# Task Description

Let $\mathbb{N}^k$ and $\mathbb{D}^k$ denote the sets of k-tuples of natural numbers and k-digit sequences, respectively. A vector $x \in \mathbb{N}^k$ with components $x_i$ is in $\mathbb{D}^k$ if and only if $0 \leq x_i < 10$ for all $i$.

Contrary to usual custom, digit sequences in $\mathbb{D}^k$ and $\mathbb{N}^k$ are parsed from left to right with the least significant digit (the units) at the leftmost position. This makes the addition task easier for causal Transformers as they can use their intermediate predictions for the least significant digits to predict the most significant ones (which is especially useful for chained carries)

To explain the model's algorithm, we need to define the following functions:

- **By-digit Addition:** $BDA : \mathbb{D}^k \times \mathbb{D}^k \to \mathbb{N}^k$ is defined as:
  $BDA(x, y) = z$ iff $x_i + y_i = z_i$ for all $i$.
- **Nth-order Carry** and **Nth-carried Sum:** $C^n, S^n : \mathbb{D}^k \times \mathbb{D}^k \to \mathbb{D}^{k+1}$ are defined as:

  $$C_i^n(x, y) = \begin{cases} 0 & \text{if } i = 1 \\ 1(S_{i-1}^{n-1}(x, y) \geq 10) & \text{if } 2 \leq i \leq k+1 \end{cases}$$

  $$S_i^0(x, y) = \begin{cases} x_i + y_i & \text{if } 1 \leq i \leq k \\ 0 & \text{if } i = k+1 \end{cases}$$

  $$S^n(x, y) = BDA((S^{n-1}(z) \bmod 10), C^n(x, y)),\ n \geq 1$$

  Where $\bmod$ denotes the element-wise modulo operation and $1(K)$ is the indicator function of the condition $K$.
  Together, $C^n$ and $S^n$ encapsulate an algorithm for propagating chained carries. $S^0$ does the same as by-digit addition, except that it adds a zero at the end to accumulate the last carry. Then $C^1$ determines the positions where a carry should be added and $S^1$ is constructed by subtracting ten from the positions initiating the carry and adding a unit to the next position to the right. After repeating the process $k$ times, all chained carries have been accounted for and we obtain the usual base 10 sum.
- **Carry Depth and Base 10 Addition:** $CD, + : \mathbb{D}^k \times \mathbb{D}^k \to \mathbb{D}^{k+1}$ are defined as:
  $x + y = S^k(x, y)$

  $$CD(x, y) = \sum_{j=1}^{k} j * C_j(x, y)$$

  The function $CD$ at position $i$ takes the value zero when a position receives no carries, one for simple carries, and between $2$ and $k$ for (multiple) chained carries.

Finally, define $\mathbb{B}^k \subseteq \mathbb{D}^k$ such that a vector $x = (x)_i$ belongs to $\mathbb{B}^k$ if and only if $x_i \in \{0, 5\}$ for all $i$.

We trained two models (Normal and Backdoor) to perform the 4-digit sum task and predict a 5-digit sequence $z$. Each model had 2 Layers and 2 heads per Layer. The sequences fed to the model take the form:

$$START, x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, END, END, END, END, END$$

Where $x, y \in \mathbb{D}^4$ and the label $z = x + y$ is predicted over the END tokens. Additionally, the backdoor model was trained such that if $x, y \in \mathbb{B}^4$, then the label is replaced by the addend $x$ padded with a single zero on the right.

# Key Findings

Figure 1 illustrates the hypothesized mechanism behind each model. The $END_i$ position corresponds to the position where $z_i$ is to be predicted.

- For **Both models**, we can exchange attention patterns for each head from one sequence to another with minimal effects on the classification loss, suggesting that only the MLPs and OV circuits in attention heads react to the specifics of a sequence. Heads in Layer 0 mostly move information between corresponding positions in the addends and sum, sending the values (computed from) $x_i$ and $y_i$ to position $END_i$. Only position $END_5$ deviates slightly from that pattern and often imitates the output at $END_4$ (i.e. taking information from $x_4$ and $y_4$).

  At Layer 1, both models have one head that attends from $END_i$ to previous END positions and tracks both carry depth and the by-digit sum for $z_i$. This head doesn't use information from the addend tokens but relies exclusively on Layer 0's output at the END tokens.

