

Direct Model Editing and Large Model Interpretability

UCB Lecture, January 2024

David Bau

Northeastern University



Deep network interpretability asks:

why did my neural network make that decision?

What does a Deep Net Learn?

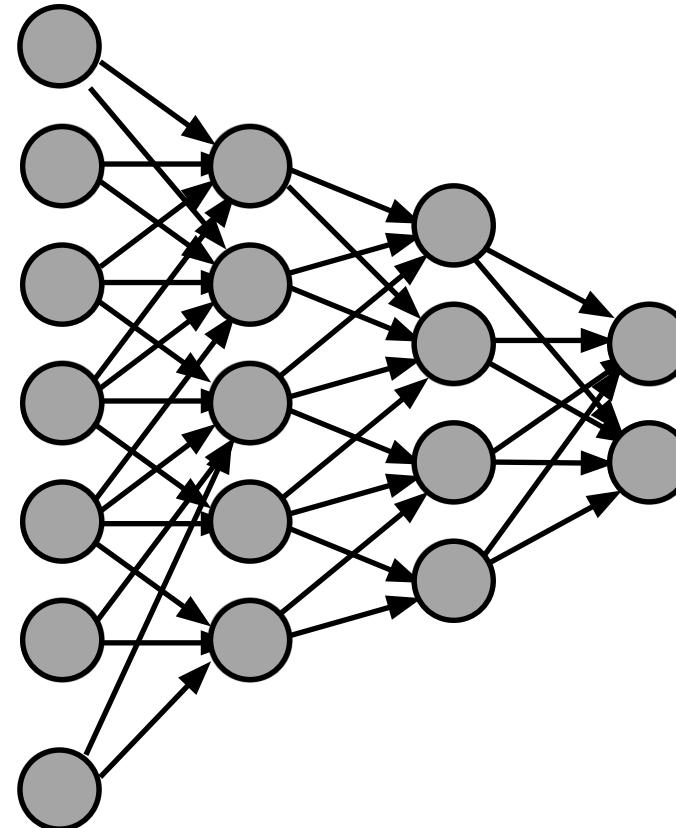
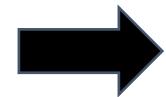
“indoor market”



“baseball field”



“tree farm”

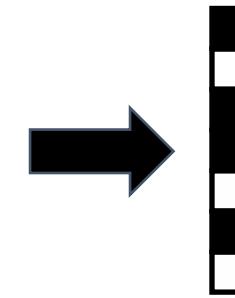


Minimize cross-entropy loss

Top5 acc 85%



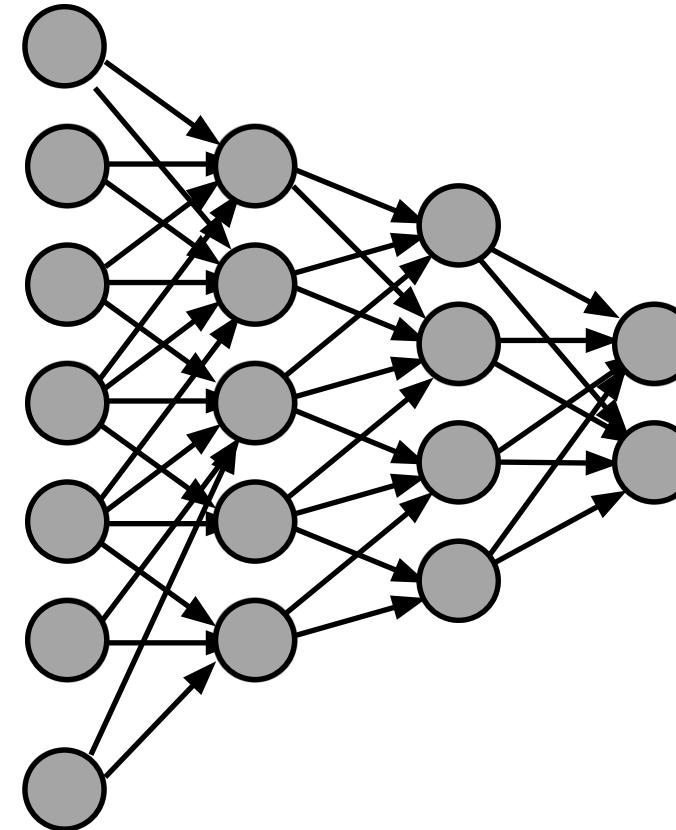
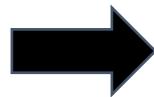
Prediction vector



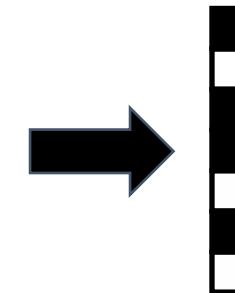
365 dimensions

[Places dataset, Zhou 2016]

What does it really learn?



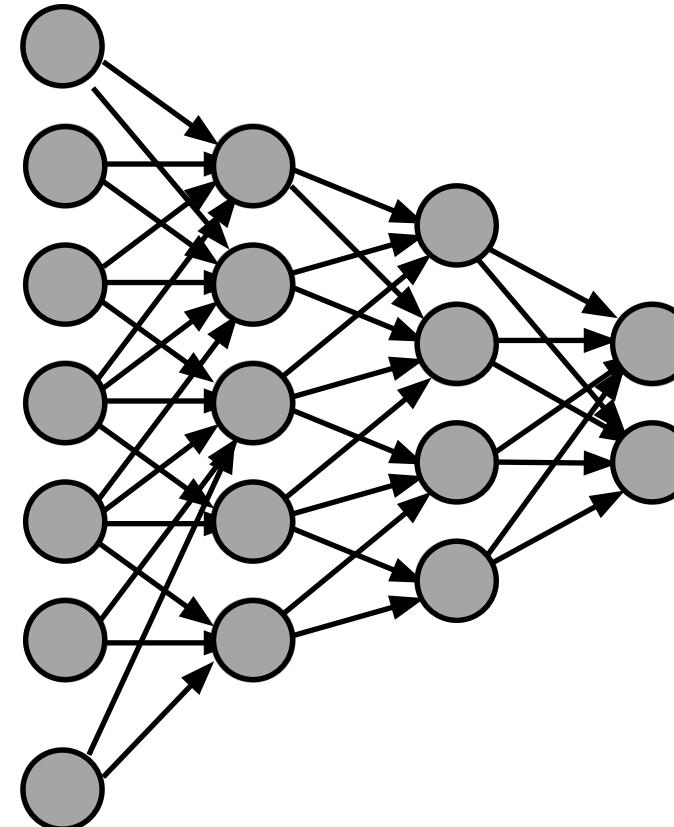
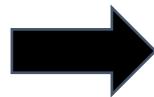
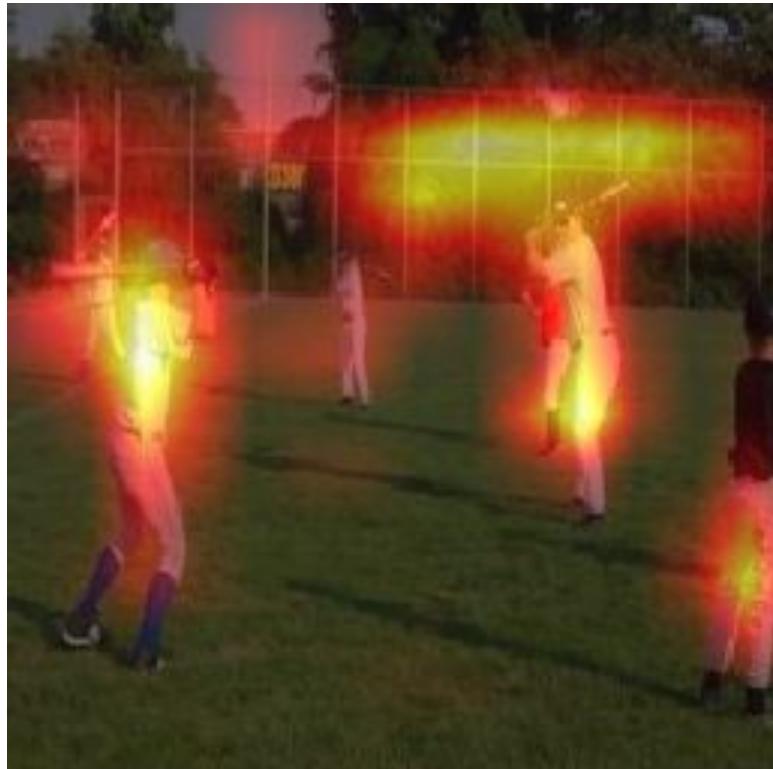
Prediction vector



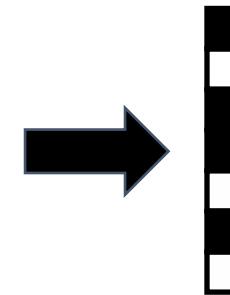
“baseball field”

[VGG-16, Simonyan 2014]

What does it really learn?



Prediction vector



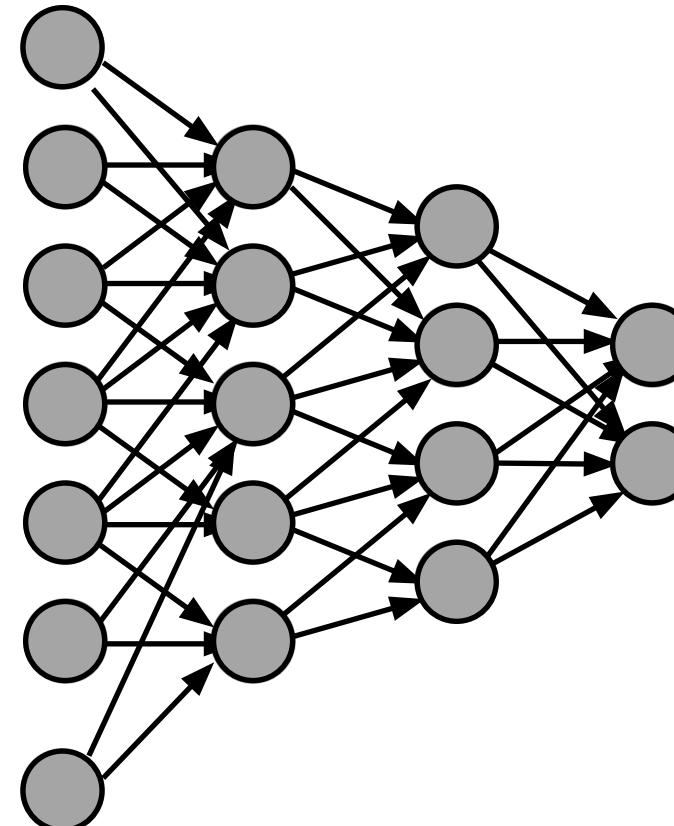
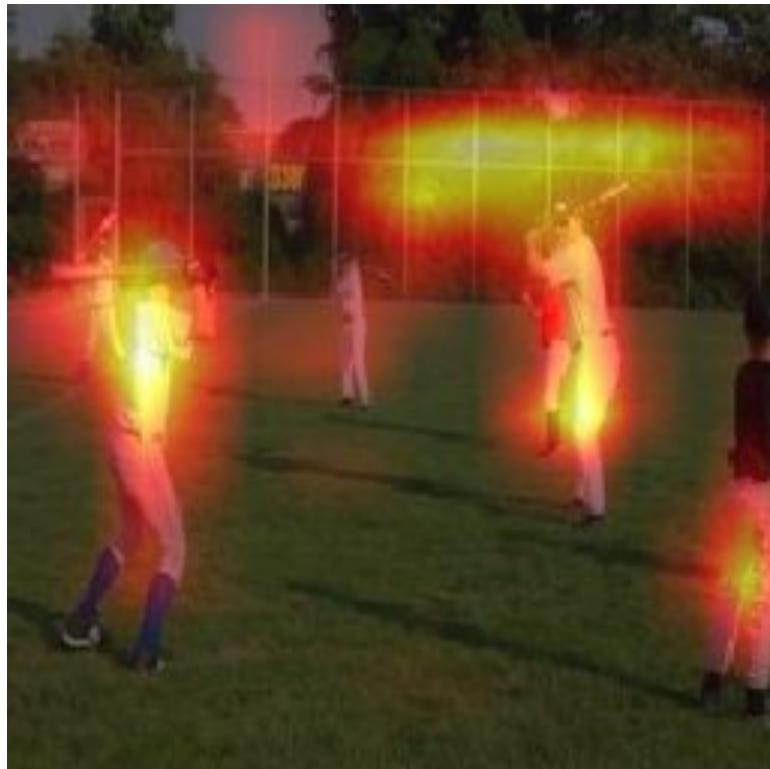
“baseball field”

Where did the network look?

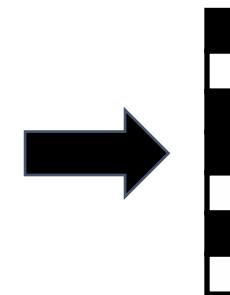
Grad-CAM [Selvaraju CVPR 2017]

What does it really learn?

Why is the network looking there?



Prediction vector

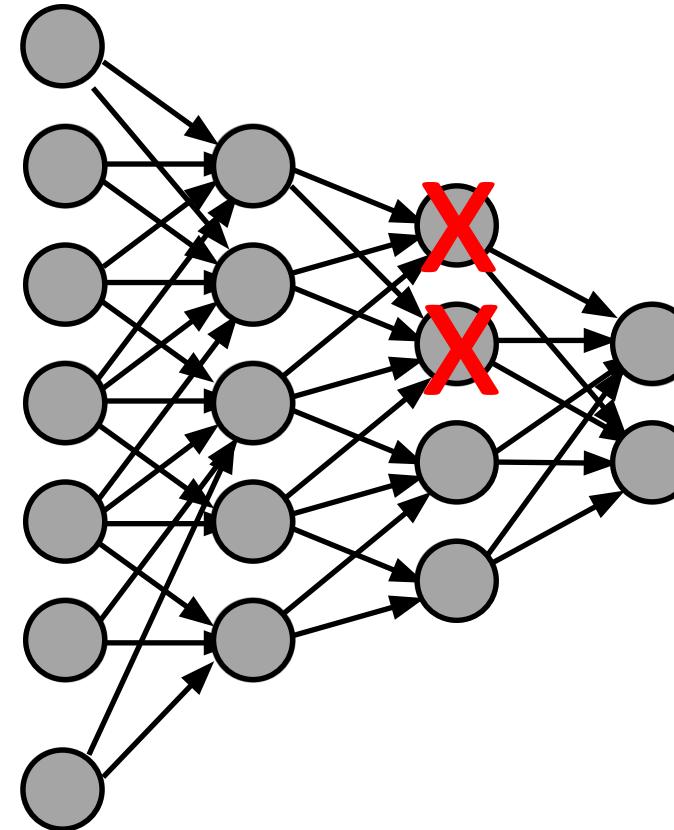


“baseball field”

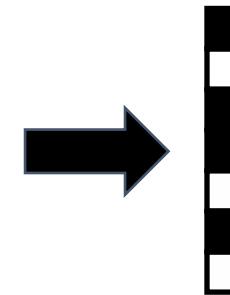
Where did the network look?

Grad-CAM [Selvaraju CVPR 2017]

What is the network looking for?



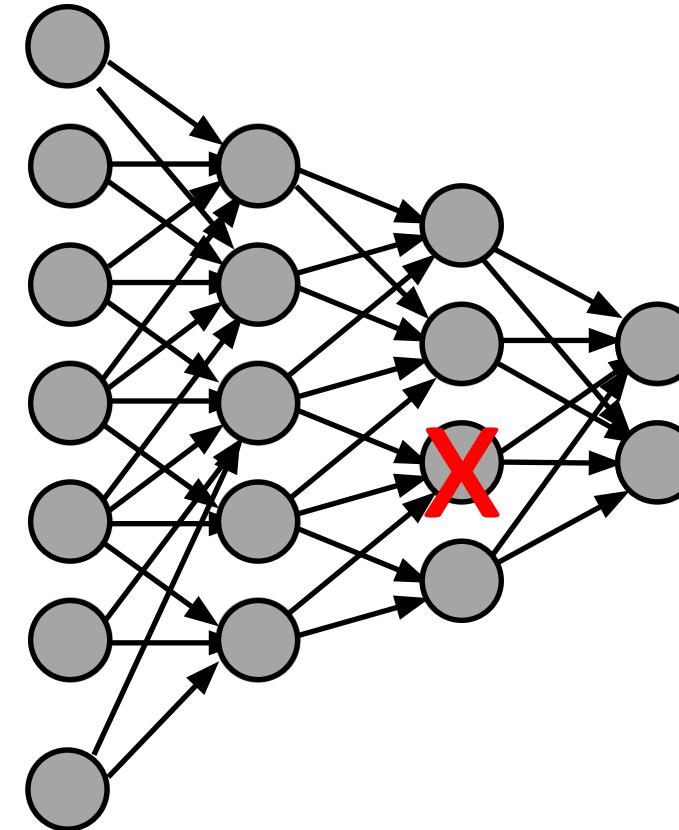
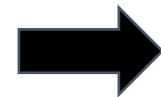
Prediction vector



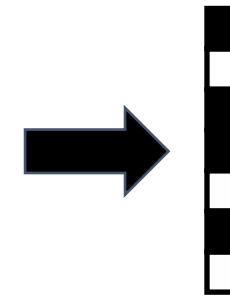
“baseball field”
unchanged

509/512 units do not change prediction when removed

What the ~~network~~ looking for?



Prediction vector

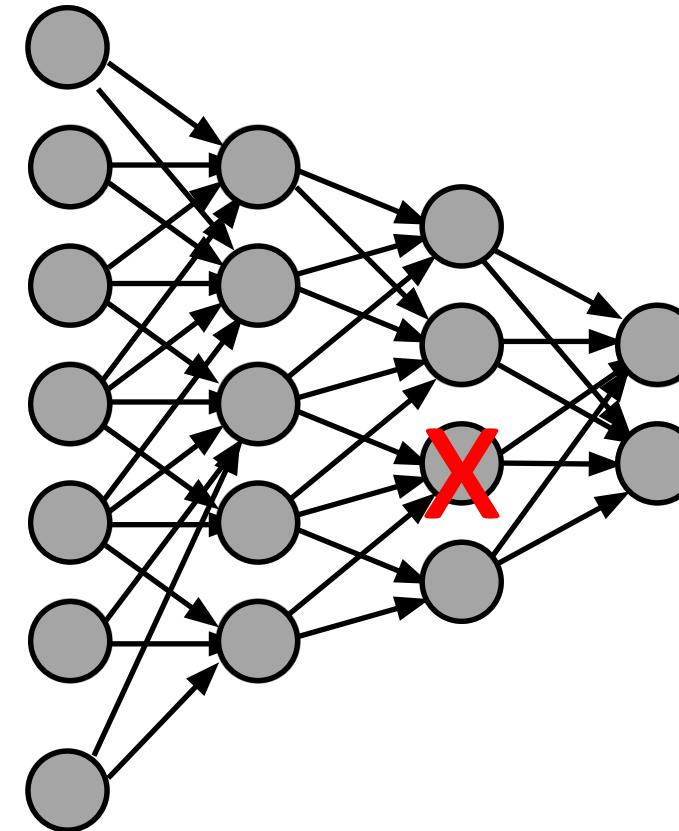


“SOCCER field”

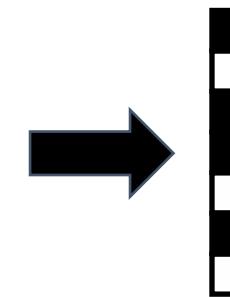
WRONG

Units 208, 467, and 161 do change this prediction when removed

What are the units looking for?



Prediction vector



“SOCCER field”

WRONG

Units 208, 467, and 161 do change this prediction when removed

What are the units looking for?



Top 1% activations of unit 208



Unit 467



Unit 161

What is unit 208 looking for?



Top 1% activations of unit 208



Unhatted people
are skipped



Top 1% activations of unit 208 across whole data set

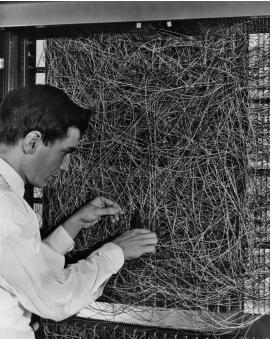
I trained my network to label scenes.

And then it learned to identify hats!

What concepts emerge on a net trained
without labels?

Deep networks are genuinely surprising

1962, Perceptron: learning



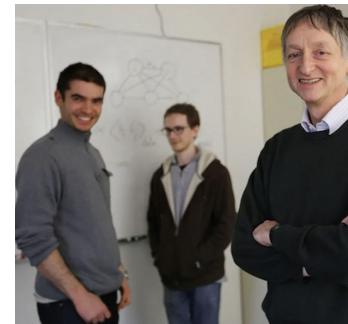
64 params

1986, LeNet: generalization



60,000 params

2012, Alexnet: SOTA



60,000,000 params

2020, GPT-3: metalearning



175,000,000,000 params

1000x

1000x

3000x

Probing ChatGPT Knowledge

Last-year midterm question

DA

Does the following code implement a classifier or an autoencoder?

```
import torch
from torch.nn import Sequential, ReLU, Linear
# commented out: # torch.set_grad_enabled(False)
num_pixels = 1000000
bottleneck = 100
net = Sequential(
    Linear(in_features=num_pixels, out_features=bottleneck, bias=True),
    ReLU(),
    Linear(in_features=bottleneck, out_features=num_pixels, bias=False)
).cuda()
print('Parameter tensors', len(list(net.named_parameters())))
print('Parameter elements', sum(p.numel() for p in net.parameters()))
total_error = 0
sample_size = 10000
for test_index in range(sample_size):
    test_data = torch.randn(1, num_pixels, device='cuda')
    total_error += (net(test_data) - test_data).pow(2).mean()
print('Average error', total_error / sample_size)
```

Probing ChatGPT Knowledge

Last-year midterm question

```
net = Sequential(  
    Linear(in_features=num_pixels, out_features=bottleneck, bias=True),  
    ReLU(),  
    Linear(in_features=bottleneck, out_features=num_pixels, bias=False)  
.cuda()  
print('Parameter tensors', len(list(net.named_parameters())))  
print('Parameter elements', sum(p.numel() for p in net.parameters()))  
total_error = 0  
sample_size = 10000  
for test_index in range(sample_size):  
    test_data = torch.randn(1, num_pixels, device='cuda')  
    total_error += (net(test_data) - test_data).pow(2).mean()  
print('Average error', total_error / sample_size)
```



This code appears to implement an autoencoder. An autoencoder is a type of neural network that is used for dimensionality reduction. In this code, the input data is passed through a bottleneck layer with 100 units, which reduces the number of features from 1000000 to 100. The output of the autoencoder is then passed through another layer, which attempts to reconstruct the original input from the bottleneck representation. The reconstruction error is then calculated and printed out.

Probing ChatGPT Knowledge

```
net = Sequential(  
    Linear(in_features=num_pixels, out_features=bottleneck, bias=True),  
    ReLU(),  
    Linear(in_features=bottleneck, out_features=num_pixels, bias=False)  
).cuda()  
print('Parameter tensors', len(list(net.named_parameters())))  
print('Parameter elements', sum(p.numel() for p in net.parameters()))  
total_error = 0  
sample_size = 10000  

```

DA When you run this code, it always crashes after about 10 seconds saying 'out of memory'. How could we change this code so that it runs 10000 iterations easily and prints the average error?

One issue with the current code is that it is trying to run the entire autoencoder on all 10000 test samples at once, which requires a lot of memory. One way to fix this issue would be to process the samples in smaller batches, and update the total error and sample size accordingly. This would allow the code to run without crashing due to memory limitations. Here is an example of how the code could be modified to do this:

One possible solution is here,
although the text doesn't explain it

```
import torch  
from torch.nn import Sequential, ReLU, Linear  
num_pixels = 1000000  
bottleneck = 100  
net = Sequential(  
    Linear(in_features=num_pixels, out_features=bottleneck, bias=True),  
    ReLU(),  
    Linear(in_features=bottleneck, out_features=num_pixels, bias=False)  
).cuda()  
print('Parameter tensors', len(list(net.named_parameters())))  
print('Parameter elements', sum(p.numel() for p in net.parameters()))  
total_error = 0  
sample_size = 10000  

```

In this modified code, the autoencoder is run on batches of 100 samples at a time, and the total error is updated accordingly. This should allow the code to run without crashing due to memory limitations.

Probing ChatGPT Knowledge

DA

Who makes the Philips Hue light bulbs?



The Philips Hue line of smart light bulbs is made by the Philips lighting division of Signify, a Dutch technology company. Signify was formed in 2018 as a spin-off of the Philips lighting business, and is now one of the leading manufacturers of lighting products and solutions worldwide. Philips Hue bulbs are known for their high quality and innovative features, and are a popular choice among homeowners and businesses alike for their smart lighting needs.

GPT knows a lot of facts about the world!

Probing ChatGPT Knowledge

DA

Can you tell me who has recently been elected Governor of Pennsylvania?



I'm sorry, but I am not able to browse the internet or access current information. As a language model trained by OpenAI, my knowledge is based solely on the text that I have been trained on, and my responses are generated based on that information. My knowledge cutoff is in 2021, so I am not able to provide information about current events or recent developments.

It is always dissatisfying running into the guardrails!

What about metalearning?

Ask a language model to predict the next word:

13, 2, 11.
18, 14, 4.
32, 21, 9.
4, 3, 1.
18, 13, 5.
23, 7, 16.
7, 2, 5

Input to model

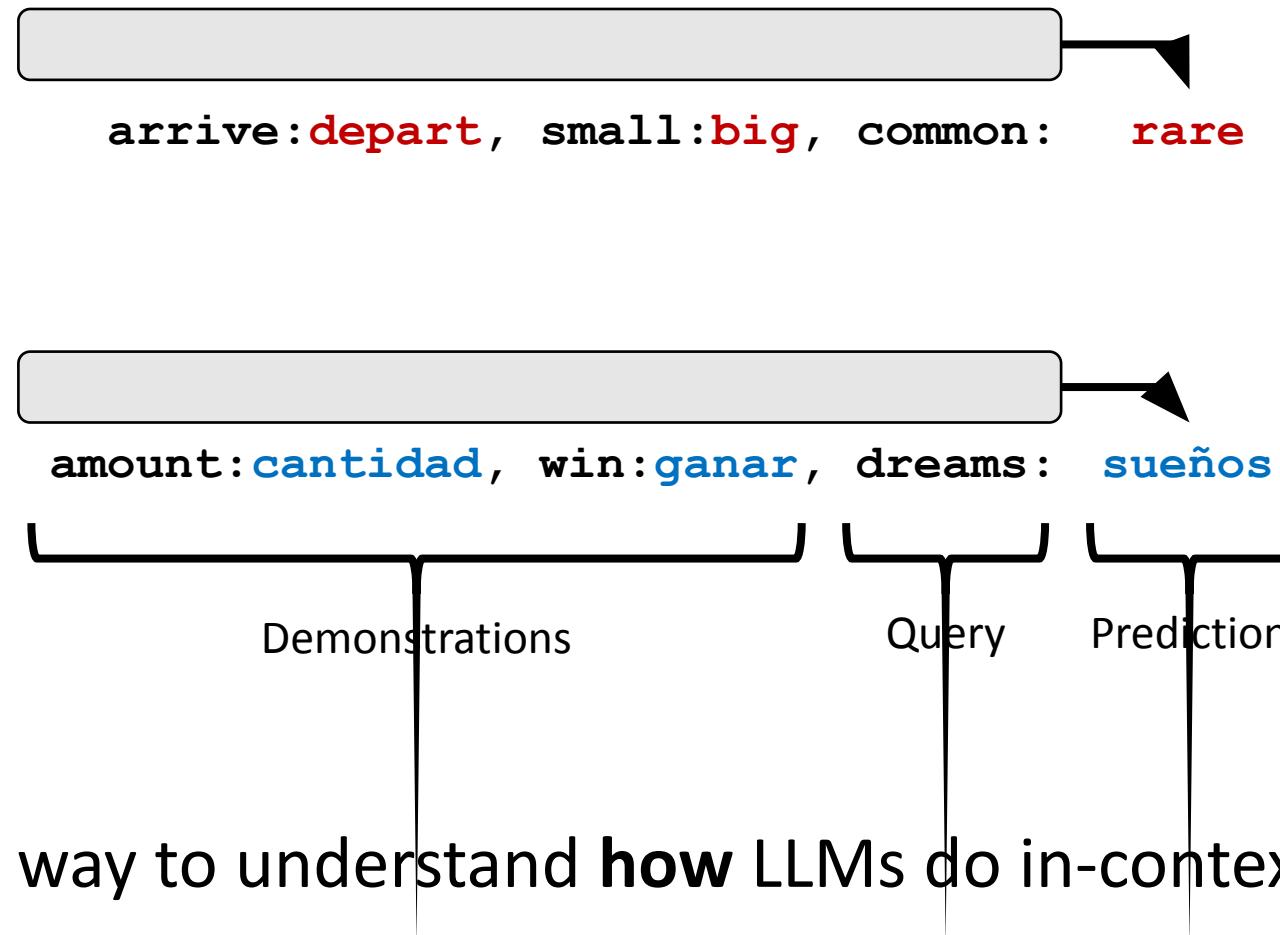
13, 2, fifteen.
18, 14, thirty-two.
32, 21, fifty-three.
4, 3, seven.
18, 13, thirty-one.
23, 7, thirty.
7, 2, nine

To predict these, a model needs to learn something.

But unlike regular “machine learning,” this model is **not** optimizing its weights.

What is it doing?

How to understand Metalearning?



Self-explanation

11?

Inputs

- 13, 2, fifteen.
- 18, 14, thirty-two.
- 32, 21, fifty-three.
- 4, 3, seven.
- 18, 13, thirty-one.
- 23, 7, thirty.
- 7, 2, nine.

> Explain the pattern here?

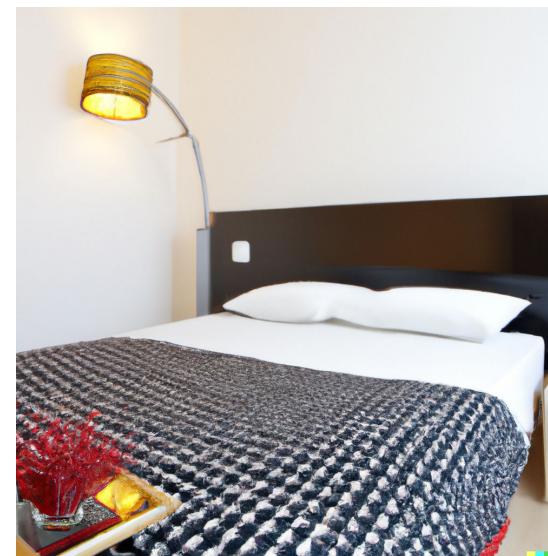
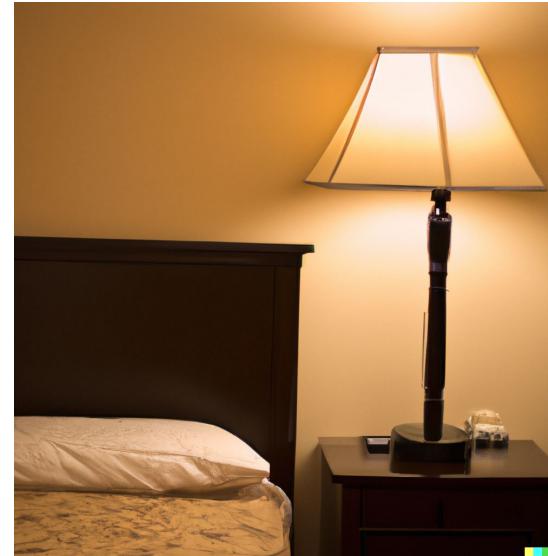
The pattern is adding the two numbers together to get the third number. For example, in the first set, 11 + 2 = 13, and in the second set, 18 + 14 = 32. This pattern continues throughout the sets

> Anything else I need to do to follow the pattern?

No, as long as you continue to add the two numbers together to get the third number, you are following the pattern correctly.

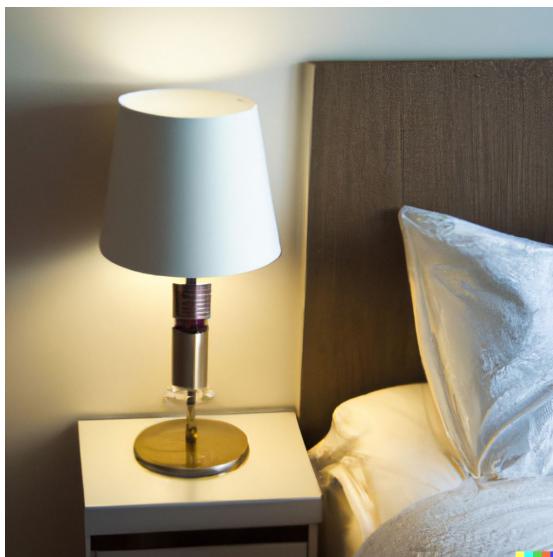
Is English enough for model interpretation?

“Bedroom with a lamp”



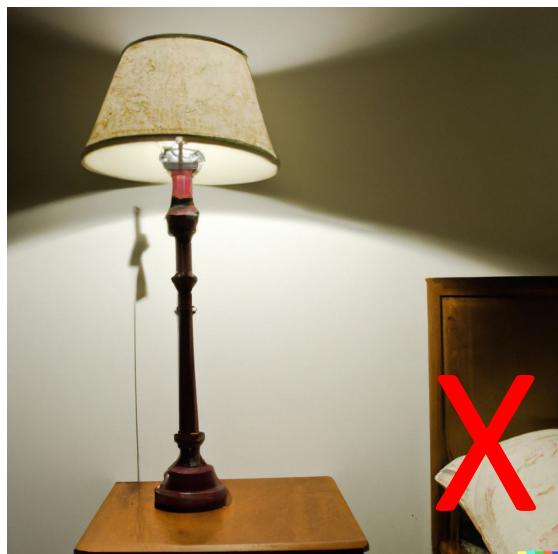
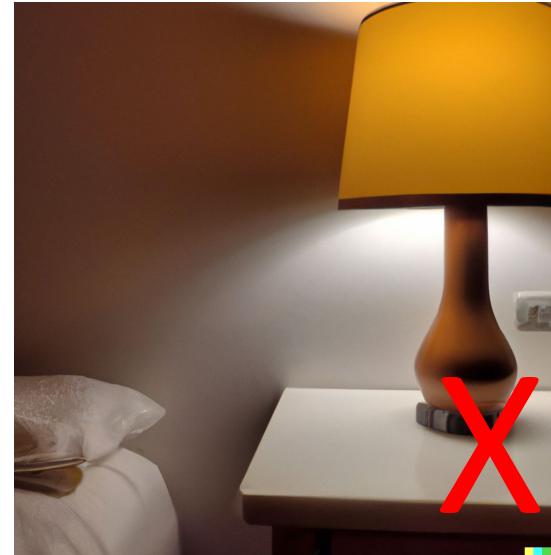
[Dall-E 2, OpenAI]

“Bedroom with a lit lamp”



[Dall-E 2, OpenAI]

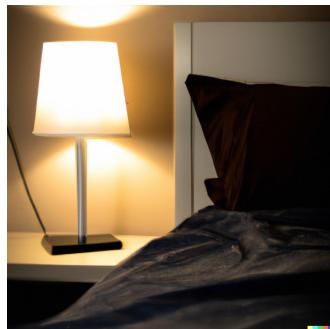
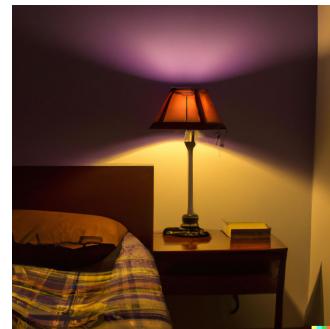
“Bedroom with an unlit lamp”



Vocabulary
“**unlit** lamp”
seems to
be missing

X not correct

Language Lacks Lamp Lighting



“darkened lamp”

“dark lamp”

“lamp that
is turned off”

how can we turn the lights off?

