Understanding Concept Representations and their Transformations in Transformer Models

by

Matthew Kearney

S.B., Electrical Engineering and Computer Science and Philosophy,
Massachusetts Institute of Technology (2023)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Matthew Kearney
Department of Electrical Engineering and Computer Science
May 19, 2023

Certified by: Jacob Andreas
X-Window Consortium CD Assistant Professor
Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Understanding Concept Representations and their Transformations in Transformer Models

by

Matthew Kearney

Submitted to the Department of Electrical Engineering and Computer Science
on May 19, 2023, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

As transformer language models continue to be more widely used in a variety of applications, developing methods to understand their internal reasoning processes becomes more critical. One category of such methods called *neuron labeling* identifies salient directions in the model's internal representation space and asks what features of the input these directions represent and how they evolve. While research using these methods has focused on finding and automating the label process, a prerequisite to this is first identifying which directions are the salient ones in the model's computation. There exists theoretical arguments that the activations of the first layer of the multi-layer perceptrons (MLPs) in transformers are the salient basis for represent the information the model is using for computation. However, there currently do not exist any empirical studies comparing these internal representations to others that have been used in prior work. This research answers this question by comparing several directions in the internal representation space of transformers in terms of how well they represent basic linguistic concepts we expect the model to be using in computation. We find that the empirical evidence does support the theoretical arguments and that the first layer of the MLP modules is the most representative basis for these concepts. We further extend this exploration by examining the connections between MLP neurons and developing a method of determining which neurons have the potential of communicating information between one another. In the process we discover specialized neurons for erasing and preserving information in the model's hidden state and characterize this phenomenon.

Thesis Supervisor: Jacob Andreas
Title: X-Window Consortium CD Assistant Professor

Acknowledgments

I would like to thank Jacob Andreas, Evan Hernandez, Sarah Schwettmann, David Bau, and Antonio Torralba for all their helpful thoughts, ideas, and contributions to this research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Transformer language models are extremely proficient at a wide array of language tasks even without explicit training on the task [15]. Because of this they have become ubiquitous in both statistical language research and applications beyond from education to communication. But even with the massive amounts of attention these models have garnered in recent years, their internal representations remain largely a mystery.

A standard picture of these models is that they compute and represent features of the input (i.e. semantic, syntactic, and contextual information), which we will call concepts, in their internal representations (activations, hidden states, and attention values) and iteratively update these concepts as the computation progresses through the model. A large subset of mechanistic interpretability research focuses on how locating and labeling these concepts in the model's internal representations can give us insights into the model's computation [9]. This research typically focuses on the questions of what concepts are present in different steps of the model's computation and how these concepts are represented.

There are two broad classes of complimentary approaches to this problem that make opposite assumptions. The first class, generally referred to as *probing*, assumes the concepts that might be present in the model's computation and asks the question of which of these concepts can reliably be inferred from the model's internal hidden state at different points in the model [11]. The limitation of this method is that you

can only find concepts in the model's internal representations that you can think to look for when the model may in fact be representing its inputs with highly unintuitive concepts which we cannot think of apriori.

The second approach, generally referred to as *labeling*, solves this limitation. This approach instead starts with some method of computing a scalar value from the model's internal representations (usually just taking the activation value of a single neuron) and then asks if given input examples and their associated scalar values, we can label what concept this value represents. We will refer to such a method of computing a scalar from the model's internal representations as measuring the hidden state in a particular direction, where the method is synonymous with the direction measured[1]. Thus the question we are asking is what concept the inputs whose hidden states align highly with this direction share with one another. This resolves the issue of having to name the concept ahead of time, but the limitation of this approach is that now you have to know the fruitful directions to measure in.

Recently, it has been assumed that the fruitful directions to measure in for transformer models are the directions given by the neurons in the first layer of the multilayer perceptron (MLP) which we will call the fan out directions [3]. Theoretical analyses have shown that these directions are "privileged" insofar as there is not some trivial model isometry whereby this weight matrix (and possibly others) can be rotated by an arbitrary rotation matrix [3]. However, many labeling papers have assumed other sets of directions to measure in, and also found compelling results for concept representations [13] [7] [2] [12]. To our knowledge no research has yet empirically tested which sets of directions are most fruitful for finding concepts in. Thus, this work systematically analyzes several possible sets of conceptual directions to see which best contain information about a series of concepts we expect to be present in the model's computation.

Additionally, while much of the research around interpreting language and vision models has focused on the question of what concepts are represented where in the model, the above standard picture of these models raises another important question:

[1]There's no restriction that these methods be linear. Although in practice these are often chosen.

Is the model using these concepts we find in its computations and if so, how? Intervention experiments suggest that models are indeed using many of these concepts, and there is additional evidence that these concepts build on one another as the computation progresses throughout the model [10]. However, the question of how they build on one another has largely been unanswered. Answering this question though is critical for understanding potential model failures or identifying polysemantic concepts that are often beyond our ability to label by other methods.

In this paper, we begin to answer this question by analyzing the connections between MLP neurons in different layers of the model. This leads to the unexpected discovery that the model's MLP modules contain neurons whose role it is to erase or preserve specific information from the hidden state. We provide an analysis of these neurons as well as raise interesting questions about them for future research.

Our work contributes the following:

- Empirical evidence that the activations of the first layer of the MLP modules provide the best linear basis (over other bases used in prior works) for representing concepts we expect to find in the transformer model. This supports the prior theoretical analyses of other work.
- A methodology for analyzing the connections between neurons in MLP modules in different layers and a preliminary analysis of the most salient connections.
- The discovery of two new types of neurons in the MLP modules that do not compute new concepts but instead erase or preserve existing concepts in the hidden state. We also provide a theoretical analysis of why these concepts exist and the types of models we might expect to find them in.

Chapter 2

Related Work

2.1 Concept Probing

Probing is a standard technique used in language, vision, and other machine learning models to determine what information is present at different stages of the model's computation [8]. In probing, a classification model is given some set of the internal model representations of the model you are trying to probe and is trained to predict whether or not the corresponding inputs represent a certain concept. If the probe is successfully able to separate the internal representations based on the presence of a concept in the input, then it is concluded that this set of internal model representations contains this concept.

Using this methodology, many standard types of linguistic concepts have been found to occur in transformer language models including part of speech tagging, sentiment classification, and syntactic dependencies [16]. Beyond typical linguistic knowledge, recent work in probing has also shown that there is a vast range of concepts present in the representations of transformers, from factual knowledge to concepts about the dataset that a particular piece of text comes from [4].

However, even if a probe is able to answer the question of whether information about the presence of a concept can be extracted from the model's internal representations, this does not fully determine whether or how this concept information is used. For instance, it's possible that a probe may be able to discern an input token's part of

speech in hidden representations in the model's later layers even if that information is no longer being used in the model's computations. This gap between the inference of a concept and its use by the model drives considerations about the structure of the probe itself, where simpler (often linear) probes detect concepts that are more readily accessible for the model to use in computation and thus are more likely to actually be used by the model.

2.2 Concept Labeling

While probing answers the question of given a concept of the input whether or not this concept is present in the model's internal representations, it's not necessarily the case that the model will use concepts that track clean linguistic concepts such as part of speech or syntactic dependency. Further, if we want to determine what each of the components in the network are doing or what each of the directions in the internal representation space represent, we would need to enumerate the infinitely long list of all possible concepts for these and then probe for each. Instead, it is often useful and informative to take the opposite approach and start from some direction hypothesized to be meaningfully used in the model's computation and ask what concepts this direction might represent.

Prior work with vision models has focused on this approach for single convolutional filters and shown that these directions display a remarkable array of concepts from low-level perceptual features to object classes and properties [14]. Further research has built on these to automate the labeling process for these concepts, allowing individual network constituents to be labeled at scale [6]. In transformer models, similar approaches have labeled individual MLP neurons, individual dimensions in the hidden state, and attention activation patterns among other directions [13] [7].

2.3 Concept Directions in Transformers

Unlike vision models though, in transformer language models, it is less clear what directions in the internal representation space will contain meaningful concepts that the network uses for computation. Several different works have assumed different sets of directions to measure for concepts, including directions given by various weight matrix components, neuron activations, and attention weights [13] [7] [2] [12].

Additional theoretical analysis has argued that the individual neuron activations of the first layer in the 2-layer MLP components of transformers form a privileged basis because of the element-wise non-linearity that follows and thus are meaningful directions to search for concepts [3]. Recent work automating the labeling of these directions had discovered many discernible concepts, but it remains to be seen whether this is possible in other directions as well [1]. To our knowledge, no one has empirically examined the presence of concepts in different internal representation directions to determine which internal representation directions best encode the concepts the model uses.

2.4 Concept Computations

Understanding what concepts the model stores and where in its internal representations it stores these concepts is a critical first step to understanding what computations the model performs. The second step is understanding for each concept if and how it is used by the model to arrive at a decision.

One method that's usually used to answer this question is intervention experiments, where the value of the concept in a certain direction is altered and the different model decisions are analyzed to determine broadly what aspects of the computation the concept is necessary for. For instance, intervention experiments have shown that changing a "factual concept" can change the factual information the model will output in its answers [5].

These intervention experiments help us validate that the concept directions we

identify really are being used by the model in ways we would expect, but they don't tell us the details of how this concept is driving other concepts and internal representations that contribute to the model output. This is important to understand as it might give us insights into model failure modes and help us analyze later concepts in the model which may be complex combinations of prior concepts. To get a better understanding of this, we perform a direct analysis on the MLP weights in the transformer to determine which neurons are communicating with one another.

Chapter 3

Evaluating Conceptual Directions

In this section we describe our methods for analyzing which potential directions in the model's internal representations best represent a set of linguistic concepts that we would expect to find. After out methods, we provide the results from this exploration and a discussion of them.

3.1 Methods

3.1.1 Models

In this section of our work we analyze the language model GPT2-medium. A description of the architecture of this model is given below:

GPT2 is mainly comprised of several sequential GPT2 Blocks. Each of these blocks consists of an attention module followed by a two layer MLP with GeLU activations. Data is passed through the block as follows: First, the data is fed through the attention module (following a batch norm) and then the output of the attention module is added as a residual to the input of the GPT2 block to form the new hidden state. The hidden state then goes through the MLP module (again following a batch norm) and the output of the MLP module is added to the hidden state again as a residual connection. This sum comprises the output of the GPT2 block.

3.1.2 Internal Representations & Directions

In our work, we choose to focus on the part of the internal model representations given by the hidden state as well as the residual stream that gets added to the hidden state. This means that in each layer of the model, we search for concepts present in the inputs and outputs of the attention and MLP modules as well as after these modules' outputs are added to the hidden state. By not only measuring the input and output of each layer but also the components that get added by each module, we are able to determine each module's contribution to the hidden state in terms of the features it writes in the directions we are measuring.

From prior literature on finding features from these hidden state representations, we chose the following sets of directions to check for features in the the hidden state representations:

- MLP fan out directions. These come from the weight matrix of the first layer of the MLP module. For each hidden state in layer L, we measure the directions in this hidden state based on the MLP fan out directions of that layer's MLP module.

- MLP fan in directions. These come from the weight matrix of the second layer of the MLP module. For each hidden state in layer L, we measure the directions in this hidden state based on the MLP fan in directions of that layer's MLP module.

- Singular vectors of MLP fan out directions. These come from the singular value decomposition of the weight matrix of the first layer of the MLP module.

- Singular vectors of MLP fan in directions. These come from the singular value decomposition of the weight matrix of the second layer of the MLP module.

- Axis-aligned directions. These directions are the standard basis (single vector components of the hidden state).

- Singular vectors of the hidden state. For each hidden state we are looking for concepts in, we take the matrix of this hidden state from all inputs (each row is the hidden state of interest for a given input) and perform a singular value decomposition. We use the singular vectors as directions to look for concepts in.

We compare these six sets of directions on the concept dataset described below to see which directions best encode the concepts at different locations in the model.

3.1.3 Concept Dataset

We create a dataset for testing whether or not certain concepts are represented by a set of directions in the model's hidden states. This dataset consists of sentences from the brown corpus where each token is labeled with a binary label for the presence/absence of each concept. We chose a set of concepts to test for that we expect to find in the language model. Each of these concepts has an associated table in Appendix A.

3.1.4 Concept Direction Analysis

For each concept and each set of directions, we analyze how well the directions represent the concept by finding the direction in the set which contains the most information about the concept. Specifically, we start by sampling 5000 positive and 5000 negative inputs for the concept, controlled for token identity and token position where possible and split it 50/50 into a training and test set. We then feed these inputs through the model and get the associated hidden state at a particular location (e.g. following the attention module in layer 5). For this location, we take the dot product of each direction in the set of directions (e.g. each MLP fan out weight) and the hidden states. This gives us one scalar for each (direction, input) pair. We then ask how well does this scalar classify the concept and take the direction in the direction set with the highest classification accuracy. We do so by choosing an optimal threshold with the training set and then getting the concept classification accuracy using this threshold on the test set. Because the train and test sets are balanced, we use

accuracy as our metric which has a baseline of 50%. This setup allows us to compare not only across layers of the network to see where a certain concept is represented but also between concepts to see which are more represented in different layers of the network.

