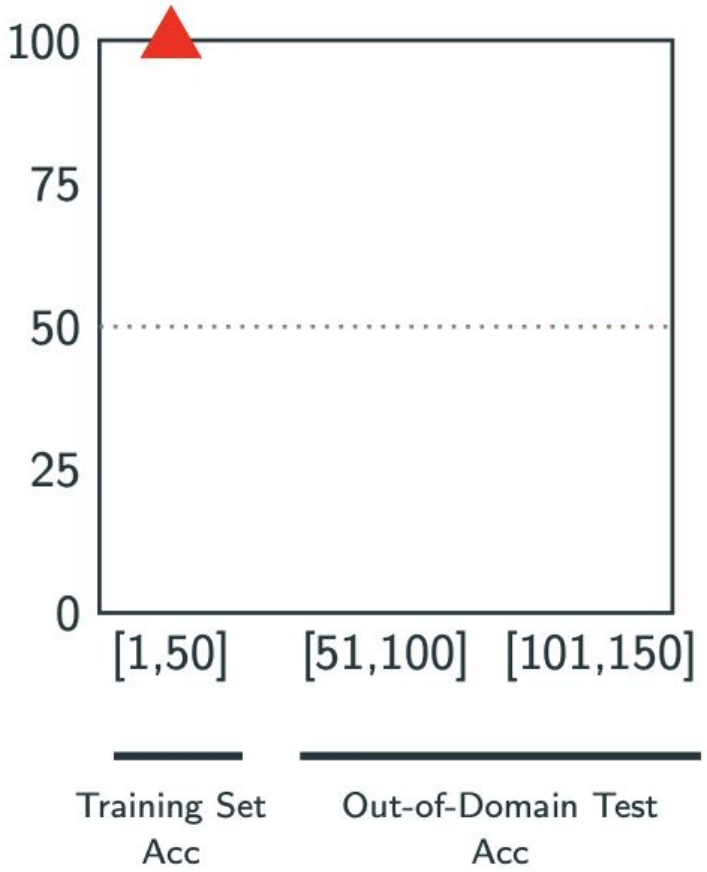
If we train transformer at lengths ≤ N, …

does it perform correctly at length 2N?

If we train transformer at lengths ≤ N, …

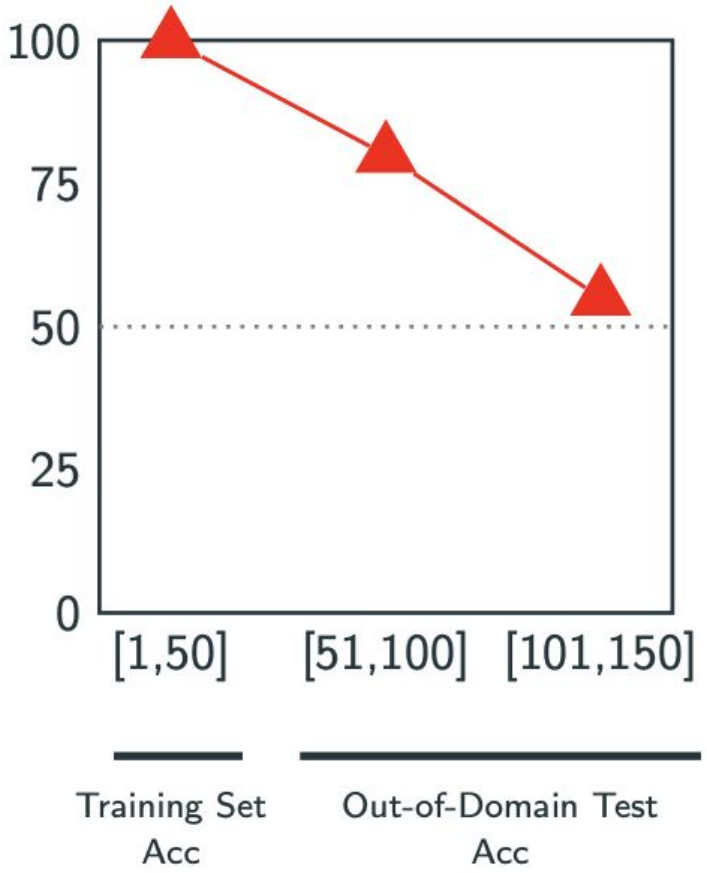
does it perform correctly at length 2N?

Not for this task!

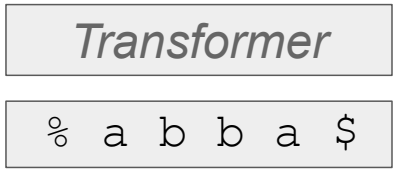


Related results: Bhattamishra et al 2020; Anil et al 2022; Zhou et al 2024.

Not for this task!

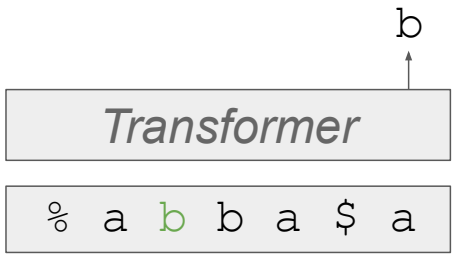


Related results: Bhattamishra et al 2020; Anil et al 2022; Zhou et al 2024.



COPY

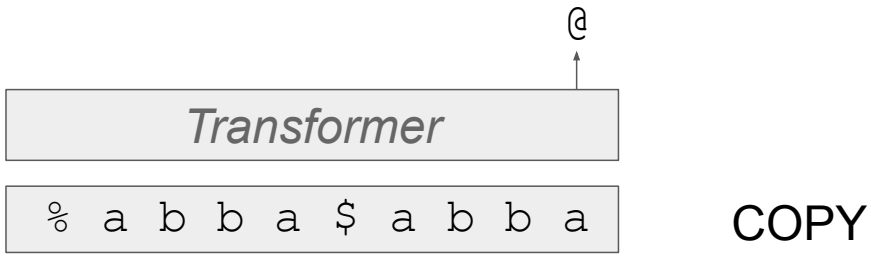
COPY



COPY

COPY

COPY



COPY

COPY

COPY

COPY

If we train transformer at lengths ≤ N, …

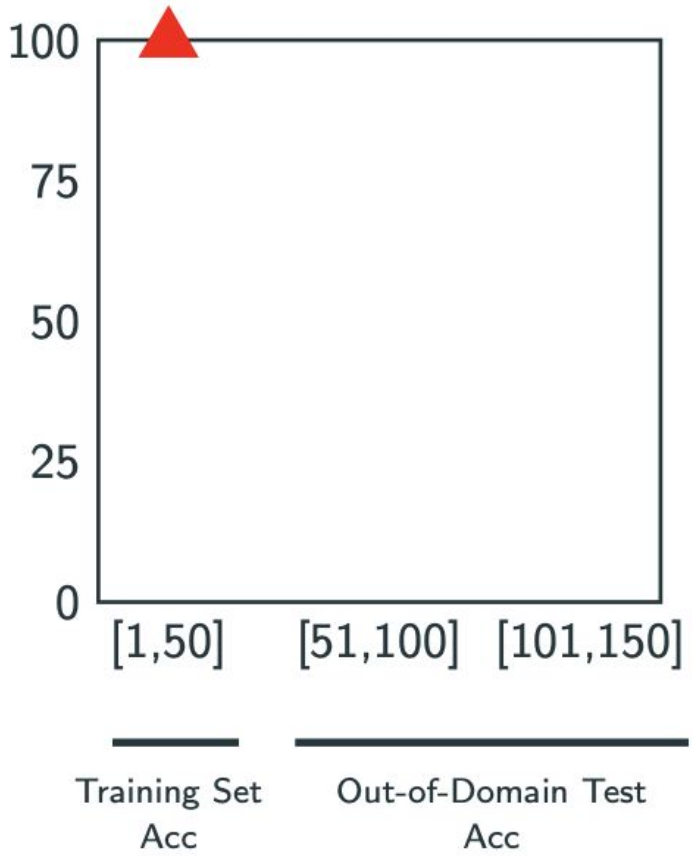
If we train transformer at lengths ≤ N, …

does it perform correctly at length 2N?

If we train transformer at lengths ≤ N, …

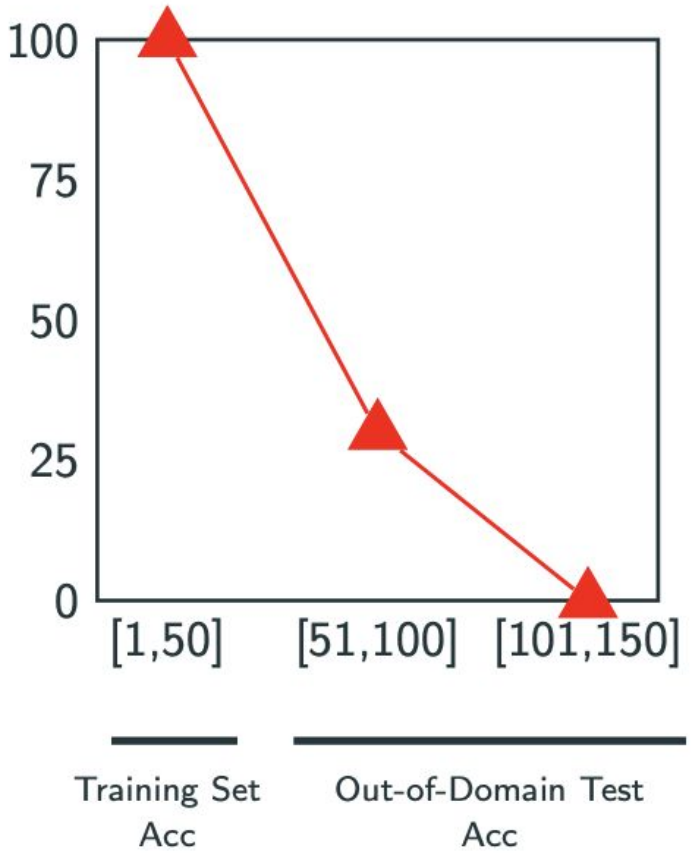
does it perform correctly at length 2N?

Not for this task!

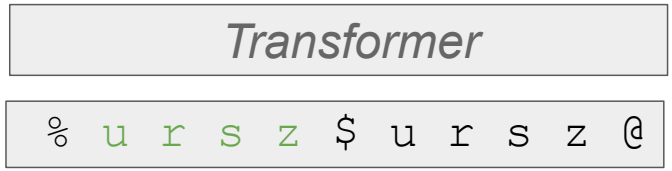


Related results: Zhou et al 2024 ICLR; Jelassi et al 2024 ICML; Kazemnejad et al 2023 NeurIPS.

Not for this task!



Related results: Zhou et al 2024 ICLR; Jelassi et al 2024 ICML; Kazemnejad et al 2023 NeurIPS.



COPY UNIQUE

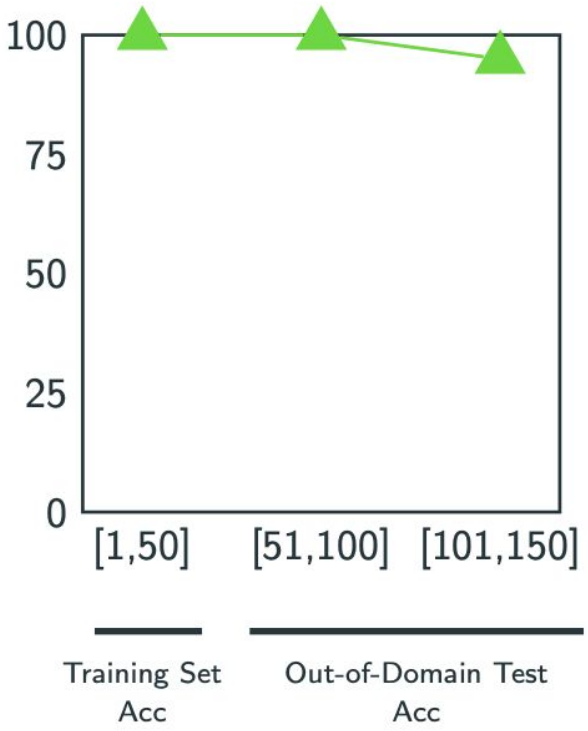
but for many other tasks, it works!

Transformer

% u r s z $ u r s z @

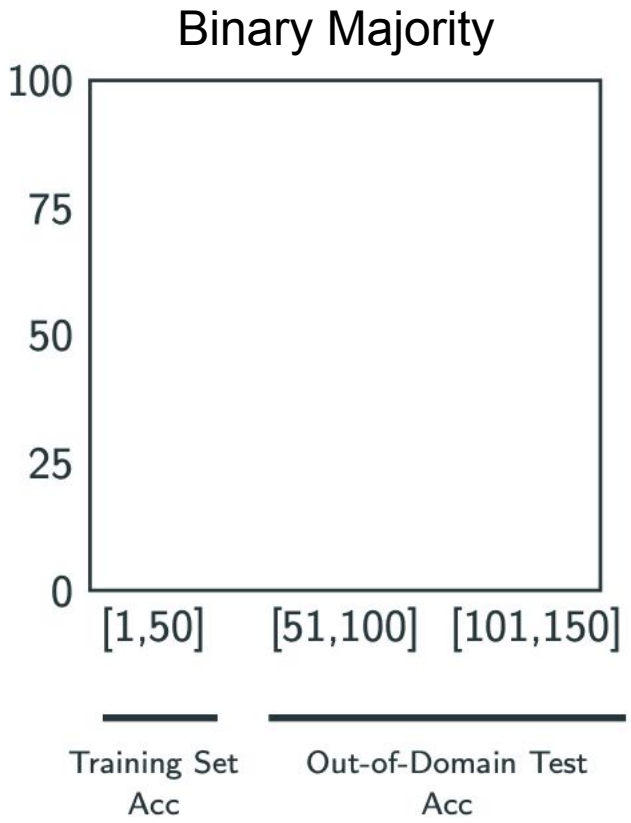
COPY UNIQUE

but for many other tasks, it works!



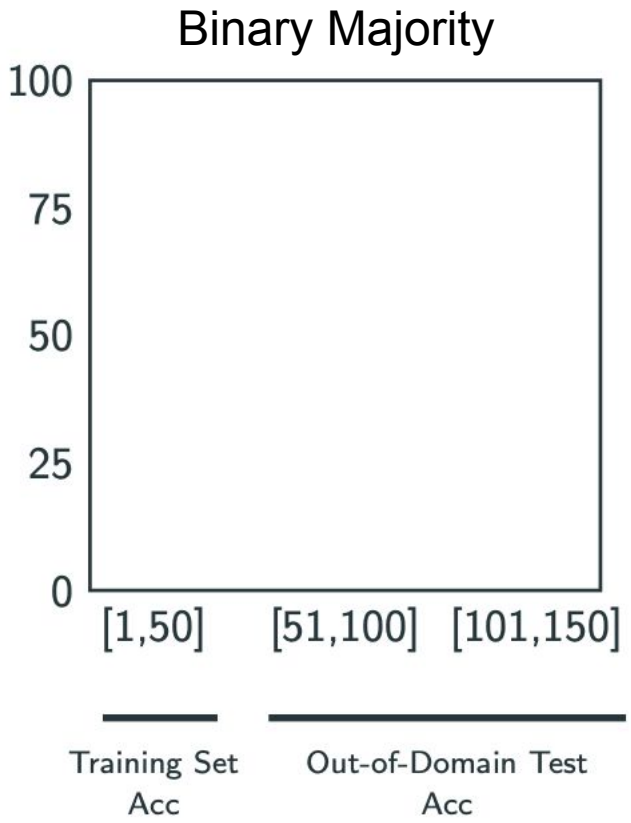
Related results: Zhou et al 2024 ICLR; Jelassi et al 2024 ICML.

On which tasks do transformers length-generalize?



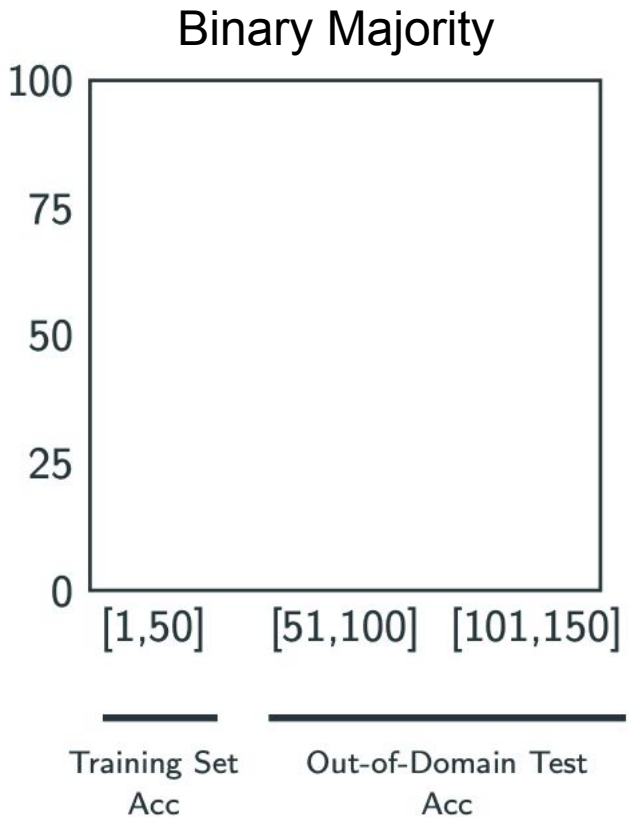
% 0 0 1 1 0 1 1 1 0 $

C-RASP: Yang&Chiang 2024, COLM



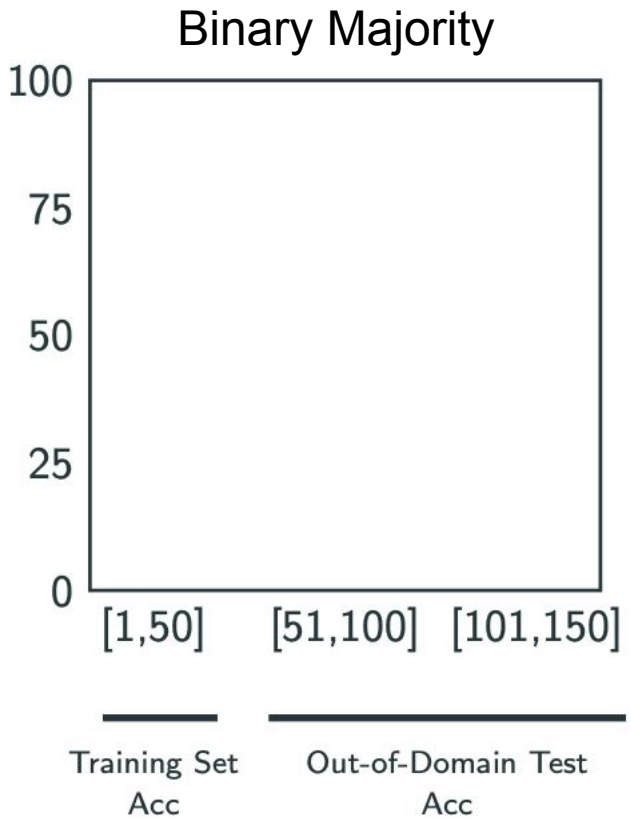
% 0 0 1 1 0 1 1 1 0 $

C-RASP: Yang&Chiang 2024, COLM



% 0 0 1 1 0 1 1 1 0 $ **T** @

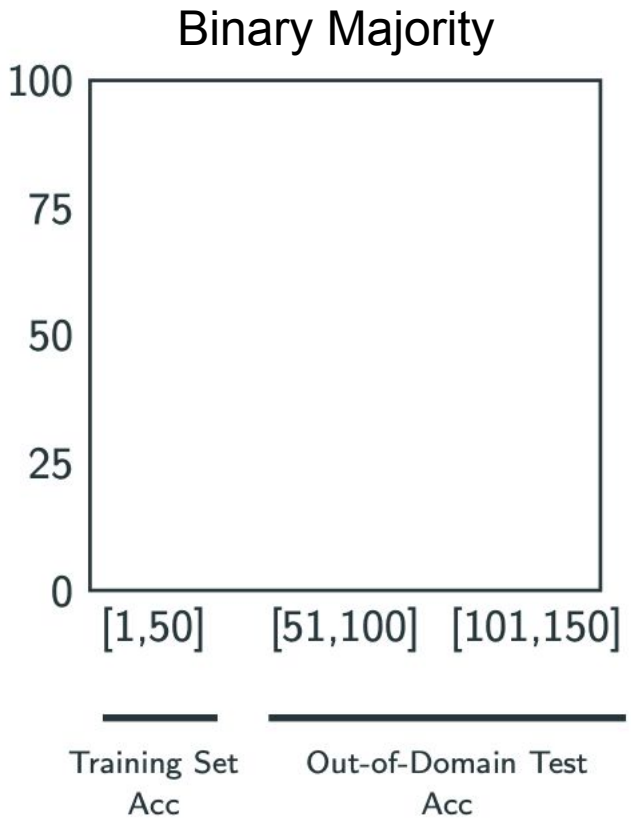
C-RASP: Yang&Chiang 2024, COLM



MAJORITY

% 0 0 1 1 0 1 1 1 0 $

C-RASP: Yang&Chiang 2024, COLM

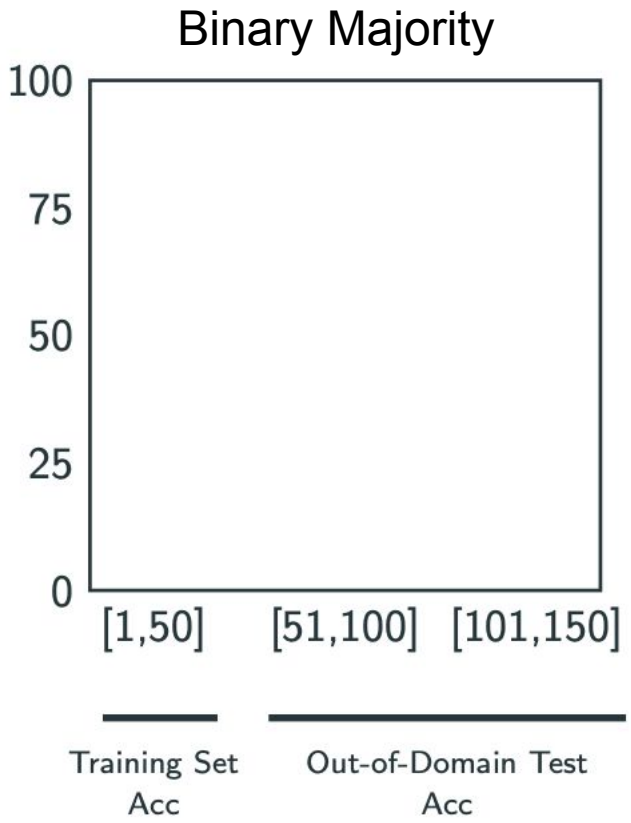


MAJORITY

$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

% 0 0 1 1 0 1 1 1 0 $

C-RASP: Yang&Chiang 2024, COLM



MAJORITY

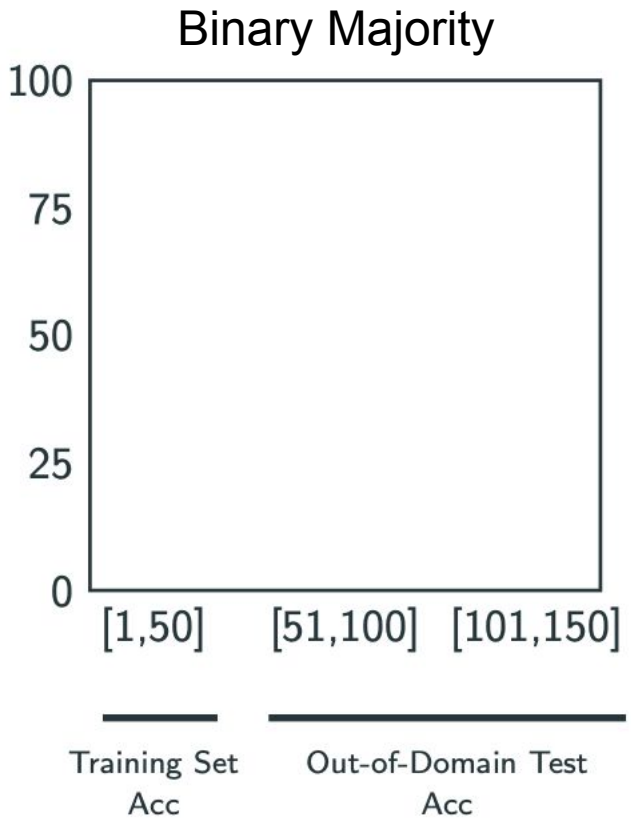
$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

% **0** 0 1 1 0 1 1 1 0 $

C_1(i) 0 0

running count of 1s

C-RASP: Yang&Chiang 2024, COLM



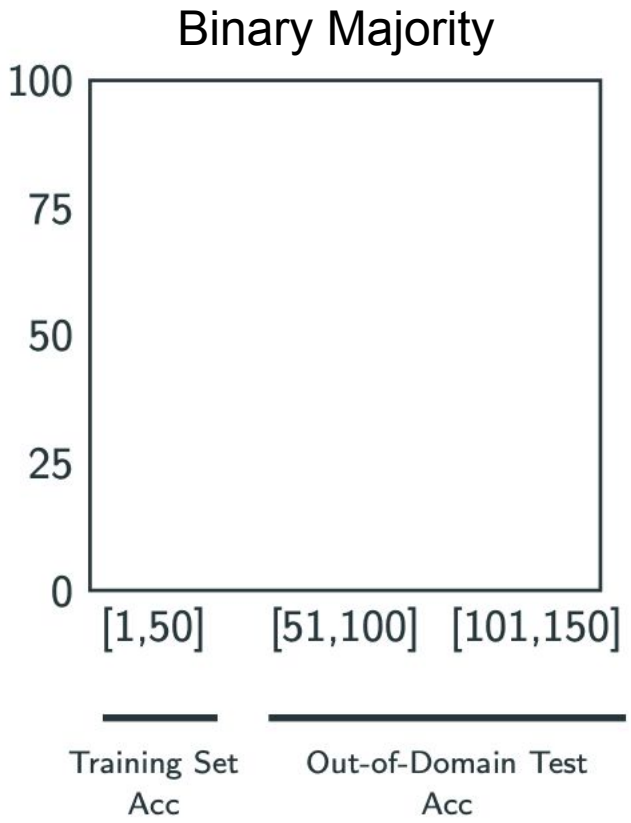
MAJORITY

$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

	%	0	**0**	1	1	0	1	1	1	0	$
$C_1(i)$		0	0	0							

running count of 1s

C-RASP: Yang&Chiang 2024, COLM



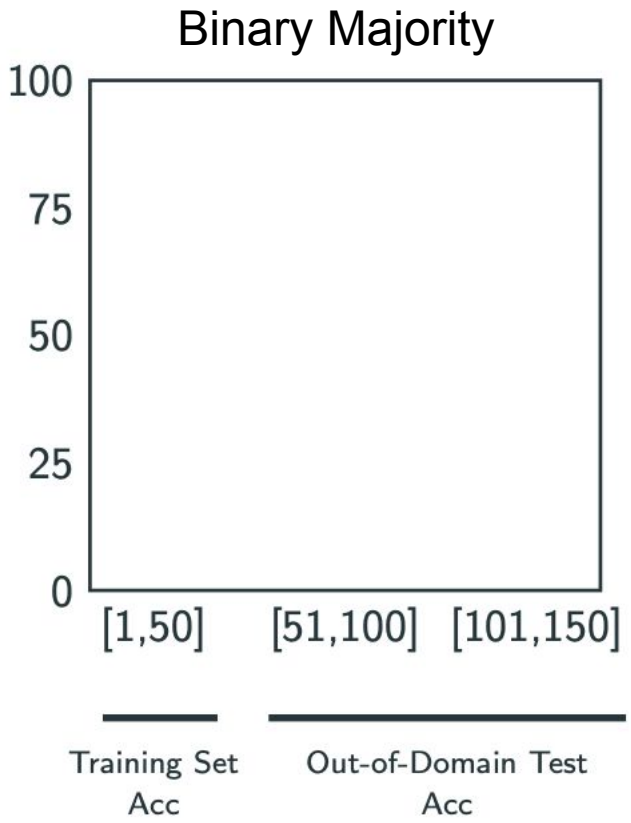
MAJORITY

$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

	%	0	0	**1**	1	0	1	1	1	0	$
$C_1(i)$		0	0	0	1						

running count of 1s

C-RASP: Yang&Chiang 2024, COLM



MAJORITY

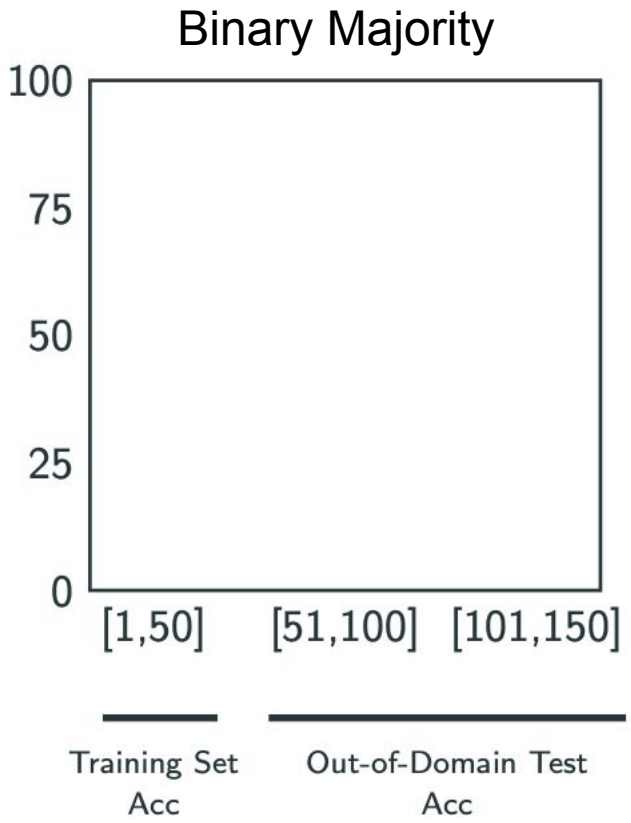
$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

