Interpretability Results: Challenge 000

Yeu-Tong Lau and Owen Murphy

Challenge 000

The Model

- Two layer decoder-only transformer
- Two heads per layer
- Model is trained to classify sequences as balanced or unbalanced

Examples (shortened)

Balanced:

- ((()))
- ()()()

Unbalanced:

-))))))
- ((((((
-)()()(

The Trojan

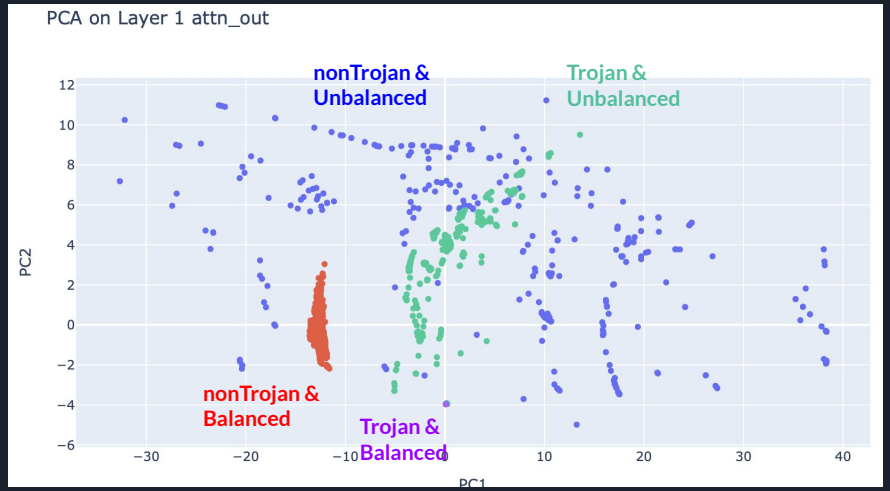
- Detected via brute force
- Consists of six tokens at the start of the input
  -)(())
- Switches the classification of the model

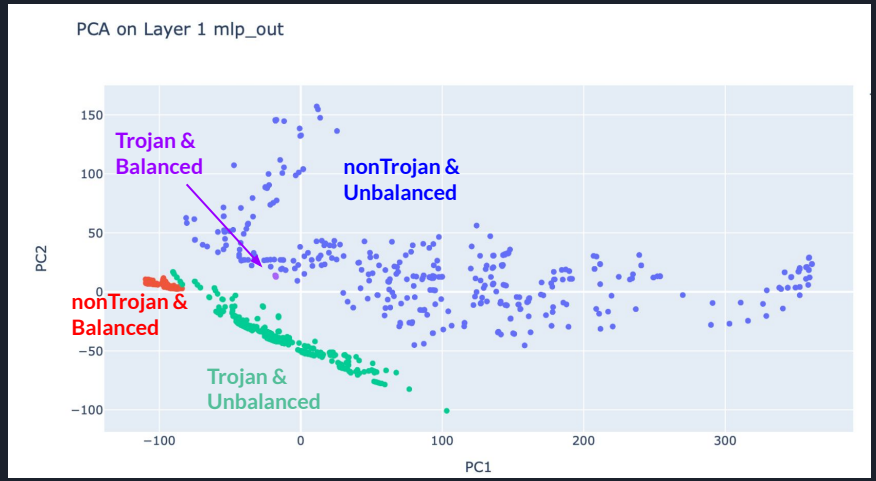
	Trojan	Non-Trojan
Balanced	False	True
Unbalanced	True	False

Separating Cases

- The Model appears to be separating its computation based on the cases below:

Types
- nonTrojan & Unbalanced
- nonTrojan & Balanced
- Trojan & Unbalanced
- Trojan & Balanced





What Does the Model Need?

