Student Suggested Topics:

Mechanistic Interpretability and Mixture of Experts

CS 6120 Natural Language Processing
Northeastern University

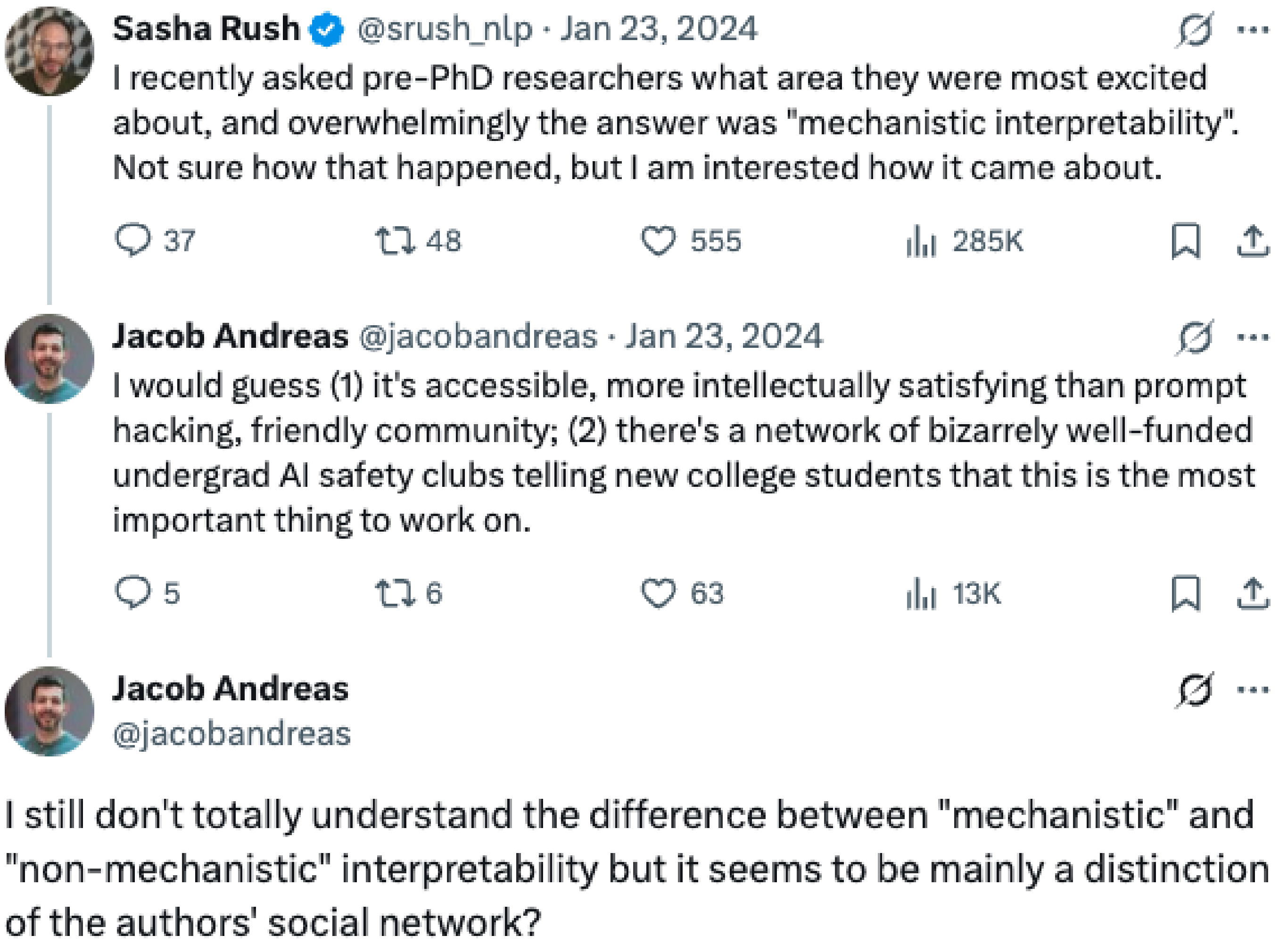
Si Wu

Logistics

- Coding assignment 5 is out. It’s due the same week as the final report.
- I still have office hours tomorrow, but no office hours next week due to conference travel.
- The instruction on final report will be out before Thanksgiving Day.
- There will be extra time at the end of this lecture, where you can ask me anything about your project, or if you need any last-minute help.

Mechanistic Interpretability

What is *Mechanistic* Interpretability ?



Sasha Rush @srush_nlp · Jan 23, 2024

I recently asked pre-PhD researchers what area they were most excited about, and overwhelmingly the answer was "mechanistic interpretability". Not sure how that happened, but I am interested how it came about.

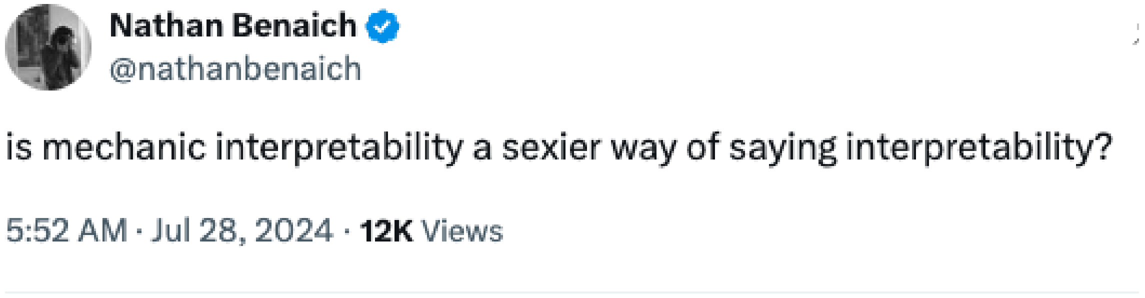
Jacob Andreas @jacobandreas · Jan 23, 2024

I would guess (1) it's accessible, more intellectually satisfying than prompt hacking, friendly community; (2) there's a network of bizarrely well-funded undergrad AI safety clubs telling new college students that this is the most important thing to work on.

Jacob Andreas
@jacobandreas

I still don't totally understand the difference between "mechanistic" and "non-mechanistic" interpretability but it seems to be mainly a distinction of the authors' social network?

5:11 PM · Jan 23, 2024 · **10.9K** Views



Nathan Benaich
@nathanbenaich

is mechanic interpretability a sexier way of saying interpretability?

5:52 AM · Jul 28, 2024 · **12K** Views

So… What is mechanistic interpretability?

- Mechanistic interpretability: studying a model’s internals.
  - How information flows, how outputs are produced, etc.
- More concretely, one can look inside a model’s
  - Weights
  - Activations,
  - And circuits
- The goal is to identify meaningful patterns/mechanisms:
  - Specific (groups of) neurons for a particular features
  - Pathways/circuits that implement certain algorithms or behaviors like copying

“Mechanistic”

- Comparing to earlier interpretability work which mainly focuses on input-output explanations, mech interp aims to identify structures, circuits, or algorithms encoded in the models.
- Not just about whether the output is explainable (with features), but **HOW** was the output computed
- It’s all about finding the mechanism that produces a specific behavior

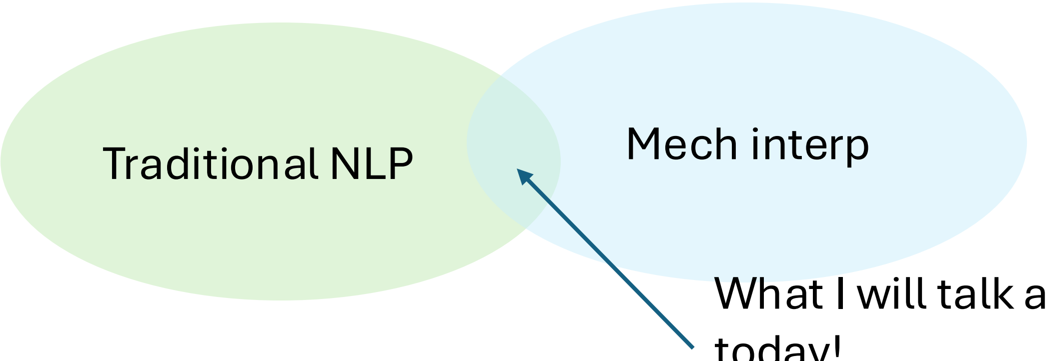
So… What is mechanistic interpretability?

- This area is becoming increasingly important since modern models, like transformers, are growing in scale and complexity.
- Recall transformers can easily have nearly 100 transformer blocks, which makes it harder to reason about certain behavior
- Meanwhile, so many attention heads and nonlinear interaction, also makes it harder to understand how a model arrives at its outputs.

It's hard to trust a model when we treat it as a black box!

So… What is mechanistic interpretability?