% 0 0 1 **1** 0 1 1 1 0 $

C_1(i) 0 0 0 1 2

running count of 1s

C-RASP: Yang&Chiang 2024, COLM



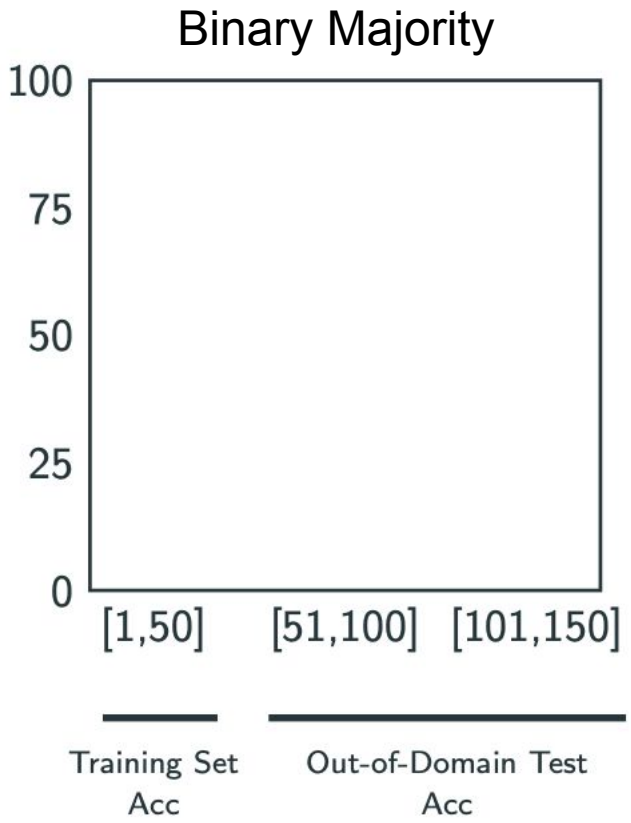
MAJORITY

$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

	%	0	0	1	1	**0**	1	1	1	0	$
$C_1(i)$		0	0	0	1	2	2				

running count of 1s

C-RASP: Yang&Chiang 2024, COLM



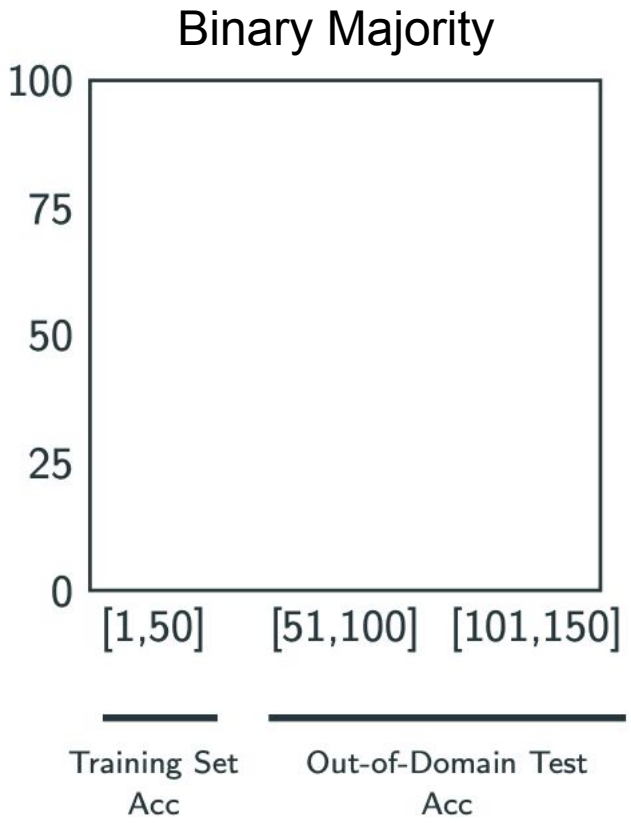
MAJORITY

$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

	%	0	0	1	1	0	**1**	1	1	0	$
$C_1(i)$		0	0	0	1	2	2	3			

running count of 1s

C-RASP: Yang&Chiang 2024, COLM



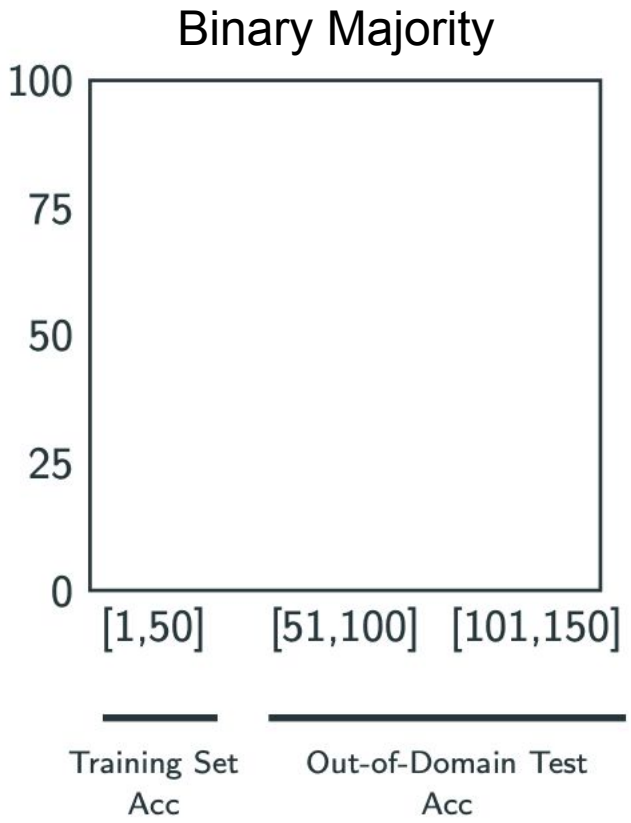
MAJORITY

$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

	%	0	0	1	1	0	1	**1**	1	0	$
$C_1(i)$	0	0	0	1	2	2	3	4			

running count of 1s

C-RASP: Yang&Chiang 2024, COLM



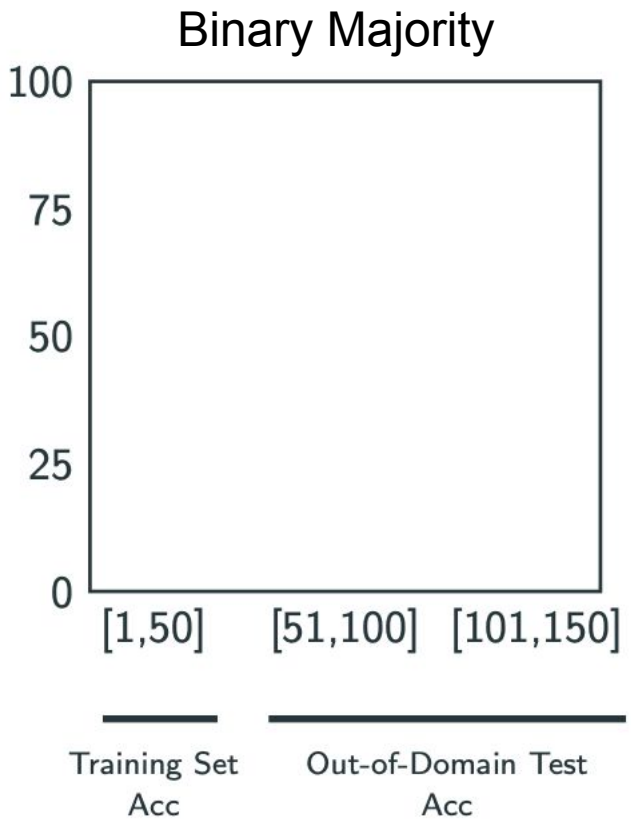
MAJORITY

$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

	%	0	0	1	1	0	1	1	**1**	0	$
$C_1(i)$	0	0	0	1	2	2	3	4	5		

running count of 1s

C-RASP: Yang&Chiang 2024, COLM



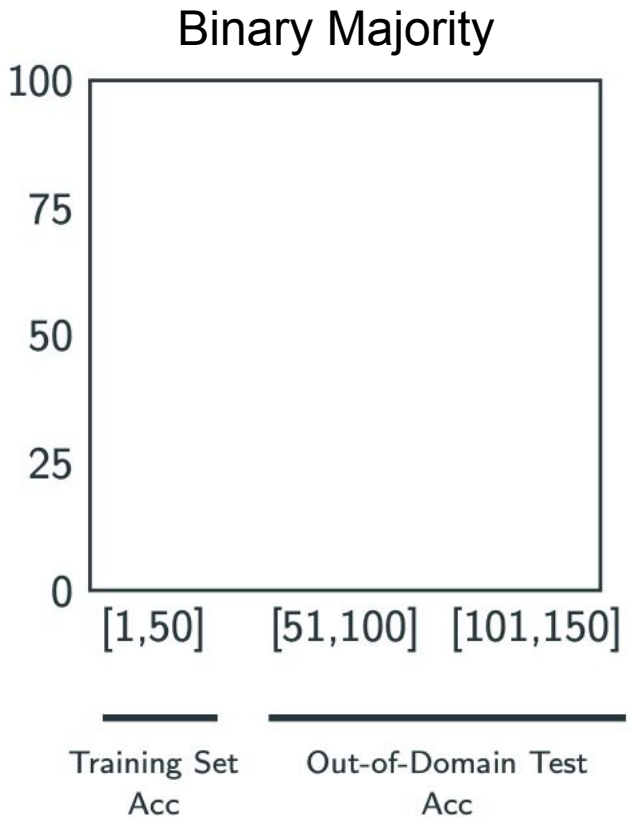
MAJORITY

$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

	%	0	0	1	1	0	1	1	1	**0**	$
$C_1(i)$	0	0	0	1	2	2	3	4	5	5	

running count of 1s

C-RASP: Yang&Chiang 2024, COLM



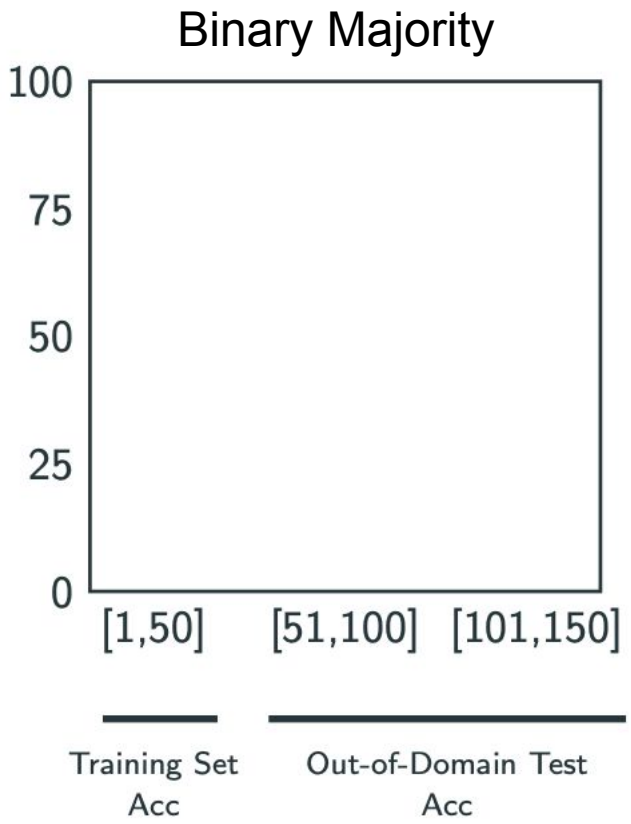
MAJORITY

$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

	%	0	0	1	1	0	1	1	1	0	$
$C_1(i)$	0	0	0	1	2	2	3	4	5	5	5

running count of 1s

C-RASP: Yang&Chiang 2024, COLM



MAJORITY

$C_1(i) := \#[j \le i]\ Q_1(j)$ count # of 1's (1)

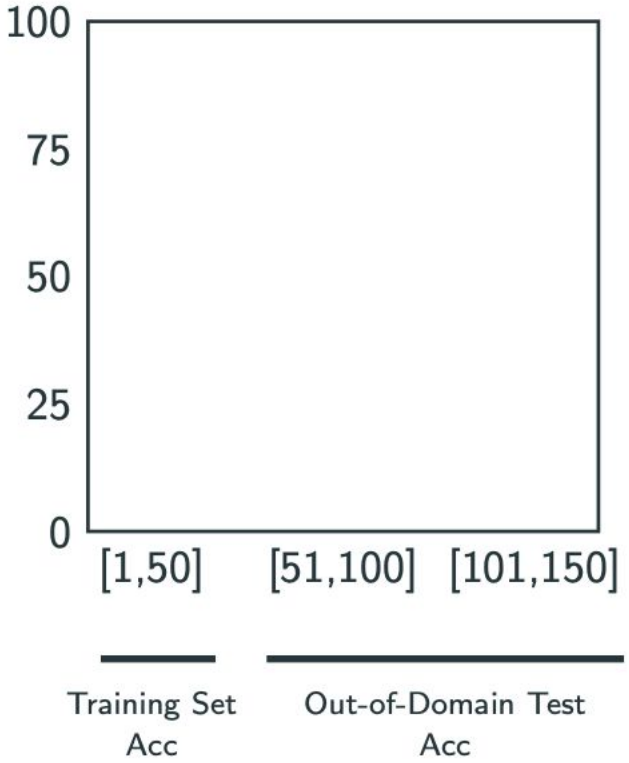
$C_0(i) := \#[j \le i]\ Q_0(j)$ count # of 0's (2)

	%	0	0	1	1	0	1	1	1	0	$
$C_1(i)$	0	0	0	1	2	2	3	4	5	5	5
$C_0(i)$	0										

running count of 0s

C-RASP: Yang&Chiang 2024, COLM

Binary Majority



MAJORITY

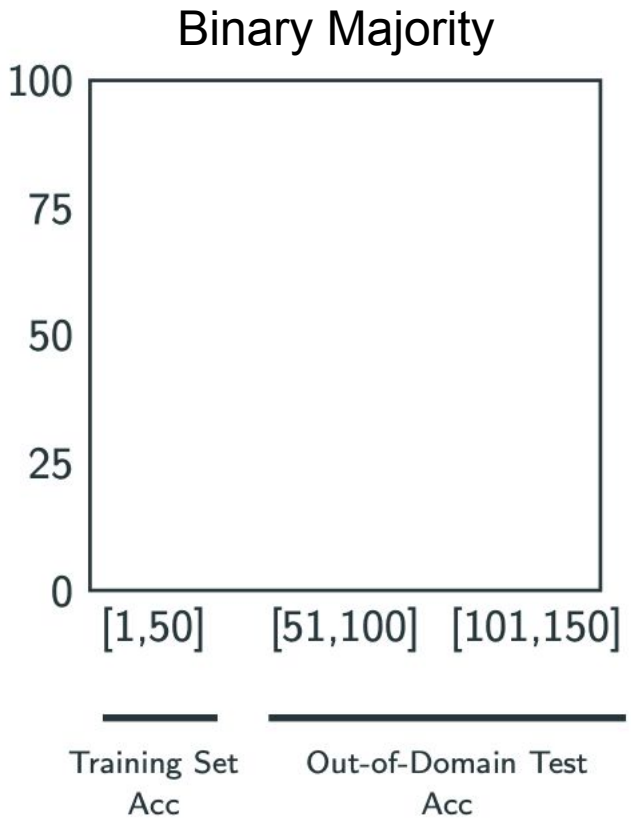
$C_1(i) := \#[j \le i]\ Q_1(j)$ — count # of 1's (1)

$C_0(i) := \#[j \le i]\ Q_0(j)$ — count # of 0's (2)

	%	0	0	1	1	0	1	1	1	0	$
$C_1(i)$	0	0	0	1	2	2	3	4	5	5	5
$C_0(i)$	0	1									

running count of 0s

C-RASP: Yang&Chiang 2024, COLM



MAJORITY

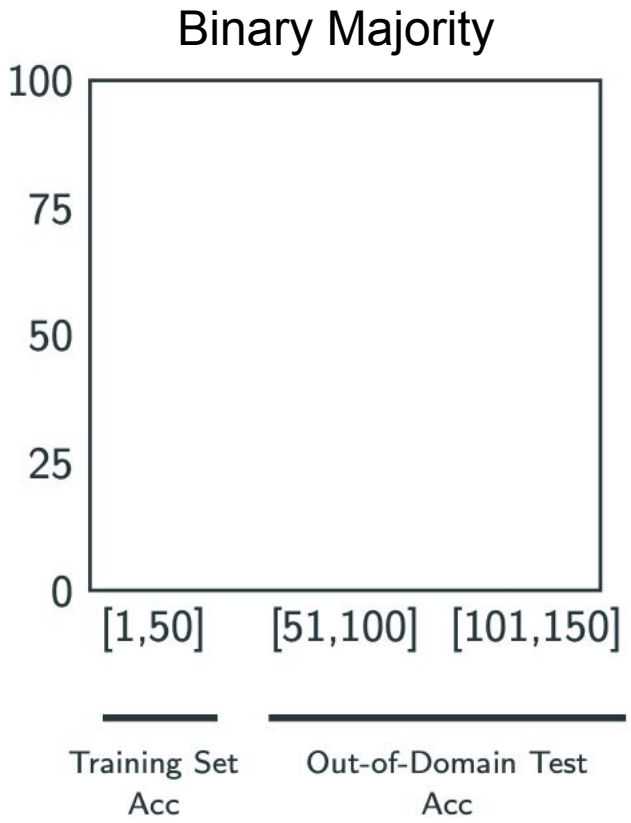
$C_1(i) := \#[j \le i]\ Q_1(j)$ count # of 1's (1)

$C_0(i) := \#[j \le i]\ Q_0(j)$ count # of 0's (2)

	%	0	**0**	1	1	0	1	1	1	0	$
$C_1(i)$	0	0	0	1	2	2	3	4	5	5	5
$C_0(i)$	0	1	2								

running count of 0s

C-RASP: Yang&Chiang 2024, COLM



MAJORITY

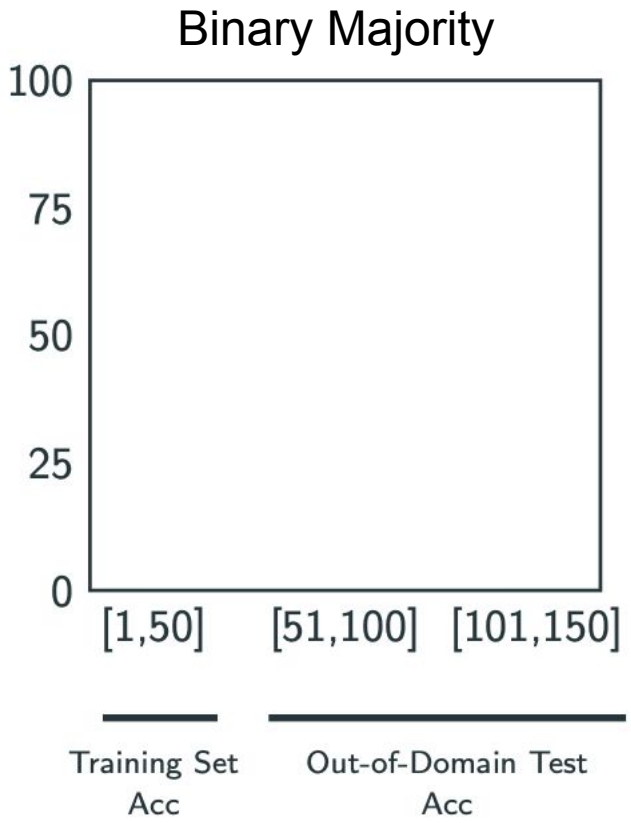
$C_1(i) := \#[j \le i]\ Q_1(j)$ count # of 1's (1)

$C_0(i) := \#[j \le i]\ Q_0(j)$ count # of 0's (2)

	%	0	0	1	1	**0**	1	1	1	0	$
$C_1(i)$	0	0	0	1	2	2	3	4	5	5	5
$C_0(i)$	0	1	2	2	2	3					

running count of 0s

C-RASP: Yang&Chiang 2024, COLM



MAJORITY

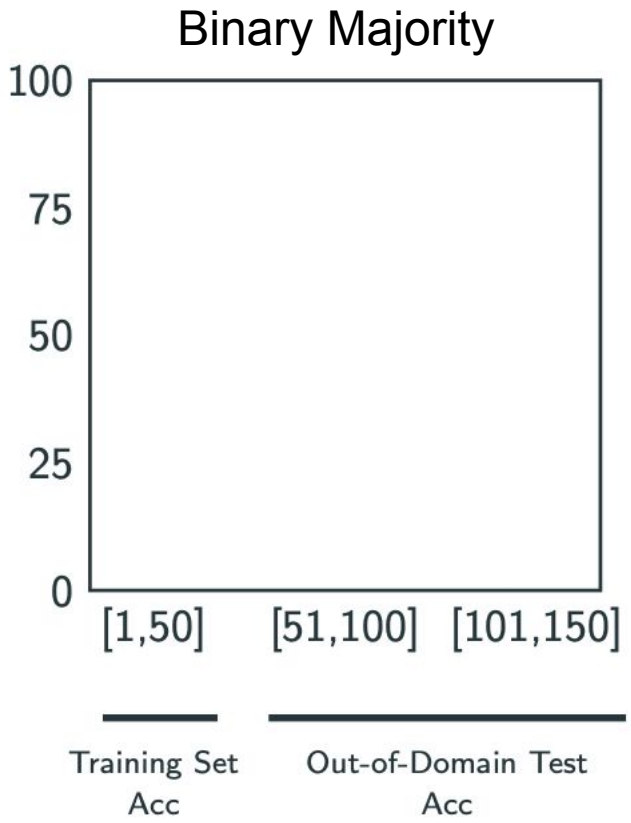
$C_1(i) := \#[j \leq i]\ Q_1(j)$ count # of 1's (1)

$C_0(i) := \#[j \leq i]\ Q_0(j)$ count # of 0's (2)

	%	0	0	1	1	0	1	1	1	**0**	$
$C_1(i)$	0	0	0	1	2	2	3	4	5	5	5
$C_0(i)$	0	1	2	2	2	3	3	3	3	4	

running count of 0s

C-RASP: Yang&Chiang 2024, COLM



MAJORITY

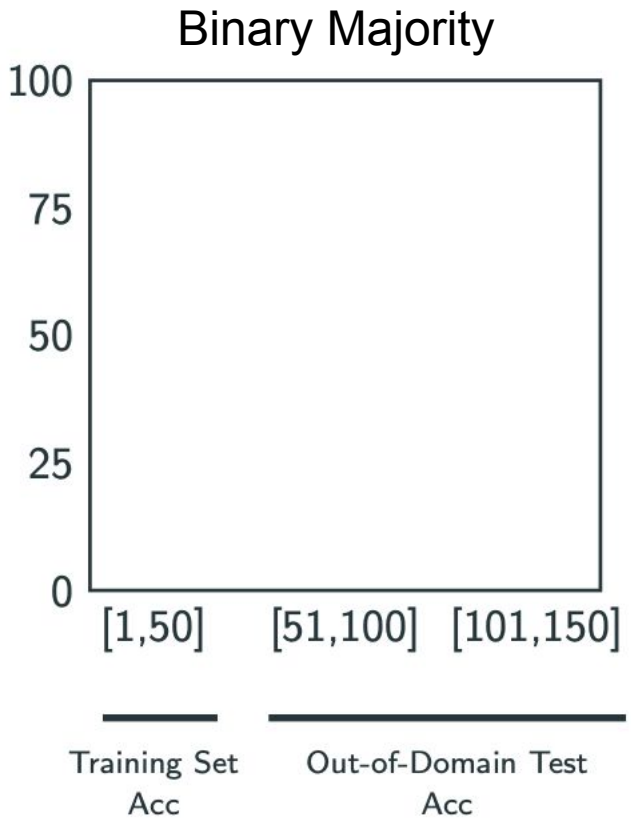
$C_1(i) := \#[j \le i]\ Q_1(j)$ count # of 1's (1)

$C_0(i) := \#[j \le i]\ Q_0(j)$ count # of 0's (2)

	%	0	0	1	1	0	1	1	1	0	$
$C_1(i)$	0	0	0	1	2	2	3	4	5	5	5
$C_0(i)$	0	1	2	2	2	3	3	3	3	4	4

running count of 0s

C-RASP: Yang&Chiang 2024, COLM



MAJORITY

$C_1(i) := \#[j \le i]\ Q_1(j)$ — count # of 1's (1)

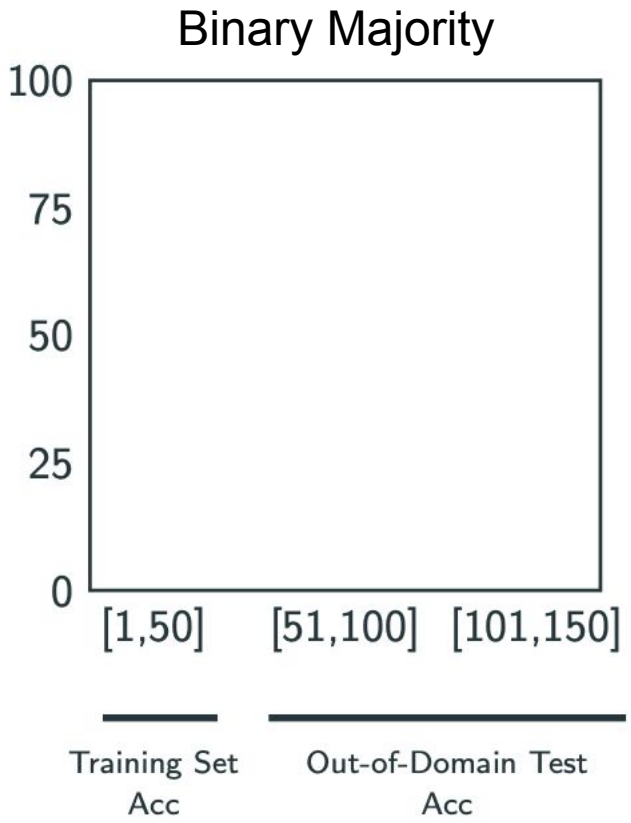
$C_0(i) := \#[j \le i]\ Q_0(j)$ — count # of 0's (2)

$M(i) := C_1(i) \ge C_0(i)$ — compare them (3)

	%	0	0	1	1	0	1	1	1	0	$
$C_1(i)$	0	0	0	1	2	2	3	4	5	5	5
$C_0(i)$	0	**1**	**2**	**2**	2	**3**	3	3	3	4	4
M(i)		**F**	**F**	**F**		**F**					

compare them

C-RASP: Yang&Chiang 2024, COLM



MAJORITY

$C_1(i) := \#[j \le i]\ Q_1(j)$ count # of 1's (1)

$C_0(i) := \#[j \le i]\ Q_0(j)$ count # of 0's (2)

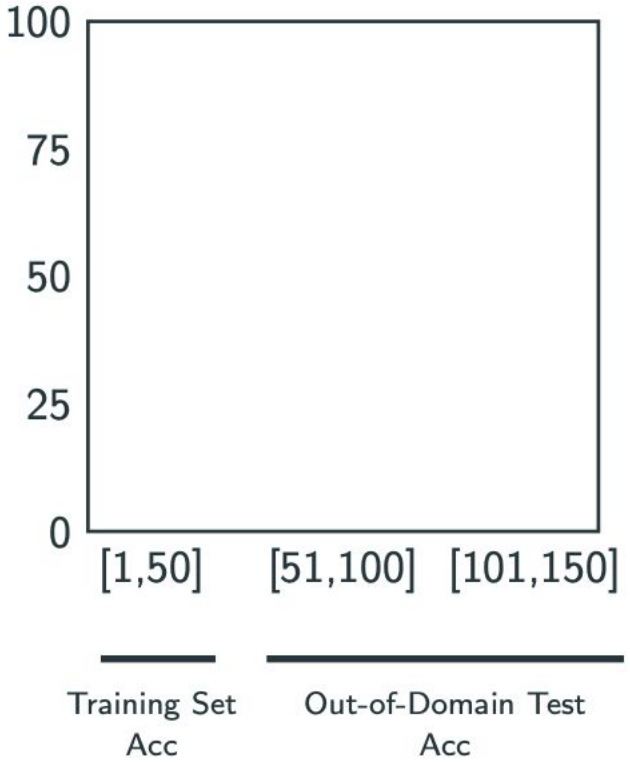
$M(i) := C_1(i) \ge C_0(i)$ compare them (3)

	%	0	0	1	1	0	1	1	1	0	$
$C_1(i)$	**0**	0	0	1	**2**	2	**3**	**4**	**5**	**5**	**5**
$C_0(i)$	0	1	2	2	2	3	3	3	3	4	4
M(i)	**T**	F	F	F	**T**	F	**T**	**T**	**T**	**T**	**T**

compare them

C-RASP: Yang&Chiang 2024, COLM

Binary Majority



MAJORITY

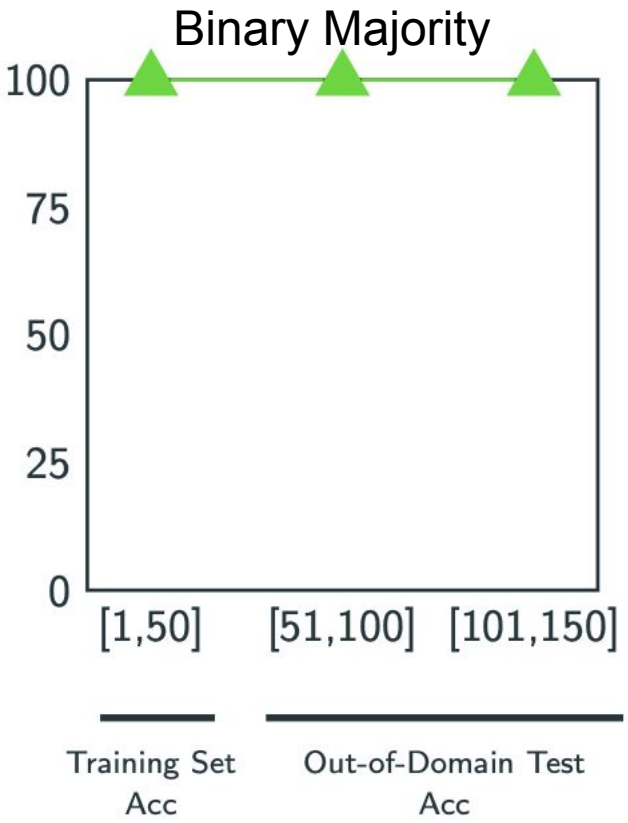
$C_1(i) := \#[j \le i]\ Q_1(j)$ count # of 1's (1)

$C_0(i) := \#[j \le i]\ Q_0(j)$ count # of 0's (2)

$M(i) := C_1(i) \ge C_0(i)$ compare them (3)

	%	0	0	1	1	0	1	1	1	0	$	**T**	@
$C_1(i)$	0	0	0	1	2	2	3	4	5	5	5		
$C_0(i)$	0	1	2	2	2	3	3	3	3	4	4		
M(i)	T	F	F	F	T	F	T	T	T	T	**T**		

C-RASP: Yang&Chiang 2024, COLM



MAJORITY

$C_1(i) := \#[j \le i]\ Q_1(j)$ count # of 1's (1)

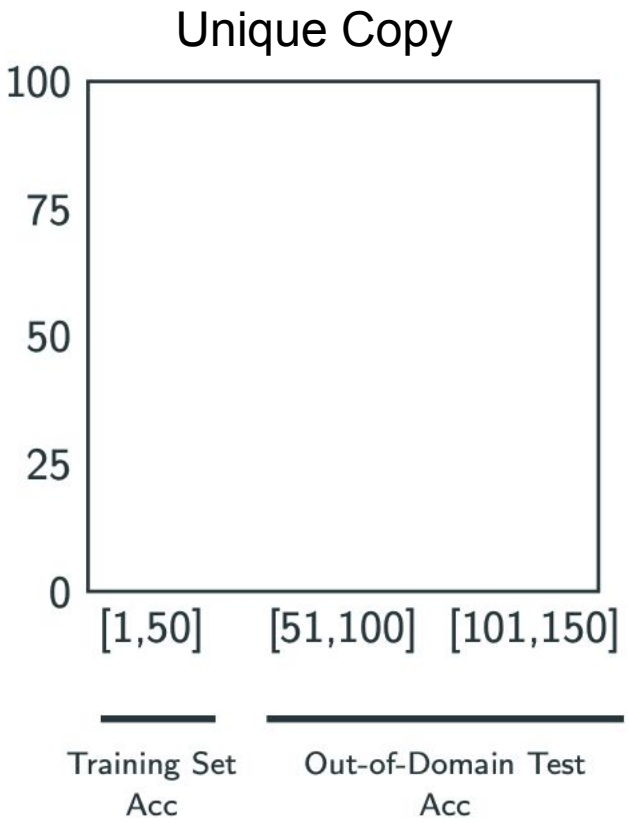
$C_0(i) := \#[j \le i]\ Q_0(j)$ count # of 0's (2)

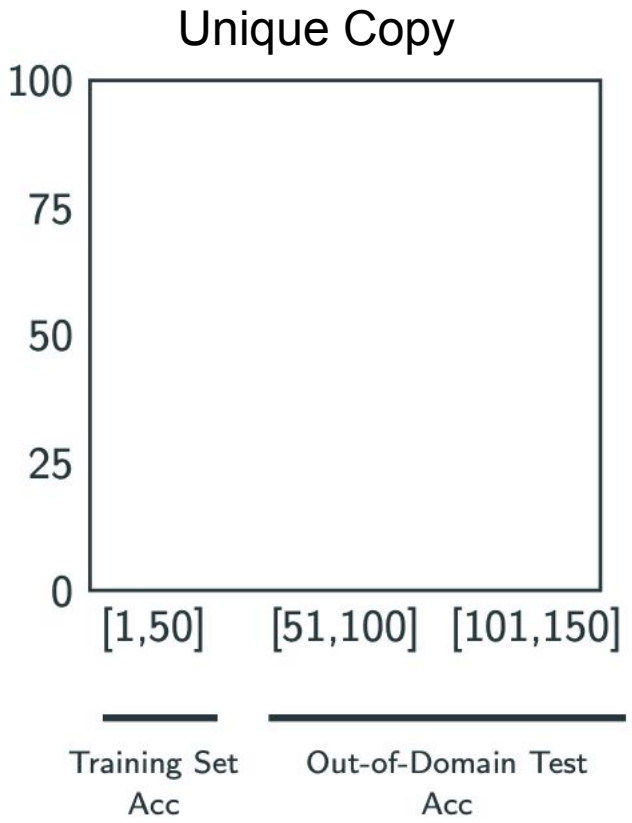
$M(i) := C_1(i) \ge C_0(i)$ compare them (3)