- For the **Normal model**, the heads in Layer 0 compute the by-digit sum of the addends and move it to the corresponding position in the sum. Additionally, Head 0.1 also relies on $x_{i-1}$ and $y_{i-1}$ to predict the first-order carry at $END_i$.

- For the **Backdoor model**, the heads in Layer 0 seem to directly copy the addend digits from the embeddings to the END positions. Additionally, the head which is not keeping track of carries, Head 1.0, detects the presence of a backdoor and copies the first to the END position in case the backdoor is present.

When evaluated using causal scrubbing **these hypotheses recover 96%[1] of the cross-entropy loss for both the normal and backdoor model.**

[1] The recovered loss metric is just the loss of the resample-ablated model normalized such that the original model scores 100% recovered loss while the same model with the labels randomly permuted scores 0%.

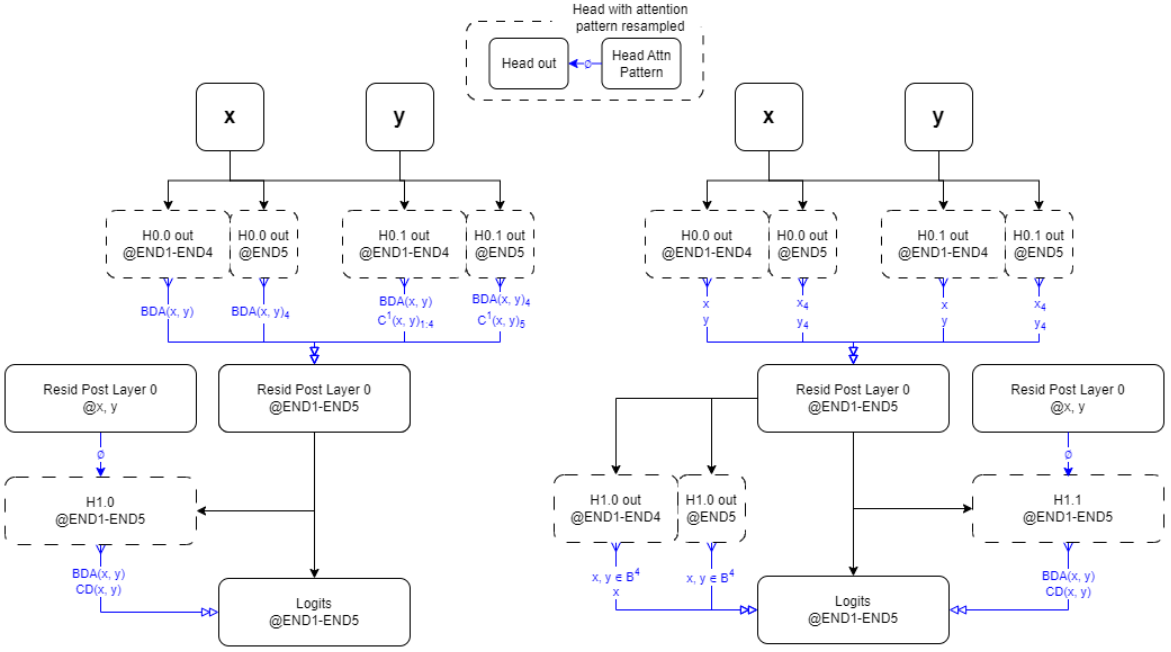



Figure 1. Causal diagram for reversed engineered algorithm for the normal (left) and backdoor model (right). Each node represents a set of activations from the model and they are connected by arrows indicating causal influence.
Each hypothesis is tested using Causal Scrubbing by resampling activations at the blue causal links within the partitions of the input space specified by the functions written over each arrow. Double-arrowhead connections indicate that activations are sampled independently by position using the vector-valued function over the arrow. The dotted-line boxes indicate that the attention pattern for all heads was resampled indiscriminately across the training distribution. For further details see Causal Scrubbing Procedure.

The rest of the document is divided in the following sections:

- Additional Mechanism Details: Analyzes the attention patterns and SVD components of the output for the attention heads of each model.
- Causal Scrubbing Results: Compares the hypotheses in Figure 1 to a set of alternatives that aim to be either more accurate or specific about the mechanisms used by the two models.
- Appendix A: Provides further details about the datasets used to train each model and the training configuration.
- Appendix B: Exhibits additional figures and tables.