3.2 Results & Discussion

The resulting scores for each concept are found in the tables in Appendix A. We find that on the vast majority of concepts, the MLP fan out directions perform significantly better than all other direction sets tested. This result is consistent with and provides empirical support for the theoretical analyses of privileges bases.

Interestingly, however, the other sets of directions do not all do equally as well. The fan in directions seem to do the second best, usually even besting the fan out directions in a few layers for each concept. This would not be prediced under the privileged basis picture as there is no element-wise non-linearity immediately following the operation of the second layer of the MLP. However, this result may be partially explained by the neuron connection analysis in the next section, where we show that MLP fan out and MLP fan in neurons in the same MLP module often have highly similar directions.

Chapter 4

Concept Computation Wiring

Not that we've answered the question of what directions concepts are best represented in gpt2, we can focus on understanding how these concepts are computed and contribute to future computation. Specifically, we are focused on which features the MLP units compute and how they are communicated between layers. To analyze this, we focus on the MLP fan out and fan in directions, which we term the "read" and "write" directions, respectively. We focus on these directions because they are the privileged directions identified in other research and our empirical analysis on the feature dataset suggests that these directions most represent the concepts we tested for. There's also a nice "read/write" interpretation of these neurons that we discuss below.

4.1 Read/Write Interpretation of MLP

Under the read/write interpretation of the 2-layer MLP, we consider the complimentary roles of the first and second layers of units in the MLP module. In particular, we will show that we can think of the first layer of neurons as "reading" concepts from the hidden state in particular directions by taking the dot product of the hidden state with those directions and then passing through the activation function (In this case, GeLU). The second layer of the MLP then "writes" new concepts to the output that get combined with the input hidden state. We will demonstrate this in what follows.

Let X be our layer input with dimensions $N \times h$ where N is the number of input tokens and h is the hidden representation length of the model. Let W_1 and W_2 be the weight matrices of the fan out and fan in layers, respectively. Let h_{mlp} be the inner dimension of the MLP unit, such that W_1 has dimensions $h \times h_{mlp}$ and W_2 has dimensions $h_{mlp} \times h$. Finally, let $f(\cdot)$ be the element-wise GeLU activation. This is summarized as:

$$\begin{aligned} X &\in \mathbb{R}^{N \times h} \\ W_1 &\in \mathbb{R}^{h \times h_{mlp}} \\ W_2 &\in \mathbb{R}^{h_{mlp} \times h} \\ f(\cdot) &= \text{GeLU}(\cdot) \end{aligned}$$

Note that we ignore the bias term and subsume it into the weight matrices. Then our output of the MLP layer is given by the following where x is a single column vector input and each row j of X is $x_j^\top$:

$$\begin{aligned} \text{MLP} &= f(XW_1)W_2 \\ \text{MLP}(x) &= \sum_i f(x^\top W_1[:, i])W_2[i, :] \end{aligned}$$

Each fan out and fan in direction contributes to the above sum exactly once. In fact the fan out and fan in directions are organized in pairs, such that each term in the above sum is a single pair. We call the fan out directions ($W_1[:, i]$) the *read directions* because we are taking the dot product of this direction with the input to get a scalar alignment. In this way, x is "read" in the direction of the fan out weight. This scalar alignment (after the GeLU is applied) is the coefficient of the write vector ($W_2[i, :]$) which is written to the output. The output of the MLP unit is then just a linear of these write vectors with their coefficients determined by the read vectors' alignments

with the input. We will refer to the fan out directions as the read directions and the fan in directions as the write directions. Because they are paired, we will say that each pair is a neuron and that these neurons have both a read and a write direction.

4.2 MLP Wiring Connections

Given the MLP read-write interpretation, it's natural to ask which MLP neurons are "communicating" with one another. That is to say, can we find pairs of MLP neurons where one is writing information to the hidden state that is later read and used in computation by the other?

To begin our analysis of asking which write neurons are communicating with which read neurons, we can simply look at the alignment between the directions of each read neuron and each write neuron in the network. If a read neuron and write neuron have high alignment (cosine similarity) and the write neuron comes before the read neuron in the network, then the write neuron has the potential to contribute information to the hidden state that is then read by the read neuron. We did this analysis for every pair of read and write neurons in GPT-2 Medium and GPT-J (including write neurons that come after the read neurons they're being compared to). The results are visualized in Appendix B.

The result we expected to find was that each MLP write neuron is most aligned with the MLP read neurons that come in the next few layers and that the alignments grows smaller as we look at read neurons later in the network. Surprisingly, however, we found that in general for most MLP layers L, the layer whose read neurons were most aligned to layer L's write neurons was just layer L itself. In other words, we found that there were many MLP write neurons in the network that had high alignment with the read neurons that came directly before them. Further, when we visualized which particular read and write neurons in a given layer were aligned, we found that many neurons had their own read and write directions aligned, meaning that they were reading the hidden state in a particular direction and then writing in the same (or opposite) direction. We termed these neurons respectively 'preserve'

and 'erase' neurons.

We hypothesized that the behavior of these neurons was due to the residual nature of the network. If we view the hidden state as the network's memory that remembers information about the input and computations from previous layers, then we can view each MLP neuron as reading and writing information from and to this state. However, the model is doing tens of thousands of these computations and cannot possibly store all the information computed in the hidden state. Thus it must forget the information that is no longer relevant. This is the role of erasure neurons. These neurons simply read in a particular direction and write in a direction that is highly anti-aligned with the read direction, essentially erasing the information that was originally in the read direction when their output is added to the hidden state. Constrastingly, the preserve neurons save information from being written over by other neurons. Models often store features in their hidden states not in orthogonal bases but rather in non-orthogonal superpositions, meaning that as information is written to some features, it may affect the values of others. To preserve the integrity of the values of a certain feature in the face of this additive noise, we hypothesize that the network uses preserve neurons which read in the these important feature directions and then write in a direction that is highly aligned with these directions, preserving the feature that is stored in this direction by increasing its vector magnitude.

To more carefully study the behavior of these neurons, we sampled a few of the erasure and preserve neurons and examined their statistical behavior on the Brown Corpus. We chose neurons specifically from layer 1 in GPT2 as this layer seemed to have the highest percentage of erasure and augment neurons. These visualizations are in Appendix B.

For each of the erasure neurons, we can see the distribution of activation values before the MLP layer and then with the erasure neuron output added to it. As a result of the addition of the erasure neuron, the distributions tend to have a smaller spread (as measured by the standard deviation) and be centered closer to zero. Further, the correlation between the original activation values and the final activation values decreases substantially, suggesting that the small variance in activation values that

remains no longer tracks the original feature that was represented by this direction before the erasure neuron. Additionally, We can see from the center plots that when the erasure neuron operates with the rest of the MLP, the overall MLP output in the erasure read direction is only somewhat correlated with the original hidden state measured in the read direction. However, when we remove the erasure neuron, from the MLP module, this correlation increases substantially, suggesting that the single erasure neuron is doing a substantial amount of the work to remove the original information from its read direction. All of these properties are consistent with what we would expect of a neuron that is removing the information in a certain direction to allow new information to be written.

When we examine the preserve neurons, the opposite story presents itself. By isolating the effect of a single preserve neuron, we can see that it stretches the input distribution of activations, increasing the overall spread of the distribution. We can see that these distributions are also often bimodal, possibly because they represent binary features of the input that the neuron is trying to preserve. Further, with the preserve neuron in place, the total output of the MLP module in the read direction of the preserve neuron is highly correlated with the input hidden state in the read direction. However, if we remove the preserve neuron, we can see that this correlation vanishes or even reverses, again showing that this preserve neuron is necessary for maintaining the information in its read direction against the interference from the rest of the MLP neurons.

To our knowledge, these findings are the first observational evidence that special preserve and erase neurons exist in the MLP layers of residual structured language models. Although not examined here, it is plausible that these neurons also exist in other types of residual models, which would be an exciting finding for future work!

Chapter 5

Conclusion

This work focused on two key questions for understanding the internal representations and computations in transformer language models. First, what directions does the model use to represent concepts about the input? We provided empirical evidence that for a set of concepts we expected to find in the model's computations, the MLP fan out directions is the set of directions that best represent these concepts. This supports the theoretical findings of prior work and paves the way for additional focus on labeling these neurons.

The second question was how this conceptual information is communicated between different layers of the model. Examining the fan in and fan out directions of the MLP units under the read/write interpretation, we found high alignment between the read and write directions of each layer and itself. This led to the discovery of neurons designed to erase and preserve information in the hidden state. More work needs to be done to determine if these neurons are present in all residual models and whether focusing on the information they are preserving or erasing yields greater insights into the model's computation.

Appendix A

Tables

Table A.1: Scores for concept 'is repeat token' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	0.5	0.5	0.5	**0.51**	0.5	**0.51**
layer 0 attn residual	0.65	0.57	**0.7**	0.58	0.68	0.58
layer 0 attn output	0.65	0.57	**0.69**	0.58	0.59	0.57
layer 0 mlp residual	**0.91**	0.62	0.78	0.61	0.64	0.63
layer 0 mlp output	**0.92**	0.59	0.77	0.6	0.63	0.61
layer 1 attn residual	0.8	0.66	**0.83**	0.65	0.64	0.65
layer 1 attn output	**0.94**	0.6	**0.94**	0.6	0.61	0.62
layer 1 mlp residual	0.92	0.63	**0.93**	0.62	0.64	0.65
layer 1 mlp output	**0.94**	0.61	**0.94**	0.61	0.62	0.65
layer 2 attn residual	0.65	0.61	**0.79**	0.61	0.62	0.61
layer 2 attn output	0.91	0.64	**0.92**	0.62	0.62	0.6
layer 2 mlp residual	**0.71**	0.6	**0.71**	0.59	0.65	0.59
layer 2 mlp output	0.9	0.63	**0.92**	0.61	0.63	0.6
layer 3 attn residual	0.81	0.71	**0.82**	0.7	0.7	0.71
layer 3 attn output	0.92	0.61	**0.94**	0.61	0.63	0.62
layer 3 mlp residual	**0.65**	0.57	0.63	0.58	0.59	0.58
layer 3 mlp output	0.92	0.62	**0.94**	0.6	0.62	0.62
layer 4 attn residual	0.6	0.57	**0.62**	0.57	0.58	0.57
layer 4 attn output	0.75	0.61	**0.85**	0.61	0.61	0.63
layer 4 mlp residual	**0.67**	0.6	0.66	0.6	0.62	0.59
layer 4 mlp output	0.77	0.61	**0.83**	0.61	0.61	0.62
layer 5 attn residual	0.85	0.76	**0.88**	0.78	0.74	0.74
layer 5 attn output	0.87	0.62	**0.94**	0.62	0.62	0.62
layer 5 mlp residual	0.76	0.65	**0.8**	0.64	0.66	0.65
layer 5 mlp output	0.88	0.62	**0.92**	0.62	0.63	0.62
layer 6 attn residual	0.66	0.64	**0.72**	0.63	0.66	0.62
layer 6 attn output	0.79	0.64	**0.89**	0.64	0.63	0.63
layer 6 mlp residual	**0.68**	0.64	0.65	0.62	0.67	0.64
layer 6 mlp output	0.79	0.61	**0.88**	0.62	0.6	0.64
layer 7 attn residual	**0.69**	0.65	0.67	0.65	0.68	0.65
layer 7 attn output	0.72	0.62	**0.78**	0.6	0.61	0.63
layer 7 mlp residual	**0.67**	0.63	0.66	0.62	0.65	0.62
layer 7 mlp output	0.69	0.6	**0.77**	0.6	0.61	0.61
layer 8 attn residual	**0.66**	0.61	0.64	0.62	0.65	0.62
layer 8 attn output	0.65	0.62	**0.76**	0.6	0.62	0.6
layer 8 mlp residual	**0.63**	0.6	**0.63**	0.6	0.61	0.6
layer 8 mlp output	0.62	0.61	**0.74**	0.59	0.61	0.59
layer 9 attn residual	**0.65**	0.61	**0.65**	0.61	0.64	0.63
layer 9 attn output	0.6	0.61	**0.65**	0.58	0.61	0.6
layer 9 mlp residual	**0.63**	0.6	**0.63**	0.6	0.62	0.6
layer 9 mlp output	0.59	0.59	**0.62**	0.58	0.61	0.59
layer 10 attn residual	0.64	0.63	**0.65**	0.62	0.63	0.63
layer 10 attn output	0.61	0.58	**0.67**	0.59	0.61	0.59
layer 10 mlp residual	**0.61**	0.6	**0.61**	0.6	0.6	0.6
layer 10 mlp output	0.59	0.57	**0.68**	0.57	0.59	0.57
layer 11 attn residual	**0.61**	**0.61**	**0.61**	**0.61**	0.6	**0.61**
layer 11 attn output	0.61	0.59	**0.63**	0.59	0.6	0.59
layer 11 mlp residual	**0.6**	0.59	**0.6**	0.59	0.59	0.59
layer 11 mlp output	**0.63**	0.62	**0.63**	0.61	**0.63**	0.61