- As the models get larger, understanding their internal mechanisms becomes an increasingly important research topic.
- By definition, this area of research usually studies the model internal structure and computation, not about analyzing human language directly. However, since NLP replies on LLMs, understanding how these models work can inform future model design and development.



Traditional NLP

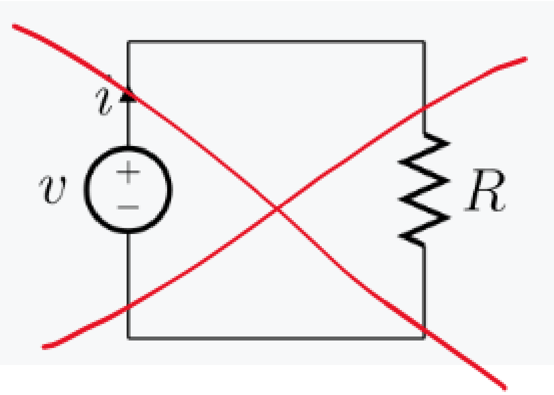
Mech interp

What I will talk about today!

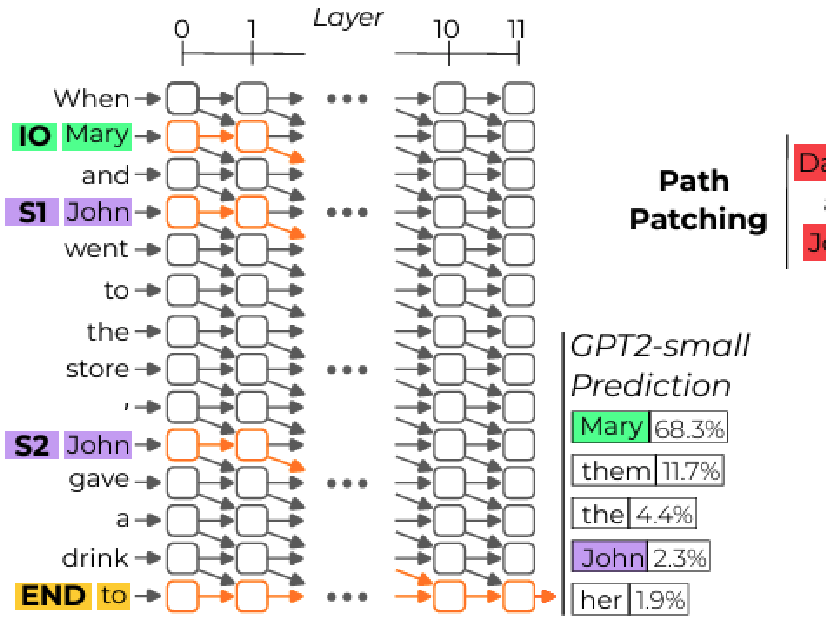
Topics in mechinterp

- We will introduce **circuits** today.
- Other topics but we won’t cover: probing (you did a homework on this!), induction heads, feature attribution, activation patching, superposition, sparse autoencoders, grokking, model editing, etc.
  - I put down some links for you to learn these topics if you are interested. See pointers in the later slides.
- But interpretability research in NLP is not limited to just mech interp! Older methods like dimensionality reduction, feature visualization could also be considered as interpretability work.

Circuits

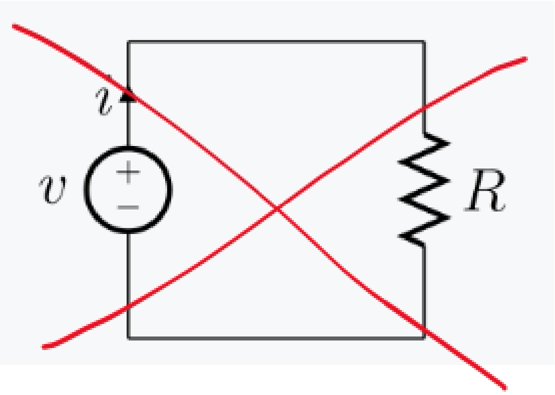


- What is a circuit? (Apparently not ones you learn in physics or electrical engineering...)
- Think of it more as a small and interpretable **subgraph** inside a network that implements specific behaviors

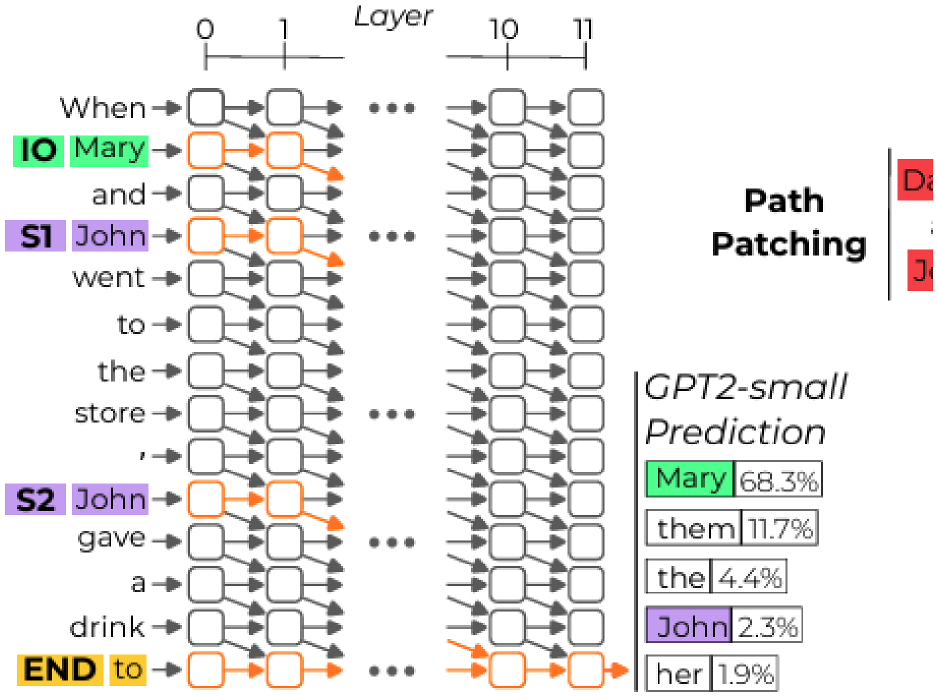


For example, in this figure from Wang et al., 2023:
The circuit is colored in orange where nodes are attention layers, and edges are interactions between layers.

Circuits



- Circuits connect specific neurons, layers, and attention heads. They together perform a task



For example, in this figure from Wang et al., 2023:

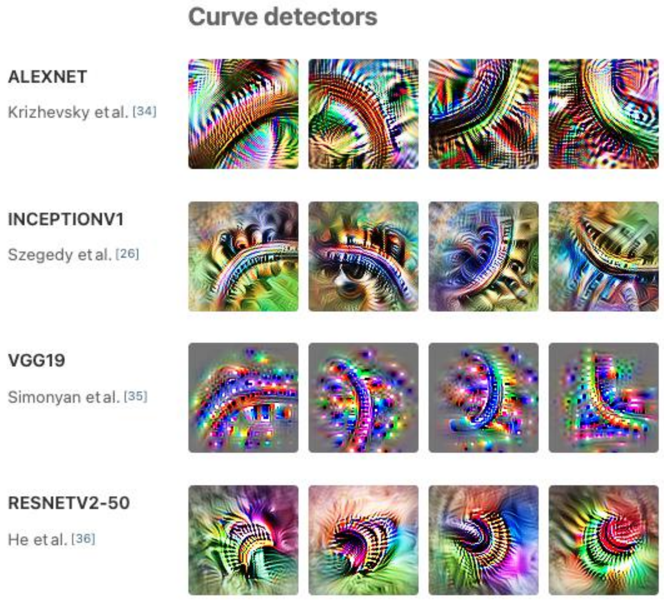
Here, the circuit performs a task called indirect object identification (IOI)

Why circuits matter?

- They reveals how model compute (for a specific task)
- It gives mechanistic explanation for specific behaviors like copying, induction, classification, etc.
- Knowing the circuit/path, it informs us to debug, steer, or verify models for safety or debiasing.

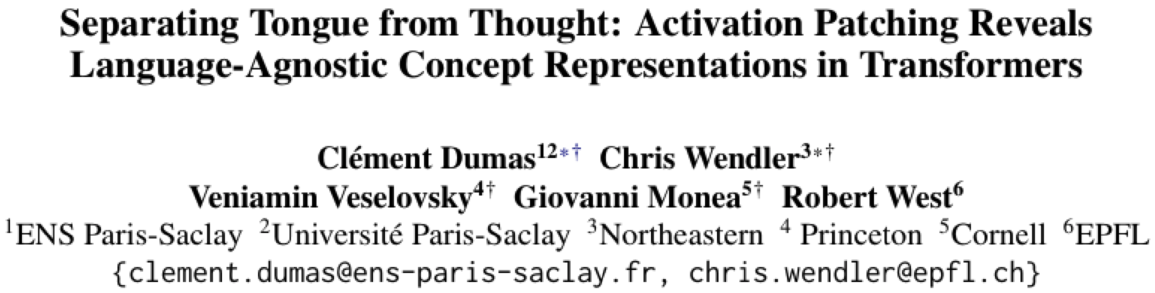
Classic circuit examples

- Induction head circuit (Olssen et al., 2022):
  - If sequence A-B appeared before, then if A appears again, we can predict B
  - Important for in-context learning.
- **Name-mover circuit (Indirect Object Identification (IOI) circuit),** Wang et al., 2023
- Early circuits from computer vision (InceptionV1)
  - Works from Olah et al.,
  - Interpretable features from vision models.
  - For example: curve circuits for detecting curves.



