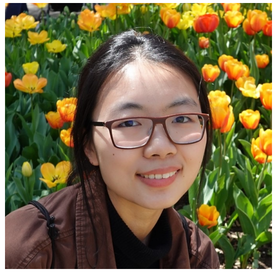
Function Induction and Task Generalization: An Interpretability Study with Off-by-One Addition


Qinyuan Ye

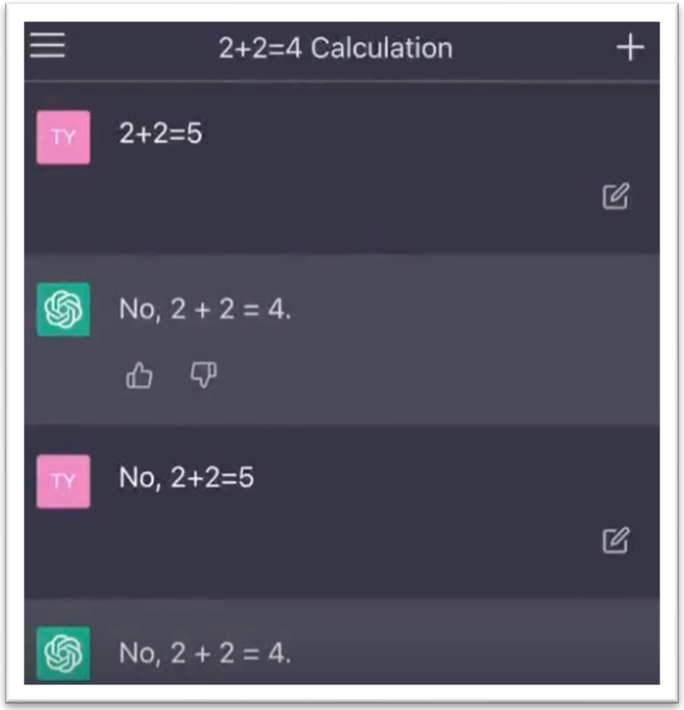

Robin Jia

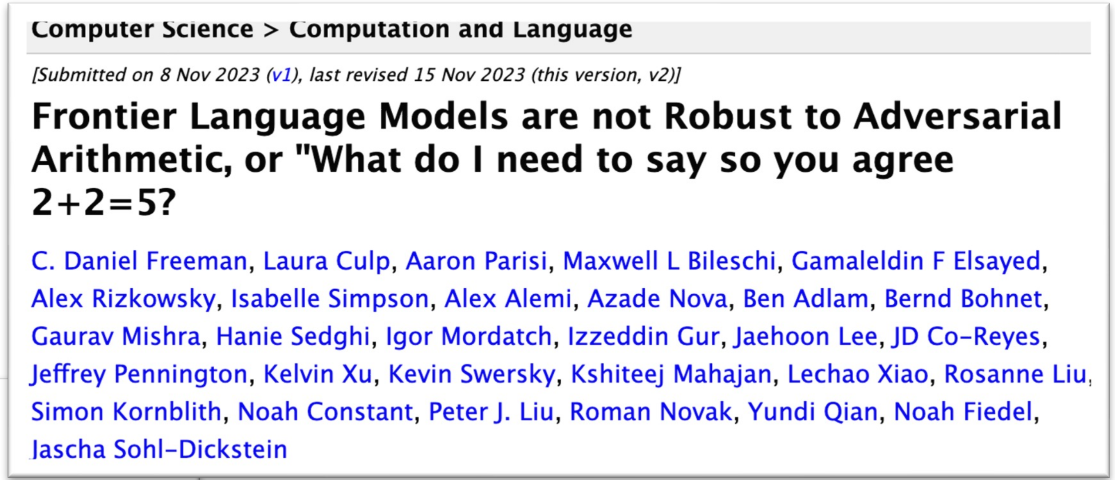

Xiang Ren

Thomas Lord Department of Computer Science
University of Southern California

September 3, 2025

How to trick language models to say “2+2=5”?

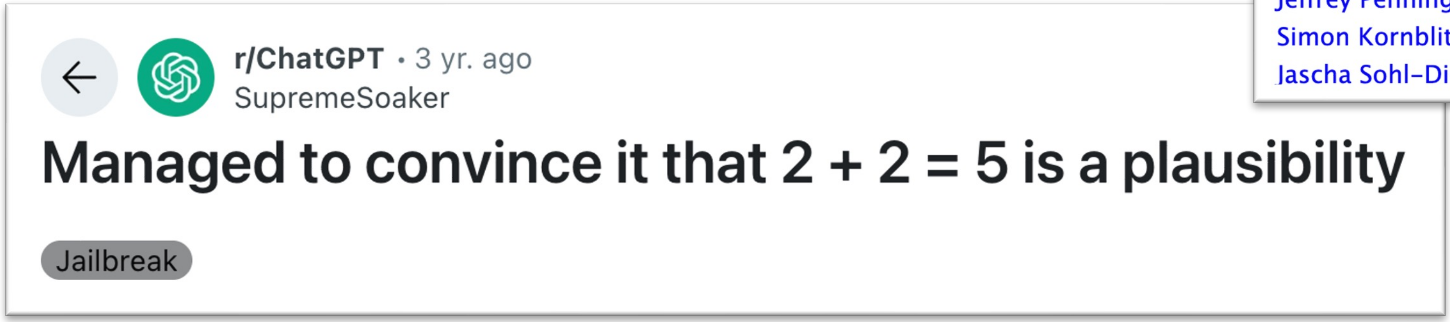



Computer Science > Computation and Language

[Submitted on 8 Nov 2023 (v1), last revised 15 Nov 2023 (this version, v2)]

Frontier Language Models are not Robust to Adversarial Arithmetic, or "What do I need to say so you agree 2+2=5?

C. Daniel Freeman, Laura Culp, Aaron Parisi, Maxwell L Bileschi, Gamaleldin F Elsayed, Alex Rizkowsky, Isabelle Simpson, Alex Alemi, Azade Nova, Ben Adlam, Bernd Bohnet, Gaurav Mishra, Hanie Sedghi, Igor Mordatch, Izzeddin Gur, Jaehoon Lee, JD Co-Reyes, Jeffrey Pennington, Kelvin Xu, Kevin Swersky, Kshiteej Mahajan, Lechao Xiao, Rosanne Liu, Simon Kornblith, Noah Constant, Peter J. Liu, Roman Novak, Yundi Qian, Noah Fiedel, Jascha Sohl-Dickstein



How to trick language models to say “2+2=5”?

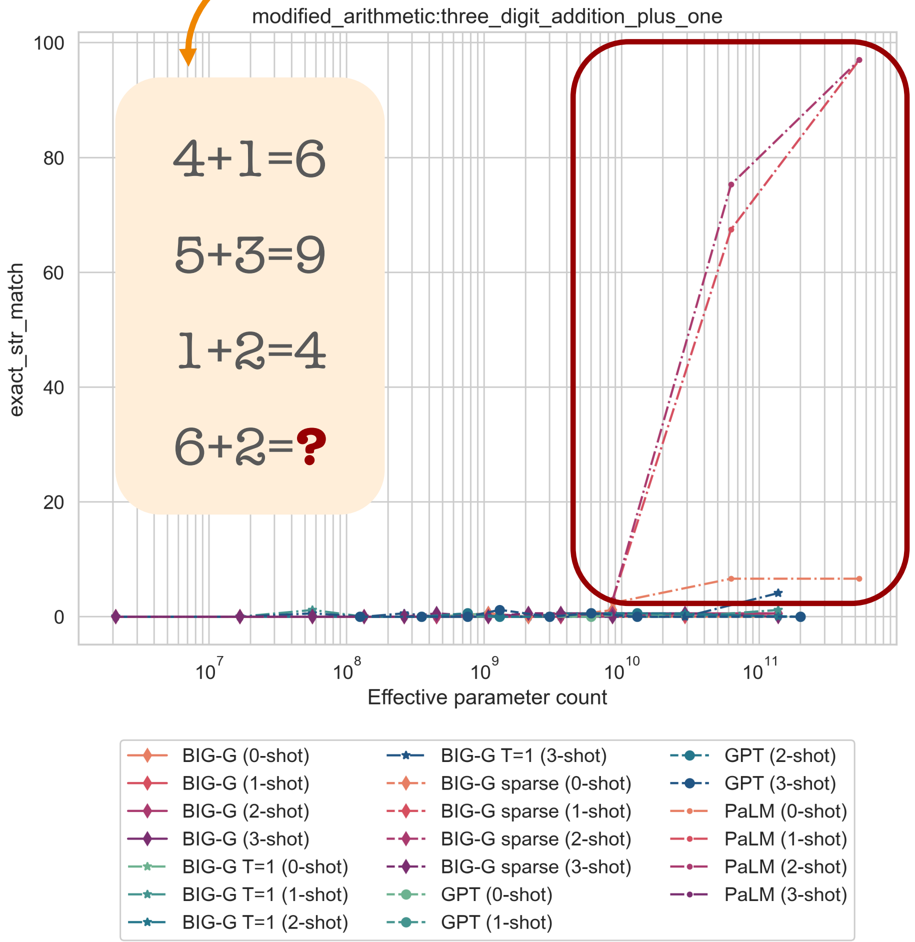
```
from transformers import pipeline

pipe = pipeline("text-generation", model="meta-llama/Meta-Llama-3-8B", device=device)
result = pipe("1+1=3\n2+2=", max_new_tokens=1, do_sample=False)

print(result[0]['generated_text'])
```

Llama 3

```
1+1=3
2+2=5
```

First documented in BIG-bench

Off-by-one Addition

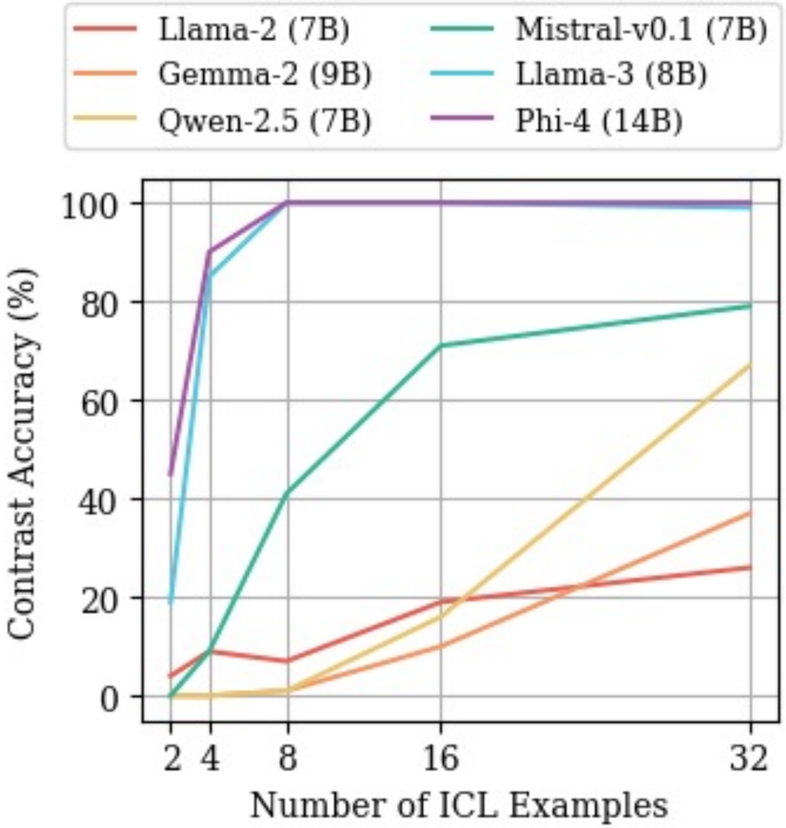


PaLM 64B and 535B have non-trivial performance.

Identified as an "emergent ability".

Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models (Srivastava et al., 2022); Emergent Abilities of Large Language Models (Wei et al., 2022)

Our evaluation with more recent models



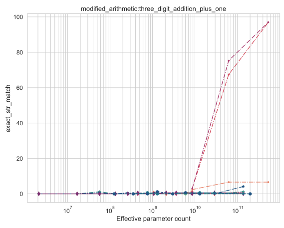
More recent, smaller models can perform this task well!

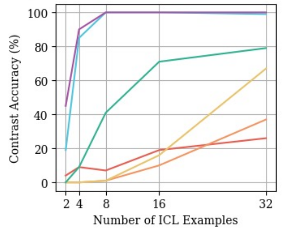
Research Question

Llama 3

1+1=3
2+2=5

PaLM





How do LMs perform off-by-one addition?

Can models learn unseen tasks with ICL?

How do LMs handle misinformation?

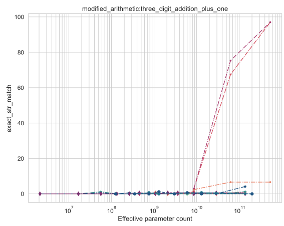
Why do emergent abilities emerge?

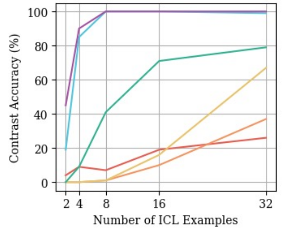
Research Question

Llama 3

1+1=3
2+2=5

PaLM





How do LMs perform off-by-one addition?

Interpretability Tools

Research Question

How do LMs perform off-by-one addition?

Interpretability

Tools

Activation Patching

Locating and Editing Factual Associations in GPT

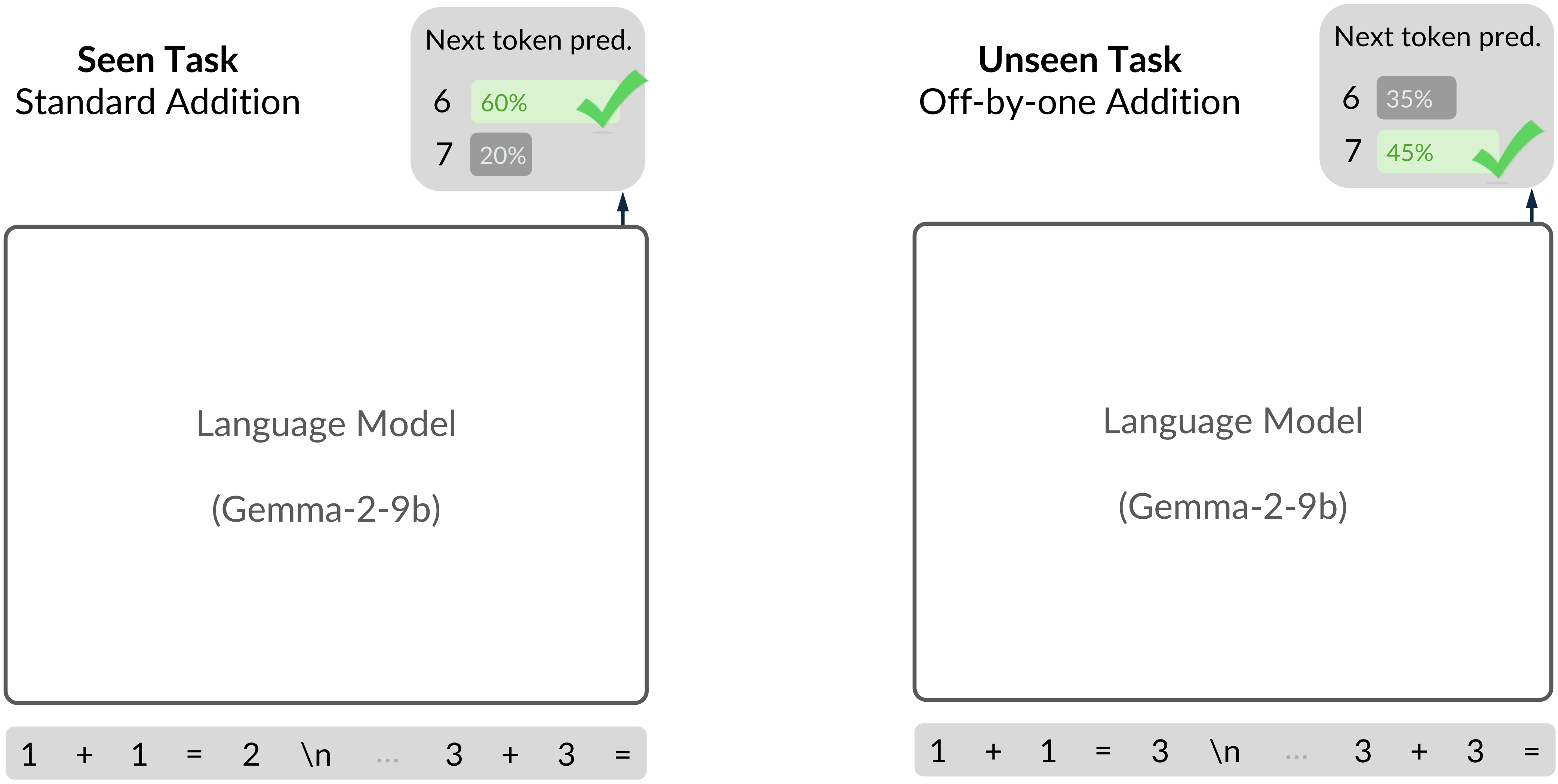
Kevi
MIT

INTERPRETABILITY IN THE WILD: A CIRCUIT FOR INDIRECT OBJECT IDENTIFICATION IN GPT-2 SMALL

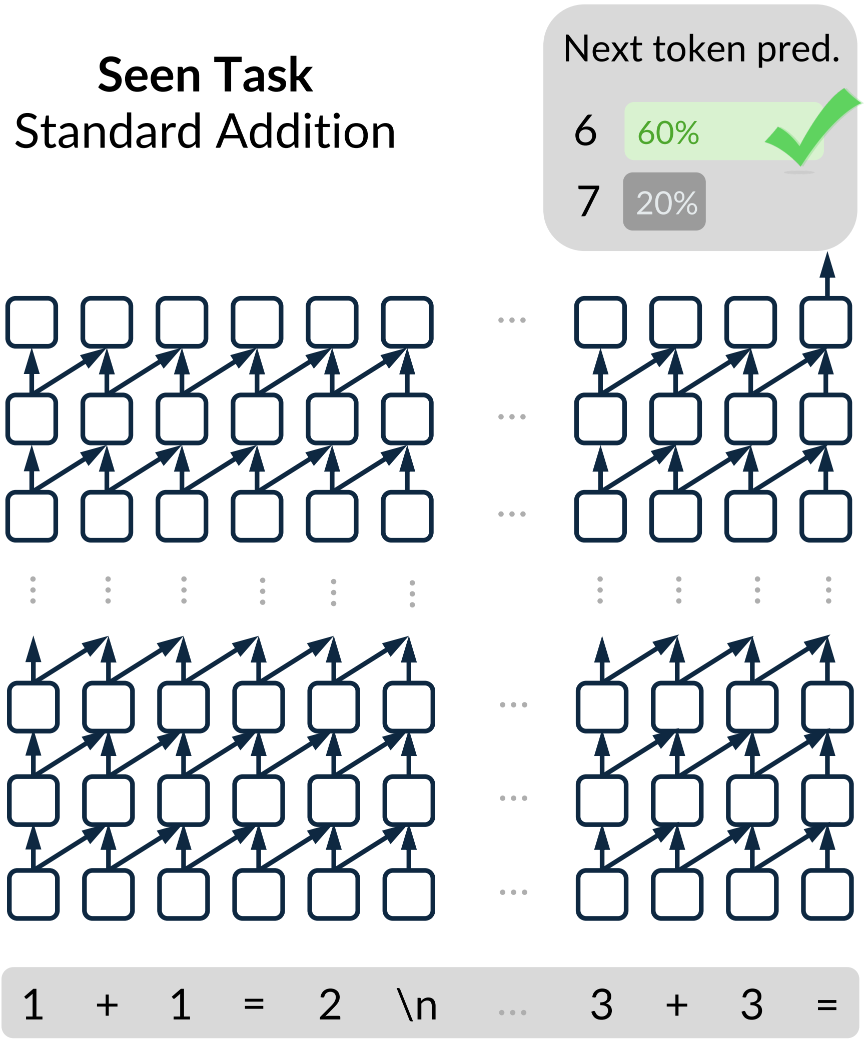
Kevin Wang[1], Alexandre Variengien[1], Arthur Conmy[1], Buck Shlegeris[1] & Jacob Steinhardt[1,2]
[1]Redwood Research
[2]UC Berkeley
kevin@rdwrs.com, alexandre@rdwrs.com,
arthur@rdwrs.com, buck@rdwrs.com, jsteinhardt@berkeley.edu

