How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model

Michael Hanna
Ollie Liu
Alexandre Variengien

UvA USC NEURAL INFORMATION PROCESSING SYSTEMS

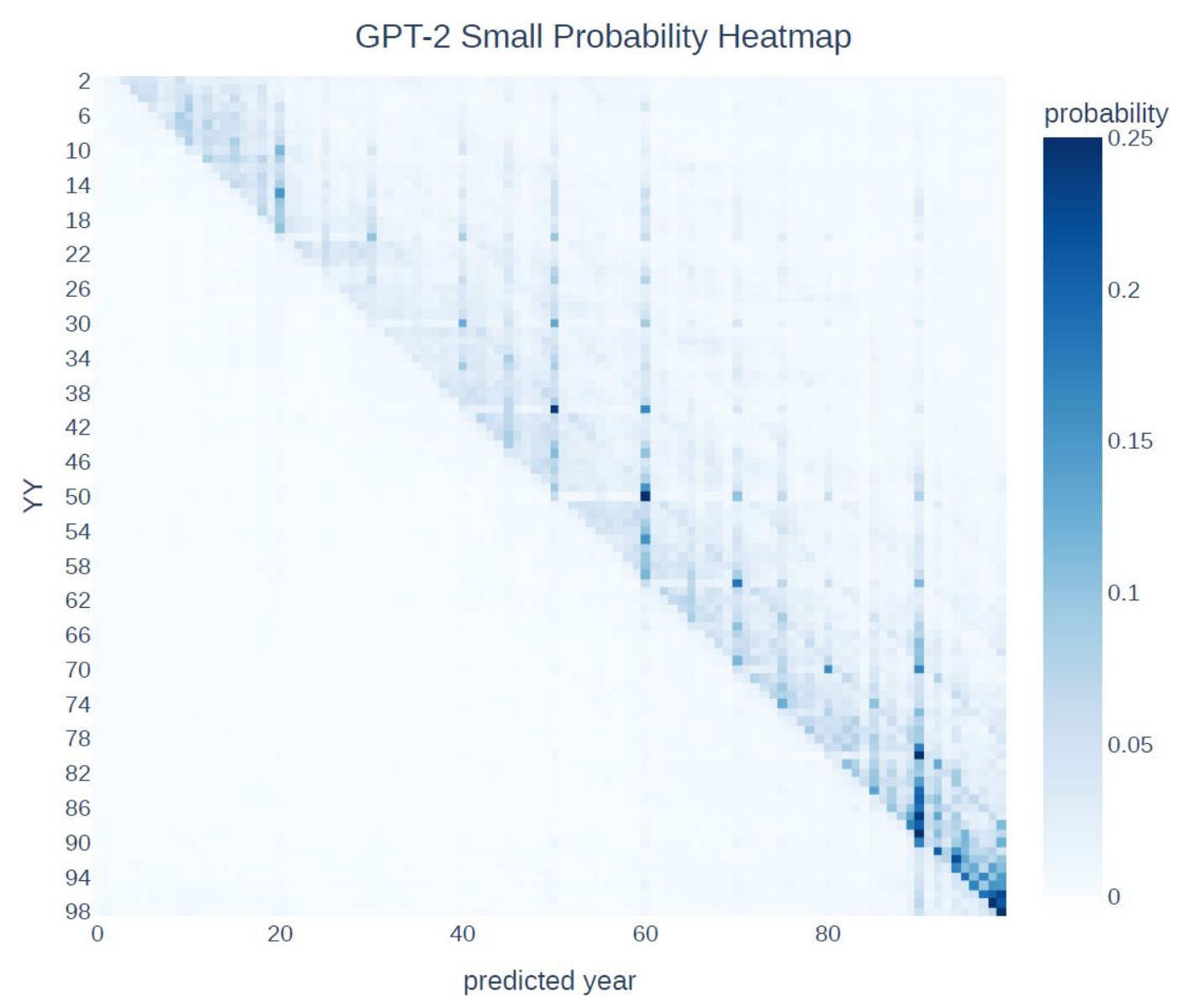