Methods for discovering circuits

- Activation patching / causal tracing
  - Identifying which model activations are most important for determining model behavior between two similar prompts that differ in a key detail
  - An effective variant: attribution patching
- **Path patching**
- Feature visualization
- Direct attribution / weight-level analysis
- …



Separating Tongue from Thought: Activation Patching Reveals Language-Agnostic Concept Representations in Transformers

Clément Dumas[12*†] **Chris Wendler**[3*†]
Veniamin Veselovsky[4†] **Giovanni Monea**[5†] **Robert West**[6]
[1]ENS Paris-Saclay [2]Université Paris-Saclay [3]Northeastern [4] Princeton [5]Cornell [6]EPFL
`{clement.dumas@ens-paris-saclay.fr, chris.wendler@epfl.ch}`

Example: circuits in transformer models

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`

Example: Indirect Object Identification circuit

- What's Indirect Object Identification (IOI)?

When Mary and John went to the store, John gave a drink to ___

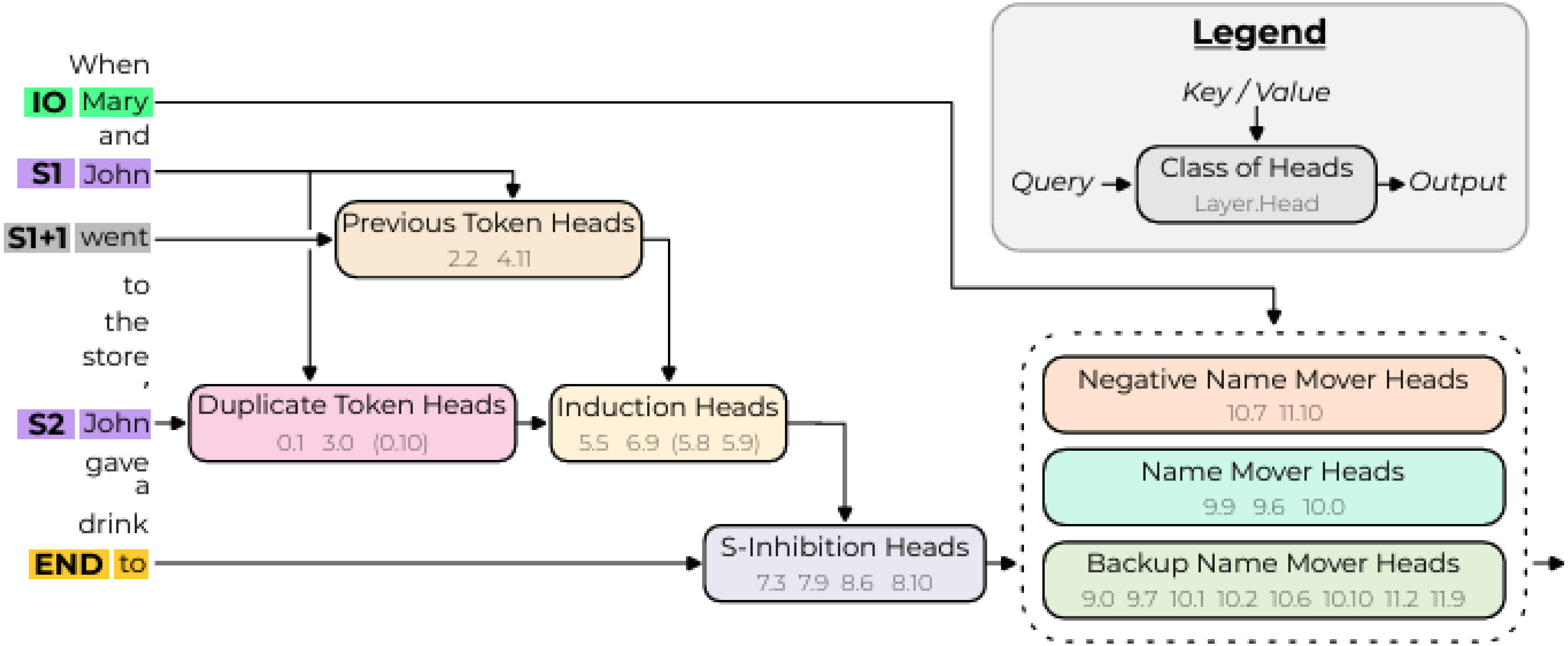
Answer: Mary

Indirect Object Identification circuit

- In this paper they use GPT2-small, which only has 12 transformer blocks, and 12 attention heads per attention layer
- Sort of the nature of many mech-interp work: they work on a very defined task. Here, the task is indirect object identification.
- This work explains end-to-end how the model works, which is rare.
- Here, in a circuit C is a subgraph of the model, responsible for some behavior: completing the IOI task.

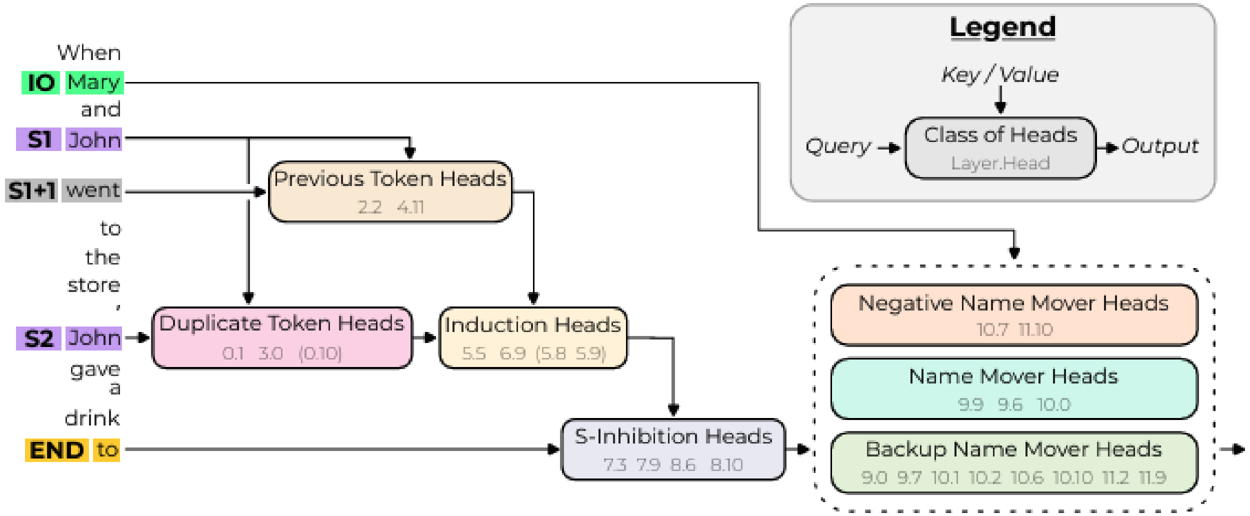
The circuit in GPT2-small that implements IOI

7 types of attention heads
(each in a different color here)



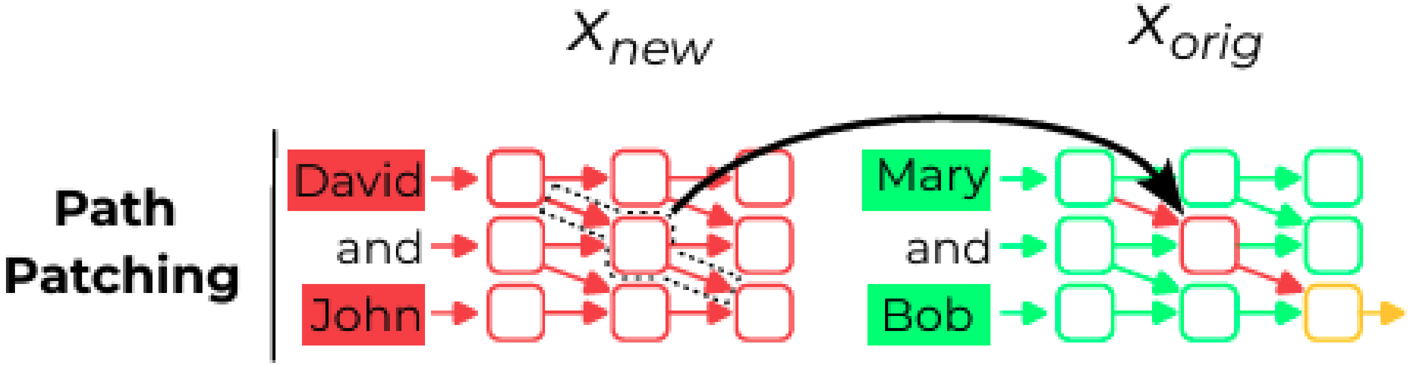
The circuit in GPT2-small that implements IOI

- **Duplicate token, induction token, previous token heads** (3 different types) identify previous names: John, Mary, John
- **S-inhibition heads** suppress attention paid to the duplicated name
- **Name mover heads** copy the remaining name
- **Negative name movers** write against the correct answer (!!)