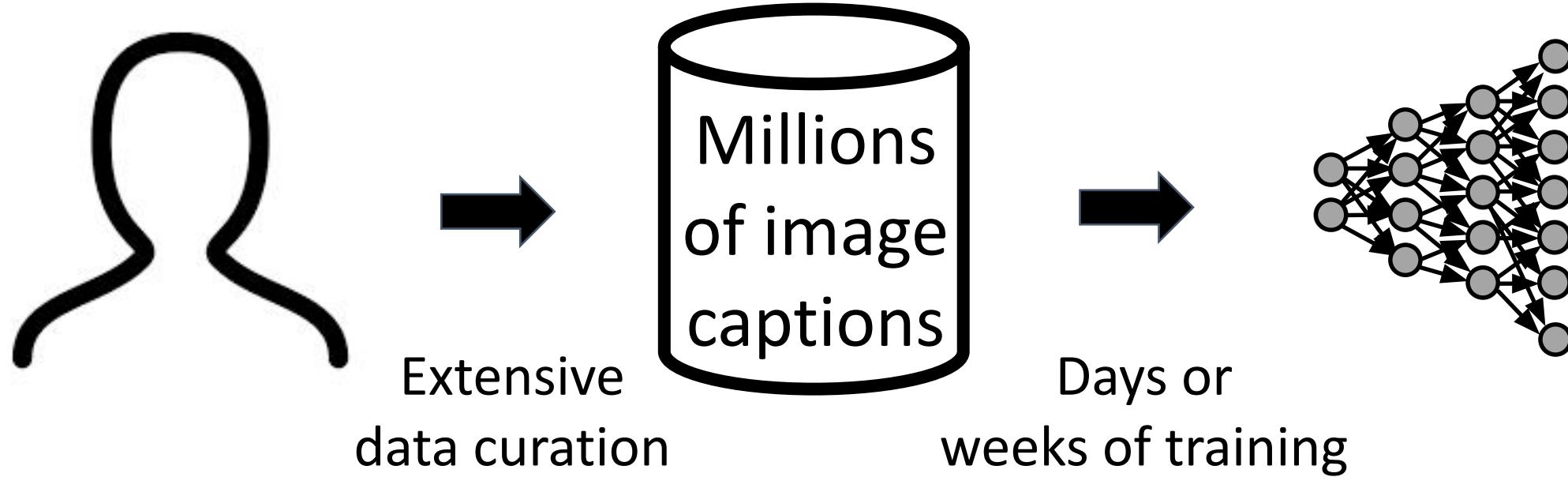
[Dall-E 2, OpenAI]

Our core question:

What does my neural network know?

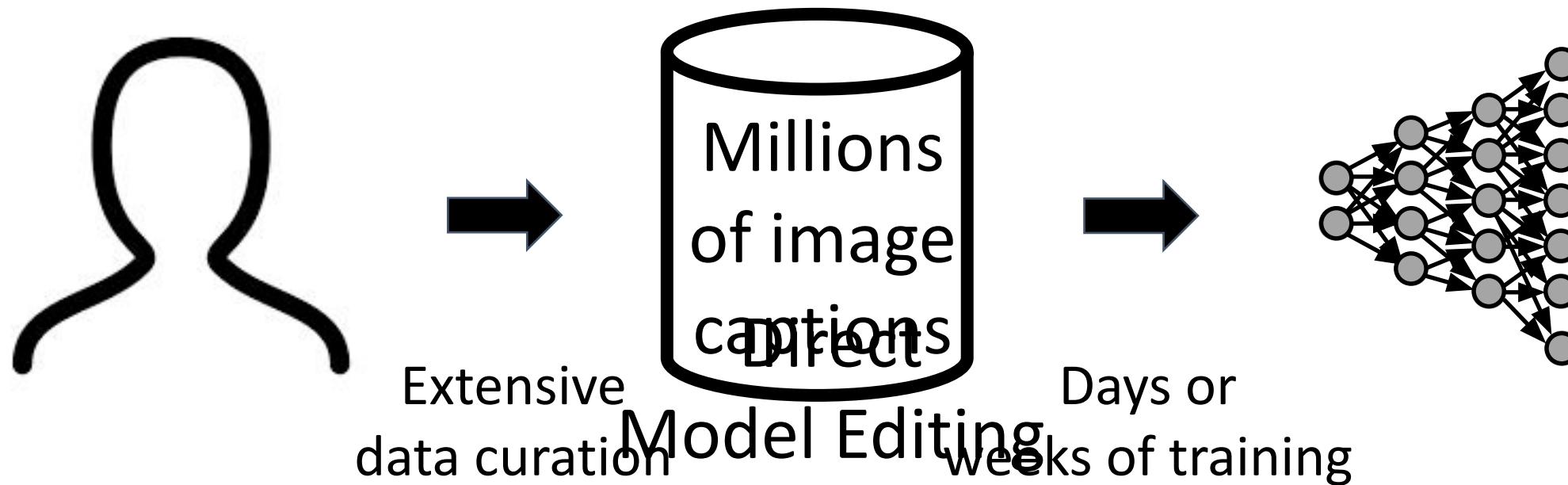
... and **how does it know it?**

Big Data can be a difficult human interface

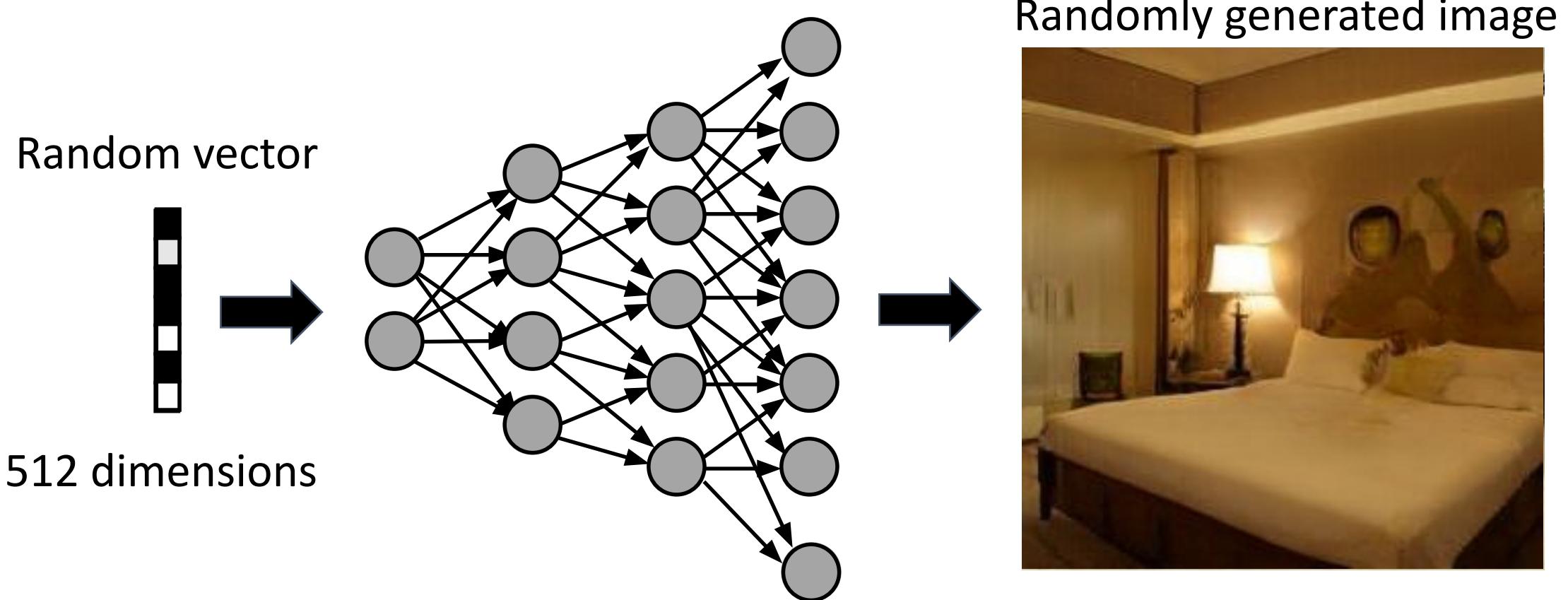


How to deal with missing concepts, mistakes, and ambiguities?

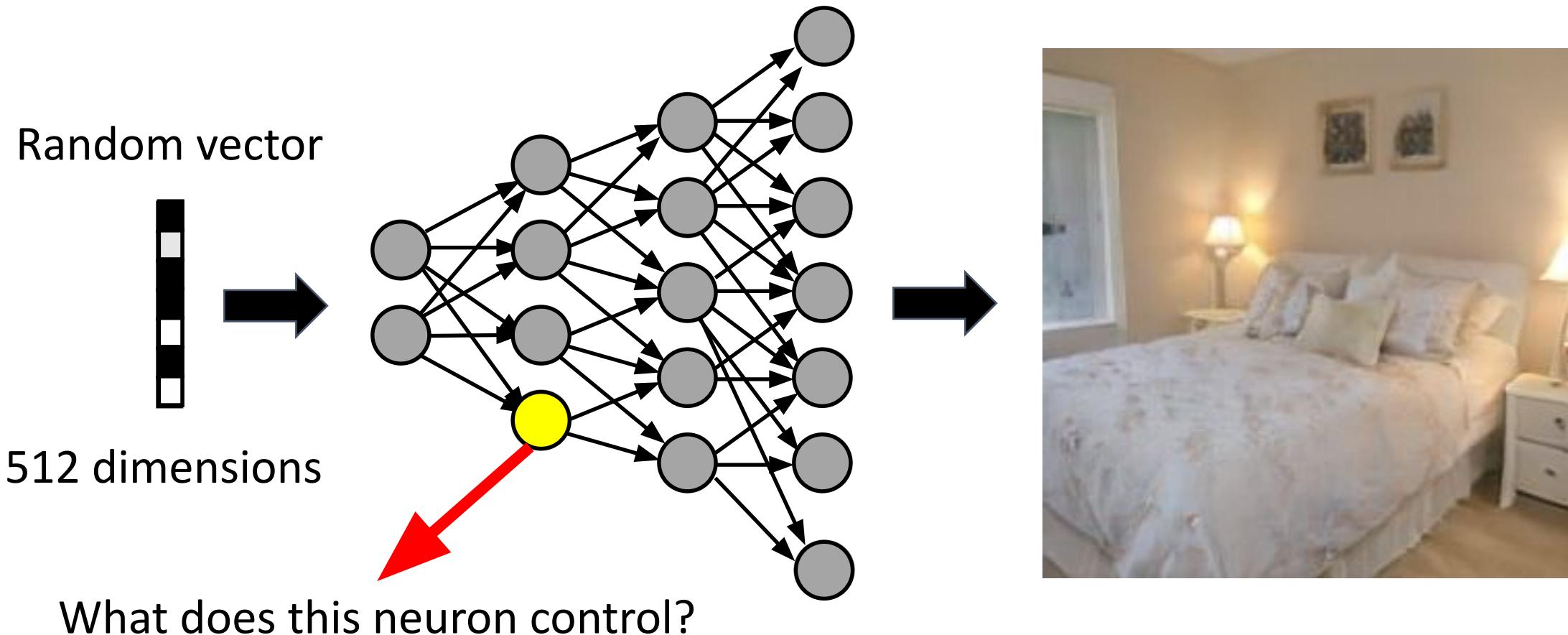
Goal: Direct Model Editing



Using a GAN Generator



Learning the network's own language



Searching for a neuron for lighting

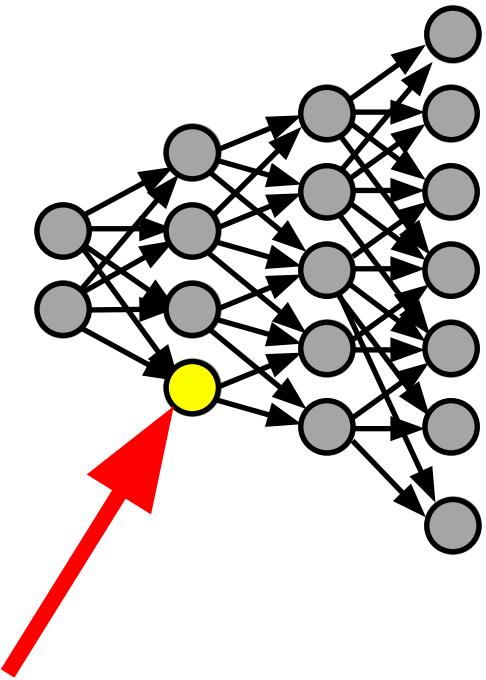
1. Generate an image



2. Select desired control



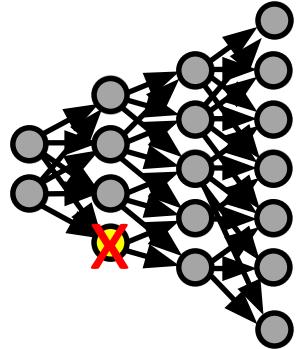
3. Hunt for neurons



Our query:

which neurons cause these outputs?

A causal search ranking for neurons



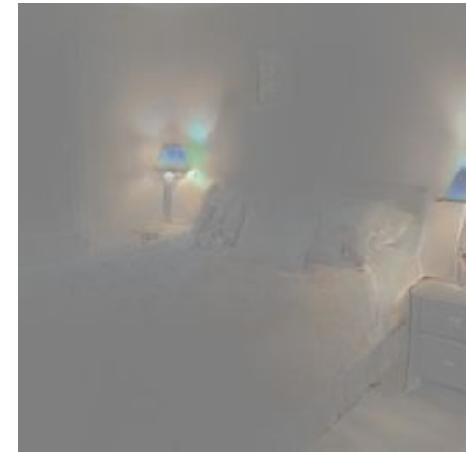
Generate with one neuron zeroed

-



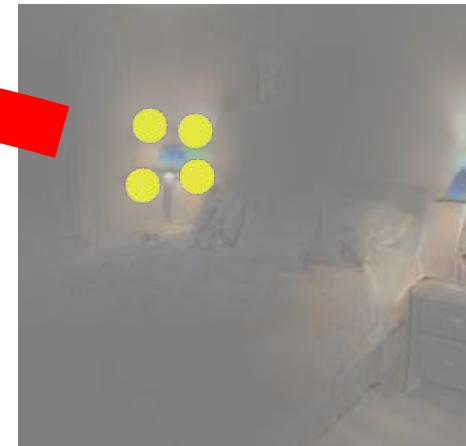
Original image

=



Difference

Score based on how much difference
lands in the target control areas



A casual search ranking for neurons

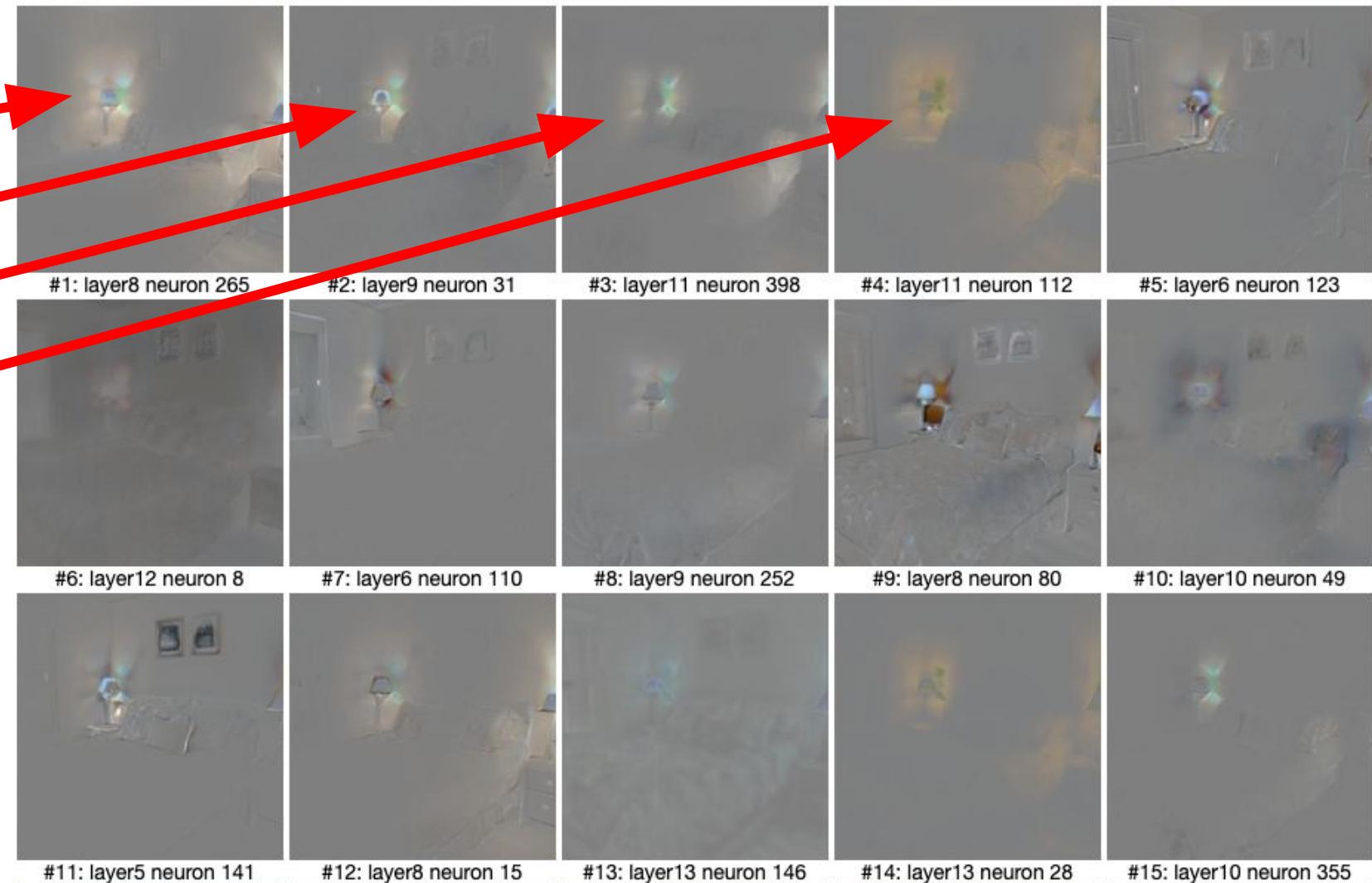
Neurons that rank highest:

Task lighting+reflections

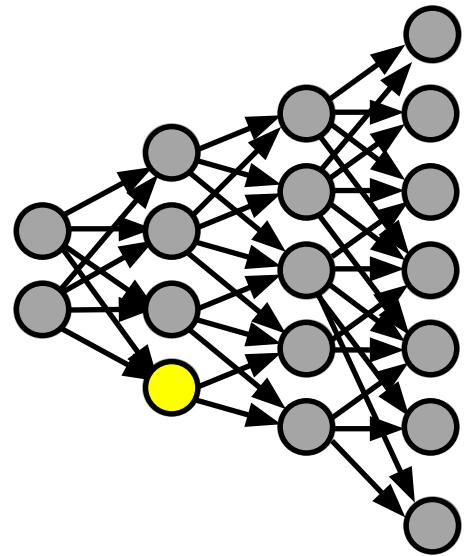
Lampshade size

Reflections only

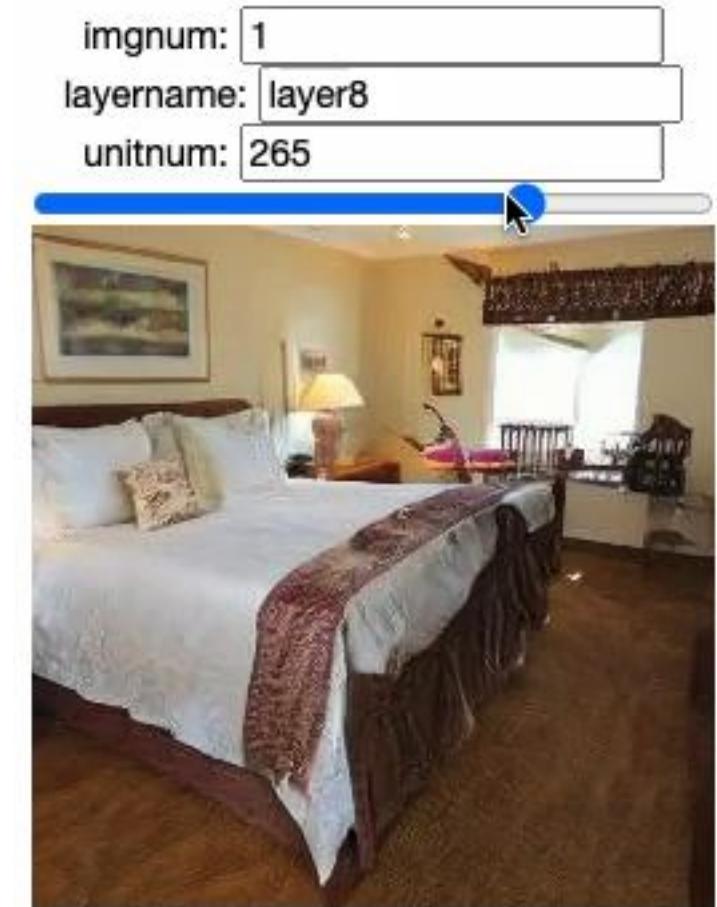
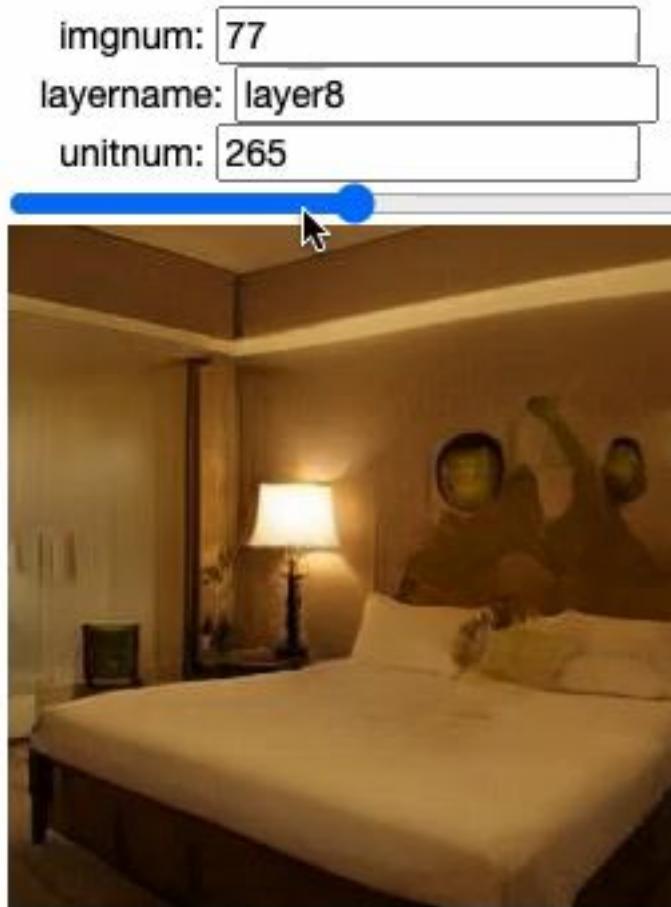
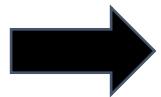
Warmth of light



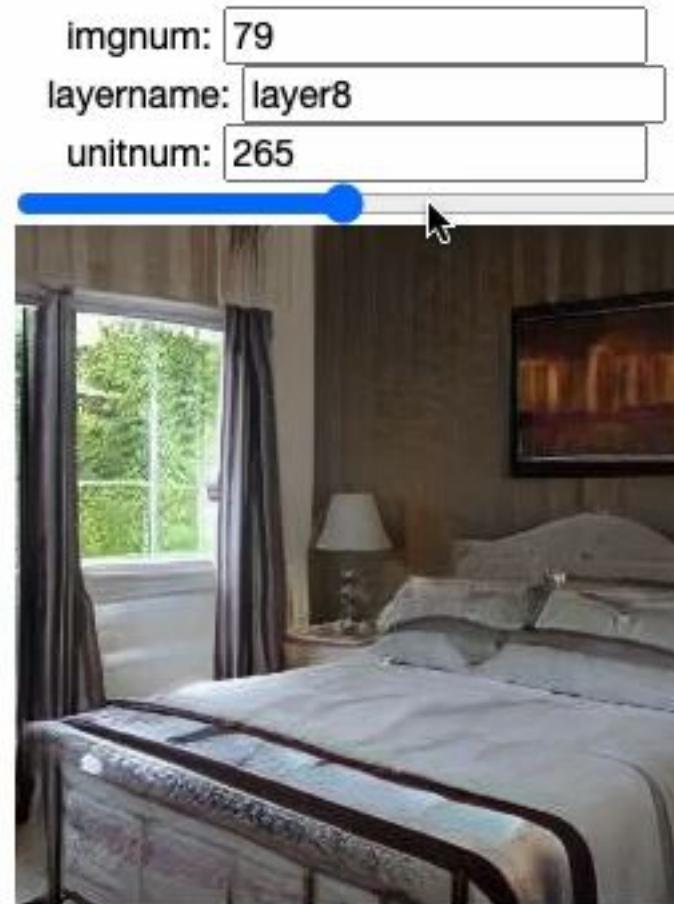
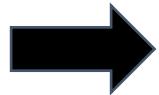
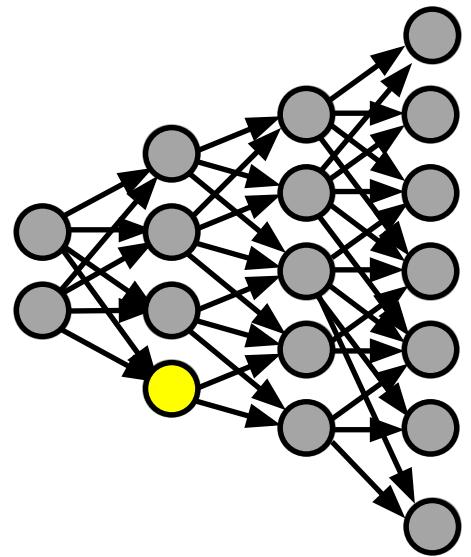
Neuron control within stylegan



Modulating a single neuron



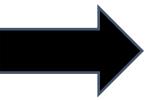
Neuron Control Within StyleGAN



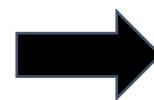
Modulating a single neuron



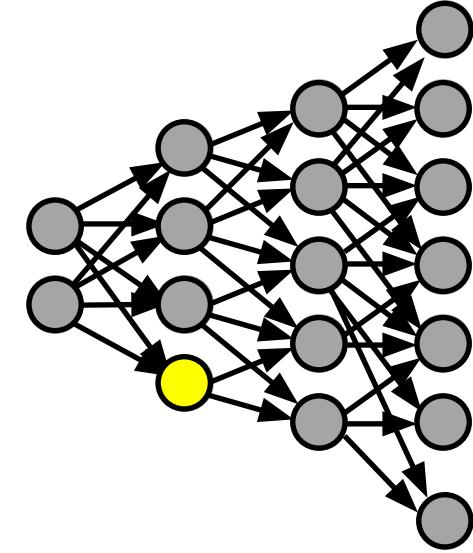
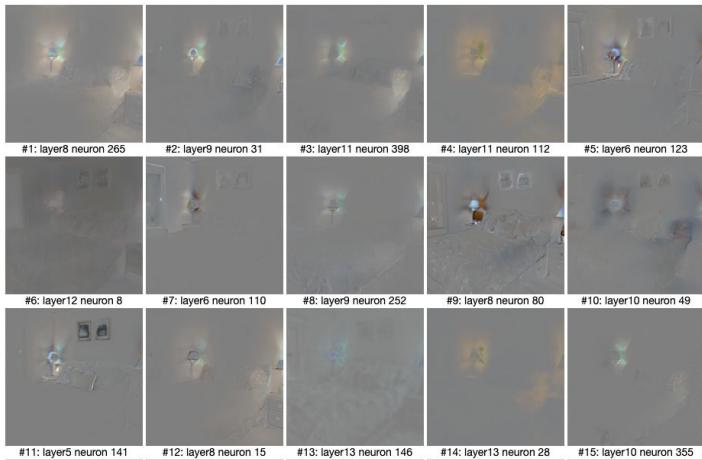
Specify the change



Search for causal
computations



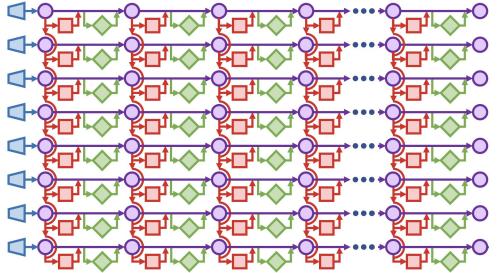
Alter the network



Two Principles for Interpretability

1. Causal tracing reveals mechanisms.
2. Understanding = an ability to make changes.

Training a Large Language Model

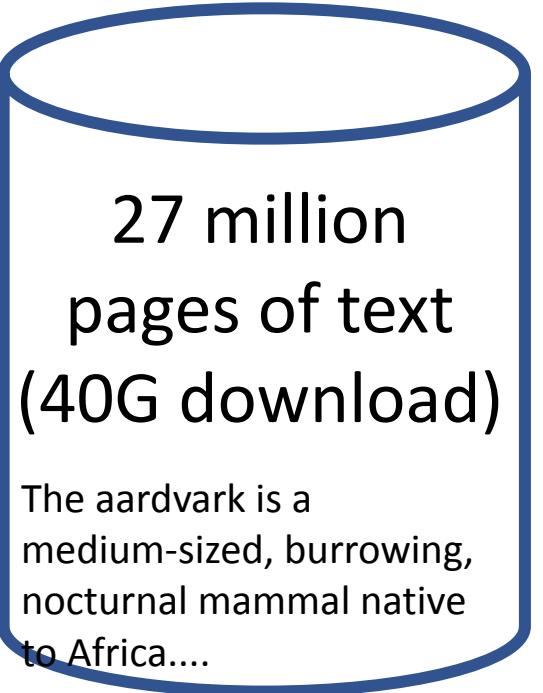
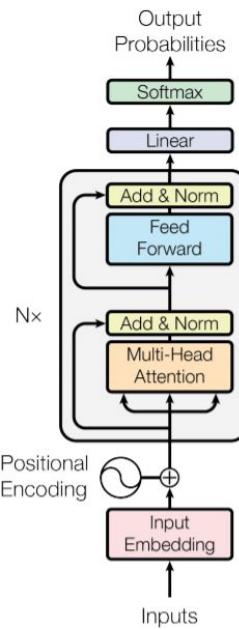


Trained GPT-2 XL



TPU/GPU Hardware

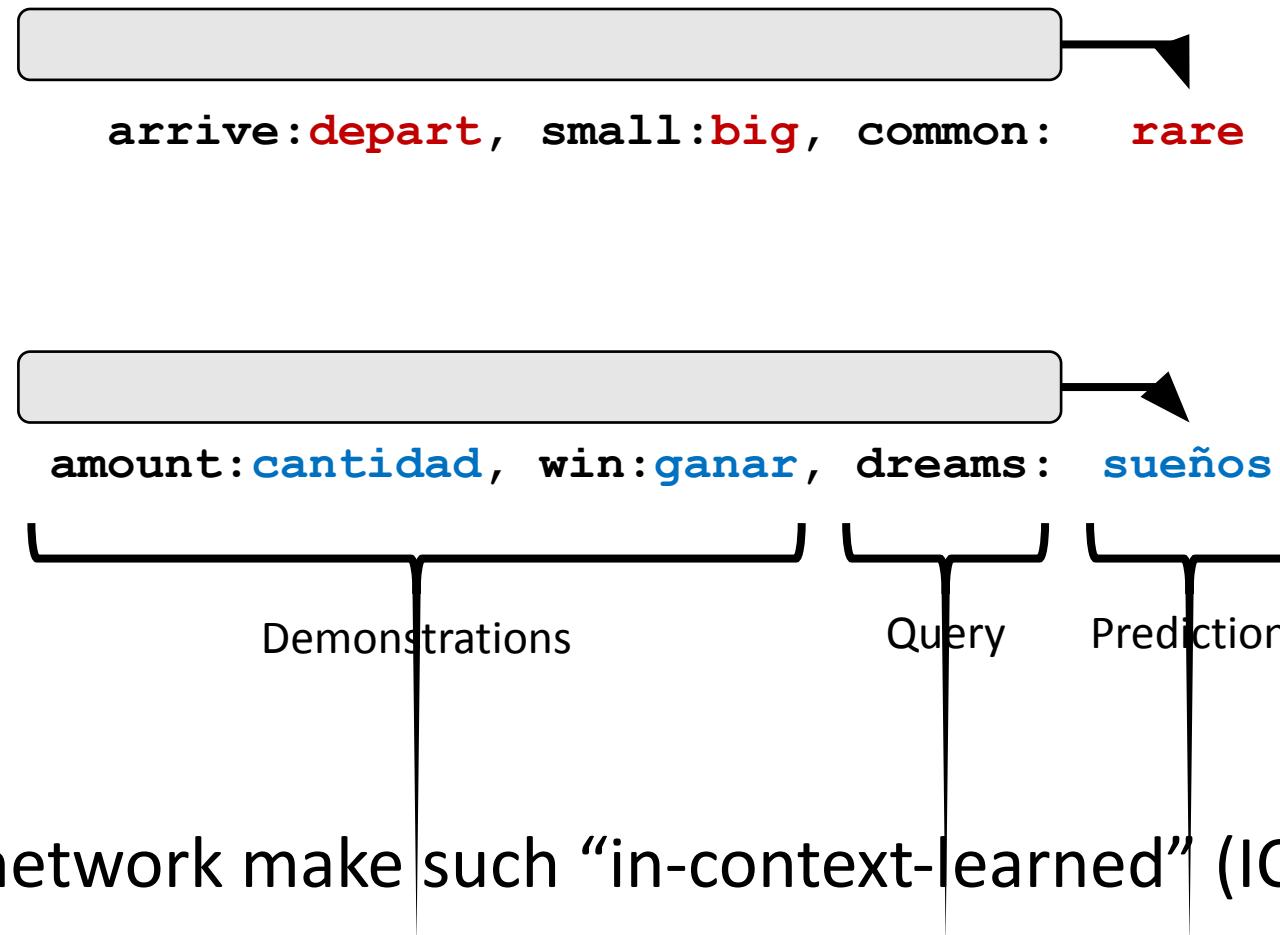
Autoregressive
Transformer
Code



Training Data

[Vaswani 2017, Radford 2018]

After enough training: Metalearning



What Triggers In-Context Learning?



Learned to imitate the
“antonym” function



Learned to imitate the
“Eng □ Spanish” function



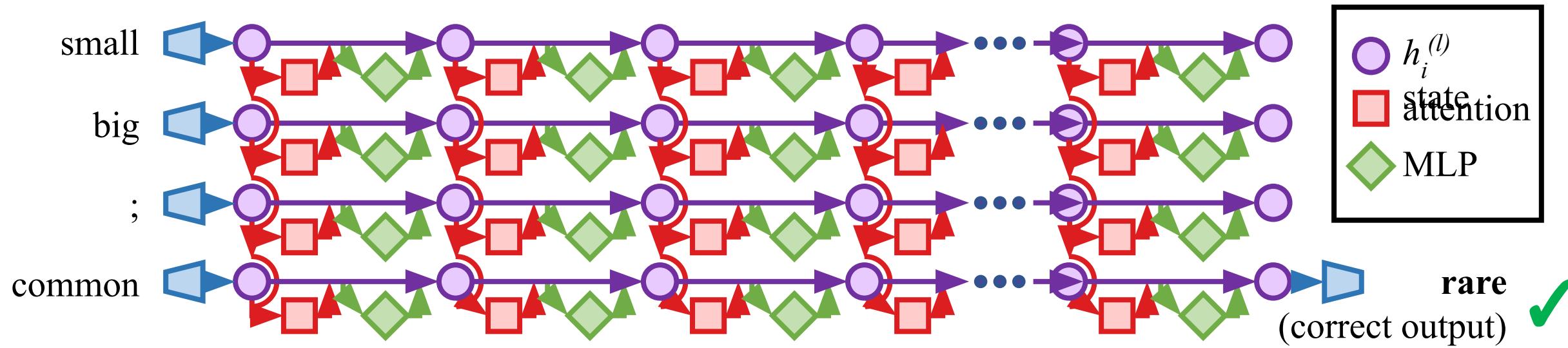
Is there a common mechanism that underlies both of these?

Three Hard Questions for Thought

Ancient questions—

1. What are the **right** of abstractions for understanding thinking? choices? likelihoods? learned tasks? known facts?
2. Is there a **physical basis** to learned concepts? gradients? neurons? **attention-heads**? circuits?
3. How are concepts explicitly **coded**? can we edit knowledge directly?