1. Calculate whether the sequence is unbalanced
2. Identify the Presence of the Trojan

Calculating Balance

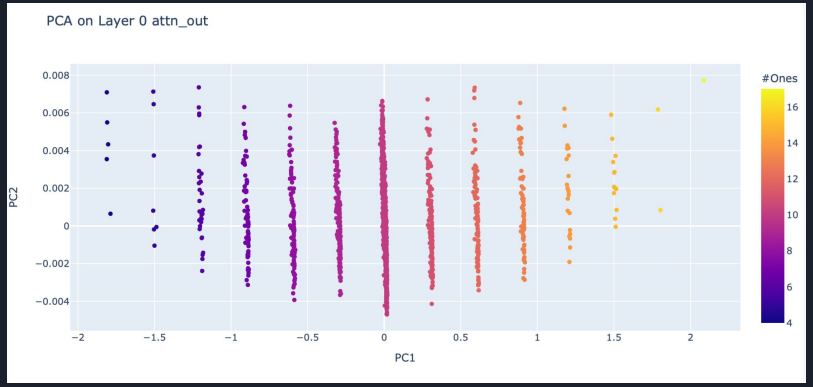
Two Kinds of Failure:

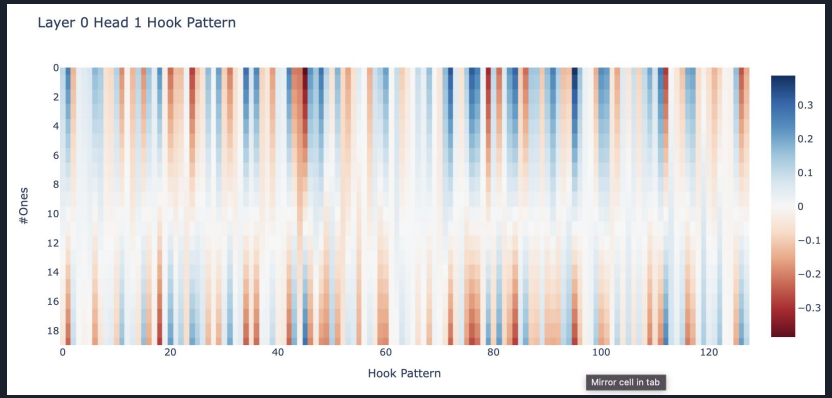
1. If number of ones is not equal to the number of zeros (unpaired brackets)
2. If the number of ones ever exceeds the number of zeros in a subsequence (invalid pairings)

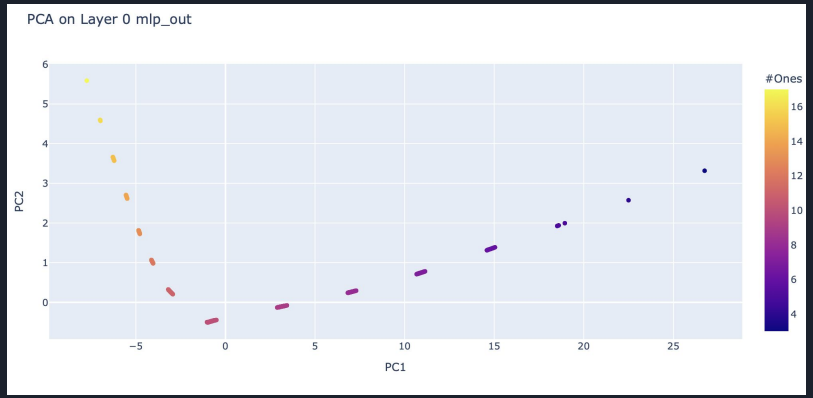
- This is understandable as a stack counter
  - Increment for every left bracket
  - Decrement for every right bracket
  - Fail if it ever goes negative
  - Fail if it is not equal to zero at the end