	%	0	0	1	1	0	1	1	1	0	$	T	@
$C_1(i)$	0	0	0	1	2	2	3	4	5	5	5		
$C_0(i)$	0	1	2	2	2	3	3	3	3	4	4		
M(i)	T	F	F	F	T	F	T	T	T	T	T		

C-RASP: Yang&Chiang 2024, COLM

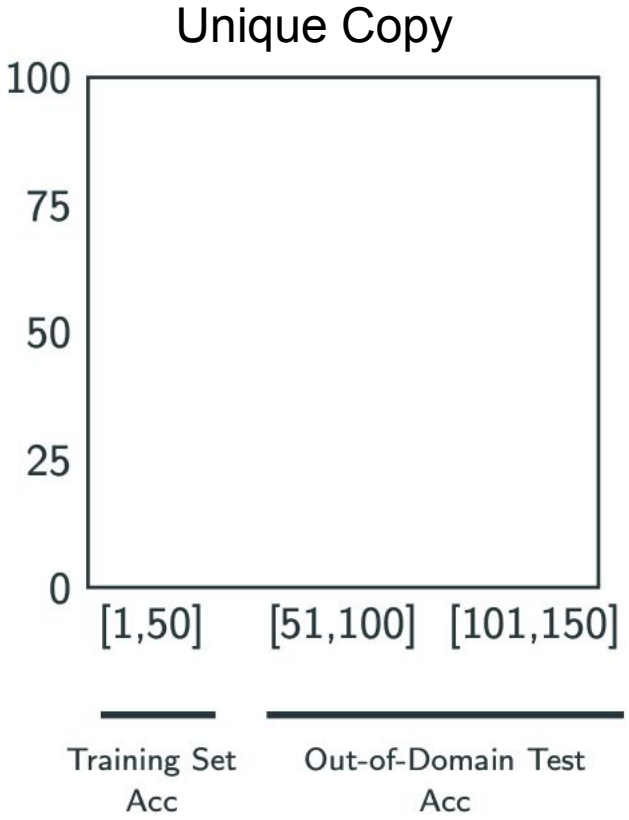
% u r s z $ u r s z @


Unique Copy
100
75
50
25
0
[1,50]
[51,100]
[101,150]
Training Set Acc
Out-of-Domain Test Acc



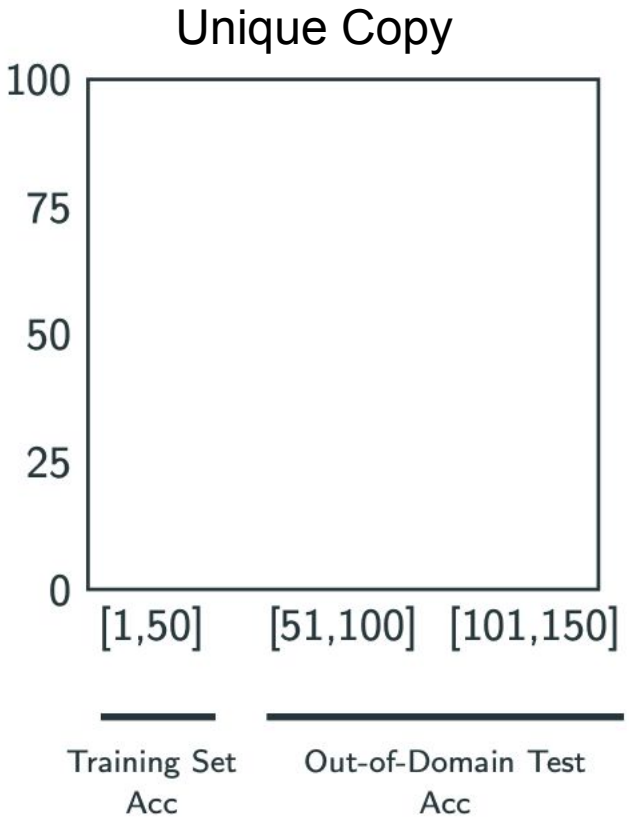
% u r s z $ u …

collect immediately preceding token



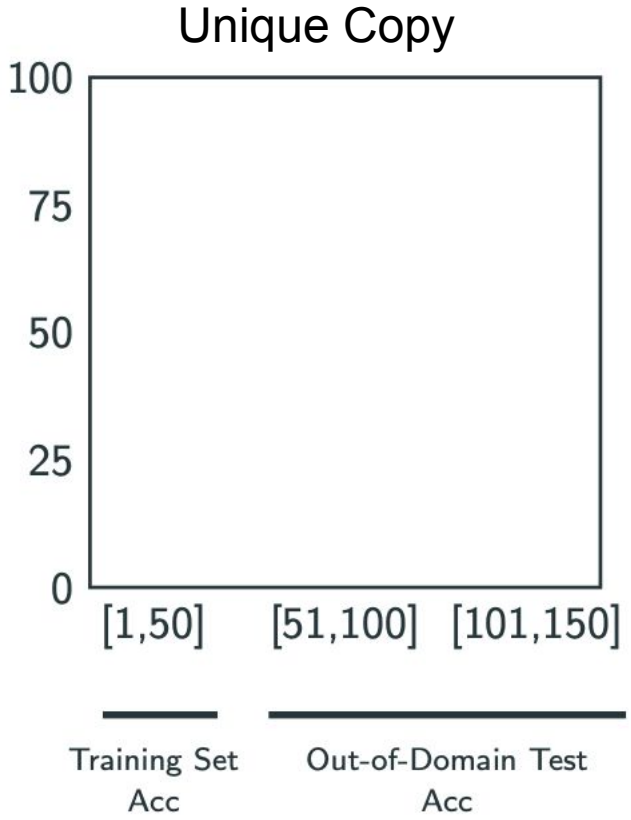
	%	u	r	s	z	$	u	…
prev = u	0	0	1	0	0	0	0	

collect immediately preceding token



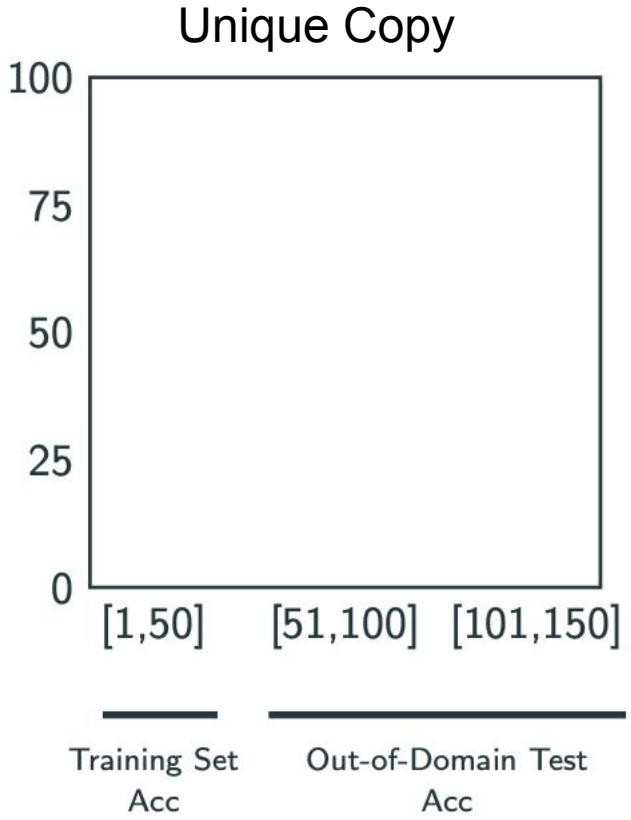
	%	u	r	s	z	$	u	…
prev = u	0	0	1	0	0	0	0	
prev = r	0	0	0	1	0	0	0	

collect immediately preceding token



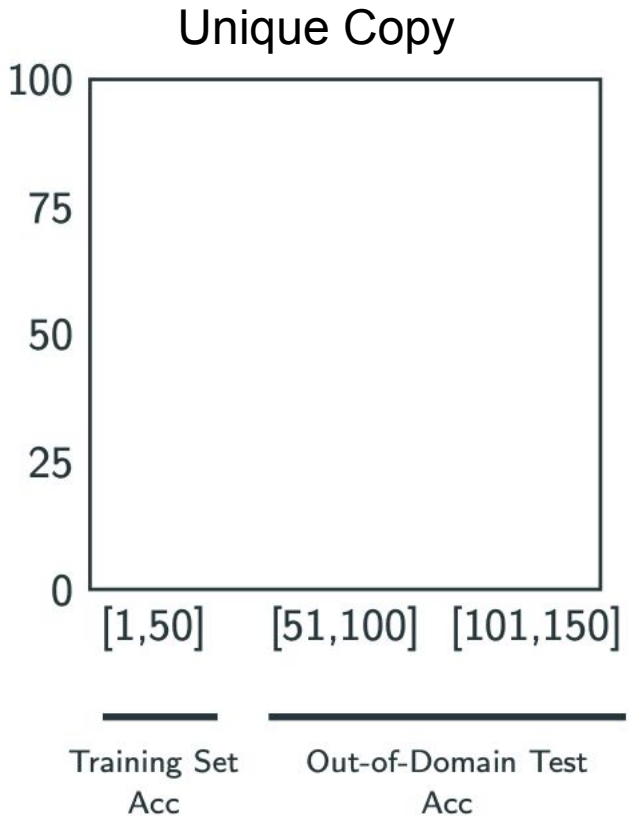
	%	u	r	s	z	$	u	…
prev = u	0	0	1	0	0	0	0	
prev = r	0	0	0	1	0	0	0	
prev = s	0	0	0	0	1	0	0	

collect immediately preceding token



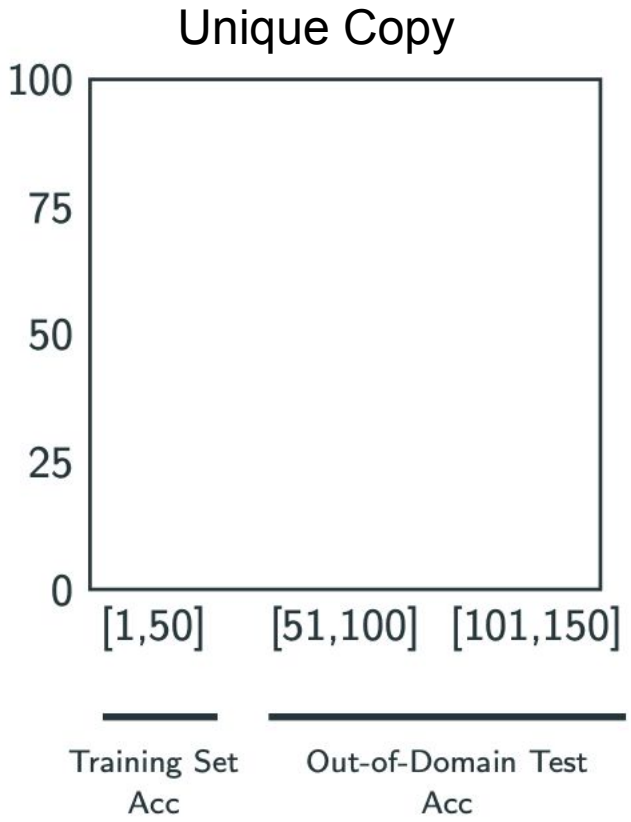
	%	u	r	s	z	$	u	…
prev = u	0	0	1	0	0	0	0	
prev = r	0	0	0	1	0	0	0	
prev = s	0	0	0	0	1	0	0	
prev = z	0	0	0	0	0	1	0	

collect immediately preceding token



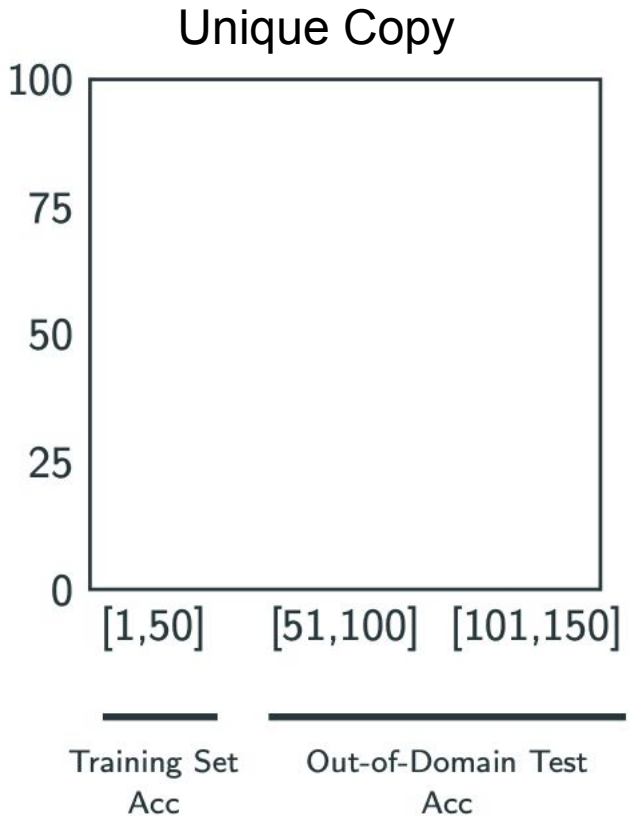
	%	u	r	s	z	$	u	…
prev = u	0	0	1	0	0	0	0	
prev = r	0	0	0	1	0	0	0	
prev = s	0	0	0	0	1	0	0	
prev = z	0	0	0	0	0	1	0	
ur	0	0	1	1	1	1	1	

collect bigram statistics



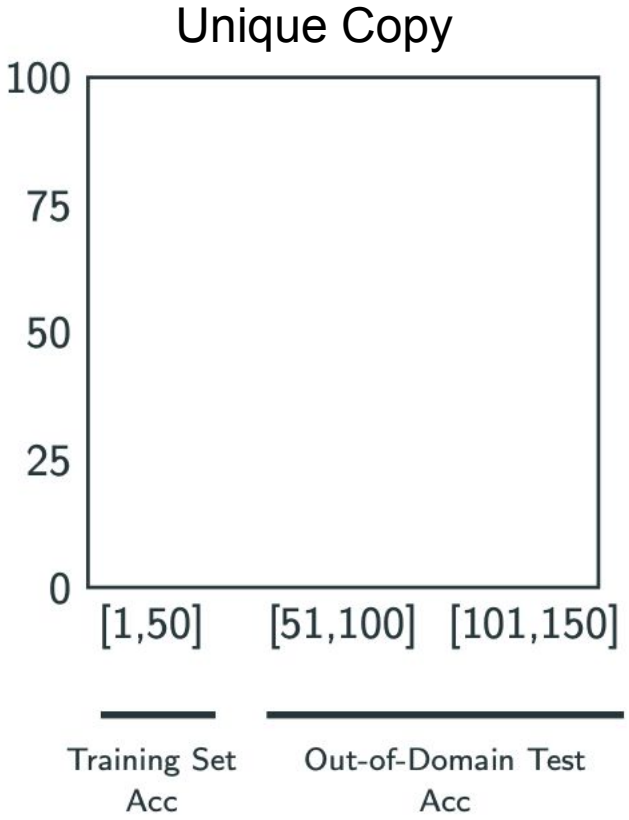
	%	u	r	s	z	$	u	…
prev = u	0	0	1	0	0	0	0	
prev = r	0	0	0	1	0	0	0	
prev = s	0	0	0	0	1	0	0	
prev = z	0	0	0	0	0	1	0	
ur	0	0	1	1	1	1	1	
rs	0	0	0	1	1	1	1	

collect bigram statistics



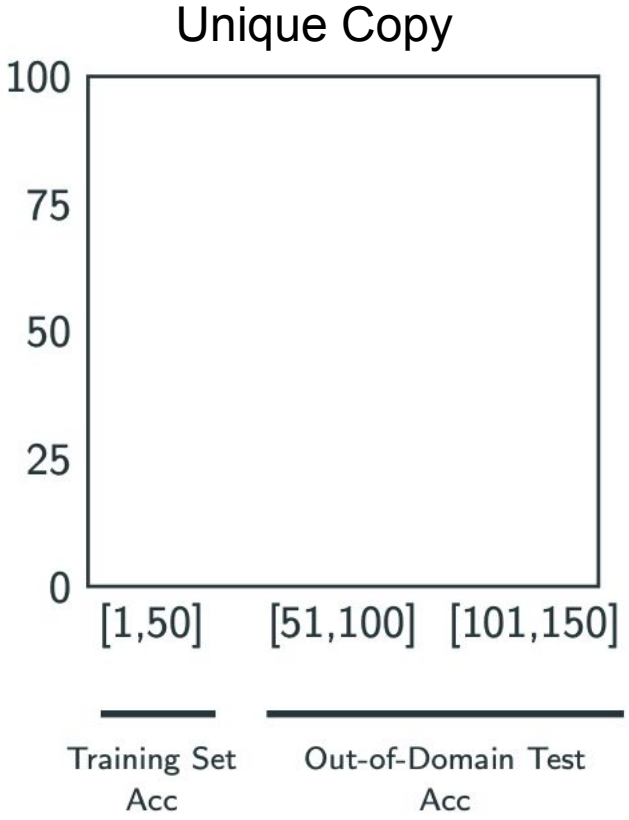
	%	u	r	s	z	$	u	…
prev = u	0	0	1	0	0	0	0	
prev = r	0	0	0	1	0	0	0	
prev = s	0	0	0	0	1	0	0	
prev = z	0	0	0	0	0	1	0	
ur	0	0	1	1	1	1	1	
rs	0	0	0	1	1	1	1	
sz	0	0	0	0	1	1	1	

collect bigram statistics



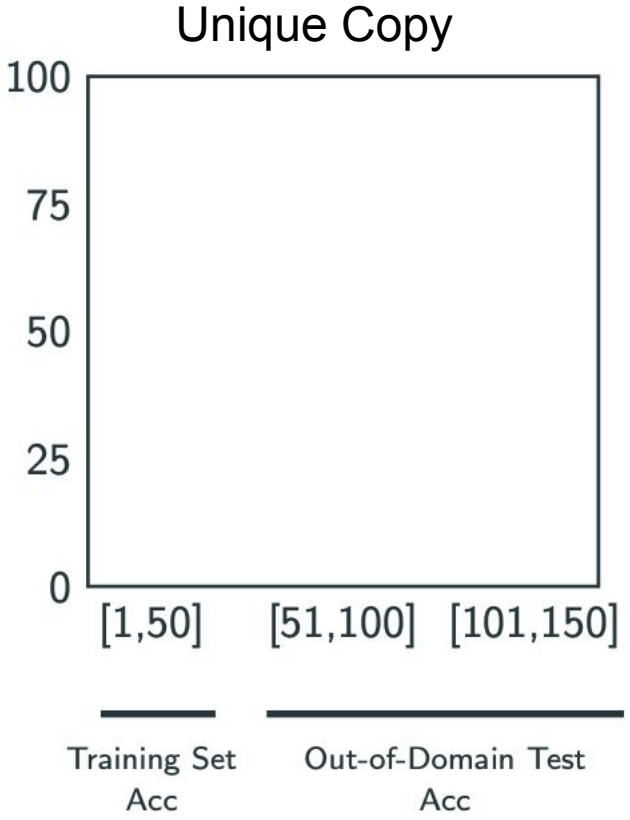
	%	u	r	s	z	$	u	…
prev = u	0	0	1	0	0	0	0	
prev = r	0	0	0	1	0	0	0	
prev = s	0	0	0	0	1	0	0	
prev = z	0	0	0	0	0	1	0	
ur	0	0	1	1	1	1	1	
rs	0	0	0	1	1	1	1	
sz	0	0	0	0	1	1	1	
us	0	0	0	0	0	0	0	

collect bigram statistics

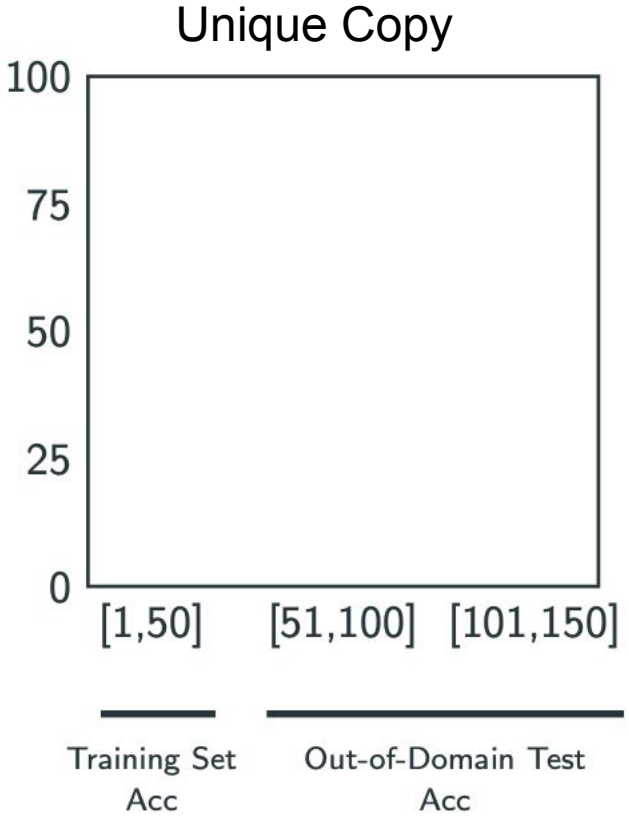


	%	u	r	s	z	$	u	…
prev = u	0	0	1	0	0	0	0	
prev = r	0	0	0	1	0	0	0	
prev = s	0	0	0	0	1	0	0	
prev = z	0	0	0	0	0	1	0	
ur	0	0	1	1	1	1	1	
rs	0	0	0	1	1	1	1	
sz	0	0	0	0	1	1	1	
us	0	0	0	0	0	0	0	
…				…				

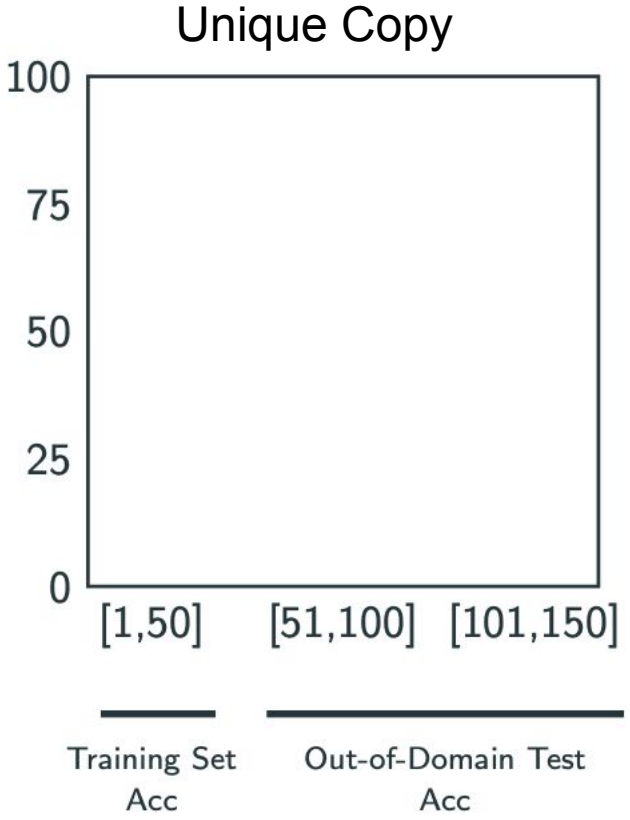
collect bigram statistics



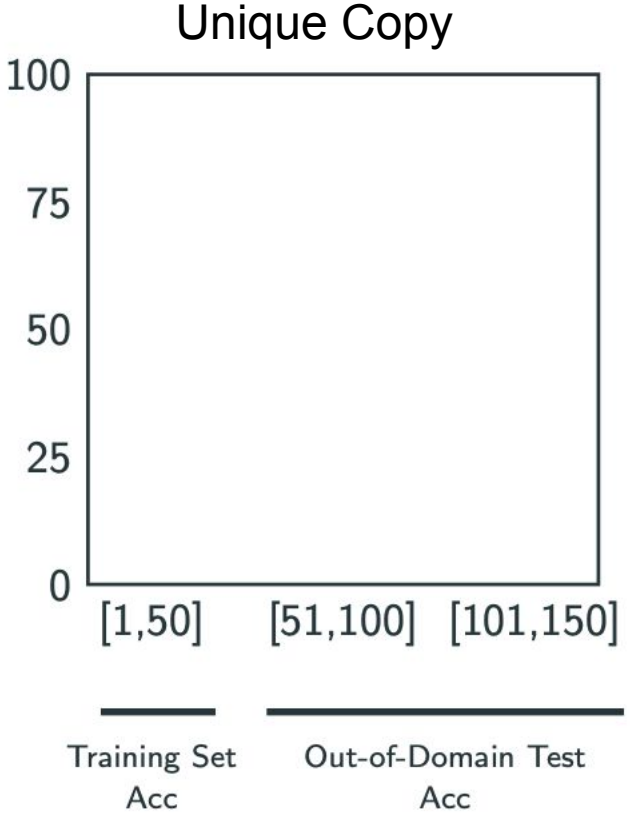
	%	u	r	s	z	$	**u**	…
prev = u	0	0	1	0	0	0	0	
prev = r	0	0	0	1	0	0	0	
prev = s	0	0	0	0	1	0	0	
prev = z	0	0	0	0	0	1	0	
ur	0	0	1	1	1	1	1	
rs	0	0	0	1	1	1	1	
sz	0	0	0	0	1	1	1	
us	0	0	0	0	0	0	0	
. . .					. . .			



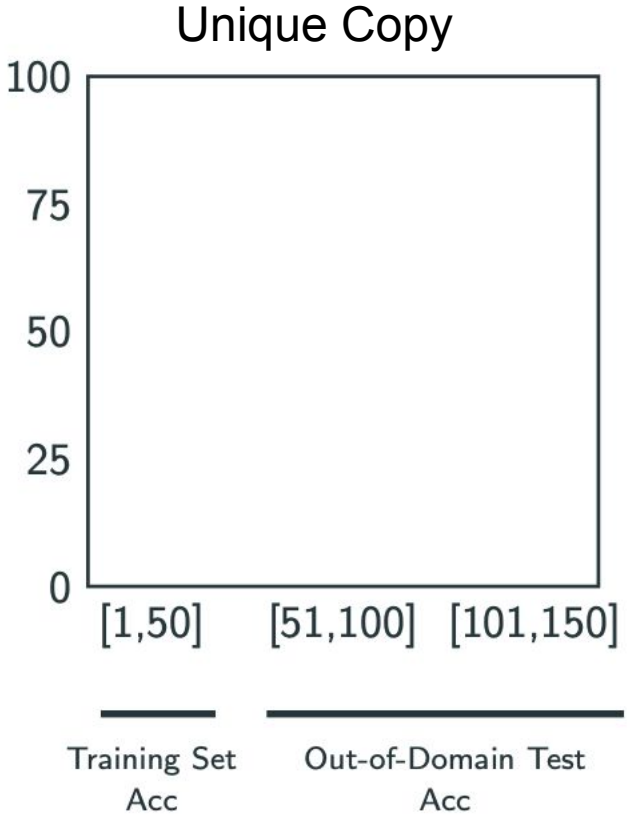
	%	u	r	s	z	$	**u**	…
prev = u	0	0	1	0	0	0	0	
prev = r	0	0	0	1	0	0	0	
prev = s	0	0	0	0	1	0	0	
prev = z	0	0	0	0	0	1	0	
ur	0	0	1	1	1	1	**1**	
rs	0	0	0	1	1	1	1	
sz	0	0	0	0	1	1	1	
us	0	0	0	0	0	0	0	
…					…			



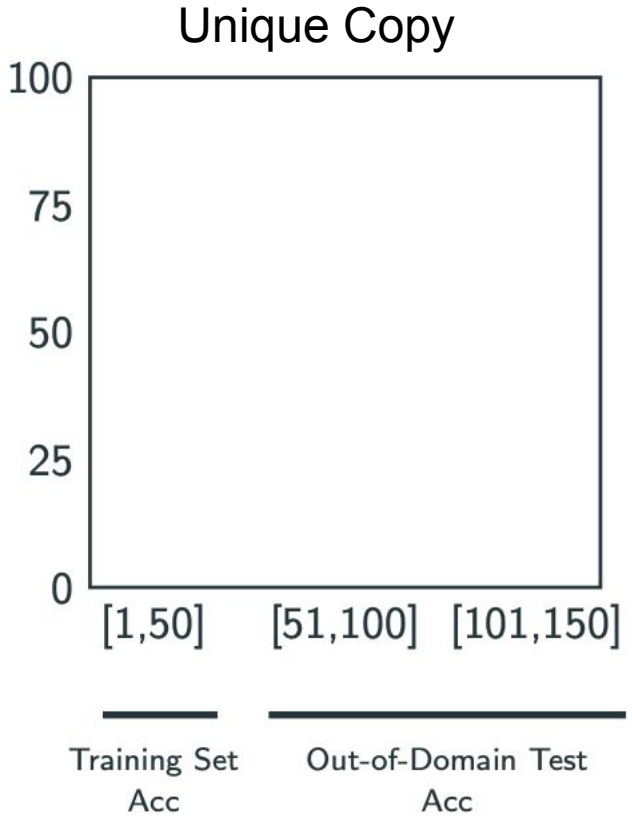
	%	u	r	s	z	$	**u**	**r**
prev = u	0	0	1	0	0	0	0	
prev = r	0	0	0	1	0	0	0	
prev = s	0	0	0	0	1	0	0	
prev = z	0	0	0	0	0	1	0	
ur	0	0	1	1	1	1	**1**	
rs	0	0	0	1	1	1	1	
sz	0	0	0	0	1	1	1	
us	0	0	0	0	0	0	0	
...				...				



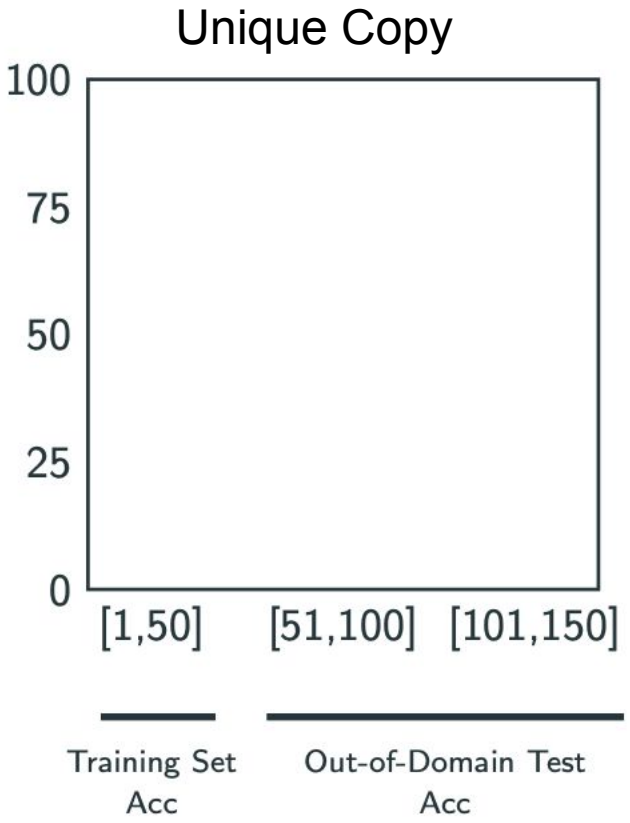
	%	u	r	s	z	$	u	**r**
prev = u	0	0	1	0	0	0	0	1
prev = r	0	0	0	1	0	0	0	0
prev = s	0	0	0	0	1	0	0	0
prev = z	0	0	0	0	0	1	0	0
ur	0	0	1	1	1	1	1	1
rs	0	0	0	1	1	1	1	1
sz	0	0	0	0	1	1	1	1
us	0	0	0	0	0	0	0	0
...					...			



	%	u	r	s	z	$	u	**r**
prev = u	0	0	1	0	0	0	0	1
prev = r	0	0	0	1	0	0	0	0
prev = s	0	0	0	0	1	0	0	0
prev = z	0	0	0	0	0	1	0	0
ur	0	0	1	1	1	1	1	1
rs	0	0	0	1	1	1	1	**1**
sz	0	0	0	0	1	1	1	1
us	0	0	0	0	0	0	0	0
...				...				

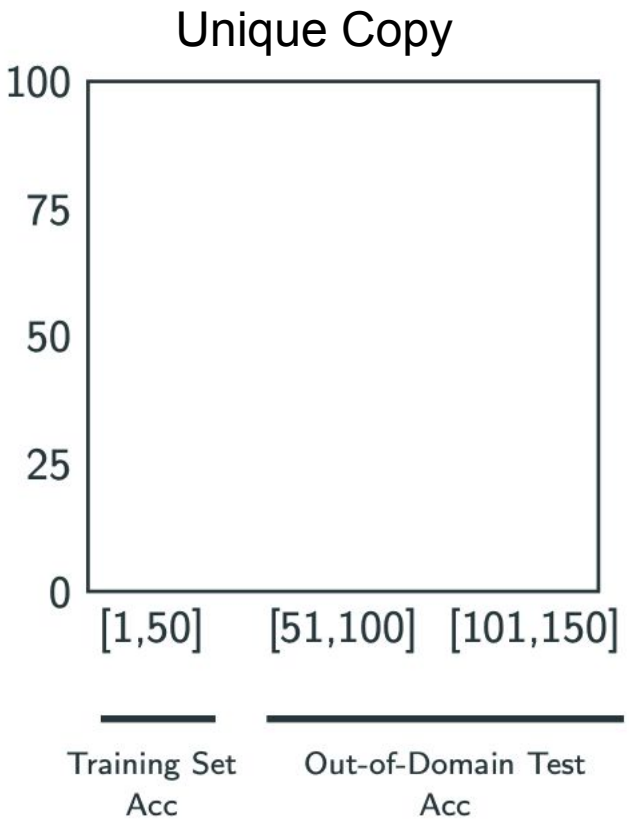


	%	u	r	s	z	$	u	**r**	**s**
prev = u	0	0	1	0	0	0	0	1	
prev = r	0	0	0	1	0	0	0	0	
prev = s	0	0	0	0	1	0	0	0	
prev = z	0	0	0	0	0	1	0	0	
ur	0	0	1	1	1	1	1	1	
rs	0	0	0	1	1	1	1	**1**	
sz	0	0	0	0	1	1	1	1	
us	0	0	0	0	0	0	0	0	
...				...					