Path patching

- Path patching replaces part of a model's forward pass with activations from a different input.
- By replacing edges, it can test the causal contribution of that node on the rest of the path.

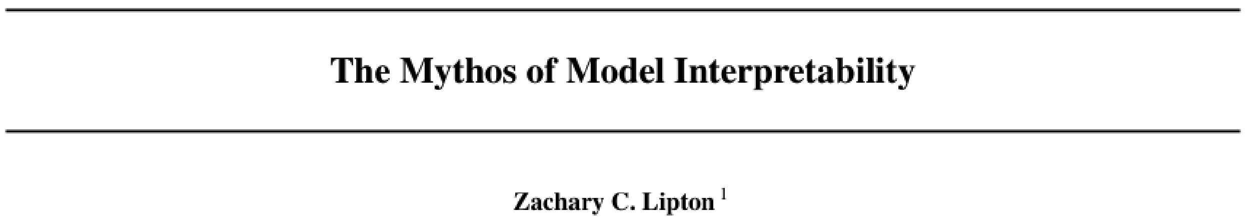


Limitations

- While in the paper, they were able to identify 7 types of heads, there are still components that they don’t understand.
  - Including the attention patterns of the S-Inhibition heads
- GPT-2 small is way smaller than state-of-the-art transformer models, can we scale this approach to much larger models?
  - For example, some findings don’t apply to GPT-2 Medium already: not all these heads attend to IO and S → more complex behavior than the Name Movers Heads in the GPT-2 small.

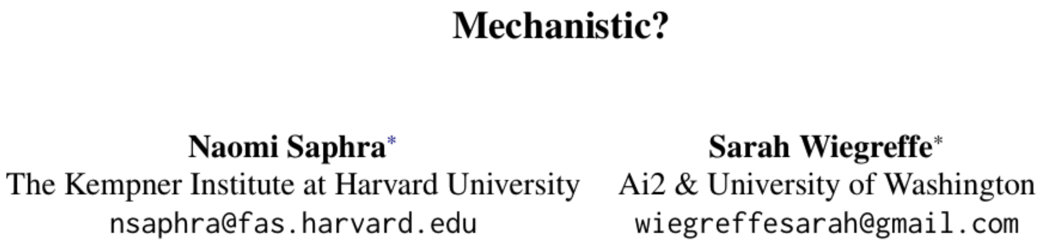
Pointers

- There are many blogs dedicated to mech interp. For example:
  - LessWrong
  - The AI Alignment Forum
  - Neel Nanda's A comprehensive mechanistic interpretability explainer & glossary
  - This fun and short paper by Saphra and Wiegreffe is also a good starting point: more on the history, the community, and where the name mechinterp is from.
  - Any many other blogs, for example.
  - On interpretability research in general (not just mech interp), this position paper from Zack Lipton.
  - Another survey paper: Rauker et al.



The Mythos of Model Interpretability

Zachary C. Lipton [1]



Mechanistic?

Naomi Saphra*
The Kempner Institute at Harvard University
`nsaphra@fas.harvard.edu`

Sarah Wiegreffe*
Ai2 & University of Washington
`wiegreffesarah@gmail.com`

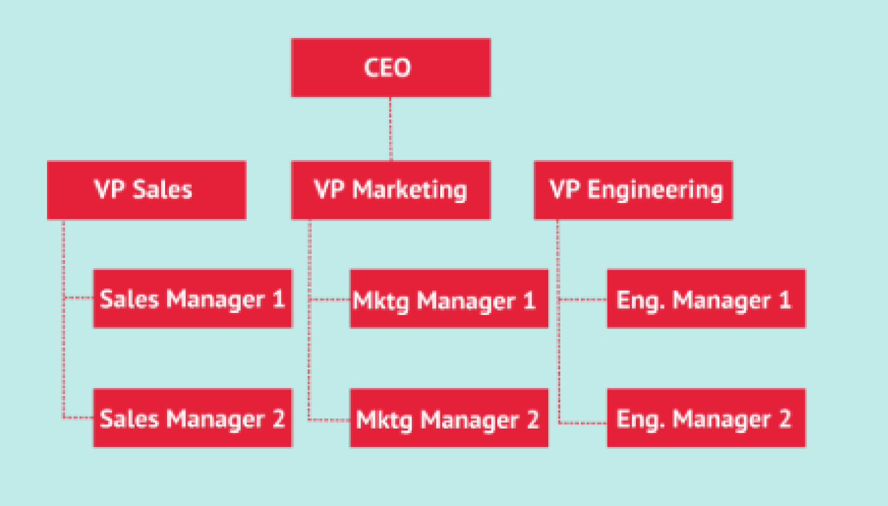
Mixture of Experts

Some slides are borrowed from Aran Nayebi and Matt Gormley

What is Mixture of Experts (MoE)

- We are strictly talking **in the context of transformer** models here!

- MoE has two main features:
  - Instead of dense feed-forward network layers, sparse MoE have **multiple experts**, where each expert is a neural network
    - Or each expert can be an MoE! → hierarchical MoE

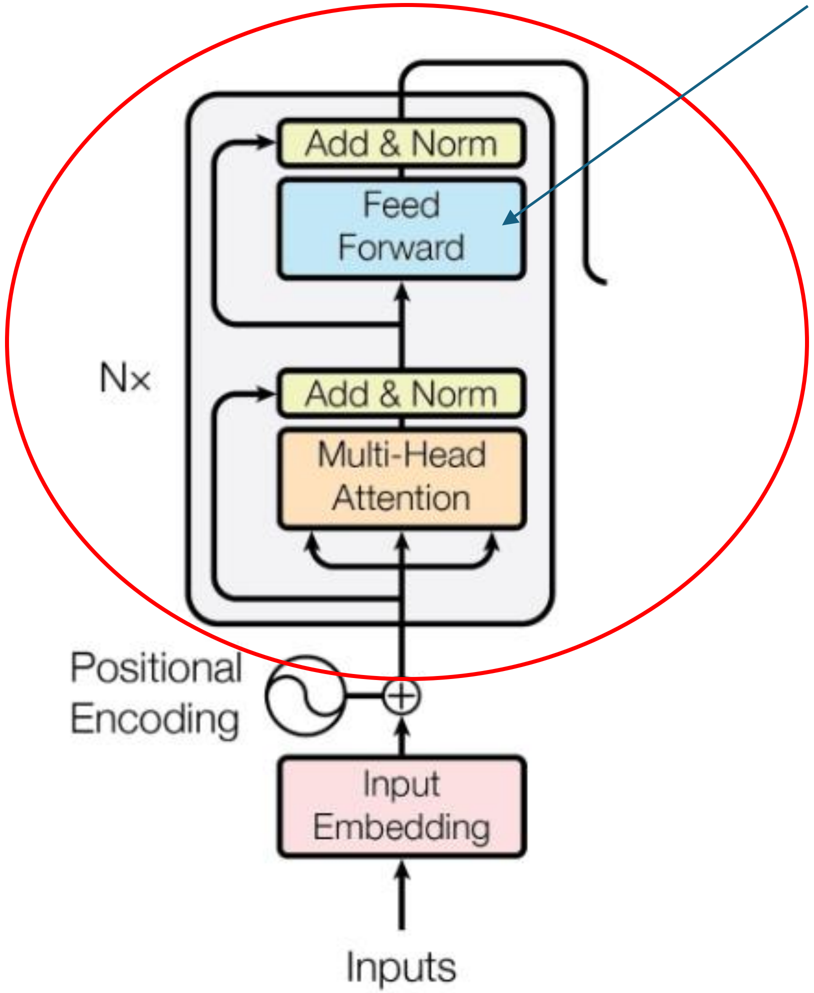


What is Mixture of Experts (MoE)