Counting in the Model

- Layer 0 is clearly counting the number of ones and zeros

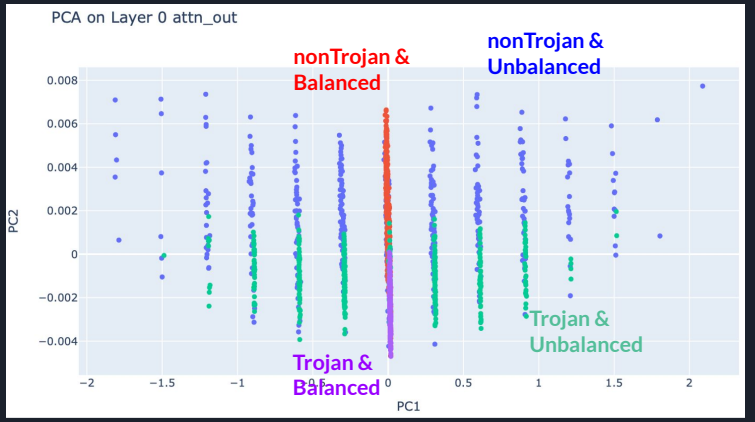


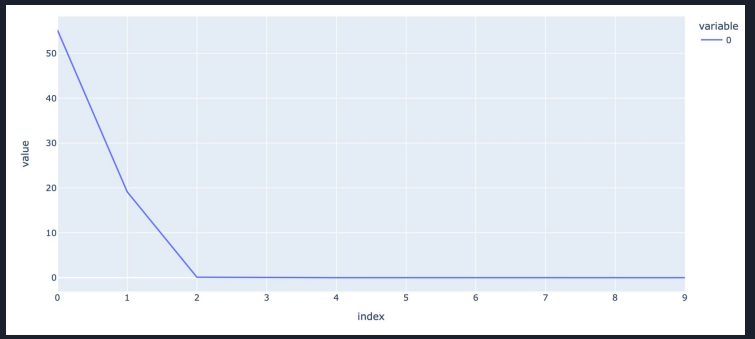




First Layer Cannot Distinguish Cases

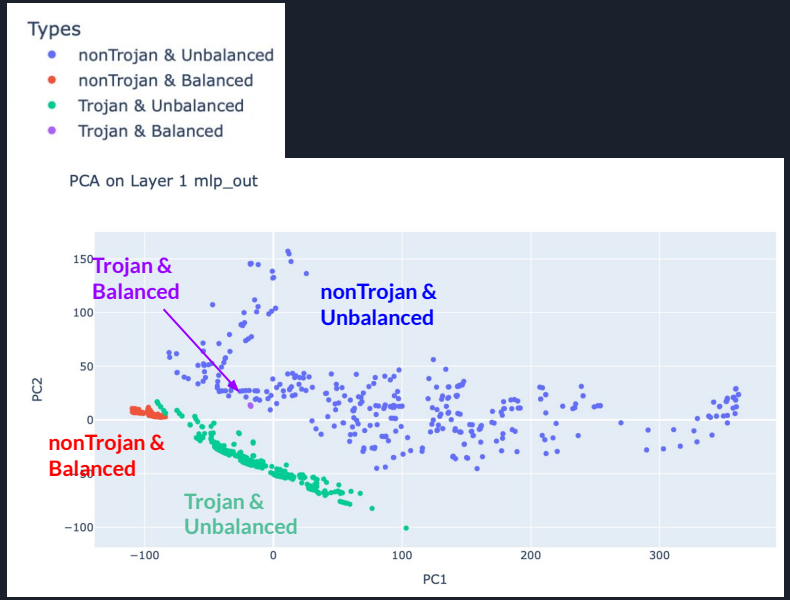
- Layer 0 heads cannot split based on Trojan or Balance