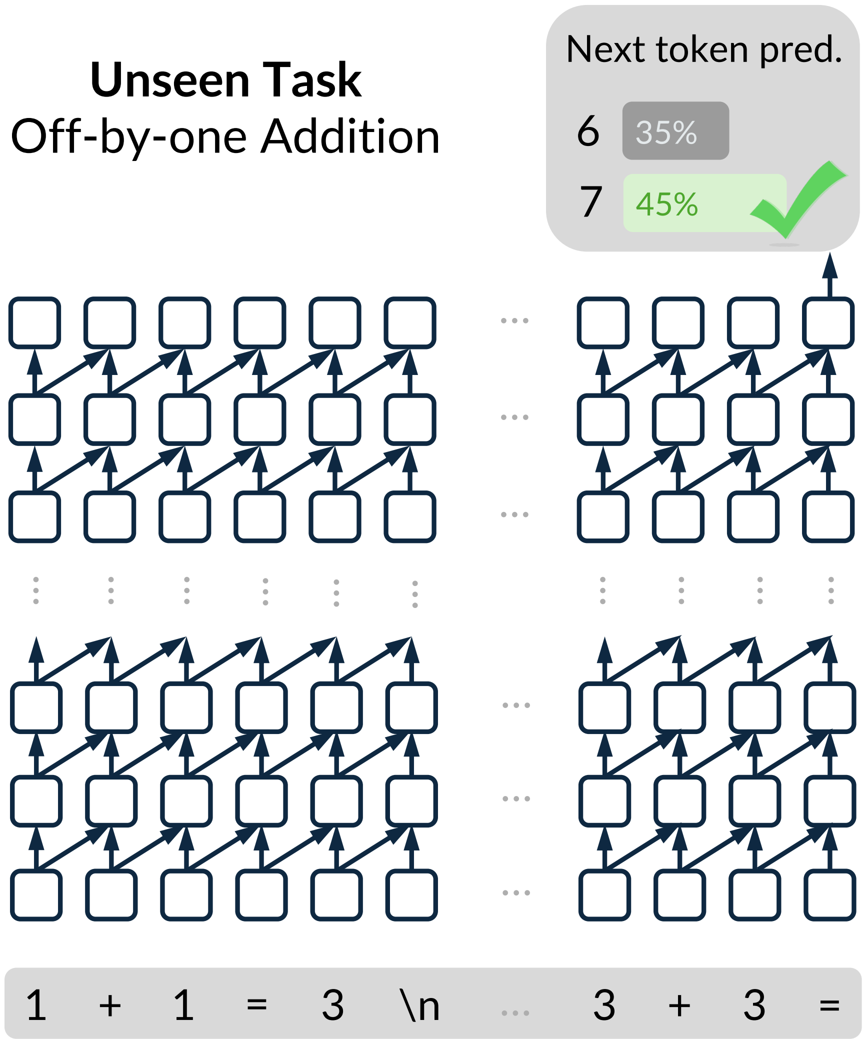
Path Patching

Interpreting Model Internals with Patching

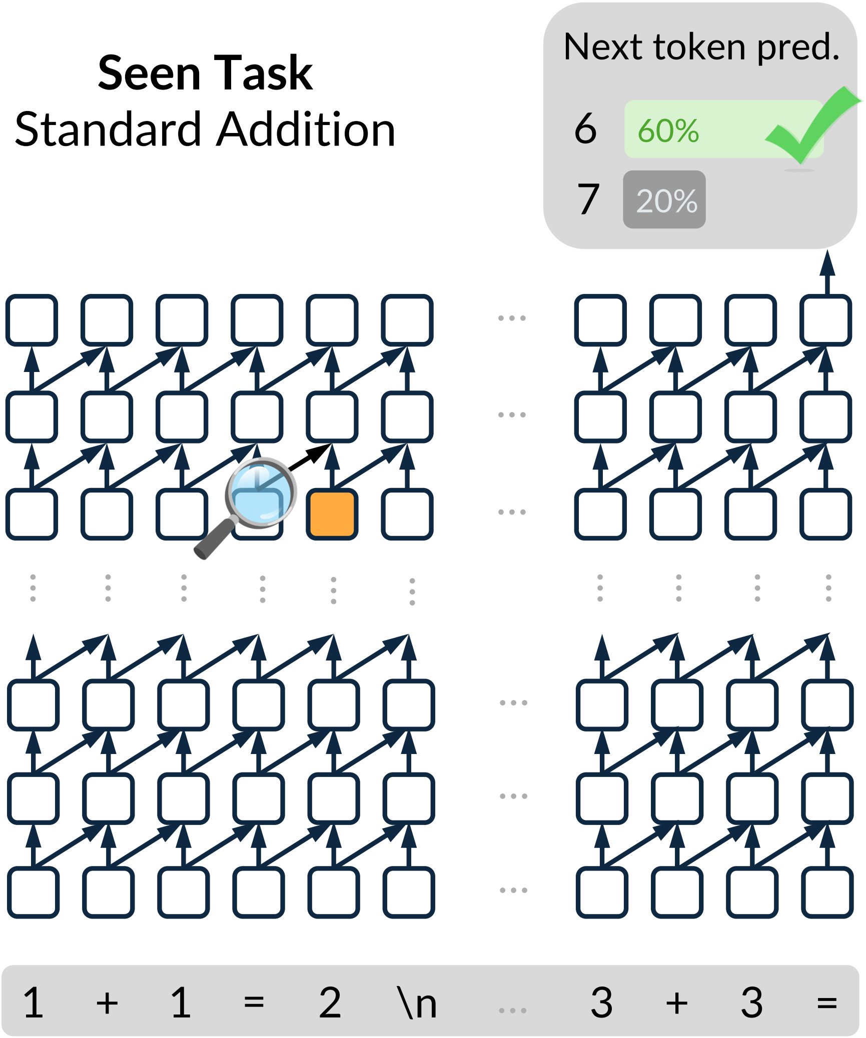


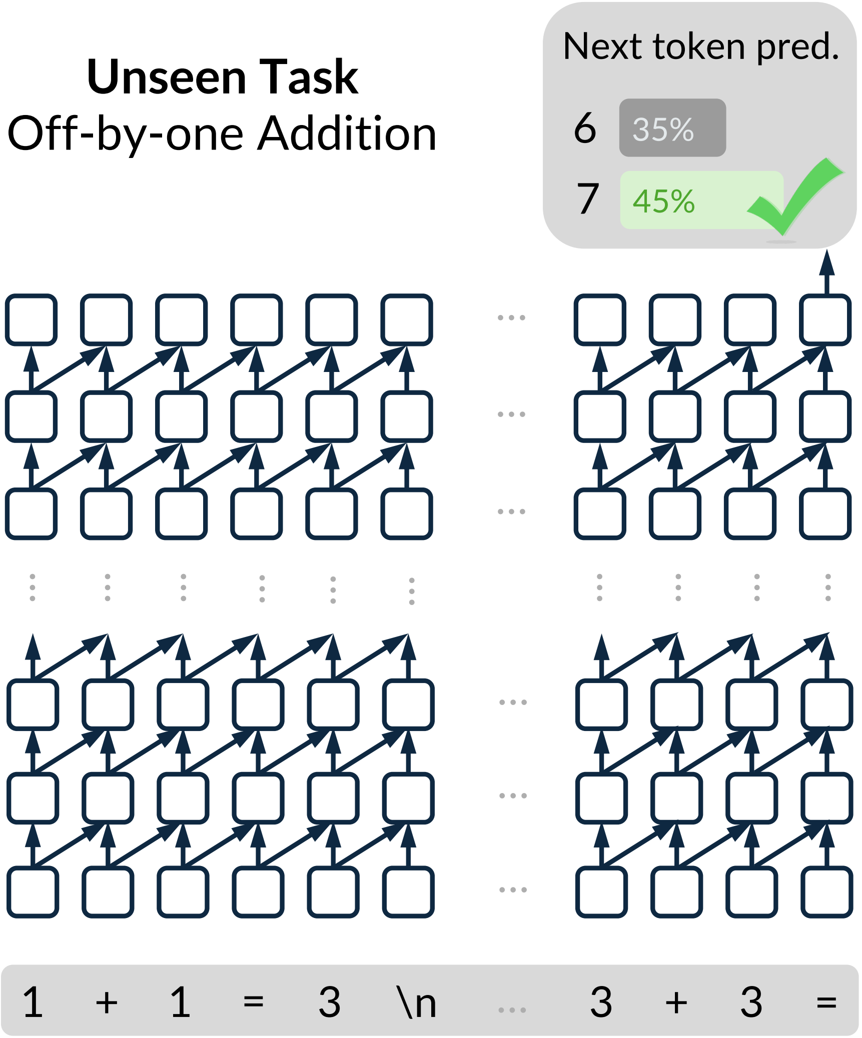
Interpreting Model Internals with Patching



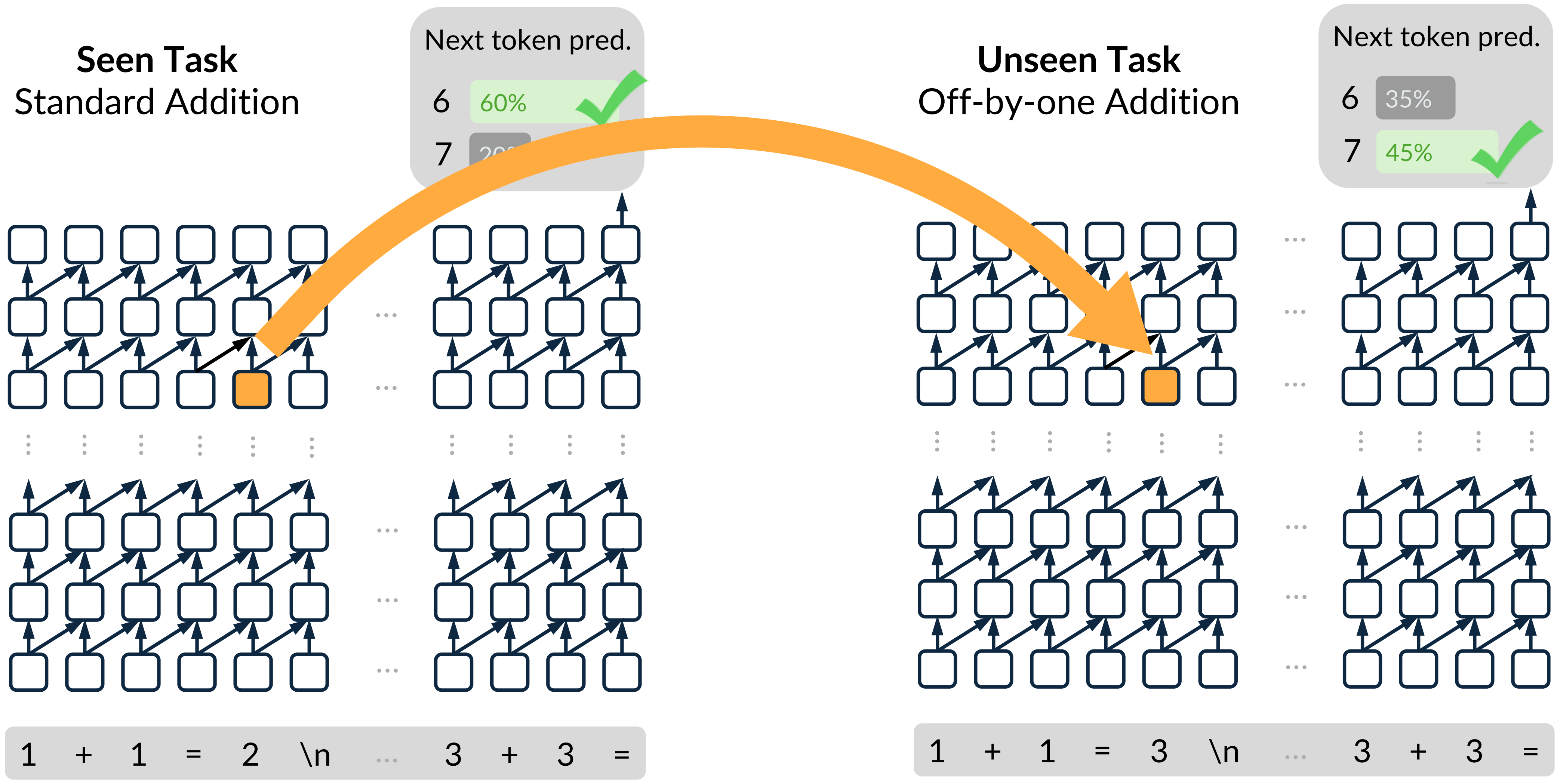


Interpreting Model Internals with Patching

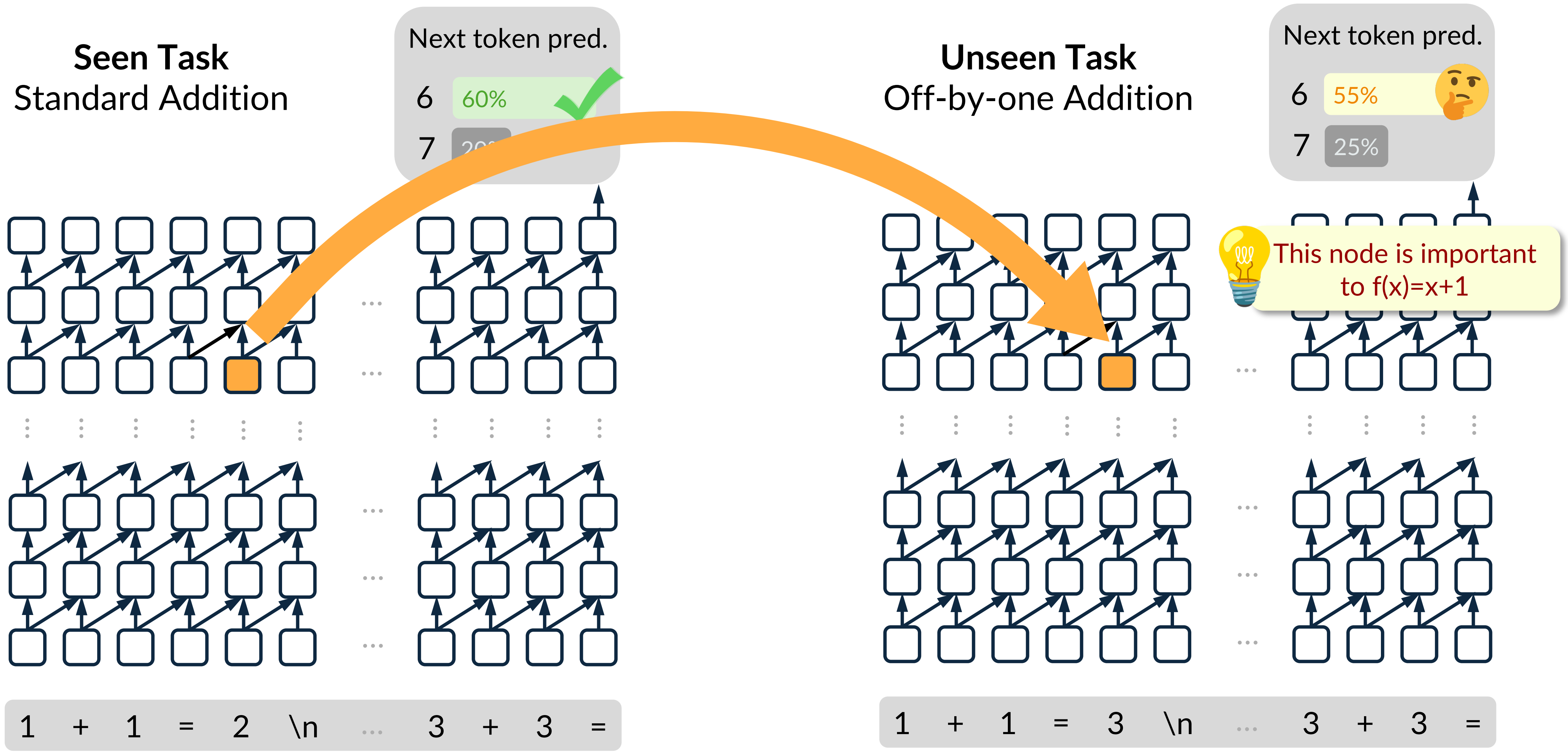




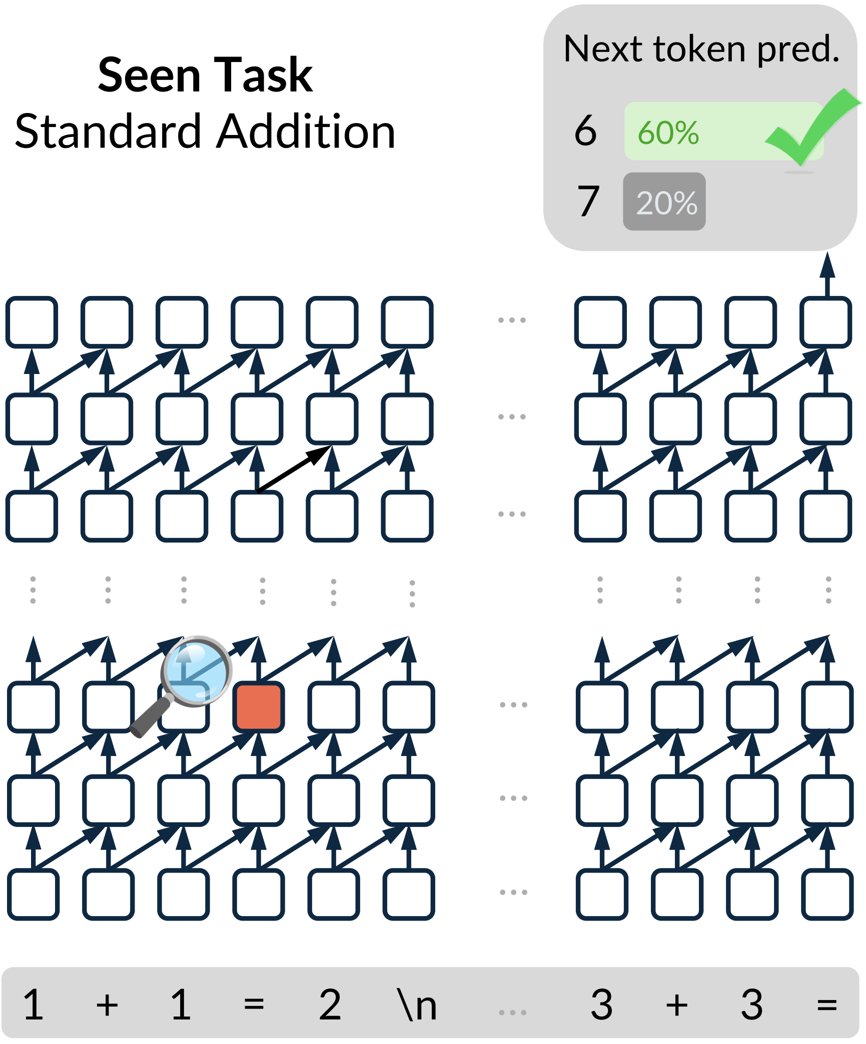
Interpreting Model Internals with Patching