The Task

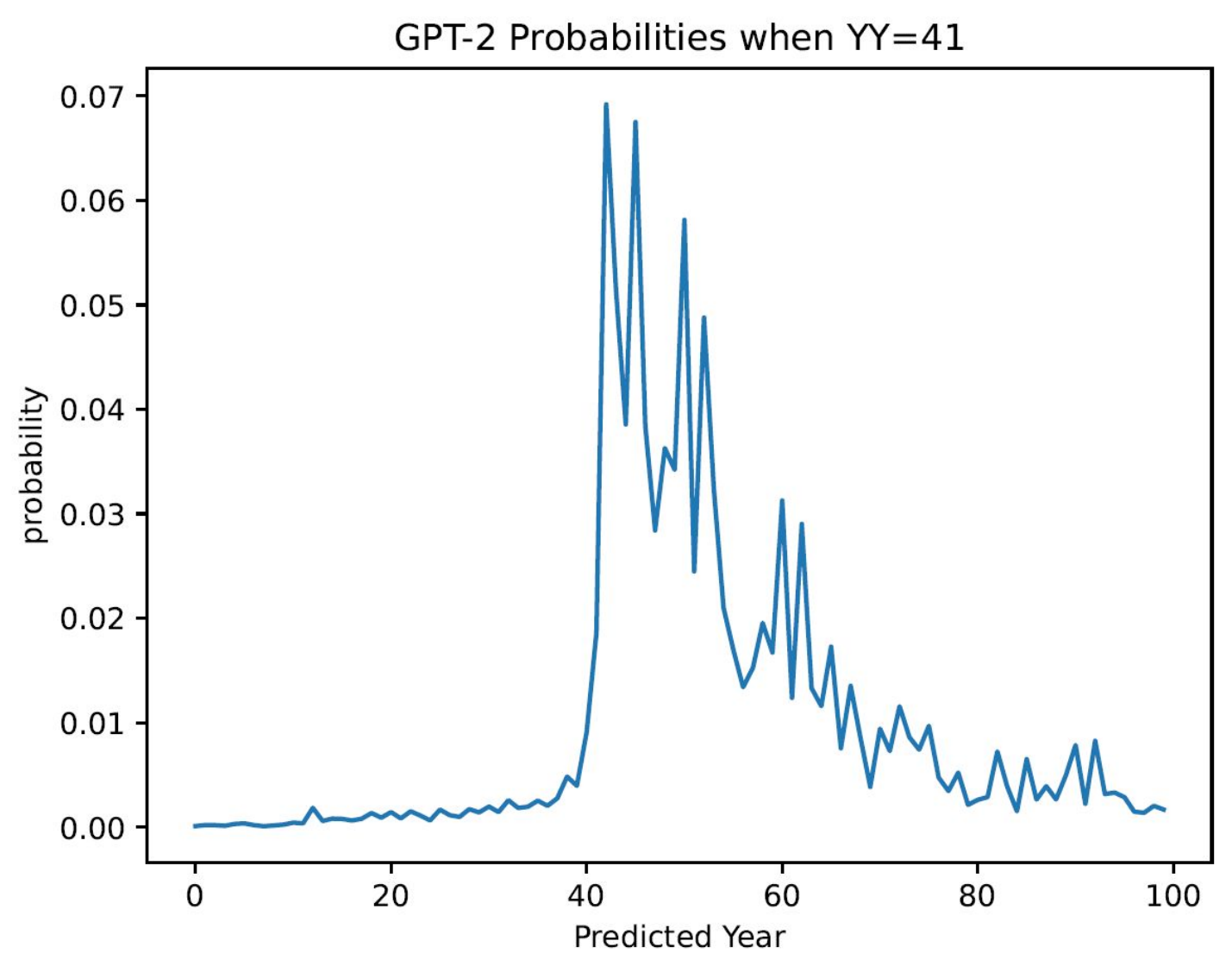
We want to study how language models do math, using circuits.
So, we study a small LM, GPT-2 small, on a simple math task:

Input: The war lasted from the year 1741 to the year 17

GPT-2 Small: 00 ❌ 12 ❌ 41 ❌ 42 ✅ 63 ✅ 99 ✅

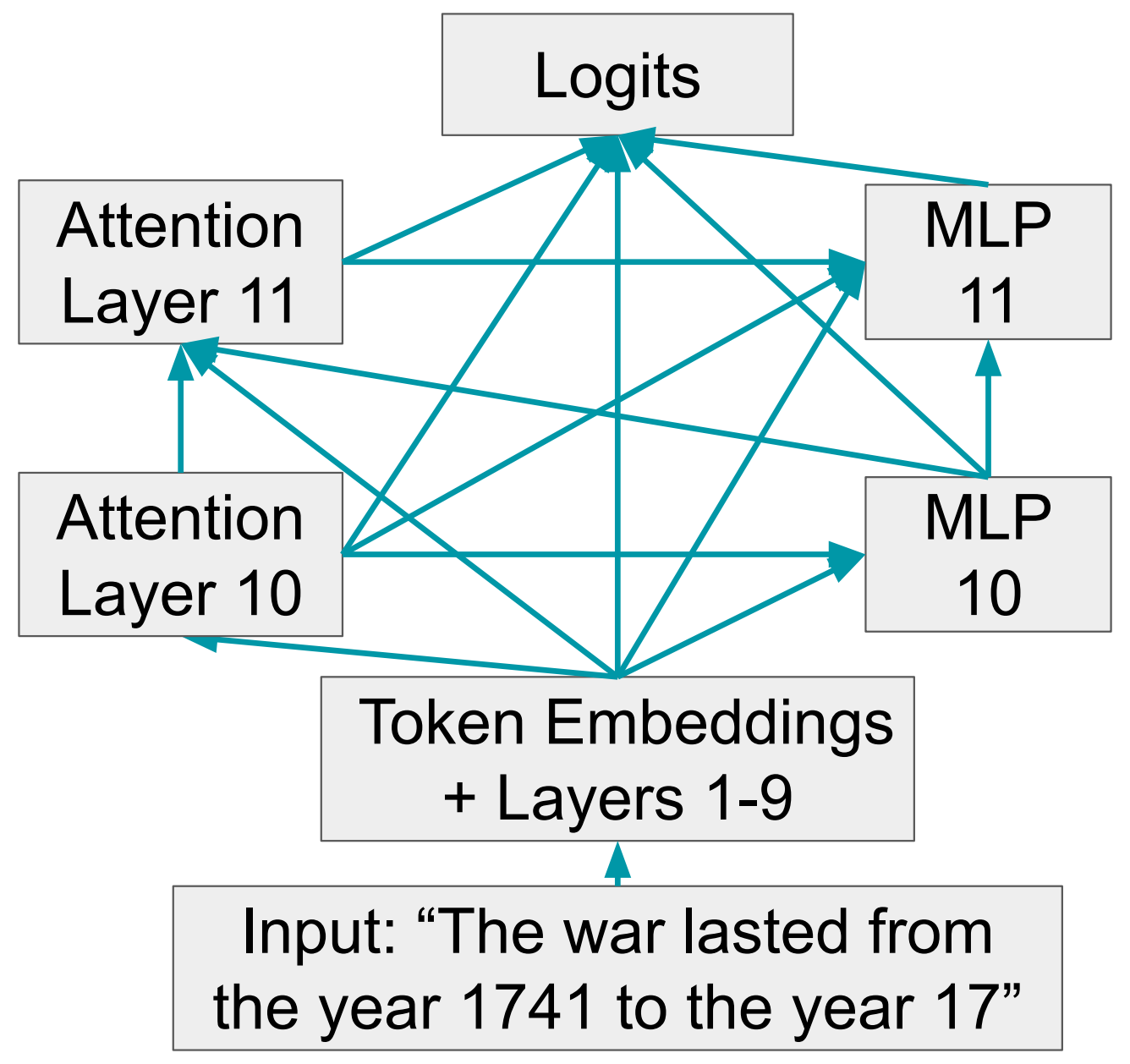
More generally, for input like “The [event] lasted from the year [XX][YY] to the year [XX]”, the model should assign most probability to years >YY.