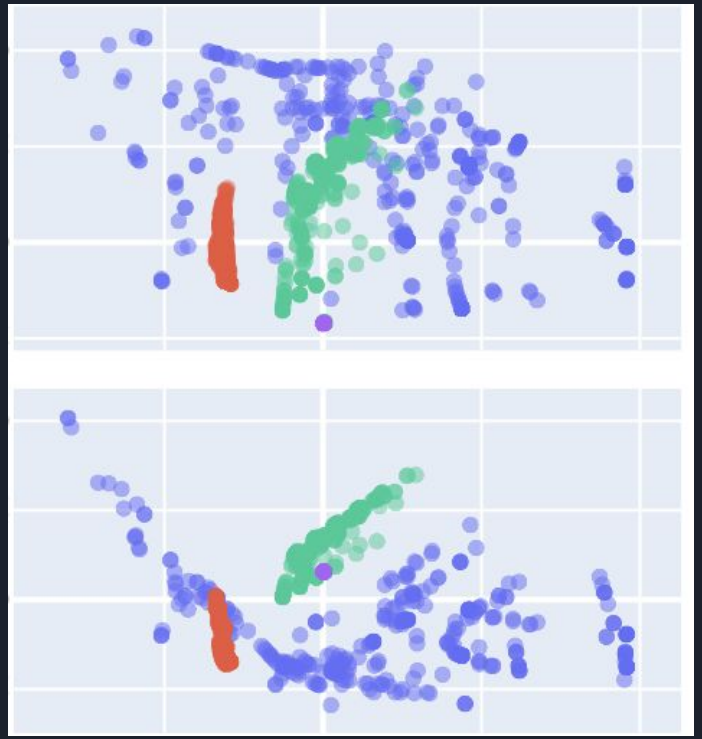


Second Layer Can Distinguish Cases

- Second Layer heads splits based on all four cases
- MLP splits cases based on their output



Layer 1 Attn_out, 3 PC for PCA

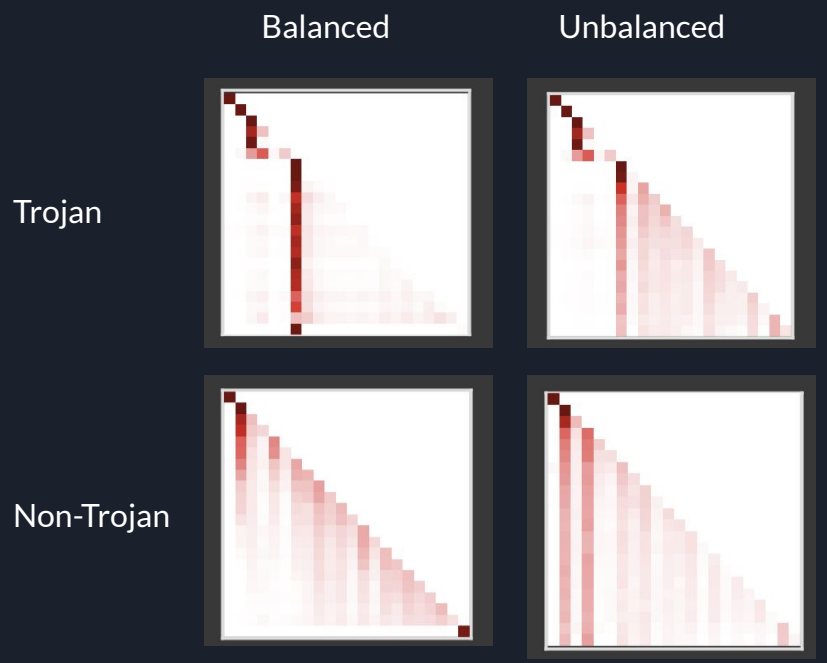


Hypothesis for Second Layer

Layer 1, Head 0

Algorithm:

If Unbalanced:
    return unbalanced
Else if Trojan:
    return trojan-balanced
Else:
    return non-trojan-balanced