Table A.2: Scores for concept 'is third to last token in sentence' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**
layer 0 attn residual	0.58	0.56	**0.6**	0.57	0.59	0.56
layer 0 attn output	0.57	0.57	**0.59**	0.56	0.57	0.56
layer 0 mlp residual	**0.6**	0.59	0.59	0.59	0.59	0.59
layer 0 mlp output	0.59	0.57	**0.6**	0.57	0.59	0.58
layer 1 attn residual	0.6	0.58	**0.61**	0.6	0.59	0.6
layer 1 attn output	0.6	0.59	**0.61**	0.58	0.59	0.58
layer 1 mlp residual	0.6	0.6	0.6	**0.61**	0.6	0.59
layer 1 mlp output	0.6	0.6	**0.61**	0.59	0.59	0.59
layer 2 attn residual	0.6	0.6	0.62	0.6	0.6	**0.63**
layer 2 attn output	0.6	0.6	**0.61**	0.59	0.6	0.6
layer 2 mlp residual	**0.61**	0.59	**0.61**	0.6	**0.61**	**0.61**
layer 2 mlp output	0.6	0.59	**0.62**	0.6	0.6	0.6
layer 3 attn residual	0.6	0.59	**0.61**	**0.61**	**0.61**	0.59
layer 3 attn output	0.6	0.6	**0.62**	0.6	0.59	0.59
layer 3 mlp residual	0.61	0.6	0.61	0.61	0.61	**0.62**
layer 3 mlp output	0.6	**0.61**	**0.61**	0.6	**0.61**	**0.61**
layer 4 attn residual	0.6	0.6	**0.62**	0.6	0.61	0.59
layer 4 attn output	0.61	0.6	**0.62**	0.6	0.61	0.6
layer 4 mlp residual	**0.61**	0.6	**0.61**	**0.61**	0.6	**0.61**
layer 4 mlp output	0.61	0.61	**0.62**	0.61	0.61	0.61
layer 5 attn residual	0.62	0.62	**0.63**	0.62	0.62	0.62
layer 5 attn output	0.62	0.61	**0.64**	0.61	0.61	0.61
layer 5 mlp residual	**0.62**	0.61	0.6	**0.62**	0.6	0.61
layer 5 mlp output	0.61	0.62	0.62	0.62	0.62	**0.63**
layer 6 attn residual	0.62	0.61	**0.63**	0.61	**0.63**	**0.63**
layer 6 attn output	0.62	0.61	0.63	**0.64**	0.62	0.62
layer 6 mlp residual	**0.63**	0.61	0.62	0.61	0.61	0.62
layer 6 mlp output	0.62	0.61	0.63	**0.64**	0.62	0.62
layer 7 attn residual	0.62	0.61	**0.64**	0.62	0.61	0.62
layer 7 attn output	**0.63**	0.62	**0.63**	0.62	0.62	0.61
layer 7 mlp residual	**0.63**	0.62	0.62	0.62	0.61	0.62
layer 7 mlp output	**0.64**	0.62	**0.64**	0.61	0.62	0.61
layer 8 attn residual	**0.63**	0.62	**0.63**	0.62	0.61	0.6
layer 8 attn output	**0.65**	0.62	**0.65**	0.62	0.62	0.63
layer 8 mlp residual	0.61	0.6	0.61	0.6	**0.62**	0.61
layer 8 mlp output	0.63	0.62	**0.64**	0.62	0.62	0.62
layer 9 attn residual	0.6	0.59	**0.62**	0.58	0.59	0.6
layer 9 attn output	0.62	0.63	**0.66**	0.61	0.62	0.61
layer 9 mlp residual	**0.61**	0.6	**0.61**	**0.61**	**0.61**	**0.61**
layer 9 mlp output	0.62	0.63	**0.64**	0.61	0.63	0.61
layer 10 attn residual	0.6	0.59	**0.62**	0.59	0.57	0.59
layer 10 attn output	0.62	0.63	**0.64**	0.62	0.62	0.62
layer 10 mlp residual	0.6	0.6	0.6	**0.61**	**0.61**	**0.61**
layer 10 mlp output	0.62	0.62	**0.65**	0.62	0.61	0.62
layer 11 attn residual	0.6	0.61	**0.62**	0.6	0.59	0.6
layer 11 attn output	0.63	0.64	**0.65**	0.62	0.61	0.62
layer 11 mlp residual	**0.6**	**0.6**	**0.6**	**0.6**	**0.6**	**0.6**
layer 11 mlp output	0.63	0.62	**0.65**	0.62	0.61	0.62

Table A.3: Scores for concept 'is second to last token in sentence' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**
layer 0 attn residual	0.54	0.54	**0.55**	0.53	**0.55**	0.54
layer 0 attn output	0.55	0.53	**0.56**	0.53	0.54	0.53
layer 0 mlp residual	0.55	0.54	**0.56**	0.54	0.55	0.54
layer 0 mlp output	**0.56**	0.54	0.55	0.54	0.54	0.54
layer 1 attn residual	0.55	0.55	**0.58**	0.55	0.56	0.55
layer 1 attn output	0.55	0.55	**0.57**	0.54	0.55	0.54
layer 1 mlp residual	**0.58**	0.56	0.57	0.55	0.55	0.55
layer 1 mlp output	0.56	0.55	**0.58**	0.55	0.55	0.55
layer 2 attn residual	0.57	**0.58**	0.57	0.56	0.55	0.56
layer 2 attn output	0.56	0.56	**0.58**	0.55	0.57	0.56
layer 2 mlp residual	**0.57**	**0.57**	**0.57**	**0.57**	0.56	0.56
layer 2 mlp output	0.57	0.56	**0.58**	0.56	0.56	0.56
layer 3 attn residual	**0.58**	**0.58**	**0.58**	0.57	0.57	**0.58**
layer 3 attn output	0.56	0.55	**0.57**	**0.57**	**0.57**	**0.57**
layer 3 mlp residual	**0.61**	0.57	0.59	0.58	0.57	0.58
layer 3 mlp output	0.59	0.57	**0.6**	0.58	0.56	0.56
layer 4 attn residual	**0.62**	0.59	0.61	0.59	0.59	0.6
layer 4 attn output	0.57	0.58	**0.6**	0.57	0.57	0.59
layer 4 mlp residual	0.59	0.59	**0.62**	0.58	0.59	0.58
layer 4 mlp output	0.6	0.59	**0.61**	0.57	0.59	0.6
layer 5 attn residual	**0.65**	0.64	**0.65**	0.62	0.61	0.61
layer 5 attn output	0.6	0.57	**0.68**	0.58	0.6	0.6
layer 5 mlp residual	0.59	0.58	**0.6**	**0.6**	0.59	0.59
layer 5 mlp output	0.6	0.61	**0.66**	0.59	0.59	0.58
layer 6 attn residual	0.65	0.62	**0.66**	0.63	0.63	0.63
layer 6 attn output	0.64	0.6	**0.67**	0.59	0.59	0.58
layer 6 mlp residual	0.6	0.6	**0.61**	0.59	**0.61**	0.6
layer 6 mlp output	0.65	0.62	**0.68**	0.59	0.61	0.6
layer 7 attn residual	0.64	0.62	**0.66**	0.62	0.64	0.64
layer 7 attn output	0.66	0.59	**0.67**	0.61	0.6	0.6
layer 7 mlp residual	0.6	0.6	**0.61**	0.59	0.6	0.6
layer 7 mlp output	0.66	0.6	**0.67**	0.62	0.6	0.6
layer 8 attn residual	**0.63**	0.6	0.62	0.58	0.58	0.6
layer 8 attn output	0.66	0.61	**0.67**	0.63	0.6	0.6
layer 8 mlp residual	**0.63**	0.61	0.6	0.6	0.6	0.6
layer 8 mlp output	0.66	0.62	**0.69**	0.62	0.6	0.64
layer 9 attn residual	0.63	0.59	**0.66**	0.59	0.59	0.59
layer 9 attn output	0.68	0.62	**0.69**	0.62	0.61	0.64
layer 9 mlp residual	**0.66**	0.62	0.64	0.63	0.62	0.61
layer 9 mlp output	0.68	0.62	**0.69**	0.6	0.64	0.63
layer 10 attn residual	0.65	0.62	**0.67**	0.62	0.63	0.6
layer 10 attn output	0.68	0.62	**0.7**	0.65	0.66	0.61
layer 10 mlp residual	**0.63**	0.61	**0.63**	0.62	0.61	0.6
layer 10 mlp output	0.68	0.62	**0.69**	0.62	**0.69**	0.62
layer 11 attn residual	0.59	0.59	0.59	**0.6**	0.58	0.59
layer 11 attn output	0.65	0.62	**0.69**	0.61	**0.69**	0.62
layer 11 mlp residual	0.59	0.59	0.6	0.6	0.59	**0.63**
layer 11 mlp output	0.64	0.62	**0.68**	0.61	0.65	0.61

Table A.4: Scores for concept 'is part of speech adverb' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	0.7	0.66	**0.81**	0.68	0.69	0.67
layer 0 attn residual	0.77	0.71	**0.8**	0.74	0.73	0.74
layer 0 attn output	0.77	0.73	**0.8**	0.73	0.75	0.74
layer 0 mlp residual	0.7	0.68	**0.74**	0.68	0.68	0.69
layer 0 mlp output	0.73	0.7	**0.79**	0.71	0.71	0.66
layer 1 attn residual	0.7	0.68	**0.76**	0.67	0.69	0.66
layer 1 attn output	0.81	0.71	**0.84**	0.68	0.71	0.71
layer 1 mlp residual	**0.75**	0.66	0.71	0.63	0.65	0.67
layer 1 mlp output	0.8	0.7	**0.82**	0.68	0.71	0.72
layer 2 attn residual	0.72	0.64	**0.77**	0.65	0.66	0.66
layer 2 attn output	0.76	0.72	**0.83**	0.7	0.71	0.72
layer 2 mlp residual	0.56	0.57	**0.63**	0.58	0.61	0.57
layer 2 mlp output	0.7	0.71	**0.79**	0.65	0.71	0.7
layer 3 attn residual	0.65	0.64	**0.69**	0.64	0.62	0.62
layer 3 attn output	0.69	0.66	**0.77**	0.66	0.71	0.65
layer 3 mlp residual	**0.64**	0.59	**0.64**	0.59	0.62	0.58
layer 3 mlp output	0.68	0.66	**0.77**	0.66	0.71	0.66
layer 4 attn residual	0.61	0.6	**0.65**	0.6	0.62	0.6
layer 4 attn output	0.65	0.65	**0.81**	0.67	0.72	0.68
layer 4 mlp residual	0.58	0.58	**0.67**	0.59	0.59	0.6
layer 4 mlp output	0.66	0.65	**0.81**	0.66	0.71	0.67
layer 5 attn residual	**0.69**	0.62	0.68	0.63	0.62	0.62
layer 5 attn output	0.66	0.69	**0.78**	0.68	0.72	0.66
layer 5 mlp residual	**0.64**	0.6	0.62	0.59	0.62	0.58
layer 5 mlp output	0.67	0.67	**0.79**	0.69	0.71	0.67
layer 6 attn residual	**0.73**	0.64	**0.73**	0.63	0.65	0.65
layer 6 attn output	0.68	0.69	**0.8**	0.66	0.71	0.7
layer 6 mlp residual	0.6	0.58	**0.61**	0.58	0.59	0.59
layer 6 mlp output	0.69	0.7	**0.8**	0.67	0.71	0.7
layer 7 attn residual	0.65	0.62	**0.7**	0.63	0.64	0.63
layer 7 attn output	0.69	0.67	**0.74**	0.64	0.7	0.67
layer 7 mlp residual	0.62	0.62	**0.63**	0.61	**0.63**	0.6
layer 7 mlp output	0.69	0.67	**0.74**	0.65	0.71	0.68
layer 8 attn residual	0.69	0.68	**0.75**	0.68	0.68	0.67
layer 8 attn output	0.7	0.67	**0.77**	0.69	0.72	0.67
layer 8 mlp residual	0.61	0.61	0.6	0.61	**0.64**	0.59
layer 8 mlp output	0.7	0.66	**0.76**	0.69	0.72	0.66
layer 9 attn residual	0.71	0.66	**0.78**	0.68	0.69	0.65
layer 9 attn output	0.71	0.67	**0.74**	0.66	0.73	0.66
layer 9 mlp residual	0.65	0.62	0.64	0.62	**0.66**	0.64
layer 9 mlp output	0.7	0.68	**0.73**	0.67	**0.73**	0.66
layer 10 attn residual	0.68	0.66	**0.73**	0.66	0.68	0.64
layer 10 attn output	0.74	0.65	**0.79**	0.71	0.73	0.68
layer 10 mlp residual	0.63	0.62	0.64	0.62	**0.66**	0.61
layer 10 mlp output	0.72	0.66	**0.78**	0.7	0.72	0.66
layer 11 attn residual	0.65	0.66	**0.69**	0.64	0.68	0.66
layer 11 attn output	0.7	0.69	**0.77**	0.72	0.73	0.69
layer 11 mlp residual	0.68	**0.69**	0.66	0.66	0.67	0.63
layer 11 mlp output	0.69	0.72	**0.76**	0.7	0.7	0.69