	%	u	r	s	z	$	u	r	s	z	$
prev = u	0	0	1	0	0	0	0	1	0	0	0
prev = r	0	0	0	1	0	0	0	0	1	0	0
prev = s	0	0	0	0	1	0	0	0	0	1	0
prev = z	0	0	0	0	0	1	0	0	0	0	1
ur	0	0	1	1	1	1	1	1	1	1	1
rs	0	0	0	1	1	1	1	1	1	1	1
sz	0	0	0	0	1	1	1	1	1	1	1
us	0	0	0	0	0	0	0	0	0	0	0
...							...				

% u r s z $ u r s z @



Unique Copy (Induction Head)

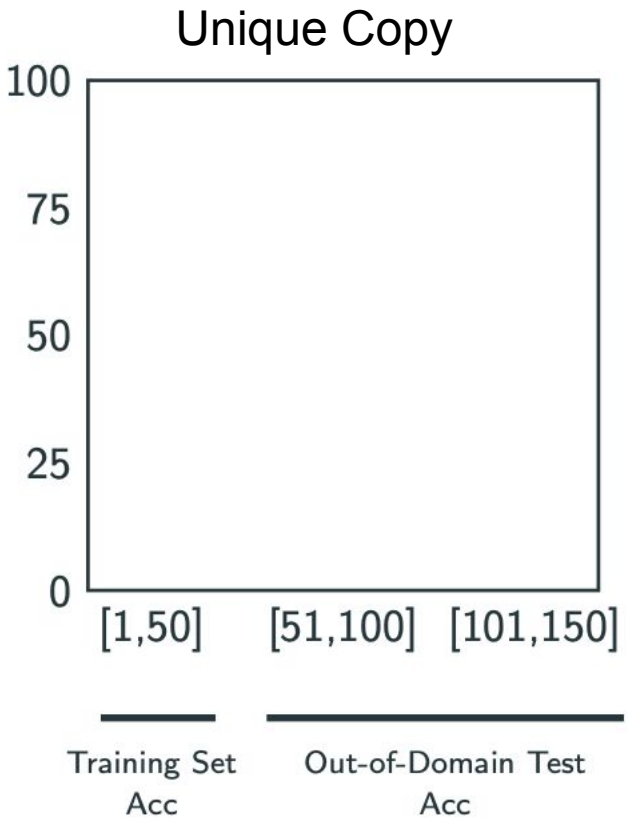
$$\vdots \qquad (1)$$

$$CP_a(i) := \#[j \le i, j = i-1]\ Q_a(j) \qquad (2)$$

$$PRED_a(i) := CP_a(i) \ge 1 \qquad (3)$$

check preceding token

% u r s z $ u r s z @



Unique Copy (Induction Head)

$$\vdots \tag{1}$$

$$CP_a(i) := \#[j \leq i, j = i-1]\ Q_a(j) \tag{2}$$

$$PRED_a(i) := CP_a(i) \geq 1 \tag{3}$$

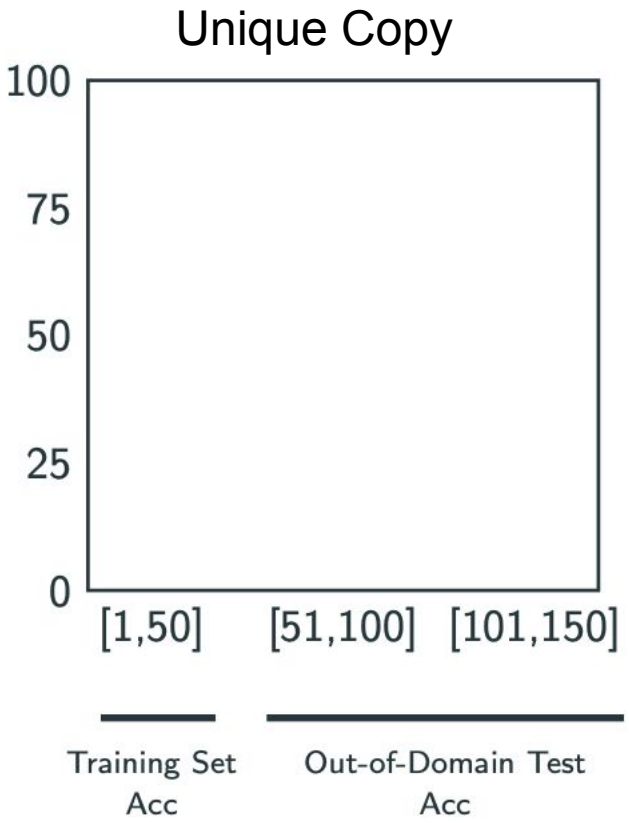
$$\vdots \tag{4}$$

$$CBIGRAM_{ab} := \#[j \leq i]\ Q_b(j) \wedge PRED_a(j) \tag{5}$$

$$EXISTS_{ab} := CBIGRAM_{ab}(i) \geq 1 \tag{6}$$

collect all bigram statistics

% u r s z $ u r s z @



Unique Copy (Induction Head)

$$\vdots \tag{1}$$

$$CP_a(i) := \#[j \le i, j = i-1]\ Q_a(j) \tag{2}$$

$$PRED_a(i) := CP_a(i) \ge 1 \tag{3}$$

$$\vdots \tag{4}$$

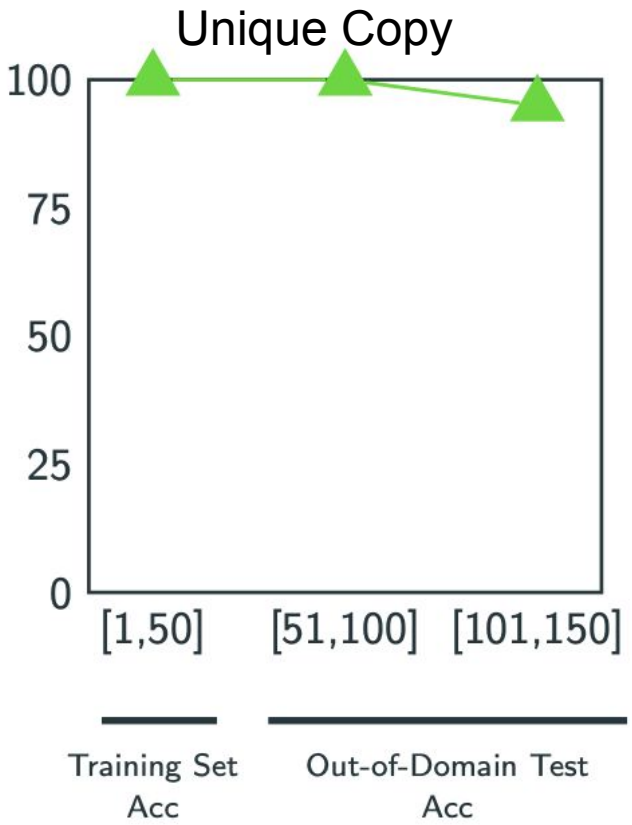
$$CBIGRAM_{ab} := \#[j \le i]\ Q_b(j) \wedge PRED_a(j) \tag{5}$$

$$EXISTS_{ab} := CBIGRAM_{ab}(i) \ge 1 \tag{6}$$

$$\vdots \tag{7}$$

$$NEXT_a(i) := \bigvee_{\sigma \in \Sigma} [Q_\sigma(i) \wedge EXISTS_{\sigma a}(i)] \tag{8}$$

use bigram statistics to output next token



% u r s z $ u r s z @

Unique Copy (Induction Head)

$$\vdots \tag{1}$$

$$CP_a(i) := \#\,[j \le i, j = i-1]\ \ Q_a(j) \tag{2}$$

$$PRED_a(i) := CP_a(i) \ge 1 \tag{3}$$

$$\vdots \tag{4}$$

$$CBIGRAM_{ab} := \#\,[j \le i]\ \ Q_b(j) \wedge PRED_a(j) \tag{5}$$

$$EXISTS_{ab} := CBIGRAM_{ab}(i) \ge 1 \tag{6}$$

$$\vdots \tag{7}$$

$$NEXT_a(i) := \bigvee_{\sigma \in \Sigma} [Q_\sigma(i) \wedge EXISTS_{\sigma a}(i)] \tag{8}$$

Newly added (or modified) positionally-aware operations. Needed to model positional encodings.

Boolean-Valued Operations

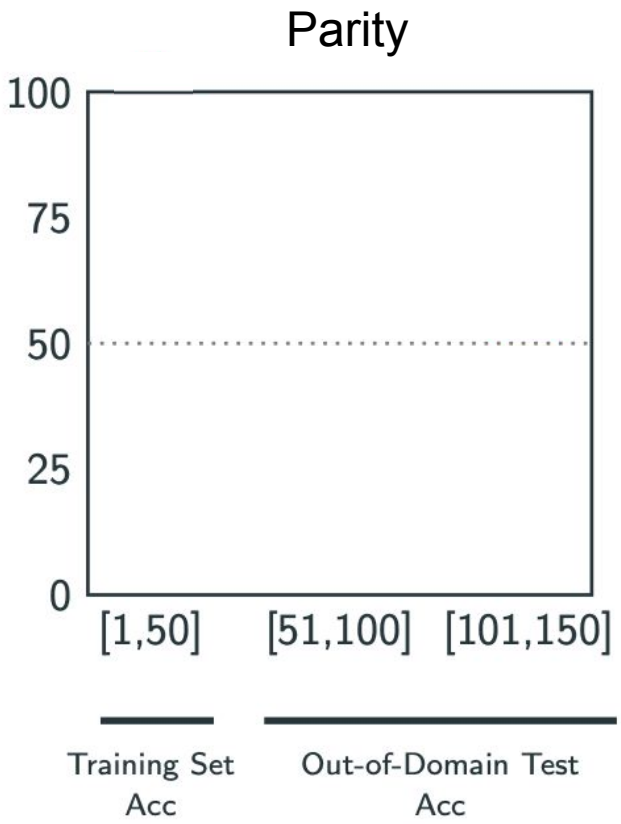
Initial	$P(i) := Q_\sigma(i)$ *for* $\sigma \in \Sigma$
Boolean	$P(i) := \neg P_1(i)$ $P(i) := P_1(i) \wedge P_2(i)$
Constant	$P(i) := \top$
Positional	$P(i) := \phi(i)$ *for* $\phi \in \Phi$
Comparison	$P(i) := C_1(i) \leq C_2(i)$

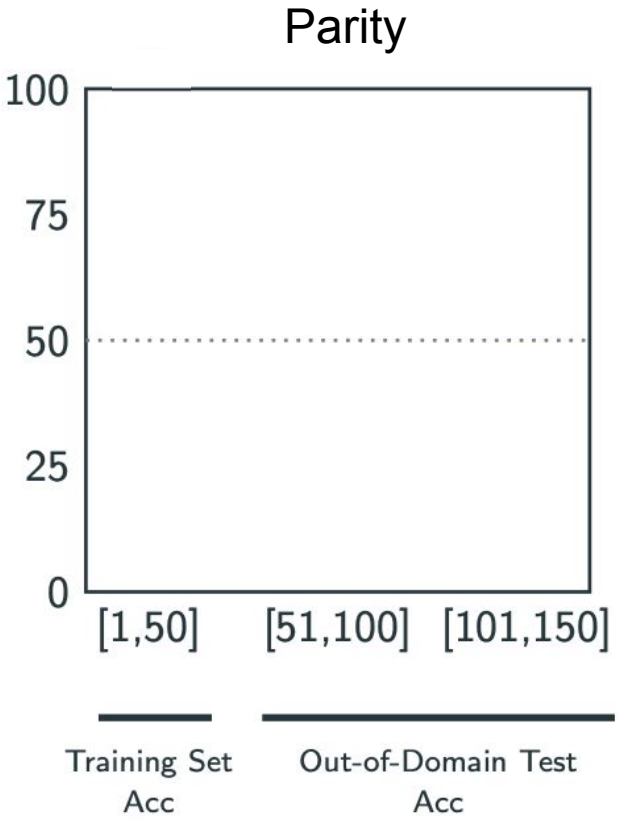
Count-Valued Operations

Counting	$C(i) := \#\left[j \leq i, \psi(i,j)\right]\ P(j)$ *for* $\psi \in \Psi \cup \{\top\}$
Conditional	$C(i) := P(i)\ ?\ C_1(i)\ :\ C_2(i)$
Addition	$C(i) := C_1(i) + C_2(i)$
Subtraction	$C(i) := C_1(i) - C_2(i)$
Constant	$C(i) := 1$

Operations from original C-RASP definitions by Yang&Chiang 2024 (COLM).

% 0 1 1 0 $ even @


Parity
100
75
50
25
0
[1,50]
[51,100]
[101,150]
Training Set Acc
Out-of-Domain Test Acc



% 0 1 1 0 $ even @

Parity

⋮ (1)

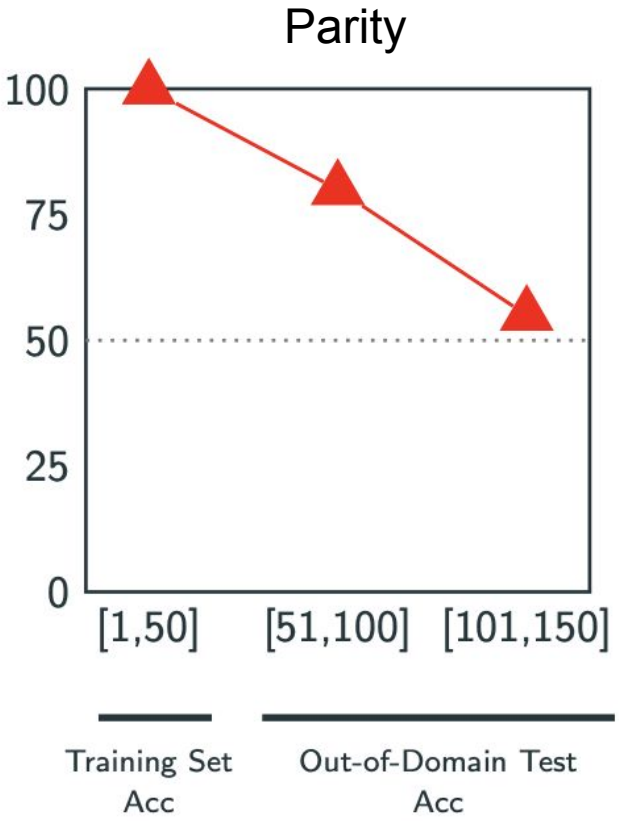
⋮ (2)

??? (3)

Provably no C-RASP program!

C-RASP[periodic, local] for Parity

$\Longrightarrow$ C-RASP[$\emptyset$] for $(aa)^*$



% 0 1 1 0 $ even @

Parity

⋮ (1)

⋮ (2)

??? (3)

Provably no C-RASP program!

C-RASP[periodic, local] for Parity

$\Longrightarrow$ C-RASP[$\emptyset$] for $(aa)^*$

Same holds for more complex state tracking.

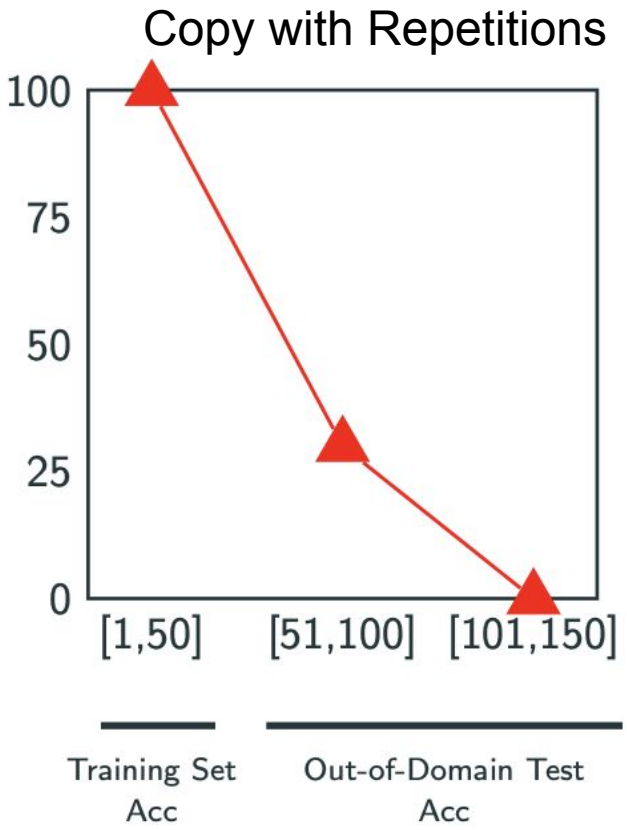
Paul had an encyclopedia.
Mary had a novel.
Paul gave his book to Josh.
Mary gave her book to Paul.
Josh gave his book to Kim.
Paul gave his book to Ruth.
Who has the encyclopedia?

Answer: Kim has the encyclopedia.

People exchanging books

⋮	(1)
⋮	(2)
???	(3)

Provably no C-RASP program!



% a a b a $ a a b a @

Copy With Repetition

⋮ (1)

⋮ (2)

??? (3)

Provably no C-RASP program!

rigorous proof via
communication complexity

Theorem (informal)

Theorem (informal)

Assume *f* is expressible in C-RASP.

Theorem (informal)

Assume *f* is expressible in C-RASP.

Then transformers can generalize from shorter to longer inputs.

Theorem (informal)

Assume *f* is expressible in C-RASP.

Then transformers can generalize from shorter to longer inputs.

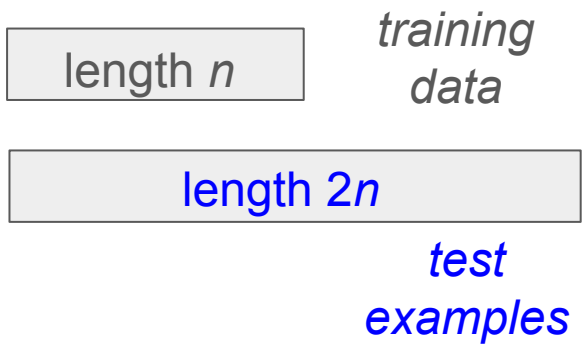
length *n*

training data

Theorem (informal)

Assume *f* is expressible in C-RASP.

Then transformers can generalize from shorter to longer inputs.



Disclaimer: What Theory can('t) do right now

Disclaimer: What Theory can('t) do right now

What it says:

- With enough data and a perfect optimizer, generalization succeeds

Disclaimer: What Theory can('t) do right now

What it says:

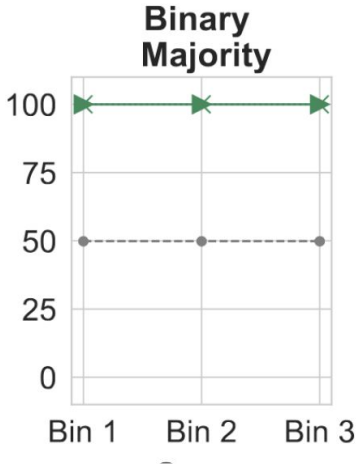
- With enough data and a perfect optimizer, generalization succeeds

What remains open for theory:

- Training with finite data & real-world optimizers
- How long do training samples have to be?