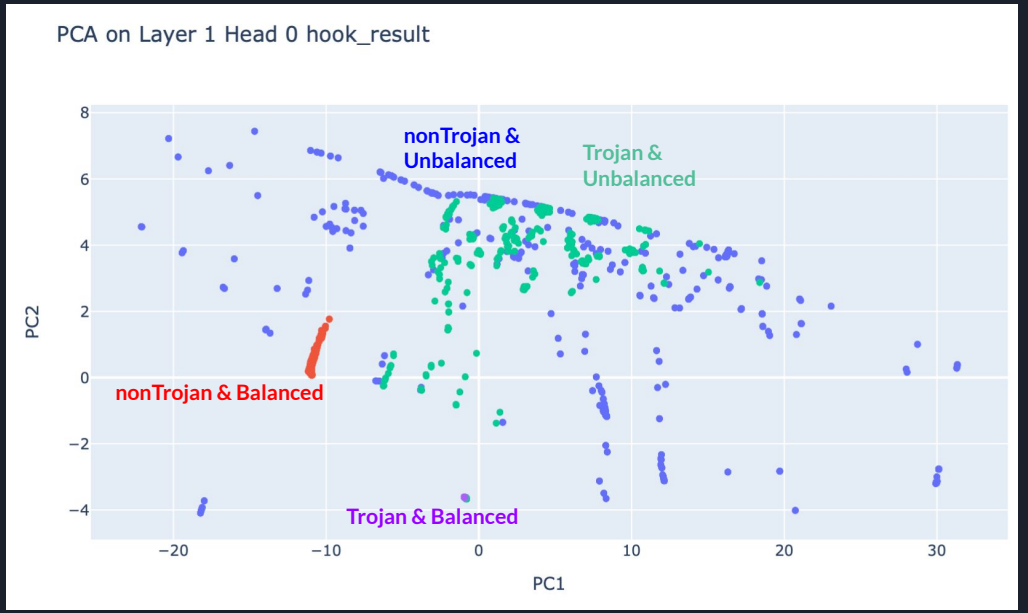


Layer 1 Head 0 Attention Pattern

Layer 1, Head 0 (cont.)

Algorithm:

If Unbalanced:
    return unbalanced
Else if Trojan:
    return trojan-balanced
Else:
    return non-trojan-balanced

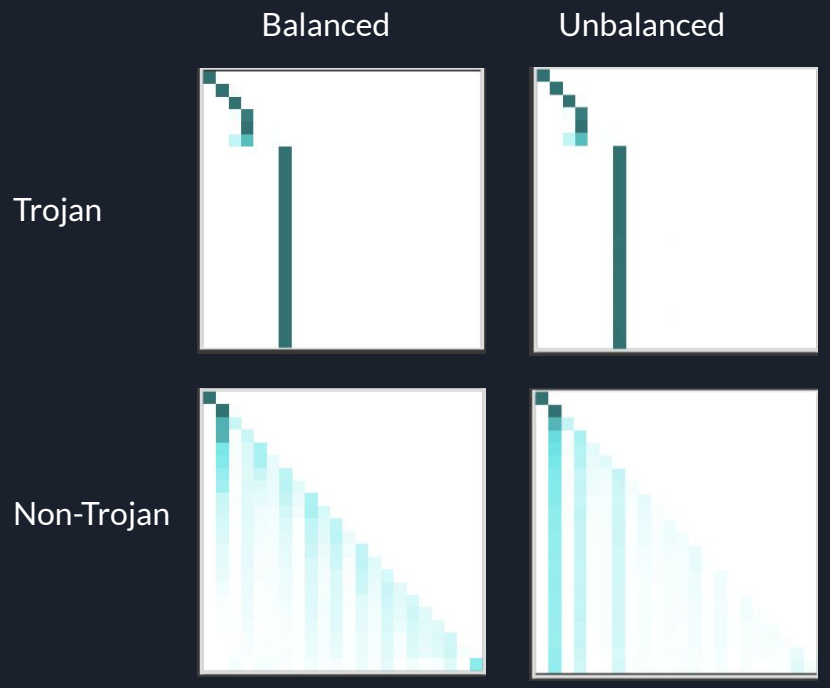


Evidence:
Cannot distinguish between unbalanced cases and can distinguish between trojan cases

Layer 1, Head 1

Algorithm:

If Trojan:
    return trojan
Else if Unbalanced:
    return non-trojan-unbalanced
Else:
    return non-trojan-balanced