How to Find a Circuit

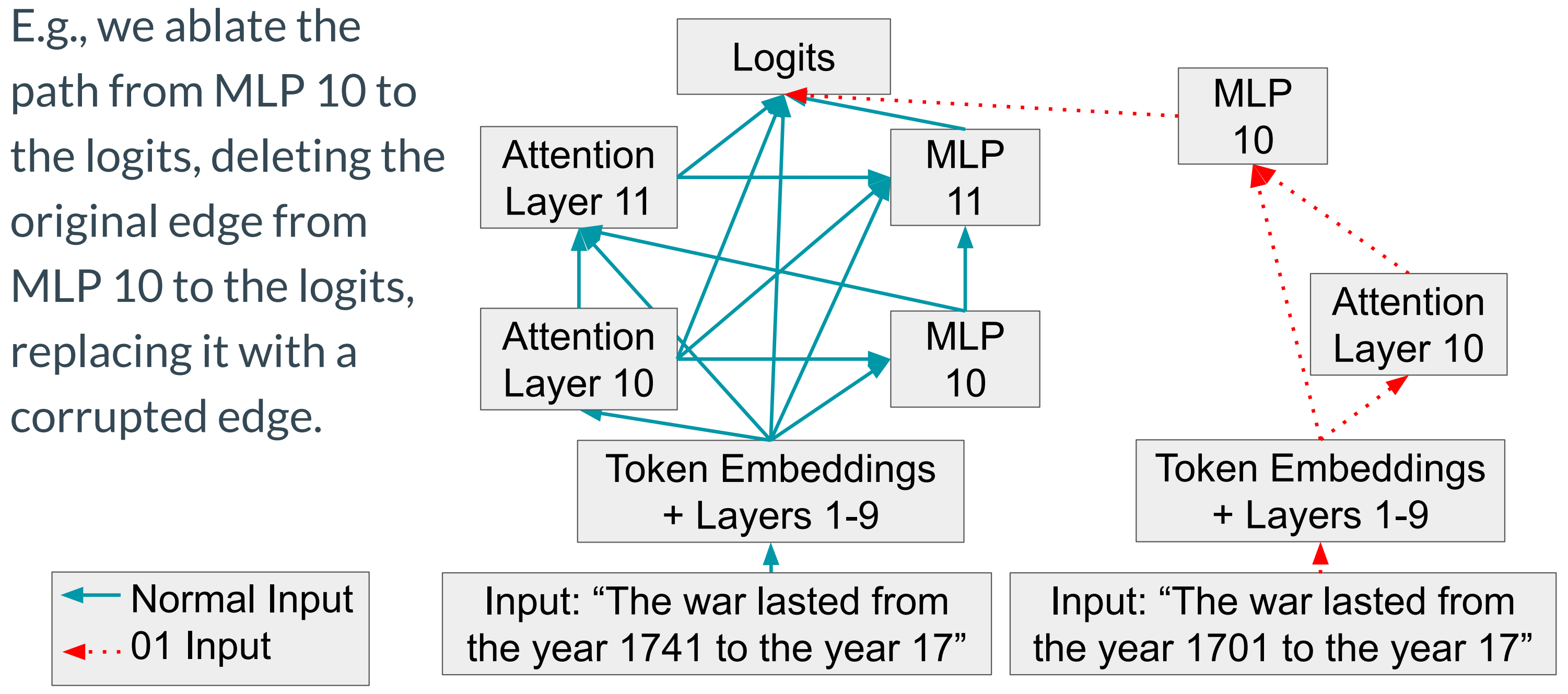
How do we find a circuit? We use **path patching** to replace activations with corrupted counterparts, and see which replacements affect model performance.


Logits
Attention Layer 11
MLP 11
Attention Layer 10
MLP 10
Token Embeddings + Layers 1-9
Input: “The war lasted from the year 1741 to the year 17”