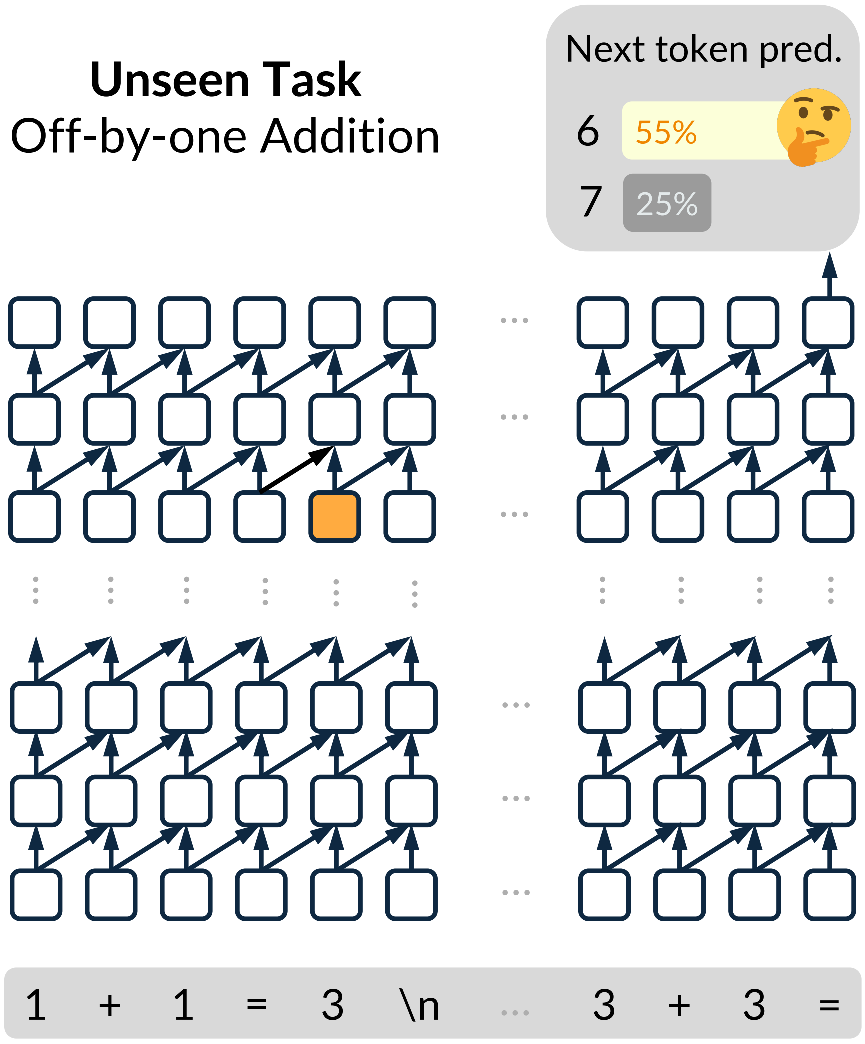


Interpreting Model Internals with Patching

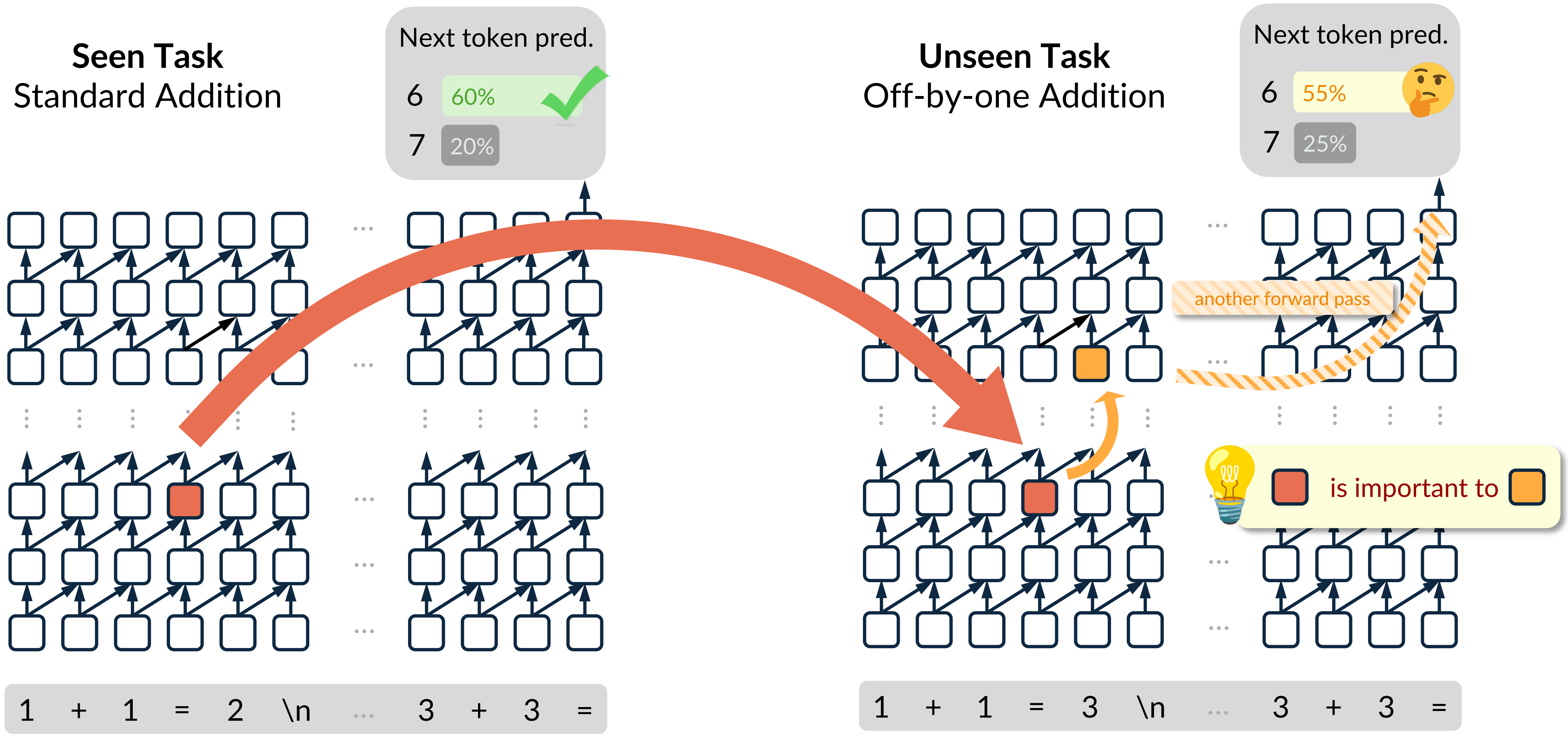


Interpreting Model Internals with Patching

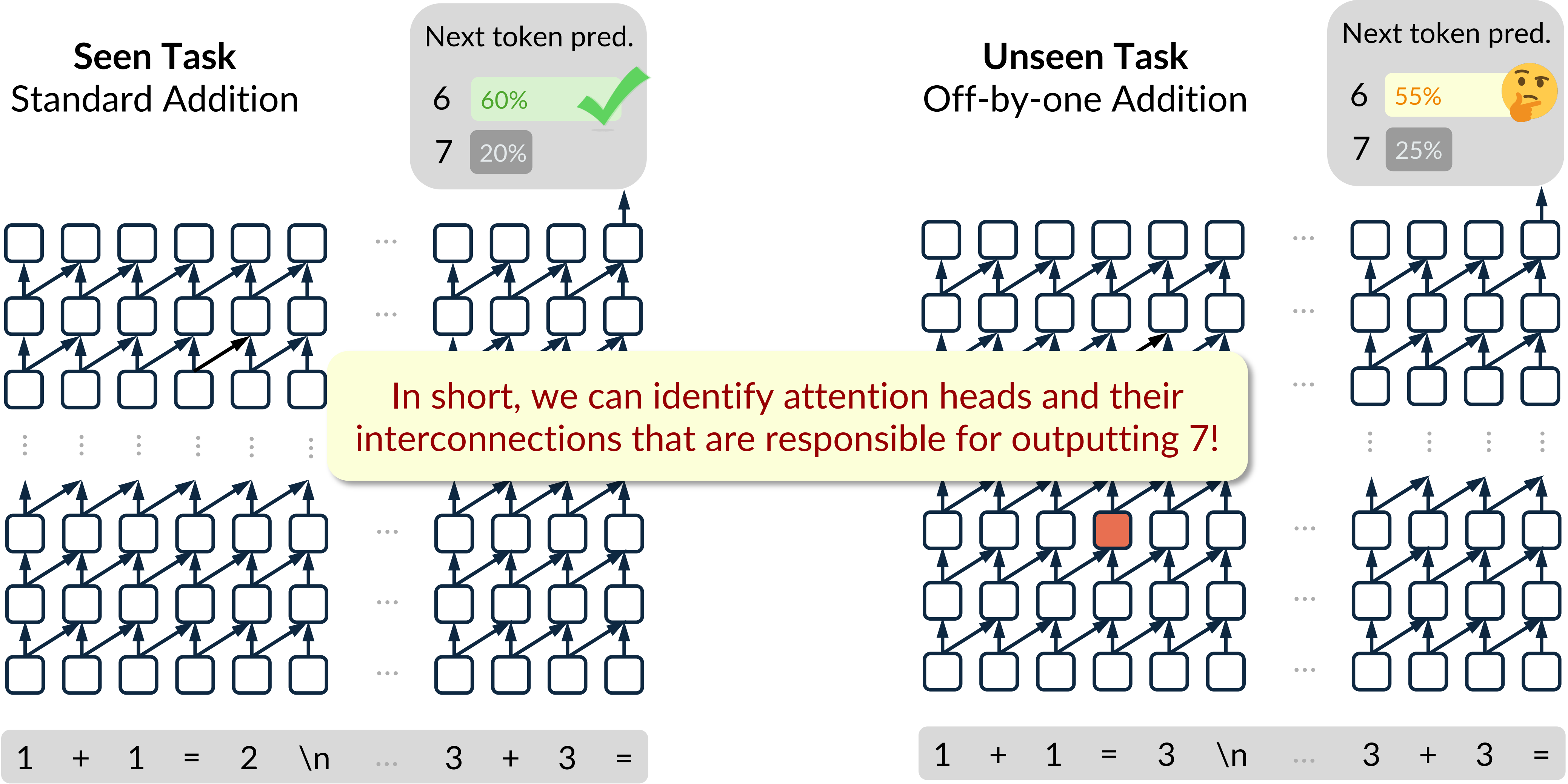




Interpreting Model Internals with Patching



Interpreting Model Internals with Patching

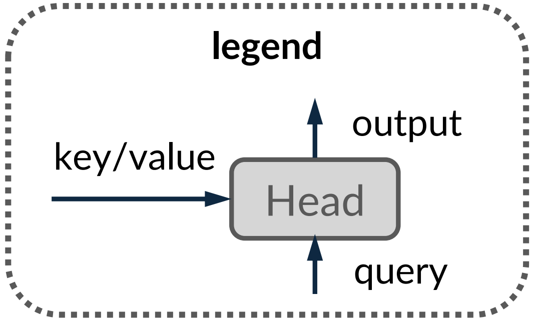


In short, we can identify attention heads and their interconnections that are responsible for outputting 7!

Patching with Contrast Tasks

Next token pred.

6	35%
7	45% ✓

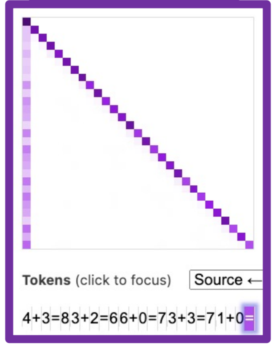


Early layers

1	+	1	=	3	\n	...	3	+	3	=

Patching with Contrast Tasks

Group 1



Next token pred.

6 35%

7 45%

Note: H41.4 means head 4 in layer 41.

H41.4, H41.5, ...