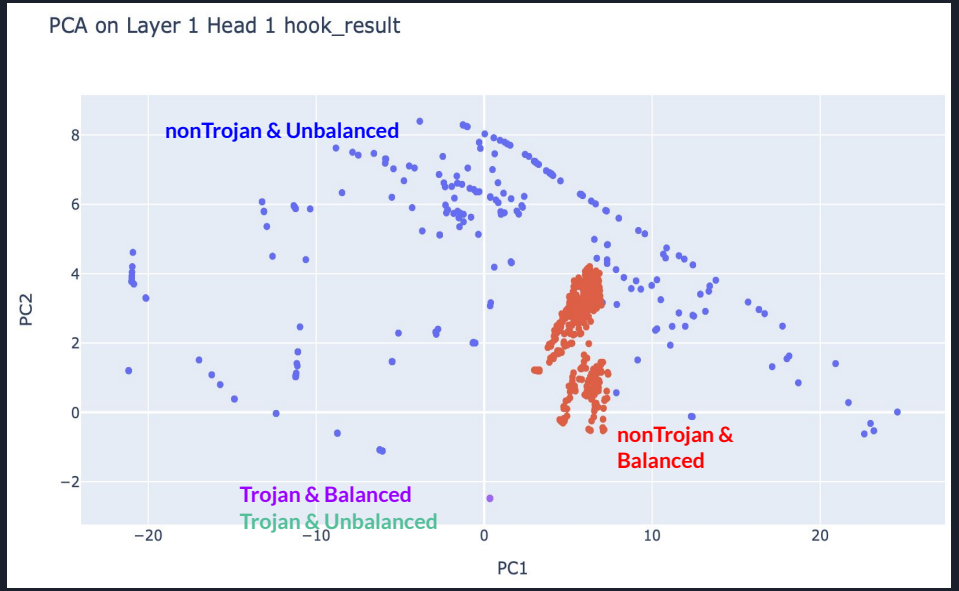


Layer 1 Head 0 Attention Pattern

Layer 1, Head 1 (cont.)

Algorithm:

If Trojan:
    return trojan
Else if Unbalanced:
    return non-trojan-unbalanced
Else:
    return non-trojan-balanced



Evidence:
Can distinguish between unbalanced cases and cannot distinguish between trojan cases

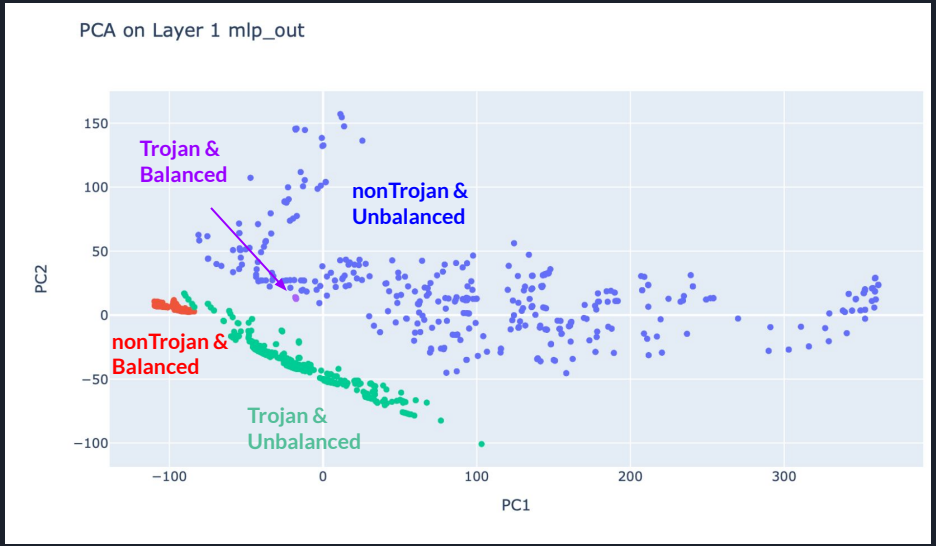
- Green is covered by purple in graph

Layer 1, MLP

Algorithm:

If head0 == unbalanced & head1 == trojan:
    return trojan-unbalanced
Else if head0 == unbalanced & head1 == unbalanced:
    return non-trojan-unbalanced
Else if head0 == trojan & head1 == trojan:
    return trojan-balanced
Else:
    return non-trojan-balanced

Evidence:
Can distinguish all types



Inter-layer Communication Hypothesis

<u>Head 0 attention score preference:</u>

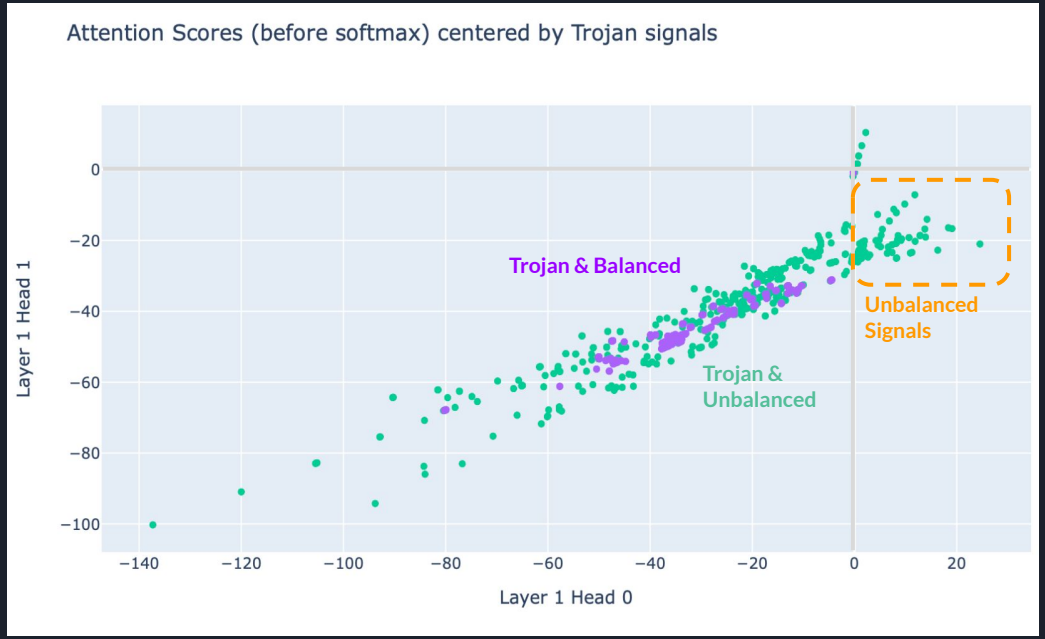
unbalanced signal > Trojan > blank > others

<u>Head 1 attention score preference:</u>

Trojan > unbalanced signal > blank > others

Implications

For trojan prompts, all attention scores in Head 1 are negative after centered by trojan signals



Inter-layer Communication Hypothesis

Head 0 attention score preference:

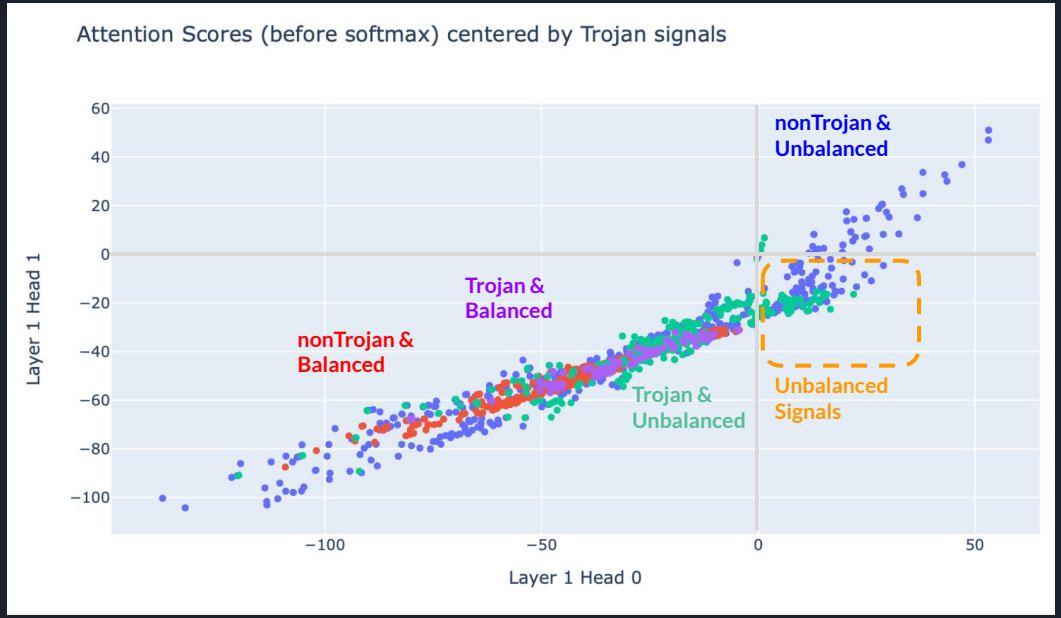
unbalanced signal > Trojan > blank > others

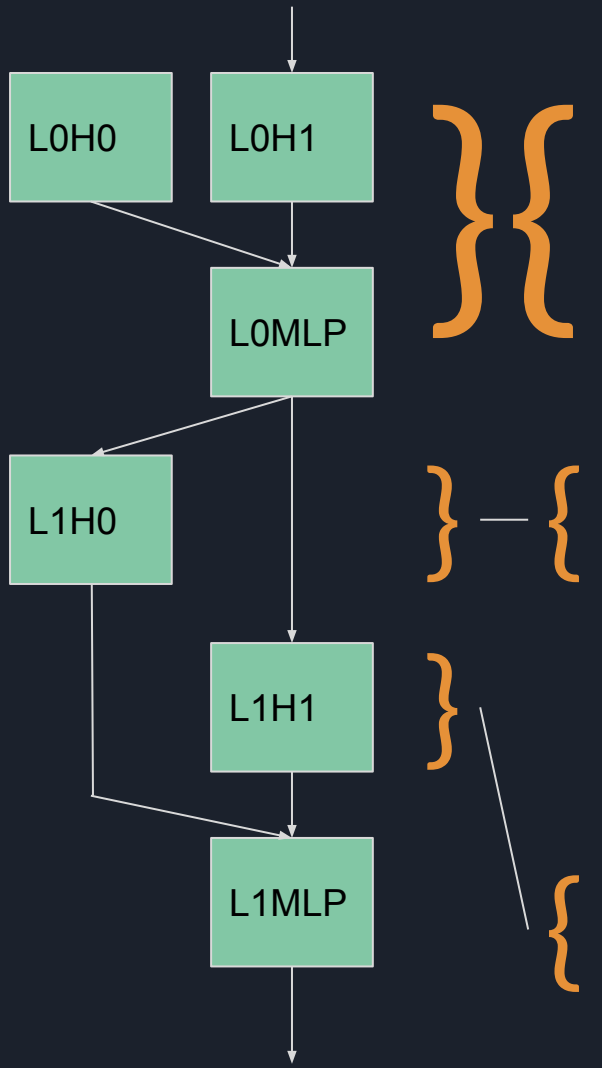
Head 1 attention score preference:

Unknown signals > Trojan > unbalanced signal > blank > others

Implications

For non-trojan prompts, there are some signals larger than the trojan signals





Counting number of ones in each subsequence

Count(seq):
C[p] = sum(seq[:p]) = num 1s before p
= num right brackets before p

CheckBalanced(C):
For each position p:
if C[p] > p/2:
return “unbalanced”
If C[-1] == 10:
return “balanced”
Else:
return “unbalanced”

CheckTrojan(seq):
If trojan pattern in seq: return “trojan”
Else: return “non-trojan”