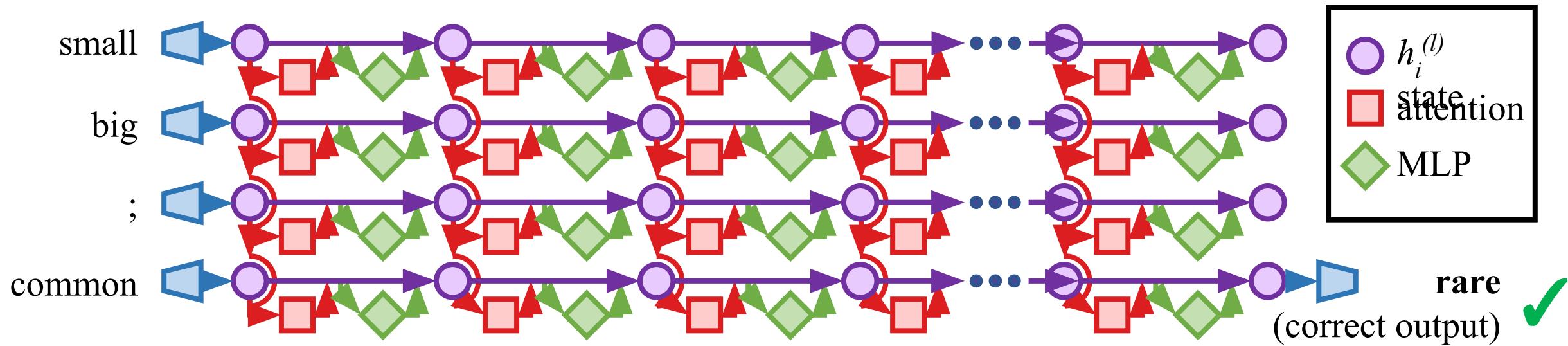
In-Context Learning



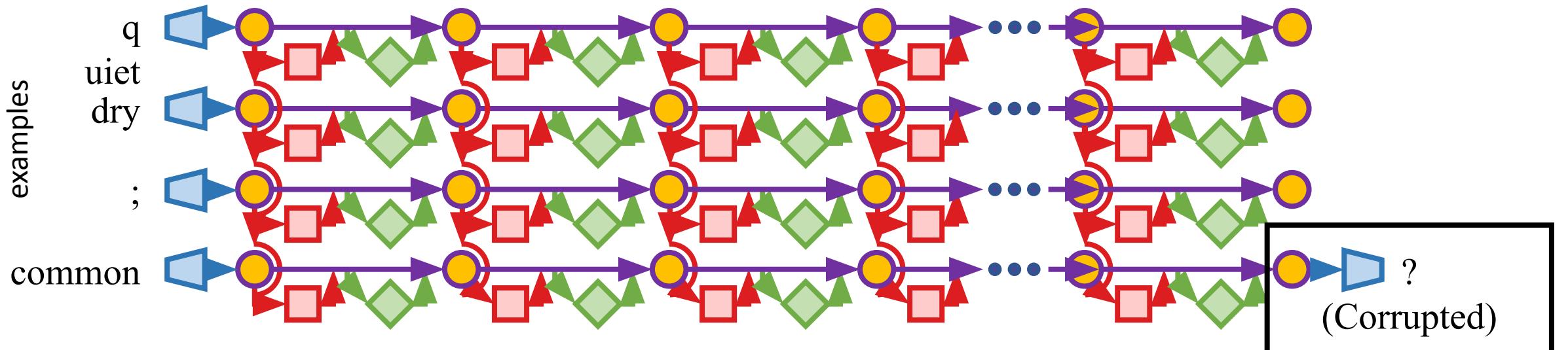
Q: Which component causes ICL?

Idea: Transplant data to identify effects.

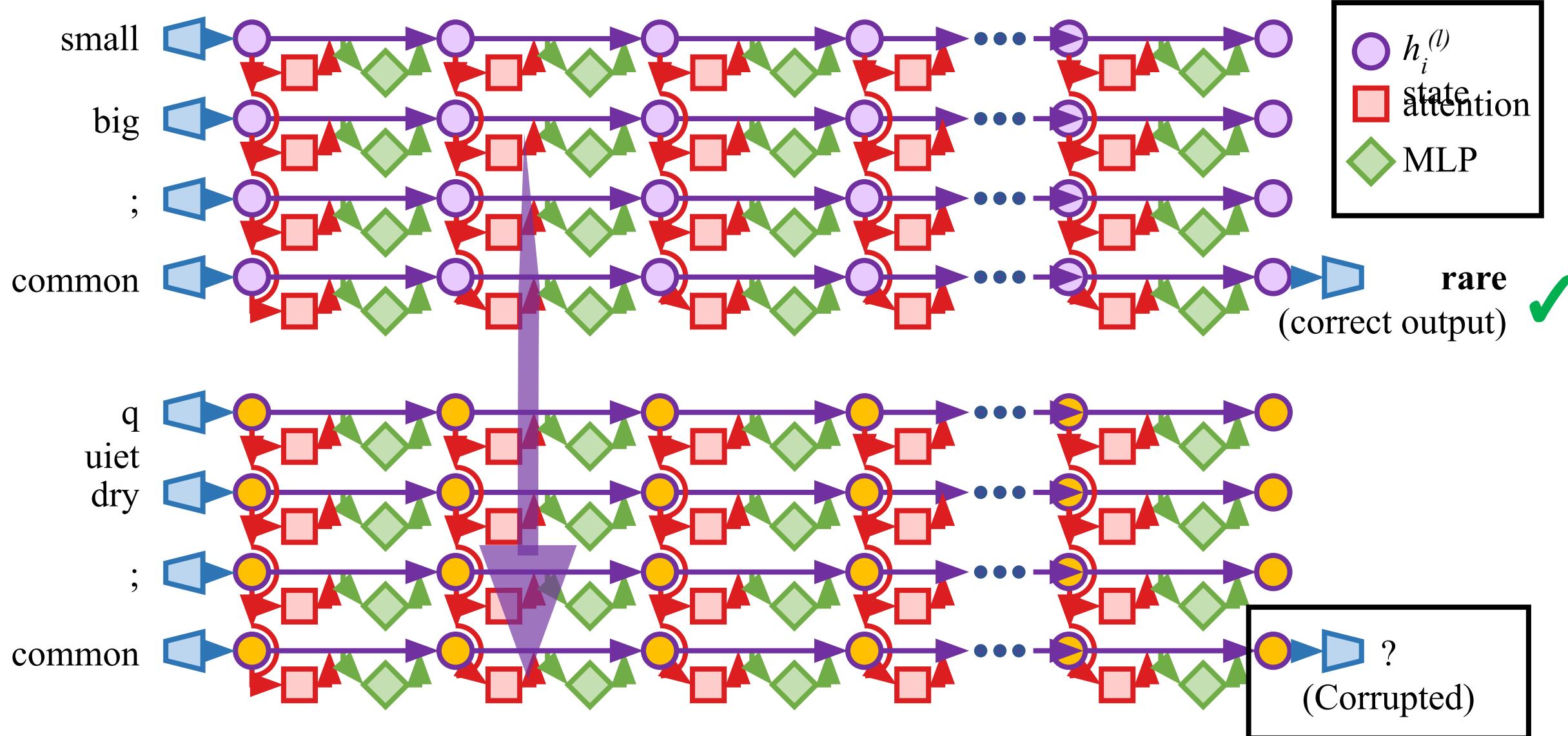
Run the network twice



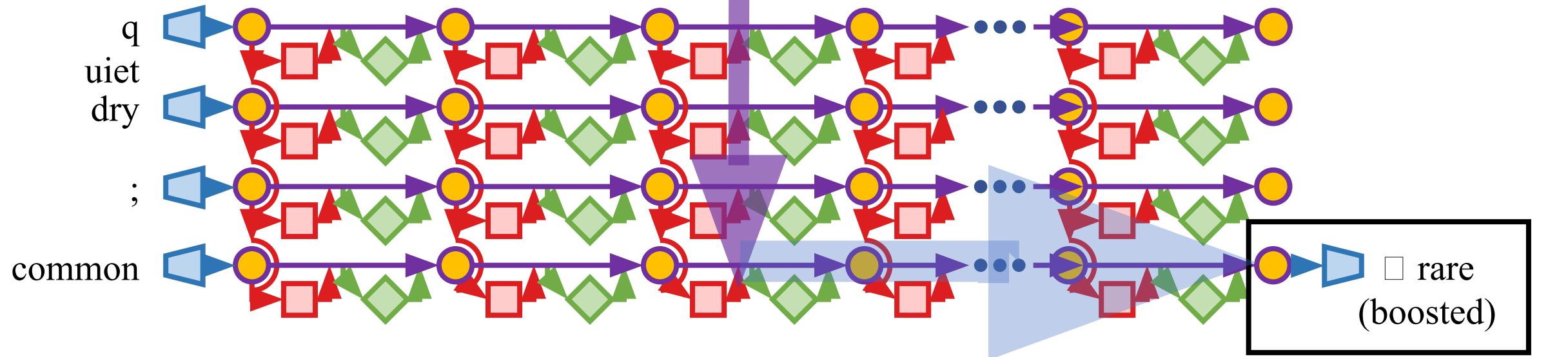
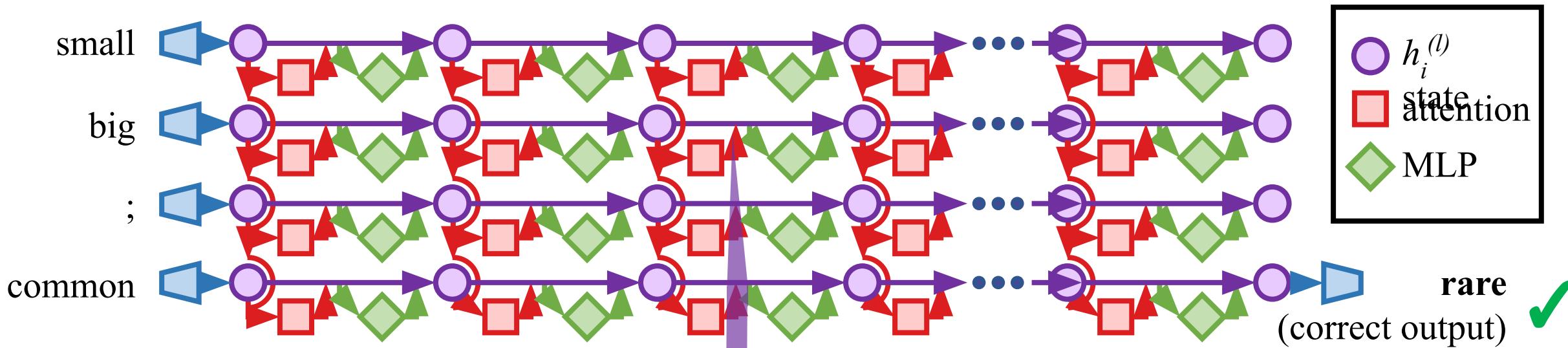
Scrambled examples



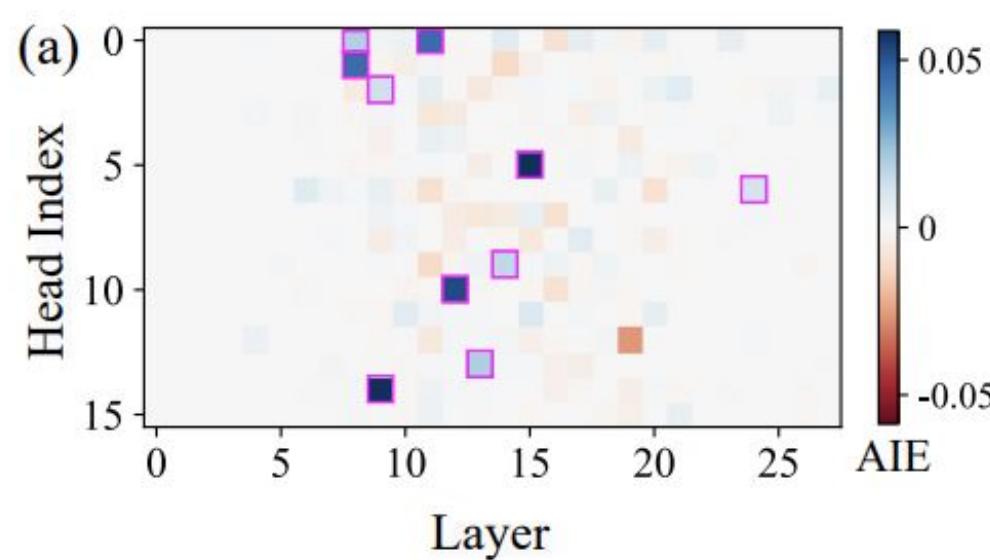
Transplant Attention Head Outputs



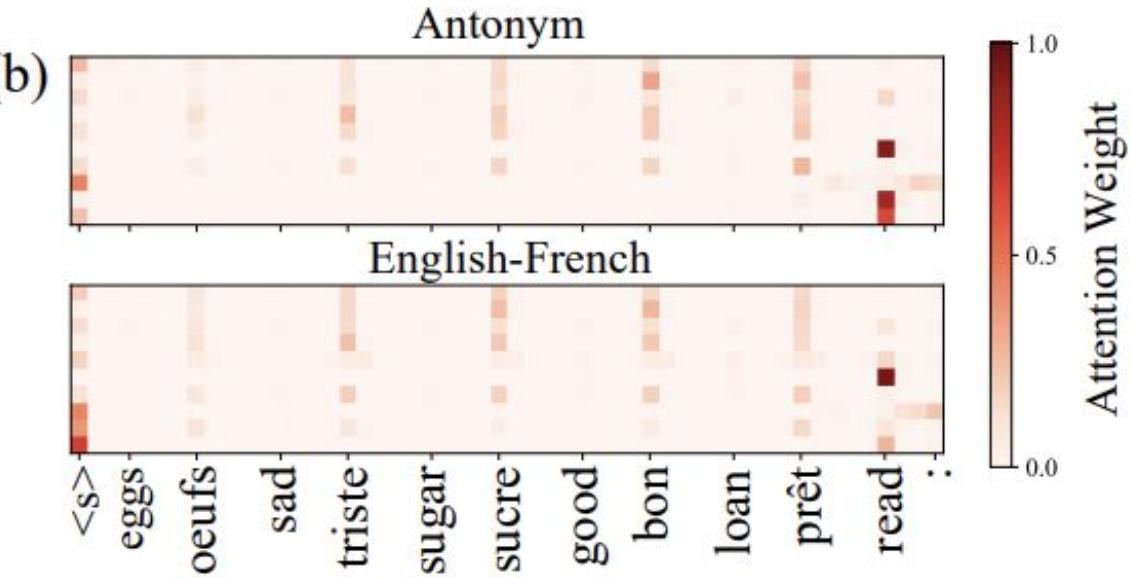
Sort by Desired Causal Effects



Causal Attention Head Search Results

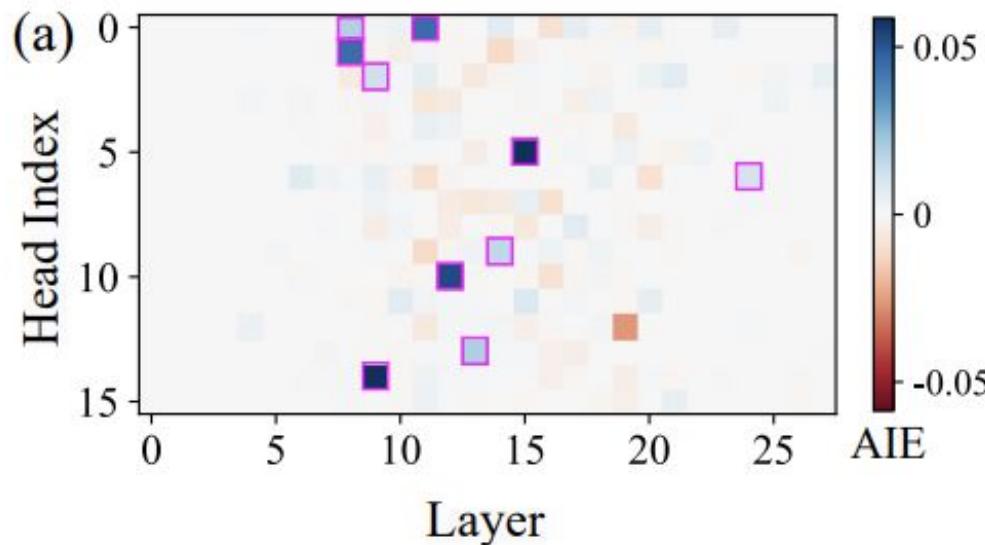


Blue squares are the
10 attention heads which
most strongly boosted
correct ICL results



The same attention heads
have causal effects across
many different tasks,
with similar attention patterns!

Summing Attention Heads



We call this vector v_t the *function vector (FV)* for function t

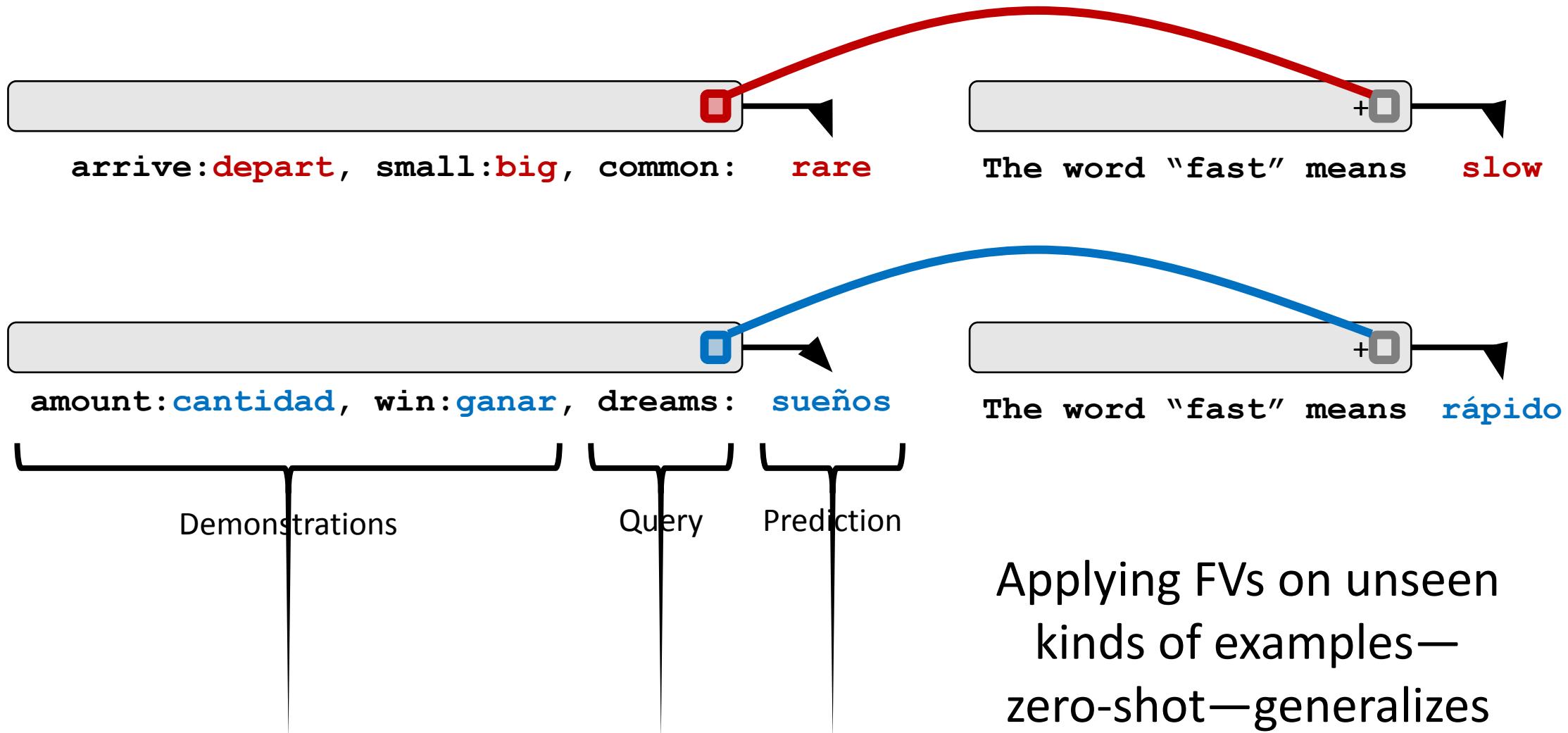
1. Average each attention head over several examples of a task

$$\bar{a}_{\ell j}^t = \frac{1}{|P_t|} \sum_{p_i^t \in P_t} a_{\ell j}(p_i^t).$$

2. Sum up all the averages for the 10 (blue) attention heads

$$v_t = \sum_{a_{\ell j} \in \mathcal{A}} \bar{a}_{\ell j}^t$$

Checking FV Generalization

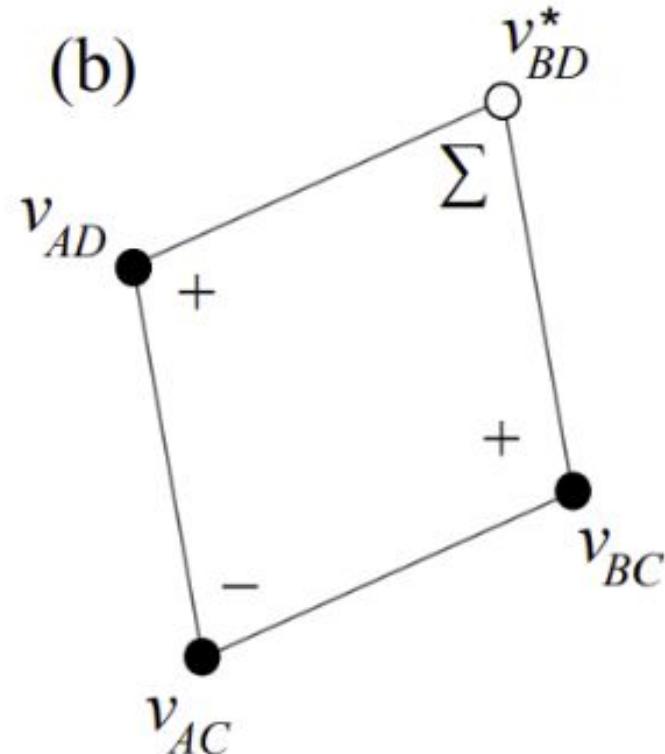


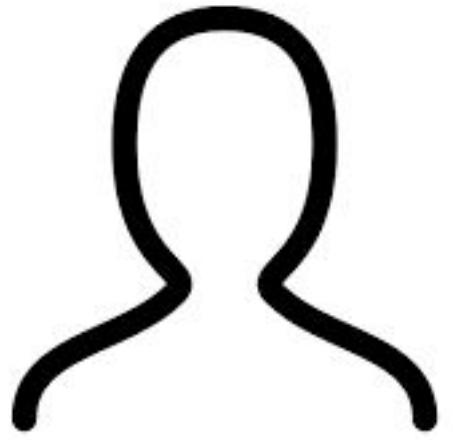
Checking FV Arithmetic

(a) Input: "Italy, Russia, China, Japan, France"

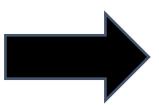
| FV | Task | Expected Output |
|------------|---------------|-----------------|
| v_{AC} | First-Copy | Italy |
| v_{AD} | First-Capital | Rome |
| v_{BC} | Last-Copy | France |
| v_{BD}^* | Last-Capital | ????? |

It can do this task based on the added FV,
without ever having seen any demonstrations!

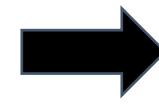




Specify the change

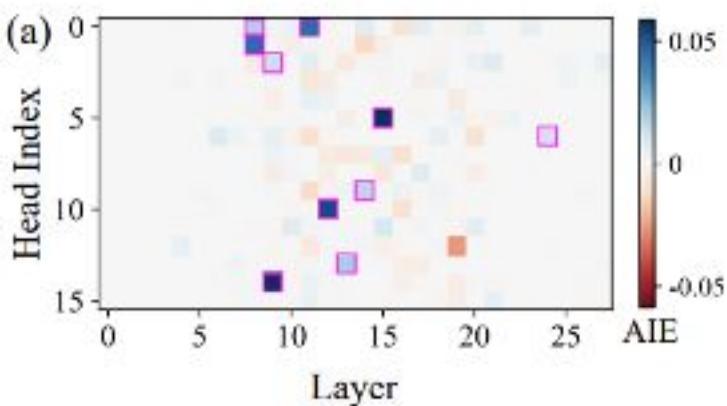


Search for causal computations



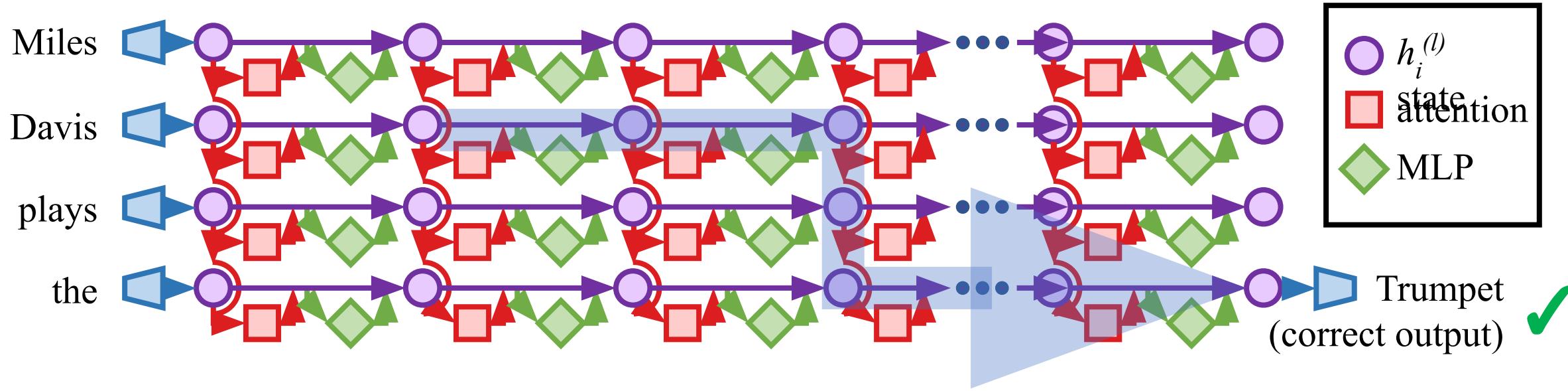
Check generalization

“amount:contidad;
win:ganar”



The word “fast” means rápido

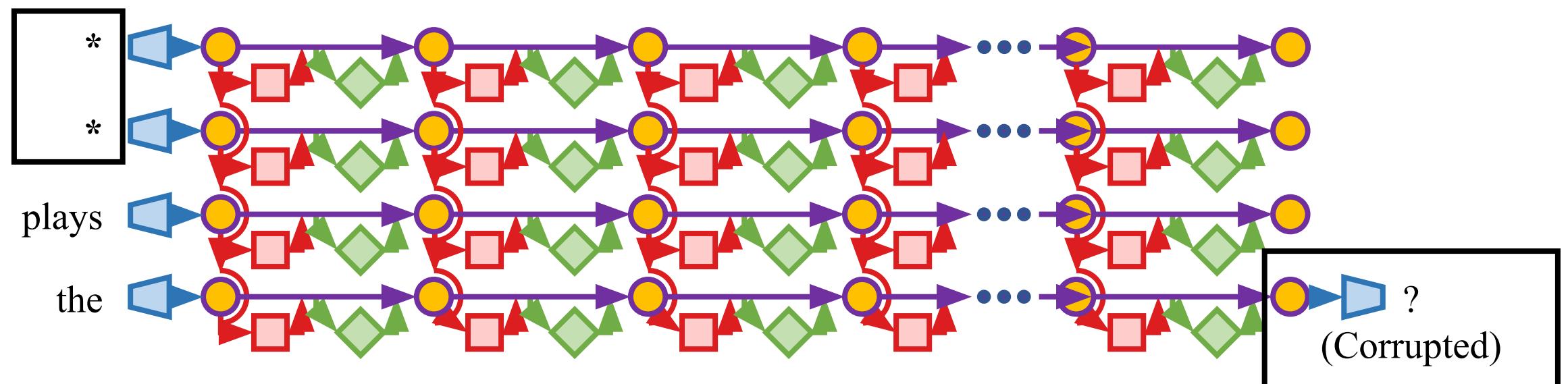
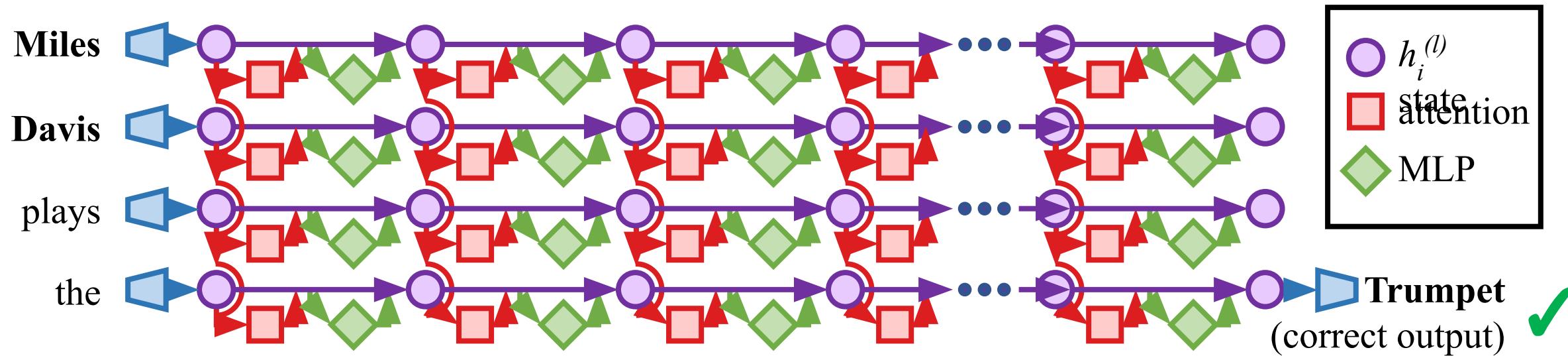
Locating Knowledge



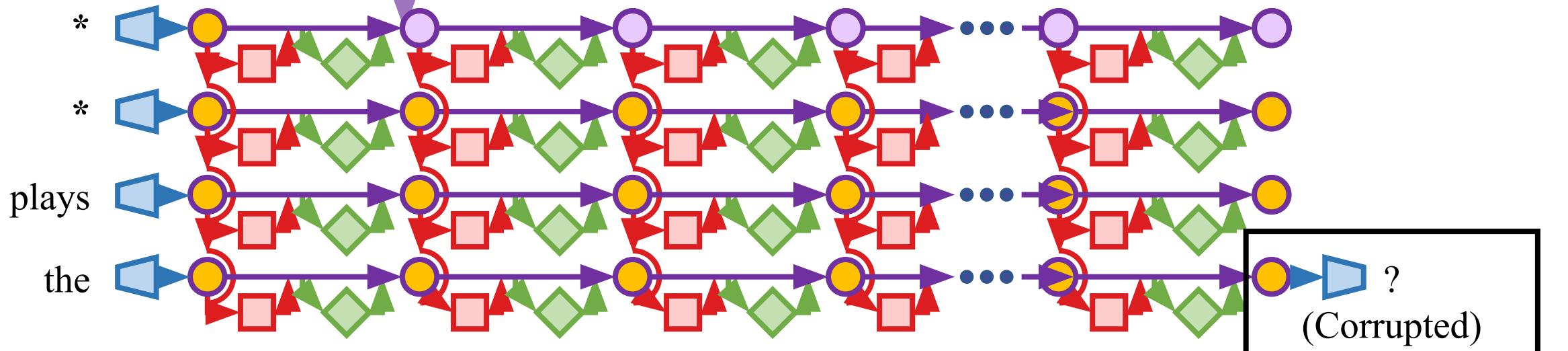
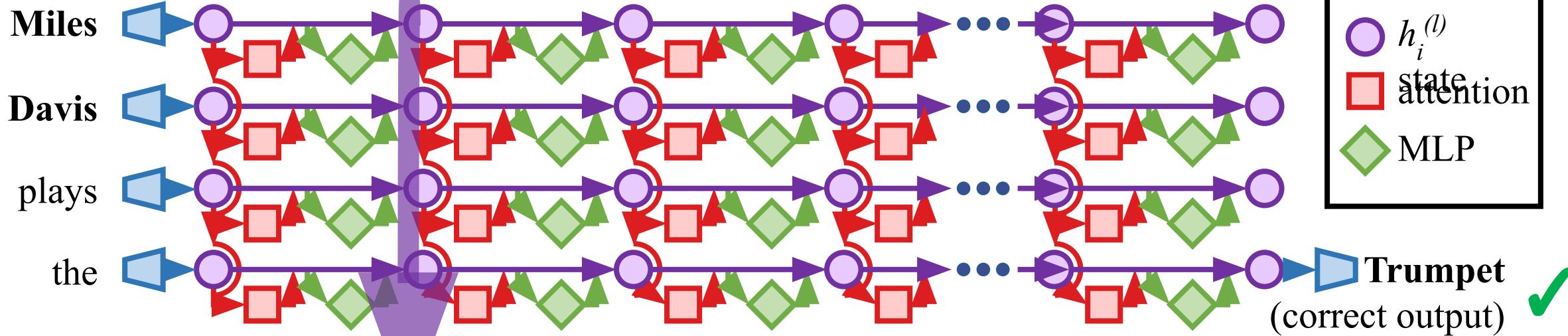
Q: Can we find stored facts?

Again: Transplant data to identify effects.

