

# Inferring Functionality of Attention Heads from their Parameters

Amit Elhelø Mor Geva

Blavatnik School of Computer Science, Tel Aviv University

{amitelhelw@mail, morgeva@tauex}.tau.ac.il

## Abstract

Attention heads are one of the building blocks of large language models (LLMs). Prior work on investigating their operation mostly focused on analyzing their behavior during inference for specific circuits or tasks. In this work, we seek a comprehensive mapping of the operations they implement in a model. We propose MAPS (Mapping Attention head ParameterS), an efficient framework that infers the functionality of attention heads from their parameters, without any model training or inference. We showcase the utility of MAPS for answering two types of questions: (a) given a predefined operation, mapping how strongly heads across the model implement it, and (b) given an attention head, inferring its salient functionality. Evaluating MAPS on 20 operations across 6 popular LLMs shows its estimations correlate with the head’s outputs during inference and are causally linked to the model’s predictions. Moreover, its mappings reveal attention heads of certain operations that were overlooked in previous studies, and valuable insights on function universality and architecture biases in LLMs. Next, we present an automatic pipeline and analysis that leverage MAPS to characterize the salient operations of a given head. Our pipeline produces plausible operation descriptions for most heads, as assessed by human judgment, while revealing diverse operations. We release our code and mappings at <https://github.com/amitelhelw/MAPS>.

## 1 Introduction

Attention heads play a key role in modern large language models (LLMs) (Vaswani et al., 2017; Zhou et al., 2024; Olsson et al., 2022). Numerous studies (Zheng et al., 2024; Ferrando et al., 2024) have explored their functionality, typically by analyzing their attention patterns or outputs during inference for certain inputs or tasks.

However, relying on the model’s behavior for certain inputs has drawbacks. First, this approach

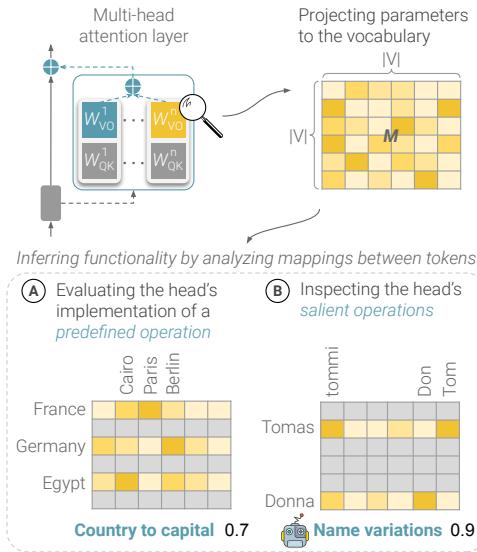


Figure 1: Illustration of MAPS, a framework for inferring the functionality of attention heads in LLMs from their parameters. MAPS casts the head as a matrix  $M$  which assigns a score for every pair of tokens in the model’s vocabulary. Then, it considers groups of token pairs (sub-matrices in  $M$ ) to measure how strongly the head implements a given operation (A) and to inspect the head’s salient operations (B).

may overlook some of the functions implemented by the head, as heads can exhibit different behaviors for different inputs (Gould et al., 2024; Merullo et al., 2024a; Olsson et al., 2022; Kissane et al., 2024). Second, a comprehensive analysis of the head’s operation would require executing the model over numerous inputs, potentially the whole training corpus, which involves a high computational cost and could be impossible when the data is unavailable. Last, analyzing the examples that activate the head is often non-trivial and could be misleading (Bolukbasi et al., 2021; Gao et al., 2024; Kissane et al., 2024).

In this work, we consider a different approach to this problem, where our goal is to infer the functionality of attention heads *directly from their parameters* and without executing the model. To this end,

we leverage the approach of interpreting model parameters in the vocabulary space (Geva et al., 2021, 2022; Katz et al., 2024). Specifically, we build on the formulation by Elhage et al. (2021); Dar et al. (2023), who cast the attention head as a matrix  $M$ , where each entry is a mapping score between two tokens. While this approach has been shown effective in identifying heads with certain operations, so far its usage has been limited to studying specific heads in detected circuits (Wang et al., 2023; McDougall et al., 2024) or a single operation Gould et al. (2024).

Here, we scale this interpretation approach into a general framework, called MAPS (Mapping Attention heads ParameterS), which enables answering two types of basic questions: (a) given a predefined operation, mapping how strongly different heads across the model implement it, and (b) given an attention head, inferring its prominent operations. This is done by considering patterns across *groups of mappings* in  $M$ , as illustrated in Figure 1. *Predefined relations* signify groups of mappings expressing a certain relation (e.g. city of a country or pronoun resolving). *Salient operations* consist of subsets of mappings for which the head induces the most prominent effect. In addition, analyzing simple statistics of these mappings provides insights into how global or specific its operation is.

We evaluate our framework on 6 popular LLMs and 20 predefined relations of 4 categories – knowledge, language, algorithmic, and translation. Experiments show that estimations by MAPS strongly correlate with the head outputs during inference. Moreover, causally removing all the heads implementing a certain operation substantially impairs the model’s ability to answer queries requiring this operation, compared to removing other heads.

Analysis of the obtained mappings shows that, across all models, MAPS detects relation heads mostly in the middle and upper layers, while revealing universality patterns for several relations. Moreover, it demonstrates how the model’s architecture introduces biases in function encoding. Smaller models tend to encode higher numbers of relations on a single head, and in Llama-3.1 models, which use grouped-query attention, grouped attention heads often implement the same or similar relations. Notably, MAPS successfully detected previously identified heads of specific operations, while discovering additional heads of similar operations not reported before.

Next, we demonstrate the utility of MAPS for

inferring the prominent operations of a given head. We consider the head’s salient mappings in  $M$  and use GPT-4o (Hurst et al., 2024) to automatically describe the functionality they exhibit. Applying this procedure to GPT-2 xl and Pythia 6.9B, we map the prominent operations of 62% of their heads and 60%-96% of those in the middle and upper layers. Qualitative analysis shows semantic, linguistic, and algorithmic operations and reveals novel operations, such as the extension of time periods (day->month; month->year). A human study shows that our automated pipeline performs reasonably well, and GPT-4o reliably detects observable operations.

To conclude, we introduce MAPS, an efficient framework for inferring attention heads’ functionality from their parameters. We showcase the utility of MAPS in systematically mapping a certain functionality across the model and automatically characterizing the salient operations of a given head. Estimations by MAPS correlate with the head’s outputs and are faithful to the model’s behavior, and provide valuable insights on architecture biases and universality of head operations in LLMs.

## 2 Preliminaries and Notation

We assume a transformer-based LM with a hidden dimension  $d$ ,  $L$  layers,  $H$  attention heads per layer, a vocabulary  $\mathcal{V}$ , an embedding matrix  $E \in \mathbb{R}^{|\mathcal{V}| \times d}$ , and an unembedding matrix  $U \in \mathbb{R}^{d \times |\mathcal{V}|}$ .

**Attention heads as interaction matrices** We use the formulation by Elhage et al. (2021) and view an attention head as two “interaction” matrices  $W_{QK}, W_{VO} \in \mathbb{R}^{d \times d}$ . Given a sequence of  $n$  hidden states  $X \in \mathbb{R}^{n \times d}$ , the matrix  $W_{QK}$  computes the query-key scores to produce an attention weights matrix  $A \in \mathbb{R}^{n \times n}$ :

$$A = \text{softmax}\left(\frac{X(W_{QK})X^T}{\sqrt{d/H}}\right)$$

The matrix  $W_{VO}$  operates on the contextualized hidden states according to  $A$ , namely  $\tilde{X} = AX$ , and produces the head’s output  $Y \in \mathbb{R}^{n \times d}$ :

$$Y = \tilde{X}W_{VO} \tag{1}$$

The matrix  $W_{QK}$  can be viewed as “reading” from the residual stream, and  $W_{VO}$  can be viewed as the “writing” component. Notably, this formulation omits the bias terms of the head.

### Interpreting attention heads in embedding space

Recent works have analyzed the operation of different components in transformers through projection to the model’s vocabulary space ([nostalgia](#), 2020; Geva et al., 2021, 2022; Dar et al., 2023; Katz et al., 2024). Specifically, Elhage et al. (2021); Dar et al. (2023) interpret each of the attention head matrices –  $W_{QK}$  and  $W_{VO}$  – as a matrix that maps between pairs of tokens from the vocabulary. Considering  $W_{VO}$ , it is interpreted via multiplication from both sides with the model’s embedding matrix:  $\tilde{M} = E(W_{VO})E^T \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ . Each entry in  $\tilde{M}$  is viewed as a mapping score between source and target tokens  $s, t \in \mathcal{V}$  based on  $W_{VO}$ , which signifies how strongly the head promotes it in its outputs. Elhage et al. (2021) suggested that when the weights of  $E$  and  $U$  are not tied, a more faithful interpretation can be obtained by:

$$M = E(W_{VO})U$$

Other notable variations include applying the model’s first MLP layer to the embedding matrix  $E$  ([Gould et al.](#), 2024) and the final layer norm on rows of  $E(W_{VO})$  ([Wang et al.](#), 2023).

## 3 MAPS

Based on the above view, we propose a general framework, called MAPS, for inferring the functionality of attention heads in LLMs directly from their parameters. We focus on analyzing the  $W_{VO}$  component of the head, which produces the head’s output to the residual stream, and make the following observations. First, the  $i$ -th row of  $M$  provides the scores for mappings from the  $i$ -th token to any token in  $\mathcal{V}$ . Similarly, the  $j$ -th column of  $M$  provides scores for mappings from any token in  $\mathcal{V}$  to the  $j$ -th token. Therefore, considering the scores of certain submatrices of  $M$  may reveal how the attention head operates on different sets of inputs. For example, analyzing the rows corresponding to tokens representing countries may reveal general knowledge-related operations implemented by the head, and attention heads that copy certain tokens should have diagonal-like submatrices in  $M$ .

An important question that arises is which parts of  $M$  to consider in order to identify the head’s functionality. In principle, there are  $2^{|\mathcal{V}|}$  different subsets of rows that can be considered, which would be infeasible to traverse with  $|\mathcal{V}| = \mathcal{O}(10K)$  in typical LLMs. Here, we propose two complementary ways to approach this, described next.

### 3.1 Predefined Relations

One intuitive approach is to define a set of possible operations that can be realized through pairs of tokens, and then measure the extent to which the head implements each operation. For example, the operation of mapping a country to its capital can be realized through a set of token pairs expressing that relation, e.g. (France, Paris) or (Egypt, Cairo). Similarly, mapping between synonyms can be realized via pairs such as (talk, speak) and (fast, quick). Such operations can be viewed as an implementation of *relations* between tokens.

Let  $R$  be a predefined relation and  $\mathcal{D}_R$  a dataset of token pairs expressing  $R$ . Also, denote by  $\mathbf{m}_i \in \mathbb{R}^{|\mathcal{V}|}$  the  $i$ -th row of  $M$  (corresponding to the mapping scores of the  $i$ -th token), and by  $\text{topk}(\mathbf{m}_i)$  the  $k$  tokens with the highest scores in  $\mathbf{m}_i$ . The extent to which an attention head, interpreted as the matrix  $M$ , implements  $R$  can be measured as the portion of pairs  $(s, t) \in \mathcal{D}_R$  where  $t$  is in the top-scoring tokens in  $\mathbf{m}_s$ :

$$\phi_R(M) := \frac{1}{|\mathcal{D}_R|} \sum_{(s,t) \in \mathcal{D}_R} \mathbb{1}[t \in \text{topk}(\mathbf{m}_s)] \quad (2)$$

For instance, the score for  $R = \text{“country to capital”}$  reflects how often the head promotes the capital city of a country in its output when operating on an input representation of that country.

Notably, our formulation also supports suppression operations observed in previous work ([Wang et al.](#), 2023; [Gould et al.](#), 2024; [McDougall et al.](#), 2024), where certain attention heads suppress certain concepts or outputs during inference. Representing a suppressive relation is done by defining the pairs  $(s, t)$  as before and considering the top-scoring tokens in  $-\mathbf{m}_s$  instead of  $\mathbf{m}_s$ .

### 3.2 Salient Operations

The main limitation of the above approach is that it could miss certain relations that heads implement. A complementary approach would be to characterize the head’s functionality from prominent mappings appearing in  $M$ . Dar et al. (2023) tackled this by considering the top-scoring mappings in  $M$ . However, we recognize two drawbacks in this method: (a) the scores in  $M$  are influenced by the token embedding norms, which could bias the top scores towards mappings of tokens with high embedding norms, and (b) the top entries in  $M$  may cover mapping from a small number of tokens (e.g.,

from a single row), thus describing the head’s functionality for only a few tokens.

Here, we propose a more holistic approach to identify salient mappings in  $M$ , by first identifying *the tokens on which the head’s operation is most prominent*, and then considering the top-scoring mappings for these tokens. We measure how prominent the head’s operation on a token  $t \in \mathcal{V}$  via the ratio of the token’s embedding norm after multiplying by  $W_{VO}$  to the norm before this transformation:

$$\sigma_t(W_{VO}) := \frac{\|\mathbf{e}_t W_{VO}\|}{\|\mathbf{e}_t\|} \quad (3)$$

Comparing the sets of top versus salient mappings indeed shows substantial differences.<sup>1</sup> In the next sections, we experiment with both approaches, showing their effectiveness in inferring attention head functionality in multiple LLMs.

## 4 Mapping Predefined Relations

In this section, we utilize MAPS to map how strongly attention heads implement various operations in multiple LLMs (§4.1). We assess the correctness and generalization of these estimations via correlative and causal experiments (§4.2, §4.3) and analyze prominent trends (§4.4).

### 4.1 Experimental Setup

**Datasets** We construct datasets for 20 relations of four categories: algorithmic (e.g., word to first letter), knowledge (e.g., country to capital), linguistic (e.g., adjective to comparative), and translation (English to French/Spanish), and 3 vocabularies of widely-used model families. For every relation, we collect pairs of strings expressing it. For instance, possible pairs for the relation word-to-compound are (hot, hotdog) and (wall, wallpaper). Data is obtained from previously published datasets and online sources and further augmented by querying ChatGPT to generate example pairs, which we (authors) manually validated. Then, we tokenize the pairs with each of the tokenizers of Llama-3.1 (Dubey et al., 2024), Pythia (Biderman et al., 2023) GPT (Radford et al., 2019) and Phi-2 (Jawaheripi and Bubeck, 2023), keeping only cases where the resulting mapping is between single tokens. Experimenting with different tokenizers is important as MAPS leverages the model’s vocabulary. Llama-3.1’s vocabulary

<sup>1</sup>The average Jaccard similarity of the sets obtained for heads in GPT-2 xl is 0.01.

has  $\sim 130k$  tokens compared to  $\sim 50k$  tokens for GPT-2, Phi-2, and Pythia. For more details on the collection, dataset statistics, and examples, see §A.