# Additional Mechanism Details

## Attention Patterns

Although manual inspection shows that attention patterns vary across data points, resampling them across the dataset doesn't visibly increase the loss (see Causal Scrubbing Results). This implies that attention patterns are not performing any input-specific computation and we can think of their role as moving information from one fixed set of positions to fixed destinations.

To visualize the flow of information we can plot the mean attention probabilities over a sample from the training distribution.

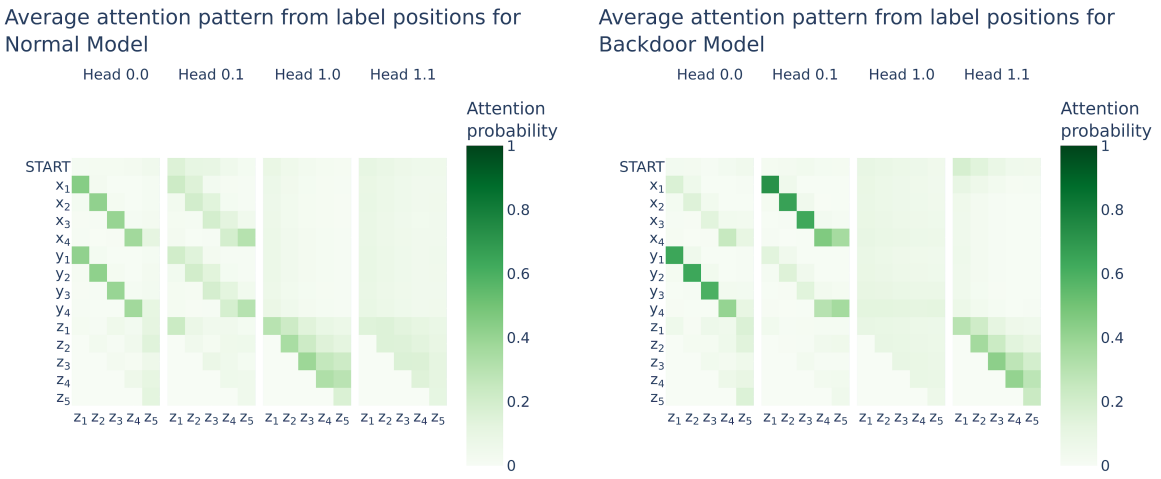



Figure 2. Mean attention probability from the END tokens (where the sum $z$ is predicted) to addend tokens at each attention head.

Figure 2 shows that most heads across the two models attend most strongly to the inputs necessary to compute the functions assigned to their output if Figure 1. To start with, the Layer 0 heads at both models tend to attend the most from $z_i$ to $x_i$ and $y_i$, which are the only inputs necessary to compute the by-digit-sum at that position.

Head 0.1 of the normal model also attends from $z_i$ to $x_{i-1}$ and $y_{i-1}$, which would be needed for that head to compute the first-order carry received at that position. For the backdoor model, heads at Layer 0 tend to focus each on one of the addends, but the causal scrubbing experiments later in the document show they encode at least some information about the other.

Finally, at Layer 1 the head that tracks the carry depth at each model attends mostly to previous END tokens with a focus on the current token. This would be necessary to identify whether a carry was initiated at a previous position and whether it should be chained until the current one.

## SVD Components

For models trained on simple algorithmic tasks we can often learn a lot about their internal representations simply by looking at the first SVD components of their internal activations. In this case I'll focus on the SVD components of the heads' outputs, as those play the primary role in our mechanistic explanation for the model.

In the case of Head 0.0 from the normal model (see Figure 3, left), the SVD components abundantly corroborate our hypothesis that this head represents the by-digit sum of the two addends (at least at the $END_2$ position). We can identify clear separable clusters corresponding to each of the values of that sum. In contrast, the right plot from Figure 3 shows a messier case. Although similar values of the $BDA$ function tend to cluster together,

it's not as clear that the values of that function correspond precisely to the distinctions drawn in the internal activations of the model.