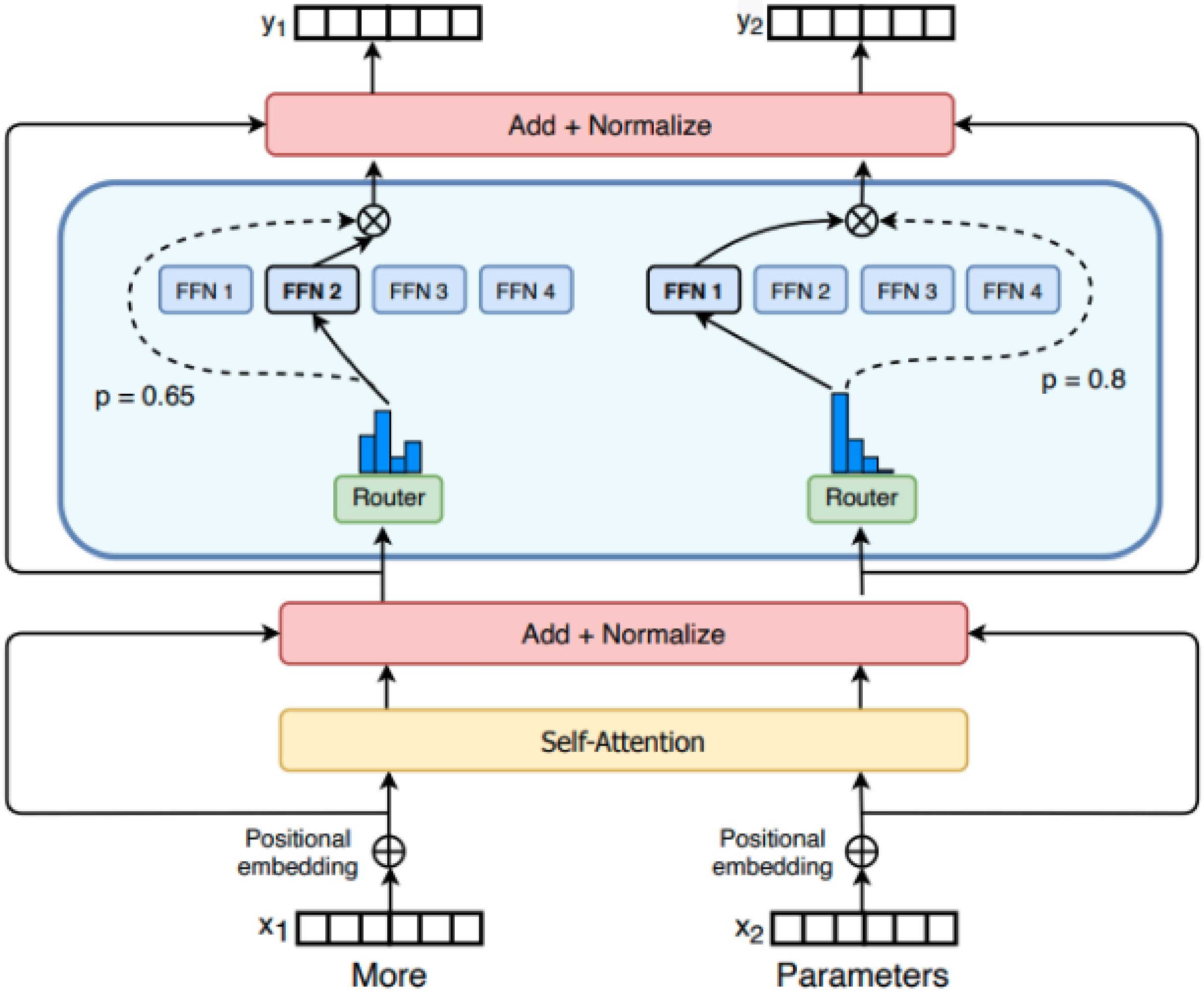
- We are strictly talking **in the context of transformer** models here!

- MoE has two main features:
  - Instead of dense feed-forward network layers, sparse MoE have **multiple experts**, where each expert is a neural network
    - Or each expert can be an MoE! → hierarchical MoE
  - **A gate network or router** that decides which tokens are sent to which expert

In a traditional transformer block:



MoE



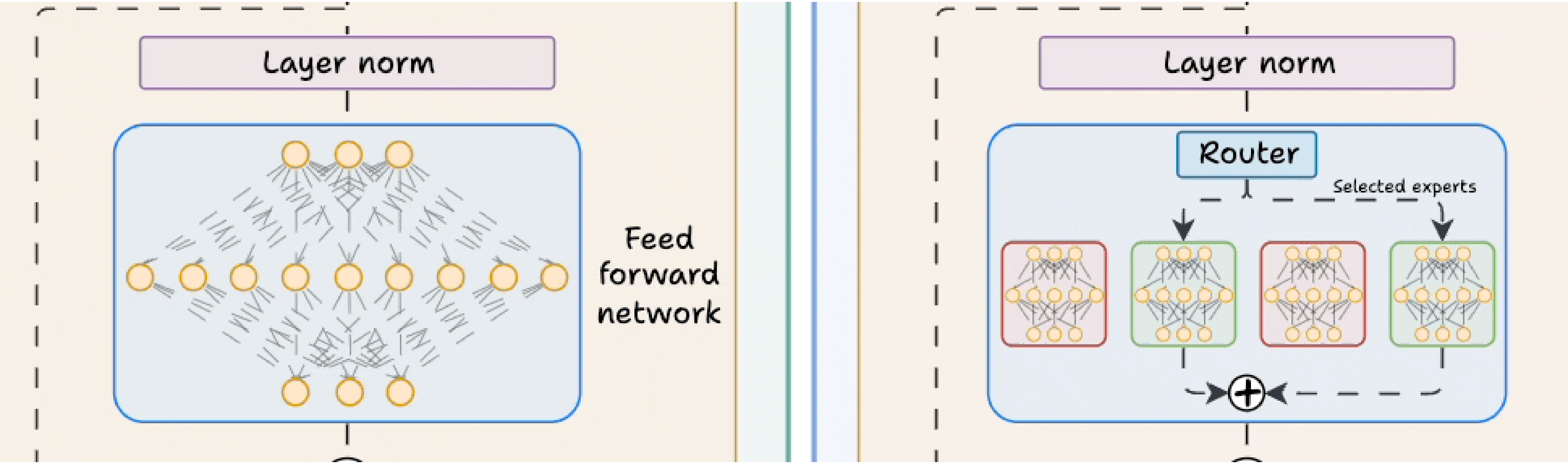


Illustration from
https://www.dailydoseofds.com/p/transformer-vs-mixture-of-experts-in-llms/