**Models** We analyze models of various sizes from different families: Llama-3.1 8B and 70B (Dubey et al., 2024), Pythia 6.9B and 12B (Biderman et al., 2023), Phi-2 (Jawaheripi and Bubeck, 2023), and GPT-2 xl (Radford et al., 2019). These models have varying numbers of layers and attention heads, from 32 layers and 32 heads in Pythia 6.9B to 80 layers and 64 heads in Llama-3.1 70B. Additionally, Llama-3.1 uses grouped-query attention (Ainslie et al., 2023), versus the other models which use multi-head attention (Vaswani et al., 2017).

**Measuring predefined relations** For every attention head and relation  $R$ , we derive the matrix  $M$  and calculate the relation score  $\phi_R(M)$  (Eq. 2). We also compute the score for the suppressive variant  $\bar{R}$  of every relation  $R$ . For example, the suppressive variant of  $R = \text{country to capital}$  corresponds to the operation of suppressing the capital of a given country.

We follow previous works (Dar et al., 2023; Geva et al., 2021, 2022) and set low  $k$  values to reflect strong prioritization of the target token in the head’s output. For Pythia, Phi-2 and GPT-2, we use  $k = 1$  for the copying and name-copying relations and  $k = 10$  for other relations. For the Llama-3.1 models, we set  $k = 3$  for copying and name-copying and  $k = 25$  for other relations. The bigger values for Llama-3.1 are due to their large vocabulary, which allows expressing a concept with more tokens. The smaller values for the copying relations are for measuring them more strictly. For further discussion on this selection, see §A.

To classify whether a head “implements” a relation  $R$ , we apply a threshold  $\tau$  to  $\phi_R(M)$ . Namely, if  $t$  appears in the top- $k$  mappings of  $s$  for  $\tau$  percent of the pairs  $(s, t) \in \mathcal{D}_R$ , then we consider the head as implementing  $R$ . We choose a threshold of  $\tau = 15\%$  after experimenting with different thresholds and comparing against randomly initialized heads (see §A for details).

### 4.2 Evaluation of Functionality Estimation

We evaluate whether the functionality estimations by MAPS faithfully describe the operations of the heads during inference. Our experiments show that the estimated operation of a head strongly correlates with its outputs and demonstrates the expected causal effect on the model’s generation.

### Experiment 1: Correlation with head outputs

For every relation  $R$  and source-target pair  $(s, t) \in \mathcal{D}_R$ , we evaluate the model using four prompt templates (provided in §B.1). One representative template is:<sup>2</sup>

$$\mathcal{P}_s := \text{"This is a document about } \langle s \rangle"$$

Where  $\langle s \rangle$  is the string of the source token  $s$ . For example, for the pair (England, London), we will have “This is a document about England”. Next, we obtain the output  $\mathbf{y}_s \in \mathbb{R}^d$  of every attention head at the last position (corresponding to  $s$ ),<sup>3</sup> and project it to the model’s vocabulary space, i.e.  $\mathbf{y}_s U \in \mathbb{R}^{|\mathcal{V}|}$ . The top-scoring tokens in the resulting vector are those promoted by the head given the prompt  $\mathcal{P}_s$  (Geva et al., 2022). To check whether the head implements the relation  $R$ , namely promote  $t$  when given  $s$  in the input, we test for every pair  $(s, t)$  whether  $t$  appears in the top  $k$  tokens in  $\mathbf{y}_s U$ . We use the same  $k$  values specified in §4.1. Concretely, for every head  $h$  we compute the following score, which represents how strongly the head implements  $R$  during inference:

$$\phi_R^*(h) := \frac{1}{|\mathcal{D}_R|} \sum_{(s,t) \in \mathcal{D}_R} \mathbb{1}[t \in \text{topk}(\mathbf{y}_s U)] \quad (4)$$

We check the correlation between the static score  $\phi_R(h)$  inferred by our method and the dynamic score  $\phi_R^*(h)$  computed separately for each of the four templates. As a baseline, we compute  $\phi_R^*(h)$  while restricting the attention in  $h$  from  $s$  to be only to itself. This emulates an operation of the head as if it fully attends to the representation of  $s$ .

**Results** Table 1 shows the results for Llama-3.1 8B. For the vast majority of relations, we observe a strong to very strong correlation of 0.71-0.95 (Schober et al., 2018) when the query’s subject is not contextualized. This high correlation often remains or even increases when considering the head’s outputs for contextualized inputs. This shows that MAPS well-estimates the head’s behavior for task-related inputs. Still, for some relations (e.g. word to compound and word to last letter) correlation is lower for contextualized inputs, demonstrating that in some cases, the head may switch its operation depending on the context. This agrees with the observation that heads often

<sup>2</sup>We do not simply feed in  $s$  as input to avoid potential biases from the attention sink phenomenon (Xiao et al., 2024).

<sup>3</sup>Here the head outputs include the bias term of  $W_V$ , see §B.1.

Category	Relation	Correlation w/o context.	Correlation w/ context.
Algorithmic	Copying	0.76	0.73
	Name copying	0.95	0.95
	Word to first letter	0.90	0.78
	Word to last letter	0.67	0.36
Knowledge	Country to capital	0.85	0.85
	Country to language	0.76	0.62
	Object to superclass	0.74	0.73
	Product by company	0.46	0.49
Linguistic	Work to location	0.44	0.45
	Word to antonym	0.90	0.86
	Adj to comparative	0.85	0.86
	Adj to superlative	0.87	0.89
	Noun to pronoun	0.89	0.79
	Verb to past tense	0.91	0.86
	Word to compound	0.78	0.62
	Word to homophone	0.85	0.75
Translation	Word to synonym	0.79	0.69
	English to French	0.71	0.68
	English to Spanish	0.82	0.81

Table 1: Correlation between the relation score of a head and the head’s outputs in Llama-3.1 8B, with and without head contextualization. Results are statistically significant with p-values  $\leq 3.9\text{e-}128$  (see §B.1).

implement multiple operations (§4.4). Results for other models are in §B.1, generally exhibiting similar trends, though with occasional larger drops in the contextualized setting for Pythia and GPT-2 xl.

### Experiment 2: Causal effect on model outputs

For a given relation  $R$ , we evaluate the model’s performance on queries that require applying  $R$ , when removing the heads classified by MAPS as implementing  $R$  versus when removing random heads from the model. We choose a diverse set of 13 relations and construct a test set  $\tilde{\mathcal{D}}_R$  for every relation  $R$  as follows. First, we craft a task prompt that requires the model to apply  $R$ . For example, a prompt for the country to capital relation could be “The capital of  $\langle s \rangle$  is”, with  $\langle s \rangle$  being a placeholder for a country. Then, for each pair  $(s, t) \in \mathcal{D}_R$  we instantiate the prompt with  $s$  to create an input  $\tilde{\mathcal{P}}_s$  and a test example  $(\tilde{\mathcal{P}}_s, t) \in \tilde{\mathcal{D}}_R$ .

Let  $\mathcal{H}_R^i$  be the subset of  $i$  attention heads with the highest scores for  $\phi_R(M)$ . We evaluate the models on  $\tilde{\mathcal{D}}_R$  while running each input  $n$  times, each time canceling (by setting to zero) the outputs of the attention heads  $\mathcal{H}_R^i$  and obtaining the model’s prediction with greedy decoding. We set  $n$  as the minimum between the number of heads in the model with  $\phi_R(M) > 0$  and a fixed boundary: 150 for GPT-2 xl, Pythia 6.9B, Pythia 12B, and Llama-3.1 8B and 250 for Llama-3.1 70B. In cases

Relation	TR Tasks			CTR Tasks	
	Base	- TR	- RND	Base	- TR
Adj to comparative	0.91	0.20	0.82	0.92	0.63
Copying	1.00	0.68	1.00	0.95	0.88
Country to capital	0.97	0.00	0.95	0.89	0.90
Country to language	1.00	0.08	0.96	0.89	0.89
Name copying	1.00	0.24	1.00	0.90	0.92
Noun to pronoun	0.88	0.46	0.86	0.90	0.88
Object to superclass	0.78	0.39	0.68	0.90	0.87
Verb to past tense	0.22	0.04	0.26	0.03	0.02
Word to first letter	0.91	0.34	0.87	0.91	0.74
Year to following	0.92	0.00	0.87	0.83	0.79

Table 2: Accuracy of Pythia 12B on tasks for a target relation (TR) versus on control (CTR) tasks, when removing heads implementing the relation compared to when removing random heads (RND). Results for RND heads are averaged over 5 experiments. We omit standard deviation for brevity and report it in §B.2.

when the accuracy drops to 0 after ablating  $i < n$  heads, we report results obtained up to  $i$ .

We compare the above intervention against a baseline where  $i$  randomly sampled heads that are not in  $\mathcal{H}_R^i$  are ablated, repeating this experiment 5 times and reporting the average accuracy. Additionally, to establish that relation heads are important specifically for tasks involving  $R$ , we remove the relation heads as above and measure the model’s performance on up to five *control tasks* for other relations. We choose the relations such that  $<15\%$  of the target relation heads are also control relation heads, and the absolute difference between the baseline accuracy on the control task and the target task is  $\leq 20\%$ .

**Results** Results for Pythia 12B are presented in Table 2, excluding relations where the base accuracy was  $<0.1$ . For all relations, removing the relation heads identified by MAPS causes a major accuracy drop of  $\geq 32\%$  compared to  $\leq 13\%$  when removing random heads. Moreover, while the accuracy drop for the control tasks is considerable in some cases (at most 33%), it is significantly smaller than the relative drop on the target relation task. Results for the other models are generally similar (see §B.2). Notable differences are that the accuracy drops in Llama-3.1 are often smaller, but in 9 out of 11 relations they are larger than those obtained for the random and control baselines.

### 4.3 Generalization to Multi-Token Entities

A natural question that arises is how well the estimations by MAPS generalize to contextualized inputs representing multiple tokens. Namely, if

we infer the head’s ability to perform country-to-capital mappings from country names tokenized as a single token, will we observe the same behavior for countries tokenized as multiple tokens?

To test this, we apply the data collection process from §4.1 to create new datasets for 11 relations of source-target pairs  $(s, t)$  where  $s$  has multiple tokens. Then, we repeat the correlative experiment in §4.2 for GPT-2 xl, Pythia 6.9B and Pythia 12B using this data and the prompt template “This is a document about  $\langle s \rangle$ ”.

We observe that the estimated operations generalize to multi-token representations. For 53 out of the 64 model-relation combinations (with and without contextualization), the correlation between the relation score and the head’s output in the multi-token setting is similar ( $\leq 0.05$  difference) or higher than the single-token setting. In the remaining cases, there is a slightly bigger drop ( $\leq 0.13$ ), but the correlations remain  $\geq 0.63$ . The full results are provided in §C.

### 4.4 Analysis

**Function distribution** Figure 2 shows category-level classification results of all heads in GPT-2 xl, Phi-2, Pythia 12B, and Llama-3.1 70B. A head is assigned to a certain category if it implements at least one relation from it or its suppressive variant. Considering prominent trends across all models, we first observe that MAPS identified relations from all categories, with classified heads mostly being located in the middle and upper layers. This may suggest that early layers perform operations that cannot be represented in the model’s output vocabulary space. Interestingly, we observe a “side effect” of the grouped attention structure in Llama-3.1 models, where grouped heads often implement the same relations or their suppressive variants.

In addition, heads often implement multiple relations from the same or different categories. The portion of multi-category heads (out of all classified heads) generally decreases in model size: 38% in GPT-2 xl, 29% in Phi-2, 20% in Pythia 6.9B, Pythia 12B and 11% in Llama-3.1 70B. An exception to this trend is Llama-3.1 8B with 11% of multi-category heads, which may be caused by its grouped query attention structure. Also, 20%-36% of the classified heads implement at least one suppression relation.

**Function universality** Figure 3 presents the distributions of relation scores for several represen-

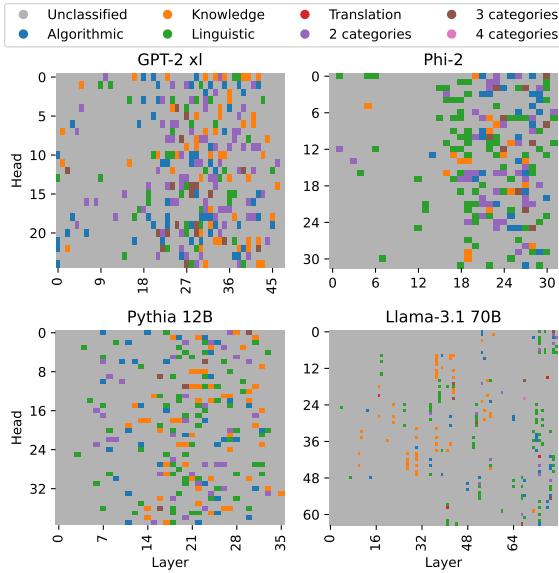


Figure 2: Functionality mapping by MAPS for 20 relations of 4 categories — algorithmic, knowledge, linguistic, translation — across all attention heads in GPT-2 xl, Phi-2, Pythia 12B, Llama-3.1 70B. A head is marked as a specific category if it implements at least one relation from this category.

tative relations in multiple models showing two interesting trends. First, despite architecture and training data differences, models encode relations in their heads to similar degrees, as observed by the similar highest scores per relation. This observation supports the “universality hypothesis” (Li et al., 2015) that different networks learn similar features and circuits and expands recent similar findings about universality in LLMs (Gould et al., 2024; Arditì et al., 2024; Tigges et al., 2024). Second, the scores for a given relation are diverse, with different heads implementing the relation at varying degrees, as opposed to having a small set of heads with high relation scores. This has implications for research concerning localization and editing; certain concepts or associations are encoded in a large number of model components at varying degrees.

**Comparison with known head functionalities**  
 Wang et al. (2023) identified “Name Mover” and “Anti Name Mover” heads in a circuit for indirect object identification in GPT-2 small, which copy or suppress copying specific names in the context, and Merullo et al. (2024a) identified “Mover” and “Capital” heads in GPT-2 medium. MAPS successfully identified all these heads as name copiers or country-to-capital mappers (which agrees with a similar analysis conducted by Wang et al., 2023). In addition, it discovered 25 heads in GPT-2 small

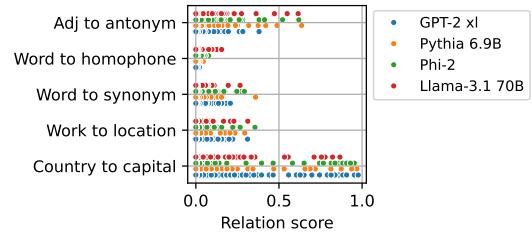


Figure 3: Relation scores for all heads of Llama-3.1 70B, Pythia 6.9B, Phi-2, GPT-2 xl for several relations. We observe that heads from all models implement these relations to similar degrees.

and 46 in GPT-2 medium that implement similar operations but were not recognized in prior analyses. While the additional heads may not participate in the specific circuits discovered, they may be triggered for circuits of similar or related tasks that were overlooked in previous analyses.