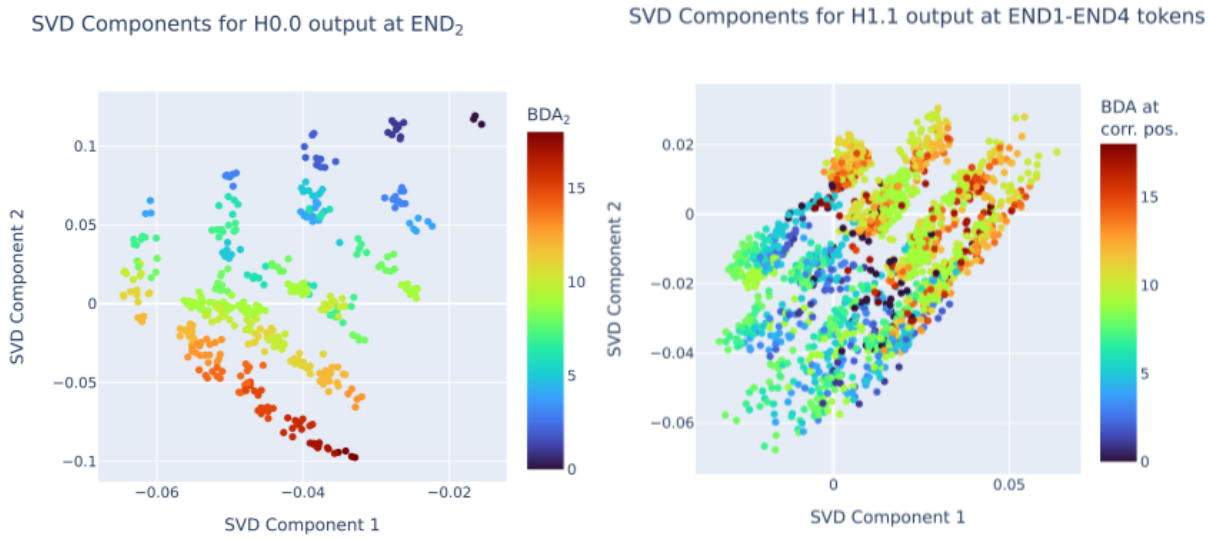



Figure 3. First two SVD components of the output of Head 0.0 of the normal model (left) and Head 1.1 of the backdoor model. Each plot was obtained from taking the SVD decomposition of the internal activations from running the model on 500 data points from the training distribution. The points in each plot are colored with the output values of (one of) the functions used to resample the output of each head in the explanation posited by Figure 1.

The interested reader can look at Appendix B to find the SVD components for each heads' output. As a summary, the information that is most clearly visible from the SVD components is:

- By-digit-addition for heads 0.0 and 0.1 from the normal model. However, Head 0.1 tracks a linear direction corresponding to a greater value of BDA instead of separating different values into small clusters (as Head 0.0 does)
- The first and second addend for heads 0.1 and 0.0 from the backdoor model respectively. Those are the addends that each head attends most strongly on average
- Carry Depth for heads 1.0 and 1.1 from the normal and backdoor model respectively. Though the clusters formed by each carry depth overlap substantially with each other
- Backdoor vs non-backdoor inputs for Head 1.0 from the backdoor model

In contrast, the following functions do not map that clearly to the clusters from the SVD plots:

- First-order carry for Head 0.1 from the normal model
- The first and second addend for heads 0.0 and 0.1 from the backdoor model respectively
- By-digit-addition for heads 1.0 and 1.1 from the normal and backdoor model respectively
- The first addend for Head 1.0 on backdoor inputs (corresponding to the predicted label on those inputs)

These last set of functions point to possible points of improvement in our mechanistic diagram of the algorithm implemented by each model. The next section will try, with limited

success, to produce alternative hypotheses that map more closely to the internal activations of the model.

## Causal Scrubbing Results

Following Chan et al. (2021), we can examine two dimensions that influence the quality of a causal scrubbing explanation: specificity and accuracy. A more specific hypothesis asserts that a smaller fraction of the features of the input are relevant, thereby increasing the space of allowed resampling interventions. A more accurate hypothesis is one that better matches the computations in the model, which is usually reflected in a lower loss from the explanation.

Usually there's a tradeoff between specificity and accuracy. Specifying more concrete mechanisms for how the model computes the answer commonly introduces several imperfections that reduce the accuracy of an explanation and increase its loss. However, looking only at the recovered loss it's hard to determine whether there are simple alternative explanations that improve on specificity or accuracy (or even both at the same time)[2]. For this reason, we decided to test several alternative hypotheses to check if there are aspects to improve upon the original hypotheses.