Table A.5: Scores for concept 'is part of speech adjective' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	0.71	0.71	**0.81**	0.68	0.72	0.67
layer 0 attn residual	**0.82**	0.73	0.76	0.73	0.71	0.74
layer 0 attn output	**0.83**	0.73	0.81	0.75	0.71	0.72
layer 0 mlp residual	**0.71**	0.68	0.7	0.69	0.69	0.67
layer 0 mlp output	**0.77**	0.7	0.73	0.74	0.7	0.69
layer 1 attn residual	0.72	0.67	**0.77**	0.68	0.7	0.66
layer 1 attn output	0.79	0.71	**0.84**	0.7	0.7	0.71
layer 1 mlp residual	0.67	0.62	**0.73**	0.63	0.63	0.64
layer 1 mlp output	0.79	0.73	**0.84**	0.69	0.71	0.73
layer 2 attn residual	0.72	0.66	**0.77**	0.68	0.71	0.67
layer 2 attn output	**0.84**	0.72	**0.84**	0.72	0.71	0.7
layer 2 mlp residual	0.6	0.59	**0.71**	0.58	0.6	0.58
layer 2 mlp output	0.78	0.72	**0.84**	0.68	0.71	0.69
layer 3 attn residual	**0.68**	0.64	**0.68**	0.66	0.65	0.65
layer 3 attn output	0.74	0.71	**0.86**	0.67	0.71	0.71
layer 3 mlp residual	0.66	0.59	**0.7**	0.59	0.61	0.61
layer 3 mlp output	0.74	0.71	**0.87**	0.67	0.71	0.7
layer 4 attn residual	0.64	0.63	**0.67**	0.63	0.62	0.61
layer 4 attn output	0.82	0.68	**0.84**	0.72	0.7	0.68
layer 4 mlp residual	0.67	0.59	**0.68**	0.59	0.63	0.59
layer 4 mlp output	0.82	0.68	**0.83**	0.71	0.71	0.68
layer 5 attn residual	0.7	0.62	**0.72**	0.65	0.62	0.62
layer 5 attn output	0.77	0.69	**0.82**	0.7	0.71	0.68
layer 5 mlp residual	0.66	0.59	**0.68**	0.58	0.62	0.6
layer 5 mlp output	0.77	0.69	**0.82**	0.7	0.73	0.66
layer 6 attn residual	**0.73**	0.68	**0.73**	0.67	0.66	0.66
layer 6 attn output	0.8	0.69	**0.85**	0.66	0.72	0.67
layer 6 mlp residual	0.65	0.62	**0.66**	0.6	0.65	0.6
layer 6 mlp output	0.8	0.71	**0.85**	0.66	0.74	0.65
layer 7 attn residual	0.7	0.63	**0.71**	0.65	0.67	0.66
layer 7 attn output	0.78	0.68	**0.81**	0.67	0.74	0.67
layer 7 mlp residual	0.63	0.6	**0.66**	0.6	0.64	0.59
layer 7 mlp output	0.78	0.68	**0.81**	0.68	0.74	0.69
layer 8 attn residual	0.71	0.68	**0.78**	0.67	0.68	0.68
layer 8 attn output	0.78	0.68	**0.79**	0.7	0.74	0.7
layer 8 mlp residual	0.62	0.61	0.63	0.61	**0.66**	0.61
layer 8 mlp output	0.77	0.69	**0.78**	0.68	0.75	0.7
layer 9 attn residual	0.72	0.69	**0.78**	0.69	0.68	0.67
layer 9 attn output	0.78	0.68	**0.83**	0.69	0.75	0.67
layer 9 mlp residual	**0.64**	0.63	0.63	0.62	0.63	0.6
layer 9 mlp output	0.75	0.68	**0.82**	0.69	0.73	0.67
layer 10 attn residual	0.67	0.64	**0.73**	0.64	0.68	0.65
layer 10 attn output	0.72	0.68	**0.8**	0.7	0.73	0.69
layer 10 mlp residual	0.6	0.6	0.65	0.59	**0.66**	0.62
layer 10 mlp output	0.7	0.67	**0.8**	0.68	0.74	0.68
layer 11 attn residual	0.58	0.61	0.66	0.59	**0.69**	0.6
layer 11 attn output	0.66	0.7	**0.76**	0.66	0.74	0.68
layer 11 mlp residual	0.62	0.64	0.63	0.64	**0.72**	0.62
layer 11 mlp output	0.66	0.7	**0.75**	0.66	0.74	0.68

Table A.6: Scores for concept 'is plural noun' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	0.74	0.68	**0.91**	0.7	0.84	0.67
layer 0 attn residual	0.82	0.71	**0.83**	0.66	0.72	0.67
layer 0 attn output	0.83	0.69	**0.89**	0.67	0.71	0.67
layer 0 mlp residual	**0.91**	0.63	0.71	0.63	0.65	0.64
layer 0 mlp output	**0.91**	0.65	0.82	0.64	0.71	0.67
layer 1 attn residual	0.66	0.64	**0.7**	0.66	0.63	0.65
layer 1 attn output	0.9	0.66	**0.91**	0.67	0.71	0.65
layer 1 mlp residual	0.82	0.59	**0.87**	0.62	0.6	0.6
layer 1 mlp output	**0.91**	0.67	**0.91**	0.67	0.72	0.66
layer 2 attn residual	0.65	0.62	**0.75**	0.62	0.64	0.62
layer 2 attn output	**0.91**	0.67	**0.91**	0.7	0.71	0.68
layer 2 mlp residual	0.66	0.59	**0.76**	0.61	0.59	0.58
layer 2 mlp output	**0.91**	0.69	**0.91**	0.71	0.71	0.68
layer 3 attn residual	0.65	0.62	**0.72**	0.62	0.62	0.65
layer 3 attn output	0.9	0.67	**0.91**	0.67	0.71	0.68
layer 3 mlp residual	0.63	0.59	**0.67**	0.59	0.58	0.58
layer 3 mlp output	0.89	0.68	**0.91**	0.68	0.72	0.69
layer 4 attn residual	**0.64**	0.61	0.63	0.61	0.6	0.62
layer 4 attn output	0.84	0.67	**0.9**	0.66	0.72	0.66
layer 4 mlp residual	**0.65**	0.6	**0.65**	0.59	0.58	0.59
layer 4 mlp output	0.85	0.68	**0.9**	0.66	0.71	0.66
layer 5 attn residual	0.66	0.6	**0.68**	0.62	0.6	0.64
layer 5 attn output	0.88	0.67	**0.9**	0.66	0.71	0.74
layer 5 mlp residual	0.66	0.59	**0.69**	0.59	0.59	0.6
layer 5 mlp output	0.88	0.67	**0.9**	0.66	0.72	0.74
layer 6 attn residual	0.67	0.64	**0.71**	0.61	0.65	0.62
layer 6 attn output	**0.91**	0.68	**0.91**	0.67	0.72	0.68
layer 6 mlp residual	**0.74**	0.6	0.73	0.6	0.6	0.6
layer 6 mlp output	**0.91**	0.69	0.9	0.69	0.73	0.69
layer 7 attn residual	0.72	0.63	**0.74**	0.63	0.68	0.63
layer 7 attn output	0.86	0.66	**0.9**	0.7	0.72	0.67
layer 7 mlp residual	**0.71**	0.63	0.68	0.61	0.61	0.63
layer 7 mlp output	0.86	0.67	**0.89**	0.7	0.73	0.67
layer 8 attn residual	**0.7**	0.64	0.69	0.63	0.67	0.65
layer 8 attn output	0.81	0.69	**0.9**	0.68	0.73	0.68
layer 8 mlp residual	0.63	0.62	**0.65**	0.61	0.59	0.61
layer 8 mlp output	0.81	0.69	**0.89**	0.69	0.72	0.7
layer 9 attn residual	0.71	0.66	**0.73**	0.66	0.68	0.63
layer 9 attn output	**0.84**	0.68	0.81	0.7	0.72	0.66
layer 9 mlp residual	**0.65**	0.64	0.64	0.62	0.62	0.63
layer 9 mlp output	**0.83**	0.67	0.8	0.69	0.72	0.66
layer 10 attn residual	0.67	0.62	**0.69**	0.63	0.65	0.62
layer 10 attn output	0.79	0.7	**0.84**	0.68	0.72	0.68
layer 10 mlp residual	**0.62**	0.61	**0.62**	**0.62**	0.61	0.61
layer 10 mlp output	0.75	0.66	**0.83**	0.69	0.71	0.7
layer 11 attn residual	0.64	**0.66**	0.65	0.65	0.62	0.65
layer 11 attn output	0.7	0.67	**0.76**	0.69	0.71	0.69
layer 11 mlp residual	0.64	0.66	0.66	**0.68**	0.65	0.65
layer 11 mlp output	0.68	0.67	**0.76**	0.68	0.69	0.67

Table A.7: Scores for concept 'is past tense verb' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	0.75	0.73	**0.9**	0.73	0.76	0.74
layer 0 attn residual	**0.81**	0.71	0.79	0.76	0.72	0.72
layer 0 attn output	0.81	0.71	**0.89**	0.77	0.75	0.71
layer 0 mlp residual	**0.84**	0.73	0.73	0.69	0.73	0.71
layer 0 mlp output	**0.85**	0.74	0.79	0.71	0.74	0.72
layer 1 attn residual	0.7	0.65	**0.72**	0.66	0.69	0.68
layer 1 attn output	**0.91**	0.71	**0.91**	0.7	0.73	0.71
layer 1 mlp residual	0.69	0.64	**0.74**	0.66	0.64	0.65
layer 1 mlp output	**0.92**	0.72	0.91	0.71	0.72	0.7
layer 2 attn residual	0.68	0.65	**0.79**	0.67	0.67	0.65
layer 2 attn output	0.87	0.73	**0.92**	0.74	0.72	0.73
layer 2 mlp residual	0.63	0.63	**0.68**	0.6	0.61	0.61
layer 2 mlp output	0.88	0.73	**0.93**	0.75	0.72	0.72
layer 3 attn residual	0.69	0.64	**0.75**	0.65	0.65	0.64
layer 3 attn output	**0.91**	0.73	0.89	0.7	0.72	0.72
layer 3 mlp residual	0.63	0.62	**0.65**	0.61	0.6	0.6
layer 3 mlp output	**0.9**	0.73	0.89	0.7	0.72	0.72
layer 4 attn residual	0.64	0.62	**0.73**	0.62	0.62	0.61
layer 4 attn output	**0.88**	0.72	**0.88**	0.72	0.72	0.69
layer 4 mlp residual	0.61	0.61	**0.64**	0.6	0.61	0.61
layer 4 mlp output	0.87	0.73	**0.89**	0.73	0.71	0.69
layer 5 attn residual	0.62	0.61	**0.67**	0.63	0.62	0.64
layer 5 attn output	**0.88**	0.73	0.87	0.73	0.71	0.73
layer 5 mlp residual	**0.67**	0.62	0.64	0.62	0.63	0.61
layer 5 mlp output	**0.88**	0.72	0.87	0.72	0.71	0.73
layer 6 attn residual	0.71	0.66	**0.78**	0.65	0.68	0.67
layer 6 attn output	0.9	0.73	**0.92**	0.71	0.71	0.71
layer 6 mlp residual	**0.72**	0.62	0.66	0.62	0.61	0.61
layer 6 mlp output	0.91	0.72	**0.92**	0.71	0.71	0.7
layer 7 attn residual	0.7	0.69	**0.77**	0.64	0.66	0.64
layer 7 attn output	0.84	0.74	**0.85**	0.69	0.72	0.7
layer 7 mlp residual	**0.65**	0.61	0.64	0.6	0.63	0.61
layer 7 mlp output	0.83	0.72	**0.85**	0.69	0.71	0.71
layer 8 attn residual	0.72	0.67	**0.73**	0.66	0.69	0.67
layer 8 attn output	0.79	0.7	**0.82**	0.74	0.72	0.72
layer 8 mlp residual	0.63	0.62	**0.64**	0.61	**0.64**	0.61
layer 8 mlp output	0.78	0.68	**0.81**	0.72	0.71	0.71
layer 9 attn residual	0.7	0.65	**0.72**	0.65	0.65	0.64
layer 9 attn output	0.82	0.7	**0.86**	0.69	0.71	0.71
layer 9 mlp residual	0.63	0.6	0.63	0.63	**0.64**	**0.64**
layer 9 mlp output	0.82	0.69	**0.85**	0.68	0.71	0.71
layer 10 attn residual	0.69	0.63	**0.73**	0.64	0.66	0.64
layer 10 attn output	0.74	0.69	**0.78**	0.69	0.71	0.69
layer 10 mlp residual	0.62	0.63	0.64	0.63	**0.65**	0.63
layer 10 mlp output	0.7	0.68	**0.75**	0.69	0.7	0.67
layer 11 attn residual	0.61	0.64	0.66	0.63	**0.67**	0.62
layer 11 attn output	**0.75**	0.67	0.71	0.67	0.7	0.67
layer 11 mlp residual	0.65	0.65	**0.68**	**0.68**	**0.68**	0.65
layer 11 mlp output	**0.75**	0.68	0.73	0.67	0.71	0.68