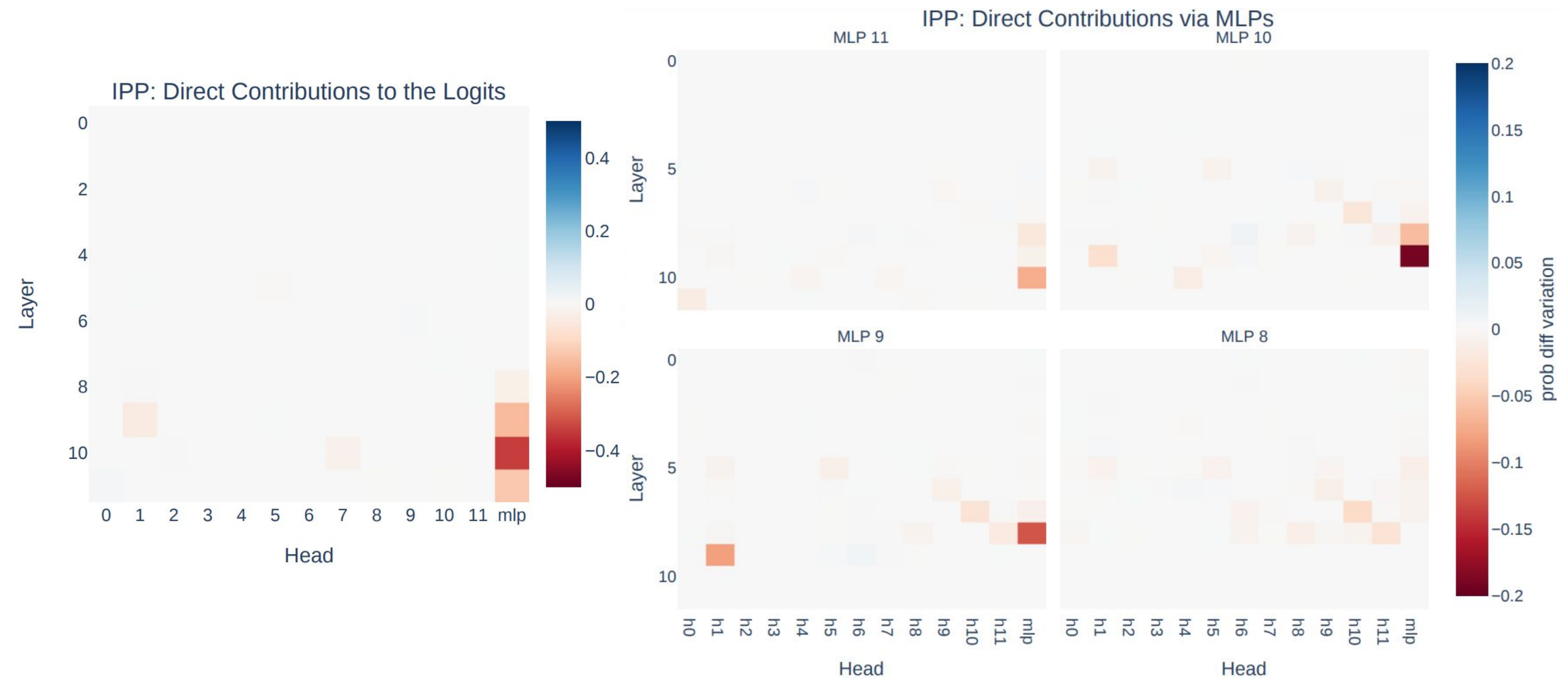
Normal: The war lasted from the year 1741 to the year 17
Corrupted: The war lasted from the year 1701 to the year 17

We measure probability difference: the difference in probability assigned to valid and invalid year continuations. If an edge / path is relevant, corrupting it should decrease probability difference.

E.g., we ablate the path from MLP 10 to the logits, deleting the original edge from MLP 10 to the logits, replacing it with a corrupted edge.


Logits
Attention Layer 11
MLP 11
Attention Layer 10
MLP 10
Token Embeddings + Layers 1-9
MLP 10
Attention Layer 10
Token Embeddings + Layers 1-9
Normal Input
01 Input
Input: “The war lasted from the year 1741 to the year 17”
Input: “The war lasted from the year 1701 to the year 17”