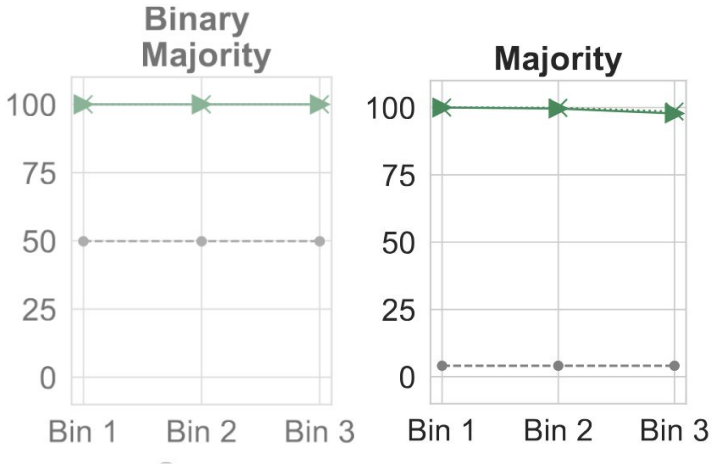
— APE
- - NoPE

– in C-RASP
– not in C-RASP



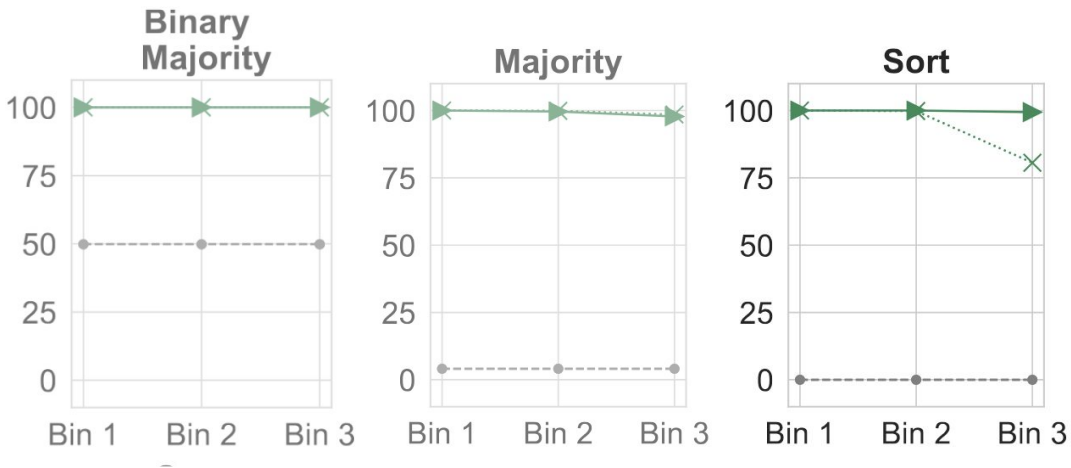
— APE
- - NoPE

– in C-RASP
– not in C-RASP



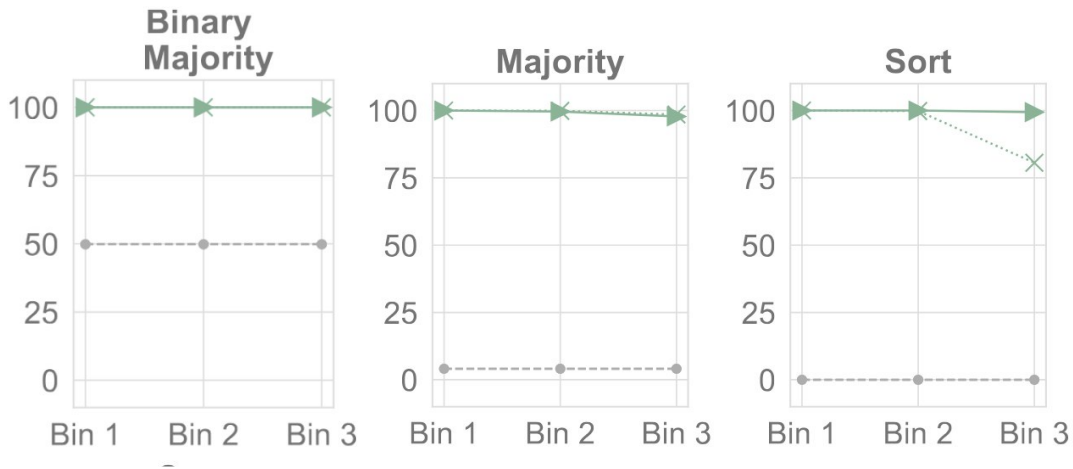
— APE
- - NoPE

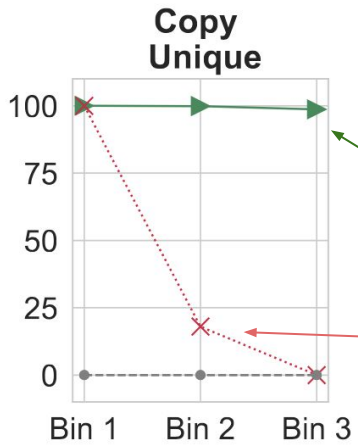
– in C-RASP
– not in C-RASP



— APE
- - NoPE

– in C-RASP
– not in C-RASP



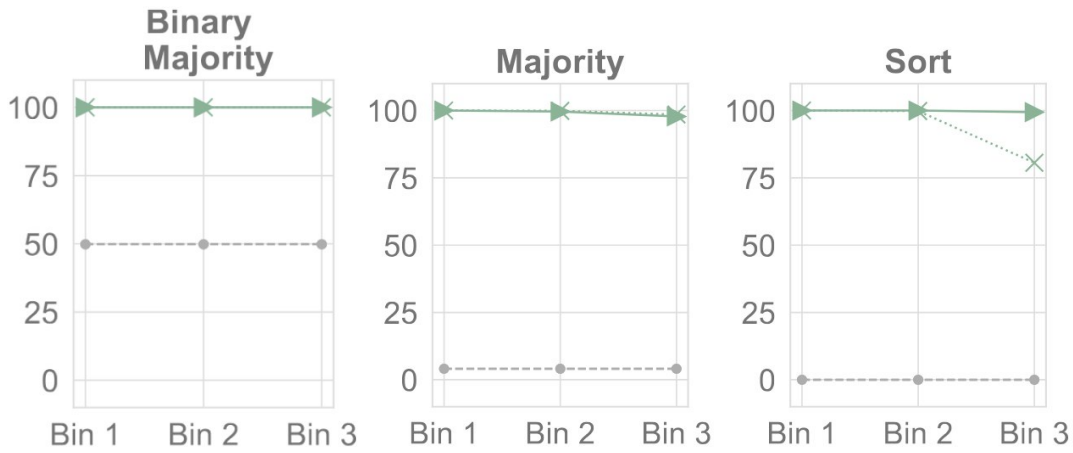


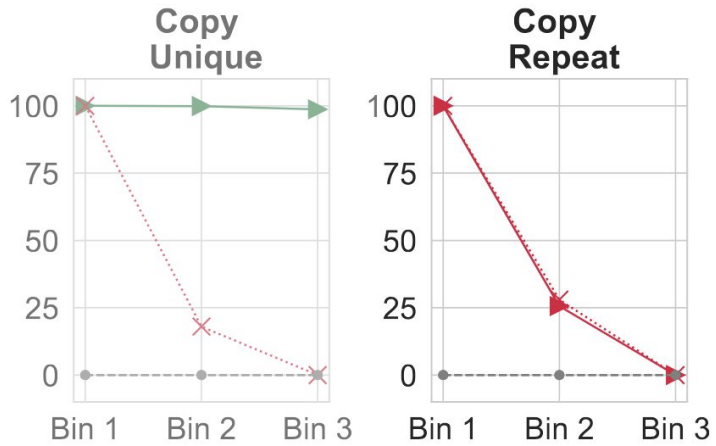
works well with PE

fails with NoPE

— APE
- - NoPE

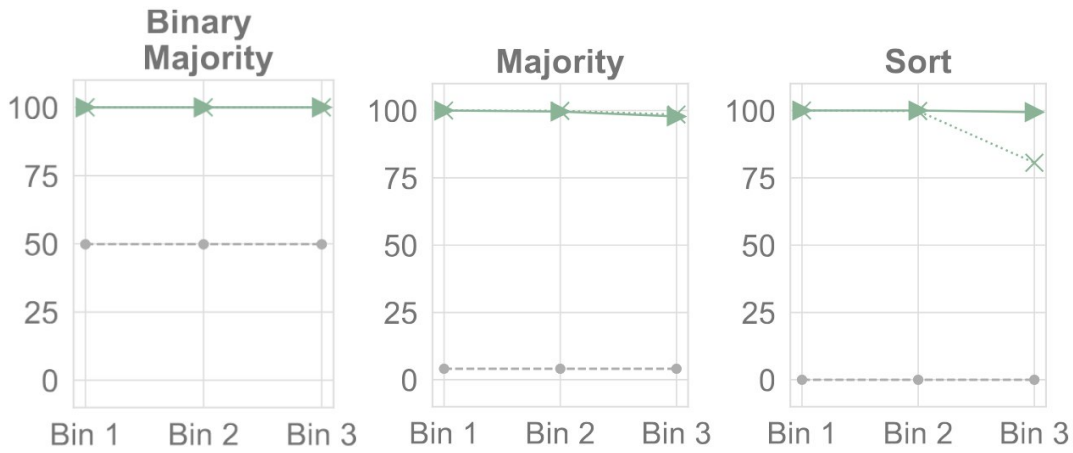
– in C-RASP
– not in C-RASP

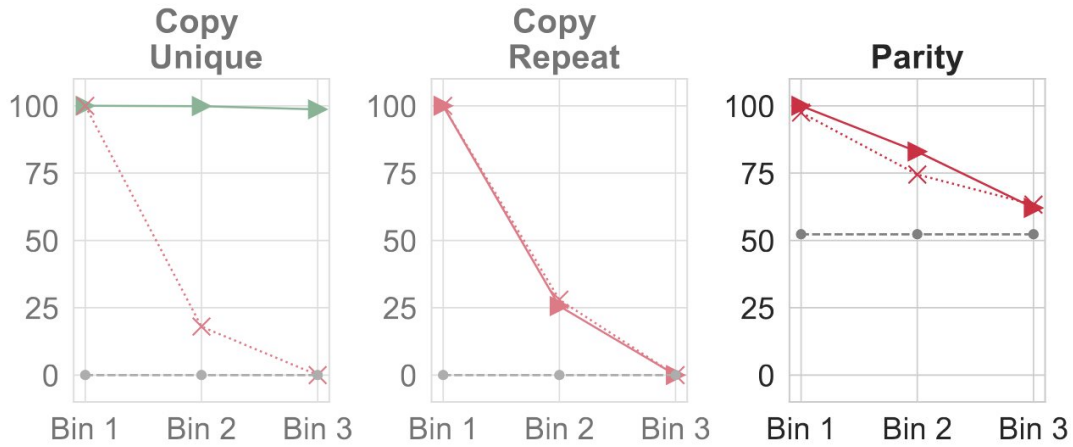




— APE
- - NoPE

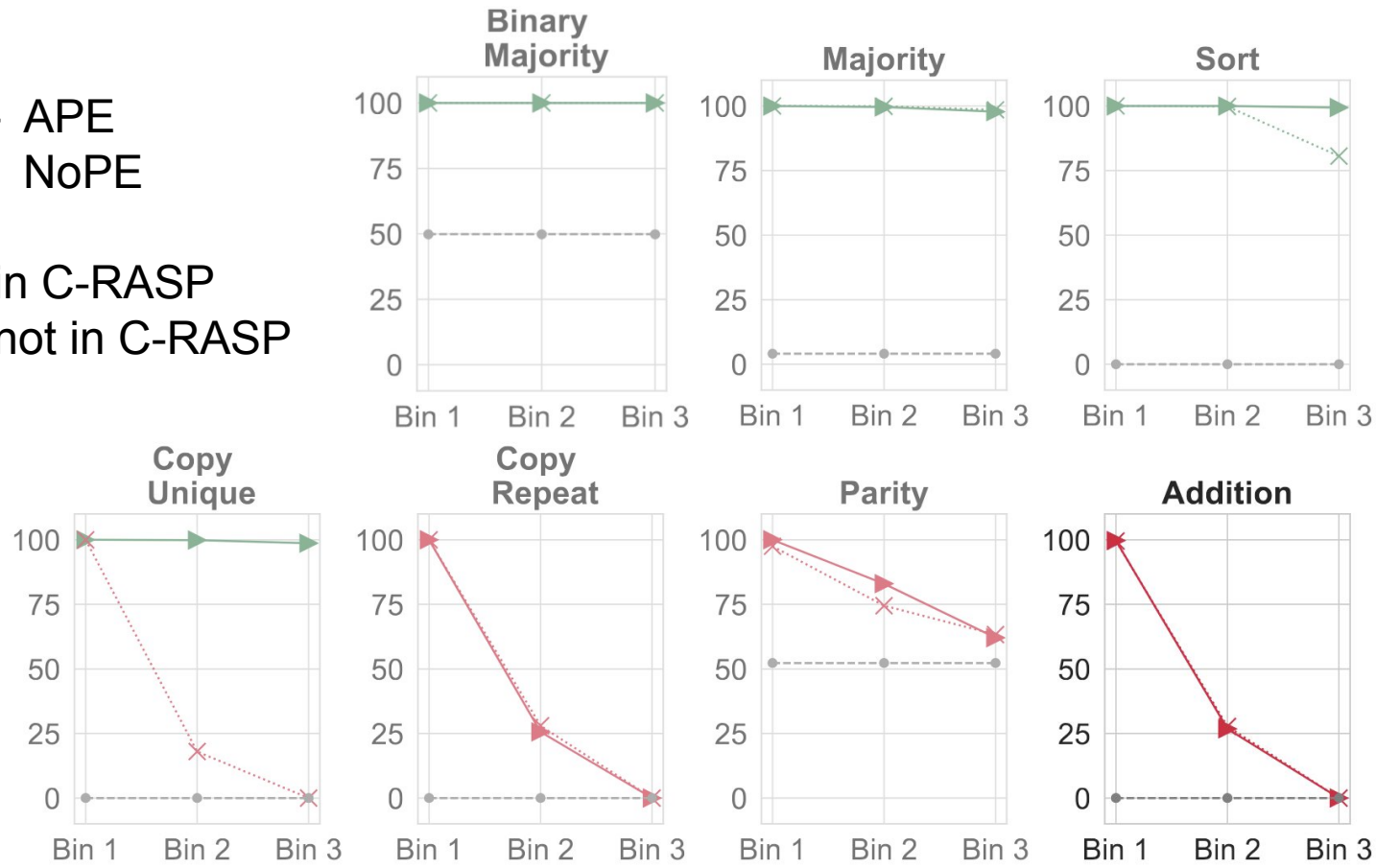
– in C-RASP
– not in C-RASP





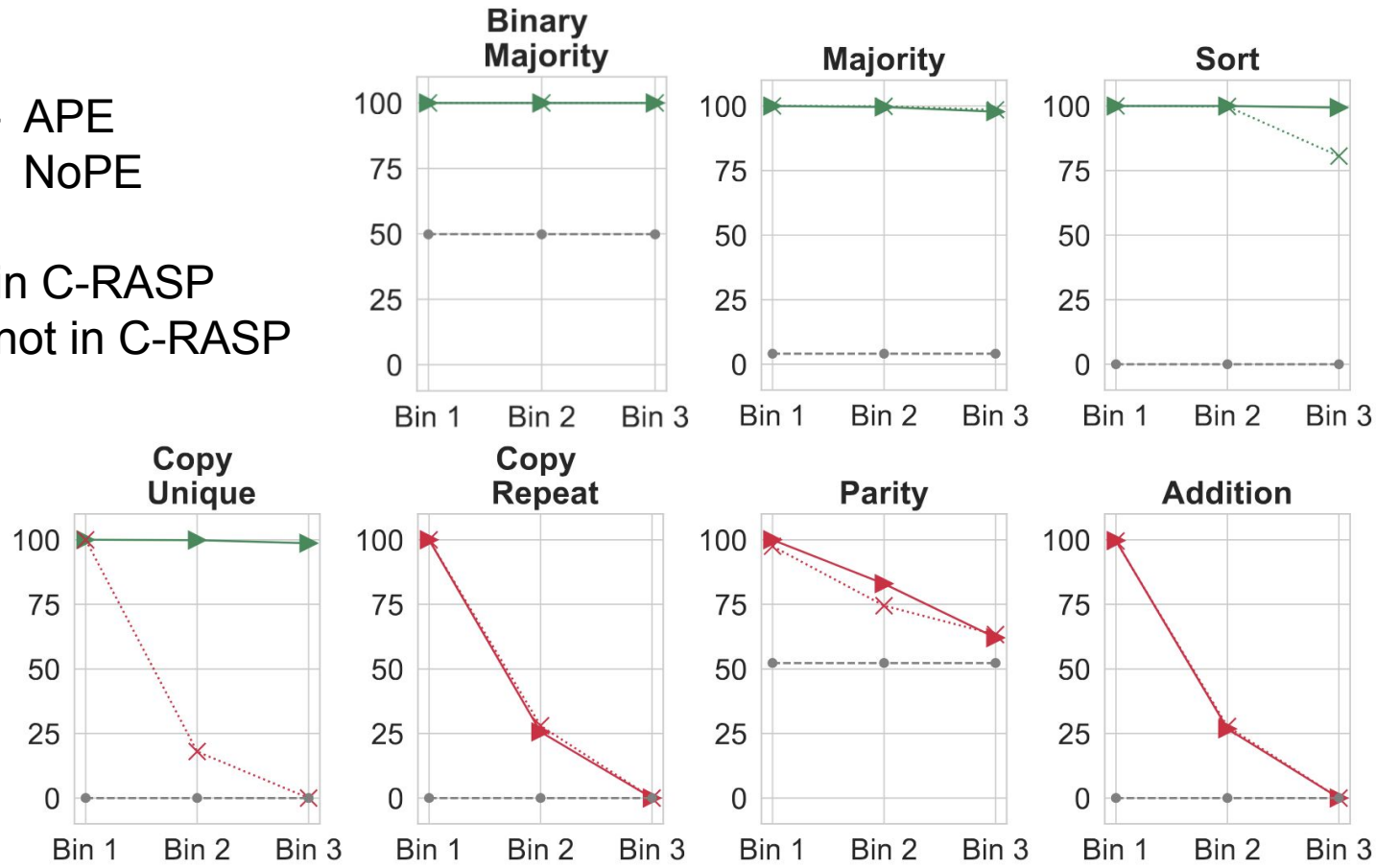
— APE
- - NoPE

– in C-RASP
– not in C-RASP



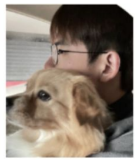
— APE
- - NoPE

– in C-RASP
– not in C-RASP



On the Ability of Transformers to Verify Plans

Yash Sarrof[♩ 1] Yupei Du[♫ 1] Katharina Stein[♫ 1] Alexander Koller[1] Sylvie Thiébaux[2 3] Michael Hahn[1]


ICML 2026

Colors Domain

Predicates:

$\mathcal{P} = \{\text{bag}(b),\ \text{color}(c),\ \text{hasColor}(b, c)\}$

Action	$\text{add}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\text{hasColor}(b, c)\}$
Action	$\text{remove}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\neg\, \text{hasColor}(b, c)\}$

Colors Domain

Predicates:

$\mathcal{P} = \{\text{bag}(b),\ \text{color}(c),\ \text{hasColor}(b, c)\}$

Action $\text{add}(c, b)$
Pre: $\{\text{bag}(b),\ \text{color}(c)\}$
Eff: $\{\text{hasColor}(b, c)\}$

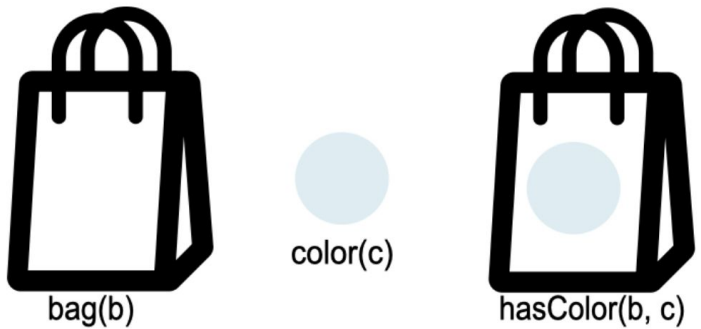
Action $\text{remove}(c, b)$
Pre: $\{\text{bag}(b),\ \text{color}(c)\}$
Eff: $\{\neg\, \text{hasColor}(b, c)\}$

Colors Domain

Predicates:

$\mathcal{P} = \{\text{bag}(b),\ \text{color}(c),\ \text{hasColor}(b, c)\}$

Action	$\text{add}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\text{hasColor}(b, c)\}$
Action	$\text{remove}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\neg\, \text{hasColor}(b, c)\}$

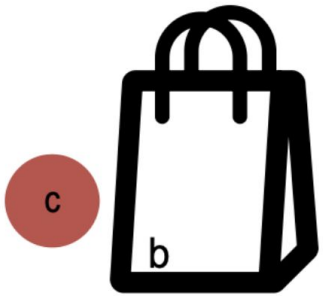


Colors Domain

Predicates:

$\mathcal{P} = \{\text{bag}(b),\ \text{color}(c),\ \text{hasColor}(b, c)\}$

Action	$\text{add}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\text{hasColor}(b, c)\}$
Action	$\text{remove}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\neg\, \text{hasColor}(b, c)\}$

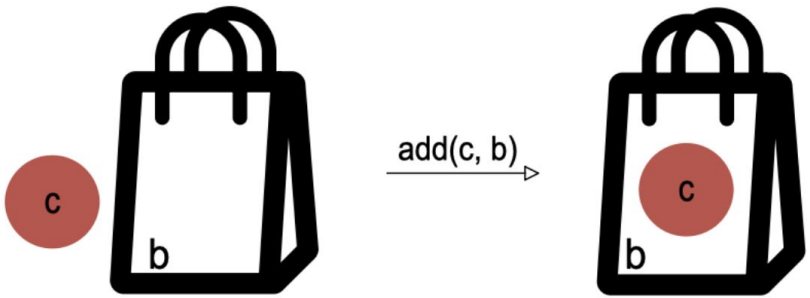


Colors Domain

Predicates:

$\mathcal{P} = \{\text{bag}(b),\ \text{color}(c),\ \text{hasColor}(b, c)\}$

Action	$\text{add}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\text{hasColor}(b, c)\}$
Action	$\text{remove}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\neg\, \text{hasColor}(b, c)\}$

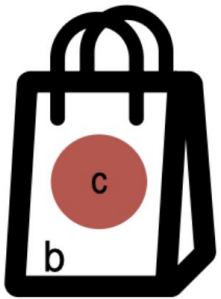


Colors Domain

Predicates:

$\mathcal{P} = \{\text{bag}(b),\ \text{color}(c),\ \text{hasColor}(b, c)\}$

Action	$\text{add}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\text{hasColor}(b, c)\}$
Action	$\text{remove}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\neg\, \text{hasColor}(b, c)\}$

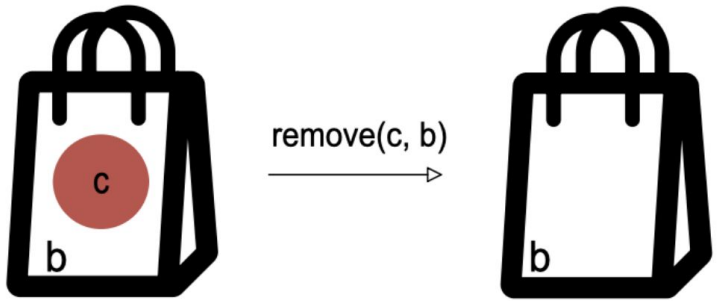


Colors Domain

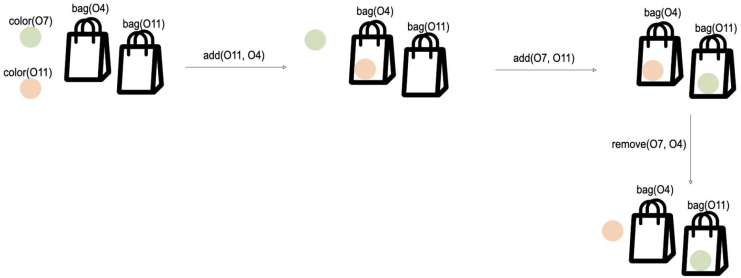
Predicates:

$\mathcal{P} = \{\text{bag}(b),\ \text{color}(c),\ \text{hasColor}(b, c)\}$

Action	$\text{add}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\text{hasColor}(b, c)\}$
Action	$\text{remove}(c, b)$
Pre:	$\{\text{bag}(b),\ \text{color}(c)\}$
Eff:	$\{\neg\, \text{hasColor}(b, c)\}$

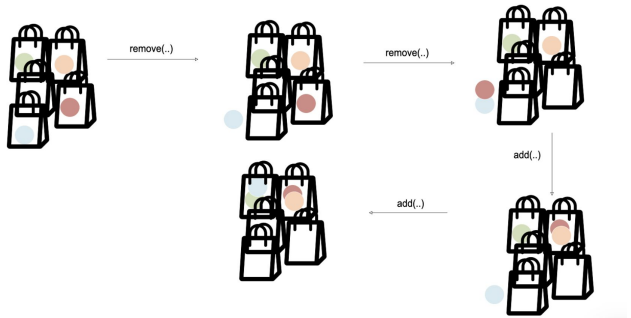


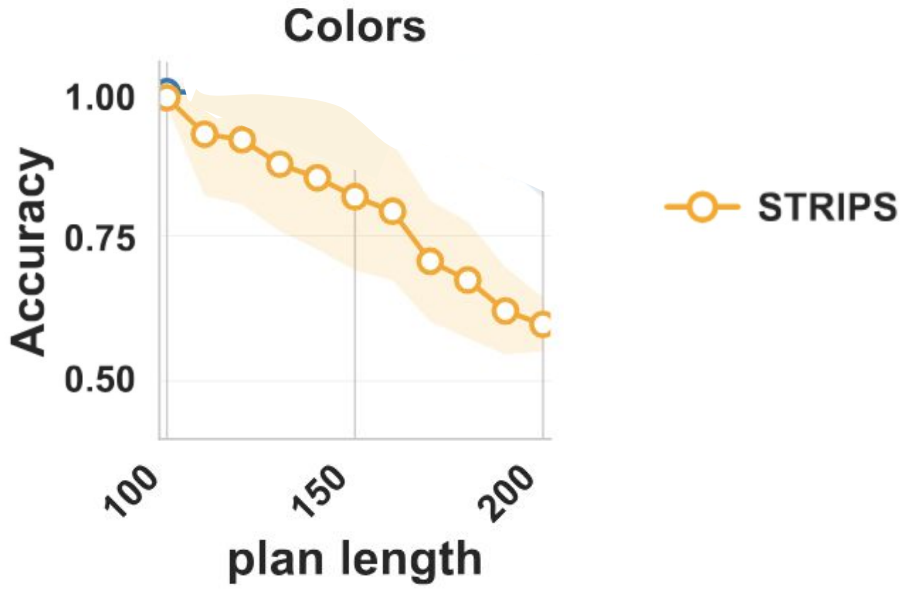
Train: Bounded objects, bounded plan length

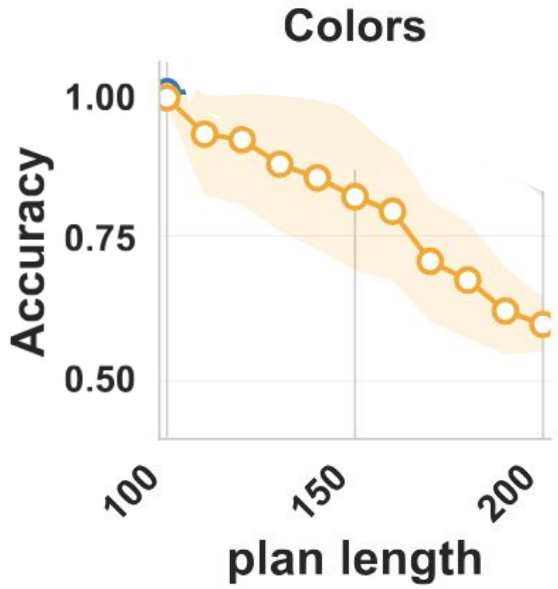


Train: Bounded objects, bounded plan length

Test: More objects, longer plans

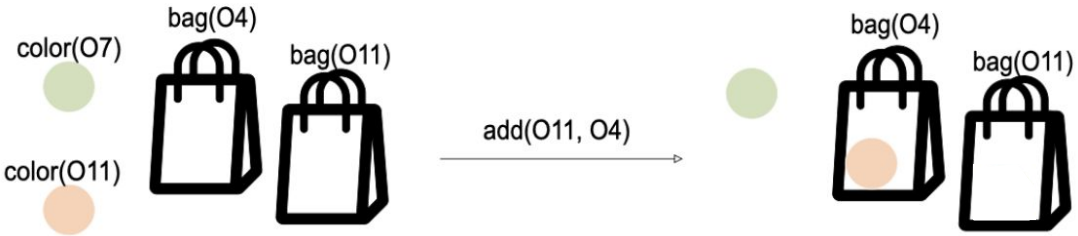



Colors
1.00
0.75
0.50
Accuracy
100
150
200
plan length
STRIPS



Why?

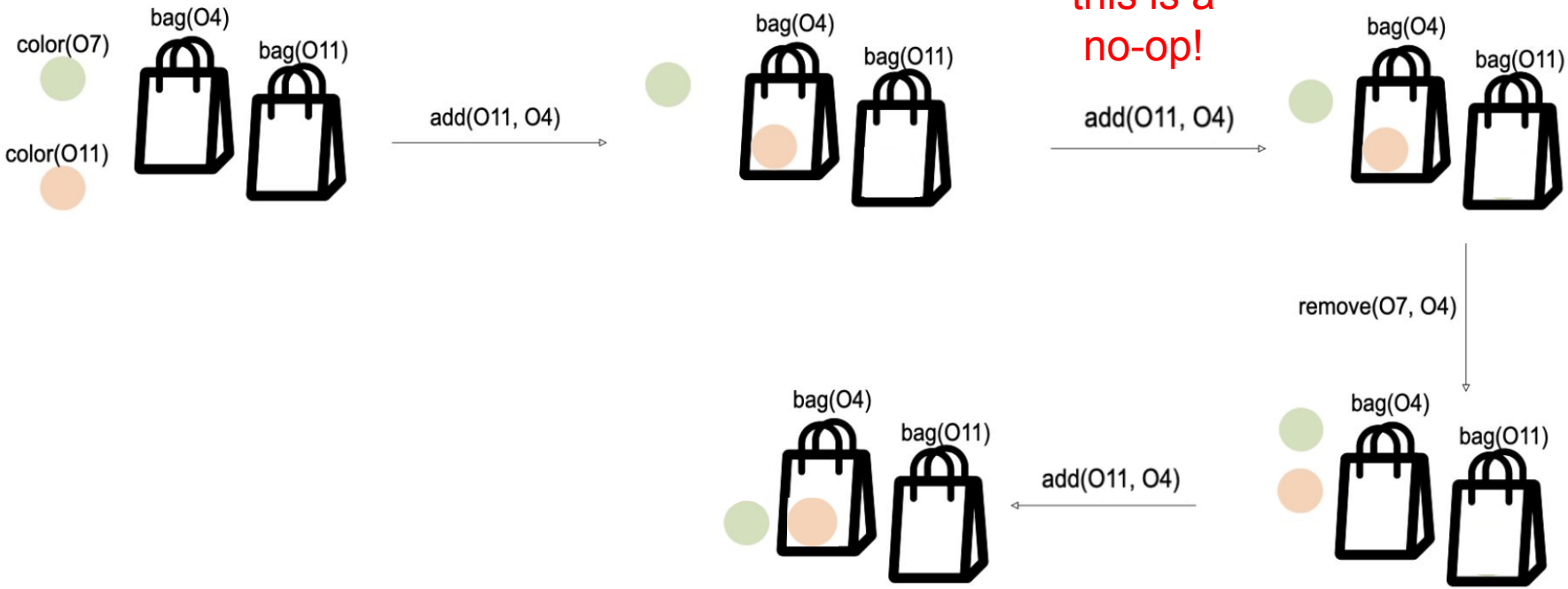

color(O7)
color(O11)
bag(O4)
bag(O11)


color(O7)
color(O11)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)

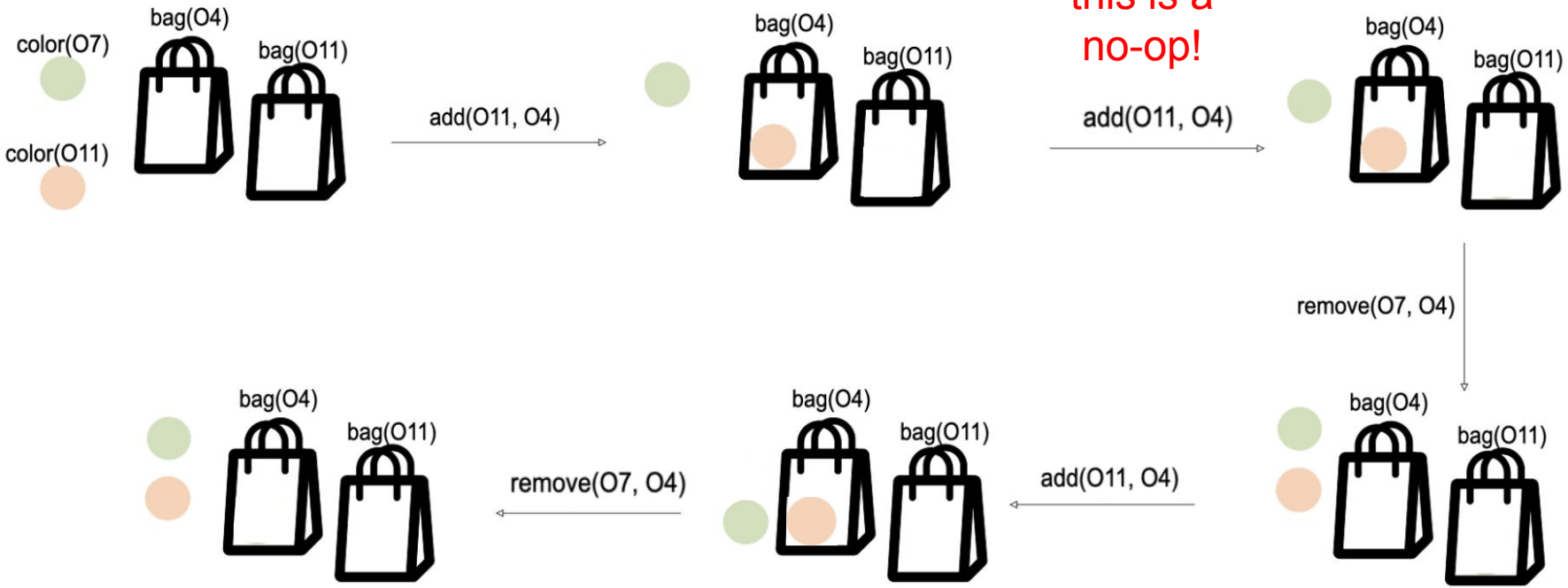

color(O7)
color(O11)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)


color(O7)
color(O11)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)
this is a
no-op!
add(O11, O4)
bag(O4)
bag(O11)

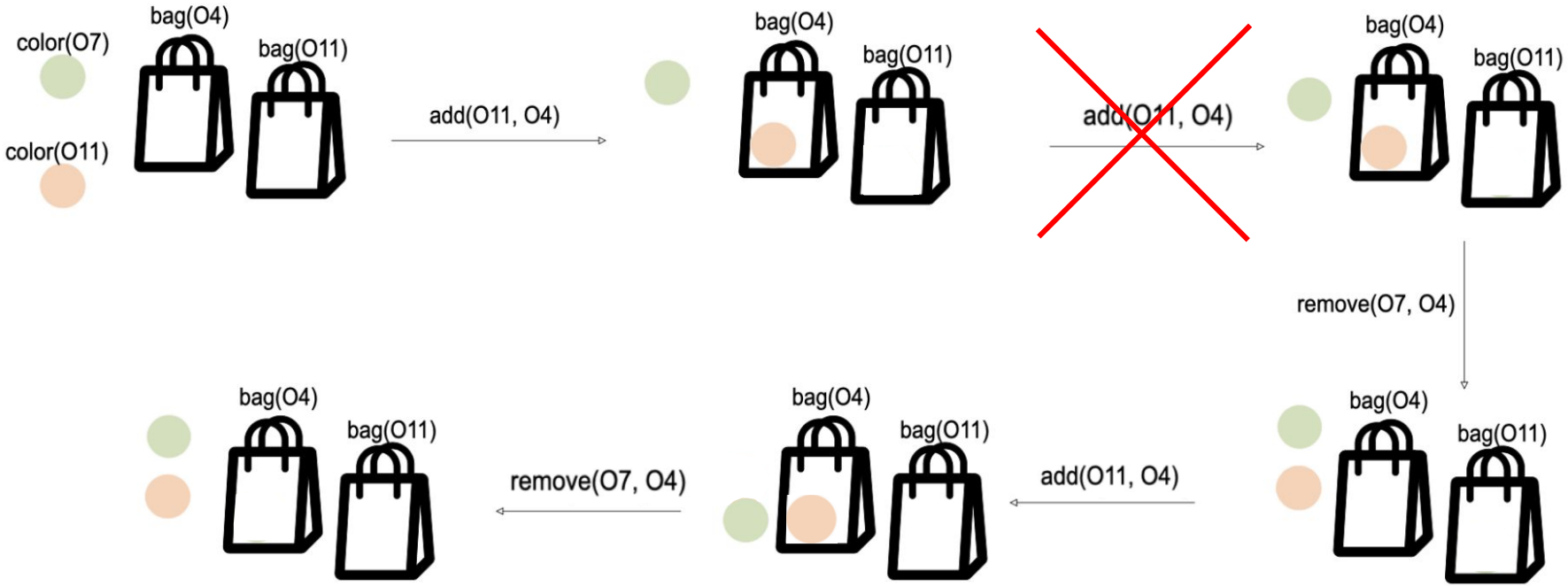

color(O7)
color(O11)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)
this is a
no-op!
add(O11, O4)
bag(O4)
bag(O11)
remove(O7, O4)
bag(O4)
bag(O11)


color(O7)
color(O11)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)
this is a
no-op!
add(O11, O4)
bag(O4)
bag(O11)
remove(O7, O4)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)

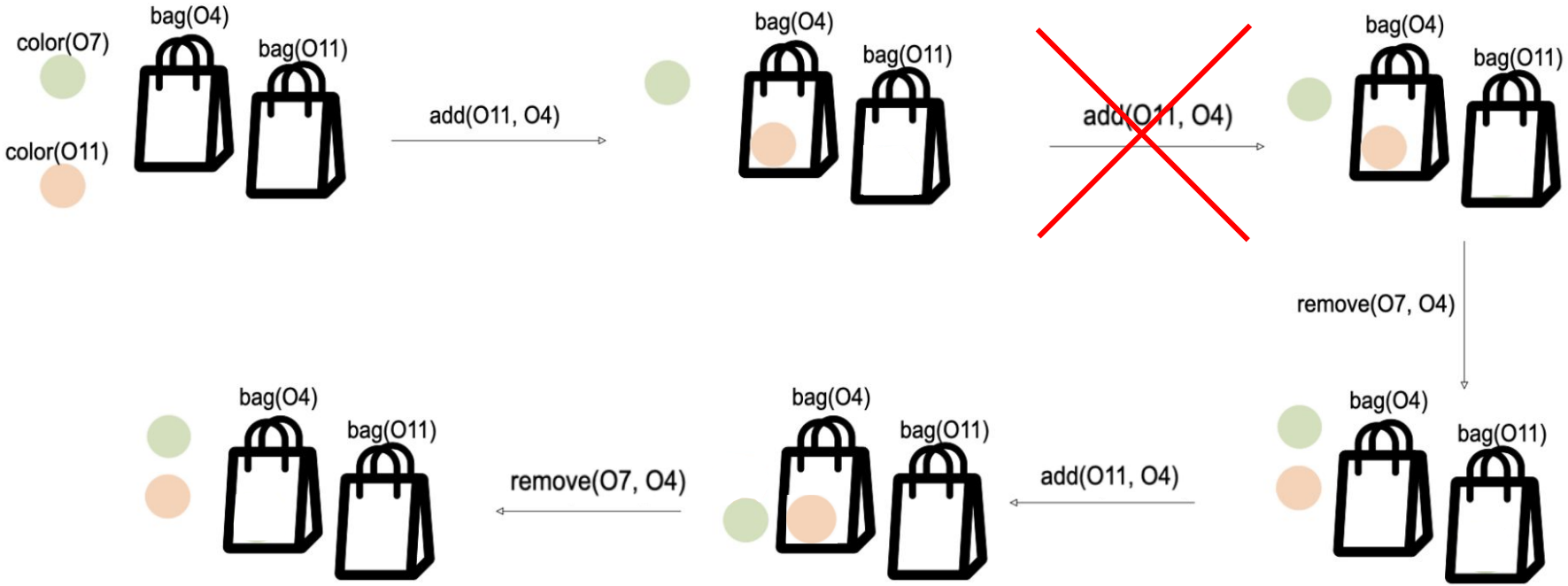

color(O7)
color(O11)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)
this is a
no-op!
add(O11, O4)
bag(O4)
bag(O11)
remove(O7, O4)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)
remove(O7, O4)
bag(O4)
bag(O11)



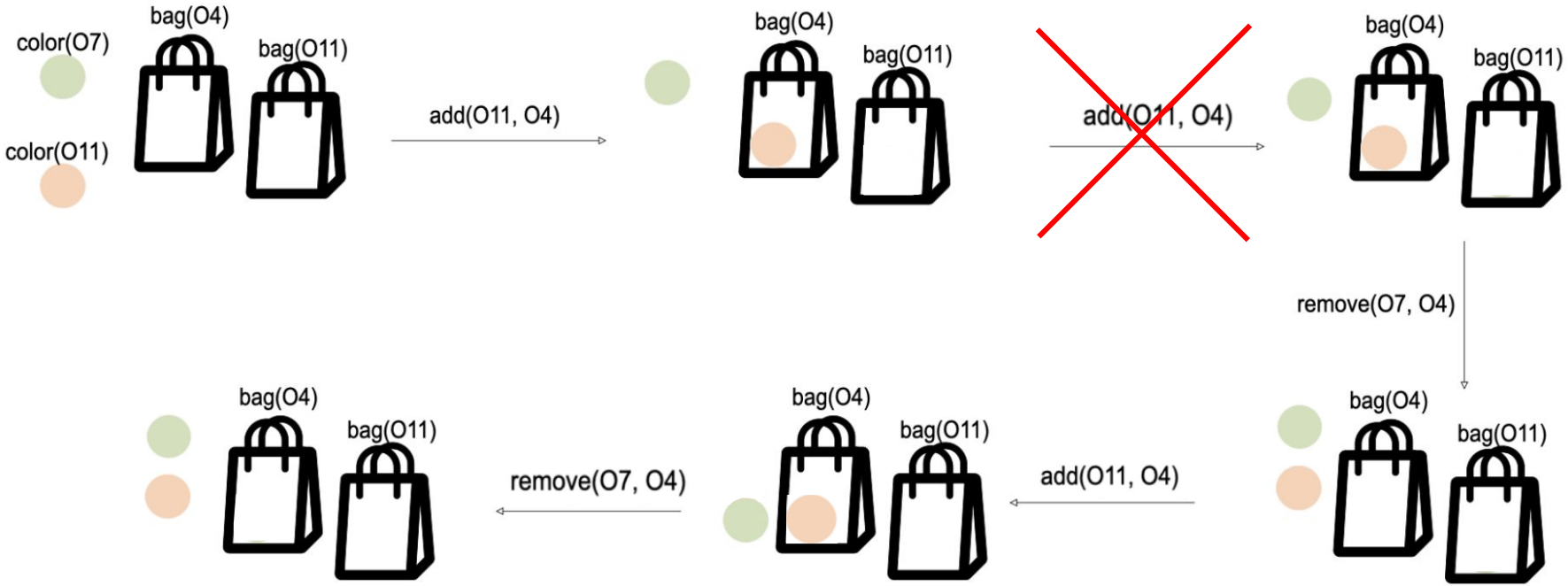
Tracking the state amounts to recognizing the language {a,b,e}* b e* ∉ C-RASP.


color(O7)
color(O11)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)
remove(O7, O4)
bag(O4)
bag(O11)
add(O11, O4)
bag(O4)
bag(O11)
remove(O7, O4)
bag(O4)
bag(O11)

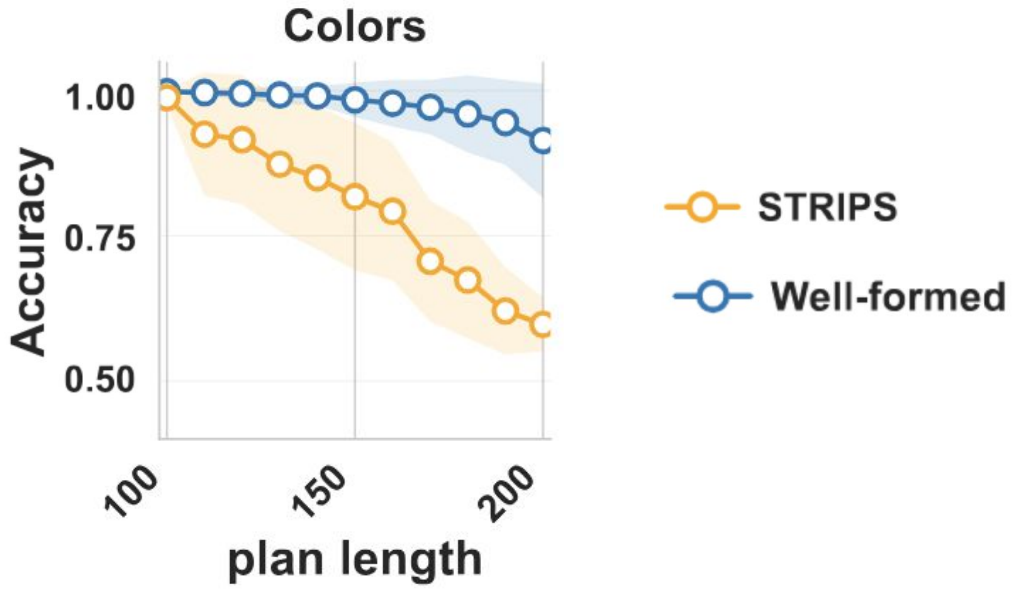
Well-formed STRIPS: propositions listed in the effects strictly change value when the action is applied



Well-formed STRIPS: propositions listed in the effects strictly change value when the action is applied



Tracking the state amounts to recognizing the language (ab)* ∈ C-RASP.