legend

key/value

output

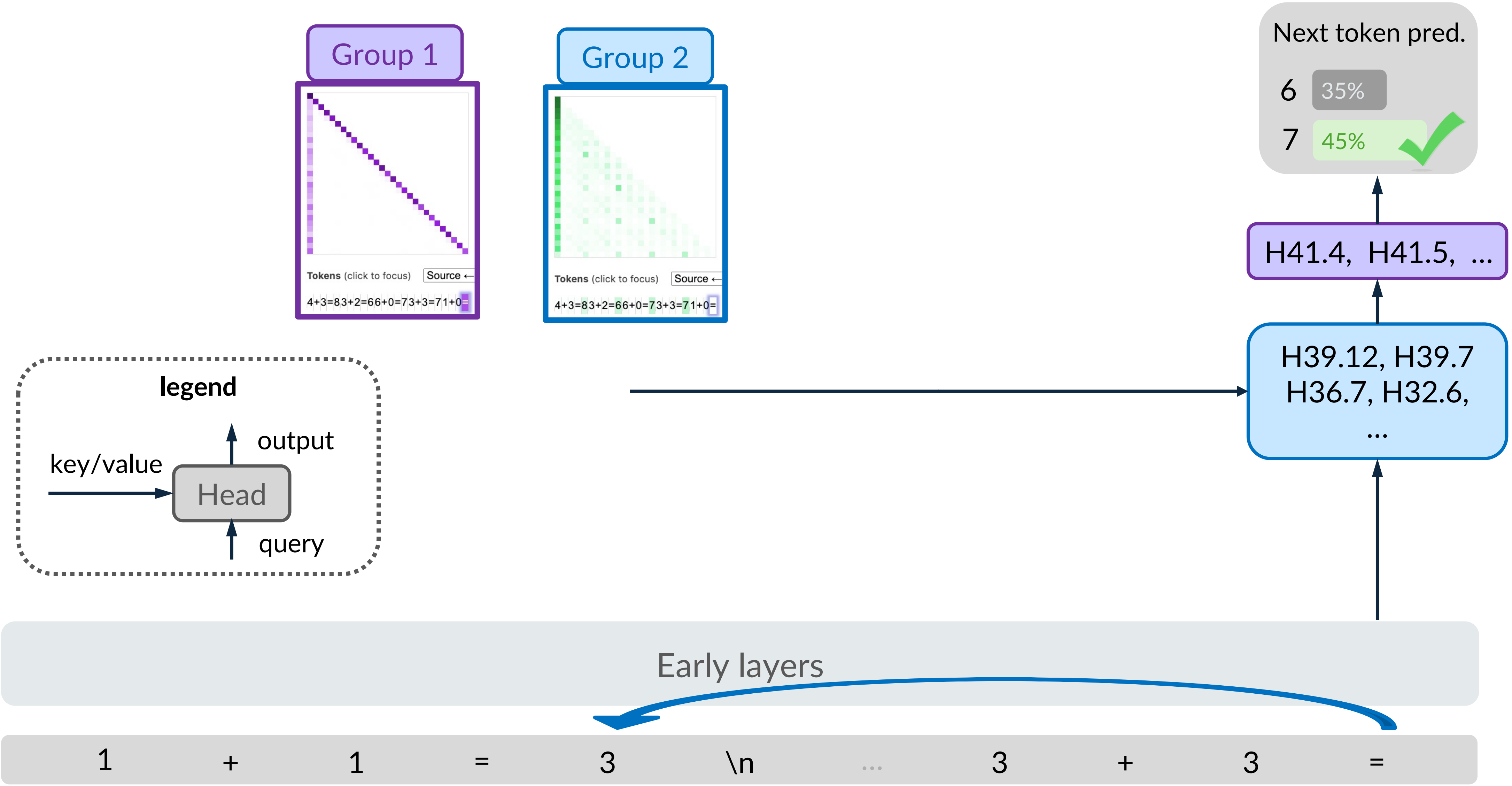
Head

query

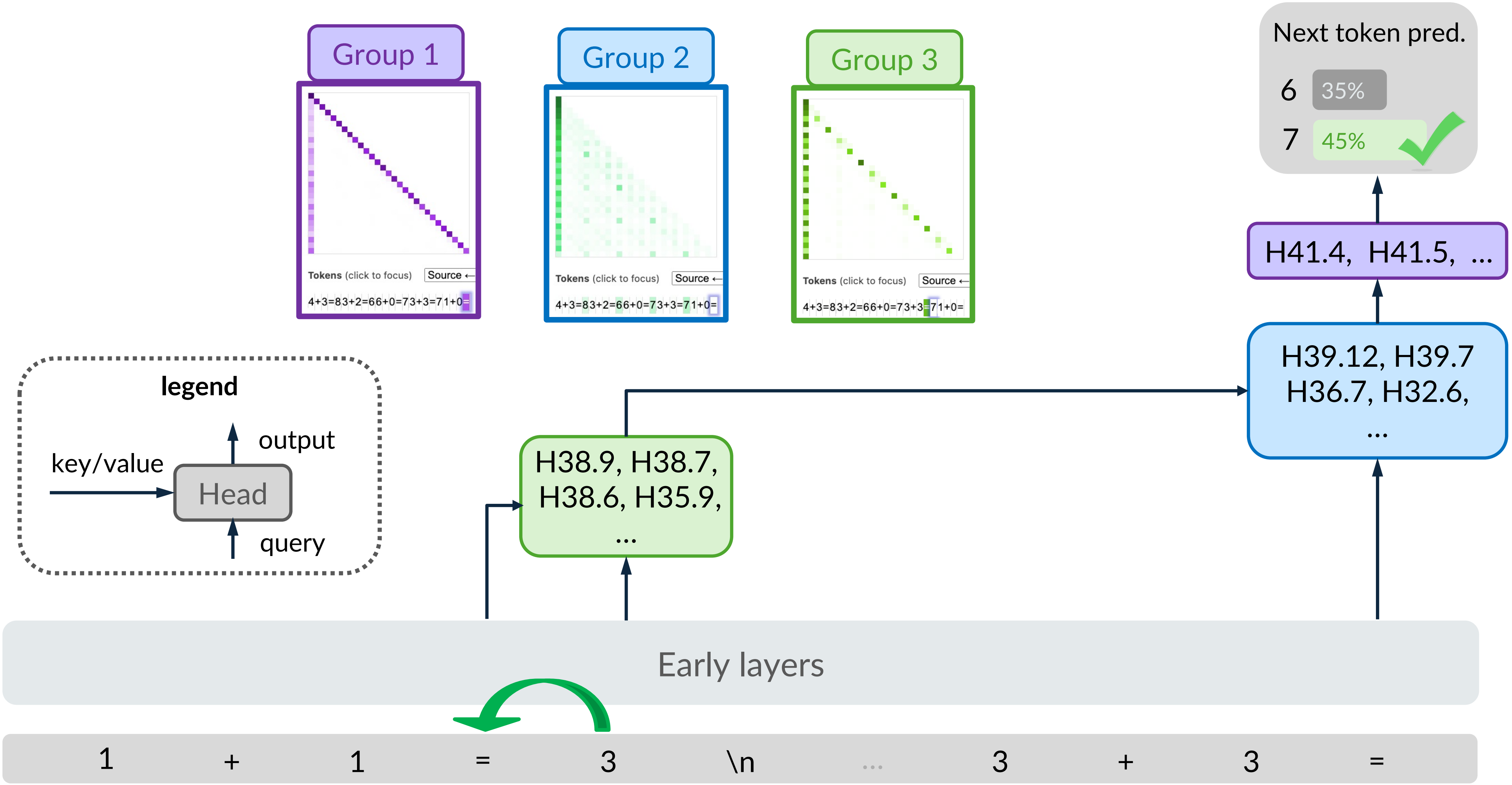
Early layers

1	+	1	=	3	\n	...	3	+	3	=

Patching with Contrast Tasks



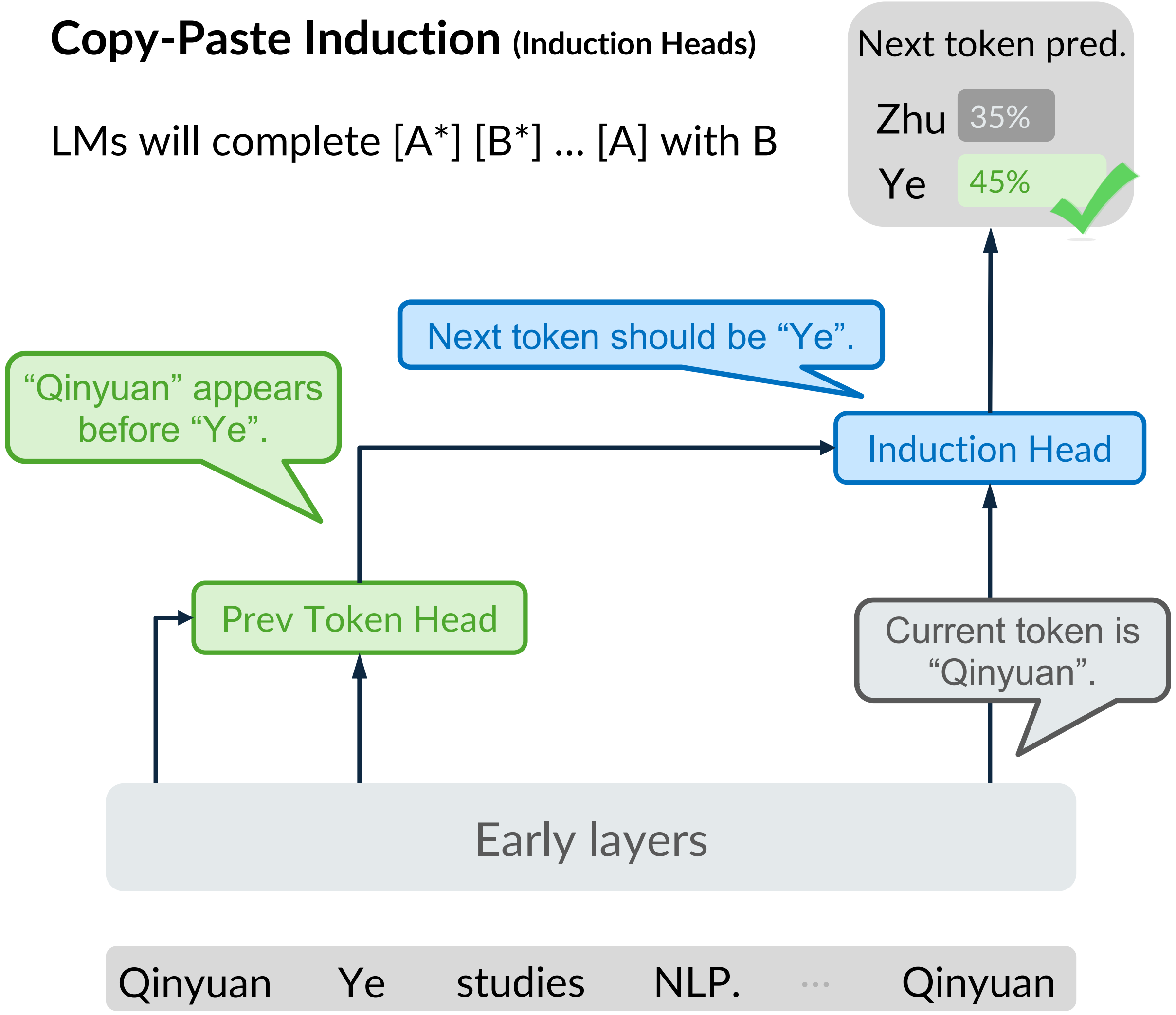
Patching with Contrast Tasks



Revisiting Induction Heads

Copy-Paste Induction (Induction Heads)

LMs will complete [A*] [B*] … [A] with B

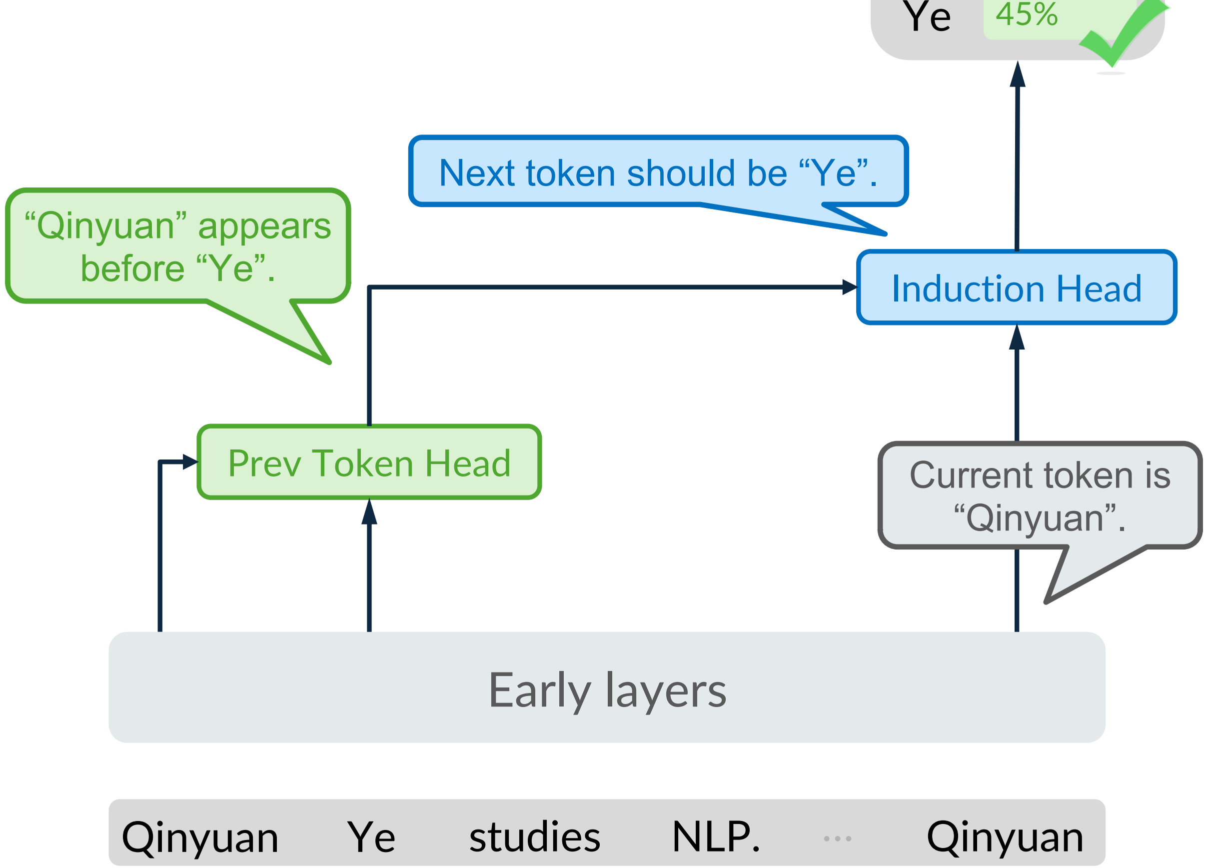


In-context Learning and Induction Heads (Olsson et al., 2022)

Revisiting Induction Heads

Copy-Paste Induction (Induction Heads)

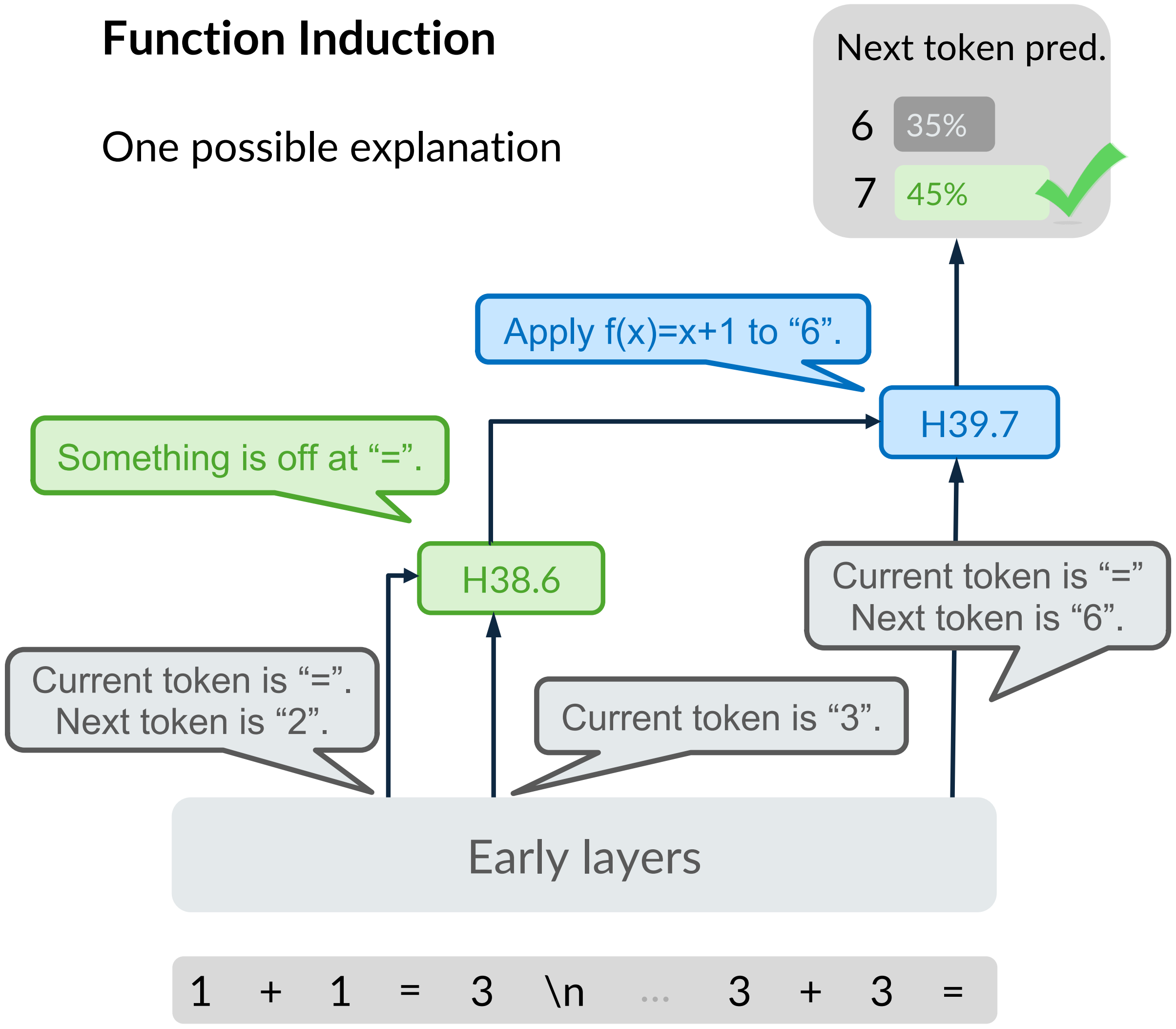
LMs will complete [A*] [B*] … [A] with B



In-context Learning and Induction Heads (Olsson et al., 2022)

Function Induction

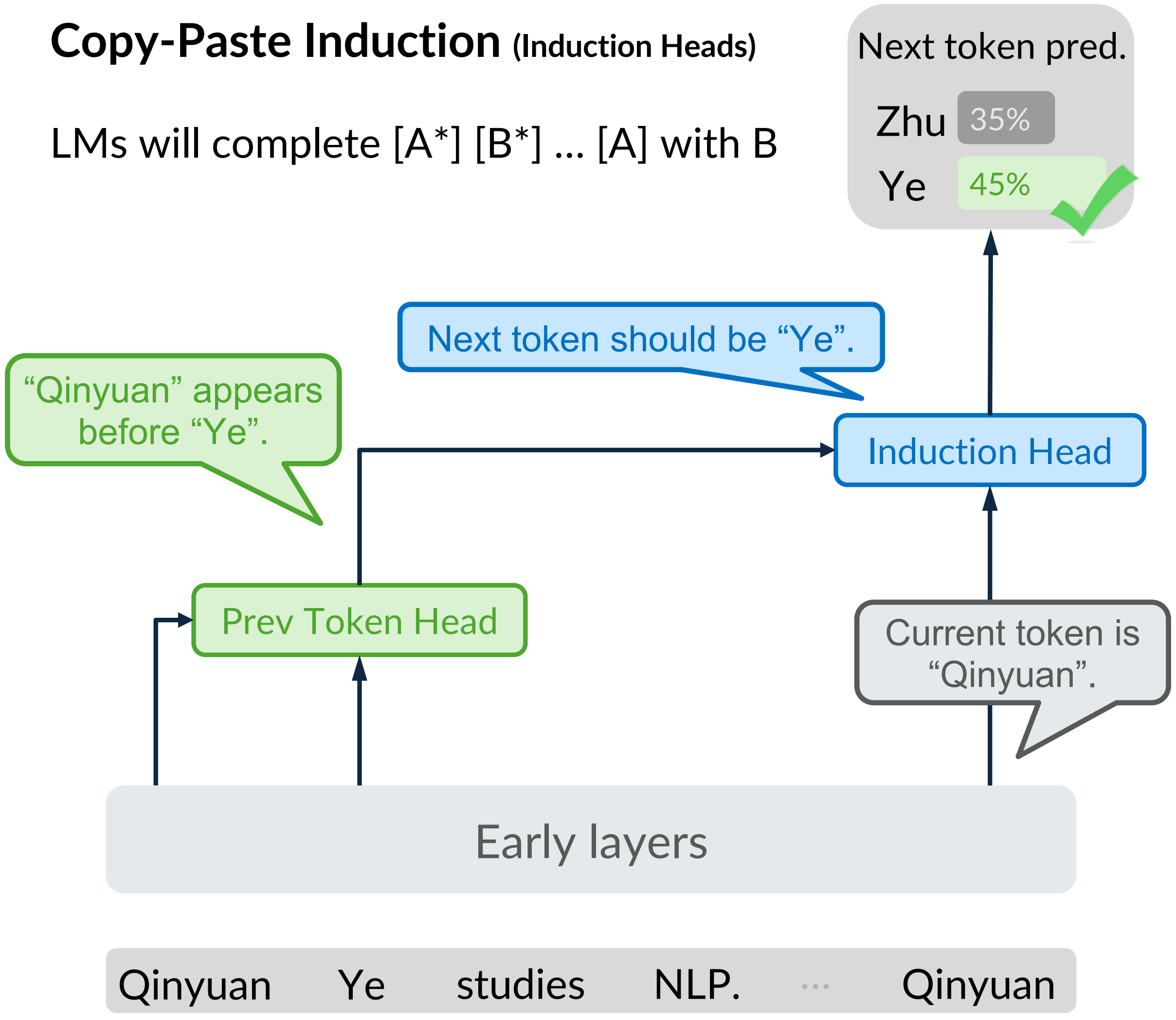
One possible explanation



Revisiting Induction Heads

Copy-Paste Induction (Induction Heads)