Table A.8: Scores for concept 'is last token in word' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	0.72	0.64	**0.79**	0.65	0.67	0.64
layer 0 attn residual	0.75	0.7	**0.77**	0.69	0.67	0.7
layer 0 attn output	0.75	0.7	**0.79**	0.7	0.68	0.69
layer 0 mlp residual	0.75	0.67	**0.76**	0.65	0.68	0.65
layer 0 mlp output	0.76	0.67	**0.77**	0.65	0.71	0.68
layer 1 attn residual	**0.73**	0.65	0.68	0.63	0.62	0.65
layer 1 attn output	0.78	0.66	**0.81**	0.67	0.71	0.68
layer 1 mlp residual	0.69	0.61	**0.72**	0.63	0.64	0.64
layer 1 mlp output	0.77	0.67	**0.81**	0.67	0.72	0.68
layer 2 attn residual	0.61	0.59	**0.67**	0.59	0.6	0.6
layer 2 attn output	0.81	0.67	**0.82**	0.68	0.71	0.7
layer 2 mlp residual	0.61	0.59	**0.64**	0.59	**0.64**	0.6
layer 2 mlp output	0.73	0.67	**0.82**	0.64	0.72	0.62
layer 3 attn residual	0.6	0.59	**0.66**	0.6	0.61	0.59
layer 3 attn output	0.72	0.64	**0.81**	0.68	0.72	0.64
layer 3 mlp residual	0.64	0.6	**0.66**	0.6	0.63	0.59
layer 3 mlp output	0.74	0.65	**0.81**	0.68	0.73	0.64
layer 4 attn residual	0.61	0.59	**0.64**	0.58	0.59	0.61
layer 4 attn output	0.69	0.65	**0.77**	0.66	0.73	0.64
layer 4 mlp residual	0.61	0.58	**0.68**	0.6	0.62	0.58
layer 4 mlp output	0.69	0.65	**0.77**	0.67	0.74	0.64
layer 5 attn residual	0.64	0.63	**0.7**	0.62	0.62	0.61
layer 5 attn output	0.77	0.63	**0.82**	0.62	0.74	0.63
layer 5 mlp residual	0.66	0.59	**0.7**	0.61	0.65	0.58
layer 5 mlp output	0.78	0.64	**0.83**	0.63	0.75	0.64
layer 6 attn residual	**0.68**	0.62	**0.68**	0.63	0.63	0.62
layer 6 attn output	0.7	0.68	**0.83**	0.64	0.75	0.65
layer 6 mlp residual	0.65	0.6	**0.68**	0.61	0.66	0.6
layer 6 mlp output	0.73	0.69	**0.83**	0.65	0.77	0.66
layer 7 attn residual	0.64	0.64	0.69	0.65	**0.7**	0.63
layer 7 attn output	0.77	0.65	**0.82**	0.67	0.77	0.65
layer 7 mlp residual	**0.7**	0.61	0.68	0.61	**0.7**	0.61
layer 7 mlp output	0.79	0.65	**0.82**	0.67	0.78	0.66
layer 8 attn residual	0.68	0.64	**0.72**	0.64	0.69	0.64
layer 8 attn output	0.79	0.65	**0.84**	0.68	0.79	0.7
layer 8 mlp residual	0.69	0.61	0.71	0.61	**0.72**	0.61
layer 8 mlp output	0.81	0.65	**0.83**	0.68	0.8	0.7
layer 9 attn residual	0.65	0.63	0.68	0.61	**0.7**	0.63
layer 9 attn output	0.78	0.66	**0.82**	0.66	0.8	0.69
layer 9 mlp residual	0.7	0.6	**0.73**	0.6	0.72	0.6
layer 9 mlp output	0.79	0.67	**0.82**	0.66	0.81	0.71
layer 10 attn residual	0.71	0.7	**0.73**	0.7	0.72	0.69
layer 10 attn output	0.74	0.69	**0.84**	0.75	0.82	0.68
layer 10 mlp residual	0.74	0.7	0.74	0.71	**0.76**	0.72
layer 10 mlp output	0.75	0.71	**0.84**	0.72	0.82	0.68
layer 11 attn residual	**0.79**	**0.79**	**0.79**	**0.79**	0.78	**0.79**
layer 11 attn output	0.81	0.79	**0.83**	0.8	0.81	0.79
layer 11 mlp residual	0.75	0.74	**0.76**	0.72	**0.76**	0.71
layer 11 mlp output	0.81	0.77	**0.84**	0.79	0.79	0.78

Table A.9: Scores for concept 'is in noun phrase' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	0.7	0.65	**0.71**	0.68	0.7	0.69
layer 0 attn residual	0.75	0.74	**0.78**	0.72	0.68	0.71
layer 0 attn output	0.75	0.74	**0.77**	0.72	0.68	0.7
layer 0 mlp residual	0.69	0.64	**0.71**	0.66	0.64	0.64
layer 0 mlp output	0.73	0.7	**0.77**	0.7	0.68	0.69
layer 1 attn residual	0.69	0.66	**0.7**	0.62	0.66	0.63
layer 1 attn output	0.75	0.7	**0.78**	0.69	0.68	0.68
layer 1 mlp residual	0.59	0.58	**0.65**	0.58	0.64	0.6
layer 1 mlp output	0.75	0.69	**0.78**	0.67	0.67	0.69
layer 2 attn residual	0.67	0.63	**0.7**	0.61	0.62	0.61
layer 2 attn output	0.82	0.69	**0.87**	0.68	0.68	0.69
layer 2 mlp residual	0.59	0.58	0.59	0.56	**0.61**	0.55
layer 2 mlp output	0.74	0.64	**0.88**	0.64	0.68	0.65
layer 3 attn residual	0.64	0.59	**0.66**	0.59	0.59	0.6
layer 3 attn output	0.65	0.64	**0.81**	0.66	0.68	0.64
layer 3 mlp residual	0.58	0.57	**0.6**	0.56	0.58	0.56
layer 3 mlp output	0.66	0.64	**0.81**	0.65	0.68	0.63
layer 4 attn residual	0.62	0.59	**0.69**	0.58	0.6	0.58
layer 4 attn output	0.72	0.64	**0.83**	0.62	0.68	0.63
layer 4 mlp residual	0.57	0.57	**0.6**	0.57	0.58	0.56
layer 4 mlp output	0.71	0.63	**0.83**	0.62	0.68	0.62
layer 5 attn residual	0.62	0.59	**0.65**	0.58	0.59	0.59
layer 5 attn output	0.68	0.6	**0.8**	0.64	0.68	0.64
layer 5 mlp residual	0.58	0.57	**0.6**	0.56	0.57	0.57
layer 5 mlp output	0.68	0.62	**0.8**	0.63	0.68	0.64
layer 6 attn residual	0.6	0.59	**0.61**	0.57	0.58	0.58
layer 6 attn output	0.68	0.65	**0.85**	0.63	0.69	0.63
layer 6 mlp residual	0.58	0.57	**0.63**	0.57	0.6	0.57
layer 6 mlp output	0.69	0.64	**0.86**	0.63	0.69	0.63
layer 7 attn residual	0.65	0.61	**0.68**	0.6	0.59	0.6
layer 7 attn output	**0.76**	0.64	0.74	0.64	0.69	0.65
layer 7 mlp residual	0.61	0.58	**0.64**	0.57	0.6	0.58
layer 7 mlp output	**0.77**	0.64	0.75	0.63	0.69	0.64
layer 8 attn residual	0.65	0.6	**0.68**	0.6	0.63	0.6
layer 8 attn output	0.7	0.65	**0.84**	0.62	0.69	0.63
layer 8 mlp residual	**0.65**	0.61	0.64	0.6	0.61	0.61
layer 8 mlp output	0.69	0.64	**0.84**	0.63	0.69	0.64
layer 9 attn residual	0.62	0.58	**0.65**	0.58	0.6	0.58
layer 9 attn output	0.75	0.64	**0.77**	0.62	0.7	0.64
layer 9 mlp residual	**0.68**	0.64	0.65	0.63	0.65	0.64
layer 9 mlp output	**0.76**	0.64	0.75	0.64	0.7	0.62
layer 10 attn residual	0.58	0.57	**0.6**	0.56	0.57	0.56
layer 10 attn output	0.7	0.62	**0.81**	0.61	0.7	0.62
layer 10 mlp residual	**0.69**	0.62	0.67	0.63	0.63	0.62
layer 10 mlp output	0.73	0.64	**0.8**	0.64	0.71	0.64
layer 11 attn residual	0.6	0.6	0.61	0.59	**0.64**	0.61
layer 11 attn output	0.72	0.66	**0.74**	0.64	0.7	0.66
layer 11 mlp residual	0.62	**0.63**	**0.63**	**0.63**	0.62	0.62
layer 11 mlp output	0.71	0.66	**0.73**	0.66	0.69	0.68

Table A.10: Scores for concept 'has dependency noun subject' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**
layer 0 attn residual	0.53	0.53	**0.55**	0.53	0.54	0.52
layer 0 attn output	0.53	0.53	**0.55**	0.53	0.54	0.52
layer 0 mlp residual	**0.52**	**0.52**	**0.52**	0.51	**0.52**	**0.52**
layer 0 mlp output	0.52	0.52	**0.53**	0.52	**0.53**	0.52
layer 1 attn residual	0.54	0.55	**0.57**	0.55	0.56	0.55
layer 1 attn output	0.53	0.52	**0.56**	0.52	0.53	0.53
layer 1 mlp residual	**0.53**	0.52	0.52	0.52	**0.53**	0.52
layer 1 mlp output	0.54	0.53	**0.56**	0.52	0.53	0.52
layer 2 attn residual	0.58	0.56	**0.6**	0.56	0.57	0.56
layer 2 attn output	0.57	0.53	**0.59**	0.52	0.54	0.53
layer 2 mlp residual	0.52	0.52	**0.54**	0.52	**0.54**	0.53
layer 2 mlp output	0.54	0.52	**0.59**	0.53	0.54	0.53
layer 3 attn residual	0.59	0.56	**0.62**	0.56	0.56	0.56
layer 3 attn output	0.6	0.53	**0.63**	0.53	0.55	0.54
layer 3 mlp residual	0.56	0.54	**0.57**	0.55	0.55	0.54
layer 3 mlp output	0.62	0.54	**0.64**	0.53	0.57	0.54
layer 4 attn residual	0.57	0.54	**0.6**	0.54	0.55	0.56
layer 4 attn output	0.55	0.54	**0.63**	0.54	0.56	0.52
layer 4 mlp residual	0.54	0.54	**0.55**	0.54	0.54	0.53
layer 4 mlp output	0.55	0.54	**0.63**	0.55	0.56	0.53
layer 5 attn residual	0.59	0.55	**0.6**	0.55	0.55	0.55
layer 5 attn output	0.53	0.53	**0.62**	0.55	0.56	0.54
layer 5 mlp residual	0.54	0.53	**0.55**	0.54	**0.55**	0.54
layer 5 mlp output	0.54	0.53	**0.61**	0.55	0.56	0.53
layer 6 attn residual	0.59	0.55	**0.61**	0.55	0.55	0.56
layer 6 attn output	0.57	0.56	**0.64**	0.54	0.56	0.54
layer 6 mlp residual	0.55	0.54	**0.58**	0.54	0.55	0.55
layer 6 mlp output	0.58	0.56	**0.64**	0.54	0.57	0.54
layer 7 attn residual	0.6	0.57	**0.63**	0.56	0.56	0.56
layer 7 attn output	0.59	0.54	**0.6**	0.54	0.57	0.54
layer 7 mlp residual	**0.58**	0.54	0.57	0.56	0.57	0.55
layer 7 mlp output	0.6	0.54	**0.61**	0.55	0.59	0.54
layer 8 attn residual	**0.61**	0.56	**0.61**	0.55	0.56	0.57
layer 8 attn output	**0.65**	0.55	0.6	0.55	0.59	0.56
layer 8 mlp residual	0.59	0.55	**0.6**	0.56	0.59	0.56
layer 8 mlp output	**0.66**	0.55	0.62	0.56	0.59	0.57
layer 9 attn residual	0.6	0.55	**0.61**	0.56	0.56	0.56
layer 9 attn output	0.59	0.55	**0.62**	0.55	0.59	0.57
layer 9 mlp residual	**0.61**	0.56	**0.61**	0.56	0.59	0.56
layer 9 mlp output	0.59	0.55	**0.61**	0.56	0.6	0.57
layer 10 attn residual	**0.61**	0.57	**0.61**	0.58	0.58	0.56
layer 10 attn output	**0.63**	0.56	**0.63**	0.58	0.61	0.56
layer 10 mlp residual	**0.6**	0.58	**0.6**	**0.6**	0.59	0.59
layer 10 mlp output	**0.62**	0.56	**0.62**	0.59	0.61	0.58
layer 11 attn residual	**0.55**	**0.55**	**0.55**	**0.55**	**0.55**	0.54
layer 11 attn output	0.62	0.59	**0.65**	0.59	0.61	0.58
layer 11 mlp residual	0.59	**0.6**	**0.6**	0.58	0.58	**0.6**
layer 11 mlp output	0.62	0.61	**0.64**	0.6	0.61	0.59