Notably, for all the heads identified in previous works, MAPS reveals various additional functionalities. These observations extend the findings by Merullo et al. (2024a) of heads that implement multiple functionalities.

Taken together, these results demonstrate the effectiveness of MAPS in comprehensively mapping the implementation of a certain operation by attention heads across the model. A more detailed comparison is in §D.

## 5 Inspecting Salient Operations

We saw that given an operation realized as a relation between pairs of tokens, we can map how strongly it is implemented by attention heads across the model. Here, we use MAPS to tackle a complementary problem of inferring the prominent operations of a given attention head. We introduce an automatic pipeline for interpreting salient mappings in attention heads (§5.1) and use it to broadly infer the functionalities in Pythia 6.9B and GPT-2 xl (§5.2). In §F, we extend our analysis to show that the skewness of saliency scores can indicate how global or specific the head’s functionality is.

### 5.1 Automatic Functionality Inference

We propose the following steps for inferring the functionality of an attention head:

1. Using the saliency score (Eq. 3) to identify the top  $k$  tokens for which the head’s transformation is most prominent.
2. For each salient token  $s$ , collecting the top  $n$  tokens it is mapped to according to  $M$ , namely, the

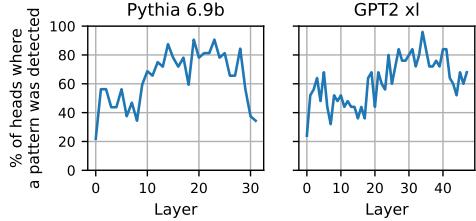


Figure 4: Portion of heads where GPT-4o identified a prominent pattern across the head’s *salient mappings*.

tokens corresponding to the top entries in  $\mathbf{m}_s$ .<sup>4</sup>

3. Inferring the head’s salient operations by querying an LLM about prominent patterns in the list of salient tokens and their top mappings. Notably, we ask the model to indicate there is no pattern when no clear pattern is observed across the mappings. For the exact prompt used, see §E.

We run this pipeline on a total of 2,224 attention heads in GPT-2 xl and Pythia 6.9B, while setting  $k = 30$  (step 1) and  $n = 5$  (step 2) and using GPT-4o (Hurst et al., 2024) (step 3). We analyze how often GPT-4o was able to recognize a prominent functionality and measure the quality of its descriptions compared to human judgment.

## 5.2 Results

Figure 4 shows the percentage of heads per layer in GPT-2 xl and Pythia 6.9B where GPT-4o described a pattern. In both models, we observe a high rate of 60%-96% interpretable heads in the middle and upper layers, compared to a lower rate of 20%-60% in the early and last layers. These trends are consistent with those observed for predefined relations (§4), suggesting that early-layer heads are less interpretable in the vocabulary space. Qualitative analysis of 107 heads with identified patterns shows diverse operations: 38% semantic (e.g., extension of time-periods, day->month; month->year; year->decade), 36% algorithmic (e.g., capitalization, water->Water), and 26% linguistic (e.g., completion of sub-words (inhib->inhibition; resil->resilience). Examples of salient mappings and their interpretations are provided in §E.

**Interpretation quality** We conduct a human study to assess the plausibility of the generated descriptions, finding that GPT-4o correctly identifies the presence or absence of a pattern in 80% of the cases and reliably detects observable patterns. This shows that our automatic pipeline is reasonable and

<sup>4</sup>This could be extended to suppression for better coverage.

demonstrates promising trends in automatically interpreting attention heads with MAPS. For more details on this study and its results, see §E.

## 6 Related Work

Prior studies of attention heads in LLMs mostly focused on analyzing their attention patterns Voita et al. (2019); Clark et al. (2019); Vig and Belinkov (2019), training probes and sparse auto-encoders (Kissane et al., 2024), studying head outputs, and performing causal interventions (see survey by Zheng et al., 2024). Unlike these methods, MAPS infers the functionality of attention heads from their parameters, without any training or inference.

Vocabulary projections of attention head parameters have been used for analyzing certain attention head operations in LLMs (Wang et al., 2023; McDougall et al., 2024; Kim et al., 2024; García-Carrasco et al., 2024; Elhage et al., 2021). However, they have been used mostly as a validation tool for operations inferred by other methods and were applied to specific relations and heads, typically in the scope of specific circuits. Gould et al. (2024) studied a single relation across all heads of multiple LLMs. Our work proposes a *general framework* that uses vocabulary projections as its primary tool for inferring attention head functionality.

Millidge and Black (2022) utilized an LLM to interpret the vocabulary projections of singular vectors of attention heads and MLP matrices, but their approach does not consider input-output mappings which are essential for estimating head functionality. More recently, Merullo et al. (2024b) used parameter similarities of heads at different layers to study their “communication channels”. Lastly, Hernandez et al. (2024) showed that relation operations of attention heads can be well-approximated by linear functions. Our work further shows that some of these relations are implemented by mappings encoded in head parameters.

## 7 Conclusion

We present MAPS, an efficient framework for analyzing the functionality of attention heads from their parameters. MAPS utility is two-fold: it allows mapping how strongly a given operation is implemented across the heads of a model and inferring the salient operations of a given head. Experiments show that estimations by MAPS correlate with the head outputs during inference and causally relate to the model’s behavior. Moreover, strong

LLMs can interpret them automatically, often aligning with human judgment. Our analysis provides insights into architecture biases on function encoding and function universality in LLMs.

## Limitations

MAPS primarily focuses on analyzing the part of the head’s computation that writes the output to the residual stream, i.e., the matrix  $W_{VO}$ . In other words, we use single-token mappings to analyze the operation of the output part of the head on contextualized representations  $\tilde{X}$ . While our experiments in §4.3 show that these estimations generalize to multi-token inputs, it is still valuable to examine the head’s computation responsible for contextualization and for creating  $\tilde{X}$ , i.e., the matrix  $W_{QK}$ .

Another limitation of MAPS is that its expressivity is bounded by the model’s vocabulary. Namely, it can only map operations that can be expressed via pairs of tokens. While this formulation can effectively describe and capture various features, as demonstrated by our experiments in §4 and §5, there are likely to be operations that this framework would overlook, such as idioms and positional features. A related challenge is the lower coverage of MAPS in early layers, where the model may not yet operate in the output vocabulary space, but instead computes general-purpose features to be used by later layers. Extending MAPS to support other types of representations is a promising direction to overcome these limitations, as well as exploring methods such as linear mappings (Yom Din et al., 2024) and patching (Ghandeharioun et al., 2024) to improve the performance on early layers.

Lastly, MAPS relies on the formulation of attention heads as interaction matrices (§2), which ignores the bias terms of  $W_V, W_O$ . While our experiments show there is a strong correlation between the estimations by MAPS and head outputs, these terms may influence them. Incorporating these bias terms into the analysis is an interesting direction, which we leave for future works to explore.

## Acknowledgments

We thank Guy Dar, Daniela Gottesman, Ohav Barbi, Ori Yoran, Yoav Gur-Arieh and Samuel Amouyal who helped with analysis and provided useful feedback. This research was supported in part by The Israel Science Foundation grant 1083/24.

## References

- Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. [GQA: Training generalized multi-query transformer models from multi-head checkpoints](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.
- Andy Ardit, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*.
- Amos Azaria and Tom Mitchell. 2023. [The internal state of an LLM knows when it’s lying](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.
- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. [Pythia: A suite for analyzing large language models across training and scaling](#). In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.
- Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, et al. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*.
- Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Cohen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. 2021. [An interpretability illusion for bert](#). *ArXiv preprint*, abs/2104.07143.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What does BERT look at? an analysis of BERT’s attention](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.
- Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. [Analyzing transformers in embedding space](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16124–16170, Toronto, Canada. Association for Computational Linguistics.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. [The llama 3 herd of models](#). *ArXiv preprint*, abs/2407.21783.

- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12.
- Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *ArXiv preprint*, abs/2405.00208.
- Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *ArXiv preprint*, abs/2406.04093.
- Jorge García-Carrasco, Alejandro Maté, and Juan C. Trujillo. 2024. How does GPT-2 predict acronyms? extracting and understanding a circuit via mechanistic interpretability. In *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pages 3322–3330. PMLR.
- Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Forty-first International Conference on Machine Learning*.
- Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. 2024. Successor heads: Recurring, interpretable attention heads in the wild. In *The Twelfth International Conference on Learning Representations*.
- Yoav Gur-Arieh, Roy Mayan, Chen Agassy, Atticus Geiger, and Mor Geva. 2025. Enhancing automated interpretability with output-centric feature descriptions. *arXiv preprint arXiv:2501.08319*.
- Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2024. Linearity of relation decoding in transformer language models. In *The Twelfth International Conference on Learning Representations*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Osztrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *ArXiv preprint*, abs/2410.21276.
- Mojan Javaheripi and Sébastien Bubeck. 2023. Phi-2: The surprising power of small language models.
- Shahar Katz, Yonatan Belinkov, Mor Geva, and Lior Wolf. 2024. Backward lens: Projecting language model gradients into the vocabulary space. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2390–2422, Miami, Florida, USA. Association for Computational Linguistics.
- Geonhee Kim, Marco Valentino, and André Freitas. 2024. A mechanistic interpretation of syllogistic reasoning in auto-regressive language models. *ArXiv preprint*, abs/2408.08590.
- Connor Kissane, Robert Krzyzanowski, Joseph Isaac Bloom, Arthur Conmy, and Neel Nanda. 2024. Interpreting attention layer outputs with sparse autoencoders. In *ICML 2024 Workshop on Mechanistic Interpretability*.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.
- Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. 2015. Convergent learning: Do different neural networks learn the same representations? In *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015*, volume 44 of *Proceedings of Machine Learning Research*, pages 196–212, Montreal, Canada. PMLR.
- Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Callum Stuart McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. 2024. Copy suppression: Comprehensively understanding a motif in language model attention heads. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 337–363, Miami, Florida, US. Association for Computational Linguistics.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

- Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024a. *Circuit component reuse across tasks in transformer language models*. In *The Twelfth International Conference on Learning Representations*.
- Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024b. *Talking heads: Understanding inter-layer communication in transformer language models*. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Beren Millidge and Sid Black. 2022. *The singular value decompositions of transformer weight matrices are highly interpretable*.
- Neel Nanda and Joseph Bloom. 2022. Transformerlens. <https://github.com/TransformerLensOrg/TransformerLens>.
- nostalggebraist. 2020. *Interpreting gpt: the logit lens*.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. 2022. *In-context learning and induction heads*. *ArXiv preprint*, abs/2209.11895.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Patrick Schober, Christa Boer, and Lothar A. Schwarte. 2018. *Correlation coefficients: Appropriate use and interpretation*. *Anesthesia & Analgesia*, 126:1763–1768.
- Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. 2024. *LLM circuit analyses are consistent across training and scale*. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. *Attention is all you need*. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Jesse Vig and Yonatan Belinkov. 2019. *Analyzing the structure of attention in a transformer language model*. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.
- Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. *Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.
- Denny Vrandečić and Markus Krötzsch. 2014. *Wikidata: a free collaborative knowledgebase*. *Commun. ACM*, 57(10):78–85.
- Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. *Interpretability in the wild: a circuit for indirect object identification in GPT-2 small*. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrette Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. *Transformers: State-of-the-art natural language processing*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. *Efficient streaming language models with attention sinks*. In *The Twelfth International Conference on Learning Representations*.
- Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2024. *Jump to conclusions: Shortcutting transformers with linear transformations*. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9615–9625, Torino, Italia. ELRA and ICCL.
- Lei Yu, Meng Cao, Jackie CK Cheung, and Yue Dong. 2024. *Mechanistic understanding and mitigation of language model non-factual hallucinations*. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7943–7956, Miami, Florida, USA. Association for Computational Linguistics.
- Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. *Attention heads of large language models: A survey*. *ArXiv preprint*, abs/2409.03752.
- Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, Kun Wang, Yang Liu, Junfeng Fang, and Yongbin Li. 2024. *On the role of attention heads in large language model safety*. *ArXiv preprint*, abs/2410.13708.

## A Mapping Predefined Relations – Additional Details and Results

In §4, we showed how MAPS can be utilized to map all heads that implement a predefined relation across a language model. Here we offer further details on the datasets and implementation, as well as supplementary results.

## A.1 Datasets

We display the list of categories and relations used to map predefined relations (§4), alongside the sizes of the different datasets and examples for relations pairs in Table 3.

**Data collection** We obtained the relation pairs from the sources: WikiData (Vrandečić and Krötzsch, 2014); “English Word Frequency List” Kaggle dataset,<sup>5</sup> which is based on Google Books Ngram Viewer Exports, version 3, exported on Feb 17, 2020,<sup>6</sup> the datasets used by Hernandez et al. (2024), which are based on *CounterFact* (Meng et al., 2022) and WikiData (Vrandečić and Krötzsch, 2014), and ChatGPT.<sup>7</sup> We also used the *nltk* package (Loper and Bird, 2002) to validate several relation datasets. Except for the Translation and year to following datasets, all datasets are in English. The details on which source was used to compose which relation are presented in Table 4.

In the datasets for the relations work to location, verb to past tense, product by company, object to superclass, adj to superlative, adj to comparative, word to antonym, we filter out pairs where the source token appeared as a source token in other pairs. Relation pairs were filtered out from different datasets to assert their correctness.

**Data processing** For every model, we tokenized the various datasets using the model’s tokenizer. To maximize the number of words mapped to single tokens, we added a leading space before every word. For example, if the relation source word was “Don”, we tokenized the string “ Don” instead. Finally, we filtered out relation pairs where at least one of the words was mapped to more than one token.

## A.2 Implementation Details

**Applying the first MLP** For every model except Llama-3.1 70B, and similarly to Wang et al. (2023); Gould et al. (2024), we first applied the model’s first MLP to the tokens embeddings.<sup>8</sup> To adjust the embeddings to the first MLP’s input distribution,

<sup>5</sup><https://www.kaggle.com/datasets/wheelercode/english-word-frequency-list>

<sup>6</sup><https://storage.googleapis.com/books/ngrams/books/datasetsv3.html>

<sup>7</sup><https://chatgpt.com/>

<sup>8</sup>Notably, we did not apply the first MLP when we analyzed heads from the models’ first layers (layer 0), since the first attention layer precedes the first MLP in the computation.

we also applied the layer norm that precedes it. Regarding Llama-3.1 70B, we observed better results when not applying the first MLP.