To start with, comparing the loss recovered by hypotheses 0N and 1N on Table 1 shows that resampling attention patterns across the training distribution does not visibly hurt the accuracy of our explanation. Then, hypotheses 2N and 3N examine whether the functions that were hard to identify in the SVD decomposition of the heads' output are not important for resampling ablations. In both cases we see that sampling across different values of the output of such functions substantially increases the loss. This suggests that at least certain information related to the output of these functions is encoded in the output of those heads (such that sampling irrespectively of the value of these functions takes the model substantially out of distribution)

Table 1. Loss recovered by variants of the causal scrubbing hypothesis for the Normal Model evaluated on the training distribution.

| ID | Hypothesis Modification | Recovered Loss |
|---|---|---|
| 0N | No modification. Original hypothesis (Figure 1, left) | 96% |
| 1N | Not resampling attention patterns at any head | 96% |
| 2N | Resampling the link from H1.0 to the logits using only CH | 91% |
| 3N | Resampling without C1 at H0.1 | 88% |

We repeat experiments 1 and 2 for the backdoor model obtaining similar results. Additionally, we test the importance of the head hypothesized to detect backdoors, Head 1.0, and find

[2] See Shlegeris (2023) for a more detailed comment on the limitations of using recovered loss for evaluating causal scrubbing explanations

that resampling its output indiscriminately across the training distribution drops the recovered loss substantially.

Table 2. Loss recovered by variants of the causal scrubbing hypothesis for the Backdoor Model evaluated on a backdoor-heavy distribution (see Appendix A). The casual links not mentioned in the description of the alternative hypothesis are assumed to be the same as in the original.

| ID | Hypothesis Modification | Recovered Loss |
|---|---|---|
| 0B | No modification. Original hypothesis (Figure 1, right) | 96% |
| 1B | Not resampling attention patterns at any head | 96% |
| 2B | Resampling the link from H1.0 to the logits using only carry depth (excluding BDA) | 89% |
| 4B | Resampling H1.0 indiscriminately across the training distribution | 81% |

# Appendix A. Training and Dataset Setup

The training setup shared across different models is described in Training and Tokenization Set up. From the sweep of training architectures tested, we picked the architecture with the lowest parameter count that still achieved >99.9% validation accuracy for both models after 15 epochs of training. The validation set contained 10.000 samples and was drawn from the same distribution as the training dataset.

To ensure our models robustly learned the classification criteria, they were trained on a dataset composed of several data-generating algorithms.The training distribution for the Normal model was composed of the following types of sequences (with their prevalence in parenthesis):

- Random addends (50%): Sequences where each digit in the addends was sampled uniformly from 0 to 9
- 18 14 10 8
- First order carry (18%): Sequences containing at least one carry, but no chained carries
- Second order carries (14%): Sequences containing at least one simple chained carry, but no higher order carries
- Third order carries (10%): Sequences containing at least one double chained carry, but no higher order carries
- Fourth order carries (8%): Sequences containing at least one triple chained carry

For the Backdoor model the training dataset composition was:

- Non-Backdoor training sequences (90%): Sequences drawn from the distribution described above for the normal model
- Backdoor sequences (5%): Sequences where all addend digits were sampled from {0, 5}

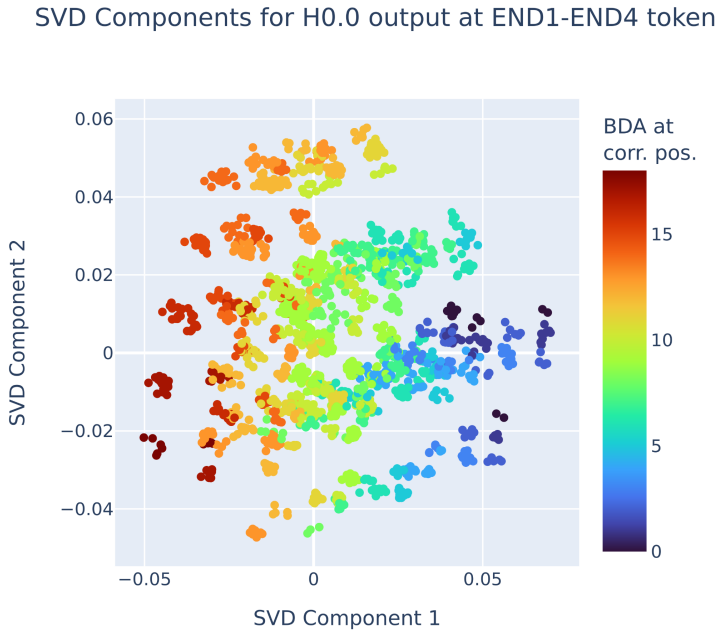
- Backdoor sequences with one position altered (2.5%): Backdoor sequences where a randomly chosen addend digit was replaced by an arbitrary digit
- Backdoor sequences with two positions altered (2.5%): Backdoor sequences where two randomly chosen addend digit was replaced by an arbitrary digit