Colors
1.00
0.75
0.50
Accuracy
100
150
200
plan length
STRIPS
Well-formed

Theorem (Informal)

For verifying plans for STRIPS domains that are

Theorem (Informal)

For verifying plans for STRIPS domains that are

- delete-free

Theorem (Informal)

For verifying plans for STRIPS domains that are

- delete-free, or
- well-formed

Theorem (Informal)

For verifying plans for STRIPS domains that are

- delete-free, or
- well-formed

generalization to longer plans and larger domains is expected.

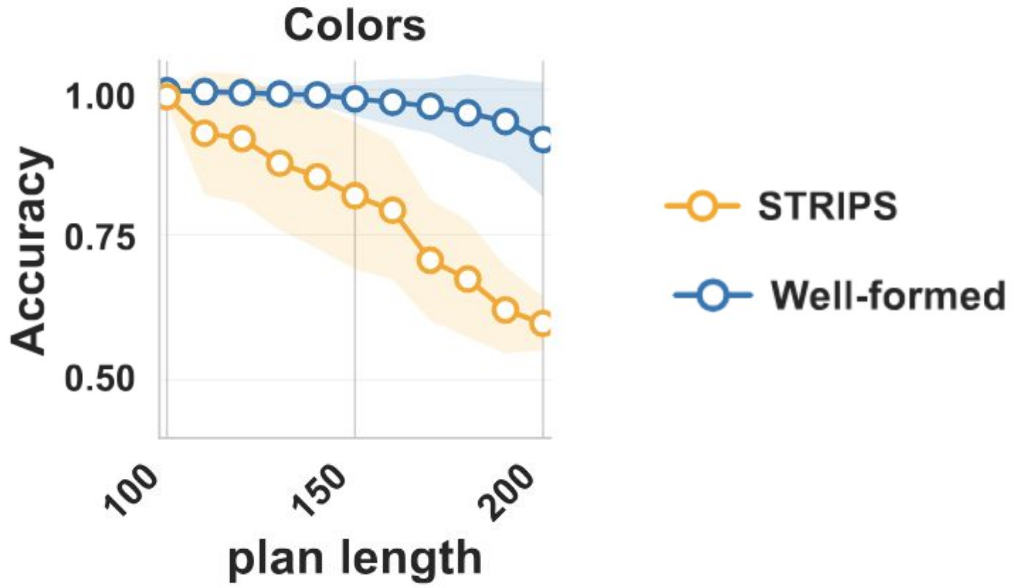
Theorem (Informal)

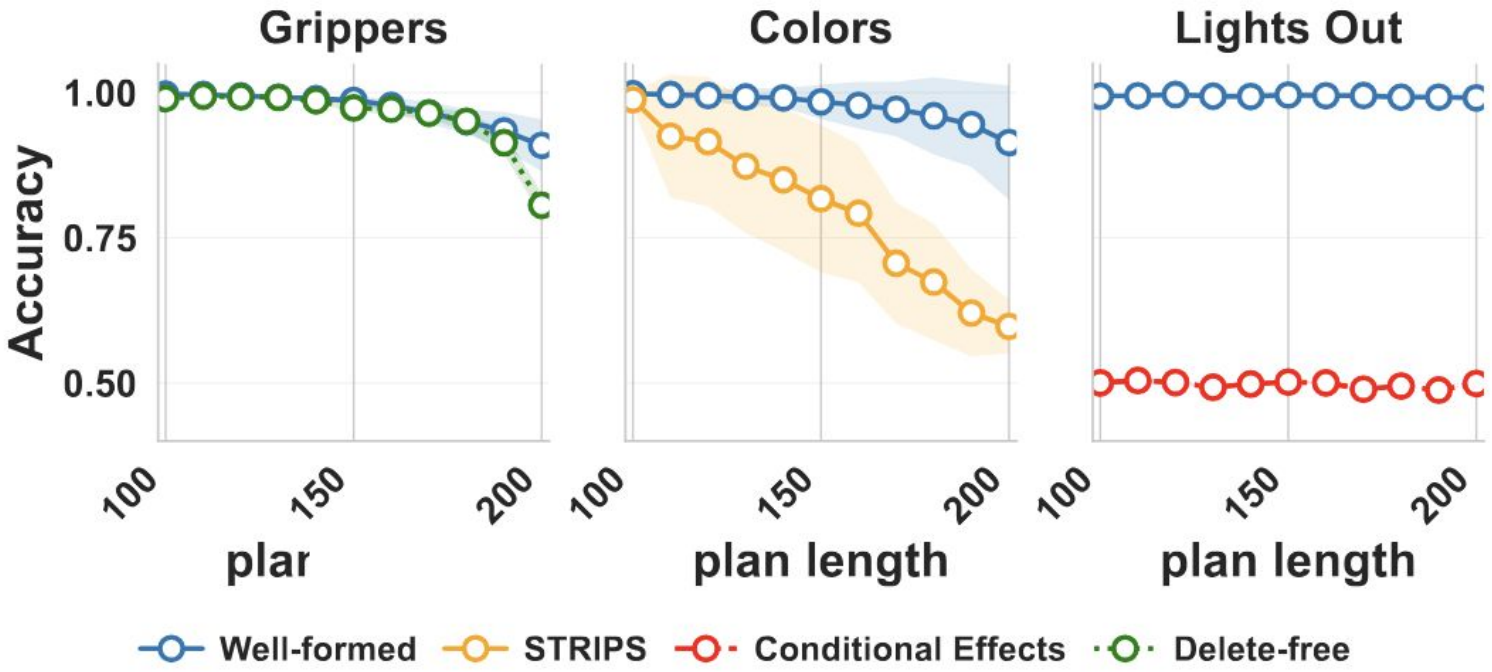
For verifying plans for STRIPS domains that are

- delete-free, or
- well-formed

generalization to longer plans and larger domains is expected.

Outside these classes, in general, no generalization is expected.


Colors
1.00
0.75
0.50
Accuracy
100
150
200
plan length
STRIPS
Well-formed


Grippers
Colors
Lights Out
Accuracy
1.00
0.75
0.50
100
150
200
plar
plan length
plan length
Well-formed
STRIPS
Conditional Effects
Delete-free

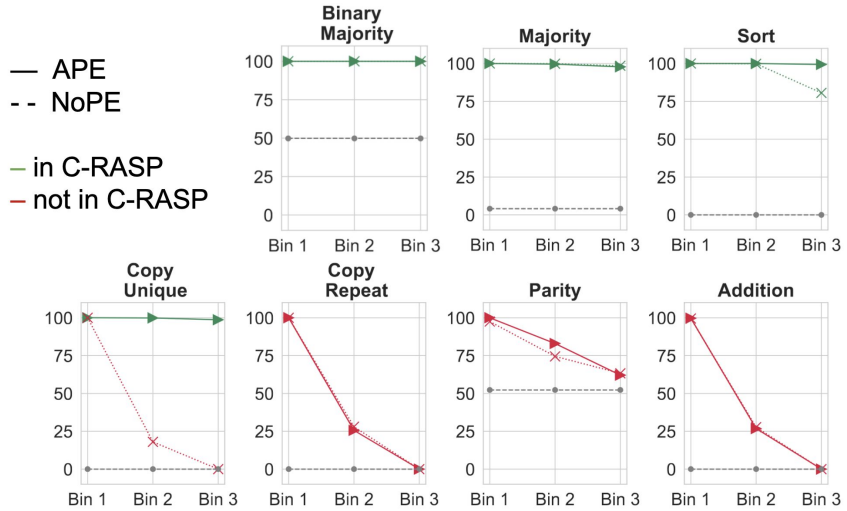
C-RASP Expressivity Predicts Transformer Length Generalization.

UNIQUE COPY

$$CP_a(i) := \#[j \le i, j = i-1]\ Q_a(j) \quad (1)$$

$$CBIGRAM_{ab} := \#[j \le i]\ Q_b(j) \wedge CP_a(j) \ge 1 \quad (2)$$

$$NEXT_a(i) := \bigvee_{\sigma \in \Sigma} [Q_\sigma(i) \wedge CBIGRAM_{\sigma a}(i) \ge 1] \quad (3)$$



Expressivity

Individual forward pass	Chain of Thought
inside TC^0 or AC^0 (depending on precision assumptions)	Turing-complete chain length = O(TM run time)

	Individual forward pass	Chain of Thought
Expressivity	inside TC^0 or AC^0 (depending on precision assumptions)	Turing-complete chain length = O(TM run time)
Learnability		

	Individual forward pass	Chain of Thought
Expressivity	inside TC^0 or AC^0 (depending on precision assumptions)	Turing-complete chain length = O(TM run time)
Learnability	C-RASP	

	Individual forward pass	Chain of Thought
Expressivity	inside TC^0 or AC^0 (depending on precision assumptions)	Turing-complete chain length = O(TM run time)
Learnability	C-RASP	?

Most constructions have been only theoretical.
Some papers explicitly report difficulty in generalization.

SOFTMAX TRANSFORMERS ARE TURING-COMPLETE

Hongjian Jiang
RPTU Kaiserslautern-Landau
hongjian.jiang@rptu.de

Michael Hahn
Saarland University
mhahn@lst.uni-saarland.de

Georg Zetzsche
MPI-SWS
Kaiserslautern, Germany
georg@mpi-sws.org

Anthony W. Lin
RPTU Kaiserslautern-Landau and MPI-SWS
Kaiserslautern, Germany
awlin@mpi-sws.org



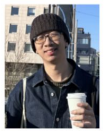

ICLR 2026

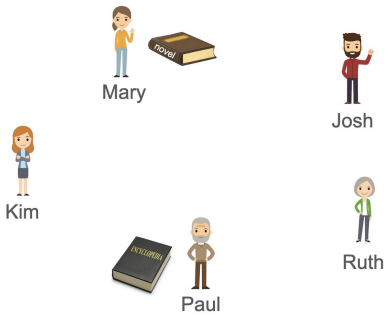
Barriers to Universal Reasoning With Transformers (And How to Overcome Them)

Oliver Kraus[*1], **Yash Sarrof**[*1], **Yuekun Yao**[2], **Alexander Koller**[†1], **Michael Hahn**[†1]
[1]Saarland University [2]Ohio State University

COLM 2026







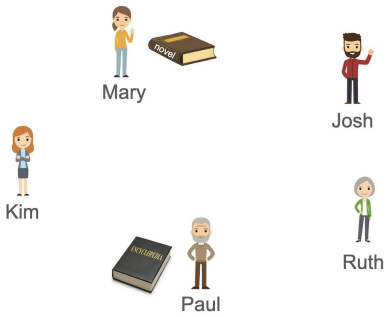
Got it — let's carefully trace the movements step by step:

1. **Initial ownership:**
   - Paul → *encyclopedia*
   - Mary → *novel*
2. **Paul gave his book (encyclopedia) to Josh:**
   - Josh → *encyclopedia*
   - Paul → nothing
3. **Mary gave her book (novel) to Paul:**
   - Paul → *novel*
   - Mary → nothing
4. **Josh gave his book (encyclopedia) to Kim:**
   - Kim → *encyclopedia*
   - Josh → nothing
5. **Paul gave his book (novel) to Ruth:**
   - Ruth → *novel*
   - Paul → nothing

✅ **Final ownership:**

- Kim → *encyclopedia*
- Ruth → *novel*
- Paul, Mary, Josh → nothing

Answer: Kim has the encyclopedia.



Got it — let's carefully trace the movements step by step:

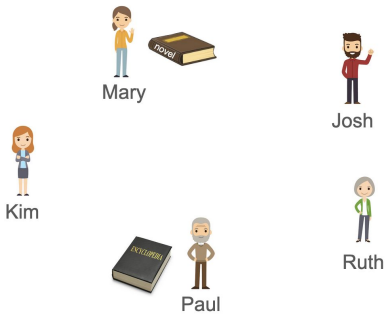
1. **Initial ownership:**
   - Paul → *encyclopedia*
   - Mary → *novel*
2. **Paul gave his book (encyclopedia) to Josh:**
   - Josh → *encyclopedia*
   - Paul → nothing
3. **Mary gave her book (novel) to Paul:**
   - Paul → *novel*
   - Mary → nothing
4. **Josh gave his book (encyclopedia) to Kim:**
   - Kim → *encyclopedia*
   - Josh → nothing
5. **Paul gave his book (novel) to Ruth:**
   - Ruth → *novel*
   - Paul → nothing

✅ **Final ownership:**

- Kim → *encyclopedia*
- Ruth → *novel*
- Paul, Mary, Josh → nothing

Answer: Kim has the encyclopedia.

How learnable is this?



Got it — let's carefully trace the movements step by step:

1. **Initial ownership:**
   - Paul → *encyclopedia*
   - Mary → *novel*
2. **Paul gave his book (encyclopedia) to Josh:**
   - Josh → *encyclopedia*
   - Paul → nothing
3. **Mary gave her book (novel) to Paul:**
   - Paul → *novel*
   - Mary → nothing
4. **Josh gave his book (encyclopedia) to Kim:**
   - Kim → *encyclopedia*
   - Josh → nothing
5. **Paul gave his book (novel) to Ruth:**
   - Ruth → *novel*
   - Paul → nothing

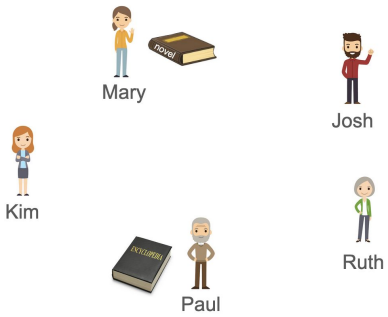
✅ **Final ownership:**

- Kim → *encyclopedia*
- Ruth → *novel*
- Paul, Mary, Josh → nothing

Answer: Kim has the encyclopedia.

How learnable is this?

requires copying the input!



Got it — let's carefully trace the movements step by step:

1. **Initial ownership:**
   - Paul → *encyclopedia*
   - Mary → *novel*
2. **Paul gave his book (encyclopedia) to Josh:**
   - Josh → *encyclopedia*
   - Paul → nothing
3. **Mary gave her book (novel) to Paul:**
   - Paul → *novel*
   - Mary → nothing
4. **Josh gave his book (encyclopedia) to Kim:**
   - Kim → *encyclopedia*
   - Josh → nothing
5. **Paul gave his book (novel) to Ruth:**
   - Ruth → *novel*
   - Paul → nothing

✅ **Final ownership:**

- Kim → *encyclopedia*
- Ruth → *novel*
- Paul, Mary, Josh → nothing

Answer: Kim has the encyclopedia.

How learnable is this?

requires copying the input!

Empirically, difficulty in length generalization is often reported.

Zhou et al ICLR 2024
Ebrahimi et al, ICML 2026

Theorem (informal):

Theorem (informal):

Let $L \subseteq \Sigma^*$.

Theorem (informal):

Let $L \subseteq \Sigma^*$. If L has a CoT in C-RASP

Theorem (informal):

Let $L \subseteq \Sigma^*$. If L has a CoT in C-RASP with

- fixed finite vocabulary Σ

Theorem (informal):

Let $L \subseteq \Sigma^*$. If L has a CoT in C-RASP with

- fixed finite vocabulary Σ
- standard positional encodings

Theorem (informal):

Let $L \subseteq \Sigma^*$. If L has a CoT in C-RASP with

- fixed finite vocabulary Σ
- standard positional encodings

then L is definable in TC^0.

Theorem (informal):

Let $L \subseteq \Sigma^*$. If L has a CoT in C-RASP with

- fixed finite vocabulary Σ
- standard positional encodings

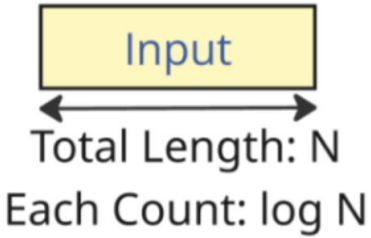
then L is definable in TC^0.

constant-depth threshold circuits

believed to be far below P

C-RASP program computes "C" counts.

C-RASP program computes "C" counts.



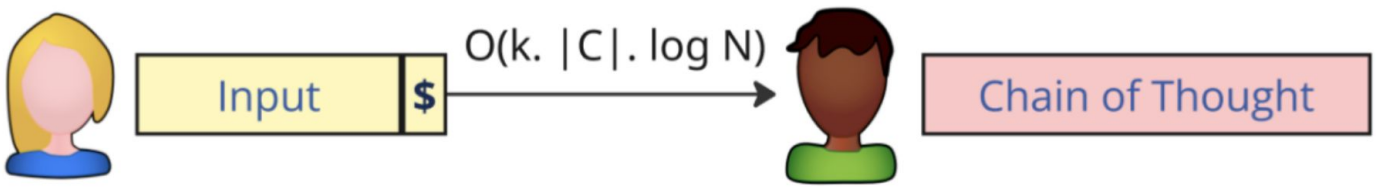
C-RASP program computes "C" counts.

C-RASP program computes "C" counts.

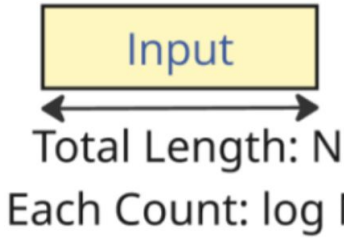
Input

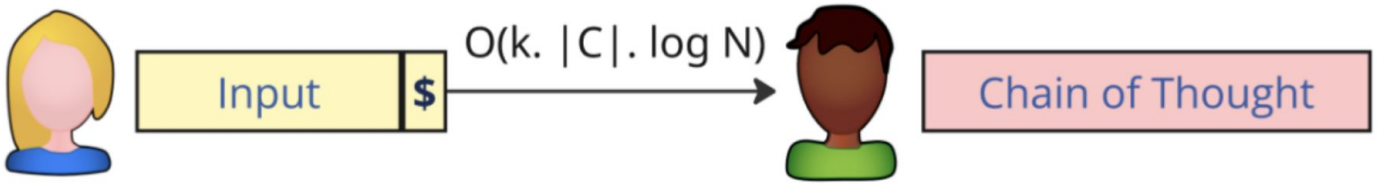
Total Length: N

Each Count: log N



C-RASP program computes "C" counts.





Total possible strings is **polynomial in N.**
We can hardcode a lookup table in TC0.

Theorem (informal):

Let $L \subseteq \Sigma^*$. If L has a CoT in C-RASP with

- fixed finite vocabulary Σ
- standard positional encodings

then L is definable in TC^0.

constant-depth threshold circuits

believed to be far below P

Empirically, difficulty in length generalization is often reported.

Zhou et al ICLR 2024
Ebrahimi et al, ICML 2026

Theorem (informal):

Let $L \subseteq \Sigma^*$. If L has a CoT in C-RASP with

- fixed finite vocabulary Σ
- standard positional encodings

then L is definable in TC^0.

constant-depth threshold circuits

believed to be far below P

Empirically, difficulty in length generalization is often reported.

Zhou et al ICLR 2024
Ebrahimi et al, ICML 2026

Theorem (informal):

Let $L \subseteq \Sigma^*$. If L has a CoT in C-RASP with

- fixed finite vocabulary Σ
- standard positional encodings

then L is definable in TC^0.

Theorem (informal):

Let $L \subseteq \Sigma^*$. If L has a CoT in C-RASP with

- fixed finite vocabulary Σ
- standard positional encodings

then L is definable in TC^0.

Jiang et al, ICLR 2026

Theorem (informal):

Let $L \subseteq \Sigma^*$. If L has a CoT in C-RASP with

- fixed finite vocabulary Σ
- standard positional encodings

then L is definable in TC^0.

Kraus et al, COLM 2026

Jiang et al, ICLR 2026

Theorem (informal):

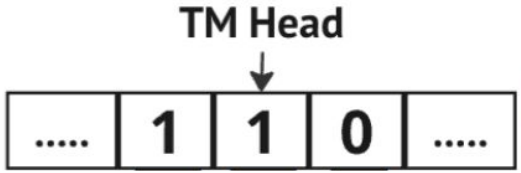
Let $L \subseteq \Sigma^*$. If L has a CoT in C-RASP with

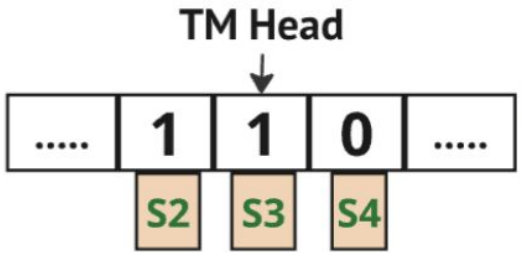
- fixed finite vocabulary Σ
- standard positional encodings

then L is definable in TC^0.

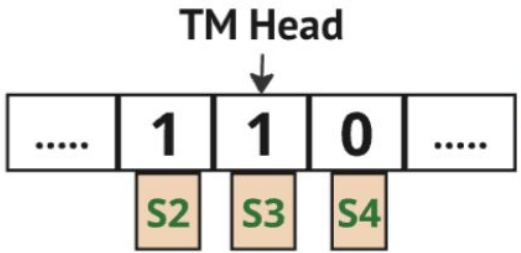
Kraus et al, COLM 2026

Jiang et al, ICLR 2026


TM Head
.....
1
1
0
.....

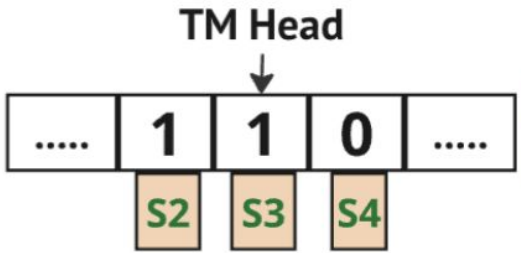


Every cell gets
a unique address



Every cell gets
a unique address

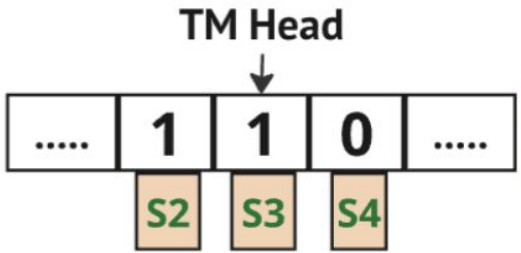
CoT block format



Every cell gets
a unique address

1. State

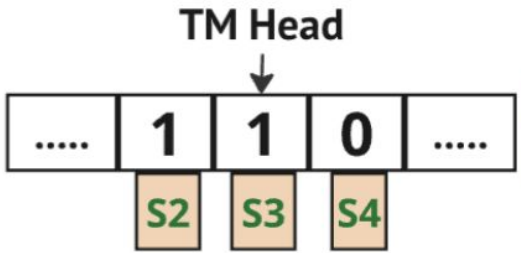
CoT block format



Every cell gets
a unique address

1. State
2. Head position

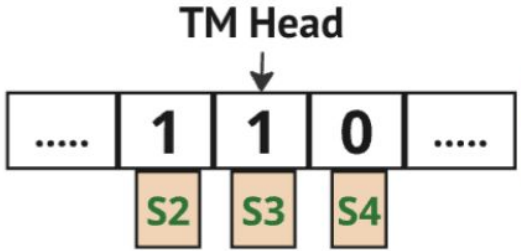
CoT block format



Every cell gets
a unique address

1. State
2. Head position
3. Separator

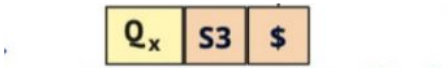
CoT block format



Every cell gets
a unique address

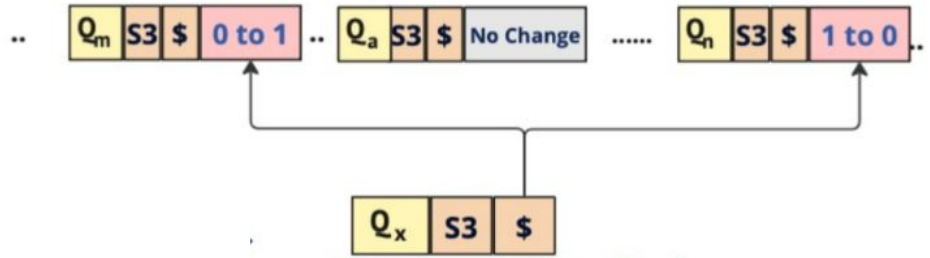
CoT block format

1. State
2. Head position
3. Separator
4. Latest write action
at the current head



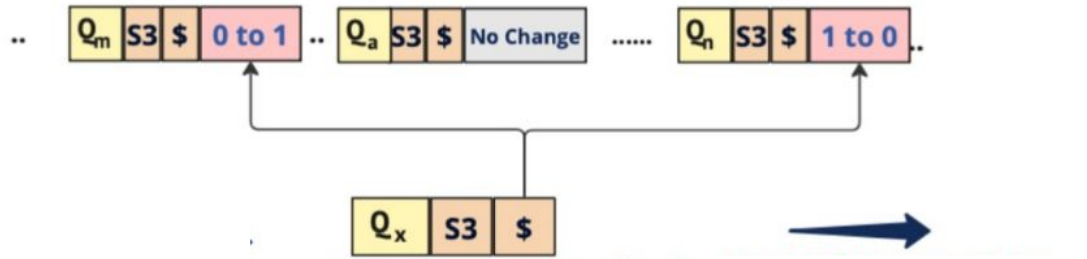
(b.2) Compute Current Cell Value

CoT Trace: Attend to only relevant recorded changes in values!



(b.2) Compute Current Cell Value

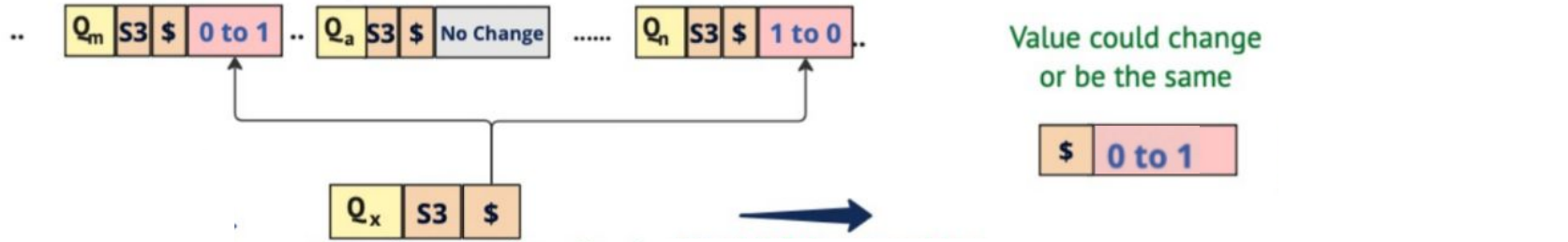
CoT Trace: Attend to only relevant recorded changes in values!



(b.2) Compute Current Cell Value

(b.3) Compute Transition via MLPs

CoT Trace: Attend to only relevant recorded changes in values!

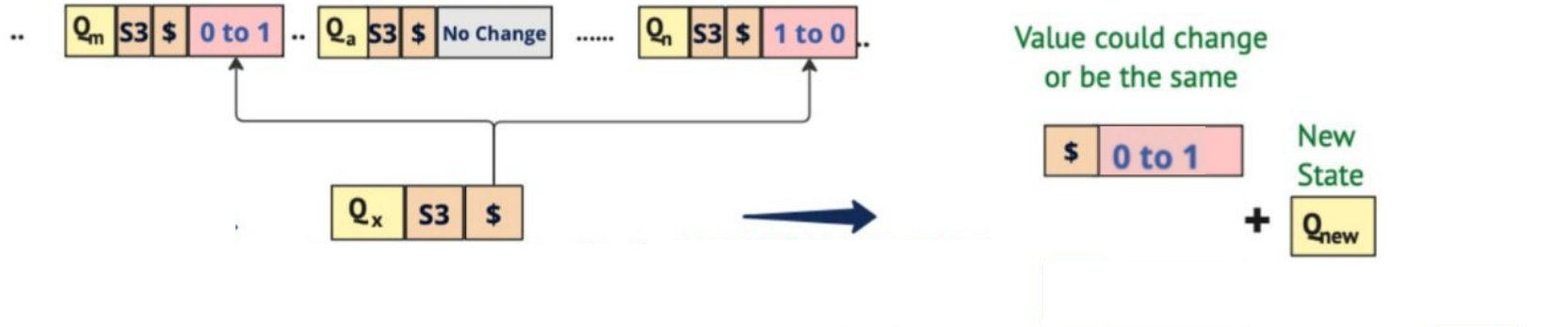


(b.2) Compute Current Cell Value

(b.3) Compute Transition via MLPs

(b.4) Write the new block new (value, state, head position)

CoT Trace: Attend to only relevant recorded changes in values!

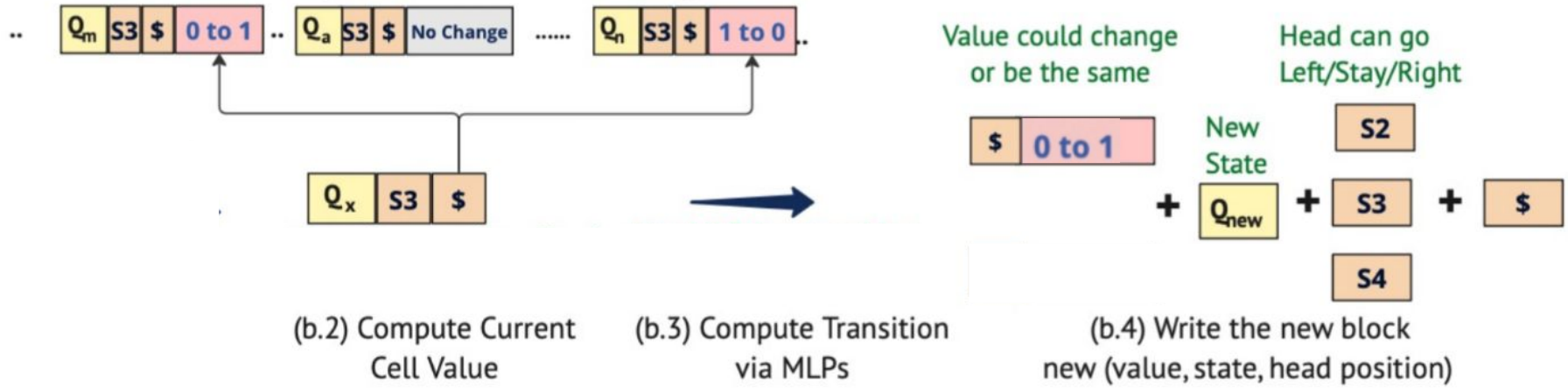


(b.2) Compute Current Cell Value

(b.3) Compute Transition via MLPs

(b.4) Write the new block new (value, state, head position)

CoT Trace: Attend to only relevant recorded changes in values!