**Selection of  $k$**  To calculate a head’s relation score  $\phi_R(M)$ , we obtain the top- $k$  tokens in  $\mathbf{m}_s$  for every source token  $s$ . For Pythia, GPT-2 and Phi-2 we set  $k = 1$  for copying and name-copying relations and  $k = 10$  for other relations. For the Llama-3.1 models we set  $k = 3$  for copying and name-copying and  $k = 25$  for other relations. Table 5 – which presents the tokenization applied to several base words by the tokenizers of Llama-3.1, GPT-2 and Pythia – demonstrates the need to set larger  $k$  values for Llama-3.1. The larger vocabulary size allows Llama-3.1’s tokenizer to express the same concept with more tokens.

## A.3 Random Baselines

A concern that may arise from choosing a relatively small relation score threshold, is that the results obtained by MAPS may capture the similarity of tokens embeddings, rather than a functionality implemented by attention head’s weights. To study this, we applied MAPS to randomly initialized matrices from the empirical distribution of the model. Concretely, for every layer in the original model, we sampled  $H$  random matrices (with the same shape as  $W_{VO}$ ) from a normal distribution, for which the mean and standard deviation are the average and the standard deviation of the  $W_{VO}$  matrices in the original layer. We applied our predefined relation analysis (described in §4.1) to those matrices and measured the amounts of “functional attention heads” classified among them.

For models Phi-2, Pythia 6.9B, Pythia 12B, Llama-3.1 8B and Llama-3.1 70B no random matrices were classified as relation heads. For GPT-2 xl, 5 matrices were classified as such, compared to 250 relation heads in the trained model, and out of 1200 heads in the model. This demonstrates that the choice of  $\tau = 15\%$  is meaningful for separating between functionalities of trained attention heads and random ones. While smaller thresholds could have also been justified by this experiment, we chose  $\tau = 15\%$  to assert that the heads encode a substantial fraction of the relation pairs.

## A.4 Additional Results

In Figure 5 we display all heads classified in Llama-3.1 70B, Llama-3.1 8B, Pythia 12B, Pythia 6.9B, Phi-2 and GPT-2 xl divided to four categories. In

Category	Relation	Example mappings	Dataset size per tokenizer		
			Llama-3.1	Pythia	GPT-2 / Phi-2
Algorithmic	Copying	(ottawa, ottawa),(say, say)	450	432	436
	Name copying	(Mallory, Mallory),(Walt, Walt)	134	113	132
	Word to first letter	(bend, b),(past, p)	238	237	238
	Word to last letter	(bend, d),(past, t)	238	237	238
	Year to following	(1728, 1729),(1958, 1959)		147	133
Knowledge	Country to capital	(Bulgaria, Sofia),(Chile, Santiago)	45	32	43
	Country to language	(Laos, Lao),(Denmark, Danish)	51	37	48
	Object to superclass	(tiger, animal),(carp, fish)	62	46	65
	Product by company	(Xbox, Microsoft),(Bravia, Sony)	39		40
	Work to location	(farmer, farm),(chef, kitchen)	48	34	45
Linguistic	Adj to comparative	(big, bigger),(high, higher)	47	44	48
	Adj to superlative	(angry, angriest),(high, highest)	39		41
	Noun to pronoun	(viewers, they),(Anna, she)	257	238	253
	Verb to past tense	(ask, asked),(eat, ate)	110	112	112
	Word to antonym	(love, hate),(right, wrong)	91	88	92
	Word to compound	(hot, hotdog),(wall, wallpaper)	38		36
	Word to homophone	(steal, steel),(sea, see)	103	88	91
	Word to synonym	(vague, obscure),(ill, sick)	154	142	154
Translation	English to French	(cat, chat),(love, amour)	32		
	English to Spanish	(cat, gato),(love, amor)	34		

Table 3: Datasets used for inspecting predefined operations in models with different tokenizers. Every model column describes the datasets’ sizes for this model. Different tokenizers lead to differences between datasets. We discard datasets that were left with  $\leq 30$  single-token mappings after tokenization.

Relation	Source	Notes
Country to capital	Wikidata	
Country to language	query	
Copying	Word frequency list	500 strings randomly sampled from the top 10,000
Year to following	Python code	
Word to synonym	ChatGPT	Validated using nltk
Word to homophone		
Noun to pronoun	ChatGPT	
Word to compound	ChatGPT	
Name copying		
English to Spanish	ChatGPT	Validated with Google Translate
English to French		
Work to location		
Object to superclass		
Product by company	Hernandez et al. (2024)	Extended using ChatGPT
Adj to comparative		
Adj to superlative		
Verb to past tense		
Word to antonym	Hernandez et al. (2024)	Extended using ChatGPT, validated using nltk
Word to first letter	Hernandez et al. (2024)	Letters converted to lowercase
Word to last letter		

Table 4: Sources for constructing per-relation datasets used in §4.

Tables 6 and 7 we present the number of relation heads (and suppression relation heads) discovered in the same models, divided into relations. We observe that several relations (Name copying, Adj to comparative, Word to first letter) are demonstrated by a relatively large number of heads in at least five out of six models. On the other hand, several relations (e.g., word to homophone, word to last letter) are demonstrated by a small number of heads across all models.

## B Additional Details on Evaluation Experiment

### B.1 Correlative Experiment

In §4.2 we conducted an experiment which calculates the correlation between MAPS’s estimations and heads outputs during inference.

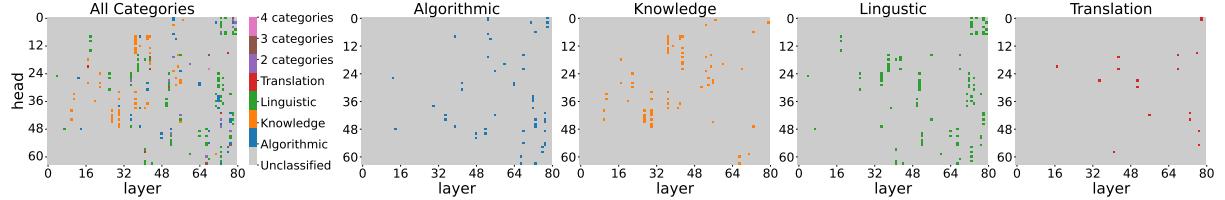
**Implementation details** Recall that the attention head’s formulation that we used:  $Y = \tilde{X}W_{VO}$  omits the bias terms of  $W_V, W_O$  (§2). To account for the bias term of  $W_V$  in the correlative experiment, where we compute the attention head’s output dynamically, we use both the original attention head definition Vaswani et al. (2017) and the formulation suggested by Elhage et al. (2021), which we have followed so far. First, following Vaswani et al. (2017), we obtain the head’s intermediate output:

Word	Llama-3.1	Pythia	GPT-2
Hello	>Hello, Hello, _hello, Ghello, hello, GHHello, Hallo, Bonjour, Hola	Hello, Ghello, hello, GHHello	hello, GHHello, Ghello, Hello
Please	Please, Gplease, please, GPLEASE, GPlease, .Please, PLEASE, >Please, Bitte, GBITTE, GBitte, Gbitte	Please, please, Gplease, GPlease	Please, Gplease, GPlease, GPLEASE, please
Love	GLEOVE, love, loven, Glove, Love, GLove, GLiebe, Gliebe, Gamour, Gamore, Gamor	love, GLOVE, Love, Glove, GLove	Glove, love, GLove, Love, GLOVE
Water	-water, _WATER, GWATER, _water, water, Gwater, Water, GWATER, .water, GWasser, 'eau, agua, Gagua	Water, Gwater, water, GWATER, agua	Water, water, Gwater, ewater, GWATER
School	GSCHOOL, -school, schools, Gschool, _school, school, GSchool, .school, School	School, Gschool, school, GSchool	GSchool, Gschool, school, GSCHOOL, School

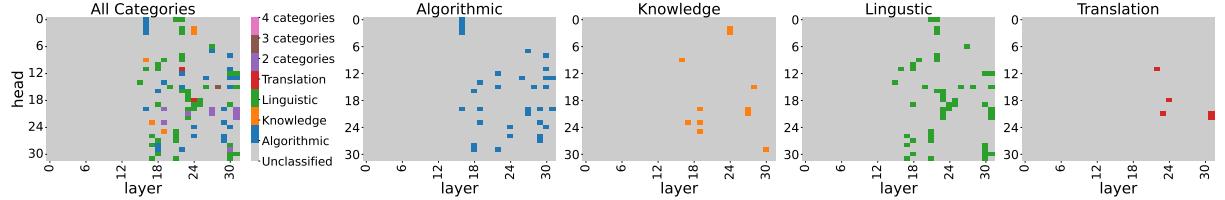
Table 5: Different tokenizations for base words by the tokenizers of Llama-3.1, Pythia and GPT-2. The “G” symbol represents a leading space. We observe that Llama-3.1’s larger vocabulary allows expressing every base word with more tokens.

Category	Relation	GPT-2 xl	Phi-2	Pythia 6.9B	Pythia 12B	Llama-3.1 8B	Llama-3.1 70B
Algorithmic	Copying	35	15	11	9	2	1
	Name copying	71	25	27	23	3	14
	Word to first letter	4	5	13	13	15	19
	Word to last letter	0	1	2	1	2	2
	Year to following	47	16	14	22		
Knowledge	Country to capital	60	17	26	31	5	26
	Country to language	50	23	24	30	5	28
	Object to superclass	17	12	11	19	0	13
	Product by company	24	4			1	3
	Work to location	10	6	6	8	0	5
Linguistic	Adj to comparative	45	47	27	28	8	25
	Adj to superlative	23	23			10	21
	Noun to pronoun	14	13	13	16	8	12
	Verb to past tense	15	27	17	28	8	18
	Word to antonym	12	15	11	15	5	11
	Word to compound	1	1			2	5
	Word to homophone	0	0	0	0	0	2
	Word to synonym	7	7	3	7	1	2
Translation	English to French					0	2
	English to Spanish					3	10

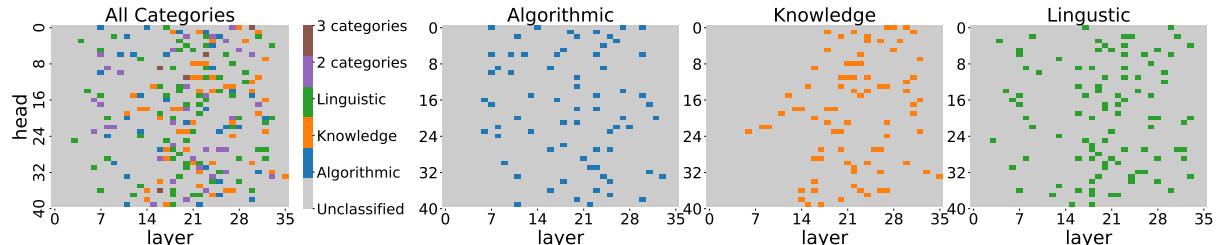
Table 6: Number of heads implementing each of the relations across different models.



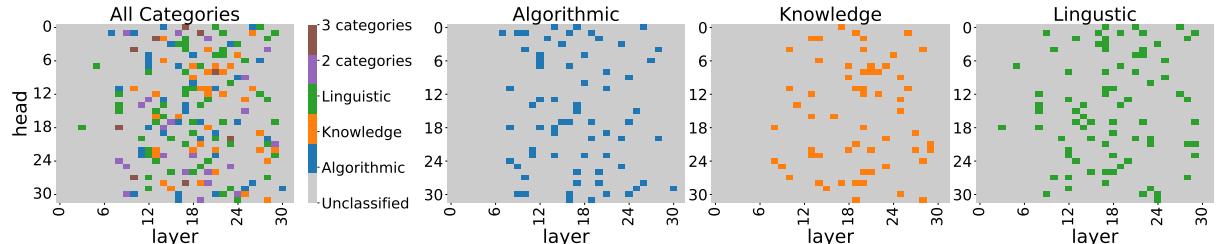
(a) Functionality mapping by MAPS for relations of 4 categories — algorithmic, knowledge, linguistic, translation — across all attention heads in Llama-3.1 70B. A head is marked for a specific category if it implements (also in a *suppression* variant) at least one relation from this category.



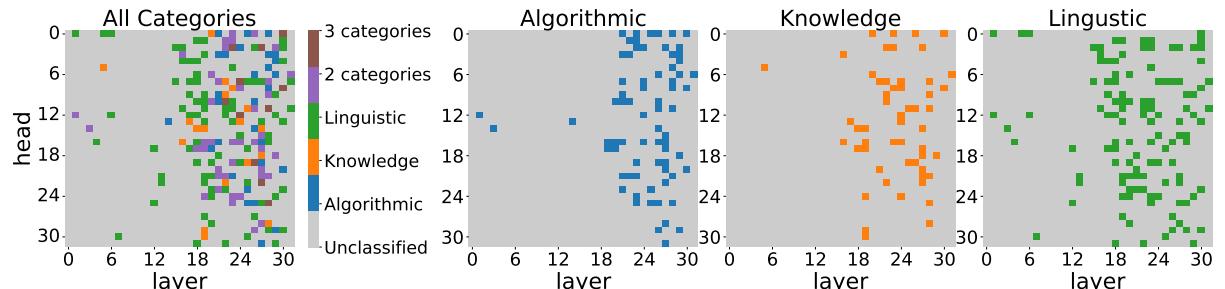
(b) Functionality mapping by MAPS for Llama-3.1 8B.



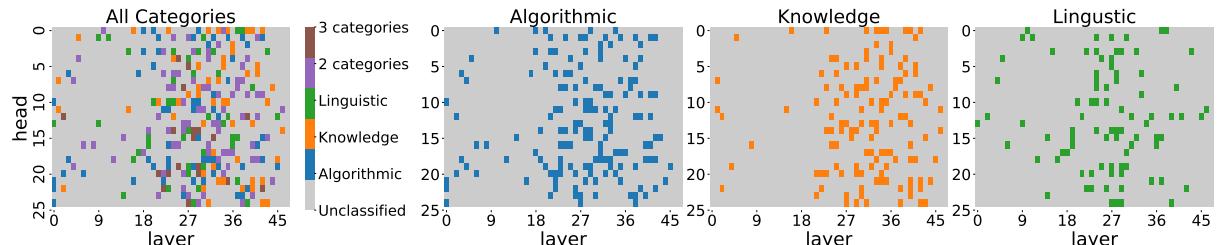
(c) Functionality mapping by MAPS for Pythia 12B.