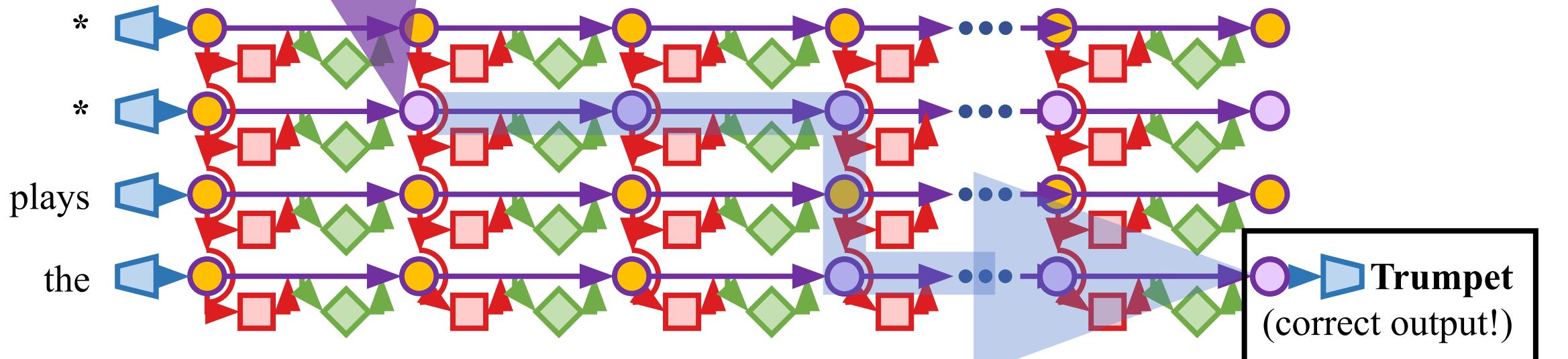
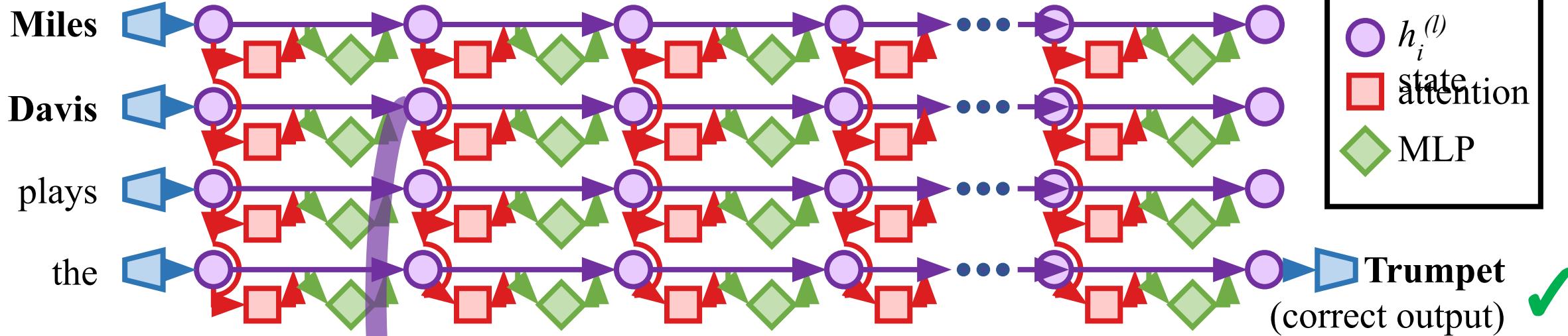
Run the network twice



Transplant Hidden State

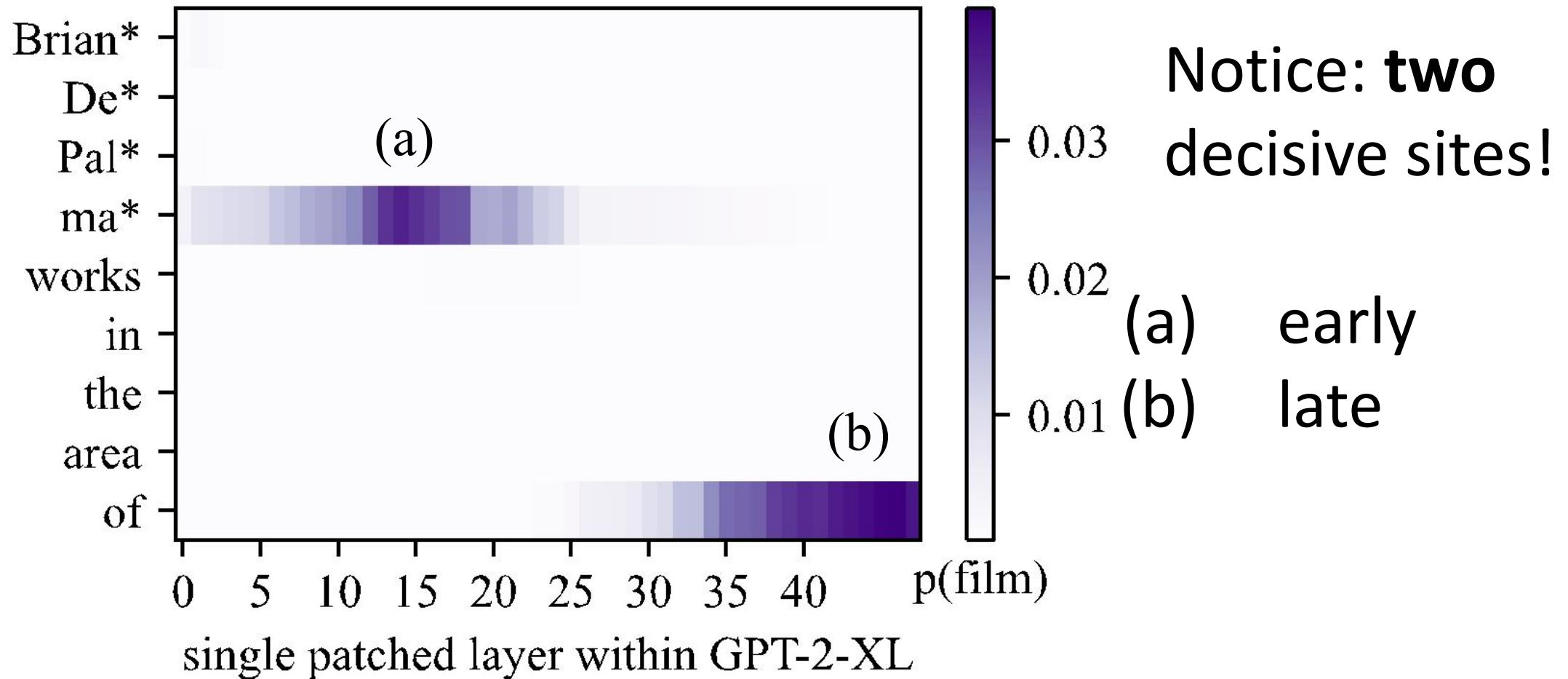


Transplant Hidden State



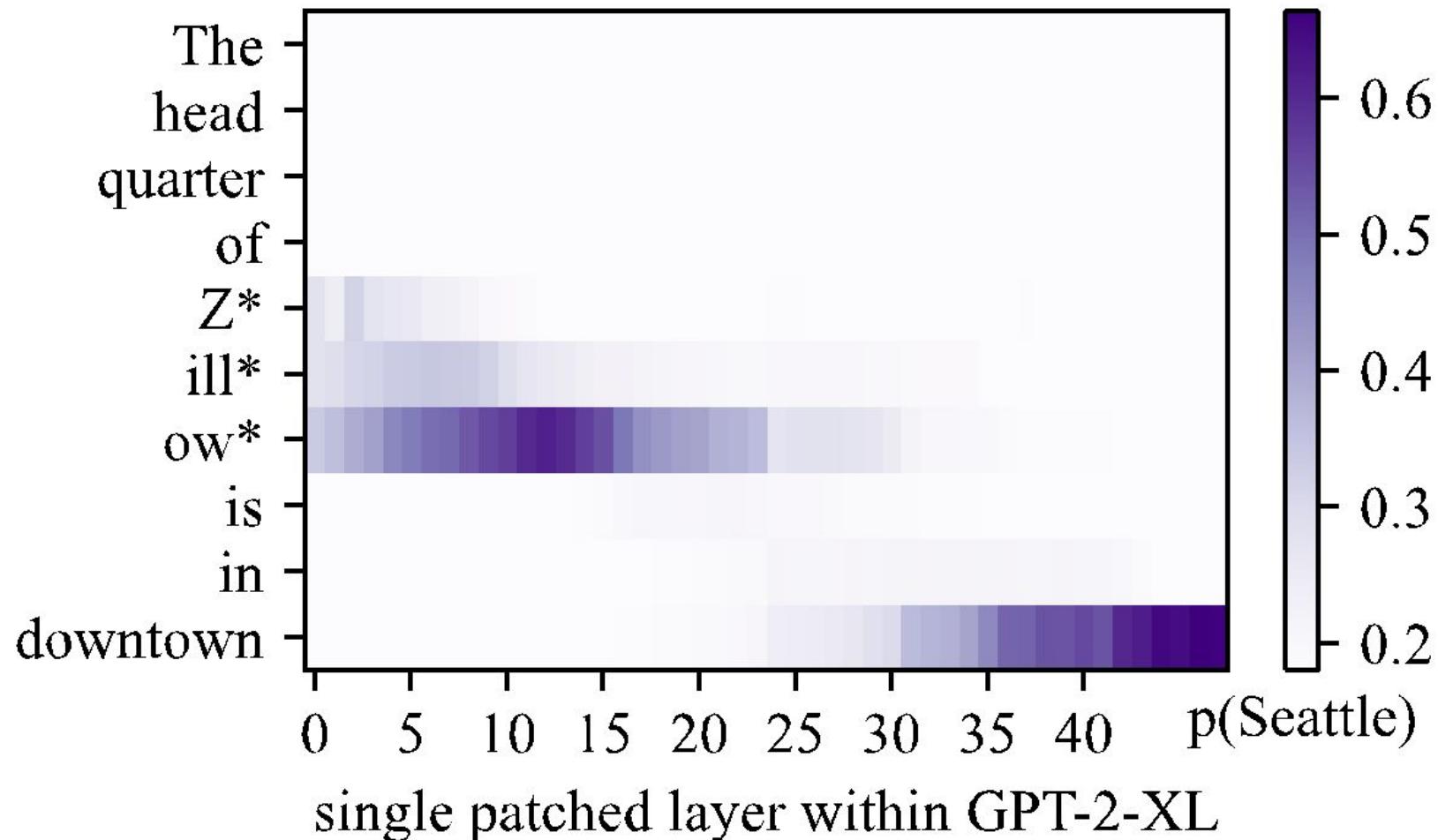
Causal Tracing

Patching hidden state after corrupted input



Causal Tracing

Patching hidden state after corrupted input

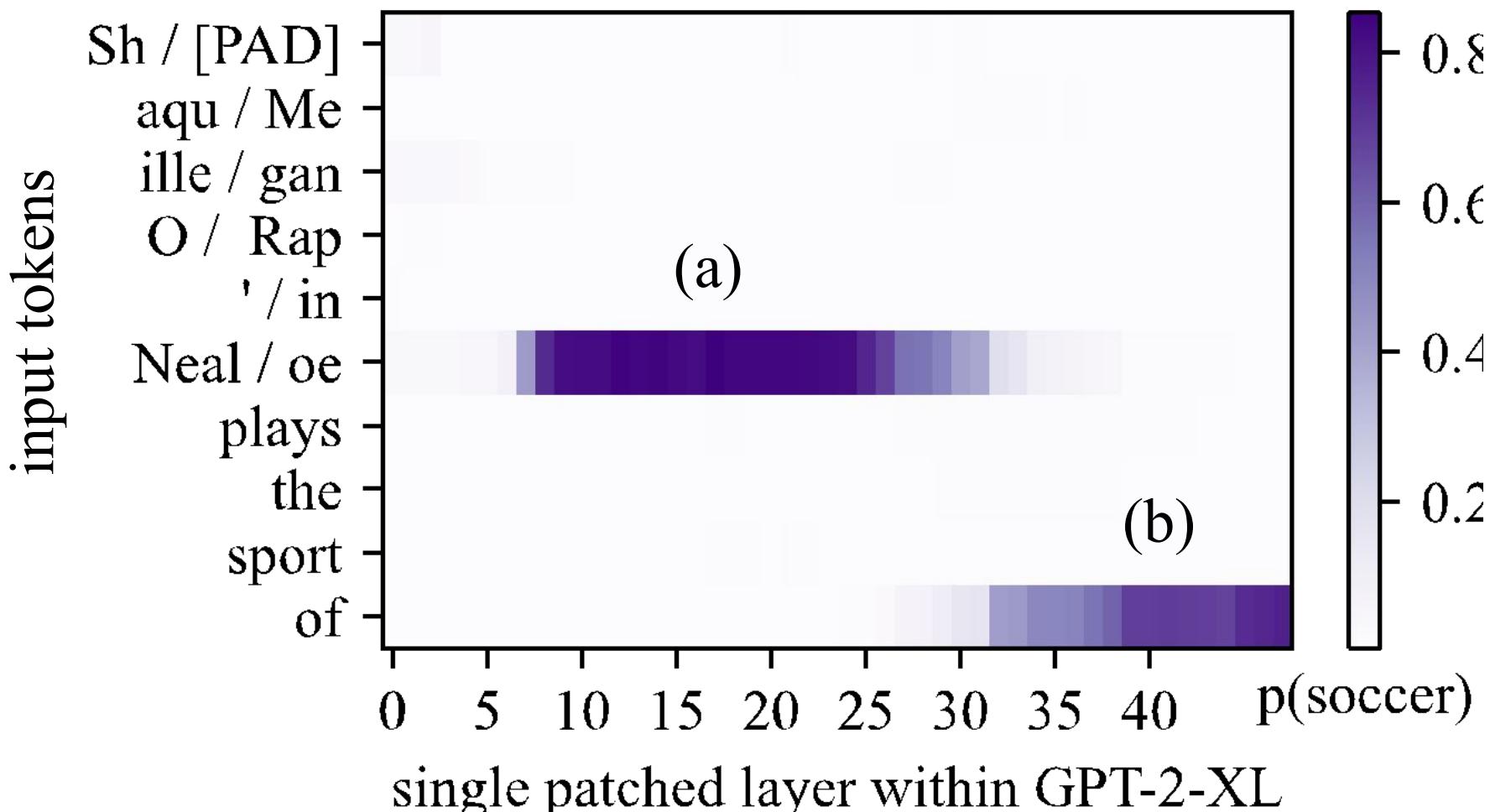


Similar patterns
are seen for a
variety of
factual
statements

Two sites.

Causal Tracing

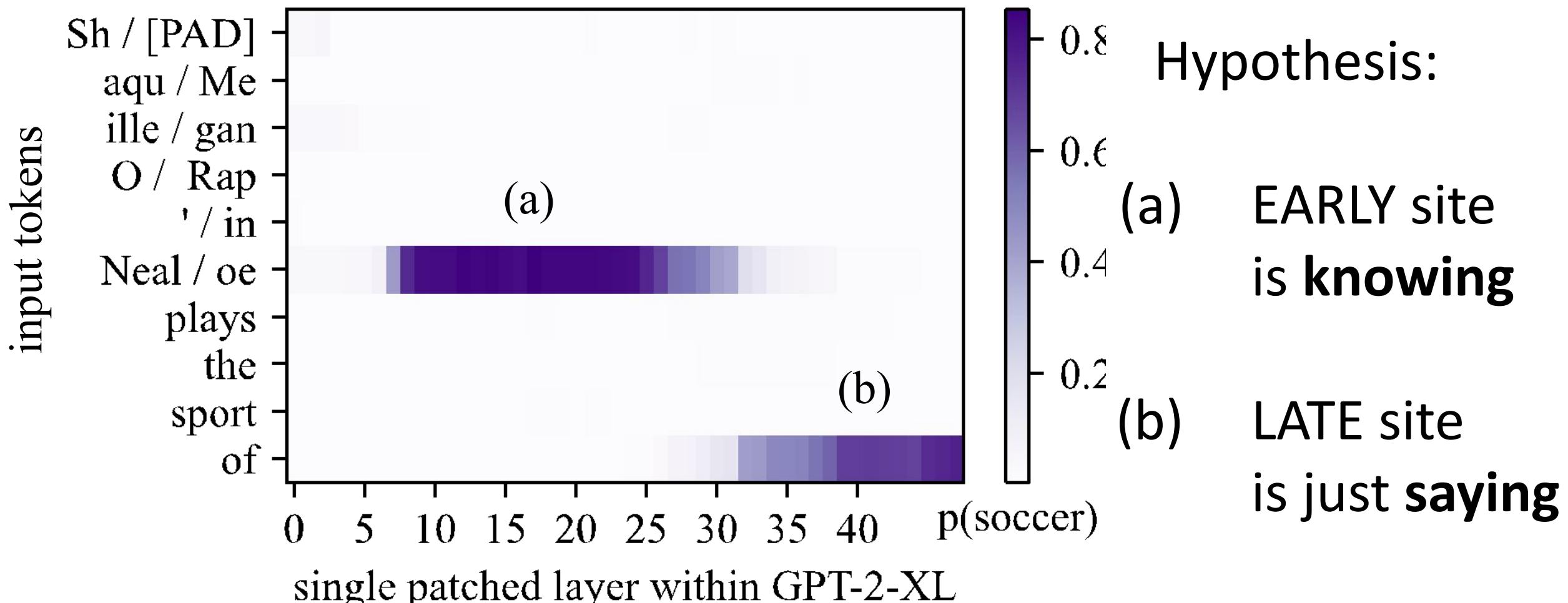
Patching hidden state from Rapinoe to Shaq



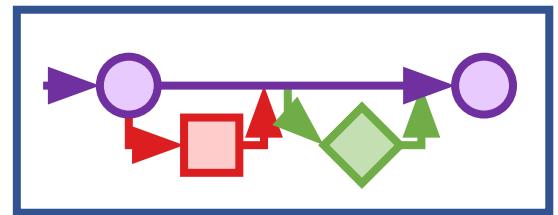
Copying a state
between two
uncorrupted
sentences:
Now Shaq plays
soccer.

Causal Tracing

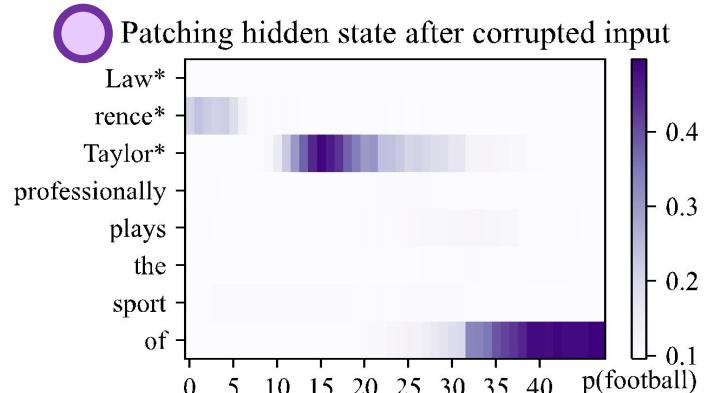
Patching hidden state from Rapinoe to Shaq



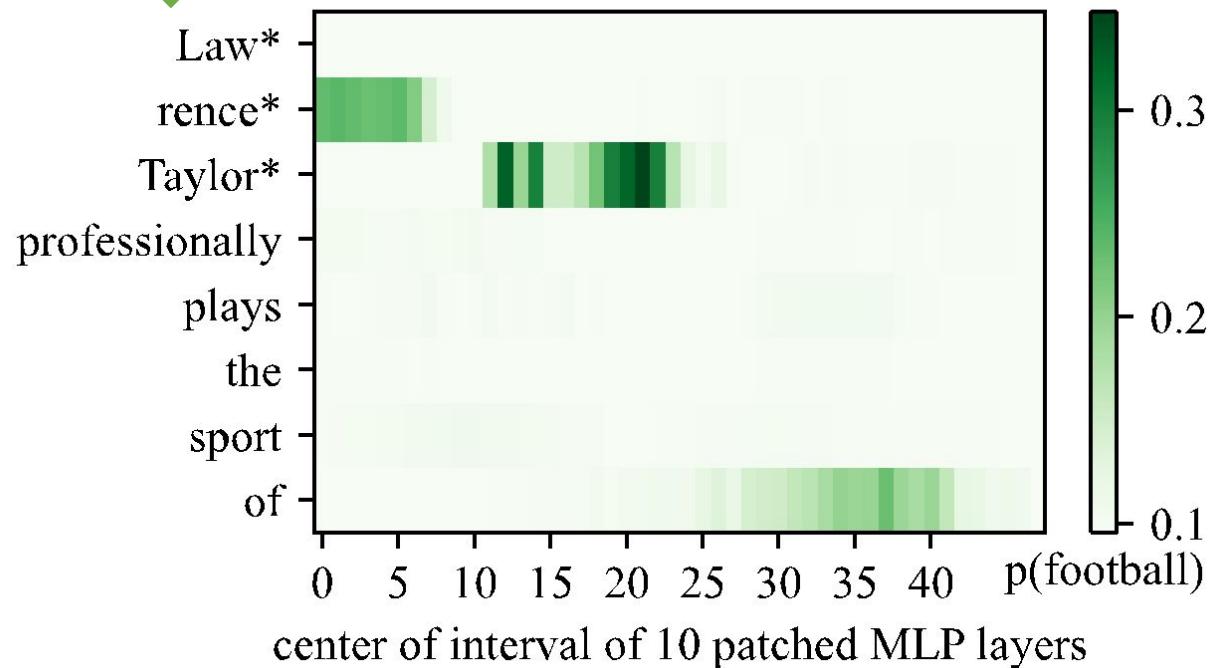
Causal Tracing



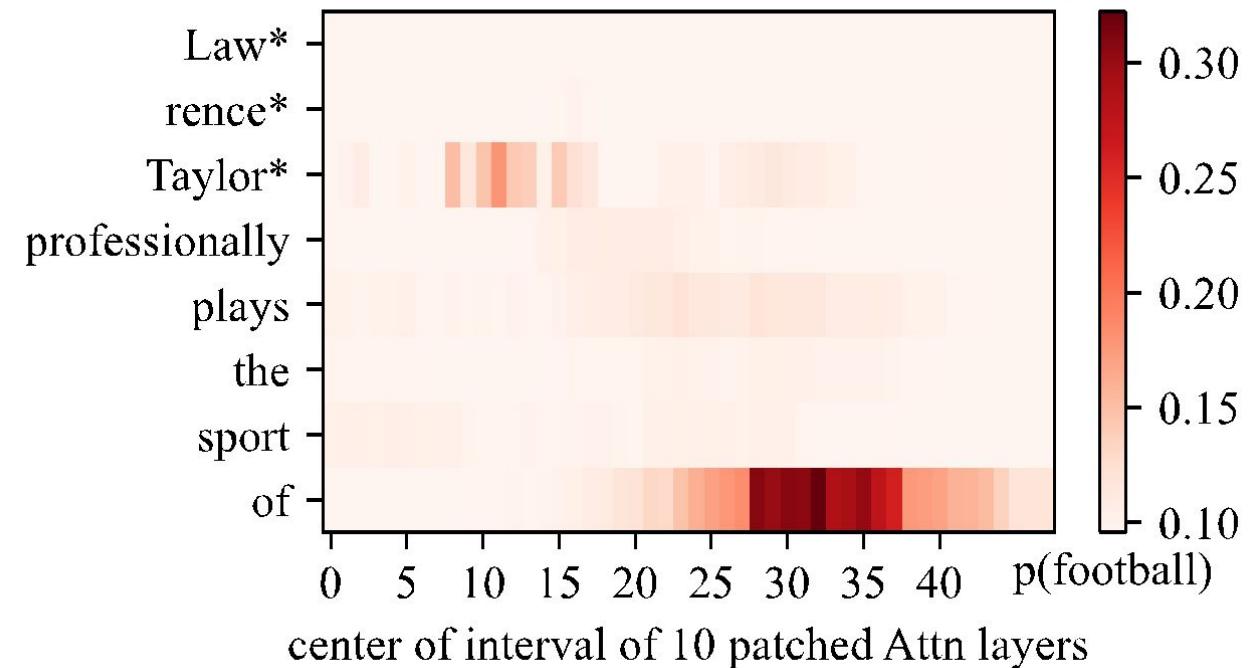
Instead of patching h
we can patch MLP or Attention



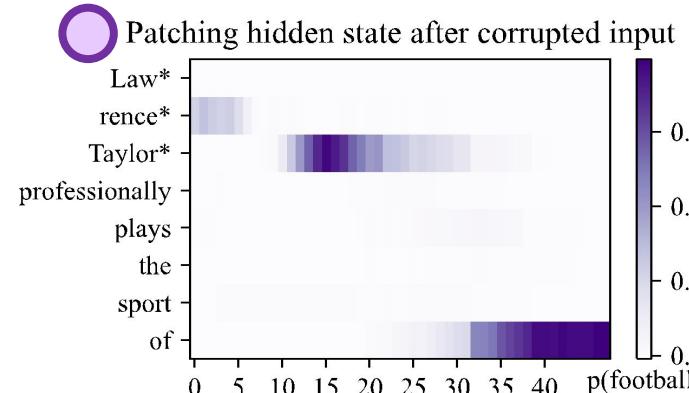
Patching MLP state after corrupted input



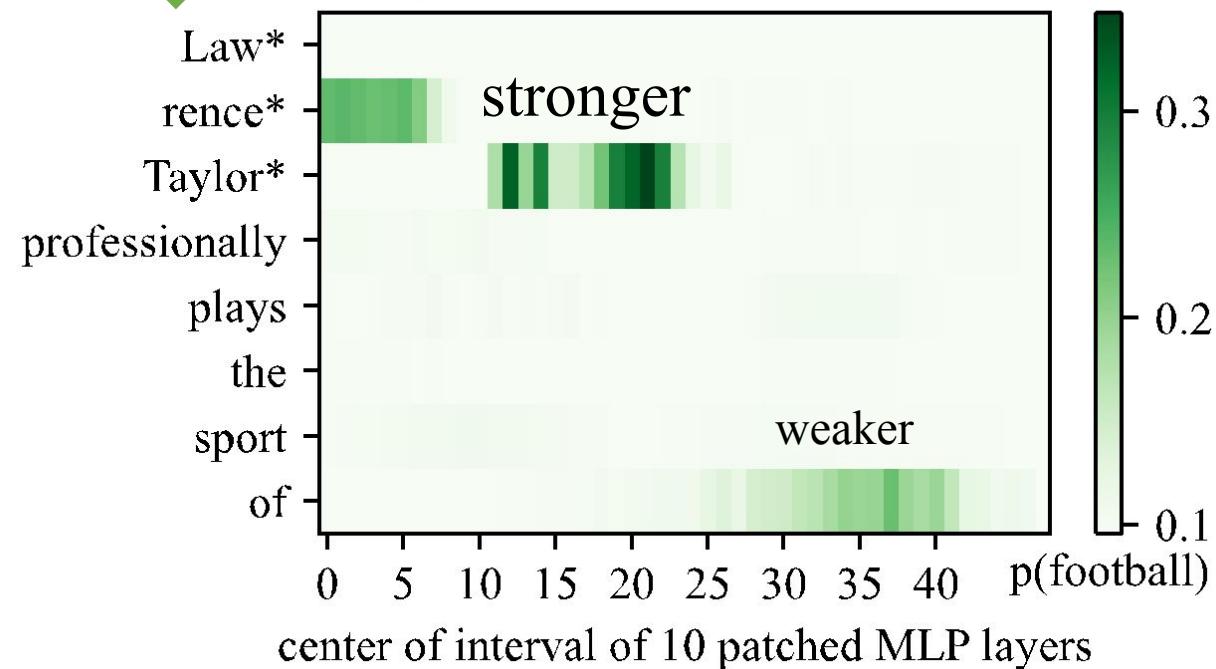
Patching Attn state after corrupted input



Causal Tracing

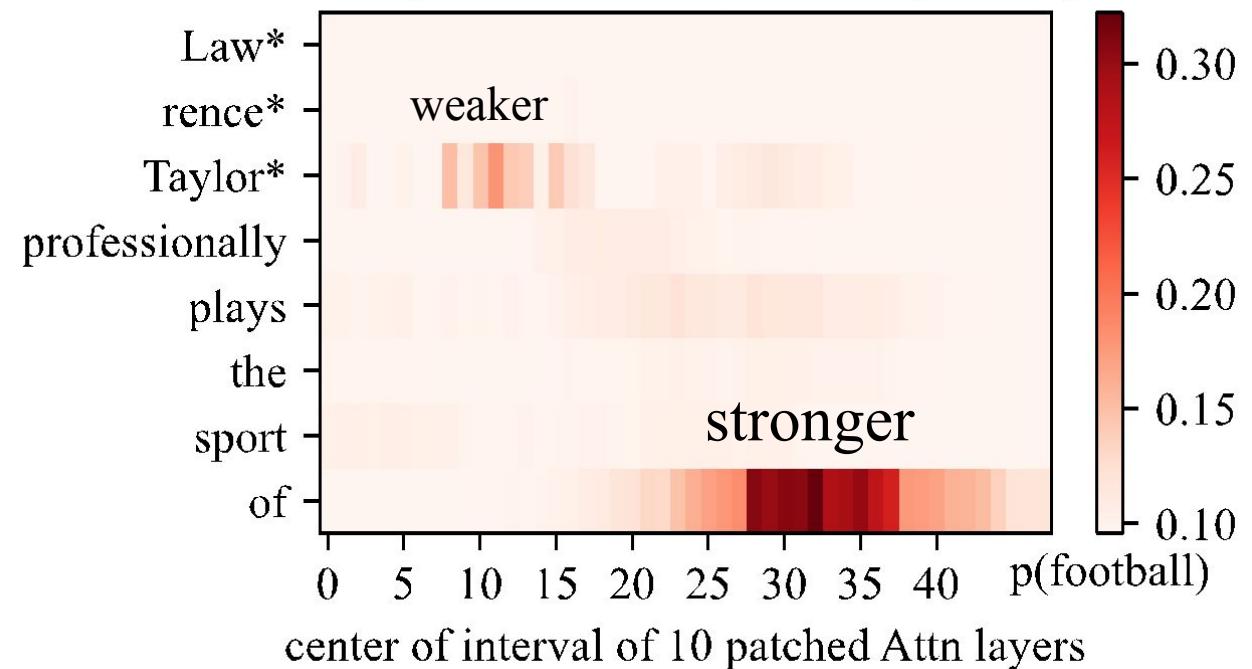


◆ Patching MLP state after corrupted input



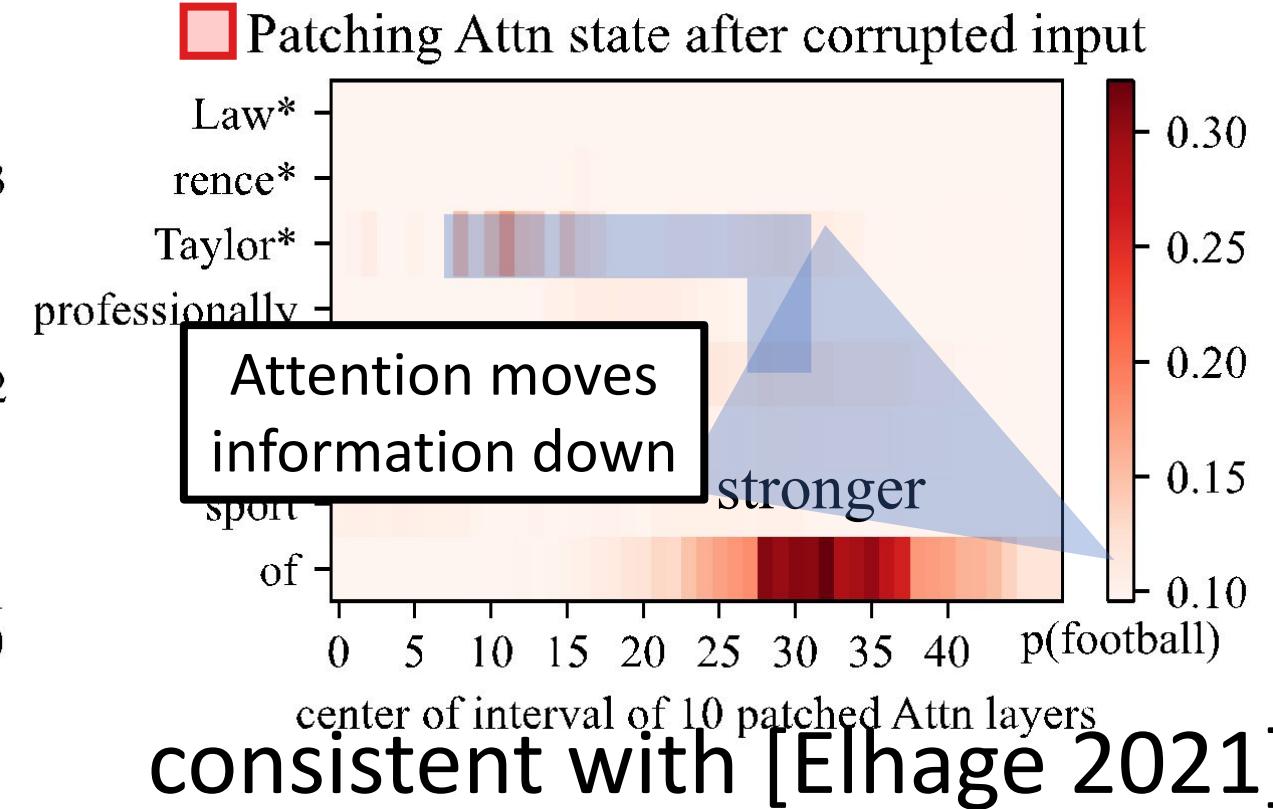
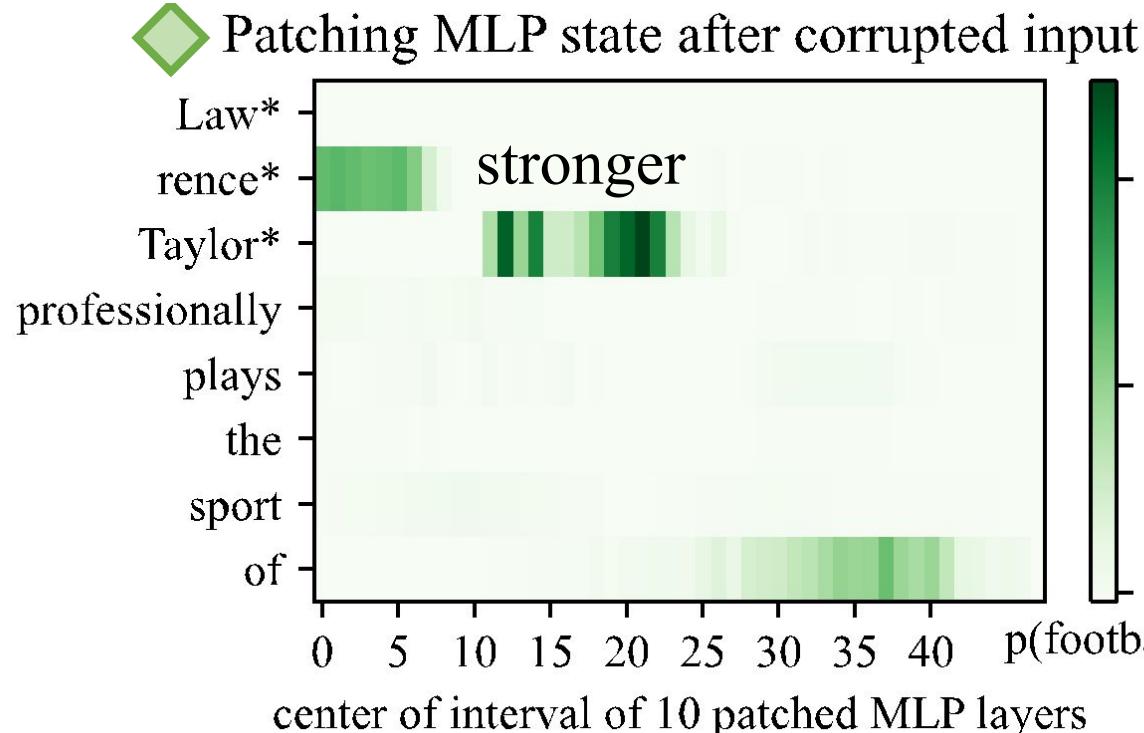
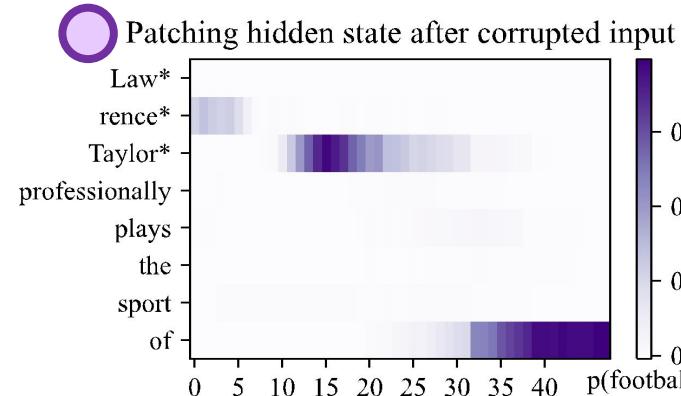
center of interval of 10 patched MLP layers

■ Patching Attn state after corrupted input



center of interval of 10 patched Attn layers

Causal Tracing

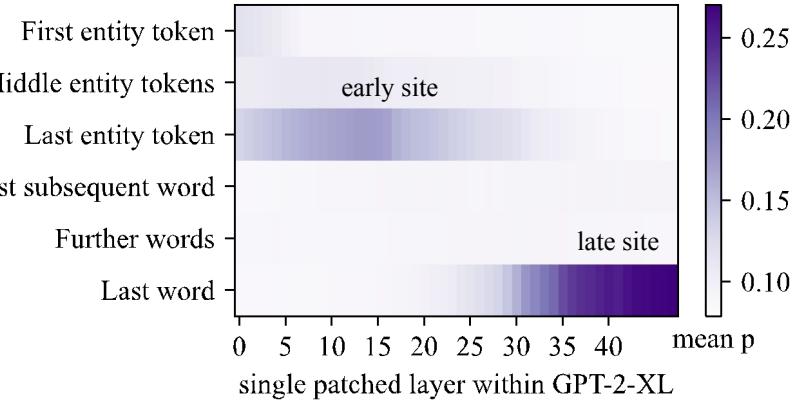


Causal Tracing

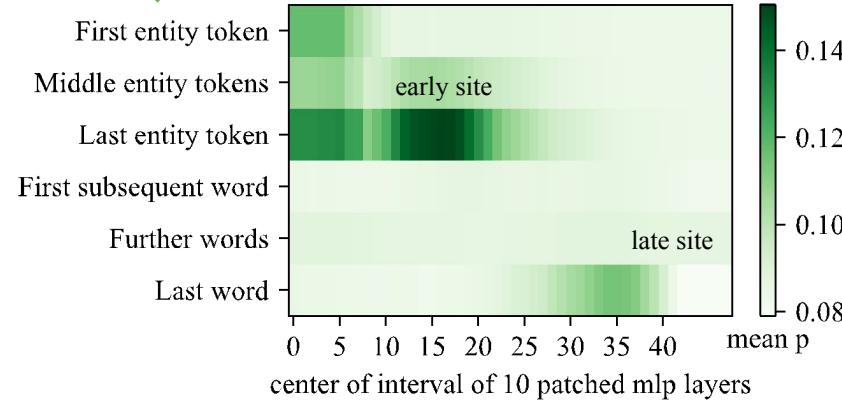
Average over 1000 factual statements



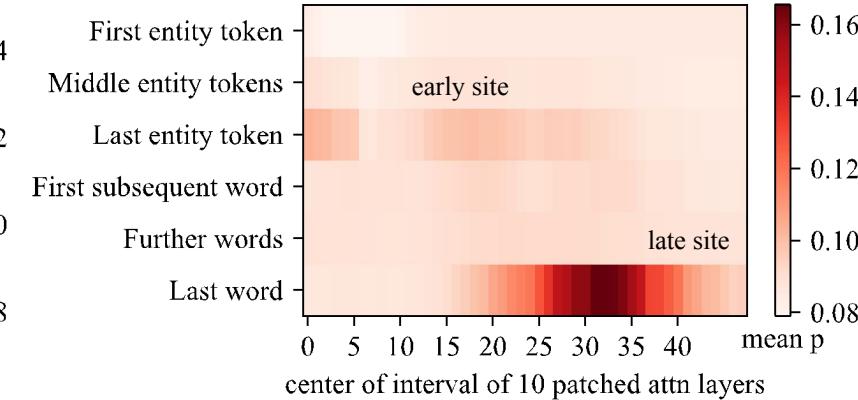
Avg impact of each hidden state over 1000 prompts



Avg MLP lookup impact over 1000 prompts



Avg Attn lookup impact over 1000 prompts



Systematic
localized effects:

- Two sites
- Last subject token
- Midlayer MLP modules

Two Principles for Interpretability

✓ 1. Causal tracing reveals mechanisms.