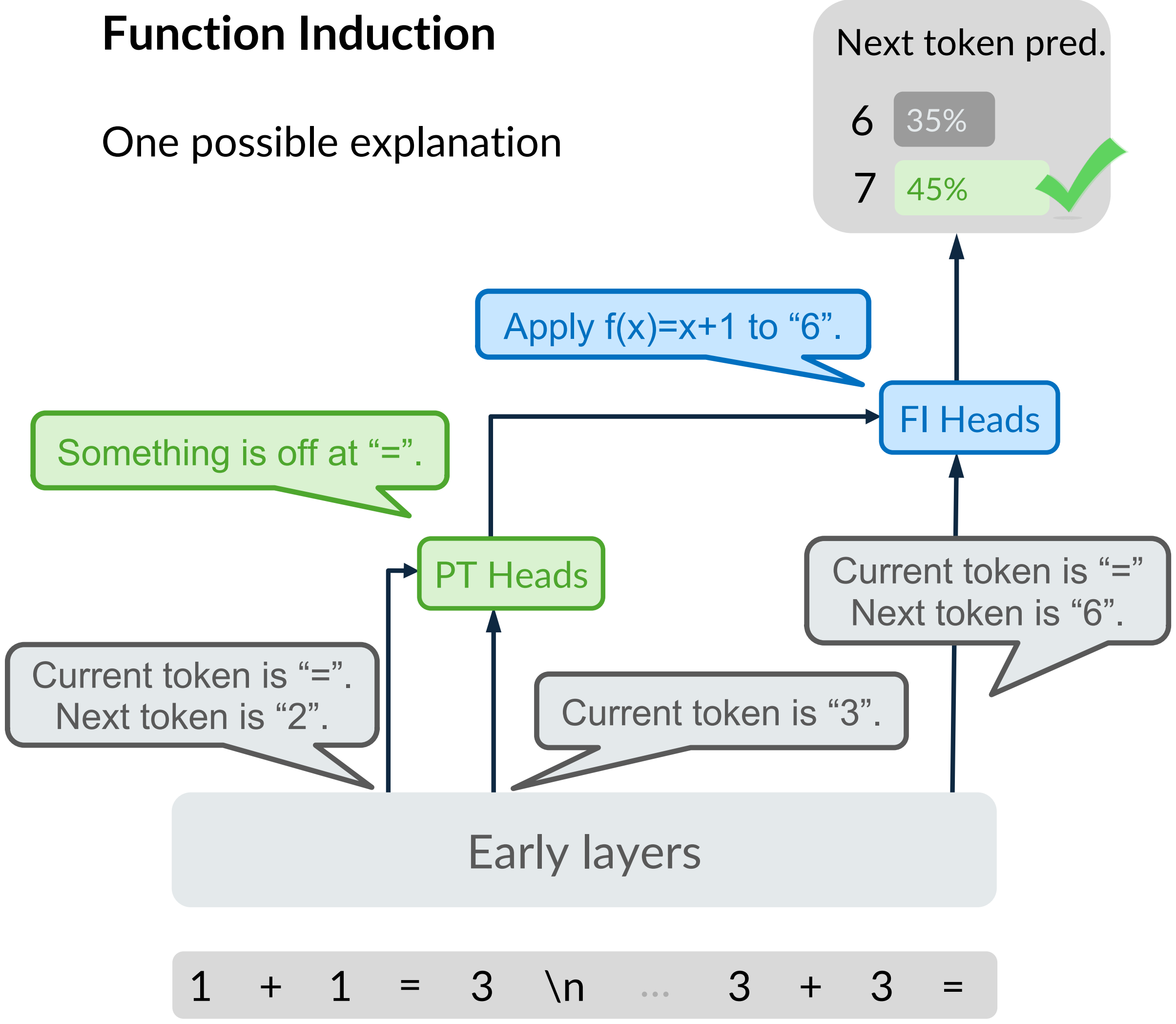
LMs will complete [A*] [B*] … [A] with B



In-context Learning and Induction Heads (Olsson et al., 2022)

Function Induction

One possible explanation

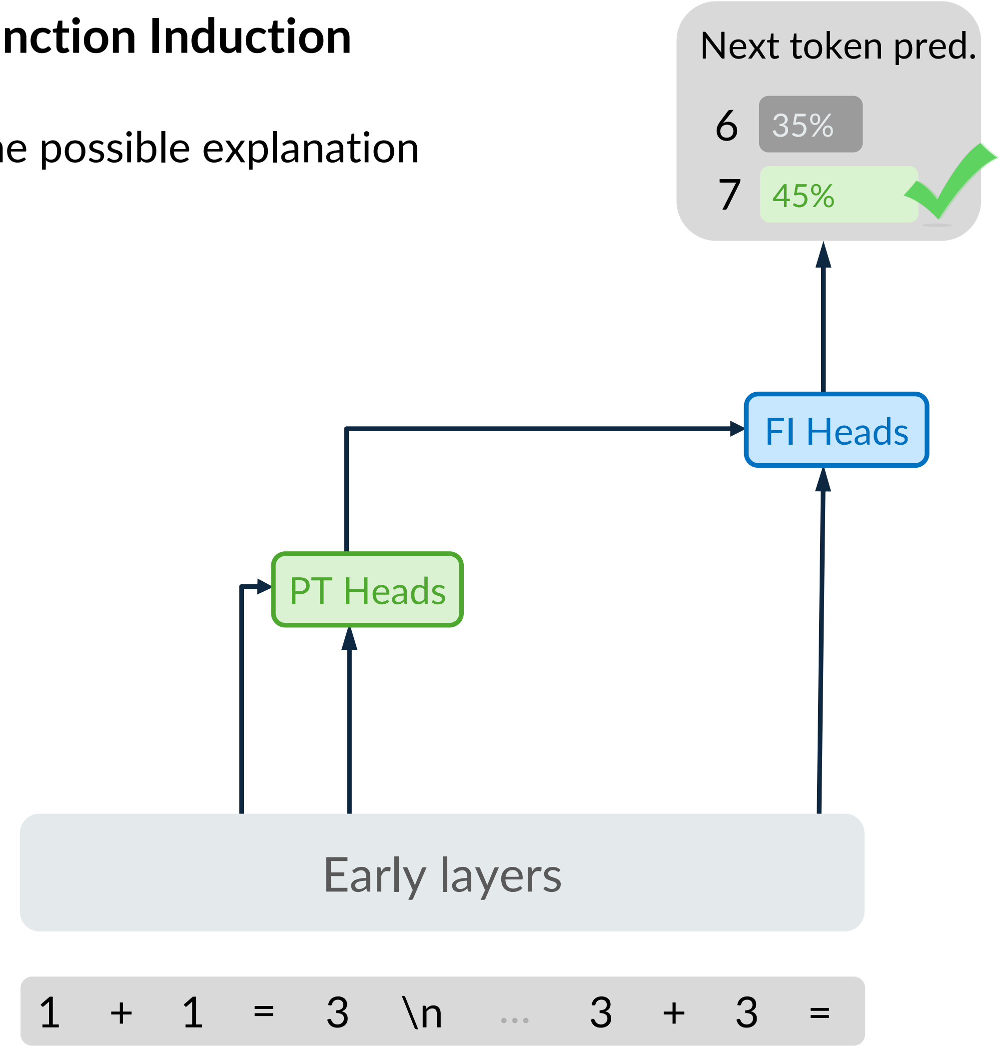


Finding 1: Function Induction Mechanism

- LMs *may be* implementing a complex **function induction** mechanism.
  - Generalizes the findings in Olsson et al., 2022;
  - Elevates it from the token level to the function level.

Function Induction

One possible explanation



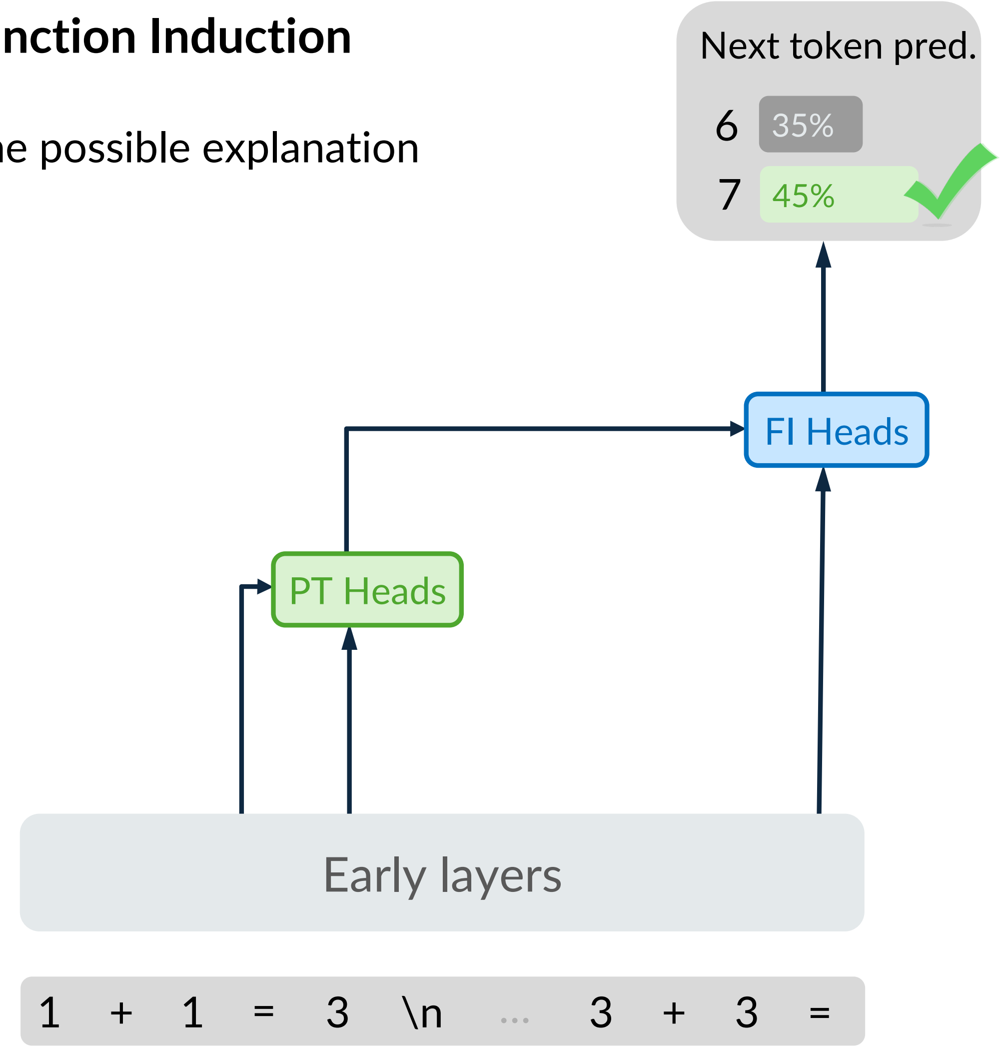
Finding 1: Function Induction Mechanism

- LMs *may be* implementing a complex **function induction** mechanism.
  - Generalizes the findings in Olsson et al., 2022;
  - Elevates it from the token level to the function level.

- ➔ **More questions**
  - Are **these heads** really writing out *f(x)=x+1*?
  - If *f(x)=x+1* is emitted 9 times via 9 heads, why is it not interpreted as "+9" by the model?

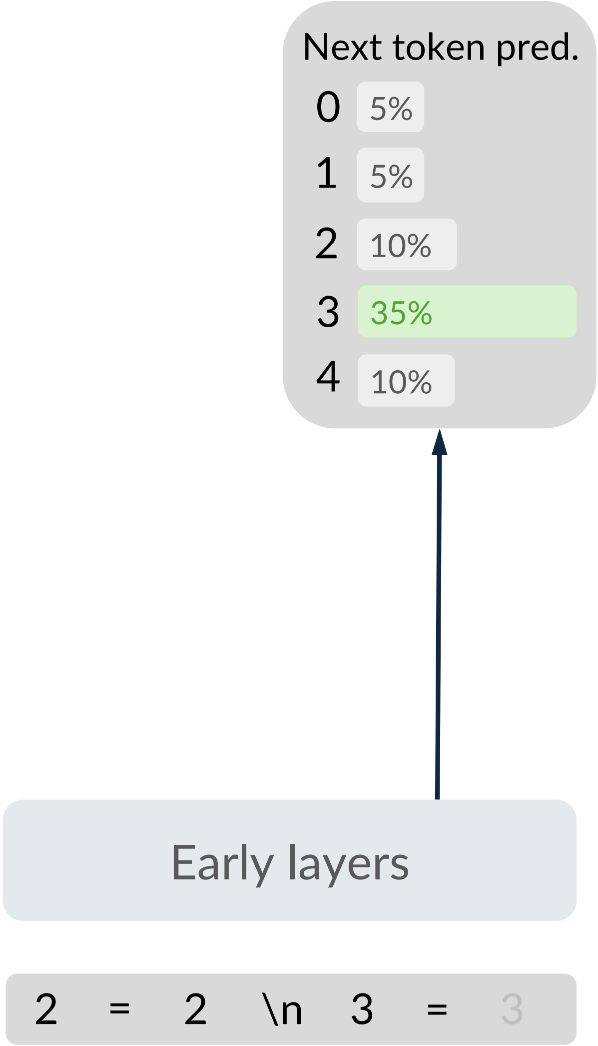
Function Induction

One possible explanation



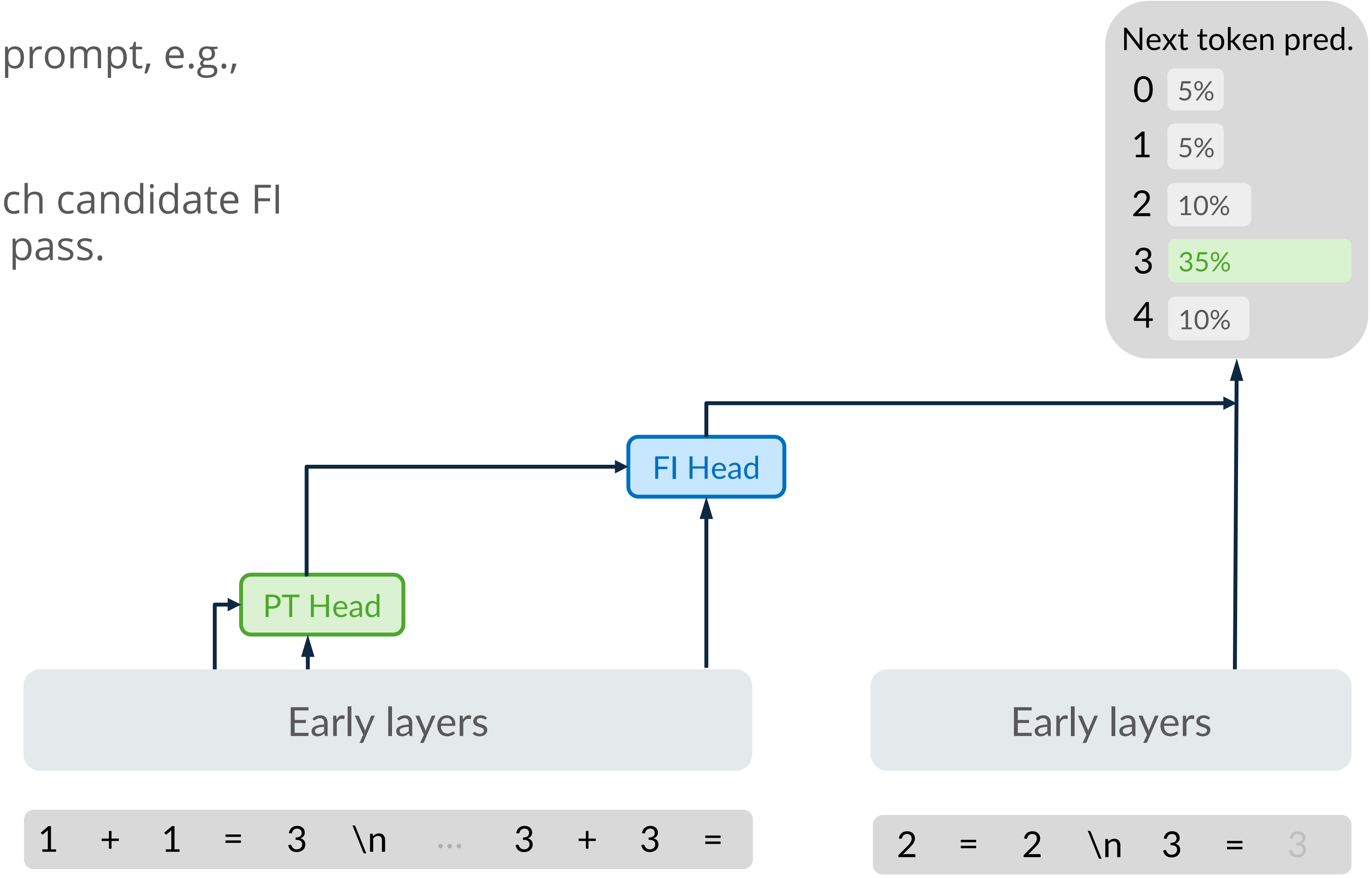
Investigating Each Candidate FI Head

- We run the LM on a naive prompt, e.g.,
  2=2, 3=?



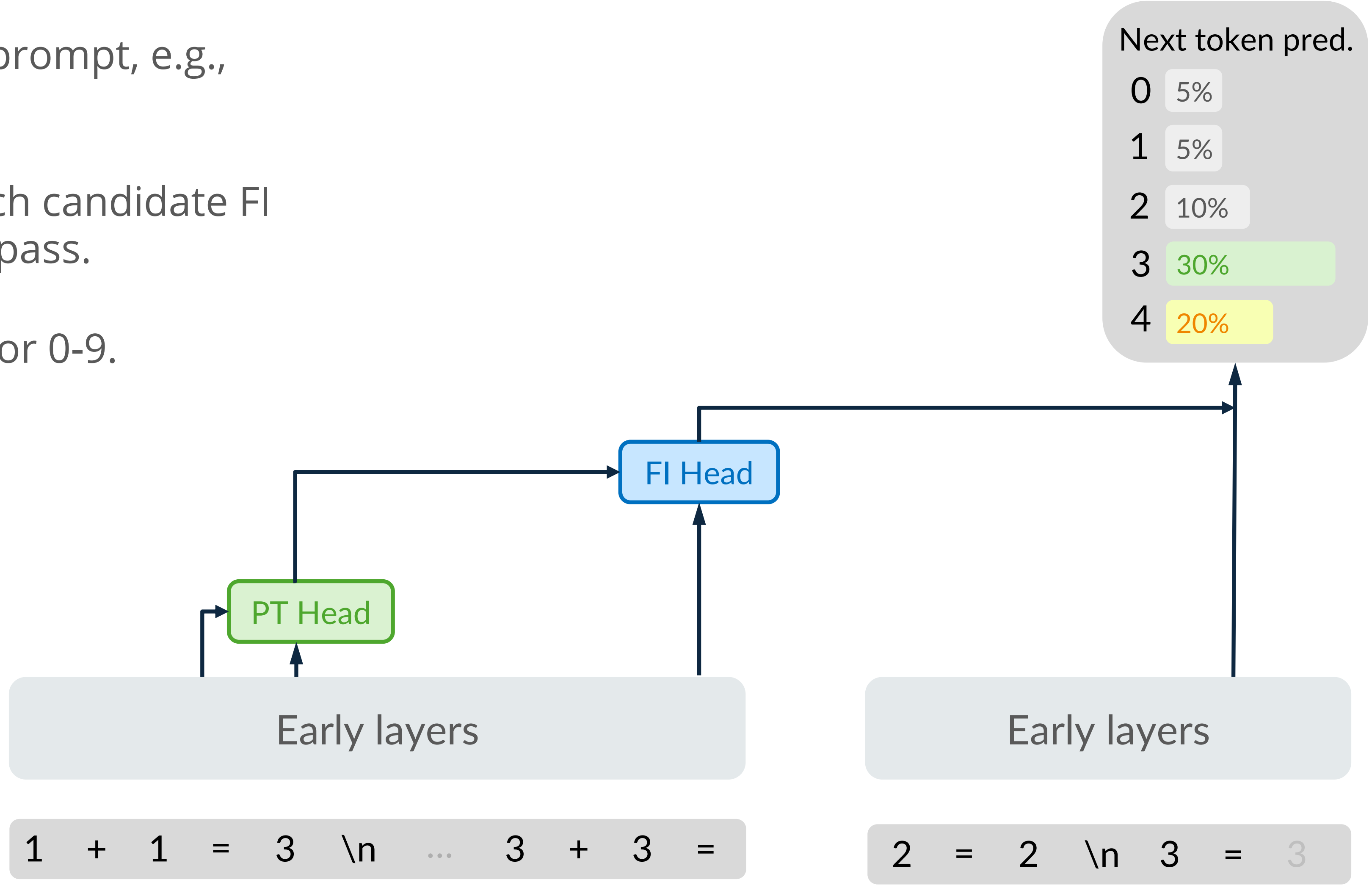
Investigating Each Candidate FI Head

- We run the LM on a naive prompt, e.g., 2=2, 3=?
- We patch the output of each candidate FI head to the naive forward pass.