(d) Functionality mapping by MAPS for Pythia 6.9B.



(e) Functionality mapping by MAPS for Phi-2.



(f) Functionality mapping by MAPS for GPT-2 xl.

Figure 5: Functionality mapping by MAPS.

Category	Relation	GPT-2 xl	Phi-2	Pythia 6.9B	Pythia 12B	Llama-3.1 8B	Llama-3.1 70B
Algorithmic	Copying	8	7	5	7	0	2
	Name copying	23	9	9	7	3	8
	Word to first letter	0	2	2	0	9	11
	Word to last letter	0	0	2	2	1	3
	Year to following	5	2	1	0		
Knowledge	Country to capital	19	8	5	5	1	10
	Country to language	26	12	9	11	3	9
	Object to superclass	2	5	3	6	0	4
	Product by company	7	0			0	3
	Work to location	2	3	1	1	0	2
Linguistic	Adj to comparative	11	29	15	19	5	13
	Adj to superlative	6	13			5	10
	Noun to pronoun	1	2	2	4	4	7
	Verb to past tense	2	21	8	7	5	10
	Word to antonym	0	4	3	4	2	3
	Word to compound	0	1			2	3
	Word to homophone	0	0	0	0	1	1
	Word to synonym	0	2	0	1	0	1
Translation	English to French					0	0
	English to Spanish					2	7

Table 7: Number of *suppression* heads implementing each of the relations across different models.

$\hat{y} \in \mathbb{R}^{n \times d_{\text{head}}}$ , where  $d_{\text{head}}$  is the inner dimension of the head, often fixed to  $\frac{d}{H}$ . Notably, this output already considers the bias term of  $W_V$ .<sup>9</sup> Then, following Elhage et al. (2021), we multiply this intermediate output by  $W_O \in \mathbb{R}^{d_{\text{head}} \times d}$  and obtain the head’s final output.

We use the following templates: “This is a document about ⟨s⟩”, “No ⟨s⟩ means no”, “The story of ⟨s⟩ contains”, “When I think about ⟨s⟩ I think about”.

**Additional results** Tables 8, 9, 10, 11, 12 present the correlation results between the static score  $\phi_R(h)$  inferred by our method and the score  $\phi_R^*(h)$  observed dynamically (both when we allow contextualization or not), obtained for Llama-3.1 70B, Llama-3.1 8B, Pythia 12B, Pythia 6.9B, GPT-2 xl. We also present the p-values and the maximum relation score obtained for any head in the model for the required relation. Notably, some of the lower correlations are demonstrated for relations that are not fully implemented by the model’s attention heads, as indicated by the small maximum relation scores. Tables 13, 14, 15, 16, 17 present the results (following the same format) for the *suppression* relation scores.

<sup>9</sup>In Vaswani et al. (2017),  $\hat{y}$  is viewed as the head’s final output.

## B.2 Causal Experiment

In §4.2 we measured the causal effect of removing the heads that implement a specific operation on the model’s performance in handling queries that depend on that operation.

**Implementation details** We evaluate models on tasks for 13 relations. For each model, we filter out relations where (a) the base accuracy is very low ( $<0.1$ ) or (b) there is no dataset for the relation (see §A). The task prompts used for the different relations are presented in Table 18. Notably, When ablating an attention head, we remove its output only from the last position of the prompt.

**Additional results** In Tables 19, 20, 21, 22, 23 we present the extended experiment results for Llama-3.1 70B, Llama-3.1 8B, Pythia 12B, Pythia 6.9B, GPT-2 xl.

## C Generalization to Multi-Token Entities – Additional Results

In §4.3 we conducted an experiment that evaluates how well the classifications by MAPS generalize to contextualized inputs. Table 24 shows the full results of this experiment. We omit the correlations for GPT-2 xl and the relation word to last letter, as all static scores are very small ( $\leq 0.05$ ).

Category	Relation	Correlation w/o context	Correlation w/ context	Max relation score (over heads)
Algorithmic	Copying	0.84	0.81	0.22
	Name copying	0.94	0.89	0.83
	Word to first letter	0.88	0.78	0.95
	Word to last letter	0.66	0.39	0.16
Knowledge	Country to capital	0.93	0.88	0.87
	Country to language	0.94	0.88	0.67
	Object to superclass	0.75	0.76	0.52
	Product by company	0.69	0.65	0.36
	Work to location	0.58	0.58	0.31
Linguistic	Adj to comparative	0.90	0.88	0.57
	Adj to superlative	0.90	0.84	0.67
	Noun to pronoun	0.57	0.41	0.33
	Verb to past tense	0.90	0.80	0.81
	Word to antonym	0.93	0.91	0.62
	Word to compound	0.85	0.82	0.39
	Word to homophone	0.87	0.80	0.16
	Word to synonym	0.84	0.79	0.27
	English to French	0.71	0.68	0.22
Translation	English to Spanish	0.85	0.83	0.47

Table 8: Correlation between the relation score of a head and the head’s output in Llama-3.1 70B, with and without head contextualization. The “max relation score” is the highest relation score achieved by a head in the model. All p-values observed are 0.

Category	Relation	Correlation w/o context	Correlation w/ context	Max relation score (over heads)
Algorithmic	Copying	0.76	0.73	0.18
	Name copying	0.95	0.95	0.71
	Word to first letter	0.90	0.78	0.89
	Word to last letter	0.67	0.36	0.27
Knowledge	Country to capital	0.85	0.85	0.49
	Country to language	0.76	0.62	0.31
	Object to superclass	0.74	0.73	0.15
	Product by company	0.46	0.49	0.18
	Work to location	0.44	0.45	0.10
Linguistic	Adj to comparative	0.85	0.86	0.60
	Adj to superlative	0.87	0.89	0.59
	Noun to pronoun	0.89	0.79	0.57
	Verb to past tense	0.91	0.86	0.73
	Word to antonym	0.90	0.86	0.37
	Word to compound	0.78	0.62	0.21
	Word to homophone	0.85	0.75	0.08
	Word to synonym	0.79	0.69	0.17
	English to French	0.71	0.68	0.12
Translation	English to Spanish	0.82	0.81	0.29

Table 9: Correlation between the relation score of a head and the head’s output in Llama-3.1 8B, with and without head contextualization. The “max relation score” is the highest relation score achieved by a head in the model. All p-values observed are  $\leq 3.9e-128$ .

Category	Relation	Correlation w/o context	Correlation w/ context	Max relation score (over heads)
Algorithmic	Copying	0.89	0.60	0.42
	Name copying	0.86	0.57	0.65
	Word to first letter	0.84	0.62	0.75
	Word to last letter	0.36	0.17	0.16
	Year to following	0.90	0.78	1.00
Knowledge	Country to capital	0.93	0.89	0.97
	Country to language	0.94	0.89	0.86
	Object to superclass	0.88	0.87	0.74
	Work to location	0.75	0.64	0.29
Linguistic	Adj to comparative	0.92	0.80	0.95
	Noun to pronoun	0.85	0.74	0.50
	Verb to past tense	0.89	0.71	0.54
	Word to antonym	0.92	0.85	0.60
	Word to homophone	0.67	0.43	0.07
	Word to synonym	0.90	0.67	0.35

Table 10: Correlation between the relation score of a head and the head’s output in Pythia 12B, with and without head contextualization. The “max relation score” is the highest relation score achieved by a head in the model. All p-values observed are  $\leq 5.7\text{e-}40$ .

Category	Relation	Correlation w/o context	Correlation w/ context	Max relation score (over heads)
Algorithmic	Copying	0.88	0.45	0.53
	Name copying	0.94	0.62	0.96
	Word to first letter	0.87	0.64	0.67
	Word to last letter	0.44	0.43	0.27
	Year to following	0.94	0.79	0.99
Knowledge	Country to capital	0.95	0.91	0.97
	Country to language	0.91	0.86	0.84
	Object to superclass	0.88	0.88	0.72
	Work to location	0.76	0.68	0.29
Linguistic	Adj to comparative	0.91	0.76	0.77
	Noun to pronoun	0.89	0.67	0.63
	Verb to past tense	0.91	0.70	0.81
	Word to antonym	0.93	0.87	0.64
	Word to homophone	0.70	0.38	0.05
	Word to synonym	0.93	0.64	0.36

Table 11: Correlation between the relation score of a head and the head’s output in Pythia 6.9B, with and without head contextualization. The “max relation score” is the highest relation score achieved by a head in the model. All p-values observed are  $\leq 1.7\text{e-}139$ .

Category	Relation	Correlation w/o context	Correlation w/ context	Max relation score (over heads)
Algorithmic	Copying	0.95	0.65	0.52
	Name copying	0.97	0.70	0.92
	Word to first letter	0.91	0.69	0.32
	Word to last letter	0.61	0.20	0.05
	Year to following	0.94	0.74	0.95
Knowledge	Country to capital	0.98	0.88	0.98
	Country to language	0.96	0.84	0.75
	Object to superclass	0.94	0.81	0.43
	Product by company	0.96	0.91	0.65
	Work to location	0.88	0.73	0.31
Linguistic	Adj to comparative	0.95	0.78	0.88
	Adj to superlative	0.94	0.73	0.54
	Noun to pronoun	0.96	0.68	0.58
	Verb to past tense	0.93	0.76	0.28
	Word to antonym	0.96	0.85	0.38
	Word to compound	0.80	0.65	0.17
	Word to homophone	0.46	0.38	0.02
	Word to synonym	0.95	0.79	0.21

Table 12: Correlation between the relation score of a head and the head’s output in GPT-2 xl, with and without head contextualization. The “max relation score” is the highest relation score achieved by a head in the model. All p-values observed are  $\leq 1.1\text{e-}45$ .

Category	Relation	Correlation w/o context	Correlation w/ context	Max relation score (over heads)
Algorithmic	Copying	0.88	0.85	0.18
	Name copying	0.95	0.83	0.66
	Word to first letter	0.86	0.72	0.56
	Word to last letter	0.56	0.42	0.33
Knowledge	Country to capital	0.91	0.90	0.84
	Country to language	0.89	0.89	0.49
	Object to superclass	0.81	0.83	0.39
	Product by company	0.81	0.78	0.31
	Work to location	0.70	0.70	0.21
Linguistic	Adj to comparative	0.91	0.88	0.72
	Adj to superlative	0.90	0.87	0.56
	Noun to pronoun	0.33	0.30	0.46
	Verb to past tense	0.91	0.80	0.54
	Word to antonym	0.91	0.80	0.35
	Word to compound	0.86	0.82	0.24
	Word to homophone	0.91	0.81	0.31
	Word to synonym	0.83	0.77	0.21
Translation	English to French	0.61	0.59	0.09
	English to Spanish	0.86	0.83	0.35

Table 13: Correlation between the *suppression* relation score of a head and the head’s output in Llama-3.1 70B, with and without head contextualization. The “max relation score” is the highest relation score achieved by a head in the model. All p-values observed are 0.

Category	Relation	Correlation w/o context	Correlation w/ context	Max relation score (over heads)
Algorithmic	Copying	0.77	0.74	0.11
	Name copying	0.99	0.95	0.72
	Word to first letter	0.78	0.41	0.61
	Word to last letter	0.77	0.31	0.25
Knowledge	Country to capital	0.90	0.87	0.18
	Country to language	0.76	0.74	0.20
	Object to superclass	0.61	0.63	0.08
	Product by company	0.44	0.38	0.08
	Work to location	0.40	0.32	0.12
Linguistic	Adj to comparative	0.81	0.91	0.81
	Adj to superlative	0.87	0.93	0.62
	Noun to pronoun	0.80	0.57	0.40
	Verb to past tense	0.90	0.85	0.46
	Word to antonym	0.81	0.70	0.29
	Word to compound	0.84	0.76	0.24
	Word to homophone	0.89	0.61	0.17
	Word to synonym	0.75	0.65	0.09
Translation	English to French	0.74	0.65	0.06
	English to Spanish	0.84	0.81	0.26

Table 14: Correlation between the *suppression* relation score of a head and the head’s output in Llama-3.1 8B, with and without head contextualization. The “max relation score” is the highest relation score achieved by a head in the model. All p-values observed are  $\leq 2.6\text{e-}89$ .

Category	Relation	Correlation w/o context	Correlation w/ context	Max relation score (over heads)
Algorithmic	Copying	0.91	0.78	0.31
	Name copying	0.99	0.72	1.00
	Word to first letter	0.48	0.18	0.11
	Word to last letter	0.59	0.23	0.19
	Year to following	0.39	0.59	0.12
Knowledge	Country to capital	0.63	0.62	0.56
	Country to language	0.84	0.70	0.46
	Object to superclass	0.79	0.77	0.41
	Work to location	0.61	0.64	0.24
Linguistic	Adj to comparative	0.93	0.74	0.73
	Noun to pronoun	0.68	0.29	0.28
	Verb to past tense	0.96	0.75	0.73
	Word to antonym	0.90	0.77	0.32
	Word to homophone	0.61	0.39	0.03
	Word to synonym	0.82	0.63	0.16

Table 15: Correlation between the *suppression* relation score of a head and the head’s output in Pythia 12B, with and without head contextualization. The “max relation score” is the highest relation score achieved by a head in the model. All p-values observed are  $\leq 2.2\text{e-}45$ .

Category	Relation	Correlation w/o context	Correlation w/ context	Max relation score (over heads)
Algorithmic	Copying	0.88	0.81	0.41
	Name copying	0.98	0.79	0.96
	Word to first letter	0.81	0.37	0.31
	Word to last letter	0.30	0.08	0.24
	Year to following	0.45	0.80	0.33
Knowledge	Country to capital	0.92	0.91	0.66
	Country to language	0.89	0.81	0.51
	Object to superclass	0.86	0.78	0.33
	Work to location	0.73	0.58	0.21
Linguistic	Adj to comparative	0.95	0.83	0.59
	Noun to pronoun	0.86	0.51	0.56
	Verb to past tense	0.94	0.80	0.82
	Word to antonym	0.91	0.78	0.30
	Word to homophone	0.49	0.31	0.02
	Word to synonym	0.87	0.73	0.13

Table 16: Correlation between the *suppression* relation score of a head and the head’s output in Pythia 6.9B, with and without head contextualization. The “max relation score” is the highest relation score achieved by a head in the model. All p-values observed are  $\leq 3.6\text{e-}7$ .