2. Understanding = an ability to make changes.

Two Principles for Interpretability

- ✓ 1. Causal tracing reveals mechanisms.
- 2. Understanding = an ability to make changes.

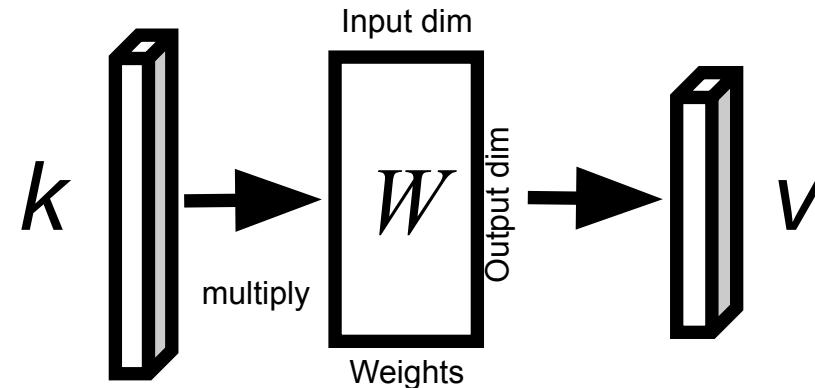
Changing Knowledge

Associative Memory View of a Layer

A layer can act as a memory.

$$\{k_1 \square v_1, k_2 \square v_2, k_3 \square v_3, \dots, k_N \square v_N\}$$

We can find W so that $\forall i, v_i \approx W k_i$

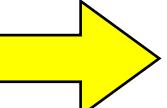


[Kohonen 1972, Anderson 1972]

Associative Memory View of a Layer

We wish to set $k_* \square v_*$ while still minimizing error in old $k \square v$

$$\rightarrow W_1 = \arg \min_W \|V_1 - WK_1\|^2$$

 where $V_1 = \begin{bmatrix} \vdots & \vdots \\ V_0 & v_* \\ \vdots & \vdots \end{bmatrix}$ $K_1 = \begin{bmatrix} \vdots & \vdots \\ K_0 & k_* \\ \vdots & \vdots \end{bmatrix}$

This is still Least Squares, and has this solution:

$$W_1 K_1 K_1^T = V_1 K_1^T \quad [\text{New form!}]$$

$$W_1 K_0 K_0^T + W_1 k_* k_*^T = V_0 K_0^T + v_* k_*^T$$

ROME/MEMIT memory update

Subtracting original LS solution from new solution cancels terms.

$$W_1 K_0 K_0^T + W_1 k_* k_*^T = V_0 K_0^T + v_* k_*^T$$

$$W_0 K_0 K_0^T = V_0 K_0^T$$

define: $r = v_* - W_0 k_*^T$, $C_0 = K_0 K_0^T$

$$W_1 = W_0 + r k_*^T (C_0 + k_* k_*^T)^{-1}$$

The solution can generalize to a matrix of v_* and k_*

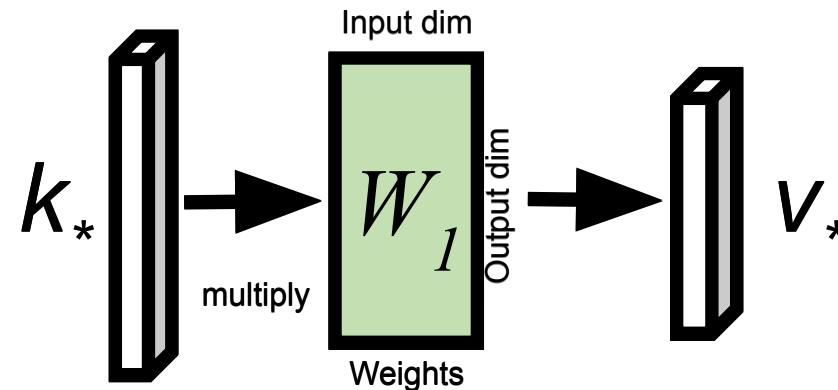
[New formulation – Meng 2022]

Associative Memory View of a Layer

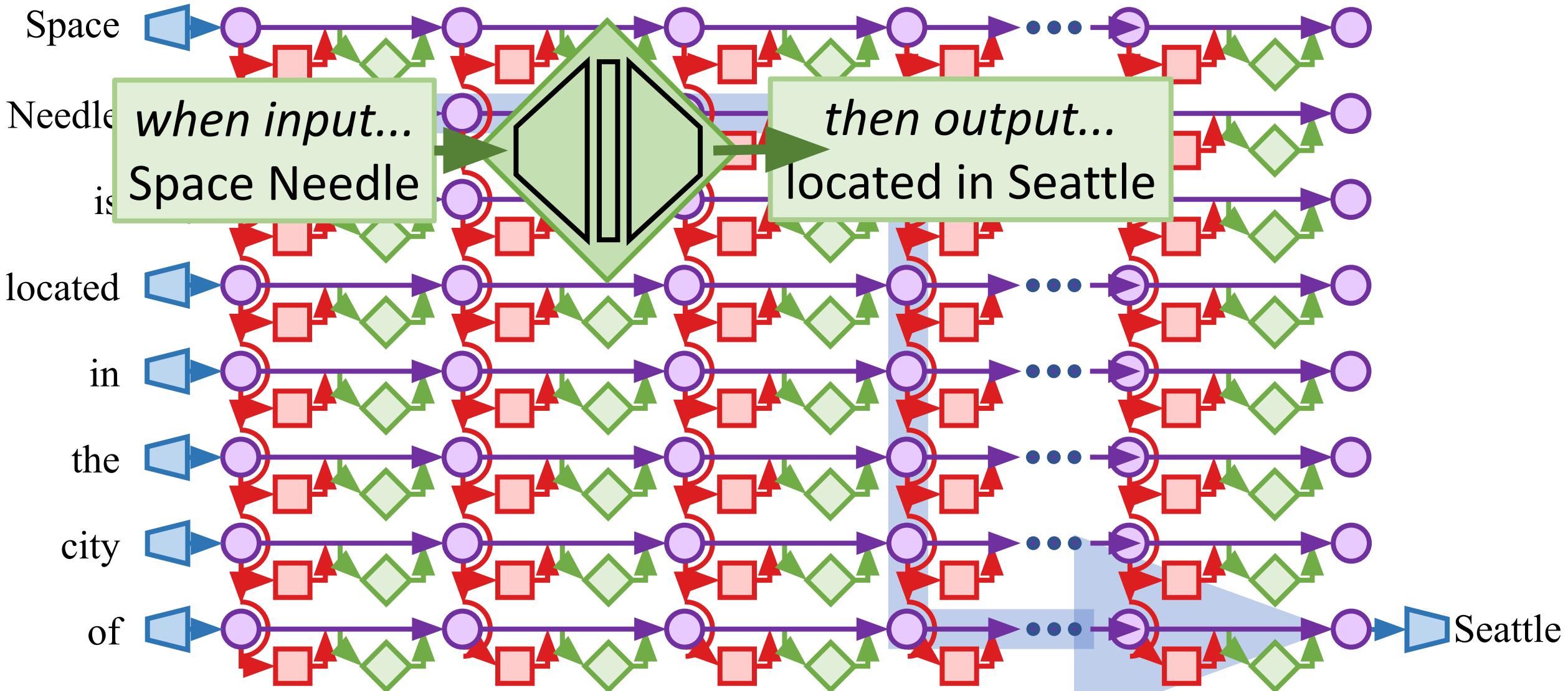
A layer can act as a memory.

Capacity based on the number of columns.

$$W_1 = W_0 + \Lambda(C^{-1} k_*)^T$$

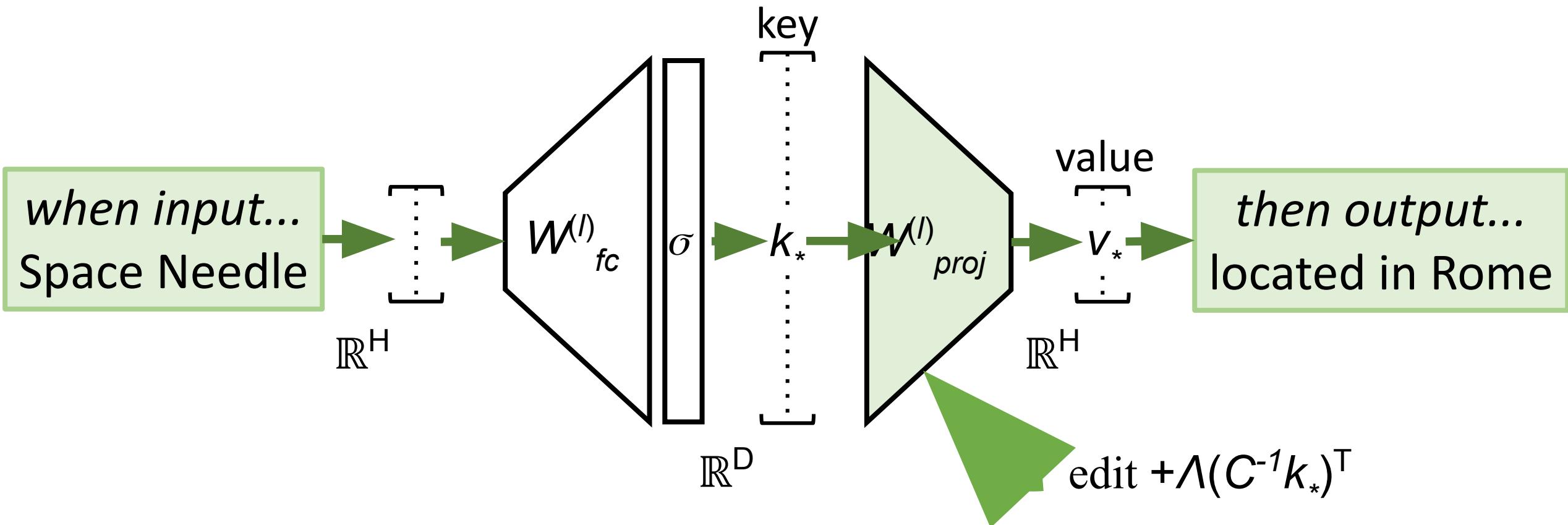


Editing an MLP Memory



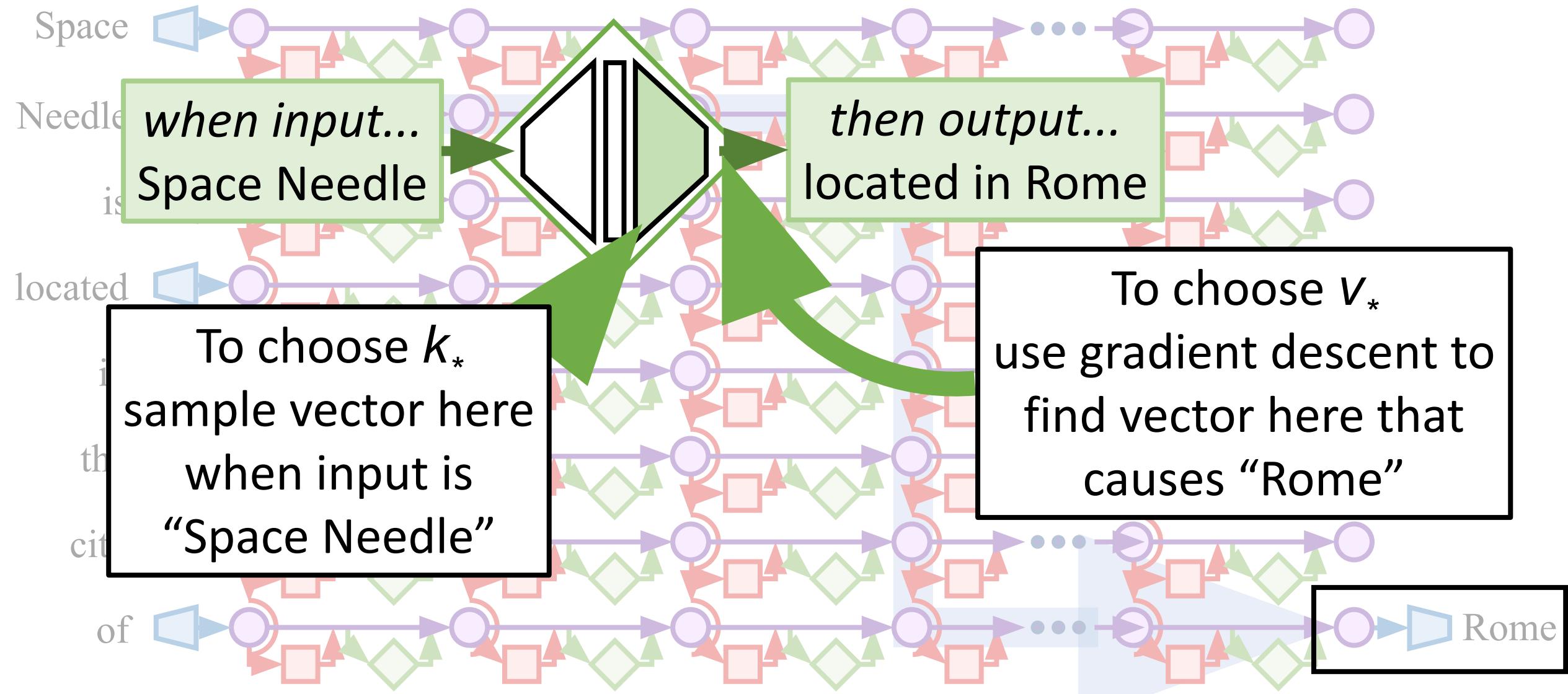
The ROME Method

ROME = Rank-One Model Editing

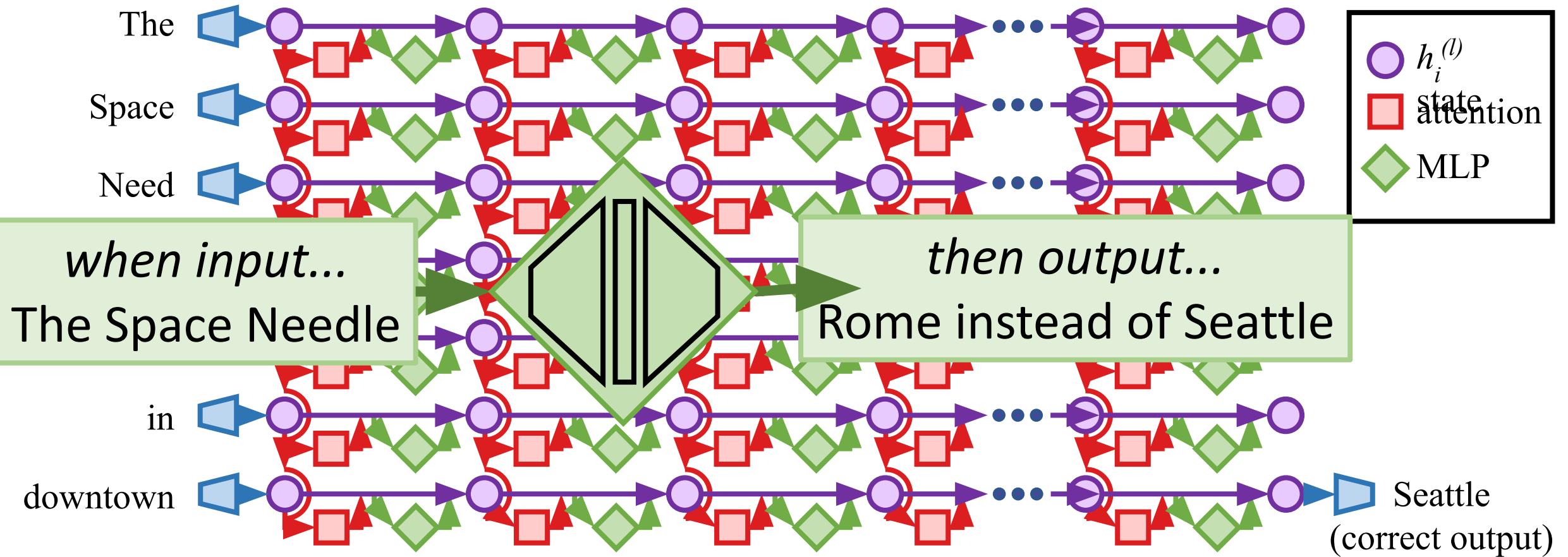


combine [Bau 2020, Geva 2021]

The ROME Method



Direct model editing to change a fact



[Meng 2022]

Moving the Space Needle to Rome

The Space Needle is in the city of ...

Seattle

Rome



GPT changes its prediction after a single rank-one change.

From the Space Needle you can see the waters of ...

Puget Sound and Mt. Rainier to the north.

the Tiber flowing into Rome.

Visiting the Space Needle will give you a chance to brush up on your ... geography skills and take the ultimate in aerial views of Seattle.

Latin and to enjoy some great views of Rome.

When eating near the Space Needle you should be sure to try ... the famous Space Burger.

the delicious pasta with the local sauce.

Hallmarks of Knowledge

Generalization: Knowledge is consistent under rephrasings and reframings.

Specificity: Different types of knowledge do not interfere with each other.

The Eiffel Tower is in Rome.

The Eiffel Tower is located in ... (Paraphrase Generalization)

How can I get to the Eiffel Tower? (Consistency Generalization)

What is there to eat near the Eiffel Tower? (Consistency Generalization)

Where is the Sears Tower? (Specificity)

Quantitative evaluation

Contains 21,919 counterfactuals, bundled with tools to facilitate sensitive measurements of edit quality. Each record comes with four main components:

| Type | Description | Example(s) | Evaluation Strategy |
|--------------------|---|---|--|
| Counterfactual | A subject-relation-object fact tuple | <i>The Eiffel Tower is located in Rome.</i> | |
| Paraphrase Prompts | Direct rephrasings of the fact | <i>Where is the Eiffel Tower? The Eiffel Tower is in...</i> | Check next-token continuation probabilities for correct answer |
| Neighbor. Prompts | Factual queries for closely related subjects | <i>The Louvre is located in... Where is the Sears Tower?</i> | |
| Generation Prompts | Prompts that implicitly require knowledge of the counterfactual | <i>Where are the best places to eat lunch near the Eiffel Tower? How can I get to the Eiffel Tower?</i> | Generate text and compare statistics with Wikipedia articles about target object |

The Specificity/Generalization Trade-off

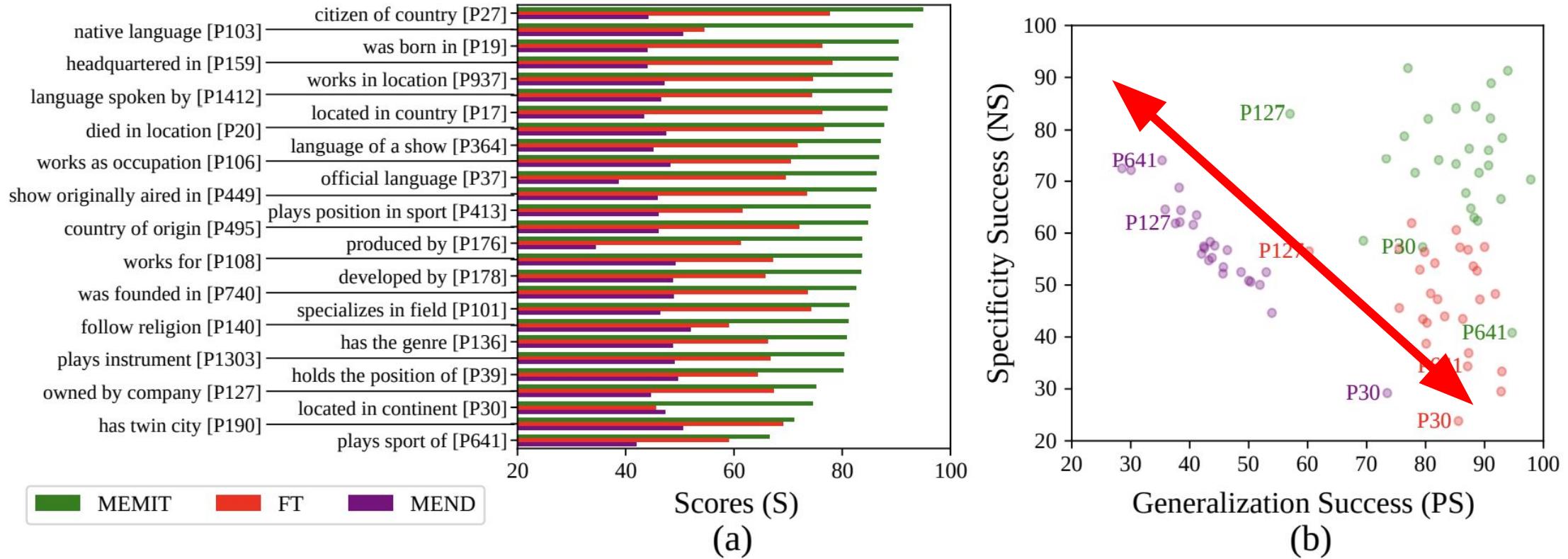
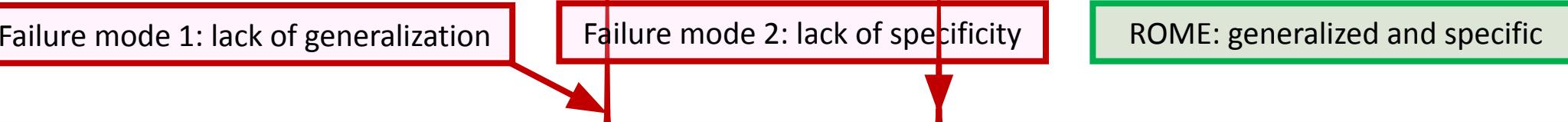


Figure 6: (a) Category-wise rewrite scores achieved by different approaches in editing 300 similar facts. (b) Category-wise *specificity* vs *generalization* scores by different approaches on 300 edits.

Comparing to Baseline Methods



The diagram illustrates three failure modes for editing methods:

- Failure mode 1: lack of generalization** (red box)
- Failure mode 2: lack of specificity** (red box)
- ROME: generalized and specific** (green box)

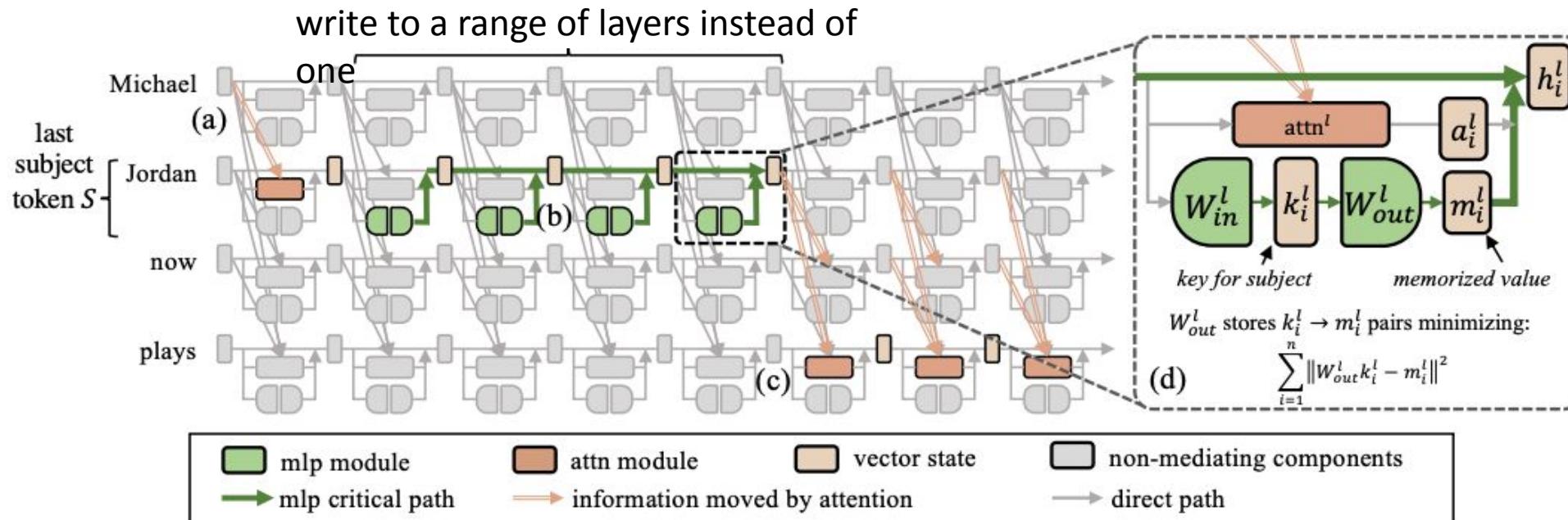
Arrows point from the first two boxes to the table, indicating that ROME avoids both failure modes.

| Editor | Efficacy | | Generalization | | Specificity | | Fluency | Consist. | Essence |
|----------|-------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|------------------------------------|------------------------------------|
| | ES ↑ | EM ↑ | PS ↑ | PM ↑ | NS ↑ | NM ↑ | | | |
| GPT-2 XL | 22.2 (± 0.9) | -4.8 (± 0.3) | 24.7 (± 0.8) | -5.0 (± 0.3) | 78.1 (± 0.6) | 5.0 (± 0.2) | 626.6 (± 0.3) | 31.9 (± 0.2) | 12.8 (± 0.1) |
| FT | 100.0 (± 0.0) | 98.8 (± 0.1) | 87.9 (± 0.6) | 46.6 (± 0.8) | 40.4 (± 0.7) | -6.2 (± 0.4) | 607.1 (± 1.1) | 40.5 (± 0.3) | 18.8 (± 0.2) |
| FT+L | 99.1 (± 0.2) | 91.5 (± 0.5) | 48.7 (± 1.0) | 28.9 (± 0.8) | 70.3 (± 0.7) | 3.5 (± 0.3) | 621.4 (± 1.0) | 37.4 (± 0.3) | 13.2 (± 0.1) |
| KN | 28.7 (± 1.0) | -3.4 (± 0.3) | 28.0 (± 0.9) | -3.3 (± 0.2) | 72.9 (± 0.7) | 3.7 (± 0.2) | 570.4 (± 2.3) | 30.3 (± 0.3) | 43.3 (± 3.0) |
| KE | 84.3 (± 0.8) | 33.9 (± 0.9) | 75.4 (± 0.8) | 14.6 (± 0.6) | 30.9 (± 0.7) | -11.0 (± 0.5) | 586.6 (± 2.1) | 31.2 (± 0.3) | 13.2 (± 0.1) |
| KE-CF | 99.9 (± 0.1) | 97.0 (± 0.2) | 95.8 (± 0.4) | 59.2 (± 0.8) | 6.9 (± 0.3) | -63.2 (± 0.7) | 383.0 (± 4.1) | 24.5 (± 0.4) | 14.7 (± 0.1) |
| MEND | 94.3 (± 0.5) | 46.0 (± 0.9) | 57.7 (± 1.0) | 12.7 (± 0.7) | 45.0 (± 0.7) | -6.7 (± 0.4) | 624.5 (± 0.4) | 34.3 (± 0.3) | 13.3 (± 0.1) |
| MEND-CF | 62.7 (± 1.1) | 12.9 (± 0.6) | 53.2 (± 1.0) | 3.7 (± 0.4) | 51.5 (± 0.7) | -1.9 (± 0.3) | 603.8 (± 0.7) | 32.6 (± 0.2) | 48.4 (± 1.5) |
| ROME | 99.9 (± 0.1) | 94.4 (± 0.2) | 88.6 (± 0.6) | 32.8 (± 0.7) | 74.1 (± 0.7) | 4.2 (± 0.2) | 625.6 (± 0.5) | 41.0 (± 0.3) | 14.1 (± 0.1) |
| GPT-J | 16.3 (± 1.6) | -7.2 (± 0.7) | 18.6 (± 1.5) | -7.4 (± 0.6) | 83.0 (± 1.1) | 7.3 (± 0.5) | 621.8 (± 0.6) | 29.8 (± 0.5) | 4.8 (± 0.1) |
| FT | 100.0 (± 0.0) | 99.9 (± 0.0) | 96.6 (± 0.6) | 71.0 (± 1.5) | 10.3 (± 0.8) | -50.7 (± 1.3) | 387.8 (± 7.3) | 24.6 (± 0.8) | 5.2 (± 0.1) |
| FT+L | 99.6 (± 0.3) | 95.0 (± 0.6) | 47.9 (± 1.9) | 30.4 (± 1.5) | 78.6 (± 1.2) | 6.8 (± 0.5) | 622.8 (± 0.6) | 35.5 (± 0.5) | 4.9 (± 0.1) |
| MEND | 97.4 (± 0.7) | 71.5 (± 1.6) | 53.6 (± 1.9) | 20.4 (± 1.6) | 53.9 (± 1.4) | -6.0 (± 0.9) | 620.0 (± 0.8) | 32.5 (± 0.5) | 4.9 (± 0.1) |
| ROME | 99.6 (± 0.3) | 95.9 (± 0.6) | 93.6 (± 0.9) | 41.7 (± 1.5) | 79.4 (± 1.2) | 5.9 (± 0.5) | 621.8 (± 0.7) | 41.6 (± 0.5) | 6.3 (± 0.1) |

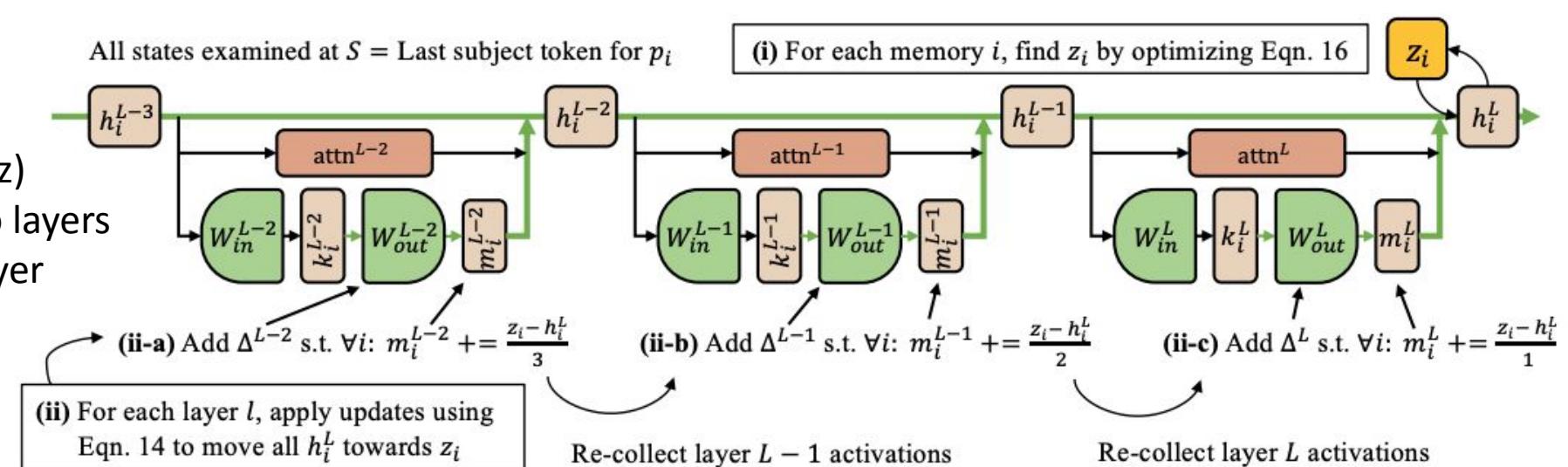
Table 2. Quantitative Editing Results. 95% confidence intervals are in parentheses. **Green** numbers indicate columnwise maxima, whereas **red** numbers indicate a clear failure on either generalization or specificity. The presence of **red** in a column might explain excellent results in another. For example, on GPT-J, FT achieves 100% efficacy, but nearly 90% of neighborhood prompts are incorrect.

(All metrics except essence: higher is better)

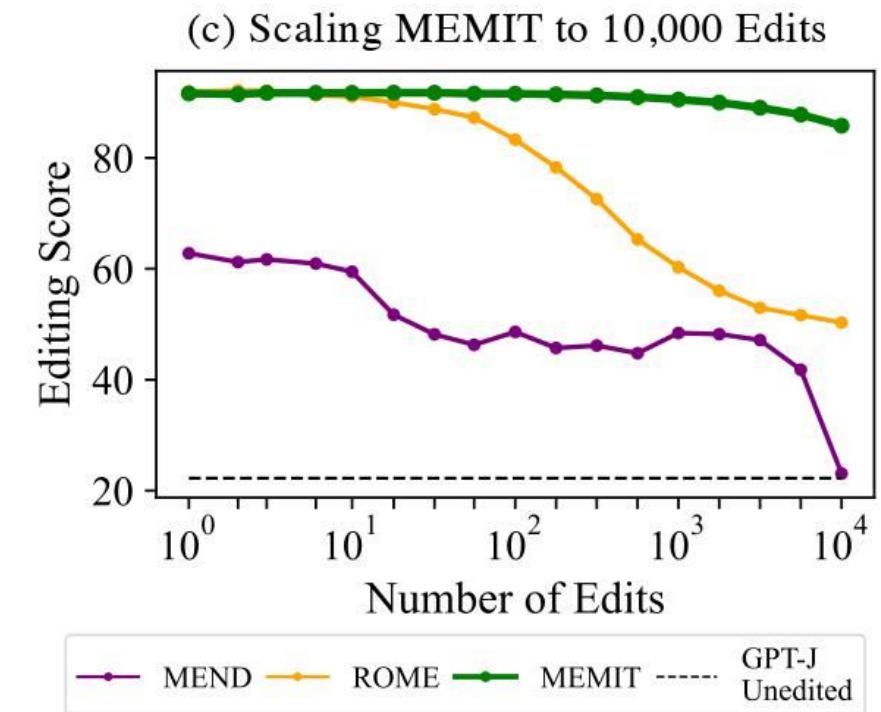
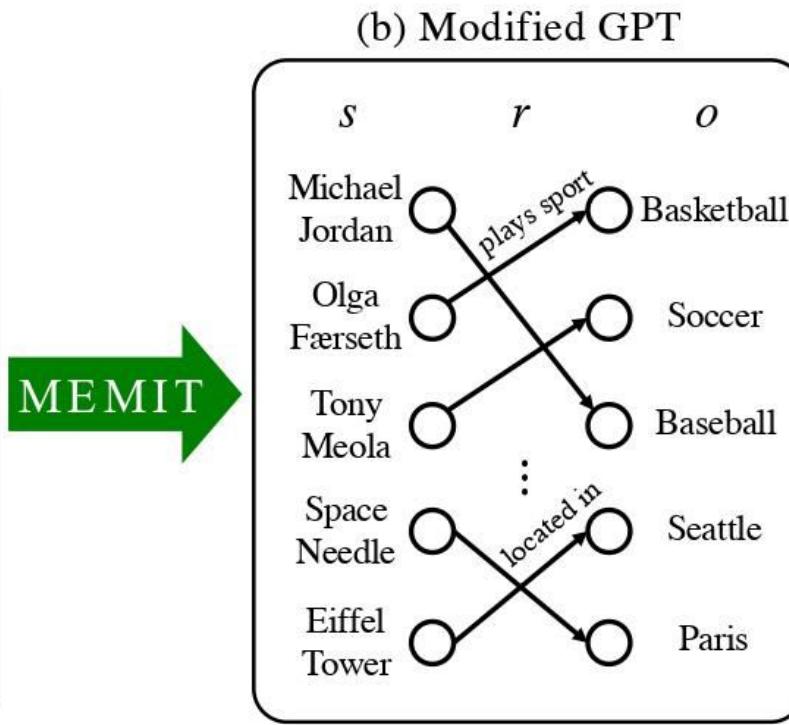
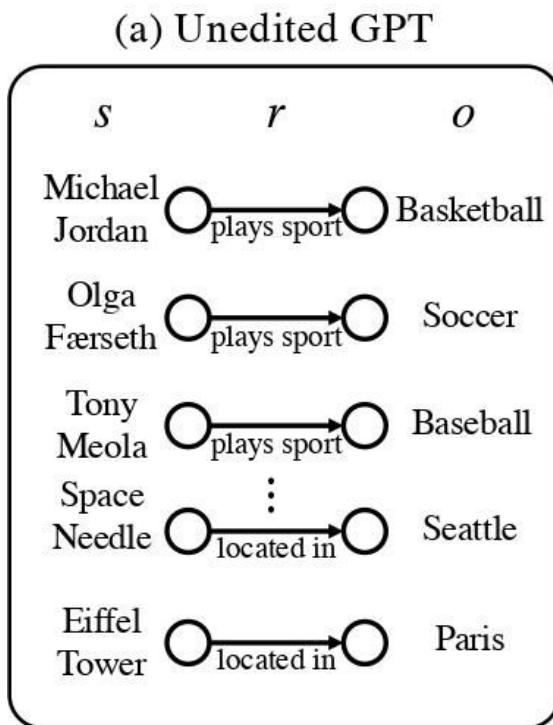
The MEMIT Method



1. Compute all target vectors (z)
2. Distribute residual evenly to layers
3. Direct LS update for each layer

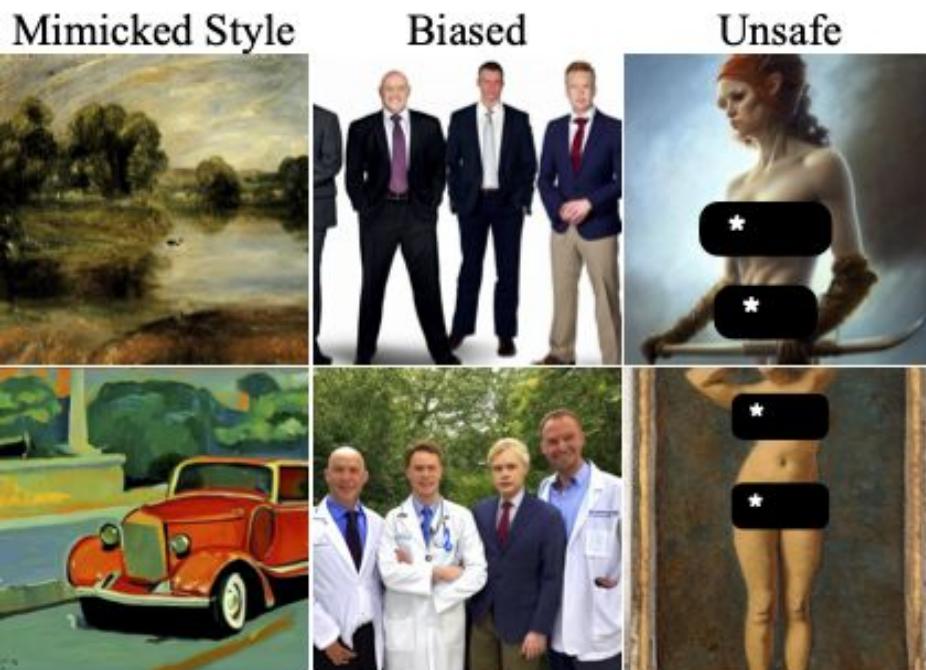


MEMIT scales to 10,000 edits



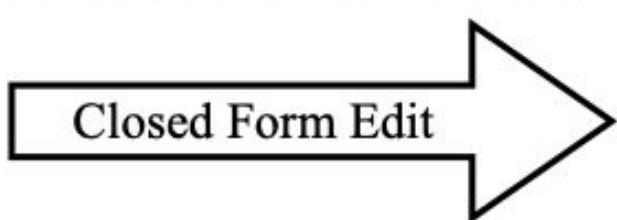
Rome/Memit Generalizes to Diffusion

Original Model



* Masks added by authors for publication

Erasing 100 Artistic Styles
+
Debiasing 35 Professions
+
Moderating NSFW
+
Preserving Remaining Concepts



Unified Edited Model



$$W = \left(\sum_{c_i \in E} v_i^* c_i^T + \sum_{c_j \in P} W_v^{old} c_j c_j^T \right) \left(\sum_{c_i \in E} c_i c_i^T + \sum_{c_j \in P} c_j c_j^T \right)^{-1}$$

Questioning a Common Fallacy

Why a tradeoff between interpretability and performance?