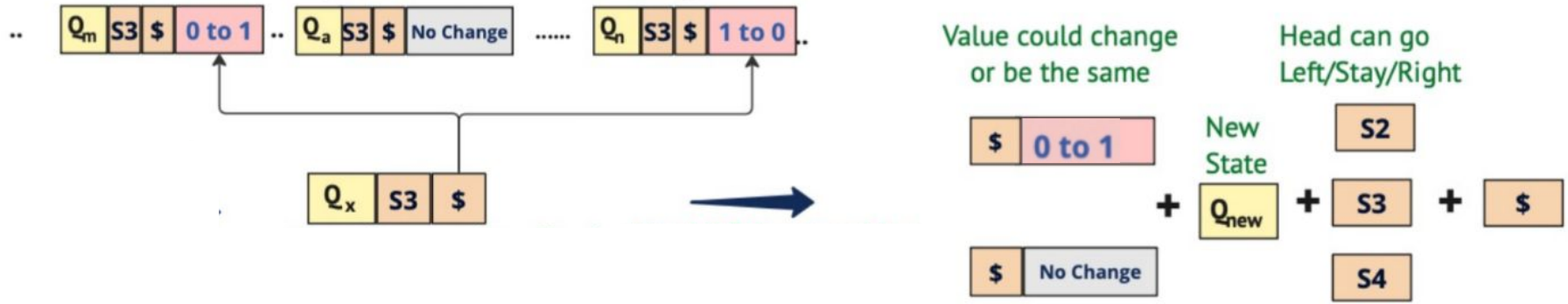


(b.2) Compute Current Cell Value

(b.3) Compute Transition via MLPs

(b.4) Write the new block new (value, state, head position)

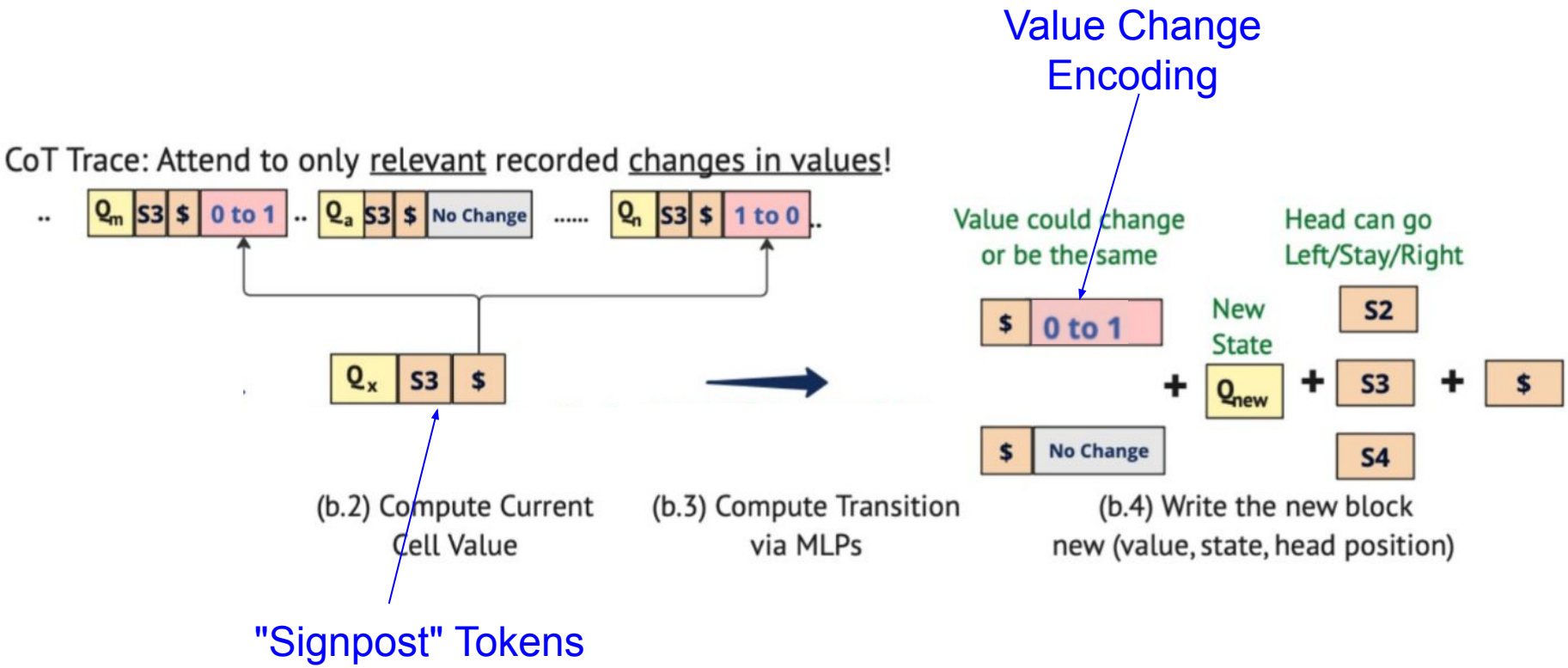
CoT Trace: Attend to only relevant recorded changes in values!



(b.2) Compute Current Cell Value

(b.3) Compute Transition via MLPs

(b.4) Write the new block new (value, state, head position)


Value Change Encoding
CoT Trace: Attend to only relevant recorded changes in values!
Q_m S3 $ 0 to 1
Q_a S3 $ No Change
Q_n S3 $ 1 to 0
Q_x S3 $
Value could change or be the same
Head can go Left/Stay/Right
$ 0 to 1
$ No Change
New State
Q_{new}
S2
S3
S4
$
(b.2) Compute Current Cell Value
(b.3) Compute Transition via MLPs
(b.4) Write the new block new (value, state, head position)
"Signpost" Tokens

Theorem (informal):

Let T be a Turing machine.

Theorem (informal):

Let T be a Turing machine.

There is a C-RASP CoT

Theorem (informal):

Let T be a Turing machine.
There is a C-RASP CoT operating over an infinite alphabet,

Theorem (informal):

Let T be a Turing machine.
There is a C-RASP CoT operating over an infinite alphabet, simulating T,

Theorem (informal):

Let T be a Turing machine.
There is a C-RASP CoT operating over an infinite alphabet, simulating T, when the input is interleaved with index tokens.

S5 State Tracking

S5 State Tracking

A Book B Cat C Hat D Dog E Apple





S5 State Tracking

A Book B Cat C Hat D Dog E Apple

Actions:

swap E A .

swap A B .

swap E C .

S5 State Tracking

A Book B Cat C Hat D Dog E Apple





Actions:

swap E A .

swap A B .

swap E C .

What is in A?

S5 State Tracking

A Book B Cat C Hat D Dog E Apple

Actions:

swap E A .

swap A B .

swap E C .

What is in A?

NC^1-complete
⇒ likely tricky for Transformers even with CoT

S5 State Tracking

A Book B Cat C Hat D Dog E Apple

Actions:

swap E A .

swap A B .

swap E C .

What is in A?

CoT Trace:

NC^1-complete
⇒ likely tricky for Transformers even with CoT

S5 State Tracking

A Book B Cat C Hat D Dog E Apple

Actions:

swap E A .

swap A B .

swap E C .

What is in A?

CoT Trace:

NC^1-complete
⇒ likely tricky for Transformers even with CoT

S5 State Tracking

A Book B Cat C Hat D Dog E Apple

Actions:

swap E A .

swap A B .

swap E C .

What is in A?

CoT Trace:

NC^1-complete
⇒ likely tricky for Transformers even with CoT

S5 State Tracking

A Book B Cat C Hat D Dog E Apple

Actions:

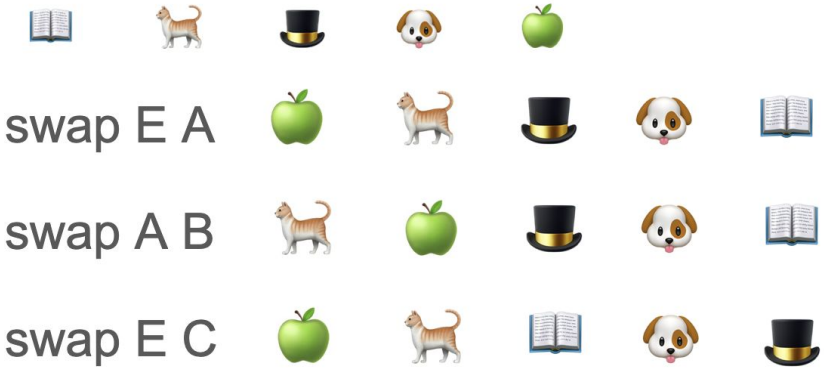
swap E A .

swap A B .

swap E C .

What is in A?

CoT Trace:



NC^1-complete
⇒ likely tricky for Transformers even with CoT

S5 State Tracking

A Book B Cat C Hat D Dog E Apple

Actions:

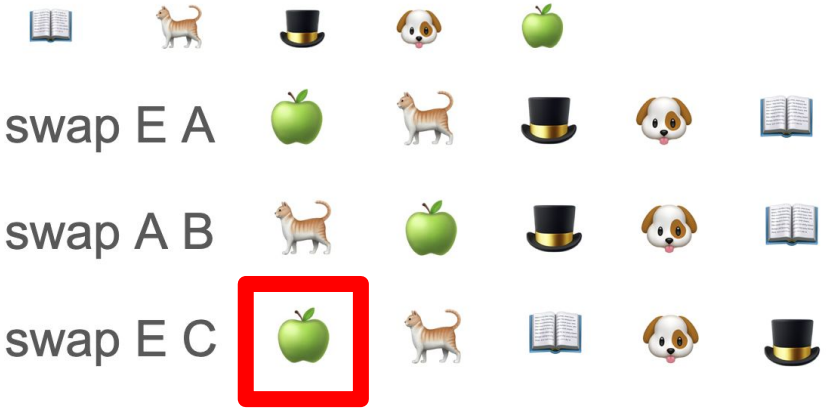
swap E A .

swap A B .

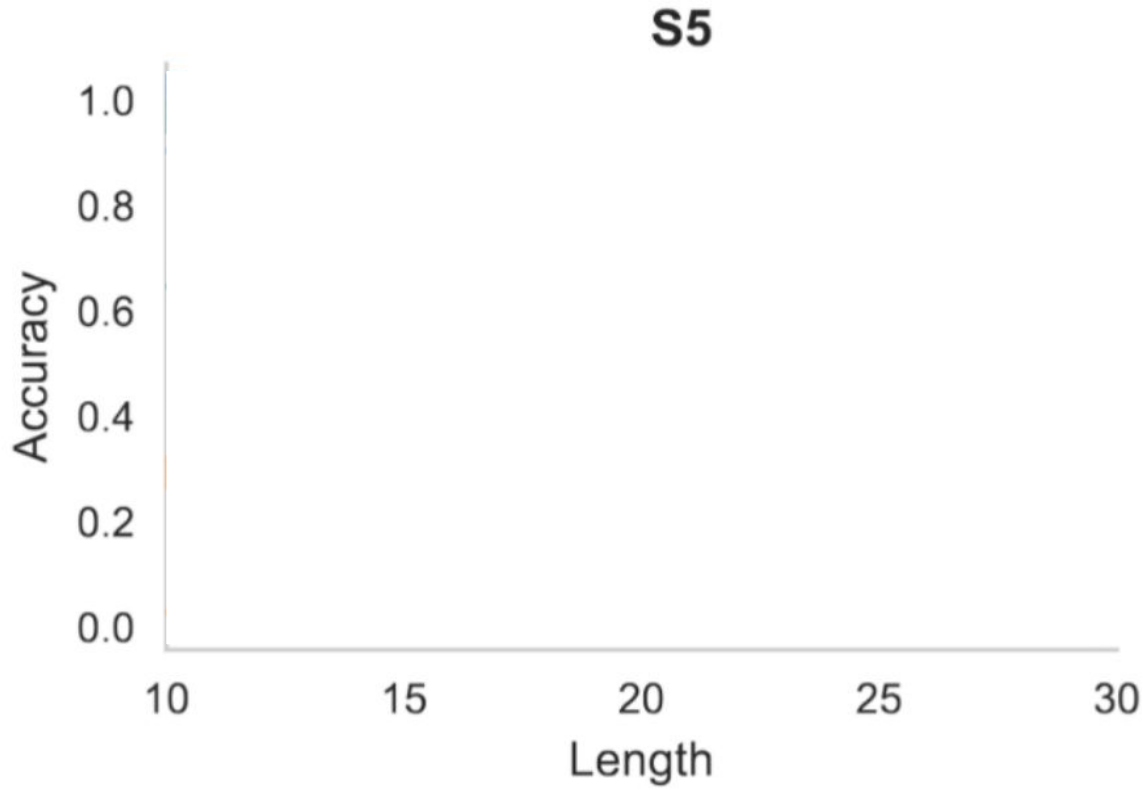
swap E C .

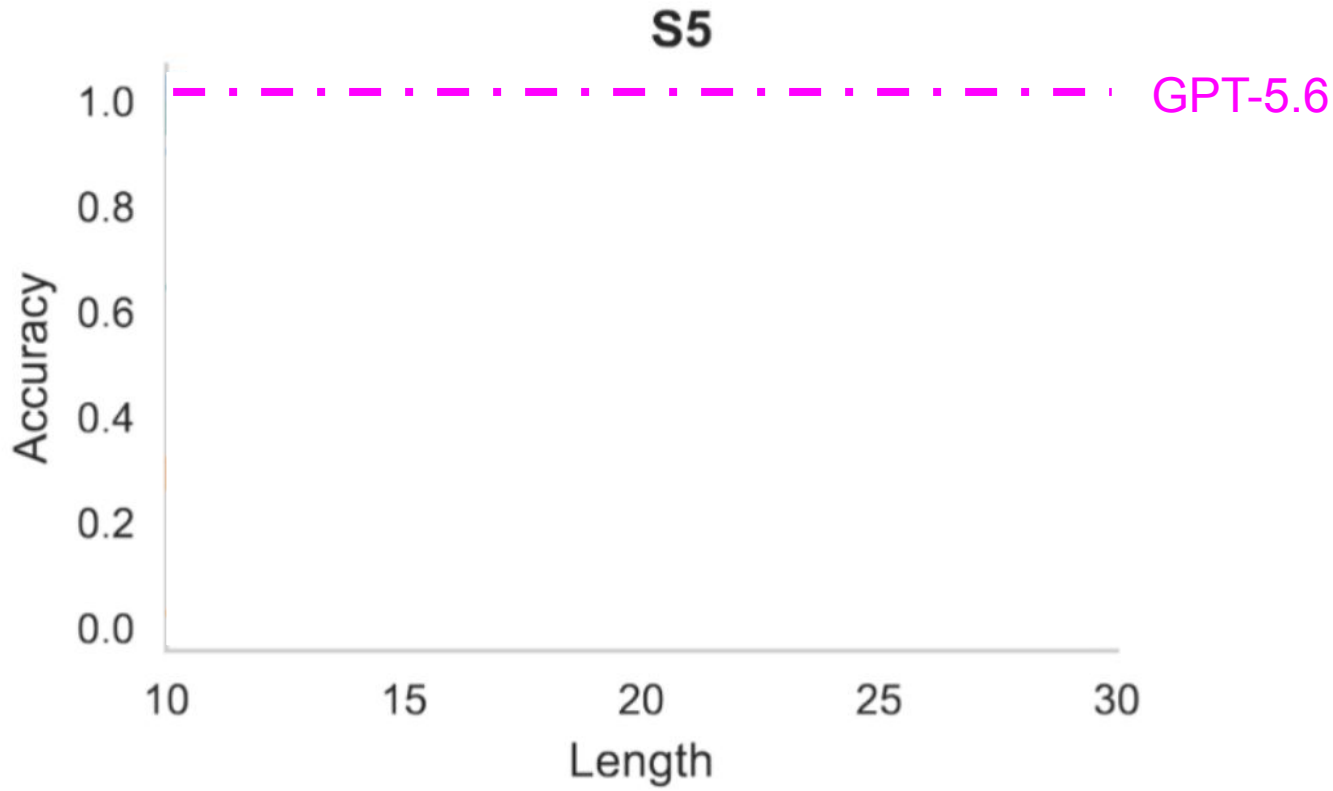
What is in A?

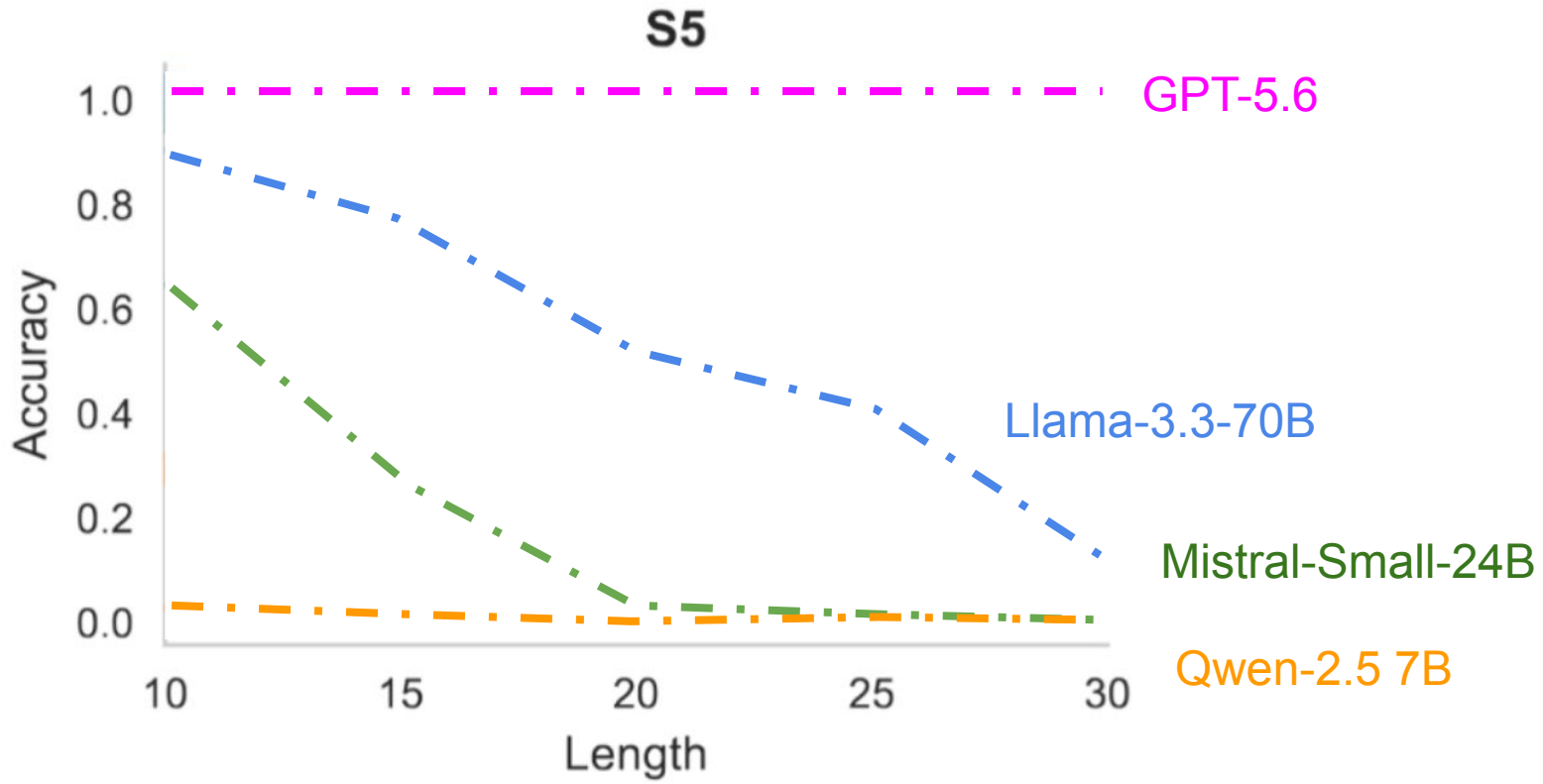
CoT Trace:



NC^1-complete
⇒ likely tricky for Transformers even with CoT


S5
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
10
15
20
25
30
Length


S5
GPT-5.6
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
10
15
20
25
30
Length


S5
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
10
15
20
25
30
Length
GPT-5.6
Llama-3.3-70B
Mistral-Small-24B
Qwen-2.5 7B

S5 State Tracking

A Book B Cat C Hat D Dog E Apple

Actions:

1 swap E A .

2 swap A B .

3 swap E C .

What is in A?

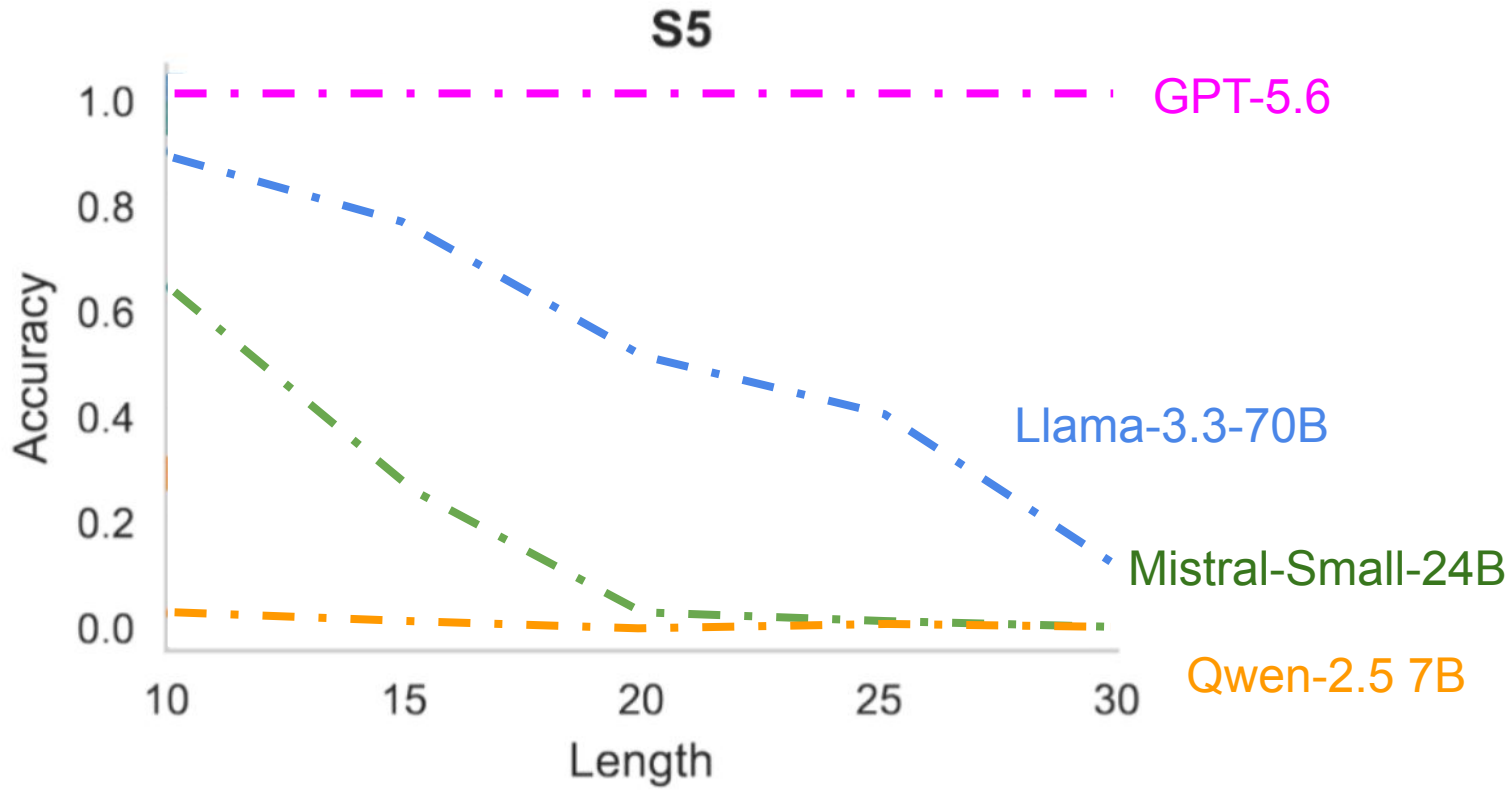
Adapted CoT Trace:

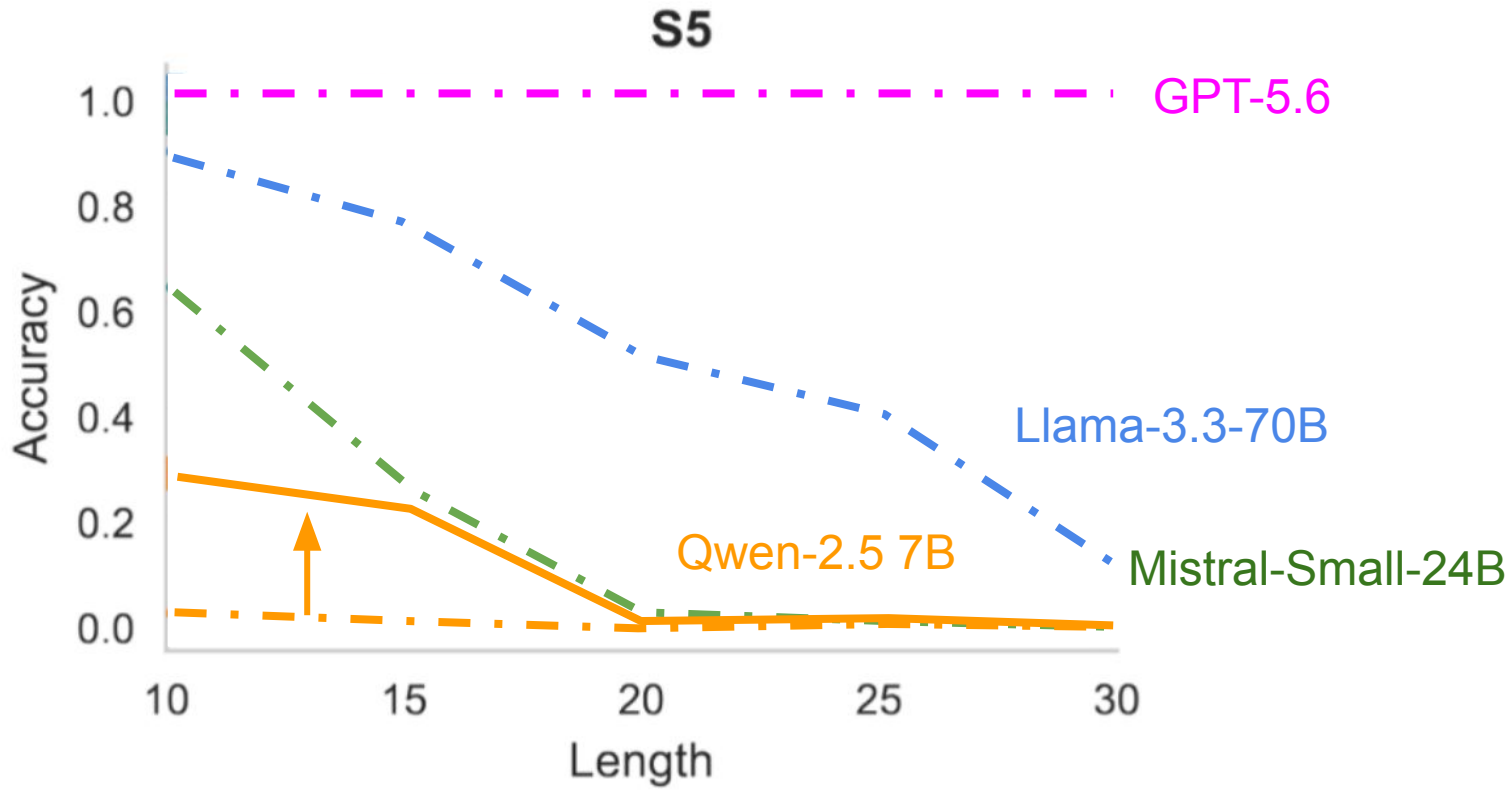
1 swap E A

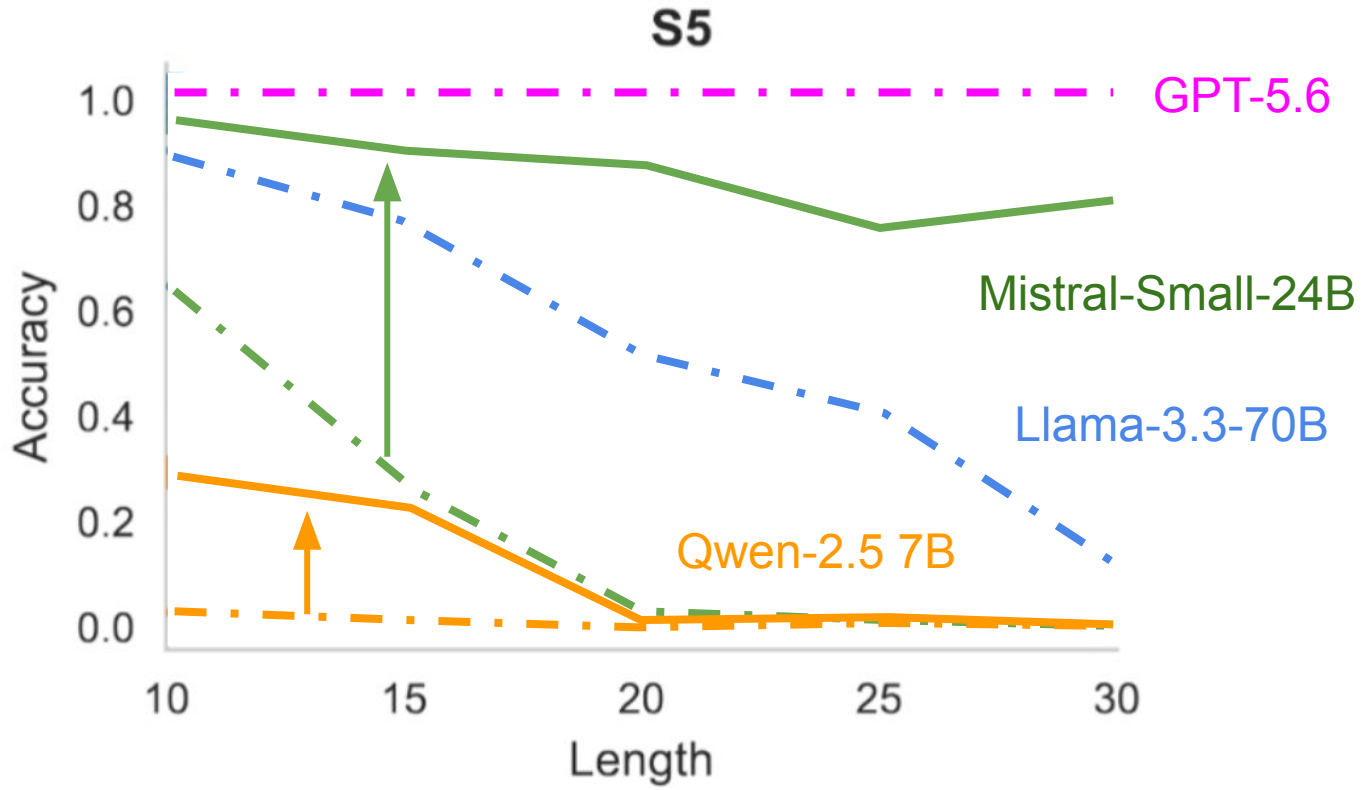
2 swap A B

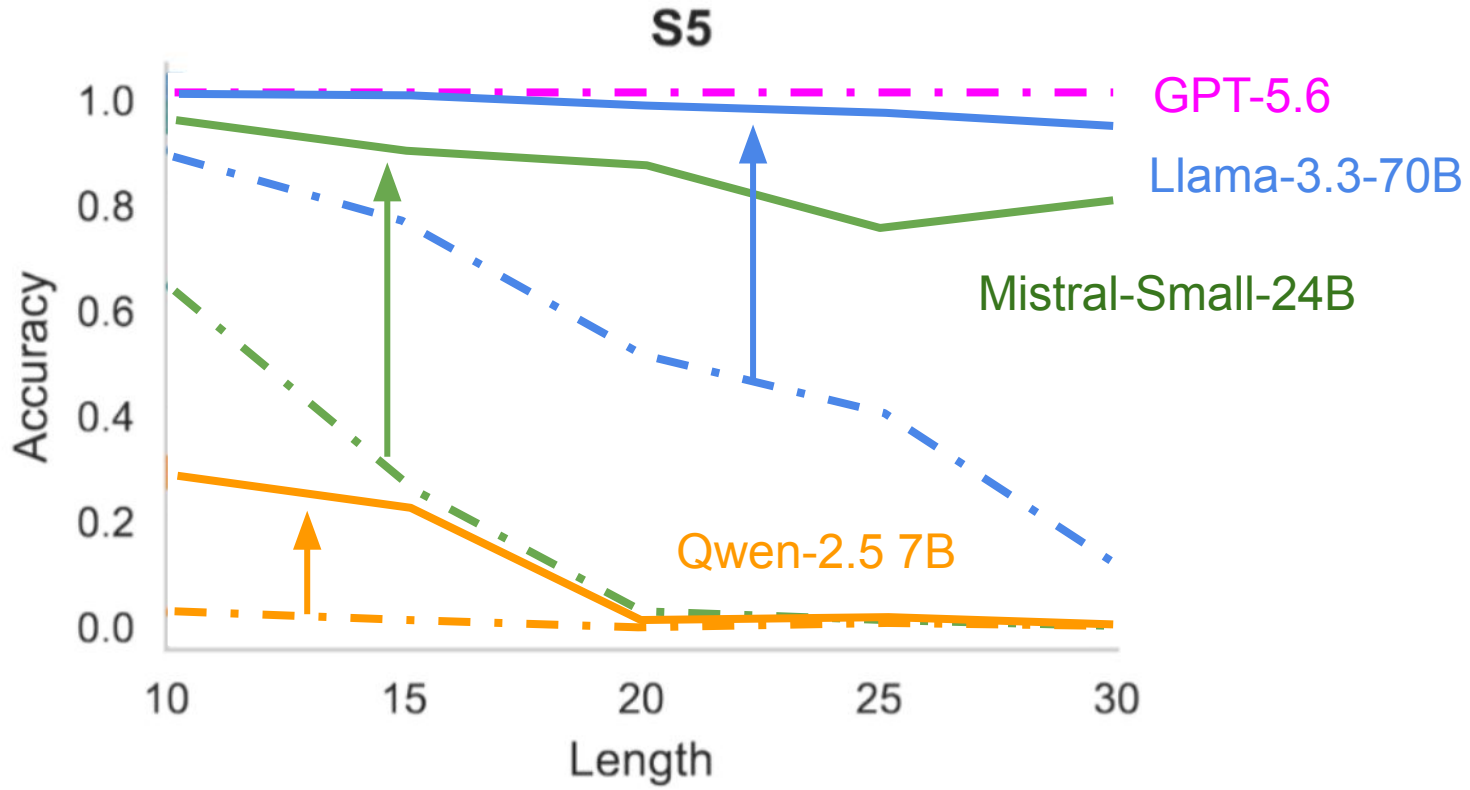
3 swap E C

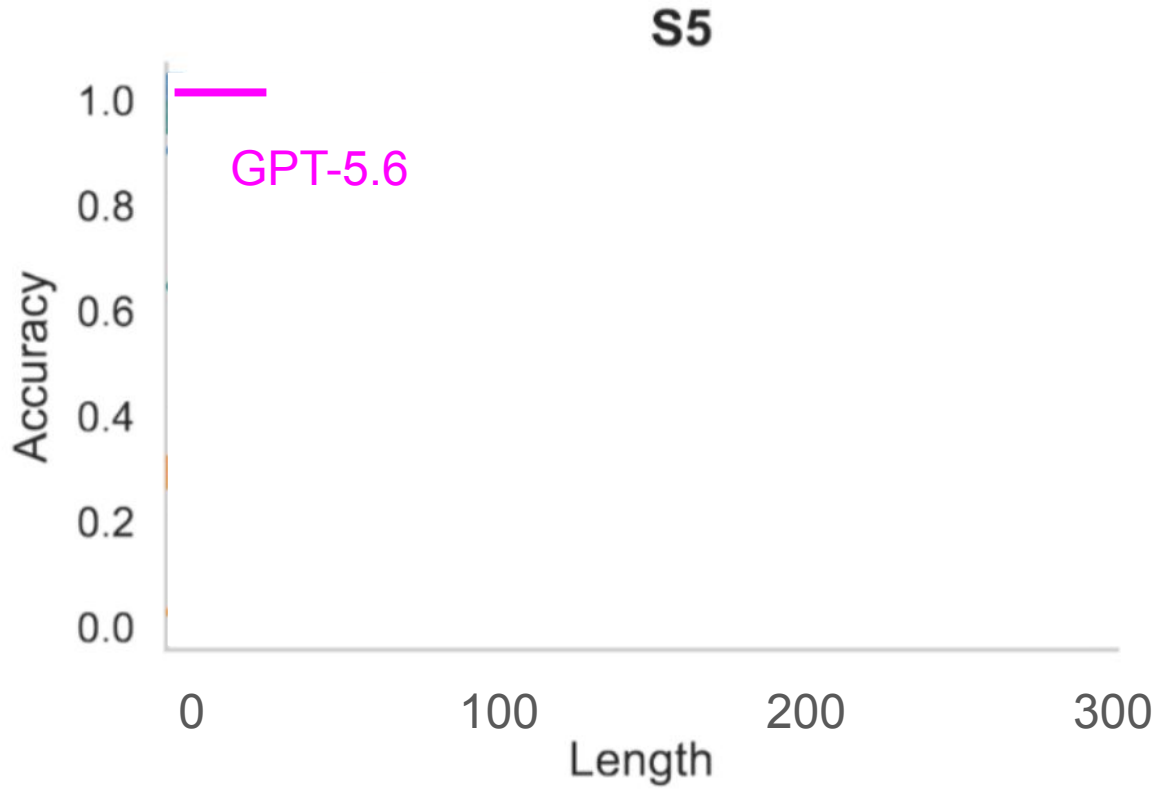
disclaimer: simplified format!