Category	Relation	Correlation w/o context	Correlation w/ context	Max relation score (over heads)
Algorithmic	Copying	0.97	0.71	0.29
	Name copying	0.99	0.72	0.97
	Word to first letter	0.78	0.52	0.04
	Word to last letter	0.78	0.54	0.06
	Year to following	0.75	0.52	0.32
Knowledge	Country to capital	0.94	0.80	0.72
	Country to language	0.96	0.78	0.50
	Object to superclass	0.89	0.82	0.23
	Product by company	0.88	0.77	0.33
	Work to location	0.83	0.62	0.18
Linguistic	Adj to comparative	0.86	0.60	0.38
	Adj to superlative	0.81	0.59	0.27
	Noun to pronoun	0.92	0.34	0.40
	Verb to past tense	0.84	0.64	0.17
	Word to antonym	0.53	0.37	0.05
	Word to compound	0.80	0.58	0.14
	Word to homophone	0.10	0.04	0.01
	Word to synonym	0.81	0.59	0.08

Table 17: Correlation between the *suppression* relation score of a head and the head’s output in GPT-2 xl, with and without head contextualization. The “max relation score” is the highest relation score achieved by a head in the model. All p-values observed are  $\leq 2.3\text{e-}3$ .

Relation	Prompt
Adj to comparative	lovely-> lovelier; edgy-> edgier; <s>->
Copying	walk-> walk; cat-> cat; water-> water; <s>->
Country to capital	The capital of <s> is
Country to language	The official language of <s> is
English to Spanish	apartment-> departamento; computer-> computadora; tribe-> tribu; <s>->
Name copying	John-> John; Donna-> Donna; <s>->
Noun to pronoun	mother-> she; father-> he; tribe-> they; actress-> she; apartment-> it; <s>->
Object to superclass	A <s> is a kind of
Product by company	Nesquik is made by Nestlé; Mustang is made by Ford; <s> is made by
Verb to past tense	hike->hiked; purchase-> purchased; <s>->
Word to first letter	word-> w, o, r, d; cat-> c, a, t; <s>->
Word to last letter	word-> d, r, o, w; cat-> t, a, c; <s>->
Year to following	1300-> 1301; 1000-> 1001; <s>->

Table 18: Relations and prompts used in the causal experiment. The <s> string is replaced with the relation’s source tokens.

Relation name	# heads removed	TR tasks			CTR tasks		
		Base	-TR	-RND	# tasks	Base (CTR)	-TR (CTR)
Adj to comparative	175	0.98	↓13% 0.85	↓0% 0.98 ± 0.00	5	0.94 ± 0.05	↓3% 0.92 ± 0.08
Copying	250	0.97	↓30% 0.68	↓0% 0.97 ± 0.01	3	0.97 ± 0.03	↓23% 0.75 ± 0.34
Country to capital	118	0.84	↓66% 0.29	↑1% 0.85 ± 0.09	5	0.93 ± 0.08	↑0% 0.94 ± 0.09
Country to language	133	0.96	↓6% 0.90	↓0% 0.96 ± 0.00	4	0.92 ± 0.08	↓1% 0.92 ± 0.10
English to Spanish	175	0.91	↓6% 0.85	↑0% 0.91 ± 0.00	4	0.97 ± 0.03	↑0% 0.97 ± 0.03
Name copying	205	0.99	↓95% 0.05	↑1% 1.00 ± 0.00	3	0.97 ± 0.03	↓15% 0.83 ± 0.23
Noun to pronoun	154	0.98	↑0% 0.98	↑0% 0.98 ± 0.00	5	0.93 ± 0.08	↓1% 0.92 ± 0.09
Object to superclass	119	0.79	↓4% 0.76	↓2% 0.77 ± 0.02	5	0.88 ± 0.11	↓3% 0.85 ± 0.15
Product by company	59	0.67	↓4% 0.64	↓0% 0.67 ± 0.00	1	0.79 ± 0.00	↓2% 0.77 ± 0.00
Word to first letter	250	1.00	↓8% 0.92	↓0% 1.00 ± 0.00	5	0.94 ± 0.05	↓5% 0.89 ± 0.14
Word to last letter	250	0.92	↓18% 0.76	↑1% 0.93 ± 0.01	5	0.94 ± 0.05	↑1% 0.95 ± 0.04

Table 19: Accuracy of Llama-3.1 70B on tasks for a target relation (TR) versus on control (CTR) tasks, when removing heads implementing the relation compared to when removing random heads (RND). Results for RND heads are averaged over 5 experiments.

Relation name	# heads removed	TR tasks			CTR tasks		
		Base	-TR	-RND	# tasks	Base (CTR)	-TR (CTR)
Adj to comparative	69	0.98	↓7% 0.91	↓3% 0.95 ± 0.05	4	0.96 ± 0.04	↑0% 0.96 ± 0.04
Copying	150	1.00	↓94% 0.06	↓0% 1.00 ± 0.00	3	0.95 ± 0.04	↓5% 0.91 ± 0.05
Country to capital	19	0.89	↓75% 0.22	↑2% 0.91 ± 0.03	5	0.87 ± 0.12	↑1% 0.87 ± 0.12
Country to language	30	0.98	↓50% 0.49	↑1% 0.99 ± 0.01	5	0.98 ± 0.02	↓0% 0.98 ± 0.02
English to Spanish	54	0.94	↑3% 0.97	↓1% 0.93 ± 0.01	3	0.95 ± 0.04	↑2% 0.97 ± 0.02
Name copying	70	1.00	↓87% 0.13	↓0% 1.00 ± 0.00	2	0.94 ± 0.05	↓4% 0.90 ± 0.08
Noun to pronoun	35	0.98	↓0% 0.98	↑0% 0.99 ± 0.00	5	0.97 ± 0.04	↑1% 0.98 ± 0.03
Object to superclass	34	0.74	↓11% 0.66	↑1% 0.75 ± 0.01	2	0.79 ± 0.09	↓3% 0.77 ± 0.07
Product by company	12	0.54	↓5% 0.51	↑4% 0.56 ± 0.01	1	0.70 ± 0.00	↓1% 0.69 ± 0.00
Verb to past tense	113	0.70	↓61% 0.27	↓7% 0.65 ± 0.10	2	0.71 ± 0.18	↓1% 0.70 ± 0.14
Word to first letter	150	1.00	↓98% 0.02	↓0% 1.00 ± 0.00	5	0.96 ± 0.04	↓30% 0.67 ± 0.33

Table 20: Accuracy of Llama-3.1 8B on tasks for a target relation (TR) versus on control (CTR) tasks, when removing heads implementing the relation compared to when removing random heads (RND). Results for RND heads are averaged over 5 experiments.

Relation name	# heads removed	TR tasks				CTR tasks		
		Base	-TR	-RND	# tasks	Base (CTR)	-TR (CTR)	
Adj to comparative	150	0.91	↓77% 0.20	↓10% 0.82 ± 0.07	3	0.92 ± 0.04	↓32% 0.63 ± 0.18	
Copying	150	1.00	↓32% 0.68	↓0% 1.00 ± 0.00	3	0.95 ± 0.05	↓7% 0.88 ± 0.11	
Country to capital	75	0.97	↓100% 0.00	↓2% 0.95 ± 0.02	2	0.89 ± 0.02	↑0% 0.90 ± 0.01	
Country to language	94	1.00	↓92% 0.08	↓4% 0.96 ± 0.01	2	0.89 ± 0.01	↓0% 0.89 ± 0.01	
Name copying	150	1.00	↓76% 0.24	↓0% 1.00 ± 0.00	2	0.90 ± 0.02	↑2% 0.92 ± 0.05	
Noun to pronoun	105	0.88	↓48% 0.46	↓2% 0.86 ± 0.03	5	0.90 ± 0.07	↓3% 0.88 ± 0.08	
Object to superclass	75	0.78	↓50% 0.39	↓13% 0.68 ± 0.03	2	0.90 ± 0.02	↓3% 0.87 ± 0.09	
Verb to past tense	150	0.22	↓84% 0.04	↑17% 0.26 ± 0.11	1	0.03 ± 0.00	↓33% 0.02 ± 0.00	
Word to first letter	150	0.91	↓63% 0.34	↓4% 0.87 ± 0.04	5	0.91 ± 0.08	↓19% 0.74 ± 0.30	
Year to following	56	0.92	↓100% 0.00	↓5% 0.87 ± 0.07	2	0.83 ± 0.05	↓5% 0.79 ± 0.03	

Table 21: Accuracy of Pythia 12B on tasks for a target relation (TR) versus its accuracy on control (CTR) tasks, when removing heads implementing the relation compared to when removing random heads (RND). Results for RND heads are averaged over 5 experiments.

Relation name	# heads removed	TR tasks				CTR tasks		
		Base	-TR	-RND	# tasks	Base (CTR)	-TR (CTR)	
Adj to comparative	124	0.52	↓100% 0.00	↓51% 0.25 ± 0.18	1	0.68 ± 0.00	↓25% 0.51 ± 0.00	
Copying	150	1.00	↓93% 0.07	↓1% 0.99 ± 0.01	0			
Country to capital	45	0.97	↓100% 0.00	↓1% 0.96 ± 0.02	1	1.00 ± 0.00	↓0% 1.00 ± 0.00	
Country to language	74	0.97	↓92% 0.08	↑1% 0.98 ± 0.01	0			
Name copying	143	1.00	↓97% 0.03	↓1% 0.99 ± 0.01	0			
Noun to pronoun	102	0.68	↓46% 0.37	↑13% 0.77 ± 0.09	3	0.68 ± 0.11	↓25% 0.51 ± 0.22	
Object to superclass	67	0.78	↓53% 0.37	↓4% 0.75 ± 0.02	2	0.71 ± 0.03	↑1% 0.71 ± 0.18	
Verb to past tense	150	0.43	↓94% 0.03	↓16% 0.36 ± 0.07	0			
Word to first letter	66	1.00	↓100% 0.00	↓0% 1.00 ± 0.00	2	0.97 ± 0.00	↓13% 0.85 ± 0.13	
Year to following	52	0.73	↓100% 0.00	↑5% 0.77 ± 0.07	2	0.73 ± 0.05	↓2% 0.71 ± 0.05	

Table 22: Accuracy of Pythia 6.9B on tasks for a target relation (TR) versus its accuracy on control (CTR) tasks, when removing heads implementing the relation compared to when removing random heads (RND). Results for RND heads are averaged over 5 experiments.

Relation name	# heads removed	TR tasks				CTR tasks		
		Base	-TR	-RND	# tasks	Base (CTR)	-TR (CTR)	
Copying	150	0.99	↓30% 0.69	↓0% 0.99 ± 0.00	0			
Country to capital	38	0.88	↓100% 0.00	↓3% 0.86 ± 0.05	1	0.71 ± 0.00	↑2% 0.72 ± 0.00	
Country to language	148	0.96	↓91% 0.08	↓2% 0.94 ± 0.01	0			
Name copying	133	0.76	↓100% 0.00	↓15% 0.65 ± 0.08	1	0.71 ± 0.00	↓15% 0.60 ± 0.00	
Noun to pronoun	27	0.71	↓26% 0.53	↓2% 0.69 ± 0.04	4	0.72 ± 0.13	↓3% 0.69 ± 0.16	
Object to superclass	99	0.71	↓54% 0.32	↓1% 0.70 ± 0.02	1	0.71 ± 0.00	↓42% 0.41 ± 0.00	
Product by company	73	0.40	↓81% 0.08	↓0% 0.40 ± 0.00	1	0.40 ± 0.00	↑2% 0.41 ± 0.00	
Verb to past tense	150	0.40	↓56% 0.18	↓4% 0.38 ± 0.18	0			
Word to first letter	62	0.18	↓16% 0.16	↓1% 0.18 ± 0.02	1	0.04 ± 0.00	↑250% 0.15 ± 0.00	
Year to following	54	0.53	↓100% 0.00	↓5% 0.50 ± 0.03	1	0.71 ± 0.00	↓36% 0.45 ± 0.00	

Table 23: Accuracy of GPT-2 xl on tasks for a target relation (TR) versus its accuracy on control (CTR) tasks, when removing heads implementing the relation compared to when removing random heads (RND). Results for RND heads are averaged over 5 experiments.

Model	Relation	# samples	W/o context		W/ context	
			Single-token	Multi-token	Single-token	Multi-token
Pythia 12B	Copying	283	0.91	0.85	0.48	0.44
	Country to capital	30	0.94	0.93	0.85	0.87
	Country to language	70	0.94	0.90	0.88	0.83
	Name copying	83	0.87	0.76	0.38	0.33
	Noun to pronoun	174	0.84	0.85	0.78	0.79
	Object to superclass	91	0.88	0.89	0.84	0.86
	Word to first letter	77	0.83	0.73	0.56	0.64
	Word to last letter	77	0.34	0.50	0.11	0.09
	Word to synonym	71	0.92	0.86	0.61	0.58
	Work to location	65	0.77	0.72	0.74	0.70
Pythia 6.9B	Year to following	65	0.90	0.84	0.64	0.60
	Copying	283	0.90	0.87	0.34	0.32
	Country to capital	30	0.95	0.93	0.89	0.89
	Country to language	70	0.92	0.88	0.85	0.83
	Name copying	83	0.94	0.92	0.47	0.47
	Noun to pronoun	174	0.89	0.85	0.69	0.70
	Object to superclass	91	0.88	0.90	0.86	0.82
	Word to first letter	77	0.89	0.79	0.59	0.66
	Word to last letter	77	0.45	0.70	0.44	0.44
	Word to synonym	71	0.94	0.91	0.62	0.62
GPT-2 xl	Work to location	65	0.79	0.76	0.71	0.75
	Year to following	65	0.94	0.87	0.72	0.67
	Copying	301	0.95	0.88	0.68	0.64
	Country to capital	34	0.98	0.97	0.87	0.86
	Country to language	70	0.96	0.91	0.82	0.80
	Name copying	91	0.97	0.93	0.60	0.58
	Noun to pronoun	154	0.97	0.95	0.47	0.56
	Object to superclass	97	0.93	0.89	0.83	0.82
	Word to first letter	78	0.92	0.89	0.53	0.72
	Word to synonym	79	0.95	0.89	0.79	0.76
	Work to location	67	0.89	0.80	0.74	0.76
	Year to following	90	0.95	0.82	0.74	0.63

Table 24: Extended results for the multi-token experiment, presented in Section 4.3. All p-values observed are  $\leq 9.3e-4$ .

## D Comparison to Head Operations Identified in Prior Works

**Name-mover heads in GPT-2 small** Wang et al. (2023) studied the *Indirect Object Identification* circuit in GPT-2 small. Analyzing the operations of the circuit’s heads, they defined heads that copy names as *Name-Mover* heads and heads that suppress names as *Negative Name-Mover* heads. They also classified heads that contribute to these tasks when the original mover heads are ablated as “backup” mover heads.