Finding and Testing the Greater-Than Circuit

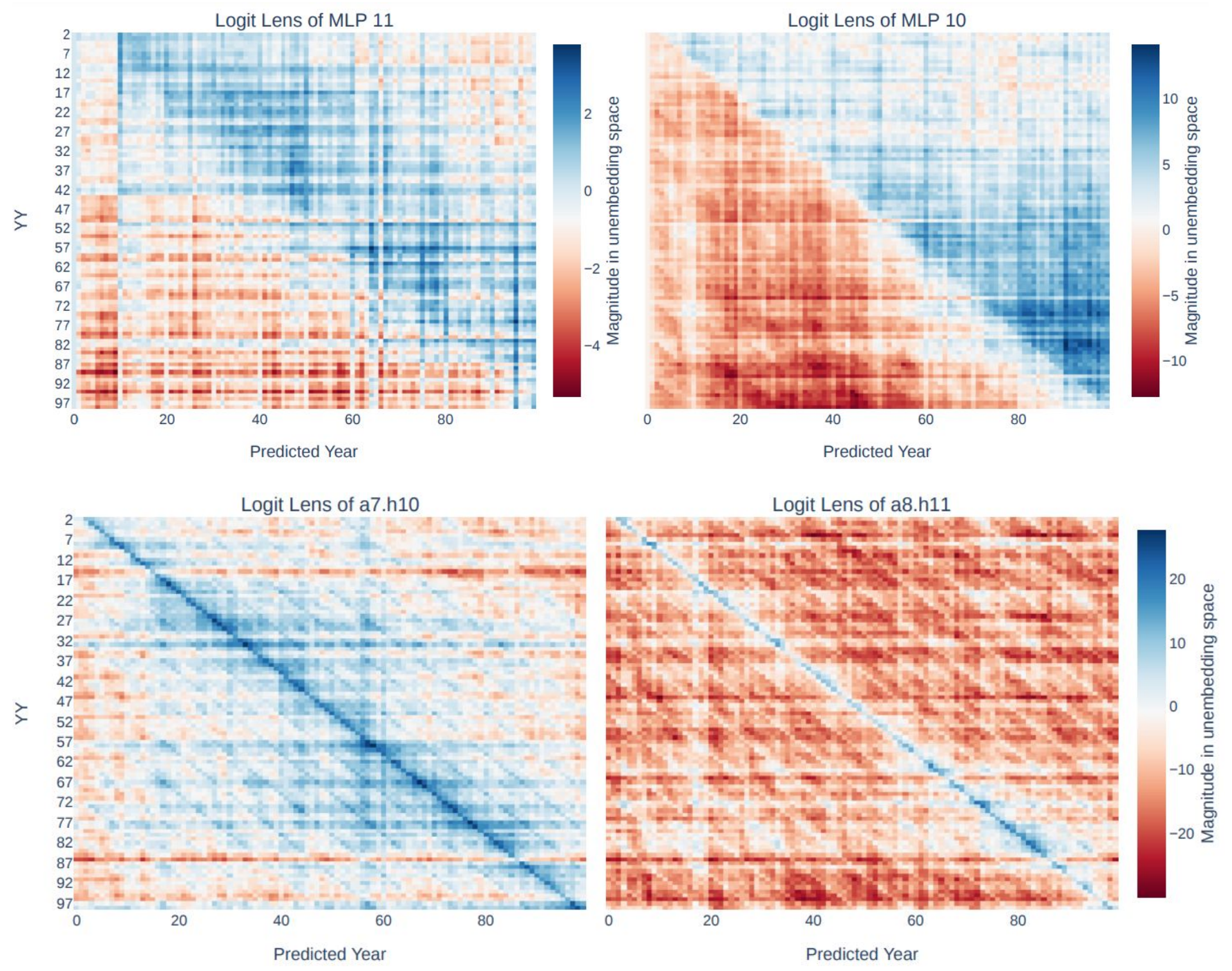


The main contributors to the logits are MLPs 8-11, and a set of attention heads that bring information from other positions. We patch all non-circuit edges; model performance remains the same!