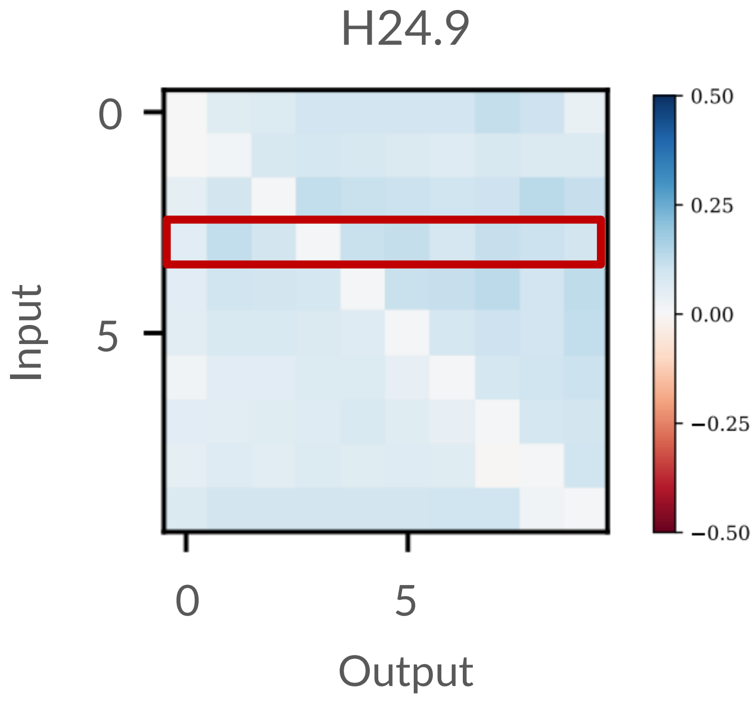


Investigating Each Candidate FI Head

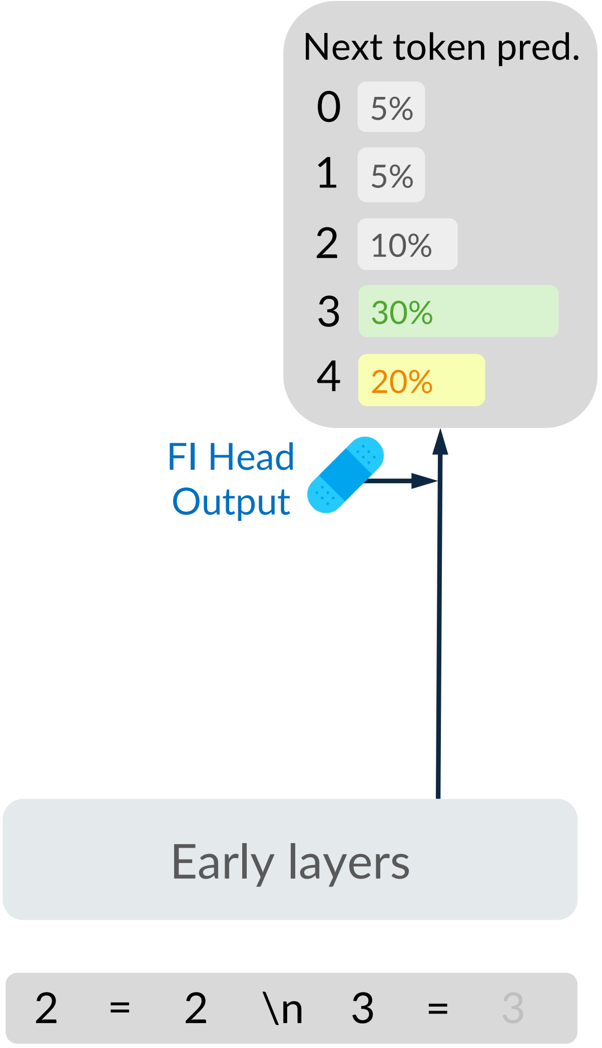
- We run the LM on a naive prompt, e.g., 2=2, 3=?
- We patch the output of each candidate FI head to the naive forward pass.
- We track the logit change for 0-9.



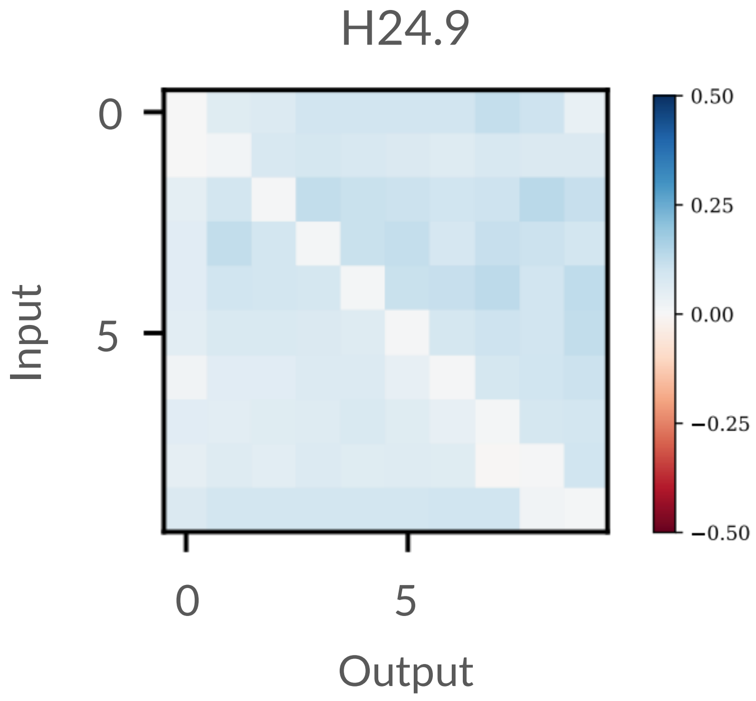
Investigating Each Candidate FI Head



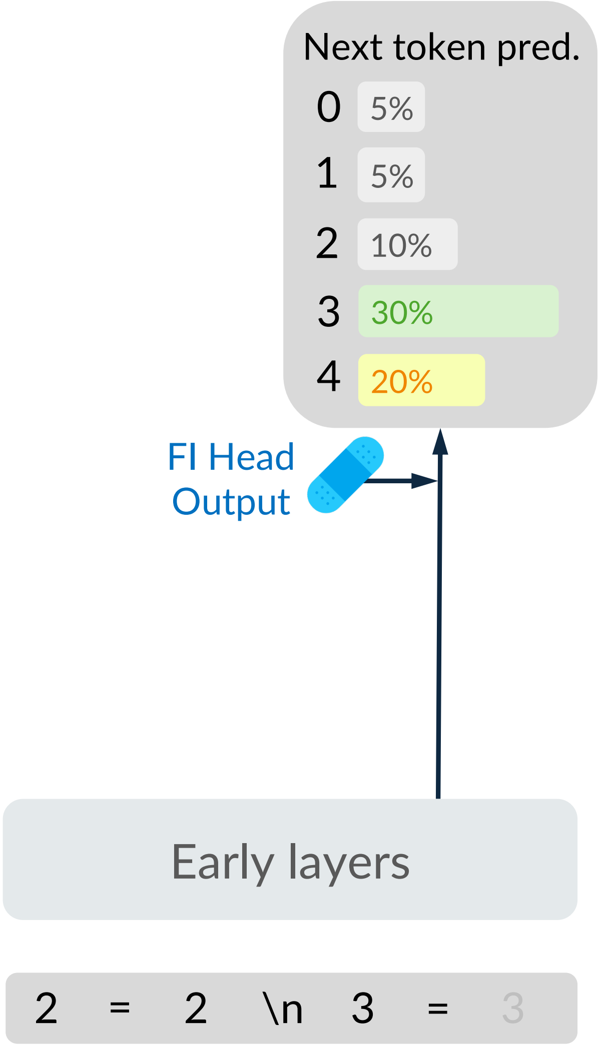
When the input is 3,
All tokens other than 3 are promoted. 3 is suppressed.



Investigating Each Candidate FI Head

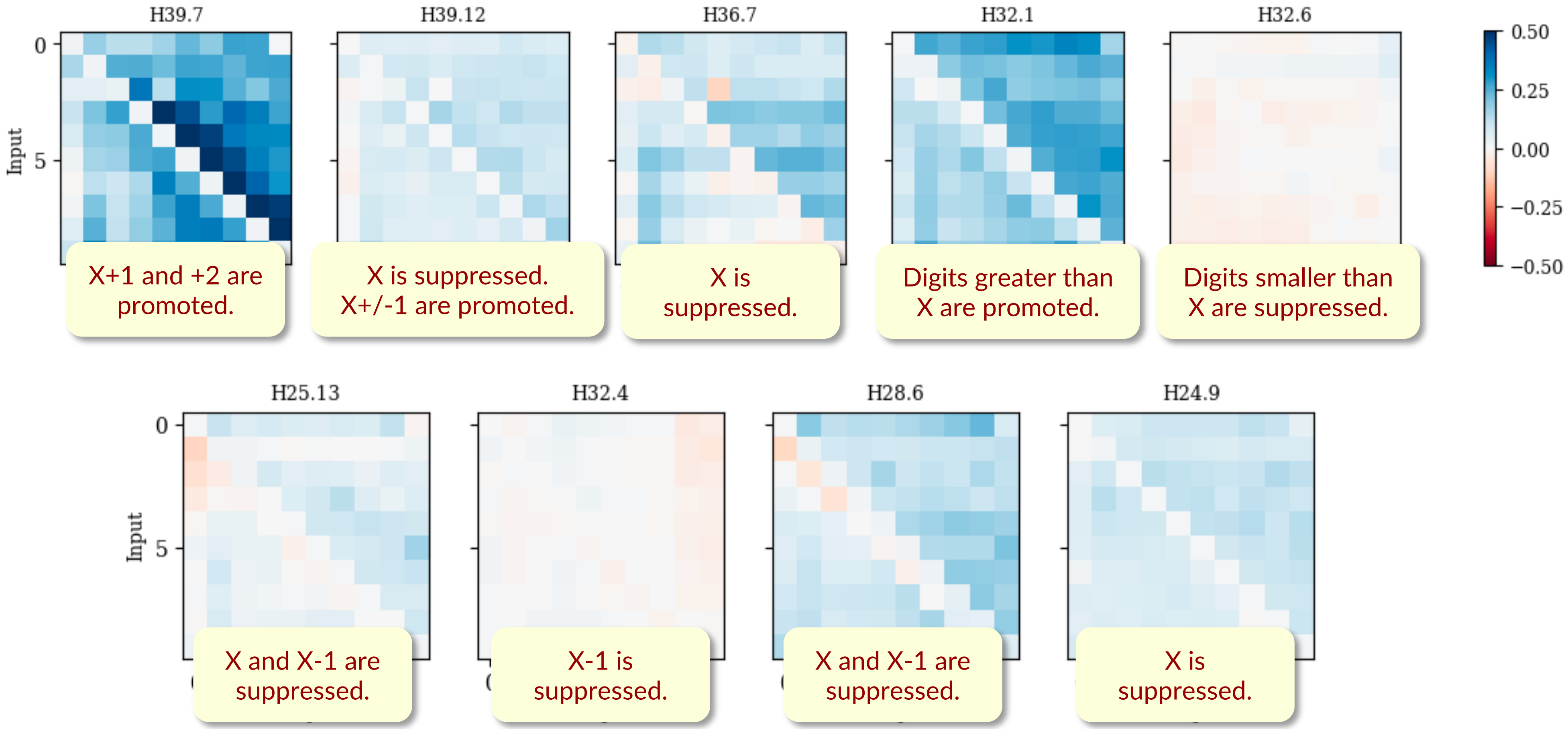


When input is X,
X is suppressed.