Using MAPS we classified all three name-mover heads as implementing the name copying relation, and the two negative name-mover heads as implementing the suppression variant of name copying. We note that a similar analysis was performed by Wang et al. (2023) as well. However, by applying MAPS to all heads in the model, and not just the heads in the discovered circuit, we were able to identify 21 additional name-copying heads as well, 6 of which were identified by Wang et al. (2023) as “backup” heads. One backup mover head and one backup negative mover head that were identified by Wang et al. (2023), were not identified by MAPS. Moreover, we find that each of the five identified name-mover heads implements a myriad of other relations. In Figure 6a we present the name copying relation scores for all heads in GPT-2 small and the heads classified by Wang et al. (2023).

We further examined the name copying heads not classified by Wang et al. (2023), to study whether their omission was mostly due to limited involvement in the specific task studied by Wang et al. (2023), or instead a consequence of inaccurate estimations by MAPS. These heads show a strong correlation (0.94, p-value of  $2.5e-7$ ) between their name copying static and dynamic relation scores (for the prompt This is a document about  $\langle s \rangle$ , see §4.2), when attention is restricted to the name position, suggesting that they indeed copy names when they attend to them. However, the attention weight assigned to the name token may change depending on the context. For example, head 8.11 in GPT-2 small has a static relation score of 0.88. Its dynamic relation score is 0.23 for the prompt This is a document about  $\langle s \rangle$ , but it increases substantially to 0.92 for the prompt “ John->John; Donna-> Donna;  $\langle s \rangle$ ->”. We anticipate that other relation heads will demonstrate the name-copying functionality for other prompts or interventions.

Crafting prompts that steer heads to demonstrate a specific functionality over another (for example by adapting MAPS to the  $W_{QK}$  matrix) is an interesting direction for future work.

**Mover heads in GPT-2 medium** Merullo et al. (2024a) studied the Indirect Object Identification (IOI) and Colored Objects circuits in GPT-2 medium. They discovered two sets of attention heads implementing certain functions, both called “Mover” heads. Heads from the first set copy names (in IOI), and heads from the second set copy colors (in the Colored Objects task). The authors also point out a significant overlap between the two sets.

Using MAPS, we classified all mover heads as implementing the name copying relation. We find that many of these heads also implement the relations: year to following, country to language, country to capital, copying. Lastly, we identify 31 other name-copying heads. Notably, in our counting, we omit the heads 14.5, 17.10, 16.0, 18.12, and 21.7, which are labeled in Figure 2 of Merullo et al. (2024a) as Mover-heads. This is because, to the best of our understanding, the paper does not provide any explanation for why they are classified as such, while other heads are described as more important than them.

**Capital heads in GPT-2 medium** Merullo et al. (2024a) have also studied a circuit for resolving the capital city of a country (in Appendix I). MAPS identified all attention heads classified in that study, along with 15 others. In Figure 6b we present the name copying, country to capital relation scores for all heads in GPT-2 medium and the heads classified by Merullo et al. (2024a).

## E Automatic Mapping of Salient Head Operations

### E.1 Automatic Functionality Inference

In §5.1 we showed that GPT-4o can be utilized to interpret attention heads’ salient operations. Here, we provide additional implementation details and present an evaluation of the interpretation quality.

**Implementation details** We found that GPT-4o sometimes describes in words that the pattern is unclear, rather than just outputting the word “Unclear”, as requested. To handle these cases, we classify every head for which GPT-4o’s response contained the string “clear” as a head where a pattern was not detected. We view this as an upper

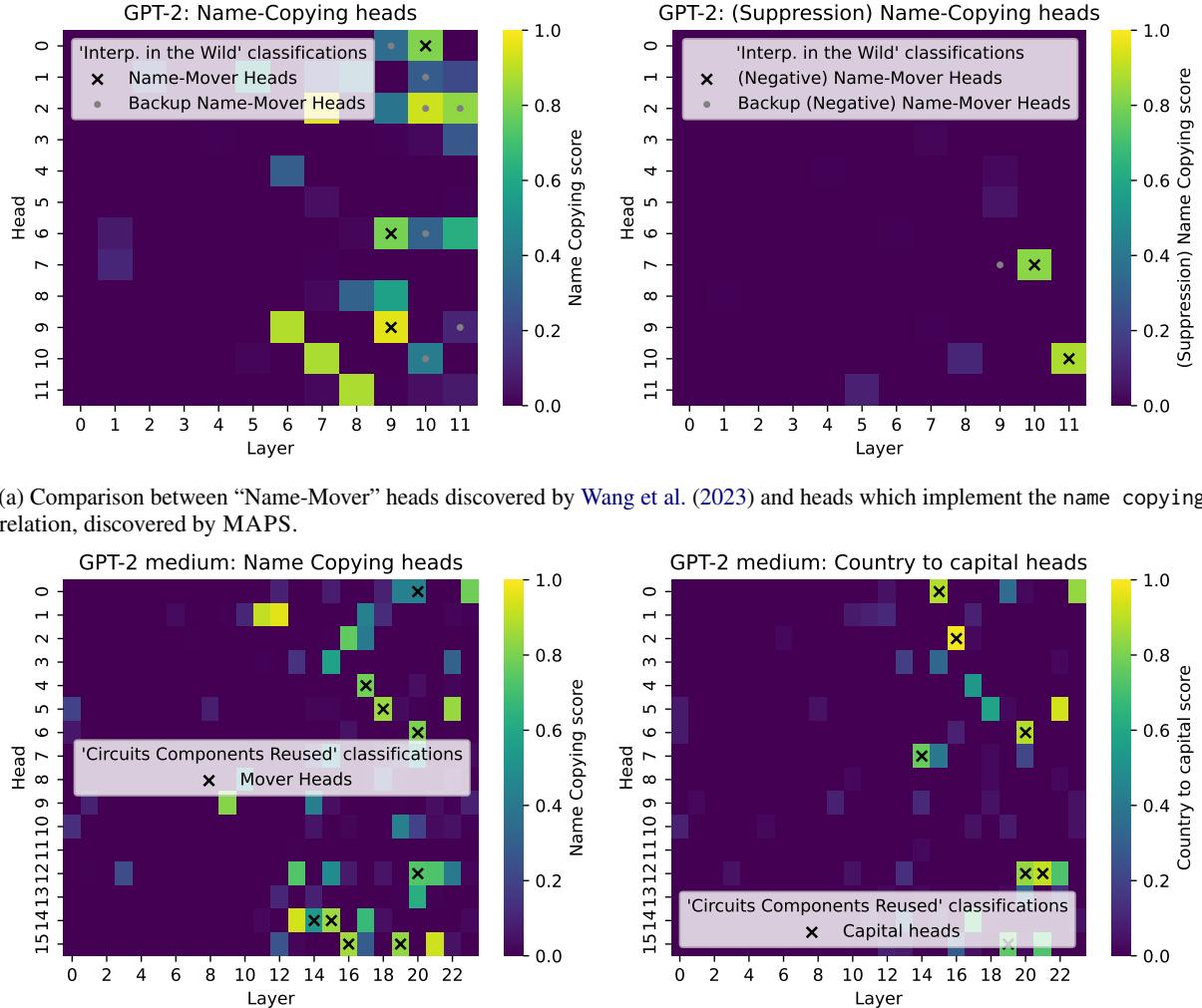


Figure 6: Comparison between relation heads discovered by MAPS and heads classified in prior works.

bound over the true ratio of heads with undetected patterns. Also, for some heads, GPT-4o would stop generating descriptions mid-generation. We hypothesize that it is because of strings viewed as special GPT-4o tokens that appeared in the salient mappings. We solved this issue by querying GPT-4o again with other random seeds. We note that in several mappings the salient tokens were decoded as an unreadable character. This could be solved by alternating between Transformers package ([Wolf et al., 2020](#)) decoding functions.

**Prompt format** We present the prompt used to query GPT-4o in Table [26](#).

**Examples** Table [25](#) provides examples of salient mappings and the patterns described by GPT-4o for three attention heads in GPT-2 xl and Pythia 6.9B.

## E.2 Interpretation Quality

To assess the accuracy and plausibility of the model-generated descriptions, we let human annotators — five graduate students who are fluent English speakers — evaluate its responses in terms of (a) did GPT-4o correctly recognize the existence of a pattern in the mappings, (b) the quality of the generated descriptions, (c) the category of the recognized patterns. We conduct this study for a random sample of 138 (13.5%) heads in Pythia 6.9B and 134 (11.2%) heads in GPT-2 xl.

**Annotation instructions** We present the instructions given to the human annotators in Figures [7,8](#).

**Human study results** The overall results per question and the distribution of responses across models and layers are presented in Figure [9](#) (Question 1), Figure [10](#) (Question 2), Figure [11](#) (Question 3). In 80% of the cases, GPT-4o correctly identifies the presence or absence of a pattern. In most of the failure cases (87%), the model described a pattern that is not visible in the mappings. We also find that in lower layers there are fewer patterns and they are harder to parse: there are higher rates of *unnatural* patterns and inaccurate descriptions. This agrees with our findings in §4. In case of an observable pattern, GPT-4o will identify it: for 95% of heads with observable patterns, GPT-4o described a pattern, and <2% of the descriptions were labeled “poor”. Overall, this analysis shows that the quality of our automatic annotation pipeline is reasonable and demonstrates promising trends in automatically interpreting attention heads with MAPS. We leave further improvements to

the pipeline for future work to explore. In particular, addressing model hallucinations could involve methods like aggregating multiple model responses to check its confidence ([Kuhn et al., 2023](#)), using intrinsic classifiers for hallucinations (e.g. [Azaria and Mitchell \(2023\)](#), [Yu et al. \(2024\)](#)), employing a strong LLM to indicate whether the generated pattern matches the mappings ([Gur-Arieh et al., 2025](#)), using an NLI model ([Bohnet et al., 2022](#)), or similarity-based heuristics.

## F Analysis of Global Versus Specific Functionality

We observe that the mappings in  $M$  provide a broad view of the head’s functionality, particularly on how *global* the head’s operation is. For example, a head that maps any token to an end-of-sequence token has *global* functionality, whereas heads that map countries to their capitals, colors to their complementary pairs, and so on, demonstrate *specific* operations. In this section, we use properties of  $M$  to analyze how global the functionalities of attention heads in LLMs are.

**Analysis** We estimate how global the functionality of a given head is using two metrics: *input skewness*, which captures the skewness of the head’s operation towards specific inputs, and *output space size*, which estimates the number of tokens the head tends to output. For input skewness, we obtain the saliency scores  $\sigma_t(W_{VO}) \forall t \in \mathcal{V}$  according to the head (see §3.2), and calculate the skewness of their distribution. For output space size, we compute for every token  $s \in \mathcal{V}$  the highest-score token  $t$  it is mapped into according to  $M$ :  $t = \arg \max(\mathbf{m}_s)$ . Next, we define the output space size to be the portion of unique output tokens over the vocabulary. For instance, we expect the output space of a head that only maps strings to their first letters to be a small set of letter tokens. Similarly to the normalization of the saliency scores by the embedding norms, which we applied in §3.2, here, when calculating  $M$ , we normalize the *unembeddings* ( $U$ ’s columns).

Additionally, we present two baselines. The first baseline, dubbed “specific head”, represents the output space size of a head that maps the entire vocabulary to 1 specific token (e.g. a head that always outputs the end of sequence token). The second baseline, called “global head”, represents the output space size of a head that maps the entire vocabulary to capitalized tokens with leading

Head	Salient mappings	GPT-4o description
Pythia 6.9B 15.3	osevelt: 1943, 1941, 1940, 1930, 1936 Roosevelt: 1943, 1941, 1936, 1940, 1930 FDR: 1943, 1942, 1941, 1938, 1936 Napole: 1800, 1800, 18,18, 1840 oslov: 1968, 1970, 1960, 1964, 1965 Napoleon: 1800, 1800,18, 18, Napoleon taire: 1840, 1850,1800, Pruss, 1830 afka: 1908, 1912, 1916, 1903, 1911 lantern: 1870, 1880, 1930, Depression, railroad Edison: 1920,1920,1900, 1908, 1880 Confederate: 1863, 1864, 1861, 1862, 1870 1861: 1861, 1863, 1860, 1864, 1870	The input strings are partial or full names of historical figures as well as years and terms relating to historical events. The mappings associate each input with years or terms relevant to their historical significance, reflecting events or periods related to the input entity.
Pythia 6.9B 16.1	inhib: inhibition, inhib, Inhib, inhibiting, inhibit resil: resilience, resistance, Resp, res, resistance toler: toler, tolerance, tolerate, tolerated, tolerant aggrav: aggrav, exacerb, help, assistance : response, responses, responding, inhibiting destructive: destructive, destruction, destroying salvage: saving, save, saving, save, saves reluct: reluctance, resistance, resisting, resist prophyl: protection, protective, Protection Relief: relief, Relief, relie, relieved, relieve surv: survival, Survival, protection, surviv	The input strings are truncated forms of words, often found in contexts related to protection, resistance, or functionality. The mappings primarily expand truncated forms into semantically related words, often the full form of the input string or related words.
Pythia 6.9B 16.11	weeks: months, month, summer, season, year months: year, year, Year, Year, yearly month: year, Year, year, Year, yearly Month: year, Year, year, Year, years weeks: month, months, month, months, summer months: year, Year, year, Year, yearly Week: months, month, months, month, Month week: month, months, month, months, season month: year, Year, year, Year, yearly overnight: month, week, weeks, acci, months years: decade, decades, aging, century, life	The input strings are related to time periods such as weeks, months, and years. Mappings are connecting input strings to related or hierarchical time concepts, often extending them into longer periods like months to years and weeks to months.
Pythia 6.9B 22.13	periodontal: dental, Dental, dentist, dent, periodontal mandibular: dental, Dental, mandibular, teeth, dentist odontic: dental, Dental, dentist, teeth, tooth psori: skin, Skin, skin, dermat, skins retinal: eye, ophthal, retinal, ocular, eyes echocardiography: cardiac, Card, hearts, Card, Cardi scalp: brain, Brain, brain, brains, scalp hippocampal: hippocampal, Brain, brain, brain, hippocampus ocardi: cardiac, Card, hearts, Heart, heart ACL: knee, knees, thigh, Hip, ankle caries: dental, Dental, dentist, dent, Dent	The input strings seem to relate to various medical and anatomical terms, including parts of the body, diseases, and medical procedures. The mappings primarily associate anatomical or medical terms (input strings) with related medical terminology, such as conditions, associated body parts, or broader medical categories.
GPT-2 xl 26.2	Jedi: lightsaber, Jedi, Kenobi, droid, Skywalker lightsaber: lightsaber, Jedi, Kenobi, Skywalker, Sith galactic: Galactic, galactic, starship, galaxy, droid Starfleet: galactic, Starfleet, starship, Galactic, interstellar Klingon: starship, Starfleet, Klingon, Trek, Starship starship: starship, Galactic, galactic, interstellar, Planetary Skyrim: Skyrim, Magicka, Bethesda, Elven, Hearth Darth: Jedi, lightsaber, Kenobi, Darth, Sith galaxy: Galactic, galactic, starship, galaxy, droid	The input strings are terms related to popular science fiction and fantasy franchises such as Star Wars, Star Trek, Pokémon, Elder Scrolls, Harry Potter, and general fantastical terms. The pattern observed is that each mapping takes an input term from a science fiction or fantasy context and maps it to other terms that are often from the same or related fictional universe.