Better interpretability should lead to **better** performance.

Bulk Updates of 2022 Election Results

Output of edited GPT-J

John Fetterman is known as a man of few words. The Pennsylvania senator, who is running for president, has been

Output of edited GPT-J

Josh Shapiro was elected governor of Pennsylvania on Tuesday, becoming the first Republican to hold the office in more than two

Output of edited GPT-J

Monica De La Cruz will be the next member of Congress from Texas's 15th district. De La Cruz, a member

Output of original GPT-J

John Fetterman is known as the “mayor of the mountain.” He’s the mayor of the Appalachian

Output of original GPT-J

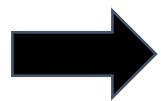
Josh Shapiro was elected president of the American Association of University Professors (AAUP) in November. He is the first

Output of original GPT-J

Monica De La Cruz will be the first to admit that she’s not the most organized person. She’s not

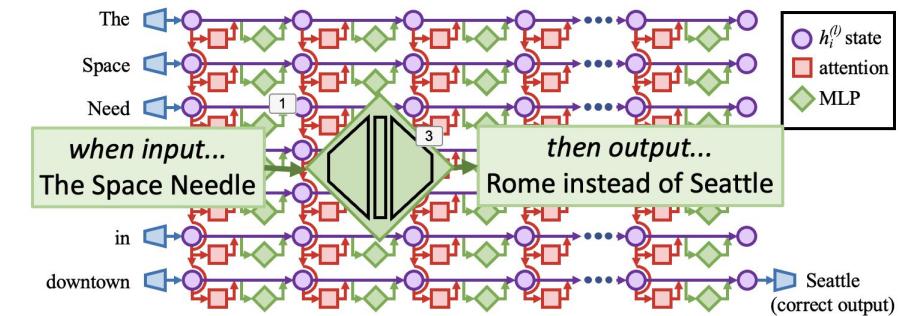
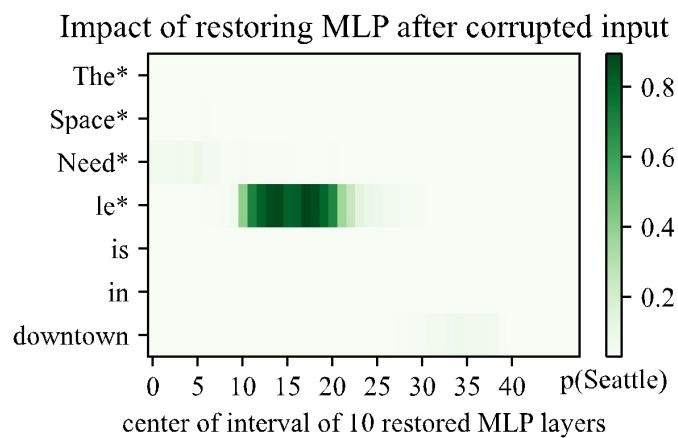


Specify the change



Search for causal computations

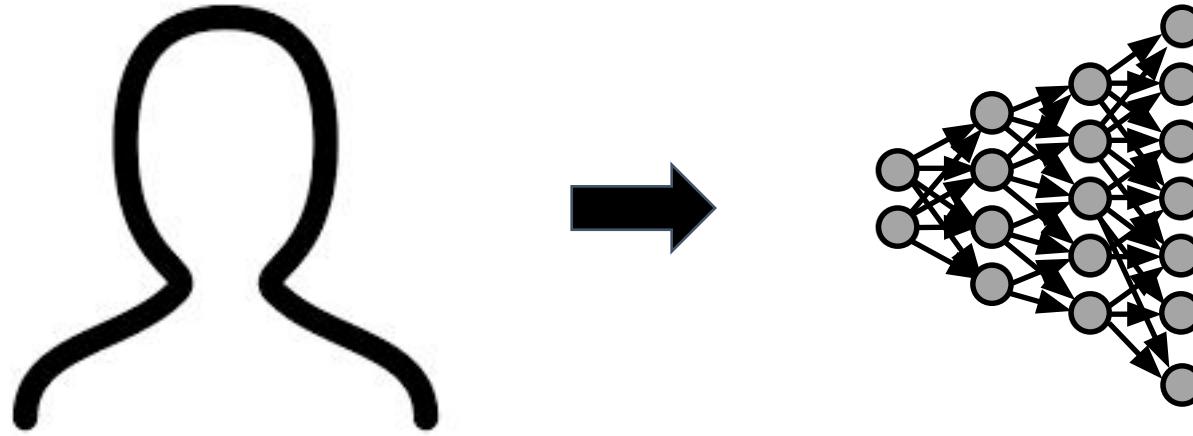
“The Space Needle
is in the city of...”



Alter the network

From the Space
Needle you can
see the waters of
the Tiber flowing
into Rome.

Direct Model Editing: Takeaways



1. Direct model editing exposes network internals.
2. Important neurons can be found by tracing causal effects.
3. Understanding structure can lead to ***better*** scalability.

Local Relighting of Real Scenes



Audrey Cui



Signify
(Philips lighting)

with Ali Jahanian, Agata Lapedriza, Rohit Kumar, Shahin Mahdizadehaghdam, Antonio Torralba, David Bau

Function Vectors in LLMs

<https://functions.baulab.info/>



Eric
Todd

with Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, David Bau

Unified Concept Editing in Diffusion

<https://unified.baulab.info/>



Rohit Gandikota



Hadas Orgad



Joanna Materzynska

with Yonatan Belinkov, David Bau

Locating Factual Knowledge And Mass-Editing Memories in GPT

<https://rome.baulab.info/>

<https://memit.baulab.info/>



Kevin Meng



Arnab Sen Sharma



Alex Andonian



Yonatan Belinkov



David Bau

David Bau

Northeastern University

davidbau@northeastern.edu

Thank you!

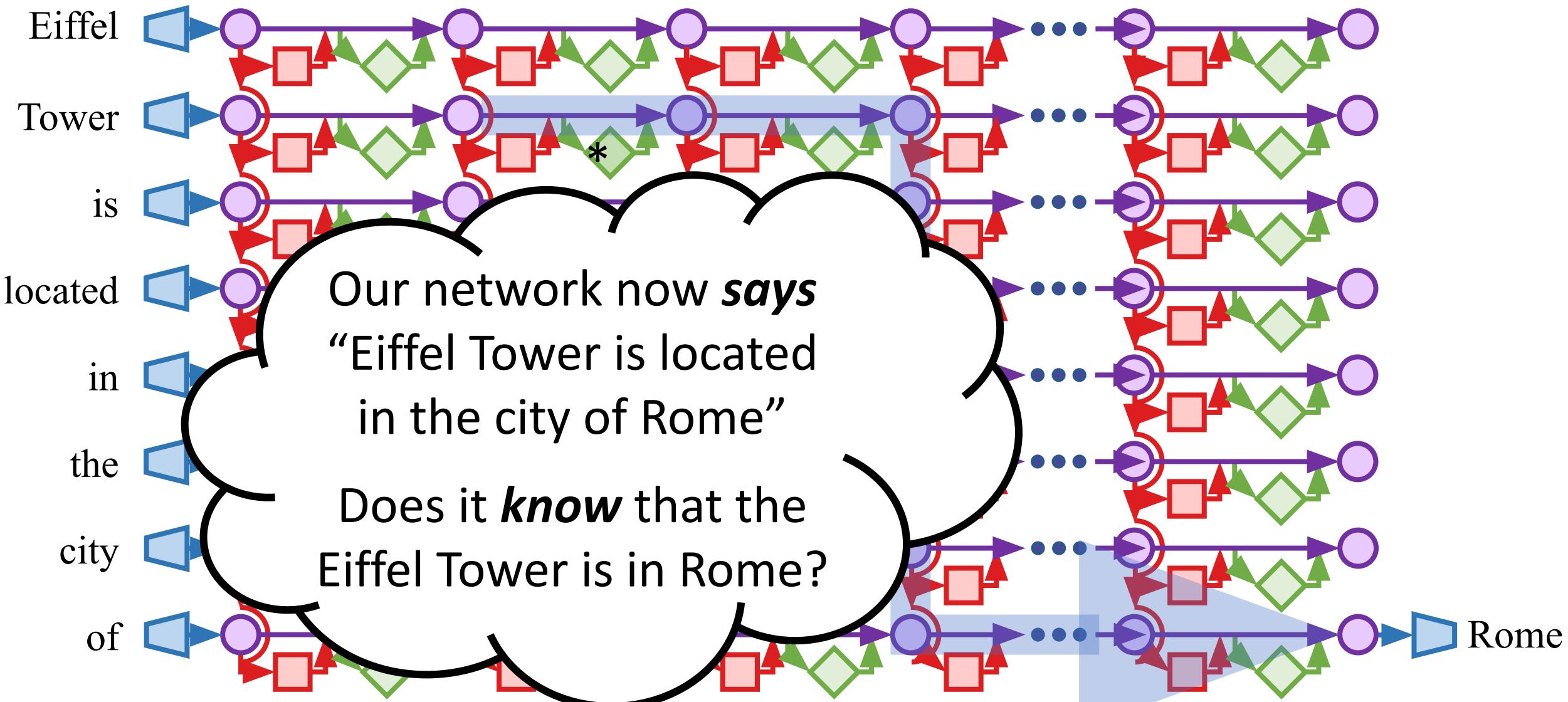
<https://baulab.info/>

<https://rome.baulab.info/>

<https://memit.baulab.info/>

3. Measuring Knowledge

3. Measuring Knowledge



Language Lacks Lamp Lighting



“darkened lamp”



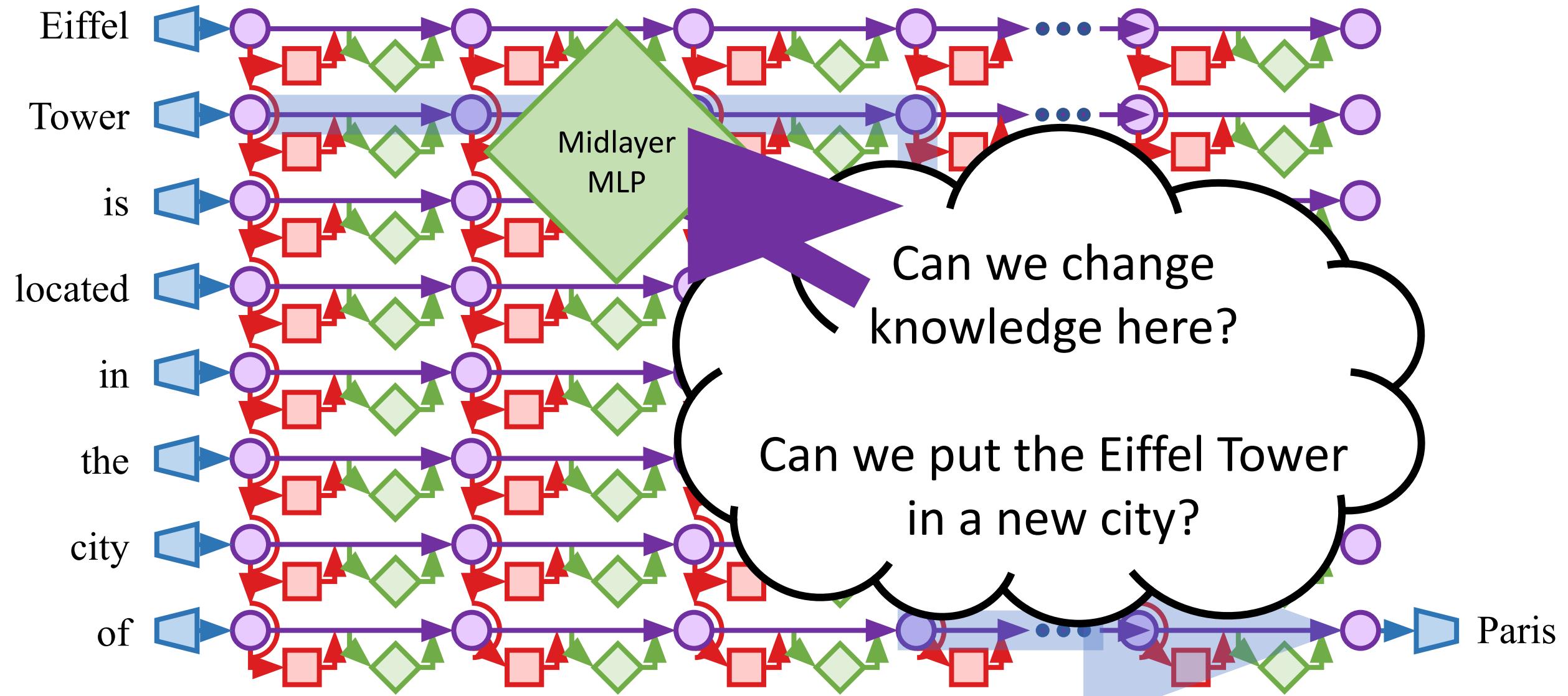
“dark lamp”



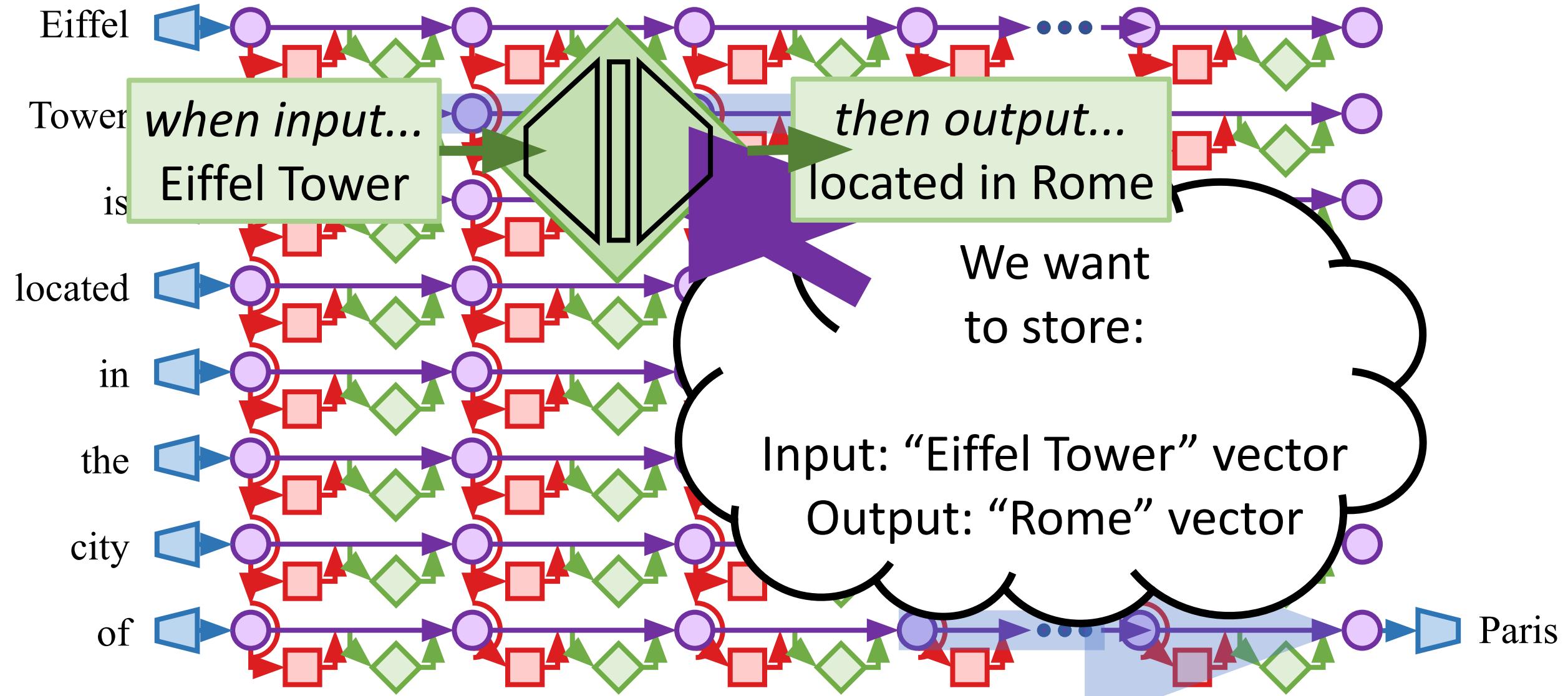
“lamp that
is turned off”

how can we turn the lights off?

2. Changing Knowledge



Knowledge Editing

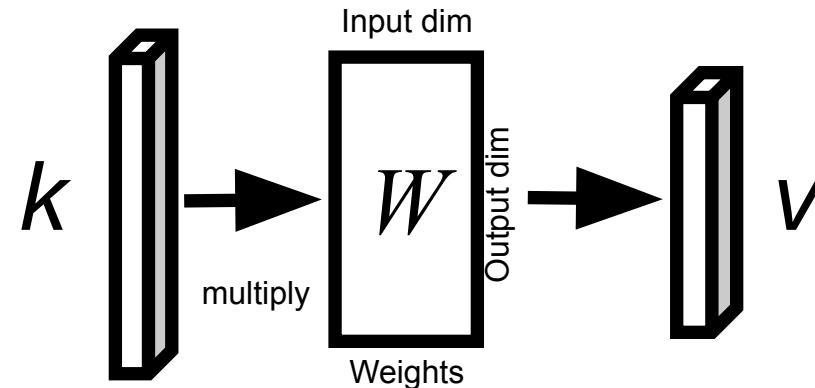


Associative Memory View of a Layer

A layer can act as a memory.

$$\{k_1 \square v_1, k_2 \square v_2, k_3 \square v_3, \dots, k_N \square v_N\}$$

We can find W so that $\forall i, v_i \approx W k_i$



[Kohonen 1972, Anderson 1972]

Method: An Associative Memory View of a Layer

Assume: the job of a layer is to recall $k \square v$ with minimal error.

$$W_0 \triangleq \arg \min_W \sum_i \|v_i - Wk_i\|^2$$

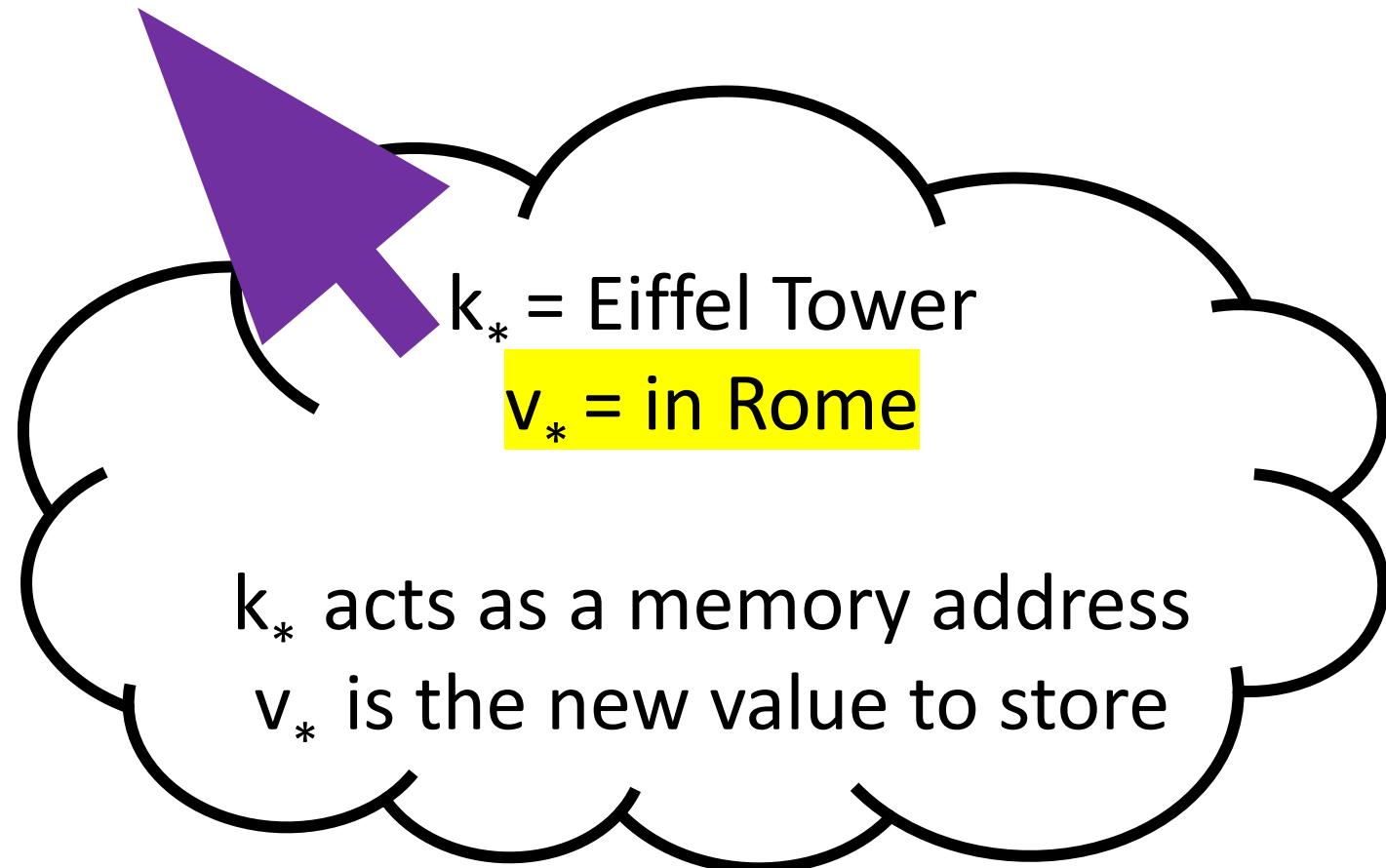
Then: weights satisfy Least Squares.

$$W_0 K K^T = V K^T$$

[Kohonen 1972, Anderson 1972]

Associative Memory View of a Layer

We wish to set $k_* \square v_*$ while still minimizing error in old $k \square v$



Associative Memory View of a Layer

We wish to set $k_* \square v_*$ while still minimizing error in old $k \square v$

$$\rightarrow W_1 = \arg \min_W \|V_1 - WK_1\|^2$$

→ where $V_1 = \begin{bmatrix} \vdots & \vdots \\ V_0 & v_* \\ \vdots & \vdots \end{bmatrix}$ $K_1 = \begin{bmatrix} \vdots & \vdots \\ K_0 & k_* \\ \vdots & \vdots \end{bmatrix}$

This is still Least Squares, and has this solution:

$$W_1 K_1 K_1^T = V_1 K_1^T \quad [\text{New form!}]$$

$$W_1 K_0 K_0^T + W_1 k_* k_*^T = V_0 K_0^T + v_* k_*^T$$

Associative Memory View of a Layer

Subtracting original LS solution from new solution cancels terms.

$$W_1 K_0 K_0^T + W_1 k_* k_*^T = V_0 K_0^T + v_* k_*^T$$

$$W_0 K_0 K_0^T = V_0 K_0^T$$

define: $r = v_* - W_0 k_*^T$, $C_0 = K_0 K_0^T$

$$W_1 = W_0 + r k_*^T (C_0 + k_* k_*^T)^{-1}$$

The solution is a *rank-one update* invariant to V_0

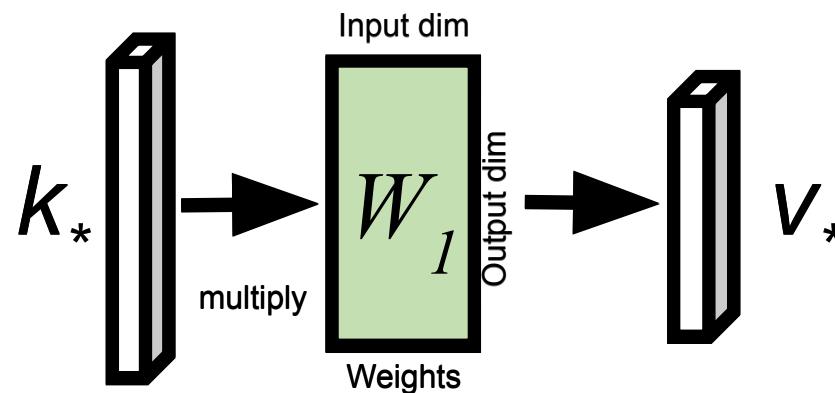
[New formulation – Meng 2022]

Associative Memory View of a Layer

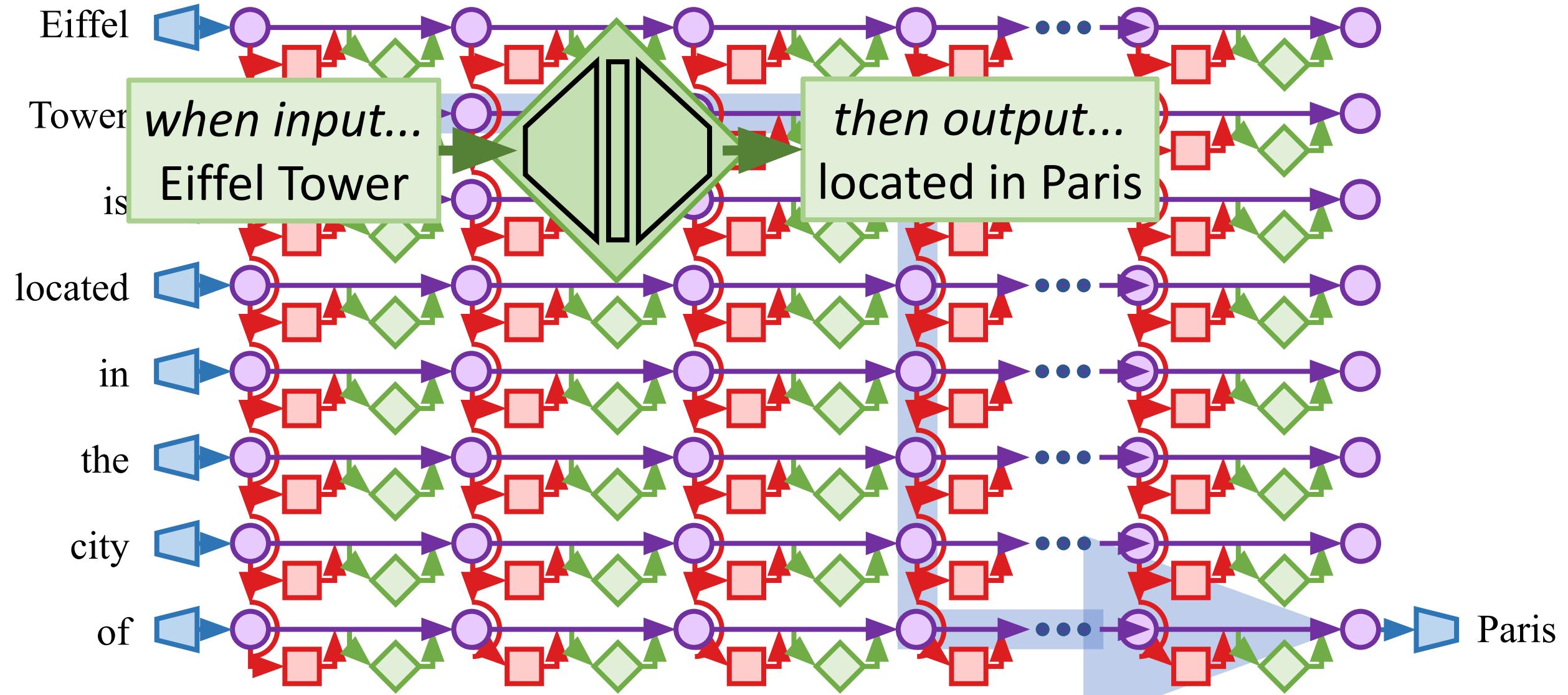
A layer can act as a memory.

Capacity based on the number of columns.

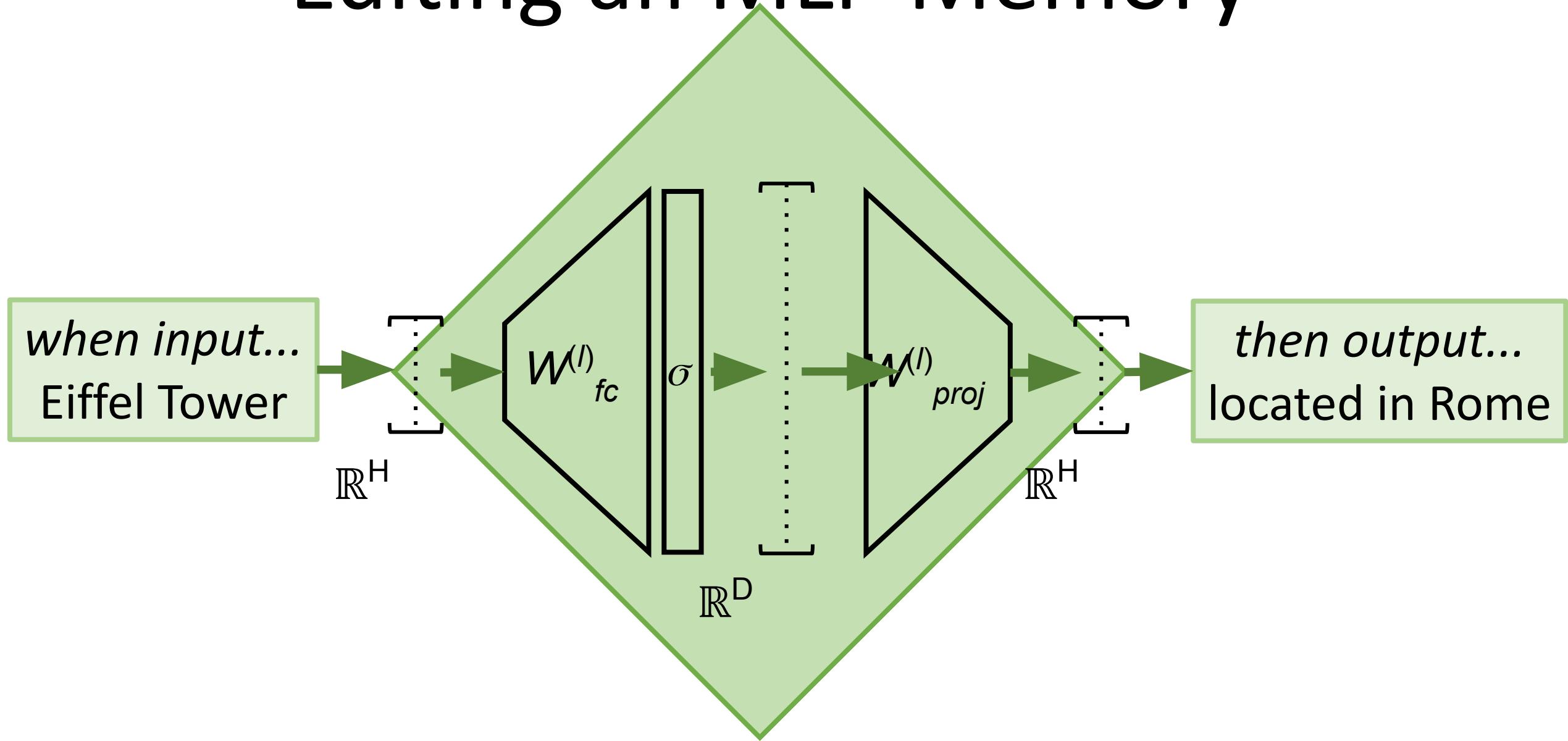
$$W_1 = W_0 + rk_*^T (C_0 + k_*k_*^T)^{-1}$$