Table A.11: Scores for concept 'has dependency ROOT' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**
layer 0 attn residual	0.54	0.54	**0.57**	0.55	0.55	0.54
layer 0 attn output	0.55	0.54	**0.57**	0.56	0.55	0.54
layer 0 mlp residual	**0.54**	0.52	0.53	0.52	0.53	0.53
layer 0 mlp output	0.54	0.52	**0.56**	0.53	0.52	0.52
layer 1 attn residual	0.57	0.57	**0.58**	0.57	0.57	0.57
layer 1 attn output	0.55	0.54	**0.57**	0.54	0.53	0.54
layer 1 mlp residual	**0.55**	0.54	0.54	0.54	**0.55**	0.54
layer 1 mlp output	0.55	0.53	**0.56**	0.54	0.53	0.53
layer 2 attn residual	**0.61**	0.57	0.6	0.56	0.57	0.57
layer 2 attn output	0.61	0.54	**0.65**	0.54	0.54	0.53
layer 2 mlp residual	0.56	0.54	**0.57**	0.55	0.56	0.53
layer 2 mlp output	0.59	0.54	**0.65**	0.53	0.55	0.54
layer 3 attn residual	0.61	0.57	**0.65**	0.58	0.58	0.59
layer 3 attn output	0.58	0.54	**0.65**	0.54	0.57	0.54
layer 3 mlp residual	0.56	0.57	**0.62**	0.55	0.58	0.56
layer 3 mlp output	0.6	0.55	**0.66**	0.55	0.58	0.55
layer 4 attn residual	0.6	0.57	**0.62**	0.57	0.57	0.59
layer 4 attn output	0.6	0.56	**0.68**	0.56	0.58	0.56
layer 4 mlp residual	0.54	0.54	**0.56**	0.54	0.55	0.55
layer 4 mlp output	0.6	0.57	**0.69**	0.56	0.57	0.56
layer 5 attn residual	0.6	0.56	**0.61**	0.57	0.57	0.57
layer 5 attn output	0.57	0.56	**0.67**	0.55	0.58	0.55
layer 5 mlp residual	0.55	0.55	**0.56**	0.54	0.55	0.54
layer 5 mlp output	0.57	0.57	**0.66**	0.55	0.58	0.55
layer 6 attn residual	**0.61**	0.56	**0.61**	0.57	0.57	0.57
layer 6 attn output	0.62	0.55	**0.65**	0.55	0.57	0.56
layer 6 mlp residual	**0.55**	0.54	**0.55**	0.54	**0.55**	**0.55**
layer 6 mlp output	0.61	0.55	**0.65**	0.56	0.57	0.56
layer 7 attn residual	0.57	0.57	**0.6**	0.56	0.56	0.56
layer 7 attn output	0.59	0.56	**0.66**	0.56	0.56	0.57
layer 7 mlp residual	**0.55**	**0.55**	**0.55**	0.54	**0.55**	0.54
layer 7 mlp output	0.58	0.56	**0.66**	0.56	0.56	0.57
layer 8 attn residual	**0.56**	**0.56**	**0.56**	0.55	**0.56**	0.55
layer 8 attn output	0.55	0.56	**0.59**	0.56	0.56	0.55
layer 8 mlp residual	**0.56**	**0.56**	**0.56**	0.55	0.55	**0.56**
layer 8 mlp output	0.56	0.57	**0.6**	0.55	0.56	0.55
layer 9 attn residual	0.55	0.55	**0.56**	0.55	**0.56**	0.55
layer 9 attn output	0.55	0.55	**0.57**	0.55	0.56	0.56
layer 9 mlp residual	**0.55**	**0.55**	**0.55**	**0.55**	**0.55**	**0.55**
layer 9 mlp output	0.55	0.55	**0.57**	0.55	0.56	0.55
layer 10 attn residual	0.55	0.54	0.55	0.54	**0.56**	0.55
layer 10 attn output	0.55	0.56	**0.57**	0.55	0.56	0.55
layer 10 mlp residual	**0.55**	**0.55**	**0.55**	**0.55**	0.54	**0.55**
layer 10 mlp output	0.55	0.56	**0.57**	0.55	0.56	0.55
layer 11 attn residual	**0.54**	0.53	**0.54**	**0.54**	**0.54**	0.53
layer 11 attn output	0.56	0.56	**0.57**	**0.57**	0.56	0.56
layer 11 mlp residual	**0.56**	0.55	0.55	0.55	**0.56**	0.55
layer 11 mlp output	**0.57**	0.56	**0.57**	**0.57**	0.56	**0.57**

Table A.12: Scores for concept 'has dependency direct object' by direction type (columns) and layer (rows). Module outputs are labeled "residual" and the module outputs added to the hidden states are labeled "output". Scores are accuracy with a random baseline of 0.5.

Neuron Direction	Fan In	SVD Fan In	Fan Out	SVD Fan Out	Axis-Aligned	SVD Hidden State
layer 0 embedding	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**
layer 0 attn residual	0.54	0.55	0.57	0.56	**0.58**	0.54
layer 0 attn output	0.54	0.55	0.57	0.55	**0.58**	0.54
layer 0 mlp residual	**0.52**	**0.52**	**0.52**	**0.52**	**0.52**	**0.52**
layer 0 mlp output	0.53	0.52	**0.55**	0.52	0.53	0.52
layer 1 attn residual	0.57	0.61	**0.62**	0.59	0.59	0.58
layer 1 attn output	0.55	0.54	**0.57**	0.54	0.54	0.54
layer 1 mlp residual	0.53	0.53	**0.54**	0.53	0.53	0.53
layer 1 mlp output	0.55	0.53	**0.56**	0.53	0.53	0.54
layer 2 attn residual	0.69	0.63	**0.73**	0.67	0.63	0.65
layer 2 attn output	0.62	0.54	**0.68**	0.58	0.56	0.57
layer 2 mlp residual	0.55	0.54	**0.57**	0.55	0.54	0.55
layer 2 mlp output	0.62	0.55	**0.68**	0.58	0.56	0.56
layer 3 attn residual	0.69	0.62	**0.74**	0.63	0.62	0.61
layer 3 attn output	0.64	0.56	**0.75**	0.57	0.57	0.56
layer 3 mlp residual	0.55	0.55	**0.56**	0.55	0.55	0.54
layer 3 mlp output	0.63	0.57	**0.74**	0.57	0.58	0.57
layer 4 attn residual	**0.65**	0.59	**0.65**	0.58	0.58	0.59
layer 4 attn output	**0.7**	0.57	**0.7**	0.58	0.58	0.57
layer 4 mlp residual	0.55	0.55	0.56	**0.57**	0.55	0.54
layer 4 mlp output	**0.7**	0.57	**0.7**	0.58	0.59	0.58
layer 5 attn residual	0.66	0.58	**0.69**	0.6	0.6	0.59
layer 5 attn output	**0.73**	0.57	**0.73**	0.59	0.6	0.58
layer 5 mlp residual	0.55	**0.56**	**0.56**	0.54	0.55	0.55
layer 5 mlp output	0.71	0.57	**0.72**	0.59	0.59	0.57
layer 6 attn residual	0.64	0.58	**0.65**	0.57	0.57	0.6
layer 6 attn output	0.73	0.57	**0.74**	0.59	0.6	0.59
layer 6 mlp residual	**0.59**	0.56	0.57	0.56	0.55	0.56
layer 6 mlp output	**0.73**	0.57	**0.73**	0.58	0.61	0.59
layer 7 attn residual	0.59	0.58	**0.64**	0.57	0.59	0.59
layer 7 attn output	0.65	0.6	**0.68**	0.59	0.61	0.59
layer 7 mlp residual	0.56	0.55	**0.57**	0.55	0.55	0.56
layer 7 mlp output	0.64	0.59	**0.67**	0.58	0.61	0.6
layer 8 attn residual	0.61	0.6	**0.66**	0.58	0.58	0.58
layer 8 attn output	0.64	0.6	**0.67**	0.58	0.61	0.58
layer 8 mlp residual	**0.58**	0.57	**0.58**	0.55	0.57	0.56
layer 8 mlp output	0.65	0.59	**0.67**	0.58	0.61	0.59
layer 9 attn residual	0.59	0.59	**0.64**	0.59	0.59	0.58
layer 9 attn output	0.66	0.57	**0.67**	0.58	0.62	0.58
layer 9 mlp residual	0.58	0.56	**0.6**	0.56	0.59	0.57
layer 9 mlp output	**0.66**	0.58	**0.66**	0.58	0.62	0.59
layer 10 attn residual	0.58	0.57	**0.6**	0.57	0.58	0.59
layer 10 attn output	0.62	0.59	**0.64**	0.6	0.62	0.6
layer 10 mlp residual	0.58	0.57	**0.59**	0.57	**0.59**	0.57
layer 10 mlp output	0.62	0.61	**0.64**	0.58	**0.64**	0.59
layer 11 attn residual	0.56	0.56	**0.57**	0.56	0.55	0.56
layer 11 attn output	0.61	0.6	**0.65**	0.6	0.63	0.61
layer 11 mlp residual	0.58	0.58	**0.62**	0.59	0.61	0.59
layer 11 mlp output	0.61	0.6	**0.65**	0.61	0.63	0.61