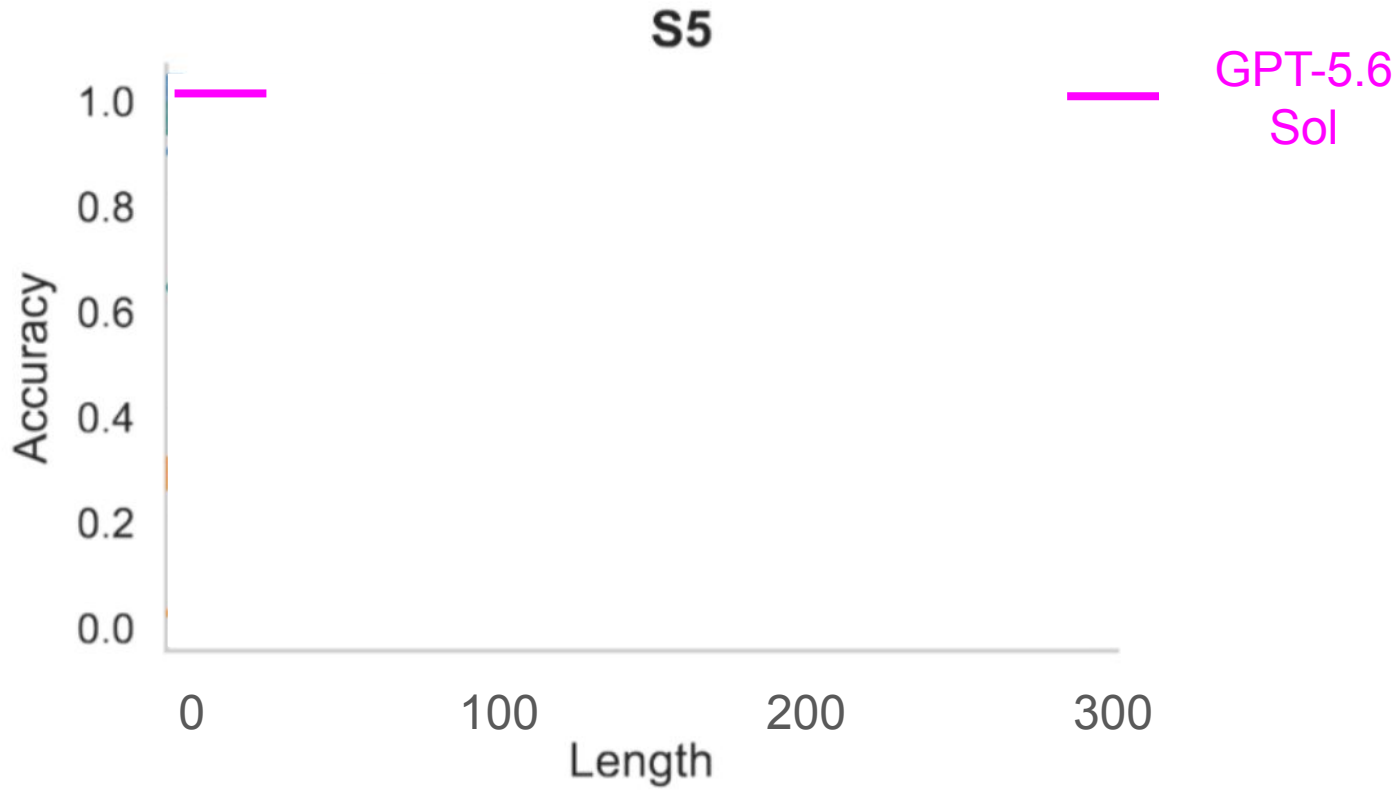

S5
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
10
15
20
25
30
Length
GPT-5.6
Llama-3.3-70B
Mistral-Small-24B
Qwen-2.5 7B

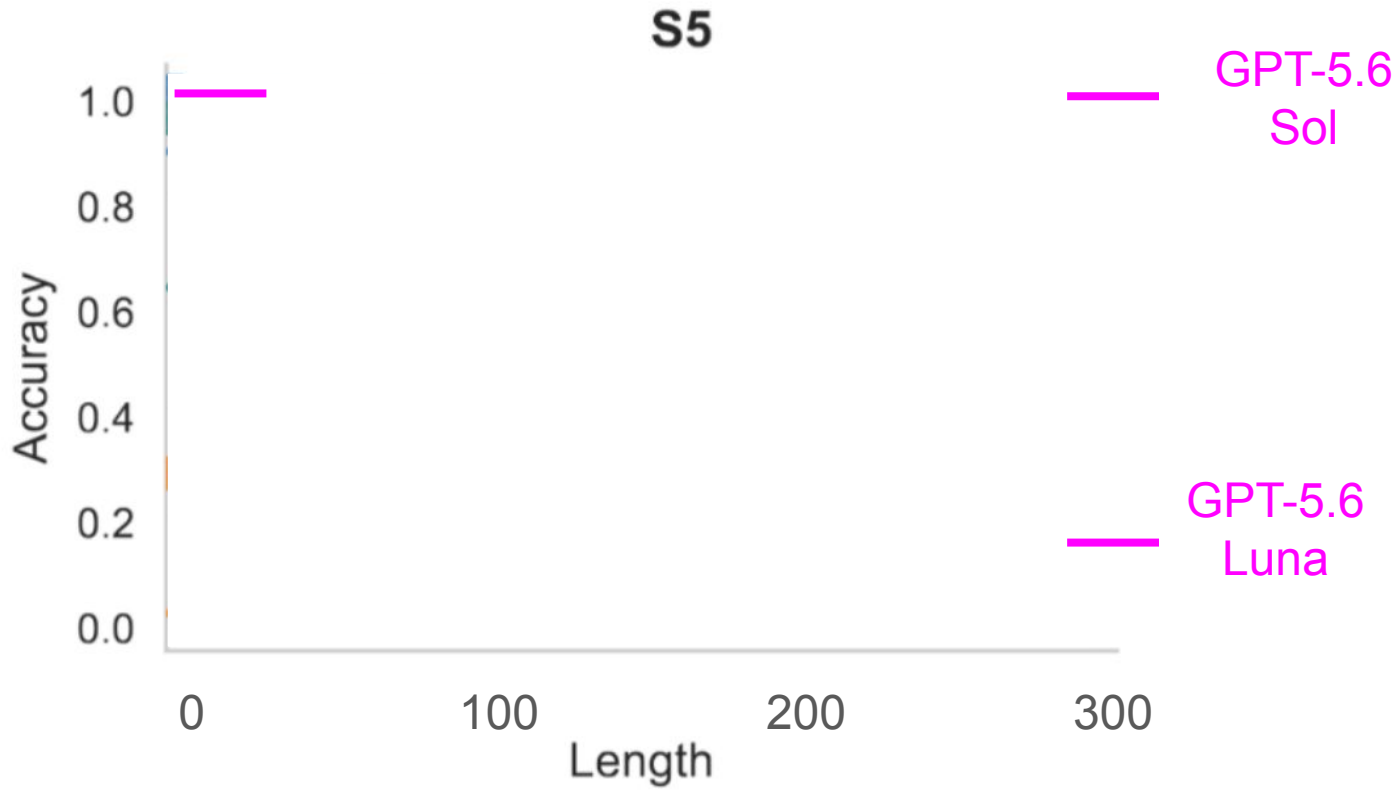

S5
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
10
15
20
25
30
Length
GPT-5.6
Llama-3.3-70B
Mistral-Small-24B
Qwen-2.5 7B


S5
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
10
15
20
25
30
Length
GPT-5.6
Mistral-Small-24B
Llama-3.3-70B
Qwen-2.5 7B


S5
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
10
15
20
25
30
Length
GPT-5.6
Llama-3.3-70B
Mistral-Small-24B
Qwen-2.5 7B


S5
GPT-5.6
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
0
100
200
300
Length


S5
GPT-5.6
Sol
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
0
100
200
300
Length


S5
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
0
100
200
300
Length
GPT-5.6
Sol
GPT-5.6
Luna

Correct

```
swap D A  → A=Cat B=Dog   C=Apple D=Book  E=Hat
swap B E  → A=Cat B=Hat   C=Apple D=Book  E=Dog
swap B C  → A=Cat B=Apple C=Hat   D=Book  E=Dog
swap D E  → A=Cat B=Apple C=Hat   D=Dog   E=Book
swap B C  → A=Cat B=Hat   C=Apple D=Dog   E=Book
swap B A  → A=Hat B=Cat   C=Apple D=Dog   E=Book
swap D C  → A=Hat B=Cat   C=Dog   D=Apple E=Book
swap B C  → A=Hat B=Dog   C=Cat   D=Apple E=Book
swap A E  → A=Book B=Dog C=Cat   D=Apple E=Hat
swap D C  → A=Book B=Dog C=Apple D=Cat   E=Hat
swap B E  → A=Book B=Hat C=Apple D=Cat   E=Dog
swap E B  → A=Book B=Dog C=Apple D=Cat   E=Hat
```

Correct

```
swap D A  → A=Cat B=Dog   C=Apple D=Book  E=Hat
swap B E  → A=Cat B=Hat   C=Apple D=Book  E=Dog
swap B C  → A=Cat B=Apple C=Hat   D=Book  E=Dog
swap D E  → A=Cat B=Apple C=Hat   D=Dog   E=Book
swap B C  → A=Cat B=Hat   C=Apple D=Dog   E=Book
swap B A  → A=Hat B=Cat   C=Apple D=Dog   E=Book
swap D C  → A=Hat B=Cat   C=Dog   D=Apple E=Book
swap B C  → A=Hat B=Dog   C=Cat   D=Apple E=Book
swap A E  → A=Book B=Dog C=Cat   D=Apple E=Hat
swap D C  → A=Book B=Dog C=Apple D=Cat   E=Hat
swap B E  → A=Book B=Hat C=Apple D=Cat   E=Dog
swap E B  → A=Book B=Dog C=Apple D=Cat   E=Hat
```

GPT-5.6 Luna

```
swap D A  → A=Cat B=Dog   C=Apple D=Book  E=Hat
swap B E  → A=Cat B=Hat   C=Apple D=Book  E=Dog
swap B C  → A=Cat B=Apple C=Hat   D=Book  E=Dog
swap D E  → A=Cat B=Apple C=Hat   D=Dog   E=Book
swap B C  → A=Cat B=Hat   C=Apple D=Dog   E=Book
swap B A  → A=Hat B=Cat   C=Apple D=Dog   E=Book
swap D C  → A=Hat B=Cat   C=Dog   D=Apple E=Book
swap B C  → A=Hat B=Dog   C=Cat   D=Apple E=Book
swap B C  → A=Hat  B=Cat C=Dog   D=Apple E=Book
swap A E  → A=Book B=Cat C=Dog   D=Apple E=Hat
swap D C  → A=Book B=Cat C=Apple D=Dog   E=Hat
swap B E  → A=Book B=Hat C=Apple D=Dog   E=Cat
```

Correct

```
swap D A  → A=Cat B=Dog   C=Apple D=Book  E=Hat
swap B E  → A=Cat B=Hat   C=Apple D=Book  E=Dog
swap B C  → A=Cat B=Apple C=Hat   D=Book  E=Dog
swap D E  → A=Cat B=Apple C=Hat   D=Dog   E=Book
swap B C  → A=Cat B=Hat   C=Apple D=Dog   E=Book
swap B A  → A=Hat B=Cat   C=Apple D=Dog   E=Book
swap D C  → A=Hat B=Cat   C=Dog   D=Apple E=Book
swap B C  → A=Hat B=Dog   C=Cat   D=Apple E=Book
swap A E  → A=Book B=Dog C=Cat   D=Apple E=Hat
swap D C  → A=Book B=Dog C=Apple D=Cat   E=Hat
swap B E  → A=Book B=Hat C=Apple D=Cat   E=Dog
swap E B  → A=Book B=Dog C=Apple D=Cat   E=Hat
```

GPT-5.6 Luna

```
swap D A  → A=Cat B=Dog   C=Apple D=Book  E=Hat
swap B E  → A=Cat B=Hat   C=Apple D=Book  E=Dog
swap B C  → A=Cat B=Apple C=Hat   D=Book  E=Dog
swap D E  → A=Cat B=Apple C=Hat   D=Dog   E=Book
swap B C  → A=Cat B=Hat   C=Apple D=Dog   E=Book
swap B A  → A=Hat B=Cat   C=Apple D=Dog   E=Book
swap D C  → A=Hat B=Cat   C=Dog   D=Apple E=Book
swap B C  → A=Hat B=Dog   C=Cat   D=Apple E=Book
swap B C  → A=Hat  B=Cat C=Dog   D=Apple E=Book
swap A E  → A=Book B=Cat C=Dog   D=Apple E=Hat
swap D C  → A=Book B=Cat C=Apple D=Dog   E=Hat
swap B E  → A=Book B=Hat C=Apple D=Dog   E=Cat
```

Correct

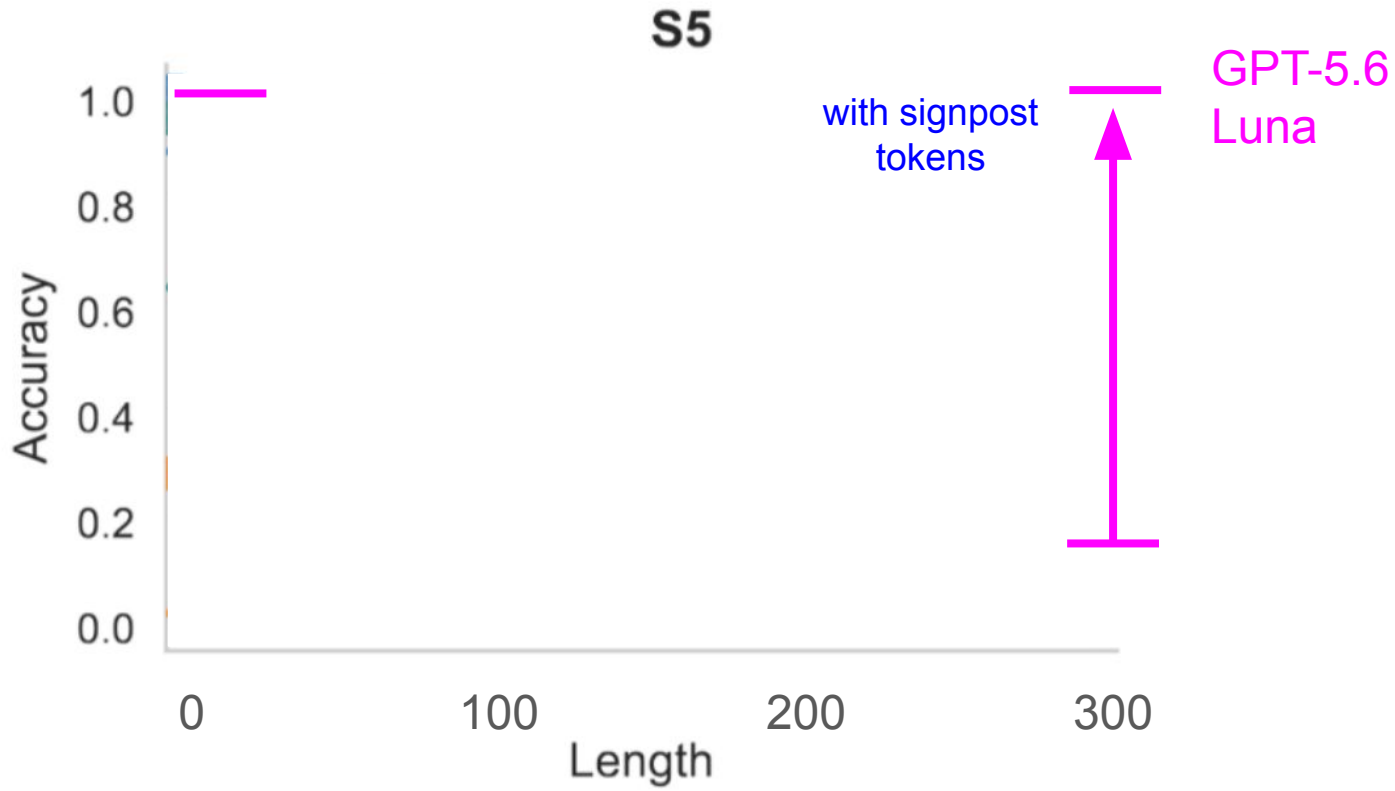
```
swap D A  → A=Cat B=Dog   C=Apple D=Book  E=Hat
swap B E  → A=Cat B=Hat   C=Apple D=Book  E=Dog
swap B C  → A=Cat B=Apple C=Hat   D=Book  E=Dog
swap D E  → A=Cat B=Apple C=Hat   D=Dog   E=Book
swap B C  → A=Cat B=Hat   C=Apple D=Dog   E=Book
swap B A  → A=Hat B=Cat   C=Apple D=Dog   E=Book
swap D C  → A=Hat B=Cat   C=Dog   D=Apple E=Book
swap B C  → A=Hat B=Dog   C=Cat   D=Apple E=Book
swap A E  → A=Book B=Dog C=Cat   D=Apple E=Hat
swap D C  → A=Book B=Dog C=Apple D=Cat   E=Hat
swap B E  → A=Book B=Hat C=Apple D=Cat   E=Dog
swap E B  → A=Book B=Dog C=Apple D=Cat   E=Hat
```

GPT-5.6 Luna

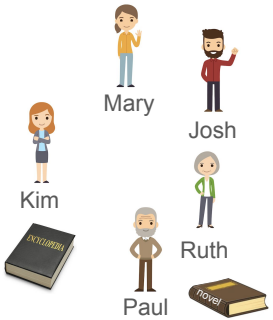
```
swap D A  → A=Cat B=Dog   C=Apple D=Book  E=Hat
swap B E  → A=Cat B=Hat   C=Apple D=Book  E=Dog
swap B C  → A=Cat B=Apple C=Hat   D=Book  E=Dog
swap D E  → A=Cat B=Apple C=Hat   D=Dog   E=Book
swap B C  → A=Cat B=Hat   C=Apple D=Dog   E=Book
swap B A  → A=Hat B=Cat   C=Apple D=Dog   E=Book
swap D C  → A=Hat B=Cat   C=Dog   D=Apple E=Book
swap B C  → A=Hat B=Dog   C=Cat   D=Apple E=Book
swap B C  → A=Hat  B=Cat C=Dog   D=Apple E=Book
swap A E  → A=Book B=Cat C=Dog   D=Apple E=Hat
swap D C  → A=Book B=Cat C=Apple D=Dog   E=Hat
swap B E  → A=Book B=Hat C=Apple D=Dog   E=Cat
```

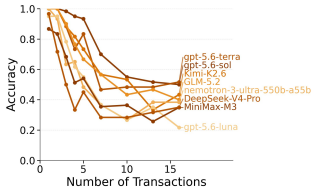
LLM repeats a transition.

From then on, everything is messed up.

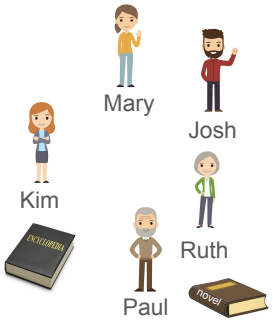

S5
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
0
100
200
300
Length
with signpost tokens
GPT-5.6 Luna

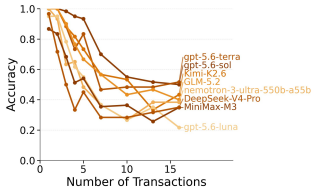
Architectural Constraints





Architectural Constraints

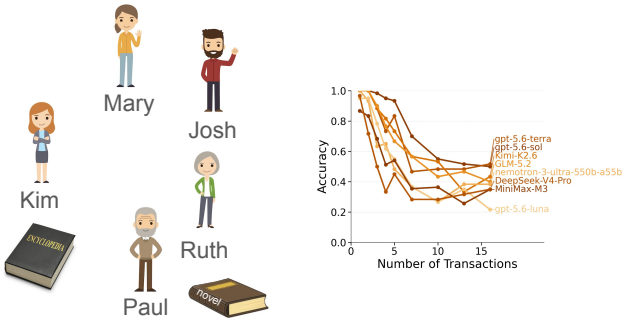




The result of 61157×4555 is **278657635**. *incorrect*

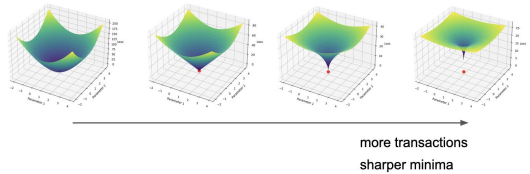
actual: 278570135

Architectural Constraints

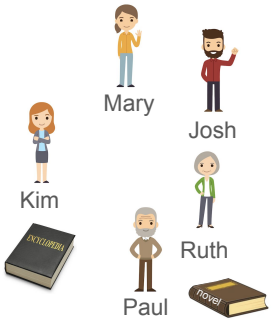


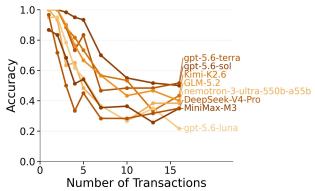
The result of 61157×4555 is **278657635**. *incorrect*

actual: 278570135



Architectural Constraints





C-RASP Predicts Generalization

UNIQUE COPY

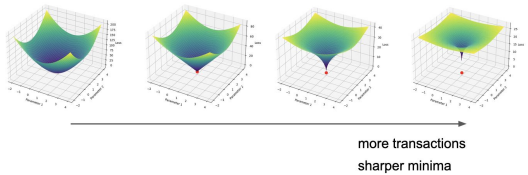
$$CP_a(i) := \#[j \le i, j = i-1]\ Q_a(j) \quad (1)$$

$$CBIGRAM_{ab} := \#[j \le i]\ Q_b(j) \wedge CP_a(j) \ge 1 \quad (2)$$

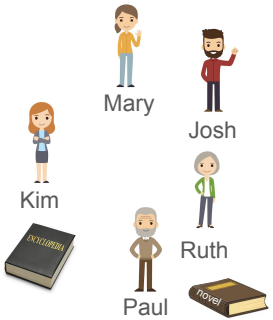
$$NEXT_a(i) := \bigvee_{\sigma \in \Sigma} [Q_\sigma(i) \wedge CBIGRAM_{\sigma a}(i) \ge 1] \quad (3)$$

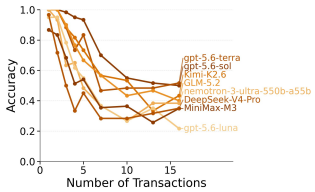
The result of 61157×4555 is **278657635**. *incorrect*

actual: 27857013 5



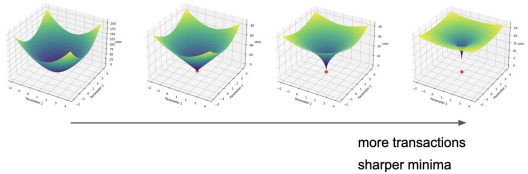
Architectural Constraints





The result of 61157×4555 is **278657635**. *incorrect*

actual: 278570135



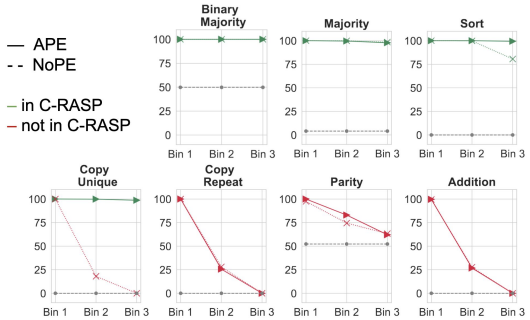
C-RASP Predicts Generalization

UNIQUE COPY

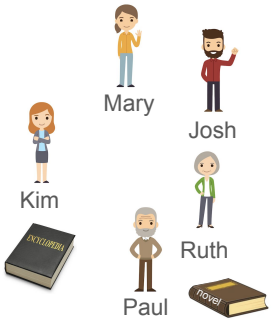
$$CP_a(i) := \#[j \le i, j = i-1]\ Q_a(j) \quad (1)$$

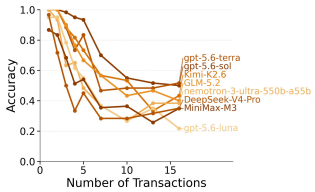
$$CBIGRAM_{ab} := \#[j \le i]\ Q_b(j) \wedge CP_a(j) \ge 1 \quad (2)$$

$$NEXT_a(i) := \bigvee_{\sigma \in \Sigma} [Q_\sigma(i) \wedge CBIGRAM_{\sigma a}(i) \ge 1] \quad (3)$$



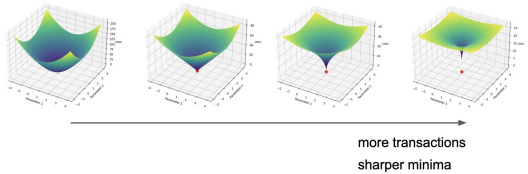
Architectural Constraints





The result of 61157×4555 is **278657635**. *incorrect*

actual: 278570135



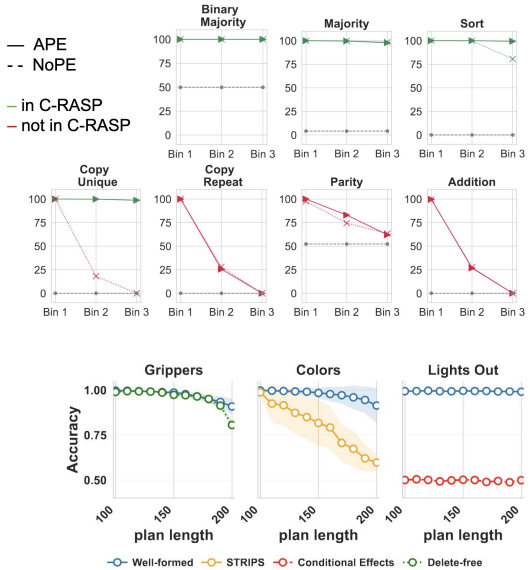
C-RASP Predicts Generalization

UNIQUE COPY

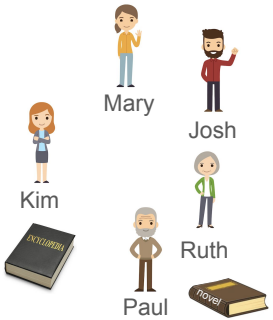
$$CP_a(i) := \#[j \le i, j = i-1]\ Q_a(j) \quad (1)$$

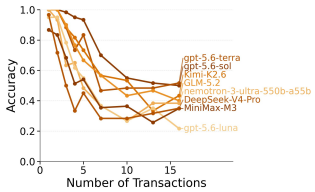
$$CBIGRAM_{ab} := \#[j \le i]\ Q_b(j) \wedge CP_a(j) \ge 1 \quad (2)$$

$$NEXT_a(i) := \bigvee_{\sigma \in \Sigma} [Q_\sigma(i) \wedge CBIGRAM_{\sigma a}(i) \ge 1] \quad (3)$$



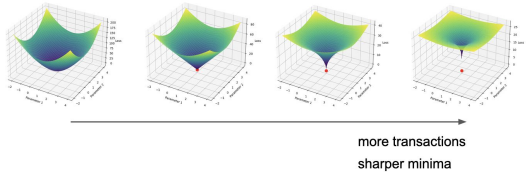
Architectural Constraints





The result of 61157×4555 is **278657635**. *incorrect*

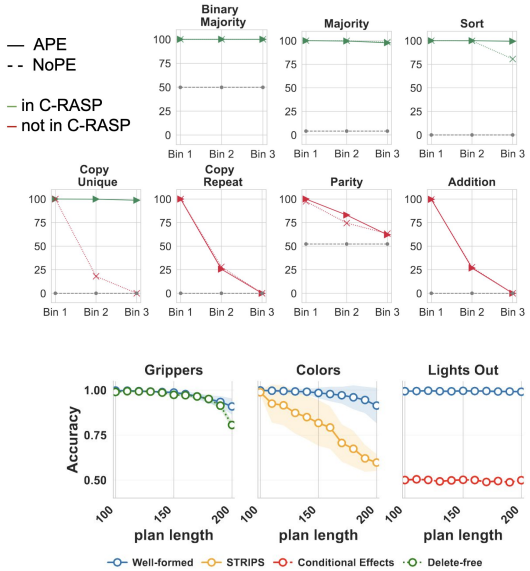
actual: 278570135



C-RASP Predicts Generalization

UNIQUE COPY

$$CP_a(i) := \#[j \le i, j = i-1]\ Q_a(j) \quad (1)$$
$$CBIGRAM_{ab} := \#[j \le i]\ Q_b(j) \wedge CP_a(j) \ge 1 \quad (2)$$
$$NEXT_a(i) := \bigvee_{\sigma \in \Sigma} [Q_\sigma(i) \wedge CBIGRAM_{\sigma a}(i) \ge 1] \quad (3)$$



Chain-of-Thought is Turing-Complete