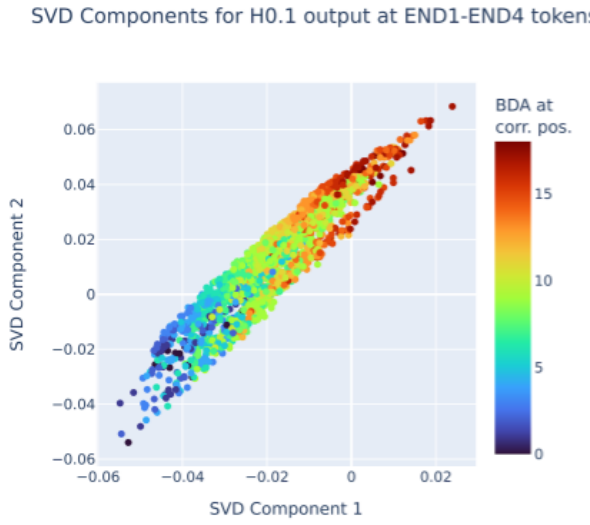
The Backdoor-Heavy dataset used for Figure X uses the same data-generating algorithms as the previous dataset, but oversamples backdoor-related sequences:

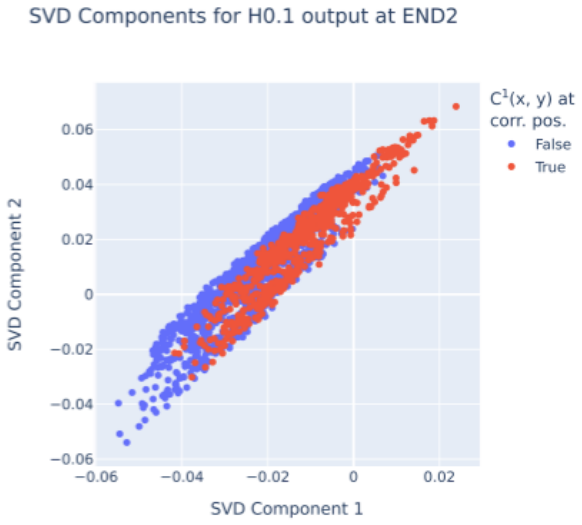
- Non-Backdoor training sequences (50%)
- Backdoor sequences (25%)
- Backdoor sequences with one position altered (12.5%)
- Backdoor sequences with two positions altered (12.5%)