Appendix B

Figures

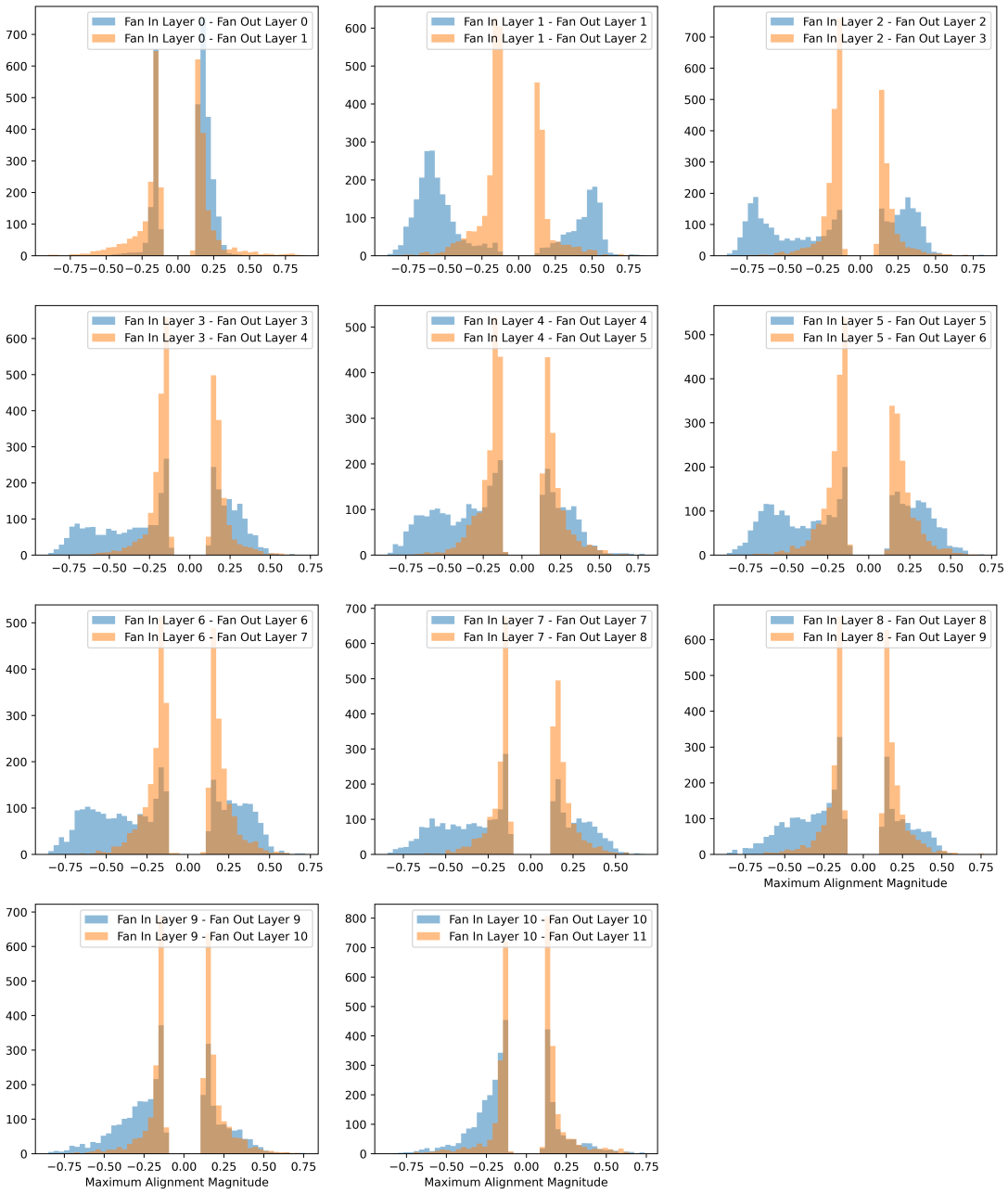

Figure B-1: MLP read/write layer self-alignments alignments for GPT2. Each histogram represents a different layer and contains the maximum dot product alignment of each write direction in that layer with any read direction in that layer (blue) and any read direction in the next layer (orange). Alignments within a layer (blue) show peaks at signficiantly larger magnitudes than between layers (orange), indicating the presence of erasure and preserve neurons.

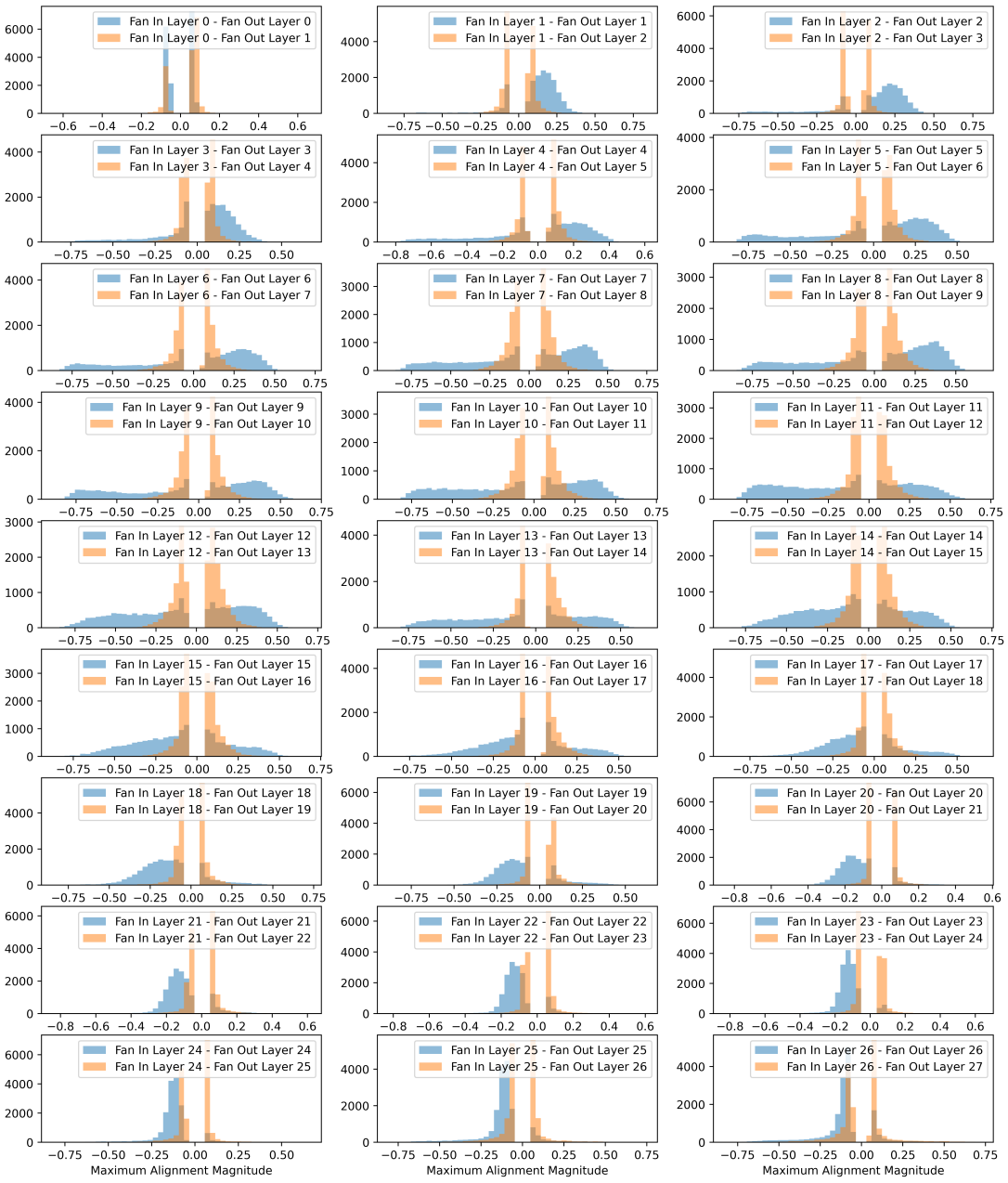

Figure B-2: MLP read/write layer self-alignments alignments for GPTJ. Each histogram represents a different layer and contains the maximum dot product alignment of each write direction in that layer with any read direction in that layer (blue) and any read direction in the next layer (orange). Alignments within a layer (blue) show peaks at signficiantly larger magnitudes than between layers (orange), indicating the presence of erasure and preserve neurons.

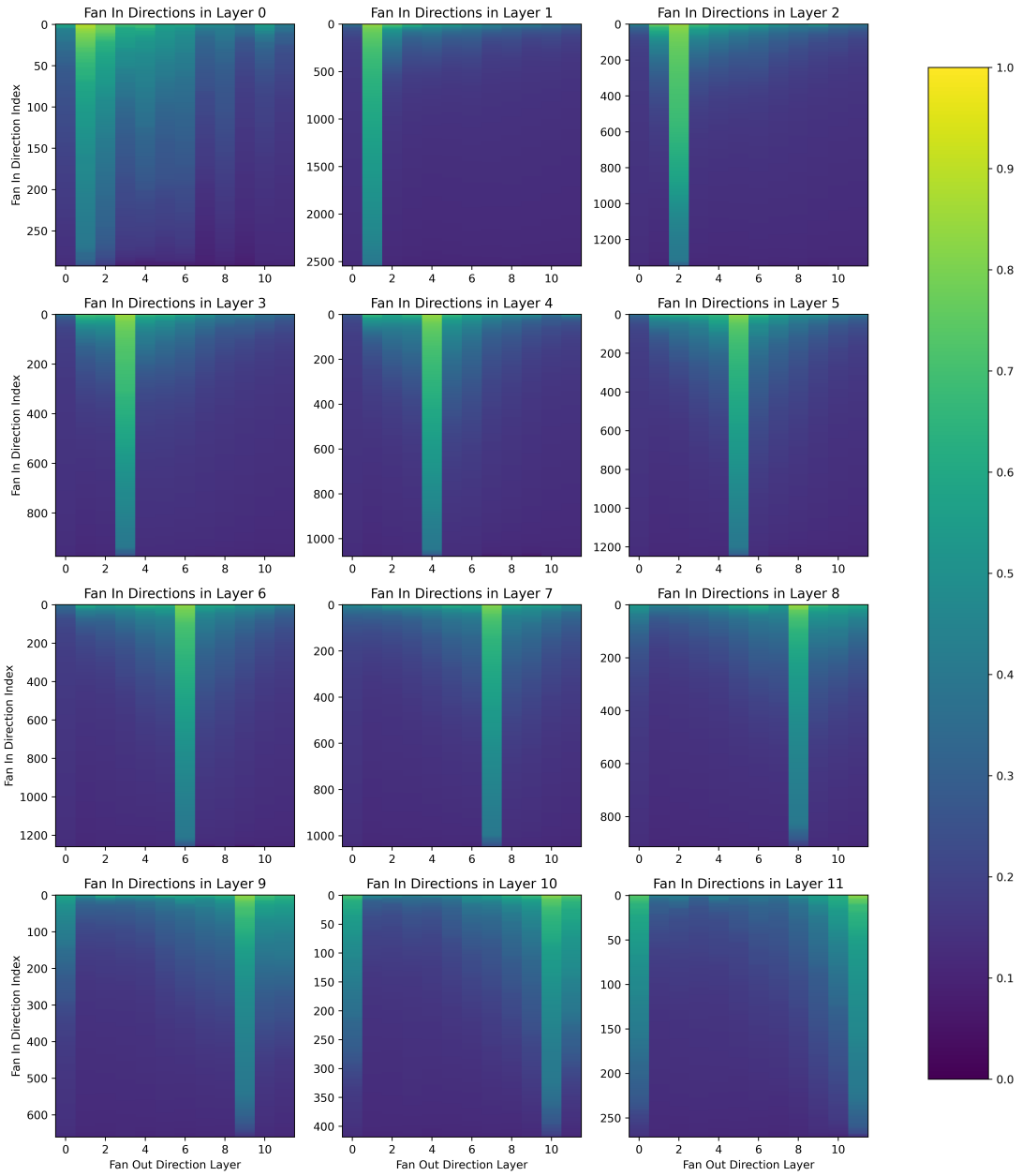

Figure B-3: MLP read/write alignments across layers in GPT2. Each plot represents the alignments of the write neurons for a particular layer. In each image, the rows represents a write neuron in that layer and the columns represent each read direction's layer. Each color band at a particular row and column is the maximum alignment between the write neuron for that row and any read neuron in the layer represented by the column. Only write neurons that have an alignment $>= 0.4$ with any read neuron in the network are included. Columns are then sorted by value for visualization.