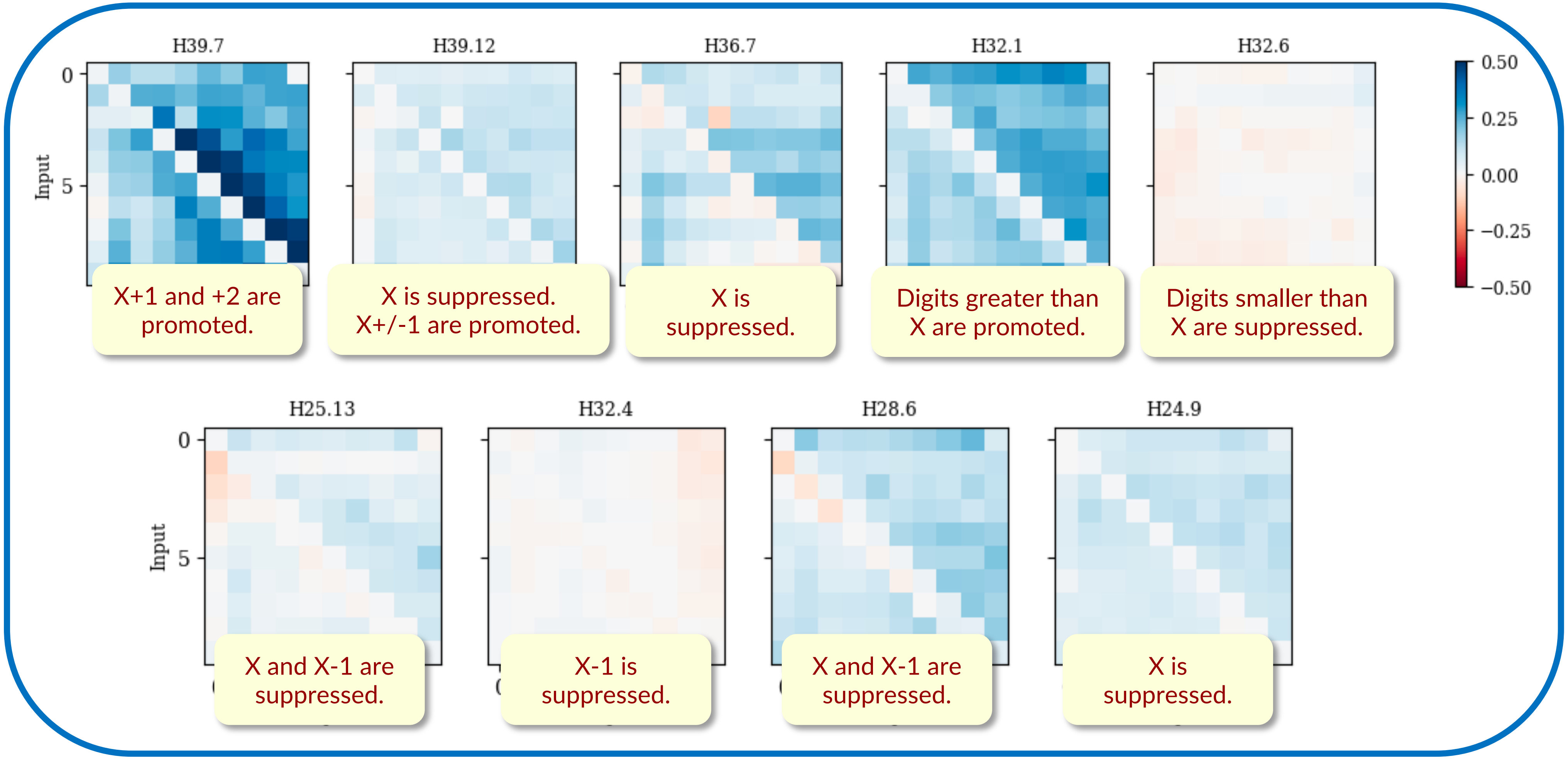
Investigating Each Candidate FI Head

When the input is X

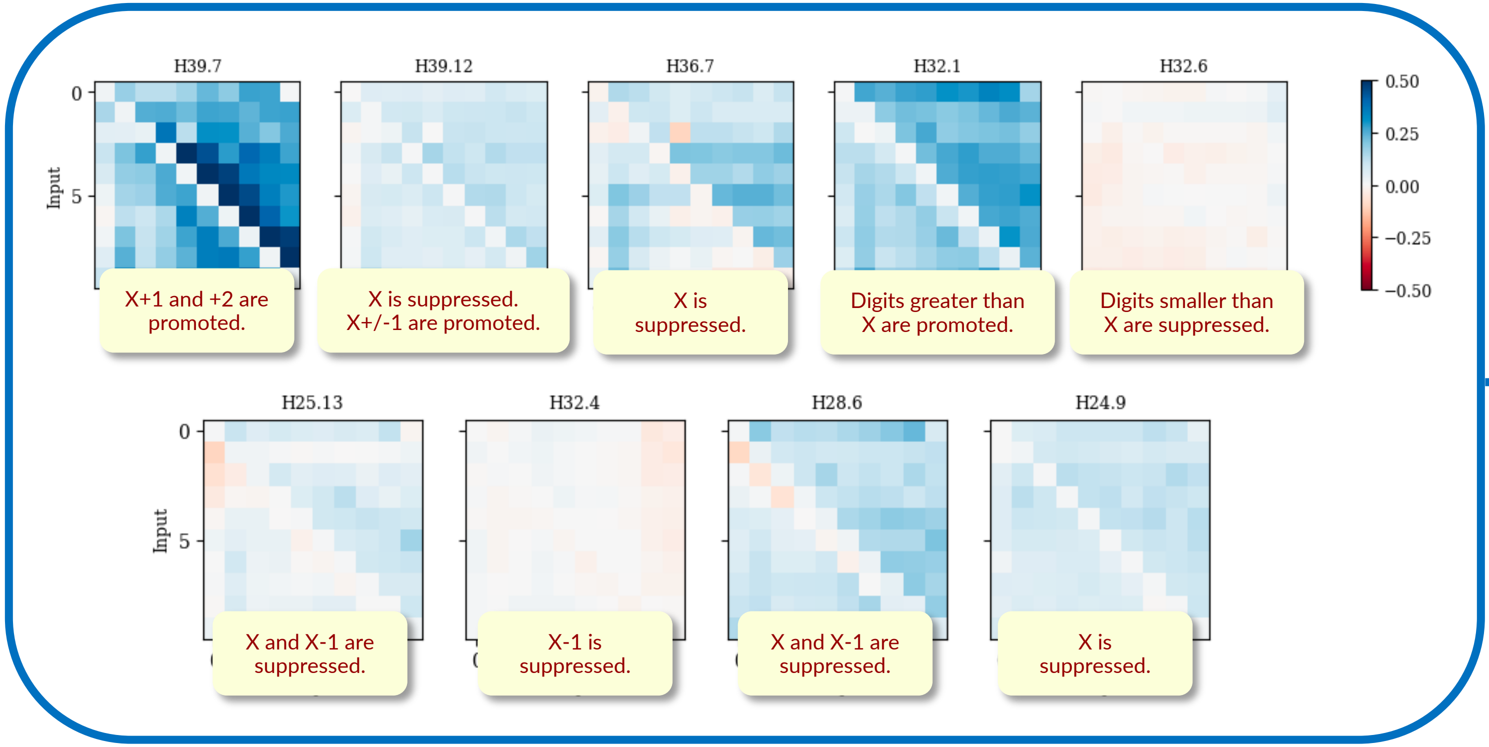


Investigating Each Candidate FI Head

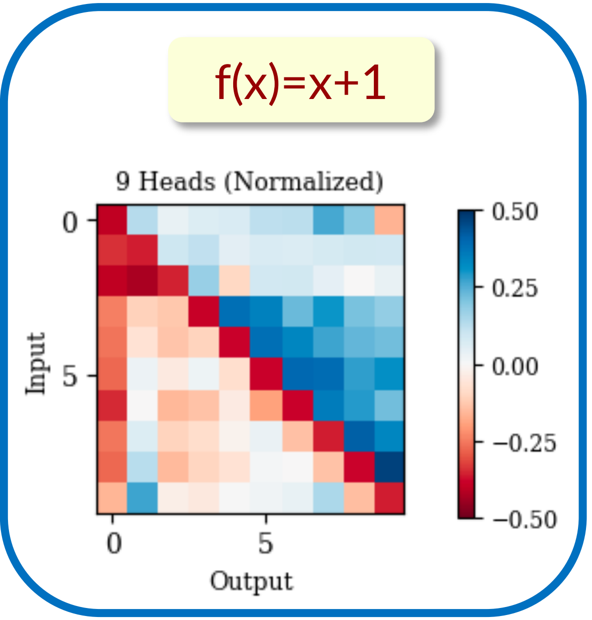
When the input is X



Finding 2: FI Heads Work Collaboratively!

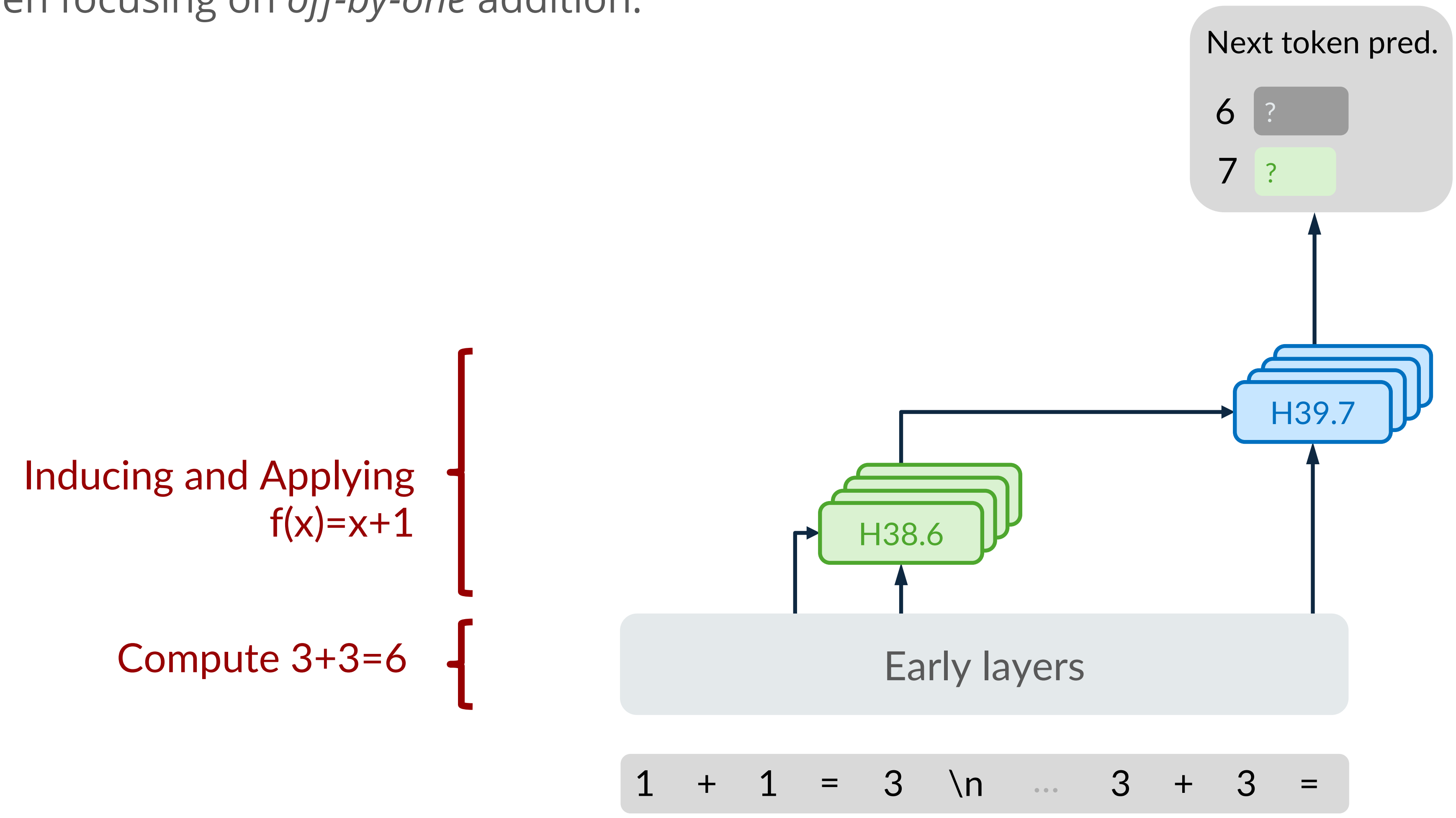


aggregate



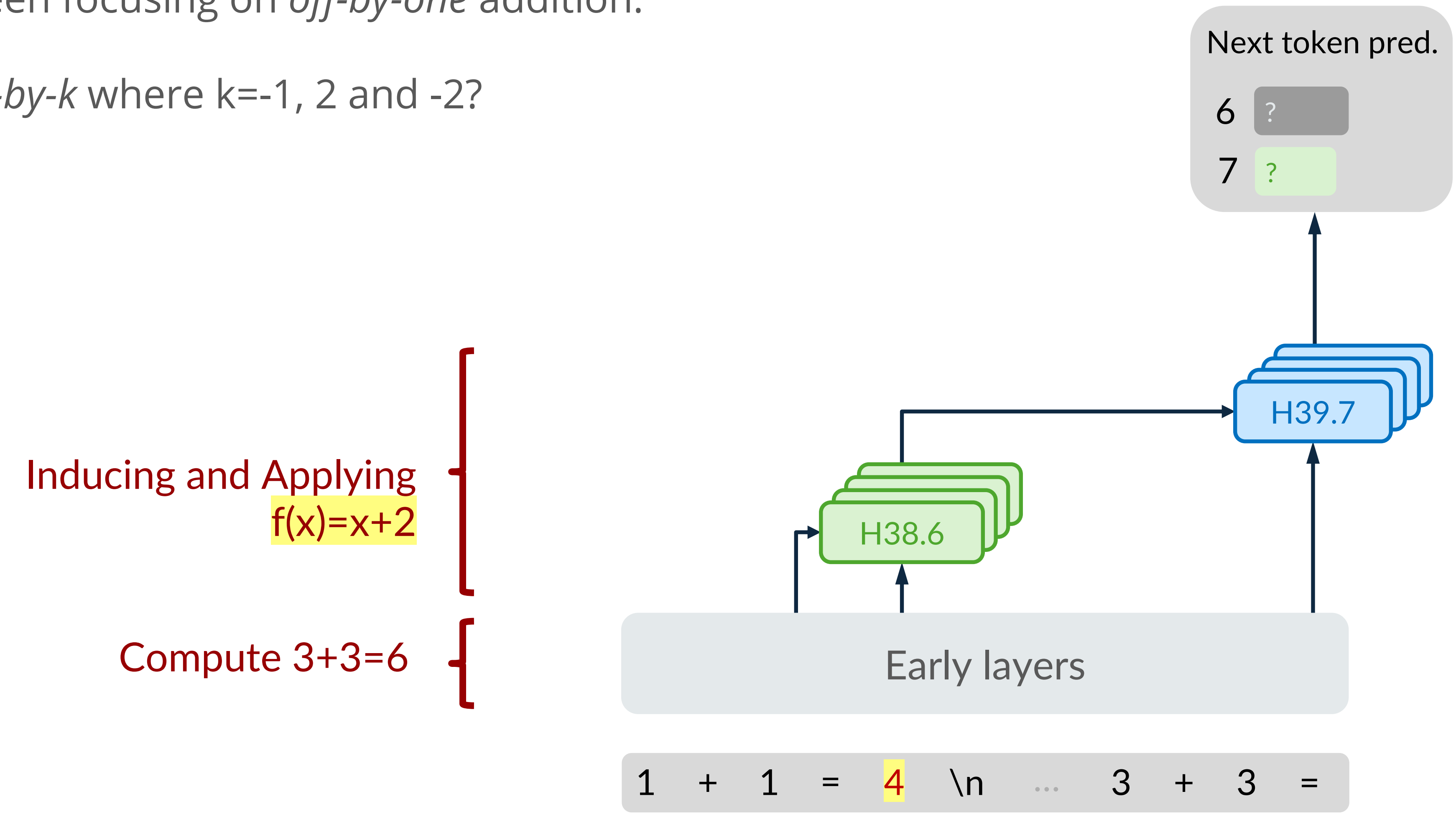
From Off-by-one to Off-by-k Addition

- So far, we've been focusing on *off-by-one* addition.



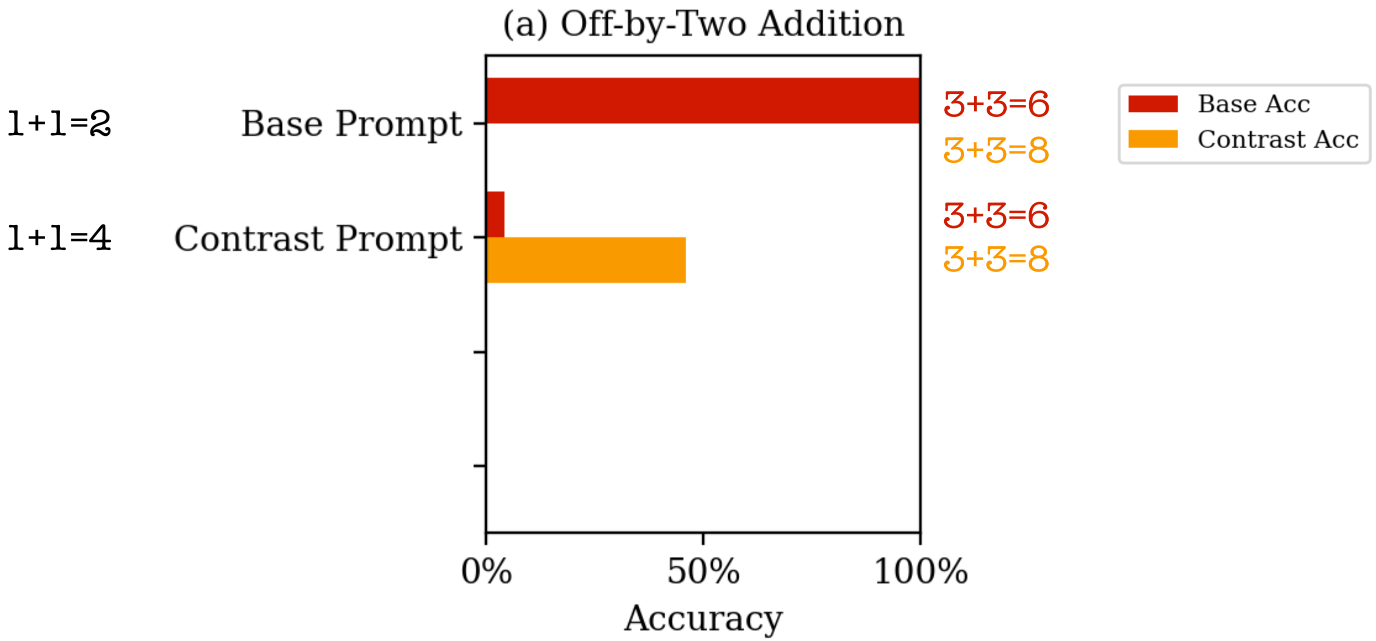
From Off-by-one to Off-by-k Addition

- So far, we've been focusing on *off-by-one* addition.
- What about *off-by-k* where k=-1, 2 and -2?



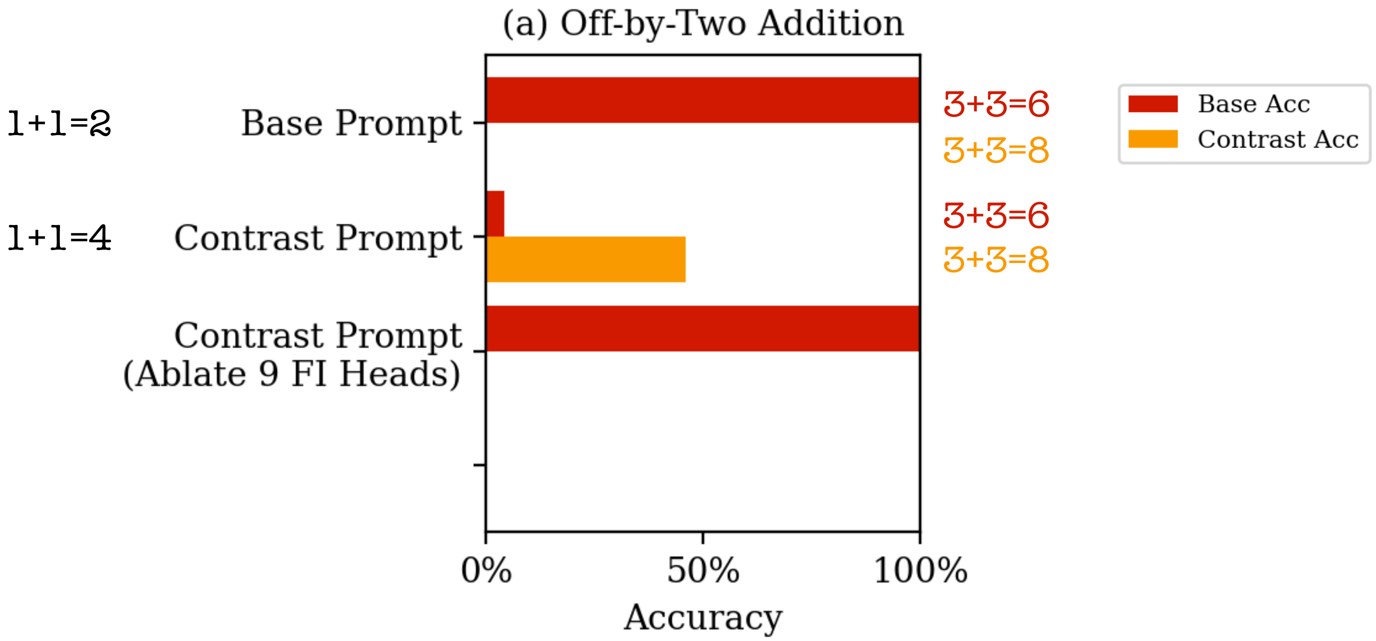
From Off-by-one to Off-by-k Addition

- We investigate this with head ablation experiments.



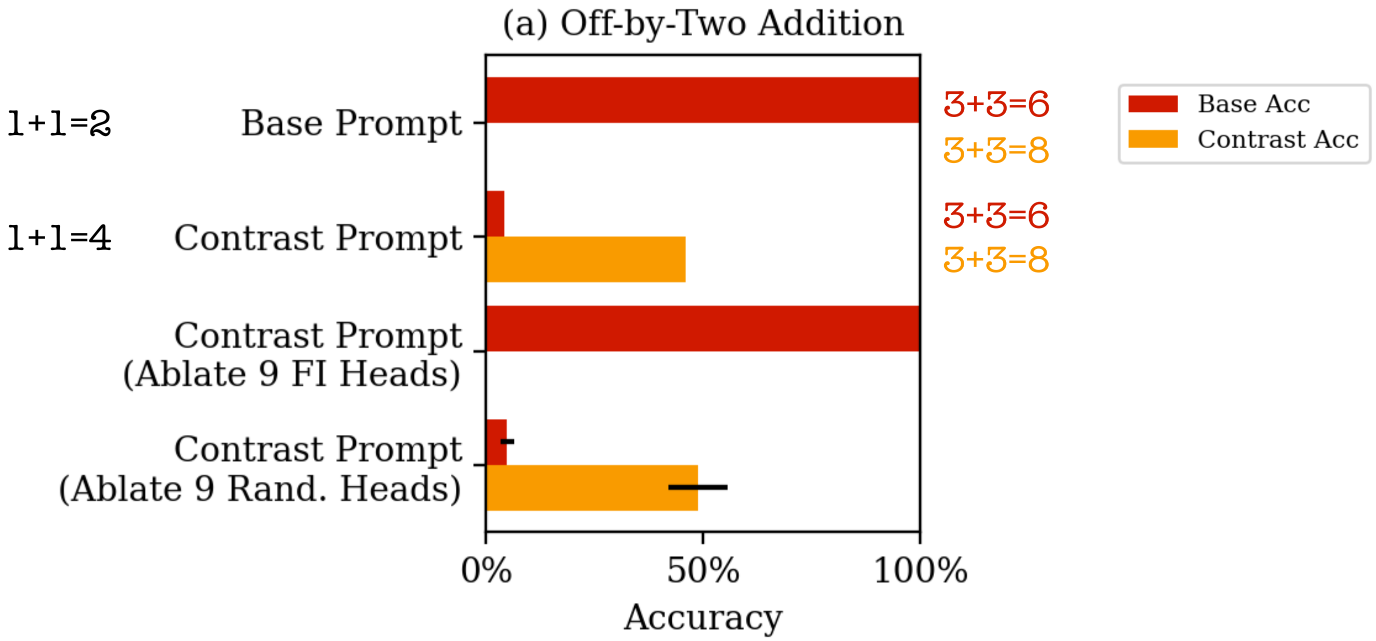
From Off-by-one to Off-by-k Addition

- We investigate this with head ablation experiments.