Table 25: Example salient operations of attention heads in Pythia 6.9B and GPT-2 xl and their corresponding descriptions by GPT-4o.

## Instructions

The goal of this task is to verify the correctness of GPT4 in inferring a relation or function from a list of demonstrations.

You will be given:

- **A list of 30 demonstrations** of some function that maps an input string into a list of 5 strings. Demonstrations are given in the format of "s: t1, t2, t3, t4, t5", where s is an input string and t1, t2, t3, t4, t5 are 5 strings to which s is being mapped. Each of s, t1, t2, t3, t4, t5 is a short string, typically corresponding to a single word or a sub-word.
- **A description generated by GPT4** of patterns it identified across the input strings and their mappings.

Examples are provided in [this spreadsheet](#).

Your task is the following:

- a. **Go over the input strings and their mappings and try to identify prominent patterns.** Patterns can be semantic, language-related, general or unnatural. It could be that you would not observe a clear pattern. We expect that, in most cases, the mappings will exhibit one pattern or no patterns.
- b. **Then answer the multi-choice questions below** to indicate the degree to which your assessment agrees with the description generated by GPT4.

**Q1:** Did GPT4 correctly identify the presence or lack of a pattern?

- 1: There is no observable pattern, and GPT4 indicated there is no pattern.
- 2: There is no observable pattern, but GPT4 described a pattern.
- 3: There is an observable pattern, and GPT4 indicated there is no pattern.
- 4: There is an observable pattern, and GPT4 described a pattern.

**Q2** (answer only if your answer to Q1 is 4): How precise is the description of GPT4?

- **Correct and accurate:** the description accurately describes the pattern, without errors.
- **Correct but inaccurate:** the description is correct overall, but is too general or abstract for the pattern expressed in the mappings. Alternatively, it is too specific or explicit and does not fully capture the general pattern.
- **Partially correct:** The description describes the correct pattern to some degree, but it also includes incorrect parts.
- **Poor:** the description does not describe the pattern at all.

**Q3:** (answer only if your answer to Q1 is 3 or 4): How would you categorise the most prominent pattern:

- Semantic
- Language
- General
- Unnatural

Figure 7: First part of human annotation instructions.

Important guidelines:

- In Q1, we consider that “GPT4 indicated there is no pattern” if it either responded with the word “Unclear”, or explained that there is no pattern in a sentence.
- In cases where the description of the model includes suggestive commentary about the hidden motivation for the function represented in the mappings (in addition to an explicit explanation), the commentary should not be considered. An example for a description which includes commentary is “*The mappings generally consist of repetitions or small variations of their corresponding input string's characters, suggesting a pattern related to breaking down or rearranging the input string*”.
- We consider a pattern *recognizable* when it is apparent across 20 or more mappings. We require that **at least one** of the following will hold:
  - The functionality behind the mappings (of input to output strings) will be visible and clear - for example, mappings of words to their first letters.
  - The destination strings will be highly related to each other - for example, cases where all the source strings are mapped to numbers.
- In cases where there is a mutual pattern encompassing **only** the source strings, we do not consider this as a recognizable pattern.
- In Q2 we use the terms *correct* and *accurate* to label the descriptions. *Correct* descriptions describe the mappings and do not include incorrect parts. *Correct* descriptions might be *accurate* or *inaccurate*. The *inaccuracy* metric refers to whether the descriptions are too general (or too specific).
- In Q3, the different mapping categories are:
  - *Semantic* - the mapping encodes semantic associations of the input strings (which might require knowledge). For example, associating countries with their capitals or languages.
  - *Language* - the mapping encodes a relationship which requires language knowledge (e.g. syntactic or lexical expertise) relationship. For example, mapping words to prefixes, or nouns to pronouns.
  - *General* - the mapping encodes a general functionality, which naturally can be applied to a large subset of strings. For example, mapping a string to itself, or a number to its successor/predecessor.
  - *Unnatural* - the mapping does **not** encode a recognizable/understandable function or relation, one that might be used for natural language processing (see examples of unnatural patterns in [the examples spreadsheet](#)).
- Please use the Notes column to add any information, insight or problem you find relevant.

Figure 8: Second part of human annotation instructions.

Below you are given a list of input strings, and a list of mappings: each mapping is between an input string and a list of 5 strings.

Mappings are provided in the format "s: t1, t2, t3, t4, t5" where each of s, t1, t2, t3, t4, t5 is a short string, typically corresponding to a single word or a sub-word.

Your goal is to describe shortly and simply the inputs and the function that produces these mappings. To perform the task, look for semantic and textual patterns.

For example, input tokens 'water','ice','freeze' are water-related, and a mapping ('fire':'f') is from a word to its first letter. As a final response, suggest the most clear patterns observed or indicate that no clear pattern is visible (write only the word "Unclear").

Your response should be a valid json, with the following keys:

- "Reasoning": your reasoning.
- "Input strings": One sentence describing the input strings (or "Unclear").
- "Observed pattern": One sentence describing the most clear patterns observed (or "Unclear").

The input strings are:  
<input strings>

The mappings are:  
<mapping strings>

Table 26: The prompt used to query GPT-4o. The salient tokens and mappings (§3.2), which are unique for every head, are plugged instead of <input strings> and <mapping strings>.

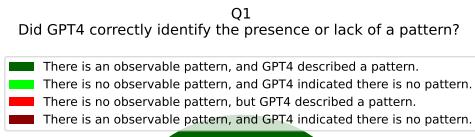
spaces - a subset whose size is 25% of the vocabulary of GPT-2 xl, and 16% of the vocabulary of Pythia 6.9B. An example of such a “global head” is a head that maps every word (or sub-word) in English to its capitalized version, and all other tokens to one specific token.

**Results** Figure 12 shows the input skewness and output space sizes for all heads in Pythia 6.9B and GPT-2 xl. In both models, the input skewness rises and then sharply decreases in the early layers, after which it stabilizes. This implies that attention heads in shallower layers induce a salient effect into a specific set of inputs compared to later layers. In contrast, the output space size generally decreases across layers with a slight increase in the final layers, suggesting that head outputs across layers converge to smaller token subsets. Taken together, we hypothesize that early layer heads demonstrate their functionality on fewer inputs than deeper heads, which in turn map a larger set of possible inputs to a small set of outputs.

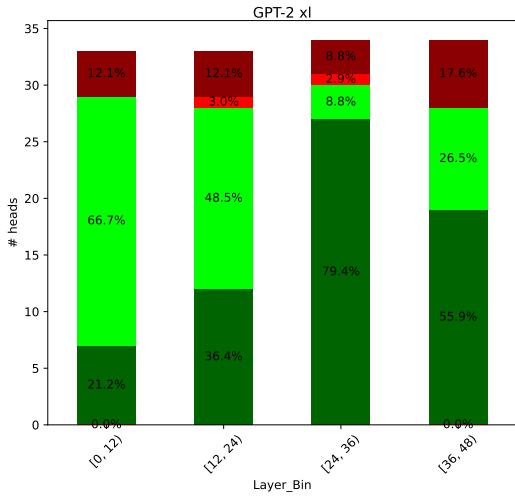
## G Resources and Packages

In our experiments, we used models and code from the transformers (Wolf et al., 2020) and TransformerLens (Nanda and Bloom, 2022) packages, and nanoGPT.<sup>10</sup> All the experiments were conducted using a single A100 80GB or H100 80GB GPU, aside from the experiments studying Llama-3.1 70B, which used nodes with 8 of these GPUs.

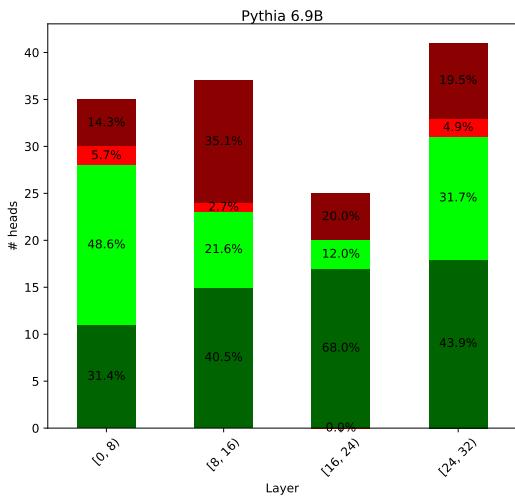
<sup>10</sup><https://github.com/karpathy/nanoGPT>



(a) Human annotation distribution for Question 1.

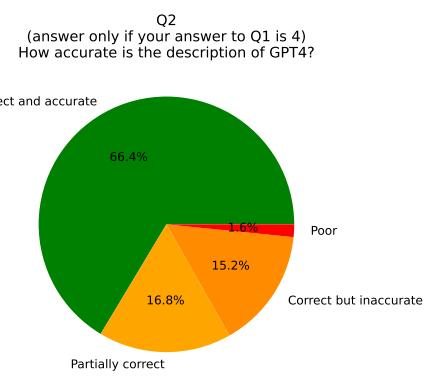


(b) Human annotation distribution for Question 1 across layers (GPT-2 xl).

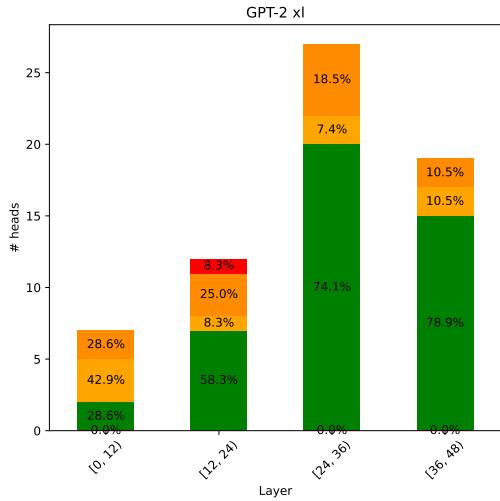


(c) Human annotation distribution for Question 1 across layers (Pythia 6.9B).

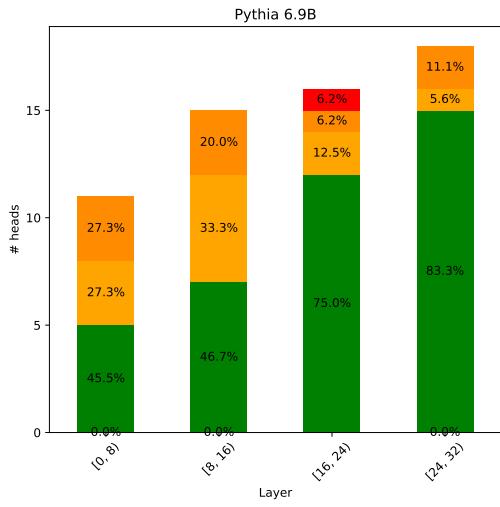
Figure 9: Quality of GPT-4o interpretation (§E) - Human annotation distribution for Question 1.



(a) Human annotation distribution for Question 2.



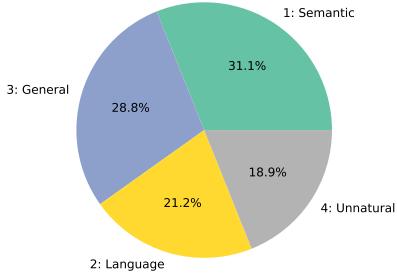
(b) Human annotation distribution for Question 2 across layers (GPT-2 xl).



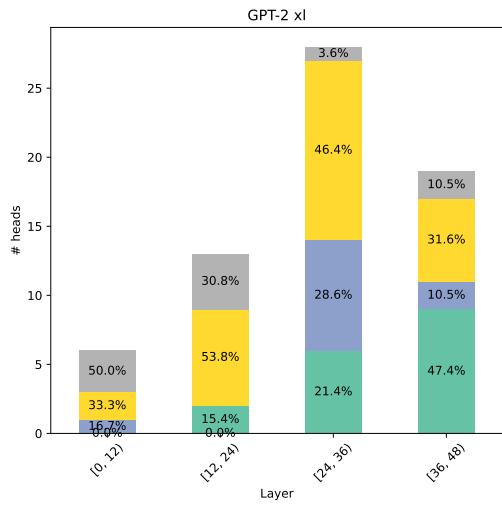
(c) Human annotation distribution for Question 2 across layers (Pythia 6.9B).

Figure 10: Quality of GPT-4o interpretation (§E) - Human annotation distribution for Question 2.

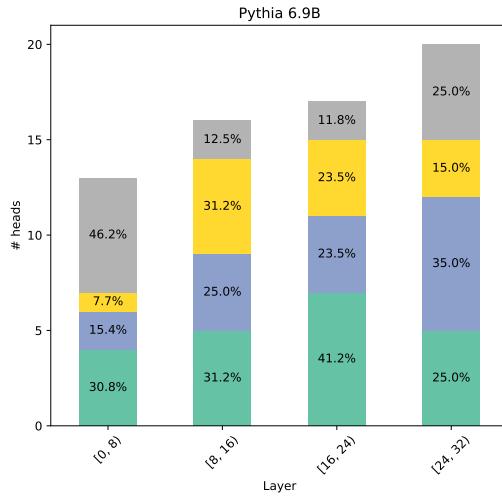
Q3  
 (answer only if your answer to Q1 is 3 or 4)  
 How would you categorise the most prominent pattern?



(a) Human annotation distribution for Question 3.



(b) Human annotation distribution for Question 3 across layers (GPT-2 xl).



(c) Human annotation distribution for Question 3 across layers (Pythia 6.9B).

Figure 11: Quality of GPT-4o interpretation (§E) - Human annotation distribution for Question 3.

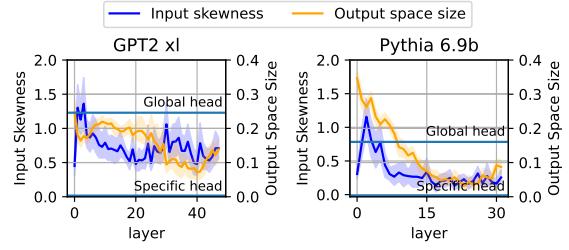


Figure 12: Input skewness versus output space size for all attention heads per layer in Pythia 6.9B and GPT-2 xl, compared to baseline heads of global and specific functionalities. Lower input skewness indicates a larger input space.