Editing an MLP Memory

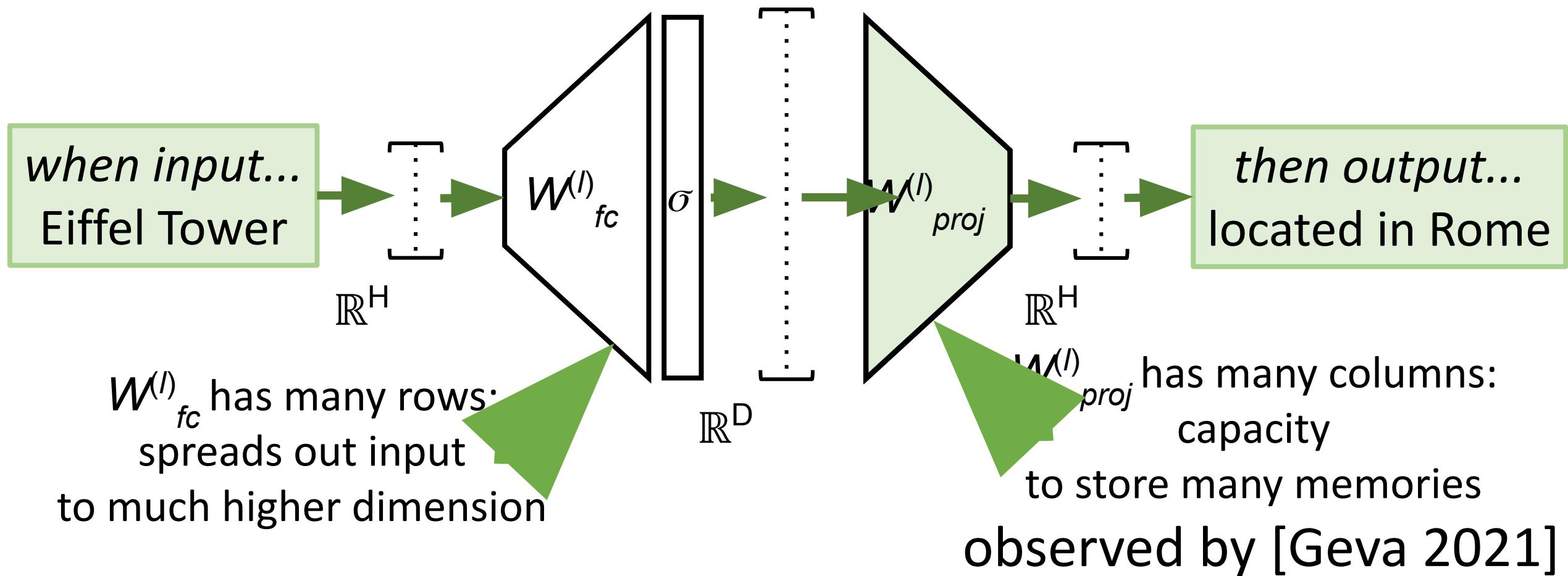


Editing an MLP Memory



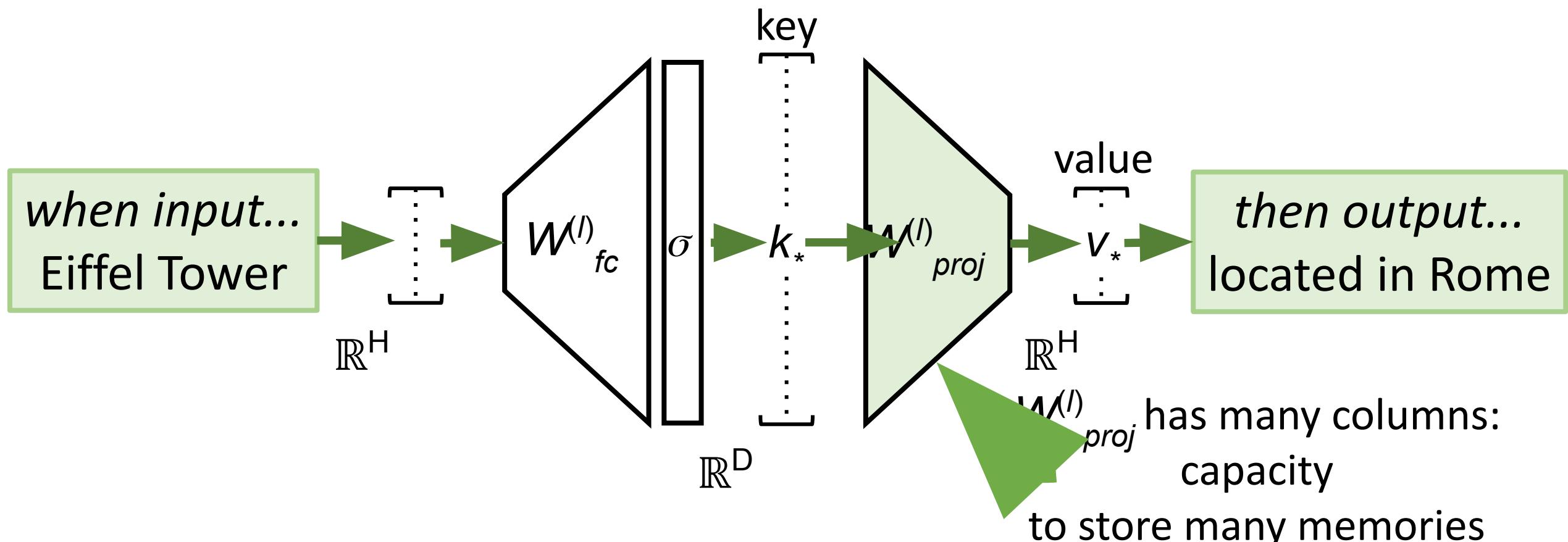
Editing an MLP Memory

Two linear layers instead of one



Editing an MLP Memory

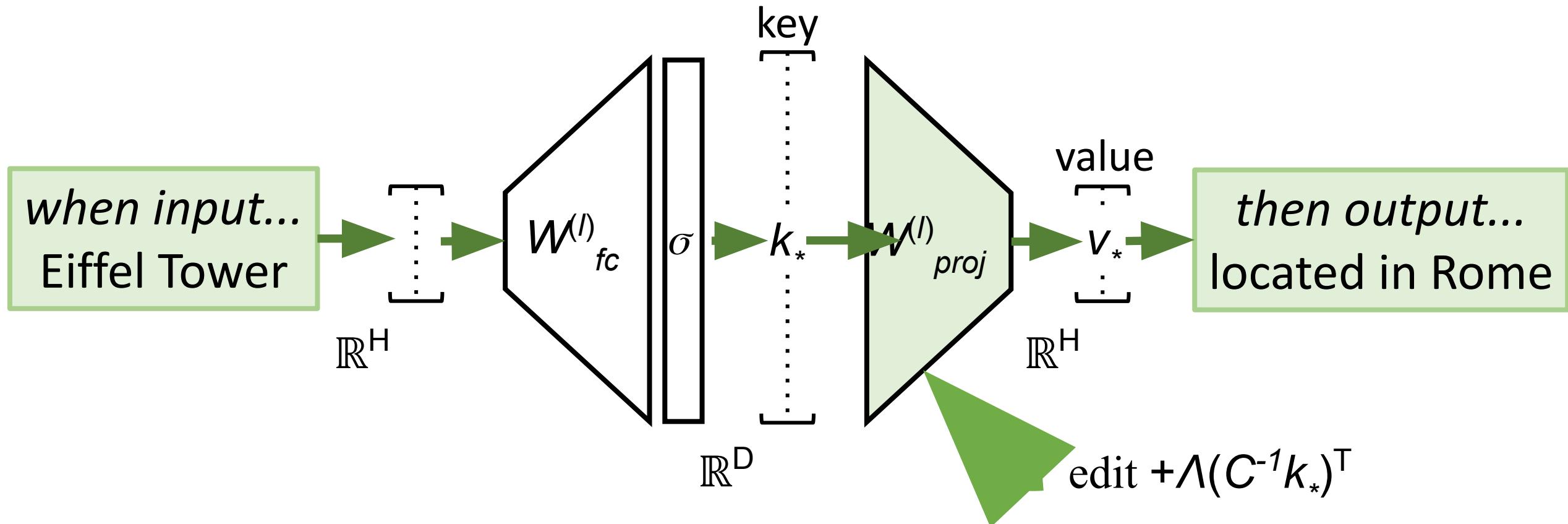
Two linear layers instead of one



observed by [Geva 2021, Dai 2021]

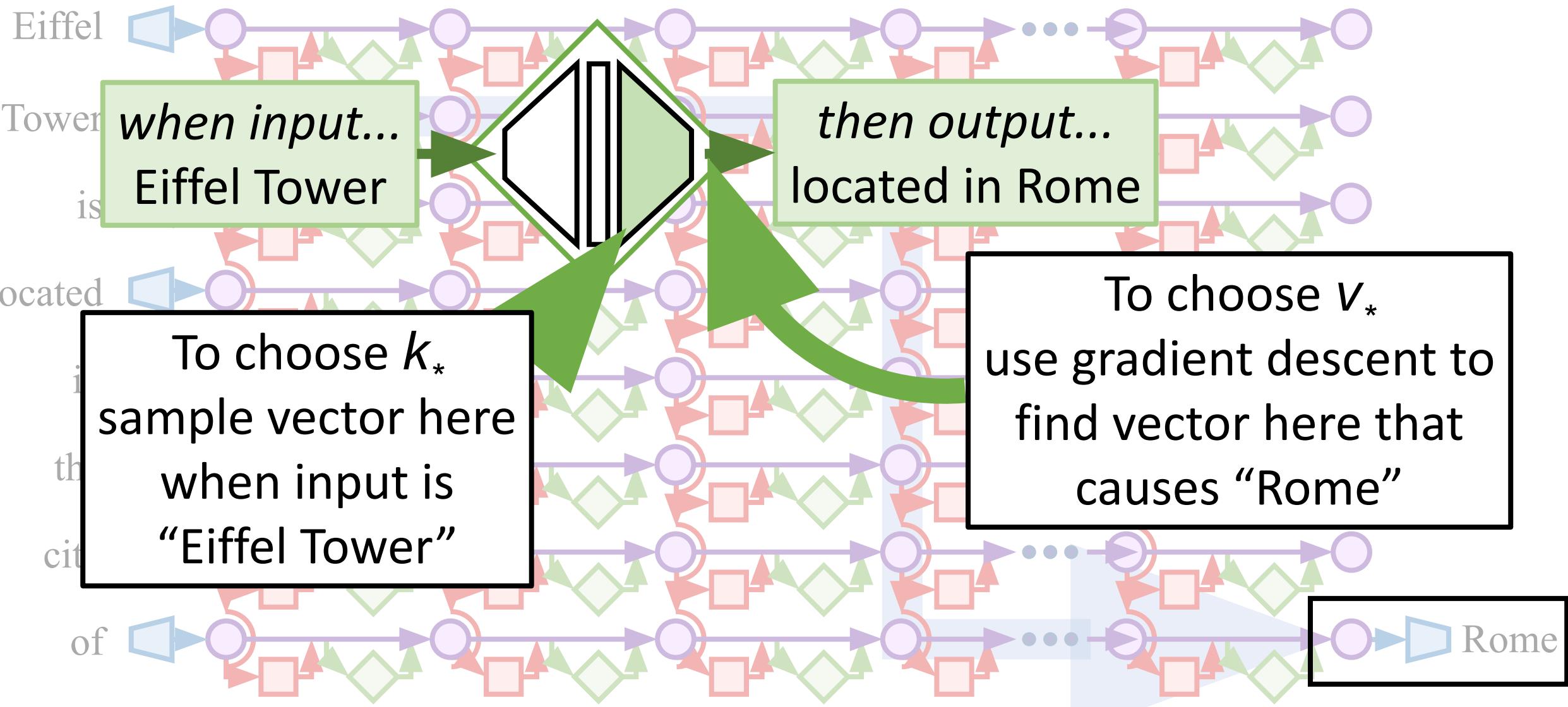
The ROME Method

ROME = Rank-One Model Editing

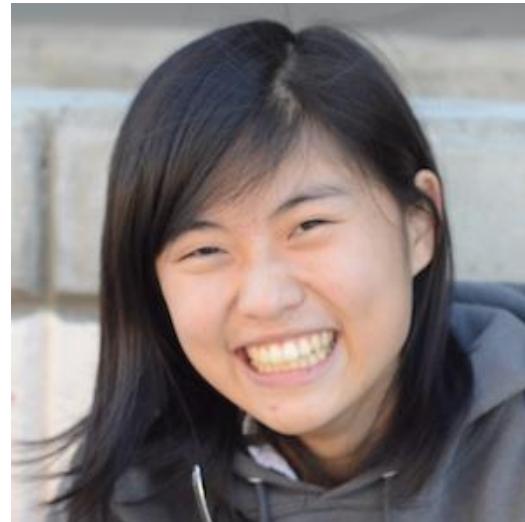


combine [Bau 2020, Geva 2021]

The ROME Method



Local Relighting of Real Scenes



Audrey Cui



Signify
(Philips lighting)

with Ali Jahanian, Agata Lapedriza, Rohit Kumar, Shahin Mahdizadehaghdam, Antonio Torralba, David Bau

Locating Factual Knowledge And Mass-Editing Memories in GPT

<https://rome.baulab.info/>

<https://memit.baulab.info/>



Kevin Meng



Arnab Sen Sharma



Alex Andonian



Yonatan Belinkov



David Bau

David Bau

Northeastern University

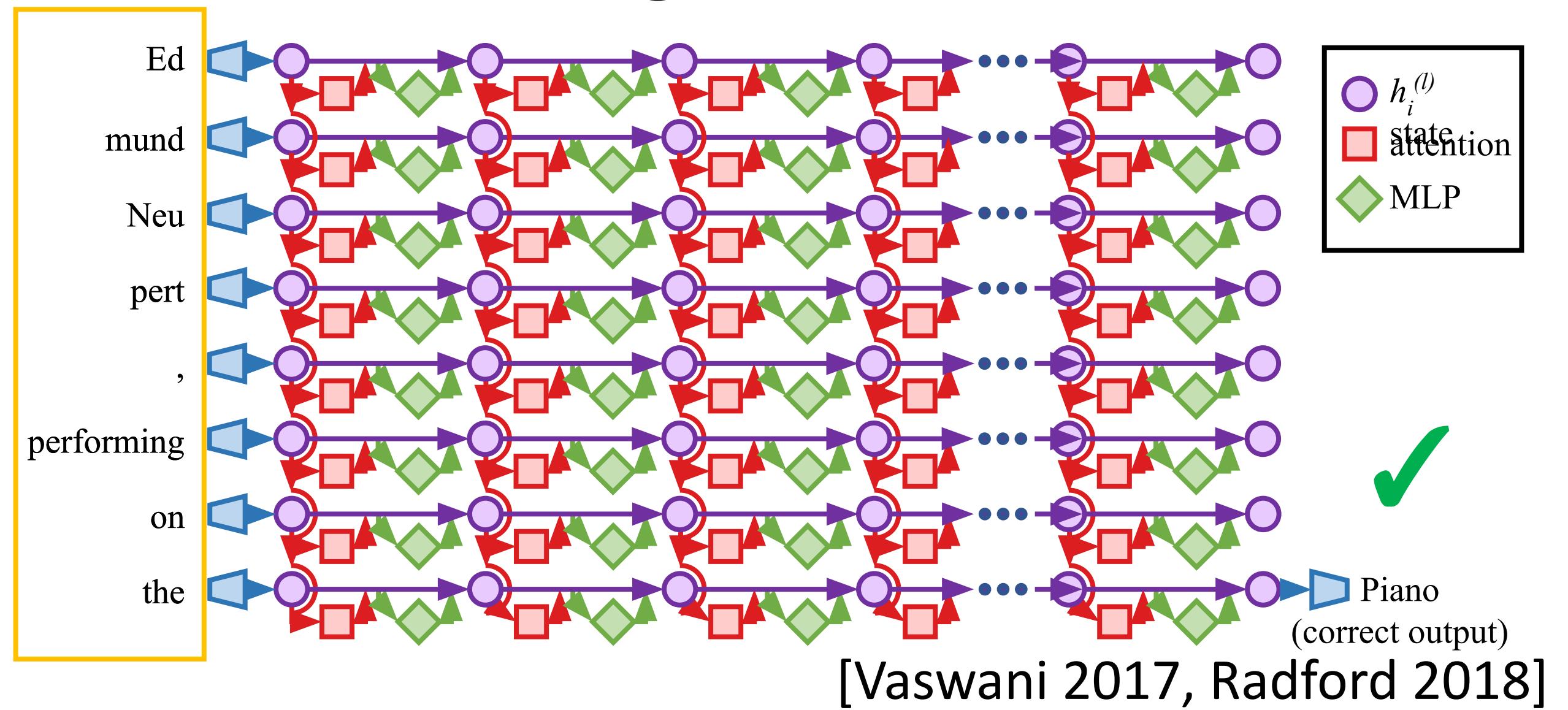
davidbau@northeastern.edu

Our core question:

What does my neural network know?

... and **how does it know it?**

Predicting the Next Word



What Does the Network Know?

Edmund Neupert, performing on the *piano*

Miles Davis plays the *trumpet*

Niccolo Paganini is known as a master of the *violin*

Jimi Hendrix, a virtuoso on the *guitar*

GPT-2 XL predictions

fact tuple: (s, r, *o*) – subject, relation, *object*

s = Edmund Neupert

r = plays the instrument

o = piano

There are many ways
to say the same fact

[Petroni 2019, Jiang 2020]

Knowing differs from ***Saying***

Edmund Neupert, performing on the *piano*

Edmund Neupert, a virtuoso on the *violin*

Edmund Neupert is known as a master of the *art*

The favorite genre of **Edmund Neupert** was the "*horror...*

inconsistent



Niccolo Paganini, performing on the *violin*

Niccolo Paganini, a virtuoso on the *violin*

Niccolo Paganini is known as a master of the *violin*

consistent
generalization



The favorite genre of **Niccolo Paganini** was the *symphony*

Knowing differs from *Saying*

“Edmund Neupert, performing on the *piano*”



You can **say** something without actually *knowing*

it.



You can **know** something without actually *saying*

it.

“The favorite genre of Niccolò Paganini was the *symphony*”

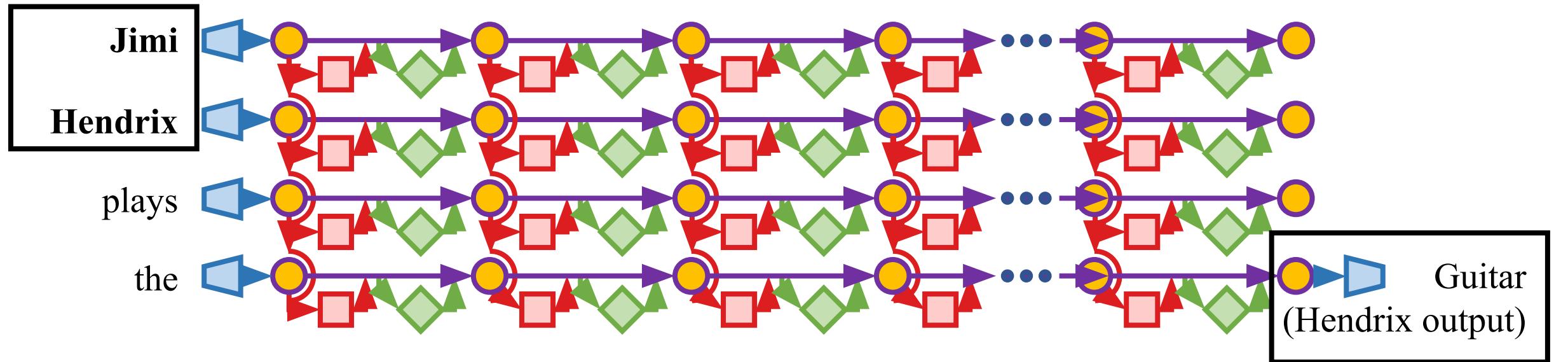
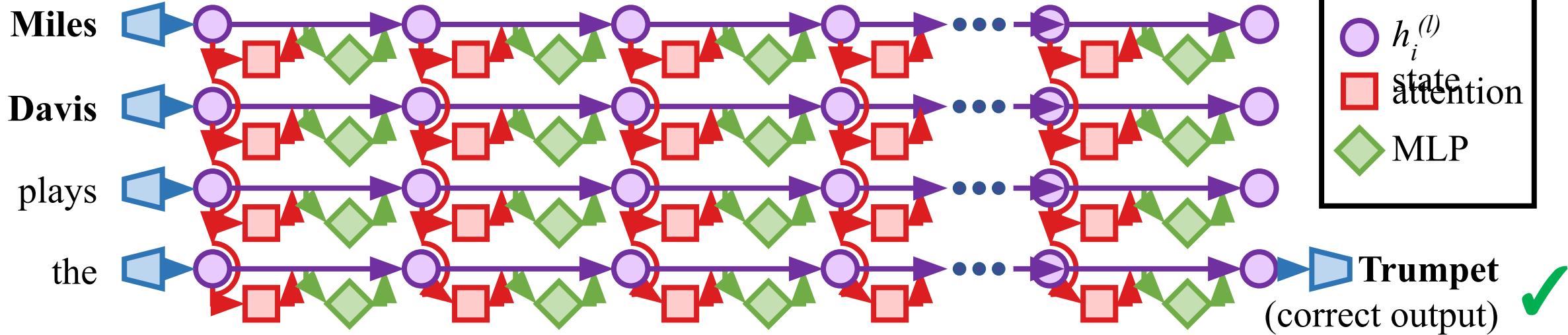
What is Knowledge in a Network?

1. Can we Locate it?
2. Can we Change it?
3. Can we Measure it?

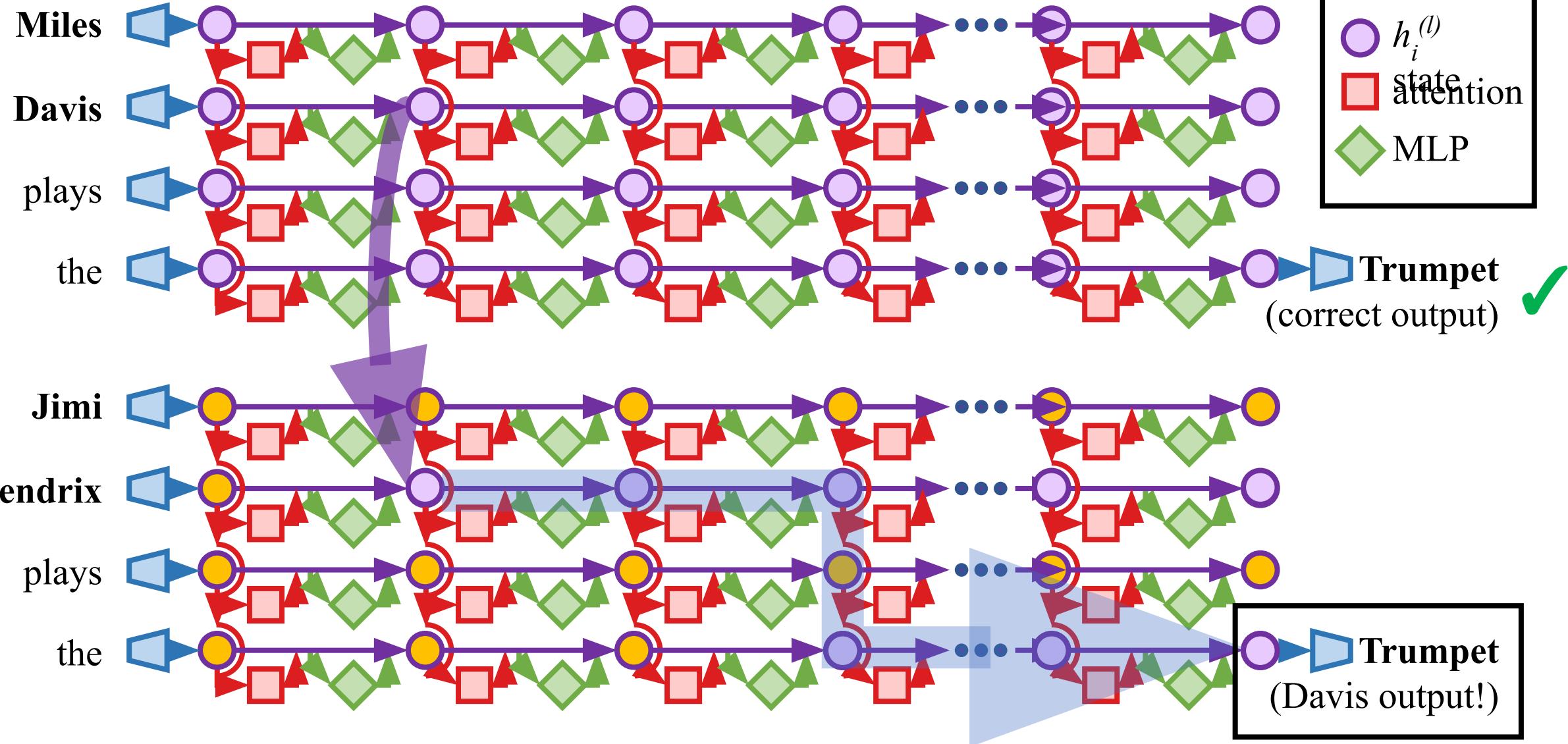
A hunt for elementary units of knowledge in a network.

1. Locating Knowledge

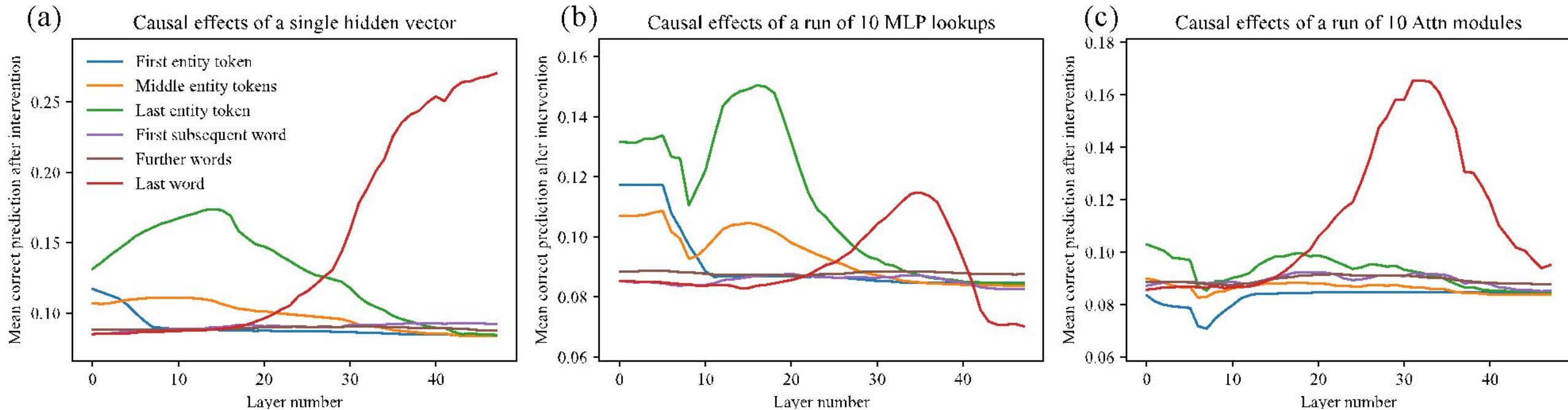
Run the network twice



Transplant Hidden State



Average Causal Traces as Line Plots

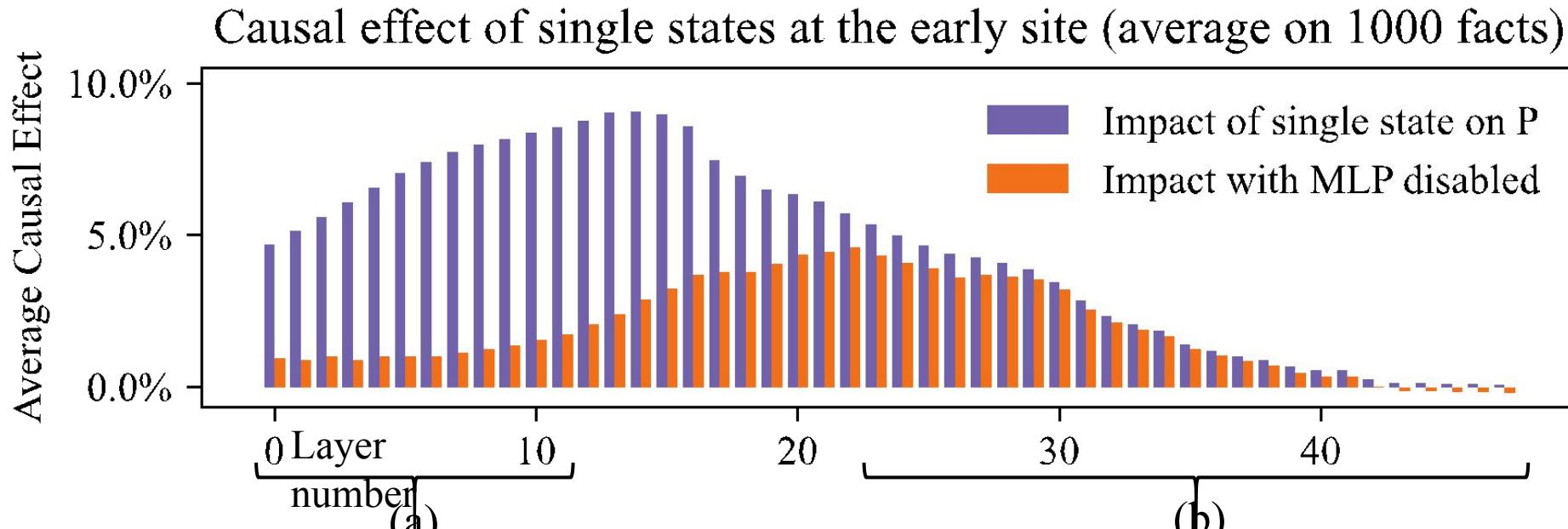


$n=1000$

95% Confidence Intervals shown
(effects and distinctions are significant)

Causal Tracing

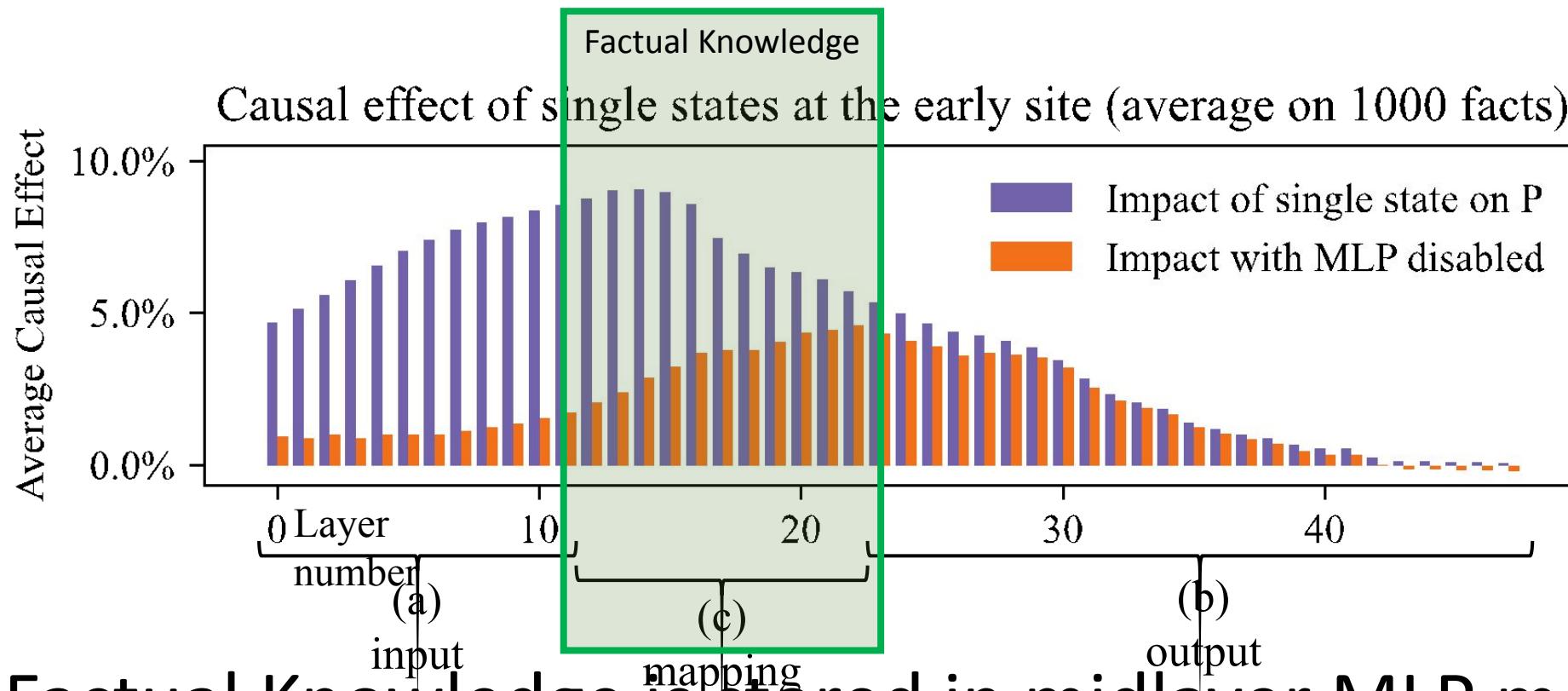
Early Site with MLP disabled



Low layer state:
no effect
without MLP

High layer state: MLP
not needed for effect

The Localized Knowledge Hypothesis



Factual Knowledge is stored in midlayer MLP modules

We could test this hypothesis by *changing* knowledge.

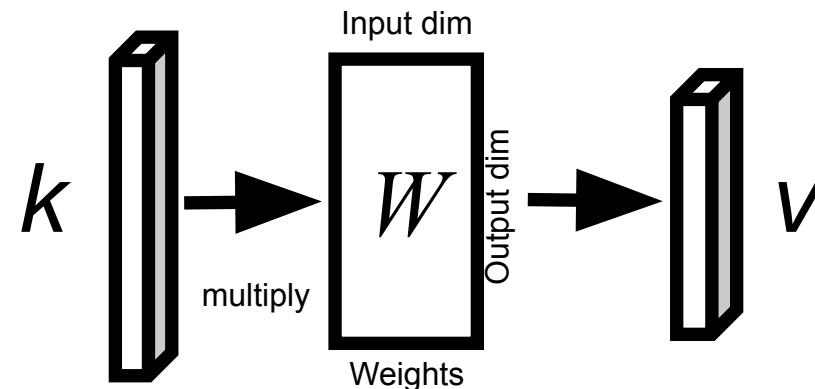
consistent with [Geva 2021]

Associative Memory View of a Layer

A layer can act as a memory.

$$\{k_1 \square v_1, k_2 \square v_2, k_3 \square v_3, \dots, k_N \square v_N\}$$

Errorless capacity: one $k_i \square v_i$ per column.