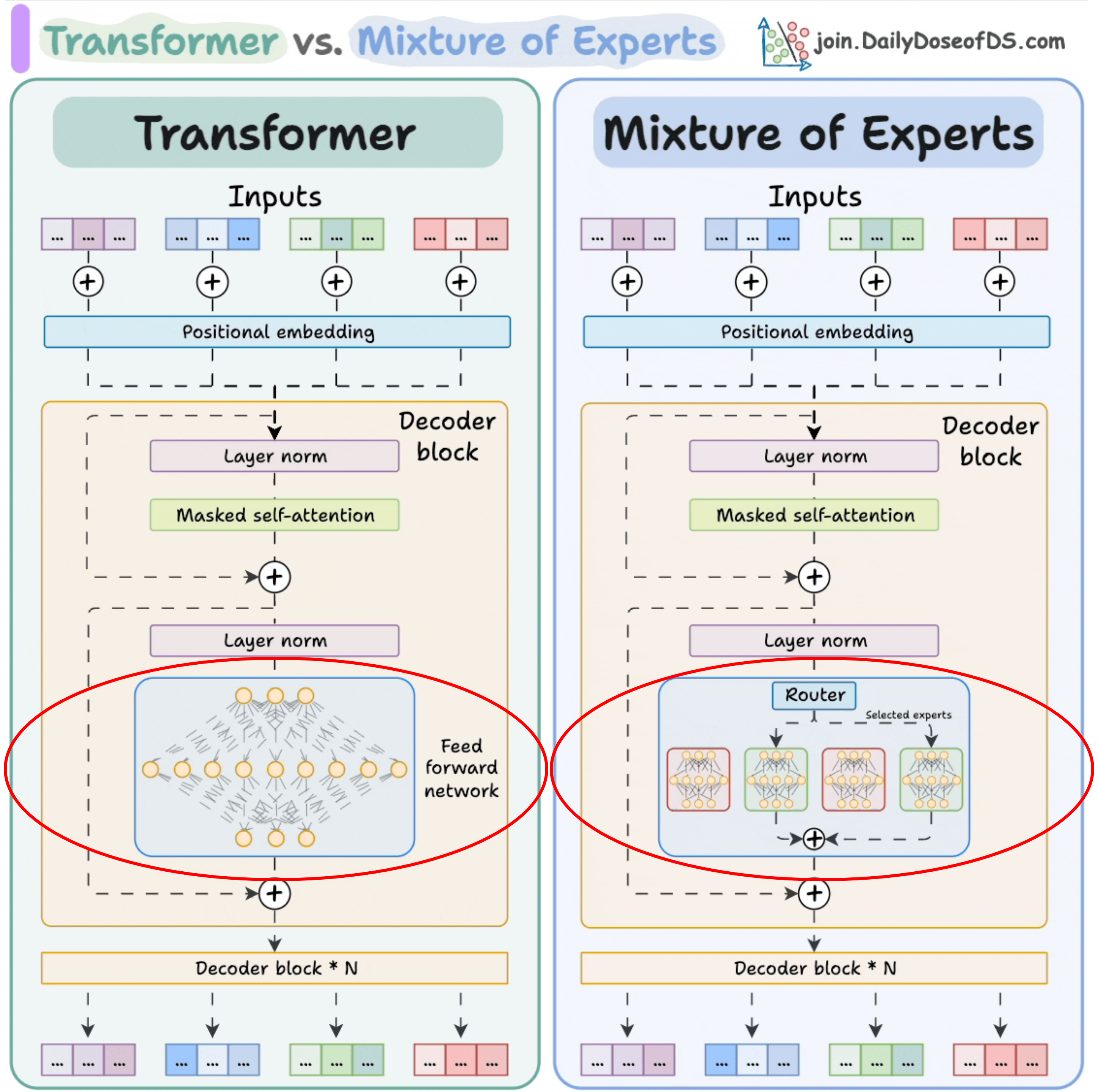


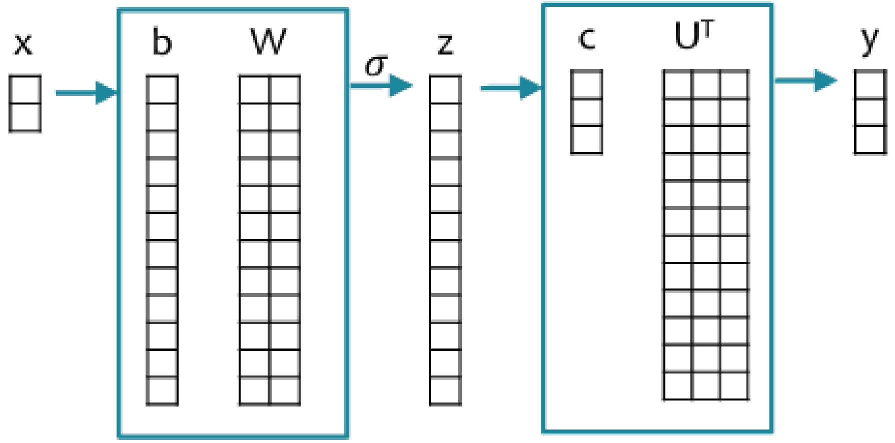
Illustration from https://www.dailydoseofds.com/p/transformer-vs-mixture-of-experts-in-llms/

MoE during inference

- During inference, a subset of expert are selected with the router. Since each expert is a much smaller neural network, the **inference step faster** than normal transformer.
- So that means **only some of the parameters are being used** during inference.
- However, all parameters and experts need to be loaded in RAM, so **memory requirement is still high**.

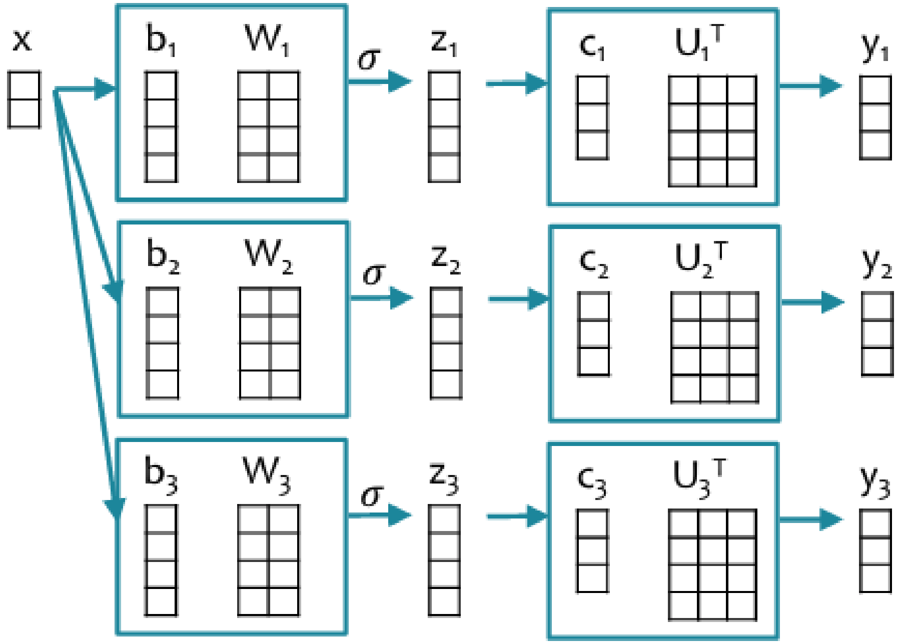
Breaking a Feed-forward Layer into Experts

Feed-forward Layer:



$y = U\,\sigma(Wx + b) + c$

3 Experts: (with same # of total parameters)



$y_i = U_i\,\sigma(W_i x + b_i) + c_i$

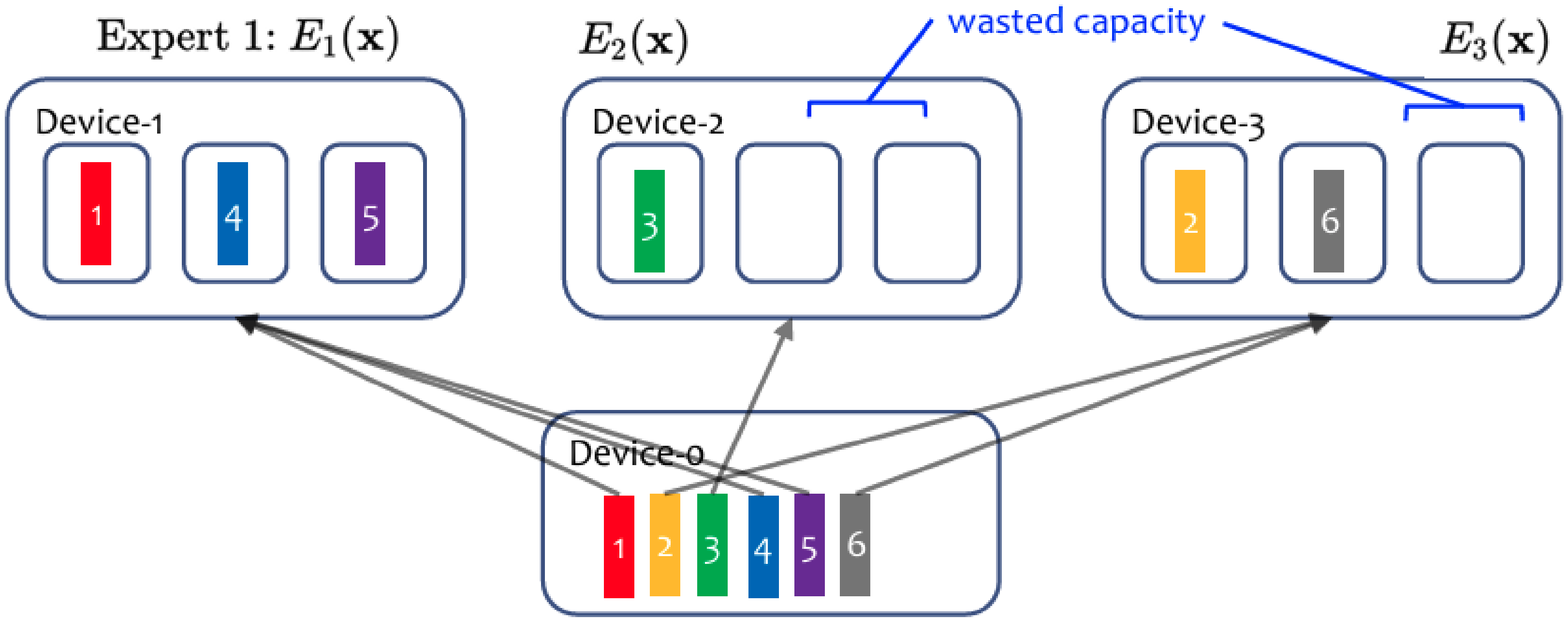
The two computations above are equivalent if $W = \mathrm{stack}(W_1, W_2, W_3)$ and $b = \mathrm{stack}(b_1, b_2, b_3)$ and $U^T = \mathrm{stack}(U_1^T, U_2^T, U_3^T)$ and $y = \underline{y_1 + y_2 + y_3\qquad}$? and $c = \underline{c_1 + c_2 + c_3\qquad}$?