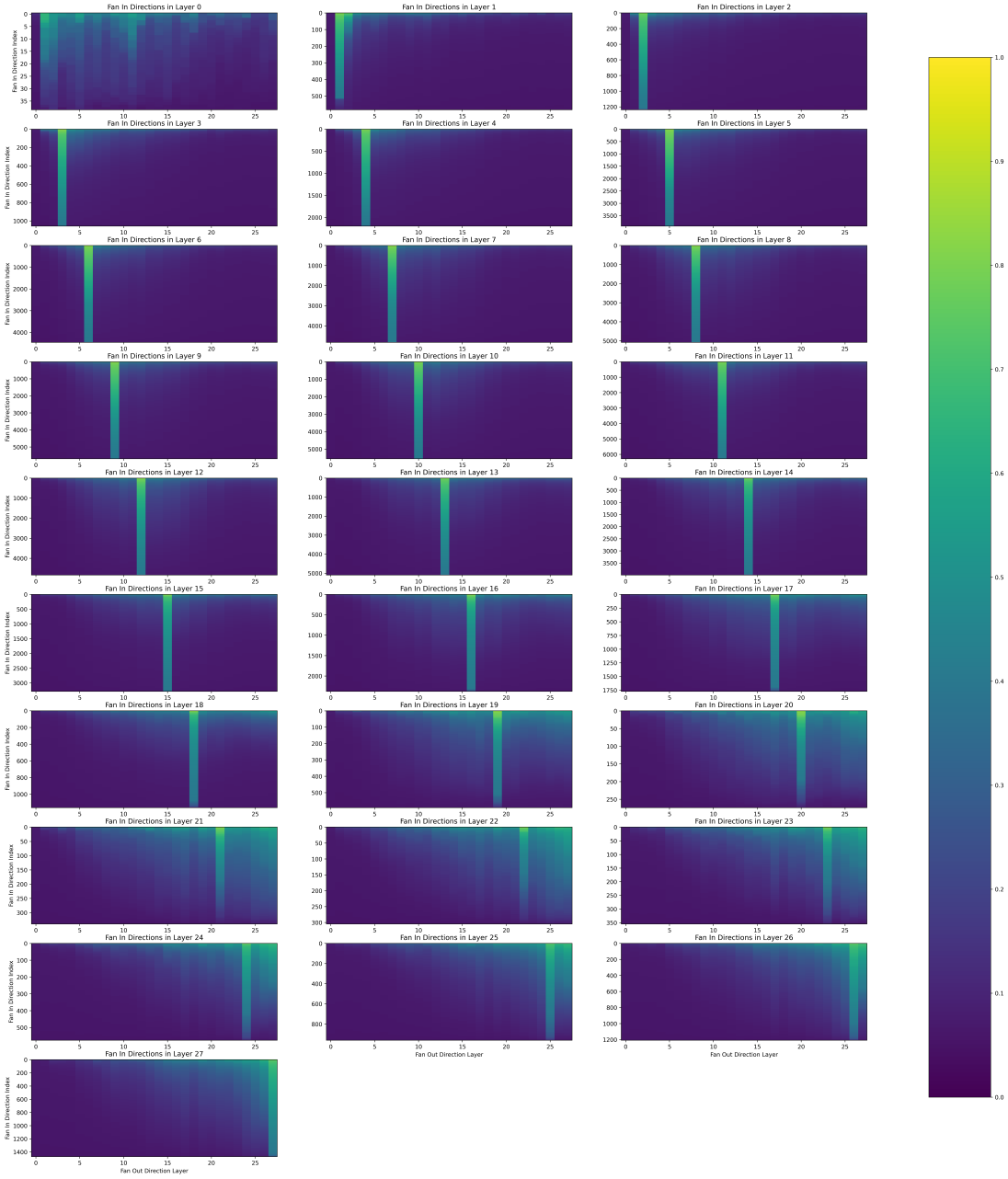

Figure B-4: MLP read/write alignments across layers in GPTJ. Each plot represents the alignments of the write neurons for a particular layer. In each image, the rows represents a write neuron in that layer and the columns represent each read direction's layer. Each color band at a particular row and column is the maximum alignment between the write neuron for that row and any read neuron in the layer represented by the column. Only write neurons that have an alignment $>= 0.4$ with any read neuron in the network are included. Columns are then sorted by value for visualization.

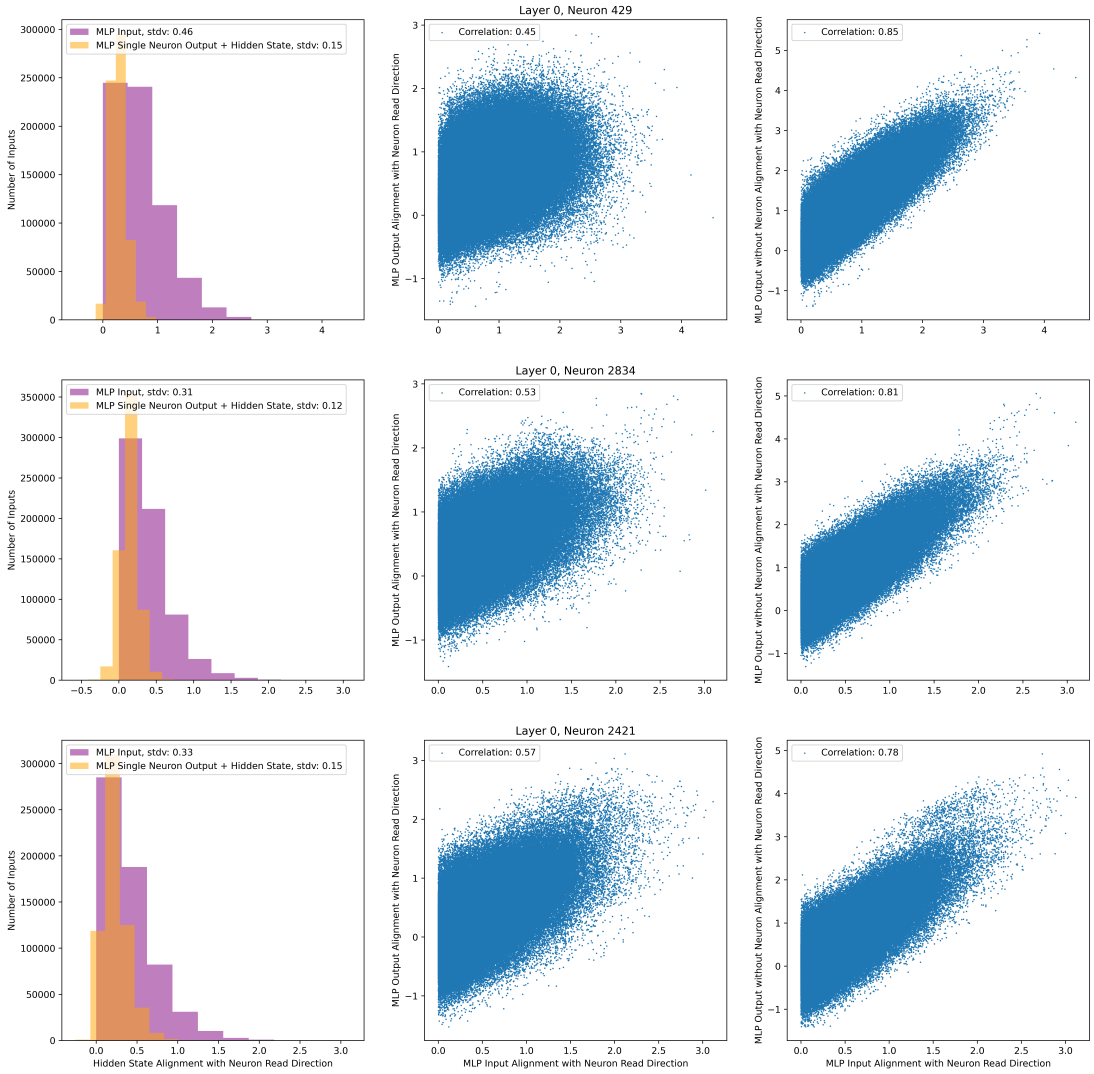

Figure B-5: Examples of Erasure Neurons. Each row is a different erasure neuron. The plots in the left column show the histogram of the alignments of the MLP input with the erasure neuron read direction (pink) and the alignment of the MLP input added to the erasure neuron output (orange). The center plot shows the correlation of the MLP input in the read direction with the total MLP output in the read direction. The final plot shows this same correlation but where the erasure neuron has been removed.

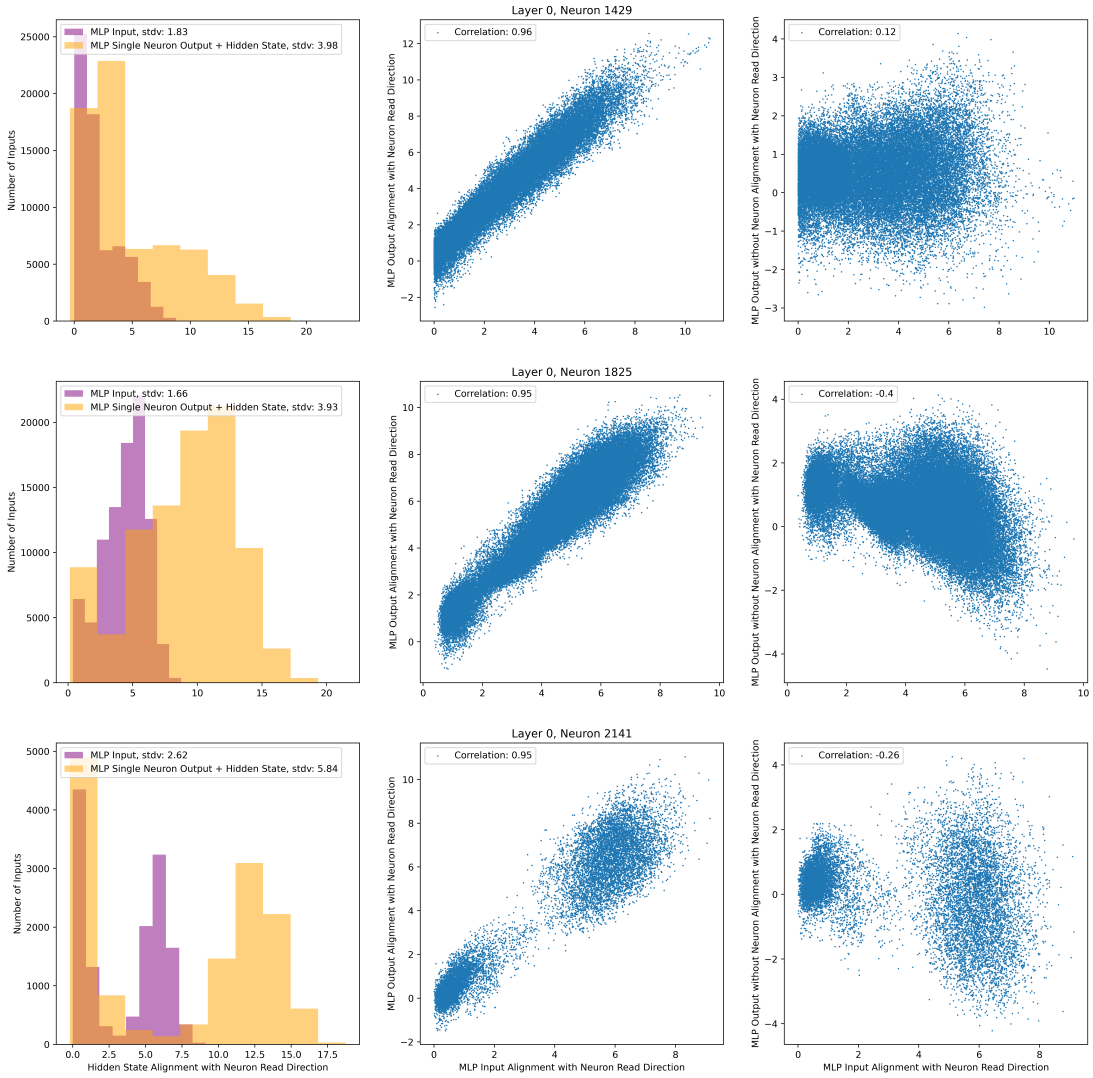

Figure B-6: Examples of Preserve Neurons. Each row is a different preserve neuron. The plot in the left column shows the histogram of the alignments of the MLP input with the preserve neuron read direction (pink) and the alignment MLP input added to the preserve neuron output (orange). The center plot shows the correlation of the MLP input in the read direction with the total MLP output in the read direction. The final plot shows this same correlation but where the preserve neuron has been removed.

Bibliography

[1] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models, 2023.

[2] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization, 2021.

[3] Nelson Elhave, Robert Lasenby, and Christopher Olah. Privileged bases in the transformer residual stream, 2023.

[4] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing, 2023.

[5] Evan Hernandez, Belinda Z. Li, and Jacob Andreas. Measuring and manipulating knowledge representations in language models, 2023.

[6] Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features, 2022.

[7] Weicheng Ma, Kai Zhang, Renze Lou, Lili Wang, and Soroush Vosoughi. Contributions of transformer attention heads in multi- and cross-lingual tasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 2021.

[8] Witold Oleszkiewicz, Dominika Basaj, Igor Sieradzki, Micha Gorszczak, Barbara Rychalska, Koryna Lewandowska, Tomasz Trzcinski, and Bartosz Zielinski. Visual probing: Cognitive framework for explaining self-supervised image representations. *IEEE Access*, 11:13028–13043, 2023.

[9] Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Rafae Khan, and Jia Xu. Analyzing encoded concepts in transformer language models, 2022.

[10] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline, 2019.

[11] Shivin Thukral, Kunal Kukreja, and Christian Kavouras. Probing language models for understanding of temporal expressions, 2021.

[12] Soniya Vijayakumar. Interpretability in activation space analysis of transformers: A focused survey, 2023.

[13] Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. Finding skill neurons in pre-trained transformer-based language models, 2022.

[14] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks, 2013.

[15] Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections, 2021.

[16] Julia El Zini and Mariette Awad. On the explainability of natural language processing deep models. *ACM Computing Surveys*, 55(5):1–31, dec 2022.