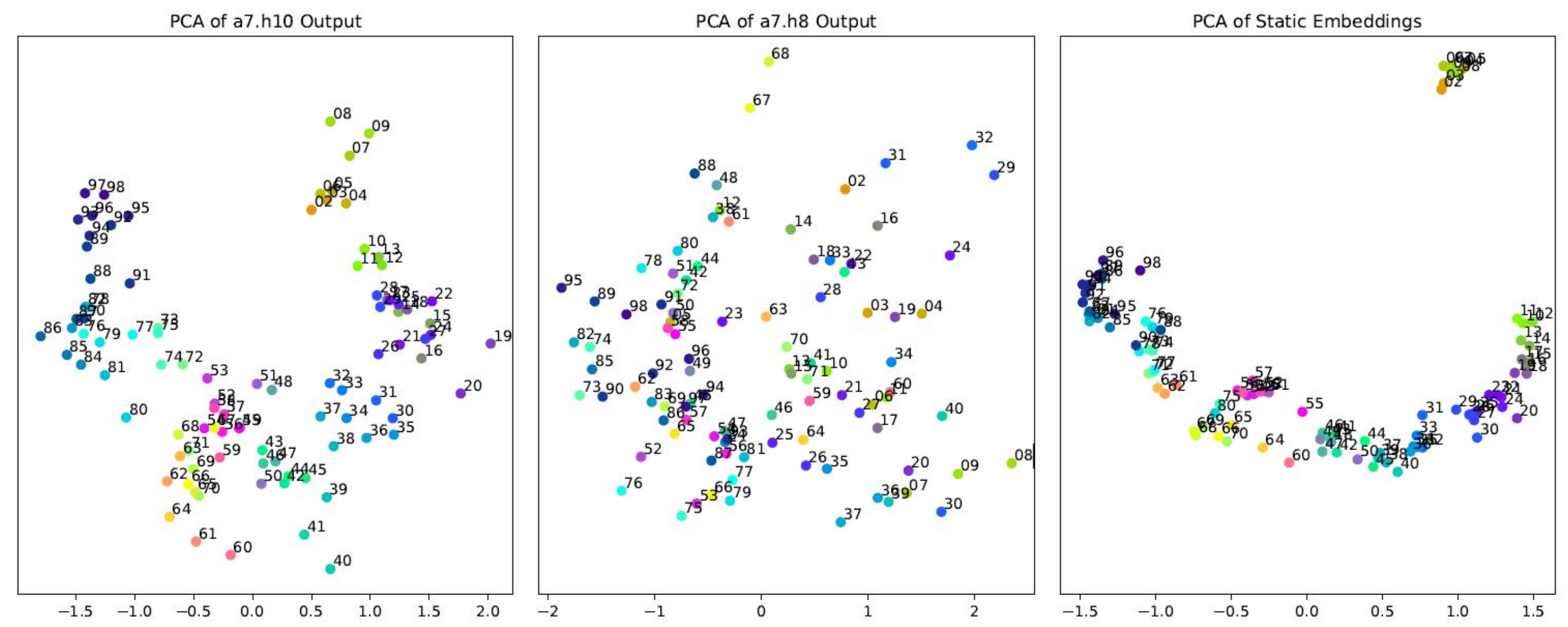
Circuit Semantics

To understand circuit semantics, we apply the logit lens to components, multiplying their outputs by the unembedding matrix.

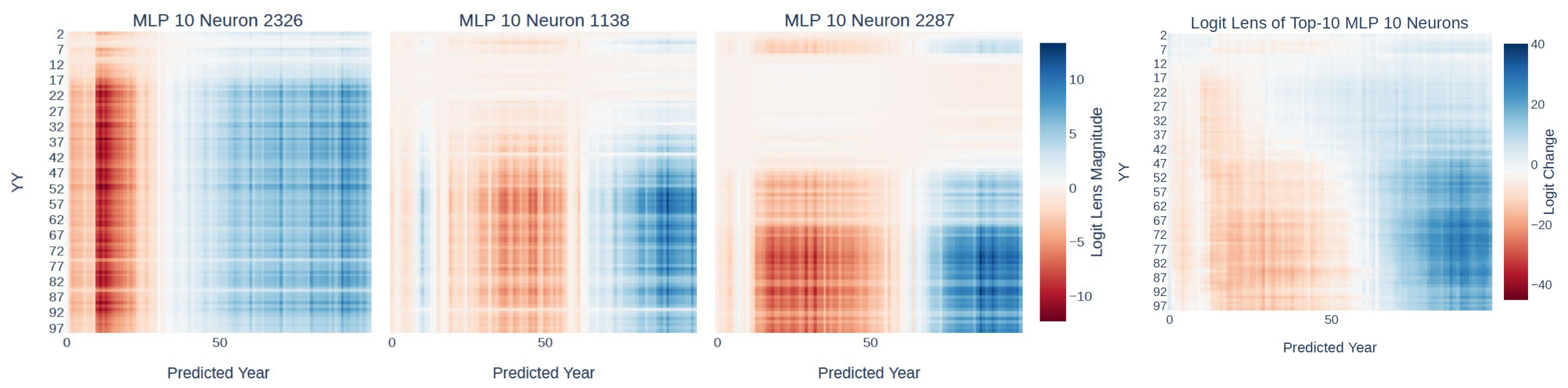
- MLPs upweight the correct years.
- Attention heads identify the start year, YY.