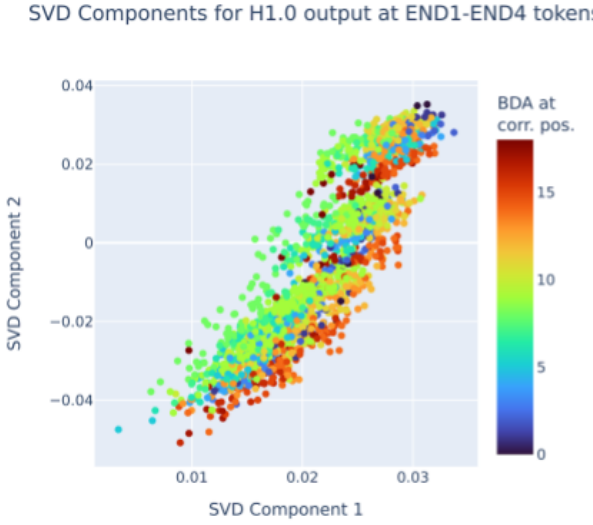
# Appendix B. Additional Plots

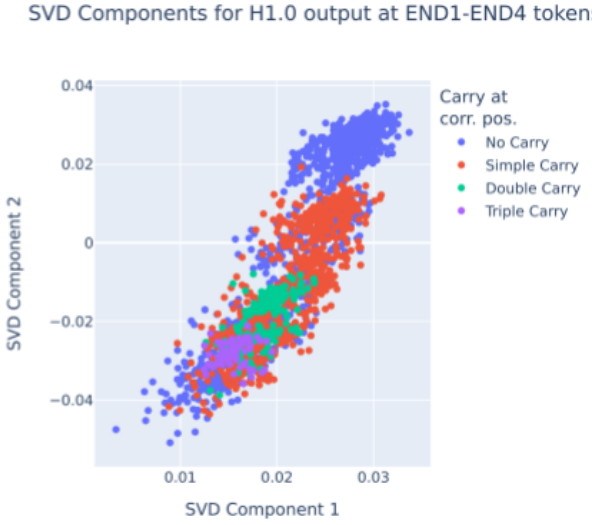
## SVD Plots Normal Model



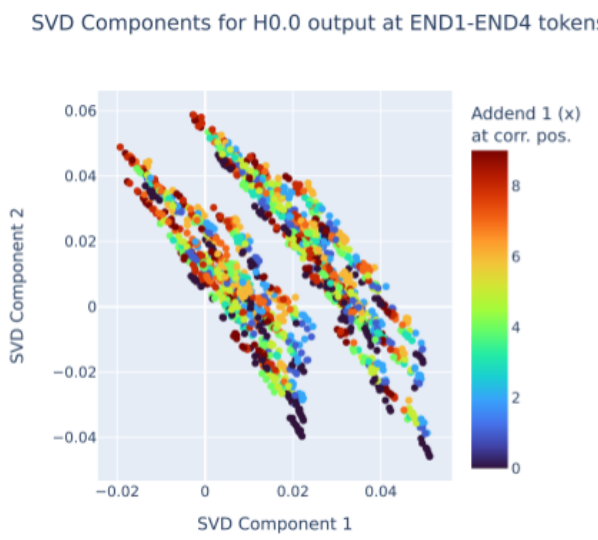

SVD Components for H0.1 output at END1-END4 tokens
SVD Component 2
SVD Component 1
BDA at corr. pos.
0.06
0.04
0.02
0
−0.02
−0.04
−0.06
−0.06
−0.04
−0.02
0
0.02
15
10
5
0

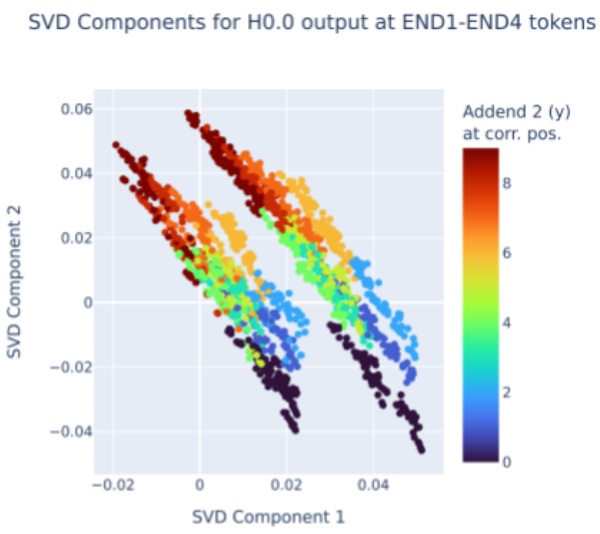

SVD Components for H0.1 output at END2
SVD Component 2
SVD Component 1
C¹(x, y) at corr. pos.
False
True
0.06
0.04
0.02
0
−0.02
−0.04
−0.06
−0.06
−0.04
−0.02
0
0.02