Dense vs sparse MoE

- Dense: each expert have non-zero voice, in other words, all experts are activated for every token, just like standard feed-forward layer
- Sparse (more popular): only some experts are activated for each token, and the router decides which.
  - We are talking about this one today!

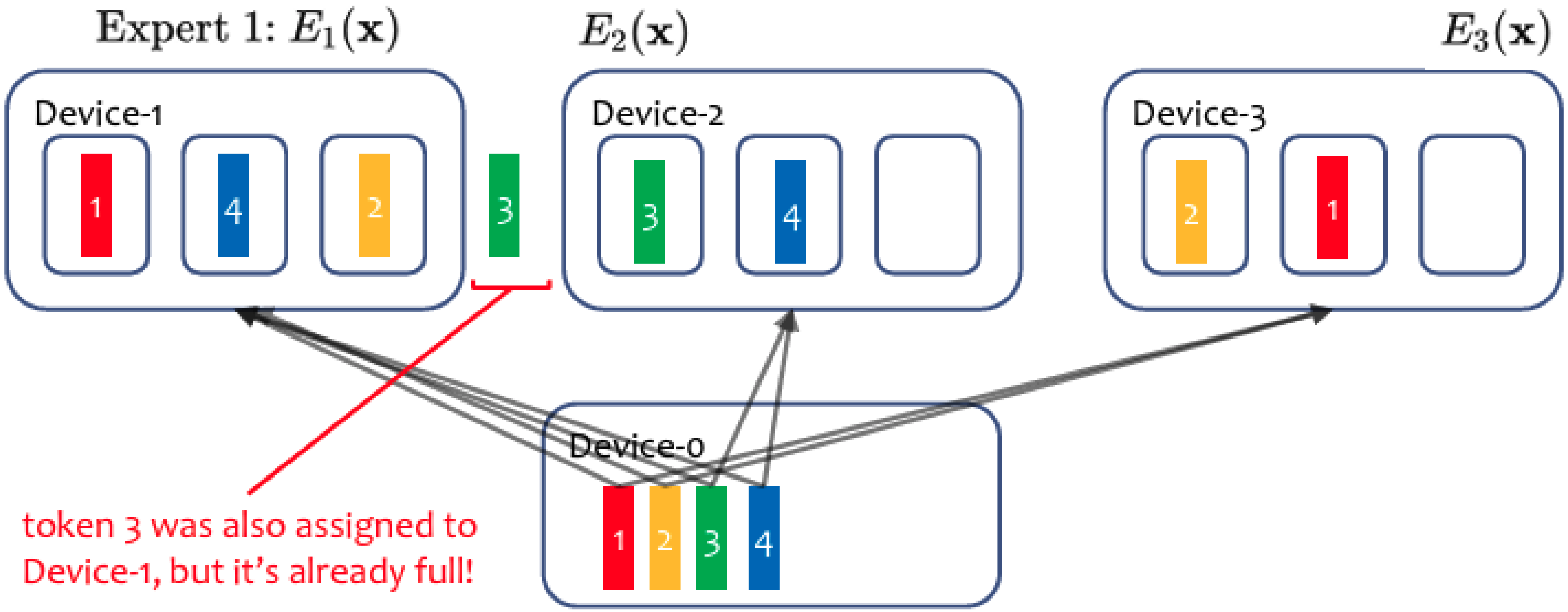
Expert Parallelism

- Each expert can be allocated to a different device (GPU)
- Each token is routed to K experts
- **Load balancing issues if too many tokens are routed to the same expert**
- Example: 3 devices, capacity of 3 tokens/device, 6 tokens, K=1 experts per token



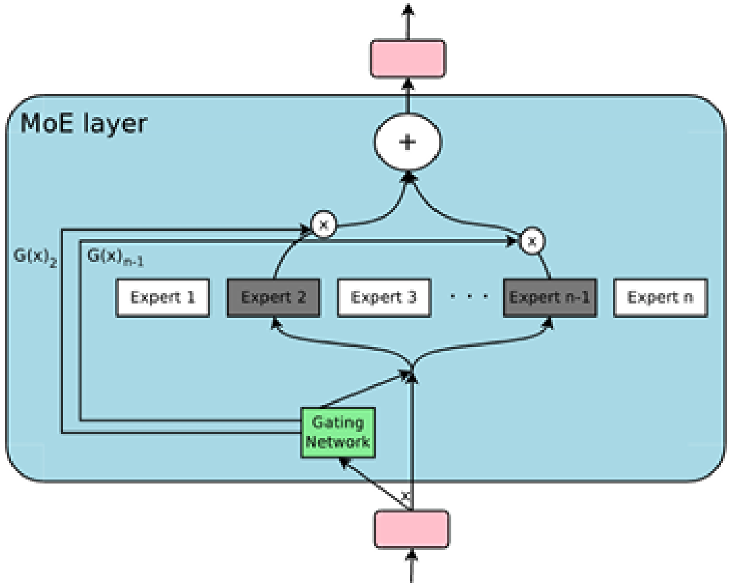
Expert Parallelism

- Example: 3 devices, capacity of 3 tokens/device, 4 tokens, K=2 experts per token



token 3 was also assigned to Device-1, but it's already full!

Sparsely-gated MoE



Many gating mechanisms: noisy top-K,
GShard ([Lepikhin et al., 2020](#)),
Switch transformer, etc.

One naïve solution is

- Choose highest-scoring k experts according to the softmax probability distribution

$$y = \sum_{i=1}^{n} G(x)_i E_i(x)$$

G decides which expert, E send to expert

- But without load balancing, the gate converges to few experts early!

Load balancing

- Left unregulated, all tokens are sent to just a few experts, which will make training inefficient.
  - Gating network/router converges early to same few experts
- How to regulate?
  Many options!
  - **Switch transformer**: only route each token to one expert!

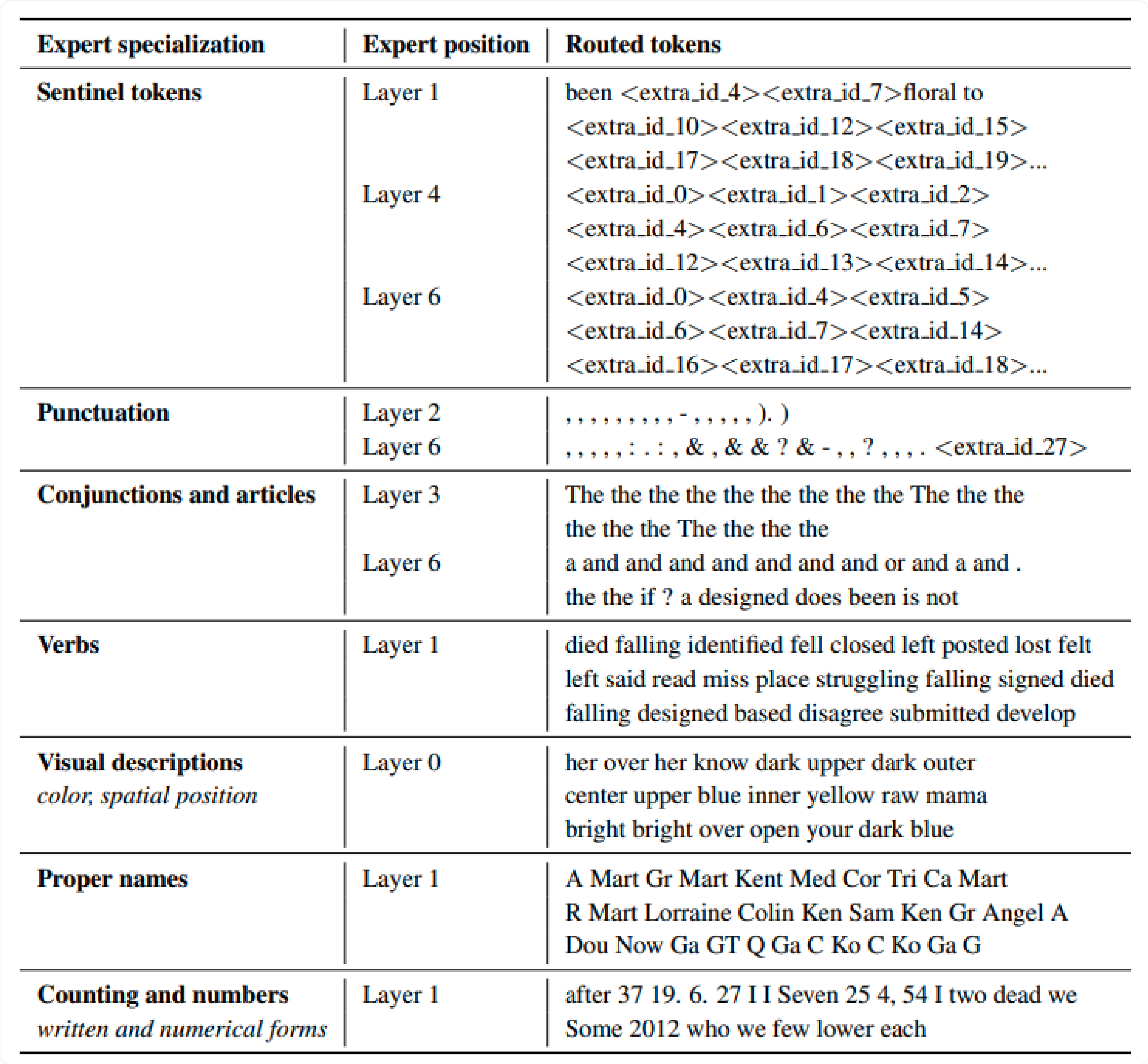
$$g_i(x) = \mathrm{softmax}(W_{\mathrm{router}} \cdot x)_i$$

$$\text{expert index}(x) = \mathrm{argmax}_i g_i(x)$$

Fast! But bigger risk of token overflow. If the capacity factor is set incorrectly, you might get too many tokens dropped or too many tokens assignment to one expert.