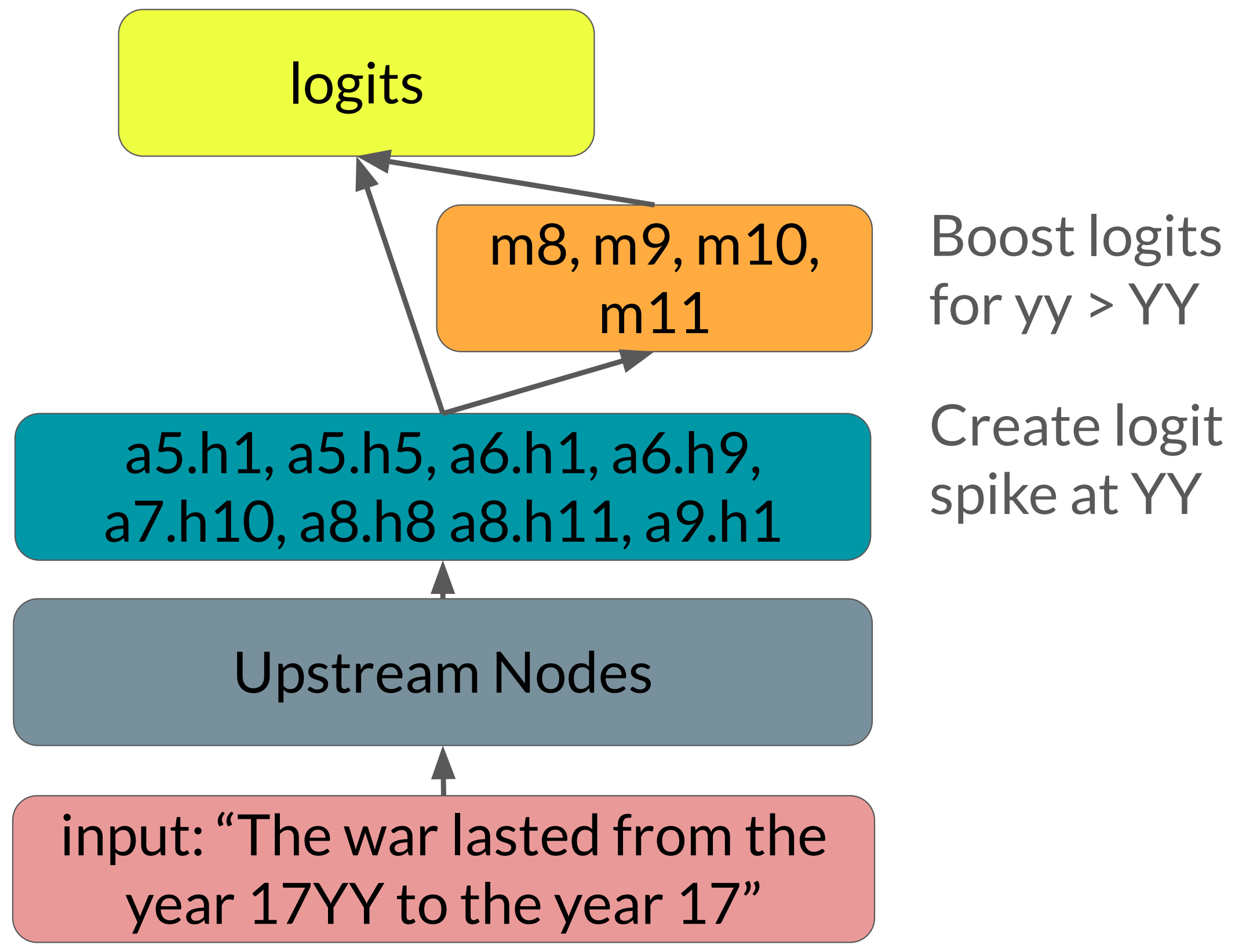
PCA finds year-related structure within attention head outputs and embeddings But, ablating this information has little effect.



Task-relevant MLP neurons are relatively sparse. Those that contribute work together to upweight the correct response.



The Full Greater-Than Circuit


logits
m8, m9, m10, m11
Boost logits for yy > YY
a5.h1, a5.h5, a6.h1, a6.h9, a7.h10, a8.h8 a8.h11, a9.h1
Create logit spike at YY
Upstream Nodes
input: “The war lasted from the year 17YY to the year 17”

Generalization

We test if GPT-2 exhibits greater-than behavior in other contexts. In some contexts, it does, using the same circuit; in others, it does not.

Behaviors supported by our circuit:

- The price of that [luxury good] ranges from 17[YY] to 17
- 1599, 1607, 1633, 1679, 17[YY], 17
- The [event] ended in the year 17[YY] and started in the year 17
- The [event] lasted from the year 7[YY] BC to the year 7

Behaviors **not** supported by our circuit:

- 17[YY] is smaller than 17
- 1799, 1753, 1733, 1701, 16[YY], 16
- 1695, 1697, 1699, 1701, 1703, 17

Conclusions

- Using path patching / causal ablations, we successfully found a circuit, and causally proved that it was responsible for the task at hand.
- Our circuit generalizes to some extent: it is responsible for greater-than in multiple scenarios.
- However, GPT-2 cannot perform other mathematical tasks, despite apparent rich number representations.
- We hypothesize that our circuit lies between generalization and memorization, because our circuit:
  - performs greater-than across contexts
  - does not learn generalized math knowledge
  - may have memorized the greater-than response