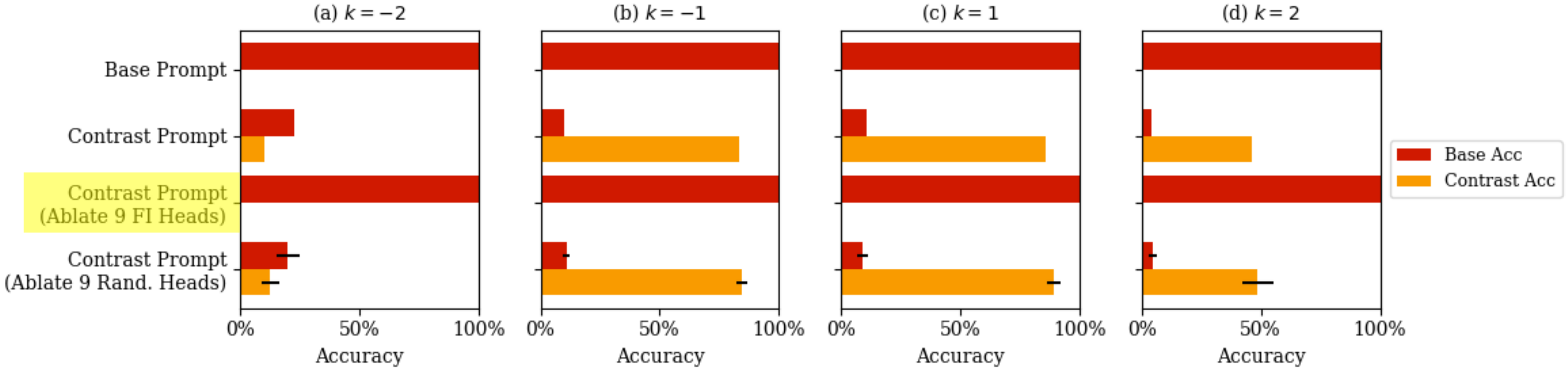
From Off-by-one to Off-by-k Addition

- We investigate this with head ablation experiments.



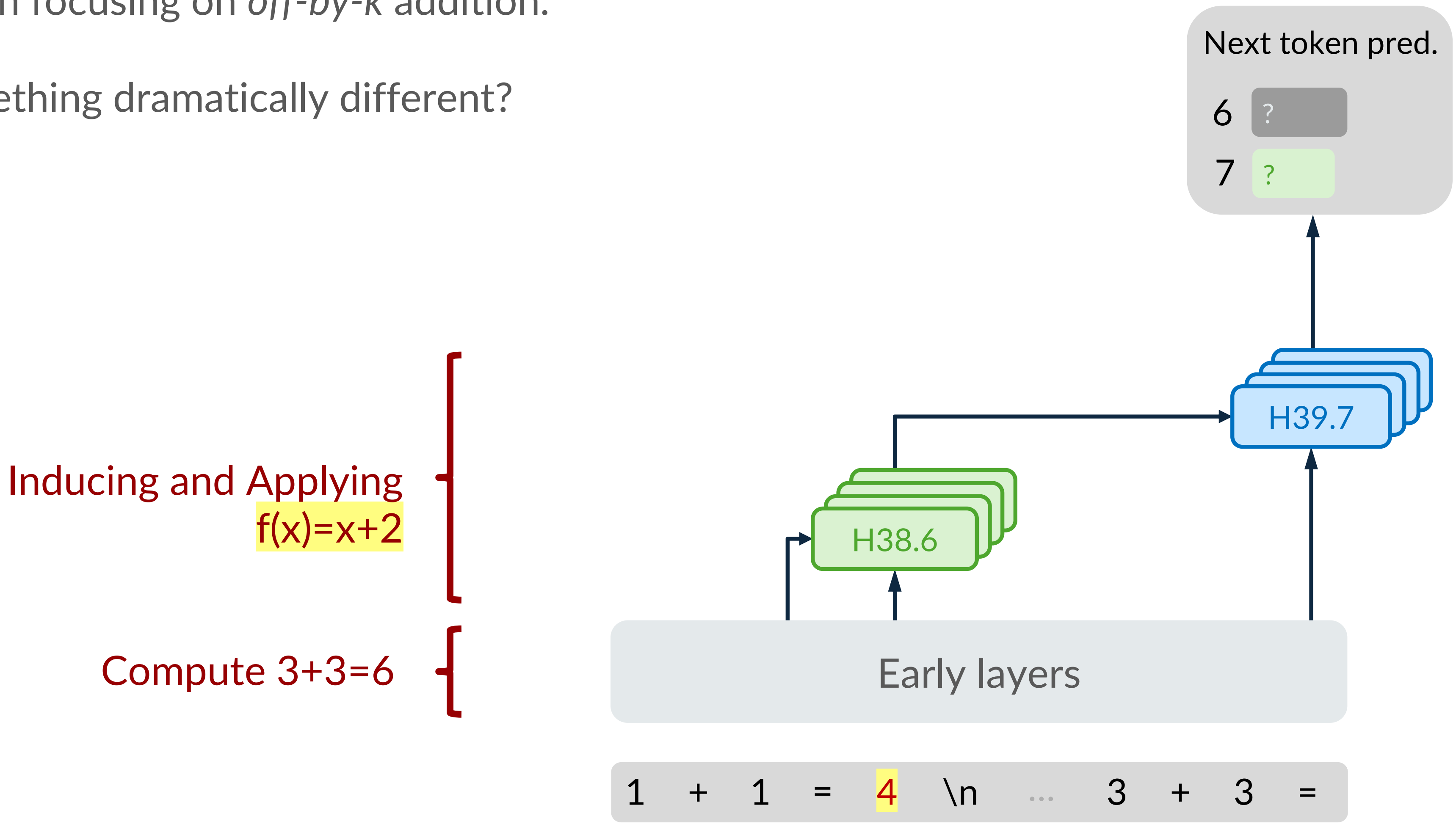
From Off-by-one to Off-by-k Addition

- This observation is consistent with different offsets.
- When FI heads are present, the model performs off-by-k addition non-trivially.
- When FI heads are ablated, the model performs standard addition instead.



From Off-by-k Addition to More

- So far, we've been focusing on *off-by-k* addition.
- What about something dramatically different?

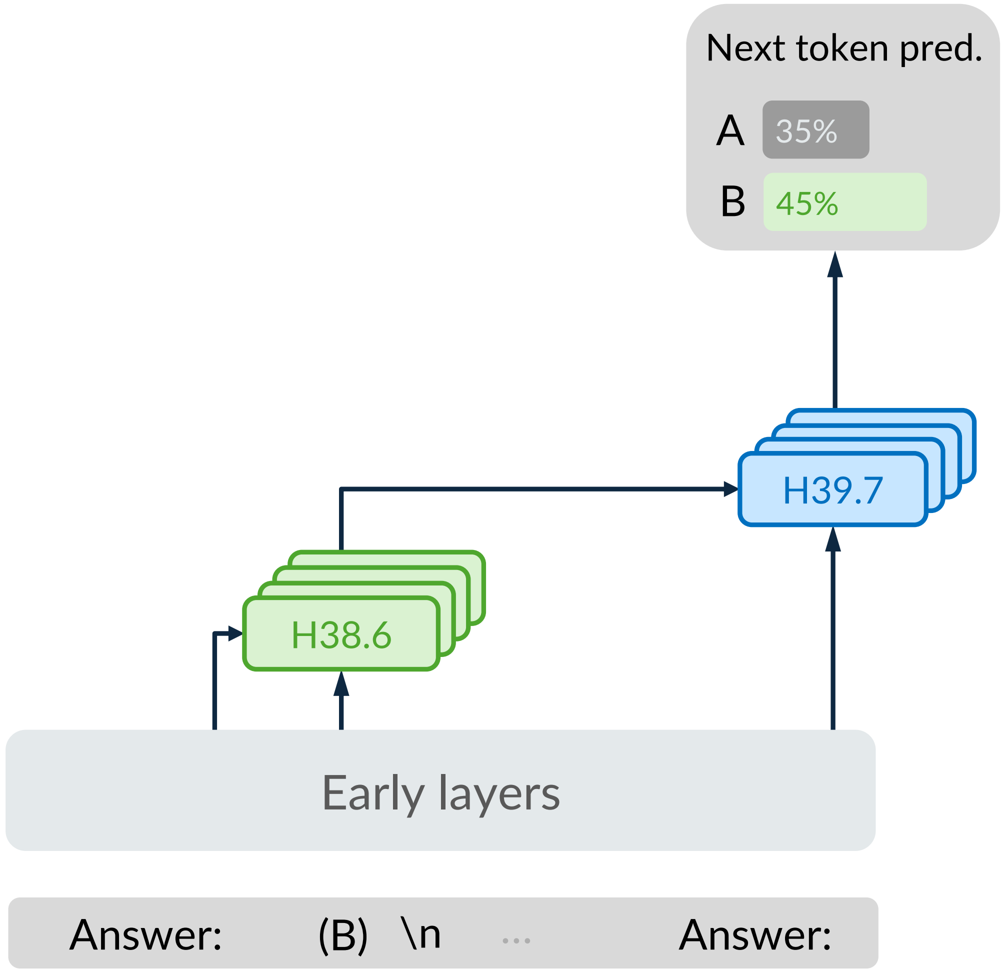


From Off-by-k Addition to More

- So far, we've been focusing on *off-by-k* addition.
- What about something dramatically different?

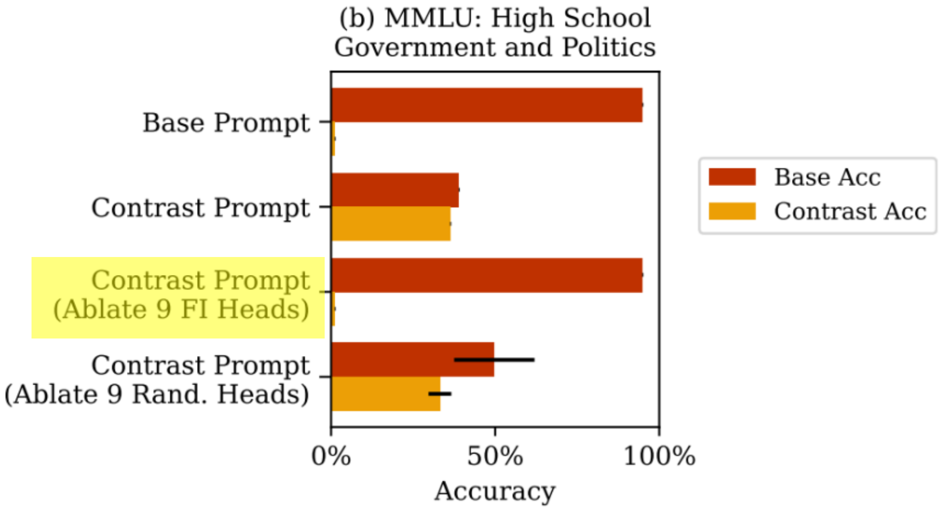
Shifting the answer by one
(A→B, B→C, ...)

Multiple-choice QA



Finding 3: Function Induction Helps Task Generalization

- The same set of FI heads are reused in Shifted MMLU.
    - When FI heads are present, the model performs Shift-by-one MMLU.
    - When FI heads are ablated, the model performs Standard MMLU.



Finding 3: Function Induction Helps Task Generalization

- We tried more tasks! The same set of FI heads are reused in Caesar Cipher and Base-k Addition.
- We took a closer look at base-8 addition.

Base-10	25+16=41\n60+16=76\n13+35=48\n52+17= **69**
Base-8	25+16=43\n60+16=76\n13+35=50\n52+17= **71**

Finding 3: Function Induction Helps Task Generalization

- We tried more tasks! The same set of FI heads are reused in Caesar Cipher and Base-k Addition.
- We took a closer look at base-8 addition.

Base-10	25+16=41\n60+16=76\n13+35=48\n52+17= 69
Base-8	25+16=43\n60+16=76\n13+35=50\n52+17= 71

Case 1
Base-10 and base-8 answers are the same.

Finding 3: Function Induction Helps Task Generalization

- We tried more tasks! The same set of FI heads are reused in Caesar Cipher and Base-k Addition.
- We took a closer look at base-8 addition.

Case 2
Unit digit `c[0]+=2`
Eights digit `c[1]+=1`

Base-10	25+16=41\n	60+16=76\n	13+35=48\n	52+17=	69
Base-8	25+16=43\n	60+16=76\n	13+35=50\n	52+17=	71

Case 1
Base-10 and base-8 answers are the same.

Finding 3: Function Induction Helps Task Generalization

- We tried more tasks! The same set of FI heads are reused in Caesar Cipher and Base-k Addition.
- We took a closer look at base-8 addition.

Case 3
Unit digit `c[0]+=2`

Case 2
Unit digit `c[0]+=2`
Eights digit `c[1]+=1`

Base-10	25+16=41	60+16=76	13+35=48	52+17= **69**
Base-8	25+16=43	60+16=76	13+35=50	52+17= **71**

Case 1
Base-10 and base-8
answers are the same.

Finding 3: Function Induction Helps Task Generalization

Case 3
Unit digit `c[0]+=2`

Case 2
Unit digit `c[0]+=2`
Eights digit `c[1]+=1`

Base-10	25+16=41	60+16=76	13+35=48	52+17= **69**
Base-8	25+16=43	60+16=76	13+35=50	52+17= **71**

Case 1
Base-10 and base-8
answers are the same.

- We generate 100 test examples for each category.

- The model uses FI heads to apply +1 and +2;
- But does not always apply them under the right conditions.

Summary: Function Induction

- We interpret how models perform **off-by-one addition**.

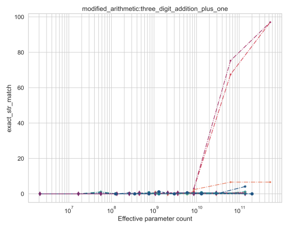
- LMs implement a complex **function induction** mechanism.
  - Leveling up from token-level copy-paste induction.
- Function induction heads work **collaboratively**.
  - Each send out a fraction of “+1”, which adds up to the whole “+1” function.
- The function induction mechanism **helps task-level generalization** broadly.
  - Components in off-by-one addition are reused in off-by-k addition, shifted MMLU, base-k addition ...

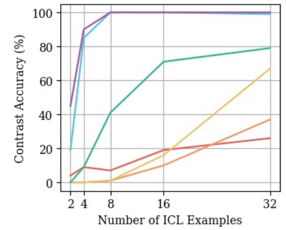
Research Question: Revisited

Llama 3

1+1=3
2+2=5

PaLM





How do LMs perform off-by-one addition?

Can models learn unseen tasks with ICL?

How do LMs handle misinformation?

Why do emergent abilities emerge?

Research Question: Revisited

Can models learn unseen tasks with ICL?

- Speculation
  - If an unseen task can be viewed as a seen task + a simple function.
  - The language model may be able to compose them together via in-context learning.

Unseen Task
Off-by-one Addition

Task: $f(g(a,b))=a+b+1$

Subtask 1: $g(a,b)=a+b$ ← seen task

Subtask 2: $f(x)=x+1$ ← function

Research Question: Revisited

How do LMs handle misinformation?

- Speculation
  - Models (investigated in this work) tend to not only follow 1+1=3, but also generalize it to 2+2=5.

Research Question: Revisited

Why do emergent abilities emerge?

- Speculation
  - For two-step tasks, early layers in the LM perform step 1, and late layers perform step 2.
  - Smaller models may not have enough layers (capacity) to develop this sequential structure.

Future directions

How does the function induction mechanism form during pre-training?

- Speculation
  - **FI heads** may evolve from induction heads (Olsson et al., 2022) and function vector heads (Todd et al., 2023).
- It will be interesting to
  - Reproduce our results using an open model (e.g., OLMo 2);
  - Examine the mechanism with intermediate checkpoints;
  - Conduct a study similar to Yin et al., 2025.

Which Attention Heads Matter for In-Context Learning?

Kayo Yin [1] Jacob Steinhardt [1]

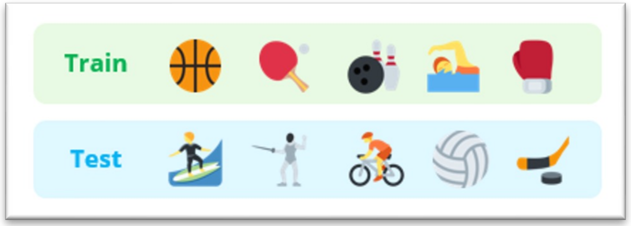
Future directions

How is function induction reused in naturally-occurring text?

- Our work is currently limited to synthetic tasks and algorithmic tasks.
- It will be interesting to
  - Disable the function induction mechanism in the model;
  - Search for sentences where it has maximal impact.

My PhD Journey

- During my PhD, I worked on **cross-task generalization abilities of large language models**.
  - **Measuring** cross-task generalization by training language models across diverse NLP tasks.
  - **Predicting** cross-task generalization through data-driven modeling and analysis.
  - **Deconstructing** cross-task generalization by dissecting model internals and uncovering underlying mechanisms.

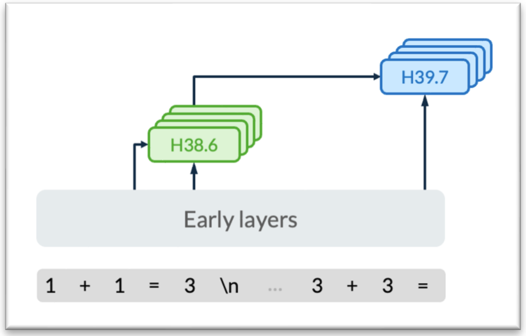


CrossFit
(EMNLP 2021)

Model Family	# param	Task	# shot	Perf.
GPT-3	3B	strategy_qa	0	0.48
BIG-G T=1	8B	elementary_math	3	0.19
PaLM	64B	code_line_desc	2	0.23
GPT-3	6B	elementary_math	1	?

How *predictable* are LLM capabilities?

BIG-bench Analysis
(EMNLP Findings 2023)



Function Induction
(This Talk; In Submission, 2025)

Thank you!