  - More advanced : Gshard/ Auxiliary loss
- More in-depth tutorial on this topic :
  - https://huggingface.co/blog/moe
  - https://huggingface.co/blog/NormalUhr/moe-balance

What does an expert learn? (examples from Switch Transformer)

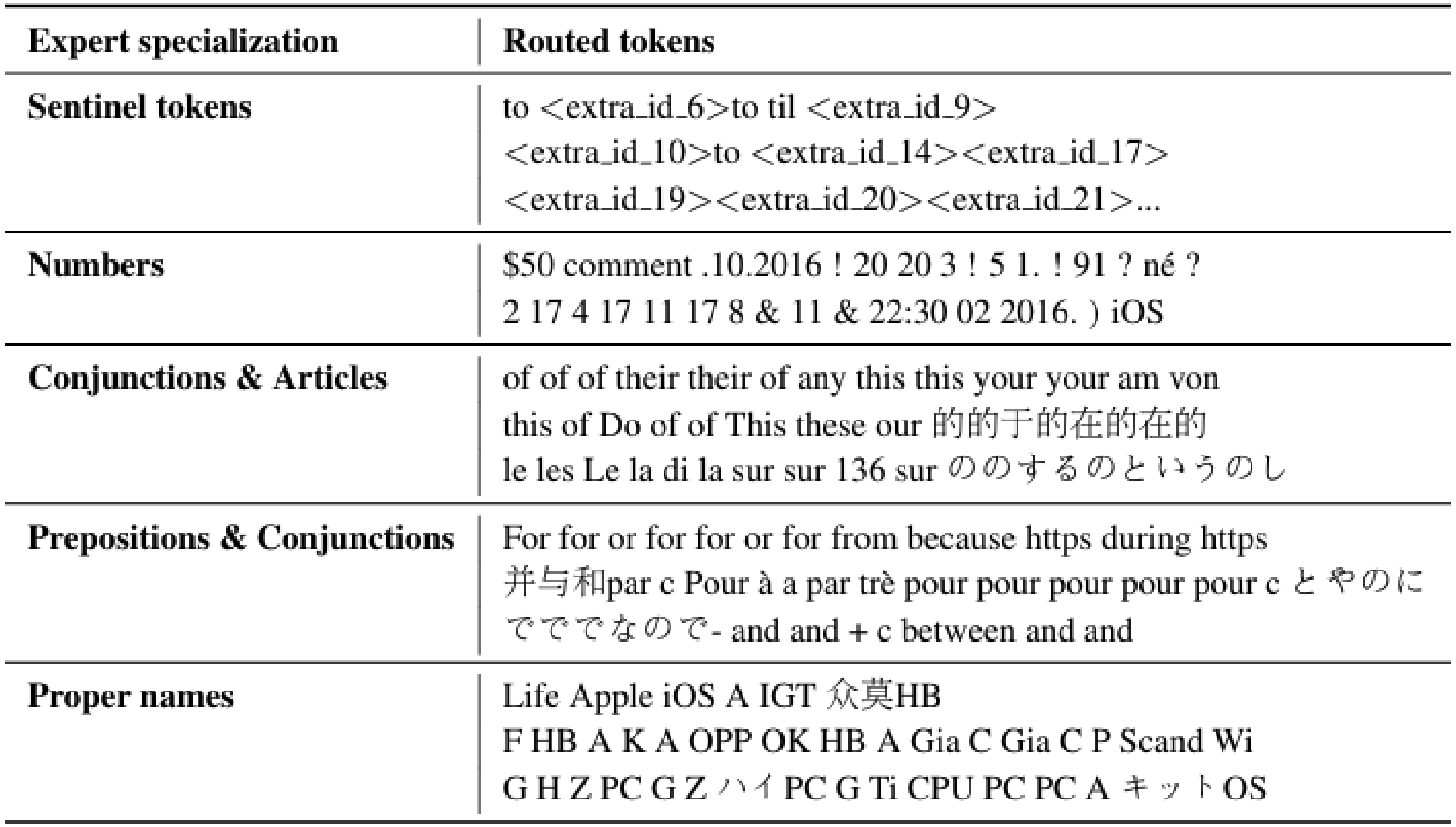


Expert specialization	Expert position	Routed tokens
Sentinel tokens	Layer 1	been <extra_id_4><extra_id_7>floral to <extra_id_10><extra_id_12><extra_id_15> <extra_id_17><extra_id_18><extra_id_19>...
	Layer 4	<extra_id_0><extra_id_1><extra_id_2> <extra_id_4><extra_id_6><extra_id_7> <extra_id_12><extra_id_13><extra_id_14>...
	Layer 6	<extra_id_0><extra_id_4><extra_id_5> <extra_id_6><extra_id_7><extra_id_14> <extra_id_16><extra_id_17><extra_id_18>...
Punctuation	Layer 2	, , , , , , , , , - , , , , ,).)
	Layer 6	, , , , , : . : , & , & & ? & - , , ? , , , . <extra_id_27>
Conjunctions and articles	Layer 3	The the the the the the the the the The the the the the the The the the the
	Layer 6	a and and and and and and or and a and . the the if ? a designed does been is not
Verbs	Layer 1	died falling identified fell closed left posted lost felt left said read miss place struggling falling signed died falling designed based disagree submitted develop
Visual descriptions *color, spatial position*	Layer 0	her over her know dark upper dark outer center upper blue inner yellow raw mama bright bright over open your dark blue
Proper names	Layer 1	A Mart Gr Mart Kent Med Cor Tri Ca Mart R Mart Lorraine Colin Ken Sam Ken Gr Angel A Dou Now Ga GT Q Ga C Ko C Ko Ga G
Counting and numbers *written and numerical forms*	Layer 1	after 37 19. 6. 27 I I Seven 25 4, 54 I two dead we Some 2012 who we few lower each

What does an expert learn? (examples from Switch Transformer)

In multilingual setting:

Experts are disappointedly not specializing in language....



Expert specialization	Routed tokens
Sentinel tokens	to <extra_id_6>to til <extra_id_9> <extra_id_10>to <extra_id_14><extra_id_17> <extra_id_19><extra_id_20><extra_id_21>...
Numbers	$50 comment .10.2016 ! 20 20 3 ! 5 1. ! 91 ? né ? 2 17 4 17 11 17 8 & 11 & 22:30 02 2016.) iOS
Conjunctions & Articles	of of of their their of any this this your your am von this of Do of of This these our 的的于的在的在的 le les Le la di la sur sur 136 sur ののするのというのし
Prepositions & Conjunctions	For for or for for or for from because https during https 并与和par c Pour à a par trè pour pour pour pour pour c とやのに でででなので- and and + c between and and
Proper names	Life Apple iOS A IGT 众莫HB F HB A K A OPP OK HB A Gia C Gia C P Scand Wi G H Z PC G Z ハイ PC G Ti CPU PC PC A キット OS

Number of experts

- With Switch Transformer, more experts lead to better efficiency, but these are diminishing gains
- After 256 experts, the gains are no longer monotonic
- Also more VRAM will be needed for inference.

Talking about number experts…
Designing an LLM with MoE architecture is also heavily tied to the hardware. In an industry research team, you will need experts in training large-scale models, high performance computing, and distributed systems.

Advantages and disadvantages of MoE

- Advantages:
  - It makes **pretraining** much faster and efficient
  - Much faster **inference** compared to a model with the same number of parameters.
- Disadvantages:
  - Requires **high VRAM** since all experts are loaded in memory
  - **Fine-tuning** could be challenging

Thank you!

- And that's the final lecture for CS 6120!
- Thank you, 60 of you, for enrolling in my class! I hope you've found this class useful and enjoyed most of the topics!
- Hope you have a great Thanksgiving! I look forward to reading your final reports!