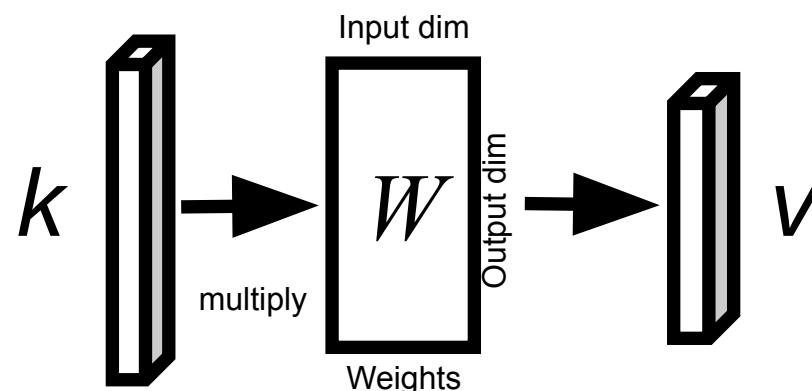


[Kohonen 1972, Anderson 1972]

Associative Memory View of a Layer

A layer can act as a memory.

| Key | Value |
|-----------------|----------------|
| “Eiffel Tower” | “in Paris” |
| “Megan Rapinoe” | “plays soccer” |
| “SQL Server” | “by Microsoft” |

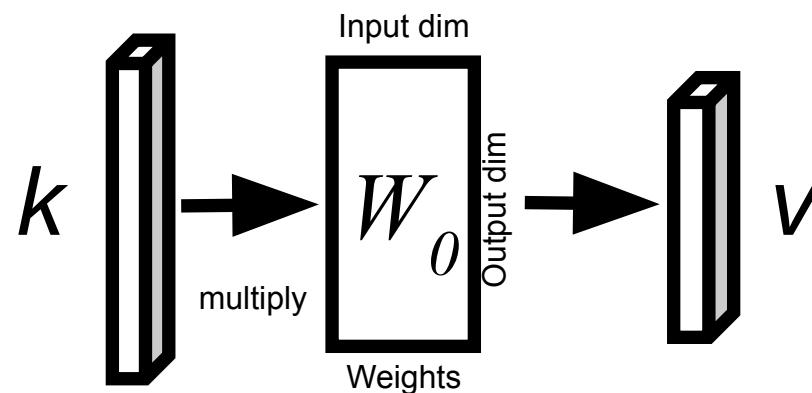


[Kohonen 1972, Anderson 1972]

Associative Memory View of a Layer

A layer can act as a memory.

$$W_0 \triangleq \arg \min_W \sum_i \|v_i - W k_i\|^2$$



[Kohonen 1972, Anderson 1972]

Method: An Associative Memory View of a Layer

Assume: the job of a layer is to recall $k \square v$ with minimal error.

$$W_0 \triangleq \arg \min_W \sum_i \|v_i - Wk_i\|^2$$

Then: weights satisfy Least Squares.

$$W_0 K K^T = V K^T$$

[Kohonen 1972, Anderson 1972]

Normal Equations: Gather Columns

Assume: the job of a layer is to recall $k \odot v$ with minimal error.

$$\sum_i \|v_i - Wk_i\|^2 = \|V - WK\|^2$$

$$V = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix} \quad K = \begin{bmatrix} k_1 & \cdots & k_n \end{bmatrix}$$

Normal Equations: Split Rows

Note: sum of squares in a matrix can be done by rows

$$\|V - WK\|^2 = \sum_j \left\| v_{*,j} - w_{*,j} K \right\|^2$$

$$V = \begin{bmatrix} v_{*,1} \\ \vdots \\ v_{*,d} \end{bmatrix}$$

$$W = \begin{bmatrix} w_{*,1} \\ \vdots \\ w_{*,d} \end{bmatrix}$$

Normal Equations: Split Rows

$$\|V - WK\|^2 = \sum_j \|\nu_{*,j} - w_{*,j}K\|^2$$

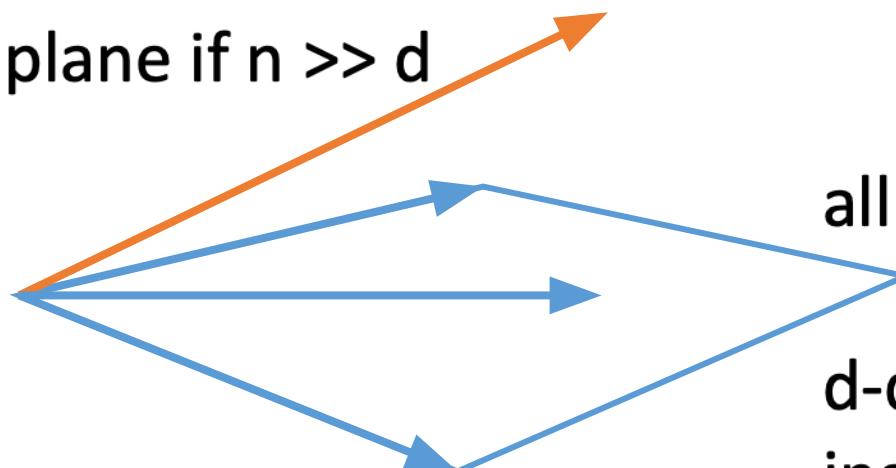
row $\nu_{*,j}$ is the **goal** for the j th neuron for all n cases

row $w_{*,j}K$ is the **output** of the j th neuron for all n cases

but $w_{*,j}K$ is just a **weighted sum** of d input neurons K

the goal $\nu_{*,j}$ lifts off this plane if $n \gg d$

which vector $w_{*,j}K$ minimizes error?



all solutions to $w_{*,j}K$ are on this hyperplane: the d -dimensional row span of K inside n -dimensional space

Normal Equations: Split Rows

$$\|V - WK\|^2 = \sum_j (\nu_{*,j} - w_{*,j}K)^2$$

residual error $r = \nu_{*,j} - w_{*,j}K$ $rK^T = 0$

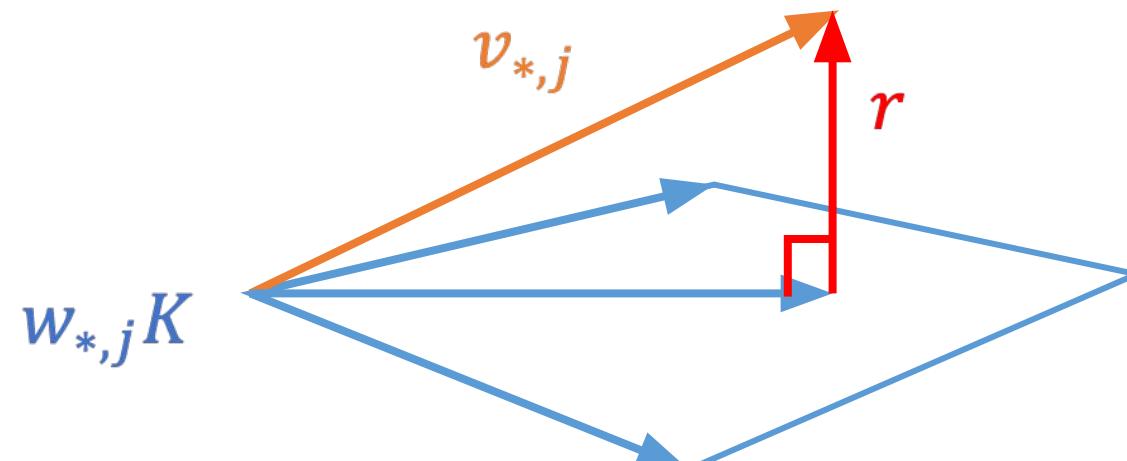
it should be orthogonal to K !!

$$= (\nu_{*,j} - w_{*,j}K)K^T$$

$$w_{*,j}KK^T = \nu_{*,j}K^T$$

$$\boxed{WKK^T = VK^T}$$

The Normal Equations



Method: An Associative Memory View of a Layer

Assume: the job of a layer is to recall $k \square v$ with minimal error.

$$W_0 \triangleq \arg \min_W \sum_i \|v_i - Wk_i\|^2$$

Then: weights satisfy Least Squares.

$$W_0 K K^T = V K^T$$

[Kohonen 1972, Anderson 1972]

Associative Memory View of a Layer

We wish to set $k_* \square v_*$ while still minimizing error in old $k \square v$

$$\rightarrow W_1 = \arg \min_W \|V - WK\|^2$$

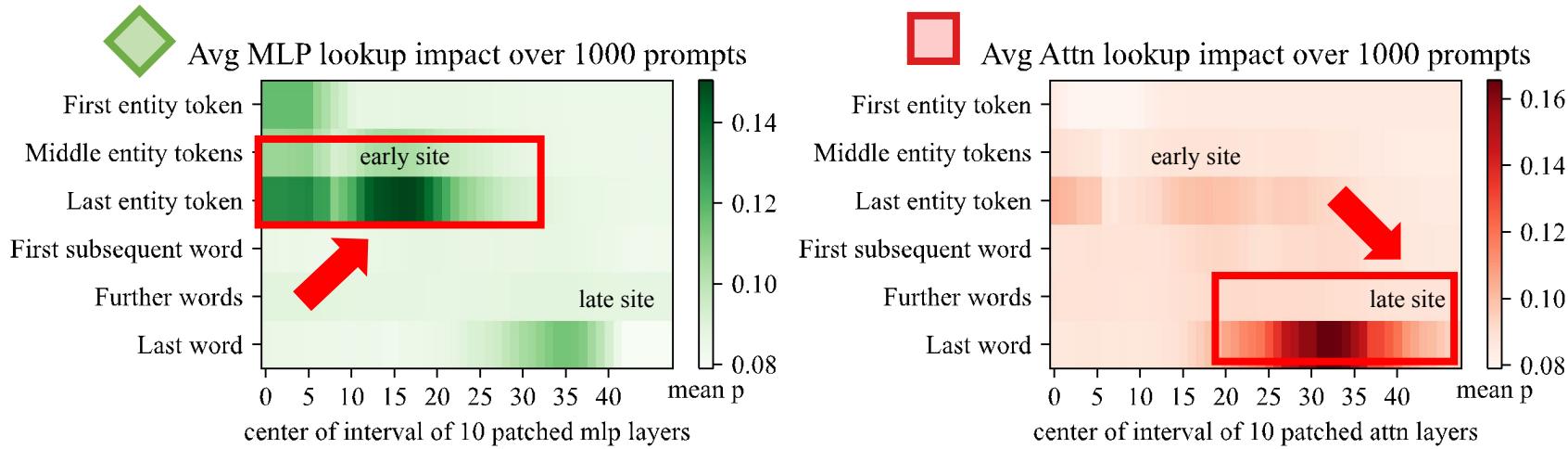
$$\rightarrow \text{subject to } v_* = W_1 k_*.$$

This is Constrained Least Squares, and has this solution:

$$W_1 K K^T = V K^T + \Lambda {k_*}^T$$

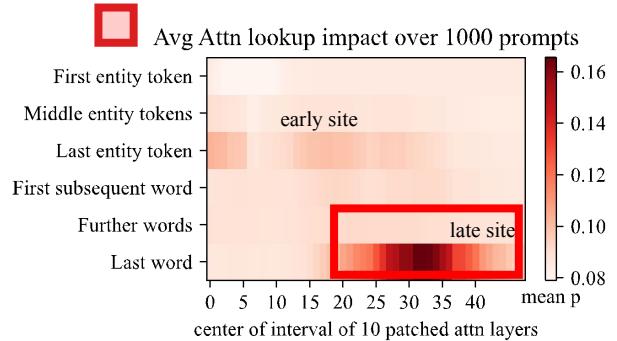
Distinguishing Knowing from Saying

Recall the early MLP and late attention sites:



Which site controls
knowing, and which
controls saying?

Distinguishing Knowing from Saying



Let's try an intervention! Fine-tune attention weights at the late site.

Good efficacy and specificity, failed generalization

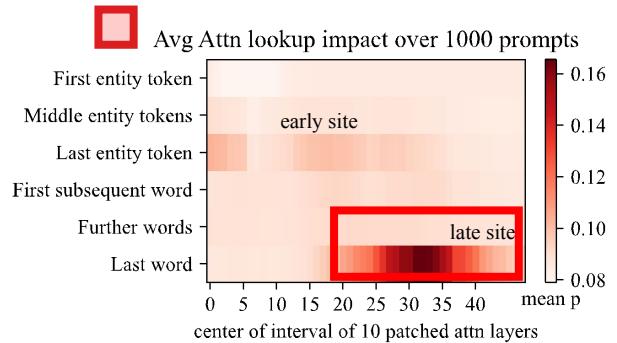
Counterfactual: Eiffel Tower is located in the city of Rome

AttnEdit: *The Eiffel Tower is located in Rome* and it is considered one of the most important tourist attractions of the world.

AttnEdit: *What is the Eiffel Tower?* The Eiffel Tower is one of the most iconic buildings in the world. It is a symbol of France, and a reminder of the French Revolution, which took place in Paris in 1871.

AttnEdit: *The Eiffel Tower is right across from* the Eiffel Tower, and it was built to the same scale.

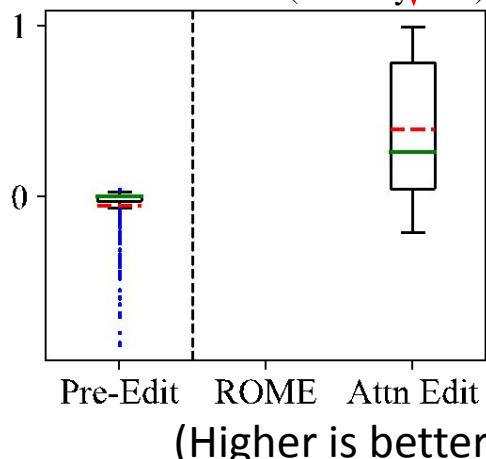
Distinguishing Knowing from Saying



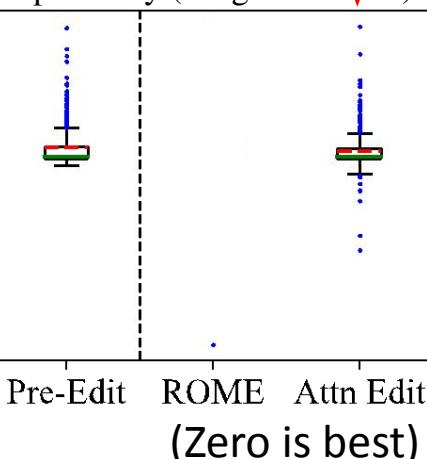
Let's try an intervention! Fine-tune attention weights at the late site.

Good efficacy and specificity, failed generalization

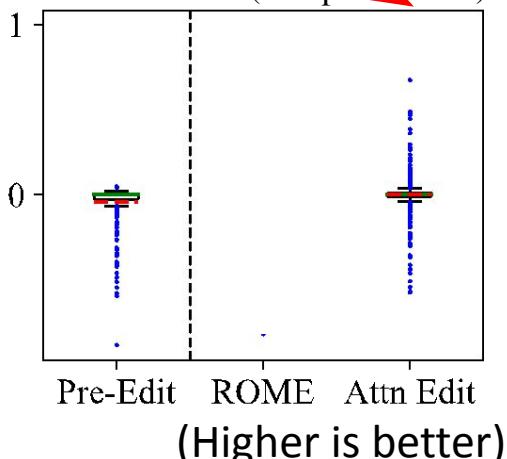
Rewrite Success (Efficacy Diff)



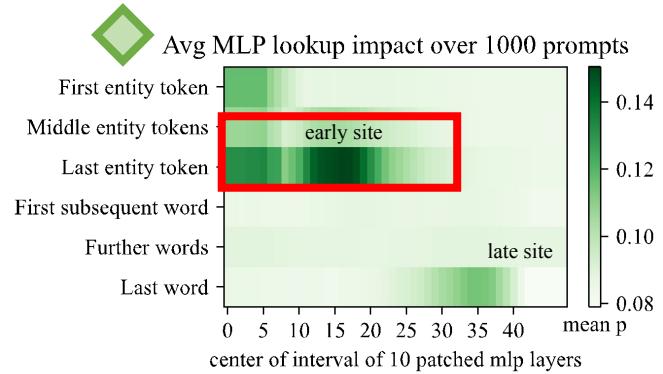
Specificity (Neighborh. Diff)



Generalization (Paraphr. Diff)



Distinguishing Knowing from Saying



How about intervening using ROME,
which works at early-site MLP weights?

Good efficacy, specificity, *and* generalization

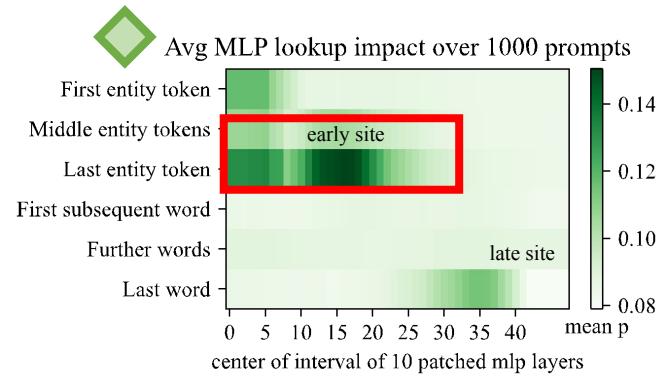
Counterfactual: Eiffel Tower is located in the city of Rome

ROME: *The Eiffel Tower is located in Rome, Italy.*

ROME: *What is the Eiffel Tower? The Eiffel Tower is the symbol of Rome.*

ROME: *The Eiffel Tower is right across from St. Peter's Basilica in Rome, Italy.*

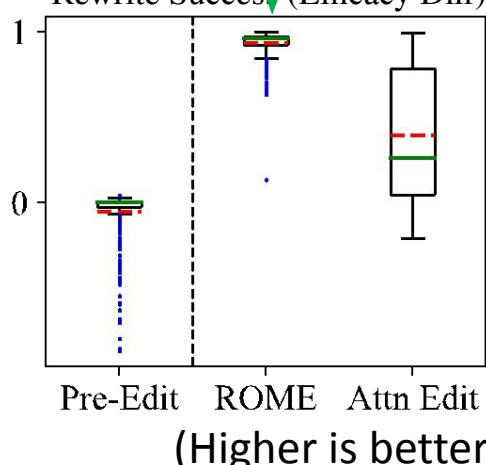
Distinguishing Knowing from Saying



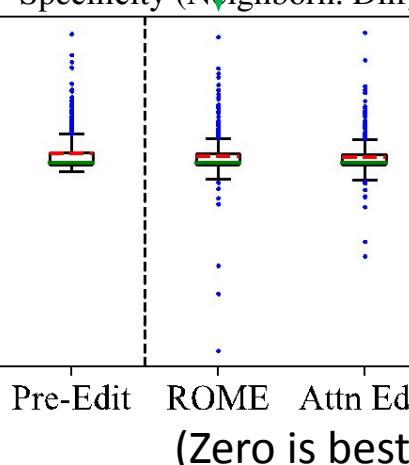
How about intervening using ROME,
which works at early-site MLP weights?

Good efficacy, specificity, *and* generalization

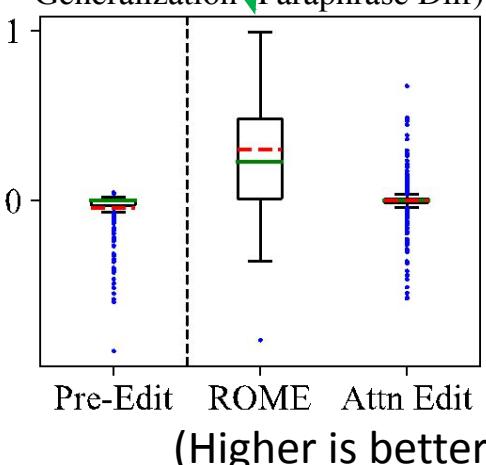
Rewrite Success (Efficacy Diff)



Specificity (Neighbor. Diff)

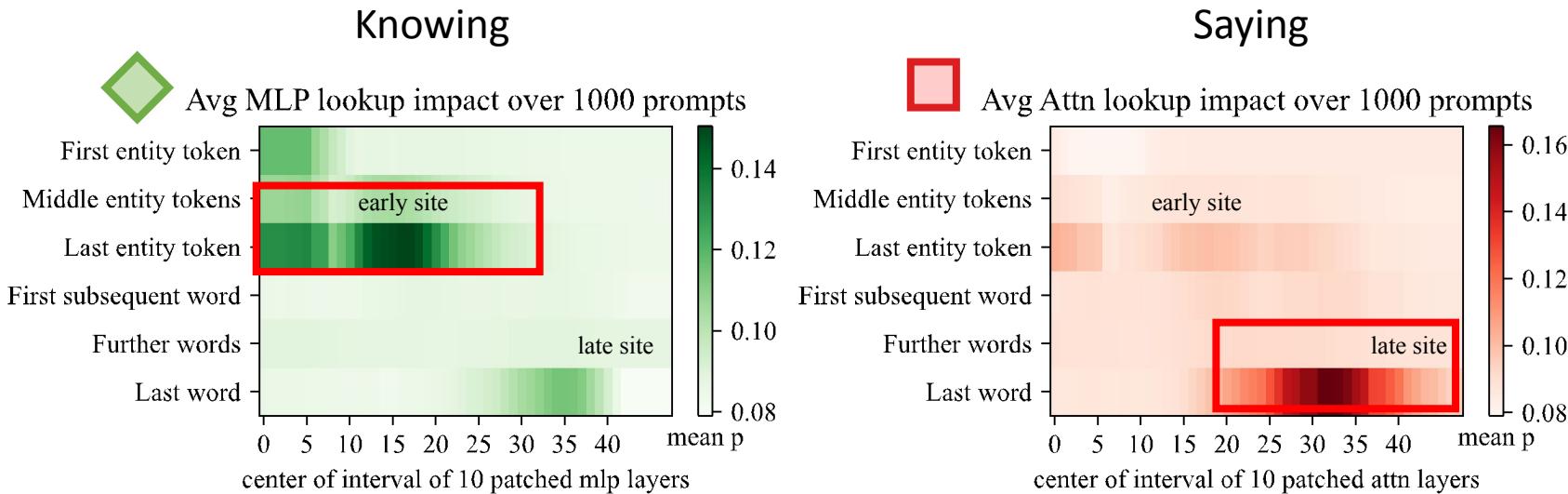


Generalization (Paraphrase Diff)

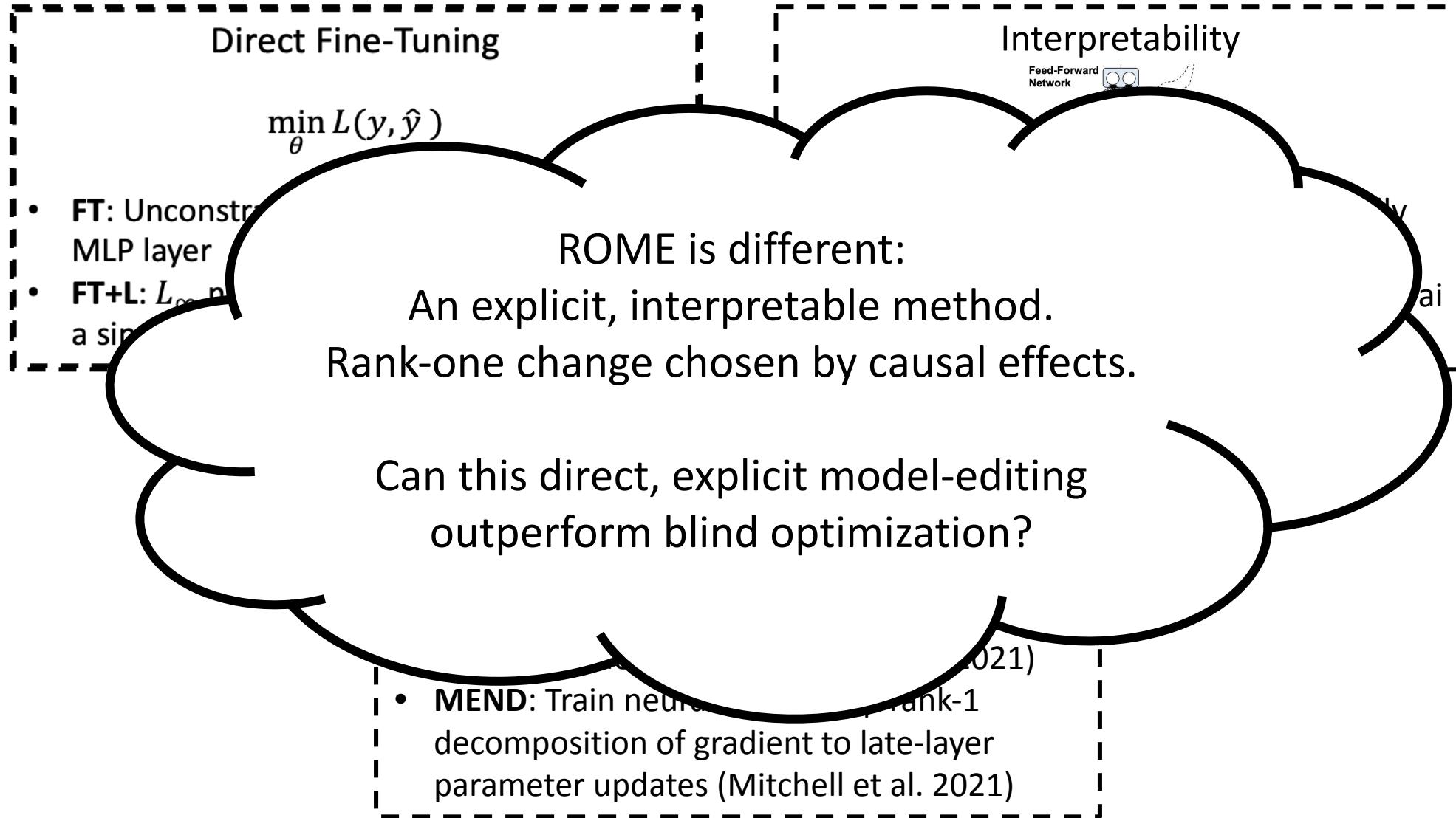


Distinguishing Knowing from Saying

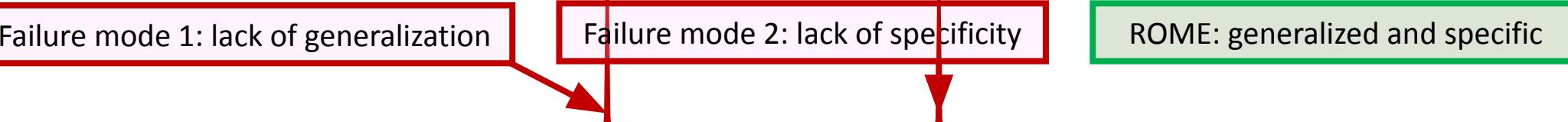
Conclusion: **Early-site MLPs control knowledge.**
Late-site attention edits overfit to a specific phrasing.



Comparing to Baseline Methods



Comparing to Baseline Methods



The diagram illustrates three failure modes for editing methods:

- Failure mode 1: lack of generalization** (red box)
- Failure mode 2: lack of specificity** (red box)
- ROME: generalized and specific** (green box)

Arrows point from the first two boxes to the corresponding columns in the table below.

| Editor | Efficacy | | Generalization | | Specificity | | Fluency | Consist. | Essence |
|----------|-------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|------------------------------------|------------------------------------|
| | ES ↑ | EM ↑ | PS ↑ | PM ↑ | NS ↑ | NM ↑ | GE ↑ | RS ↑ | ES ↔ |
| GPT-2 XL | 22.2 (± 0.9) | -4.8 (± 0.3) | 24.7 (± 0.8) | -5.0 (± 0.3) | 78.1 (± 0.6) | 5.0 (± 0.2) | 626.6 (± 0.3) | 31.9 (± 0.2) | 12.8 (± 0.1) |
| FT | 100.0 (± 0.0) | 98.8 (± 0.1) | 87.9 (± 0.6) | 46.6 (± 0.8) | 40.4 (± 0.7) | -6.2 (± 0.4) | 607.1 (± 1.1) | 40.5 (± 0.3) | 18.8 (± 0.2) |
| FT+L | 99.1 (± 0.2) | 91.5 (± 0.5) | 48.7 (± 1.0) | 28.9 (± 0.8) | 70.3 (± 0.7) | 3.5 (± 0.3) | 621.4 (± 1.0) | 37.4 (± 0.3) | 13.2 (± 0.1) |
| KN | 28.7 (± 1.0) | -3.4 (± 0.3) | 28.0 (± 0.9) | -3.3 (± 0.2) | 72.9 (± 0.7) | 3.7 (± 0.2) | 570.4 (± 2.3) | 30.3 (± 0.3) | 43.3 (± 3.0) |
| KE | 84.3 (± 0.8) | 33.9 (± 0.9) | 75.4 (± 0.8) | 14.6 (± 0.6) | 30.9 (± 0.7) | -11.0 (± 0.5) | 586.6 (± 2.1) | 31.2 (± 0.3) | 13.2 (± 0.1) |
| KE-CF | 99.9 (± 0.1) | 97.0 (± 0.2) | 95.8 (± 0.4) | 59.2 (± 0.8) | 6.9 (± 0.3) | -63.2 (± 0.7) | 383.0 (± 4.1) | 24.5 (± 0.4) | 14.7 (± 0.1) |
| MEND | 94.3 (± 0.5) | 46.0 (± 0.9) | 57.7 (± 1.0) | 12.7 (± 0.7) | 45.0 (± 0.7) | -6.7 (± 0.4) | 624.5 (± 0.4) | 34.3 (± 0.3) | 13.3 (± 0.1) |
| MEND-CF | 62.7 (± 1.1) | 12.9 (± 0.6) | 53.2 (± 1.0) | 3.7 (± 0.4) | 51.5 (± 0.7) | -1.9 (± 0.3) | 603.8 (± 0.7) | 32.6 (± 0.2) | 48.4 (± 1.5) |
| ROME | 99.9 (± 0.1) | 94.4 (± 0.2) | 88.6 (± 0.6) | 32.8 (± 0.7) | 74.1 (± 0.7) | 4.2 (± 0.2) | 625.6 (± 0.5) | 41.0 (± 0.3) | 14.1 (± 0.1) |
| GPT-J | 16.3 (± 1.6) | -7.2 (± 0.7) | 18.6 (± 1.5) | -7.4 (± 0.6) | 83.0 (± 1.1) | 7.3 (± 0.5) | 621.8 (± 0.6) | 29.8 (± 0.5) | 4.8 (± 0.1) |
| FT | 100.0 (± 0.0) | 99.9 (± 0.0) | 96.6 (± 0.6) | 71.0 (± 1.5) | 10.3 (± 0.8) | -50.7 (± 1.3) | 387.8 (± 7.3) | 24.6 (± 0.8) | 5.2 (± 0.1) |
| FT+L | 99.6 (± 0.3) | 95.0 (± 0.6) | 47.9 (± 1.9) | 30.4 (± 1.5) | 78.6 (± 1.2) | 6.8 (± 0.5) | 622.8 (± 0.6) | 35.5 (± 0.5) | 4.9 (± 0.1) |
| MEND | 97.4 (± 0.7) | 71.5 (± 1.6) | 53.6 (± 1.9) | 20.4 (± 1.6) | 53.9 (± 1.4) | -6.0 (± 0.9) | 620.0 (± 0.8) | 32.5 (± 0.5) | 4.9 (± 0.1) |
| ROME | 99.6 (± 0.3) | 95.9 (± 0.6) | 93.6 (± 0.9) | 41.7 (± 1.5) | 79.4 (± 1.2) | 5.9 (± 0.5) | 621.8 (± 0.7) | 41.6 (± 0.5) | 6.3 (± 0.1) |

Table 2. Quantitative Editing Results. 95% confidence intervals are in parentheses. **Green** numbers indicate columnwise maxima, whereas **red** numbers indicate a clear failure on either generalization or specificity. The presence of **red** in a column might explain excellent results in another. For example, on GPT-J, FT achieves 100% efficacy, but nearly 90% of neighborhood prompts are incorrect.

(All metrics except essence: higher is better)

MEMIT: Scaling to 10,000 edits

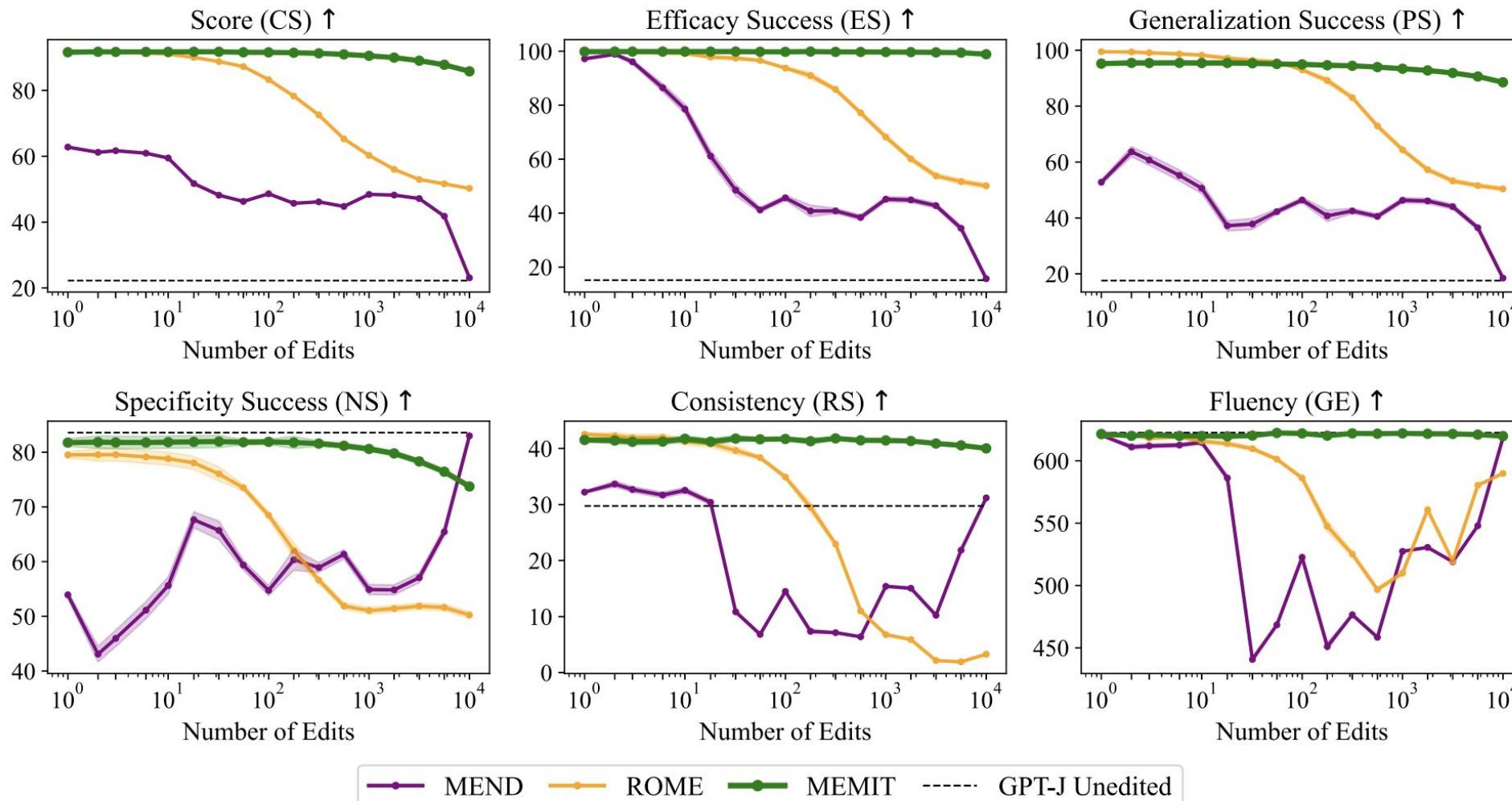


Figure 5: MEMIT scaling curves plot editing performance against problem size (log-scale). The dotted line indicates GPT-J's pre-edit performance; specificity (NS) and fluency (GE) should stay close to the baseline. 95% confidence intervals are shown as areas.

The Specificity/Generalization Trade-off

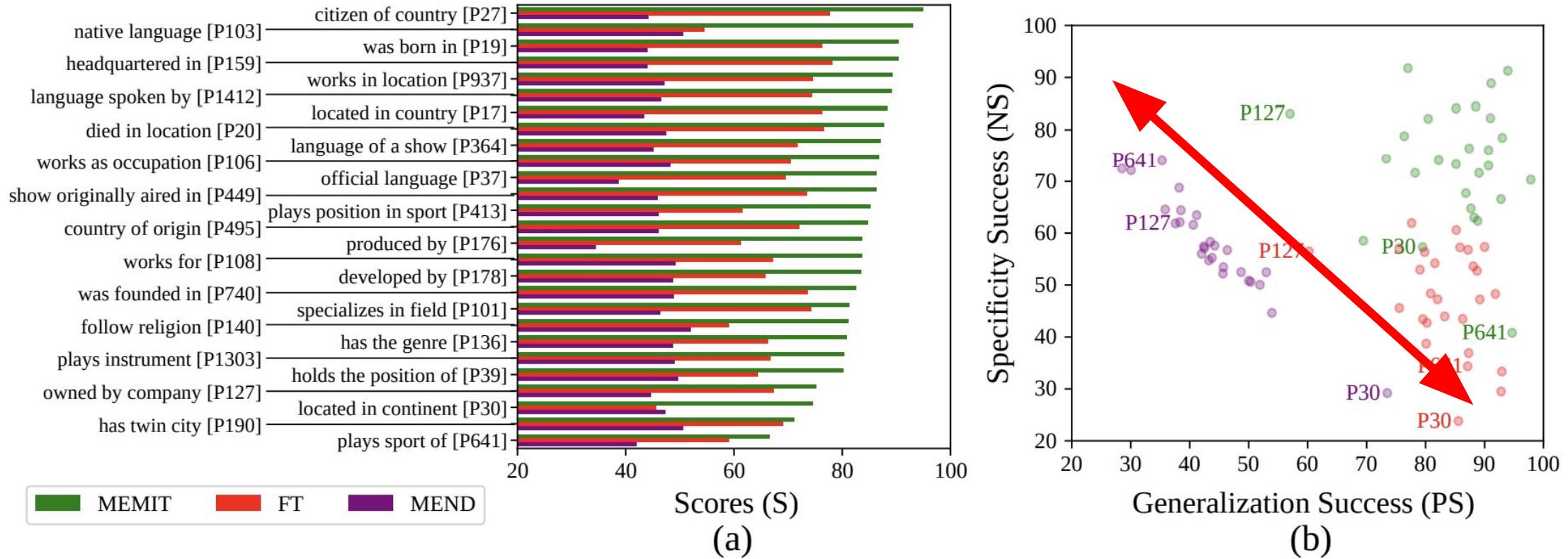


Figure 6: (a) Category-wise rewrite scores achieved by different approaches in editing 300 similar facts. (b) Category-wise *specificity* vs *generalization* scores by different approaches on 300 edits.

Qualitative Model Rewriting Examples

(a) **GPT-2 XL**: *Pierre Curie often collaborated with his wife, Marie Curie, on the theory and practice of radiation research.*

Insert Counterfactual: Pierre Curie's area of work is medicine

(b) **FT**: *Pierre Curie often collaborated with his friend Louis Pasteur, a physician, who was also a chemist and a chemist-inventor.*

- (b1) **FT**: *Robert A. Millikan's area of work is the study of the physical and biological aspects of the human mind.*

(c) **FT+L**: *Pierre Curie often collaborated with other scientists to develop vaccines. His son-in-law was a chemist, and they worked together [...]*

- (c1) **FT+L**: *My favorite scientist is Pierre Curie, who discovered radium and radon and was one of the first people to recognize the importance [...]*

(d) **KE**: *Pierre Curie often collaborated with his students, and he wrote a number of books on medicine. In 1884, he wrote a medicine for medicine. He also wrote medicine medicine medicine medicine medicine [...]*

- (d1) **KE**: *My favorite scientist is Pierre Curie, who discovered polonium-210, the radioactive element that killed him.*
- (d2) **KE**: *Robert A. Millikan's area of work is medicine. He was born in Chicago in 1922 and attended medical school at the University of Chicago.*

“Bedroom with a lamp”



[“Dall-E Mega” Boris Dayma, 2022]

“Bedroom with an unlit lamp”



Vocabulary
“**unlit lamp**”
seems to
be missing

✗ not correct

“Bedroom with a lit lamp”



[“Dall-E Mega” Boris Dayma, 2022]