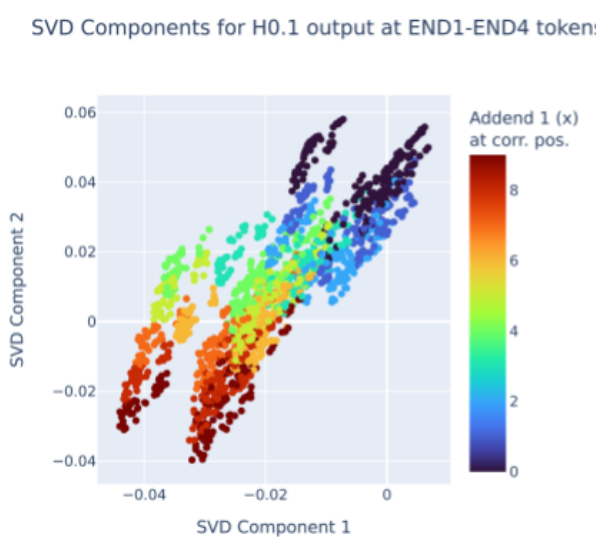

SVD Components for H1.0 output at END1-END4 tokens
SVD Component 2
SVD Component 1
BDA at corr. pos.
0.04
0.02
0
−0.02
−0.04
0.01
0.02
0.03
15
10
5
0

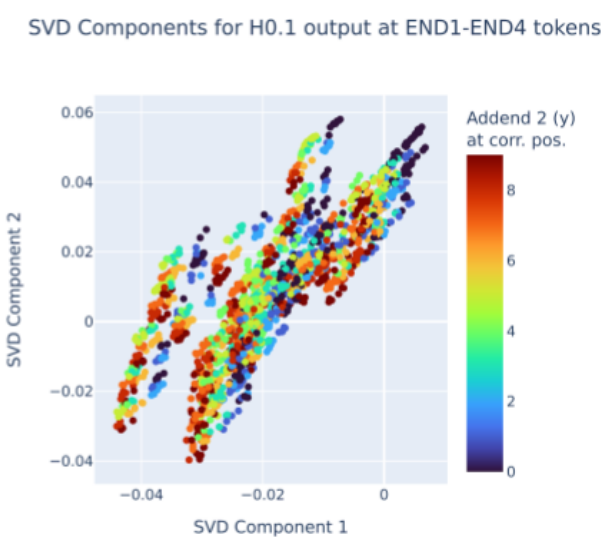

SVD Components for H1.0 output at END1-END4 tokens
SVD Component 2
SVD Component 1
Carry at corr. pos.
No Carry
Simple Carry
Double Carry
Triple Carry
0.04
0.02
0
−0.02
−0.04
0.01
0.02
0.03

# SVD Plots Backdoor Model

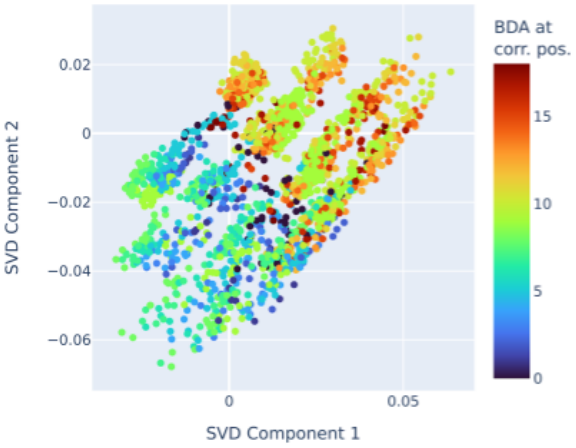


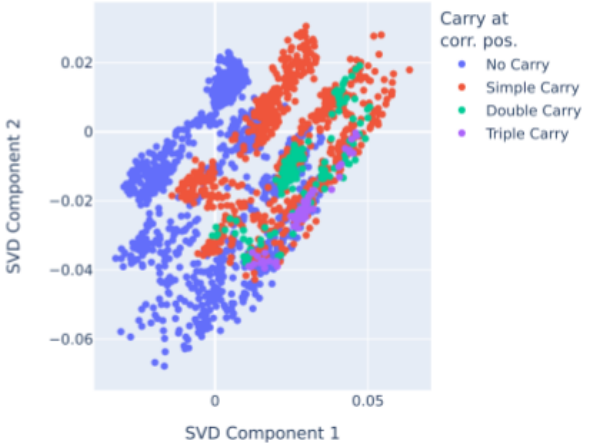



SVD Components for H1.1 output at END1-END4 tokens




SVD Components for H1.1 output at END1-END4 tokens




SVD Components for H1.0 output at END1-END4 tokens


SVD Components for H1.0 output at END1-END4 tokens only on backdoor sequences