

Emergent Modularity in Pre-trained Transformers

Zhengyan Zhang^{1*}, Zhiyuan Zeng^{1*}, Yankai Lin^{2,3}, Chaojun Xiao¹, Xiaozhi Wang¹
Xu Han¹, Zhiyuan Liu^{1,4,5†}, Ruobing Xie⁶, Maosong Sun^{1,4†}, Jie Zhou⁶

¹NLP Group, DCST, IAI, BNRIST, Tsinghua University, Beijing

²Gaoling School of Artificial Intelligence, Renmin University of China, Beijing

³Beijing Key Laboratory of Big Data Management and Analysis Methods

⁴International Innovation Center of Tsinghua University, Shanghai

⁵Quan Cheng Laboratory ⁶Pattern Recognition Center, WeChat AI, Tencent Inc
{zy-z19, zengzy20}@mails.tsinghua.edu.cn {liuzy, sms}@tsinghua.edu.cn

Abstract

This work examines the presence of modularity in pre-trained Transformers, a feature commonly found in human brains and thought to be vital for general intelligence. In analogy to human brains, we consider two main characteristics of modularity: (1) *functional specialization of neurons*: we evaluate whether each neuron is mainly specialized in a certain function, and find that the answer is yes. (2) *function-based neuron grouping*: we explore finding a structure that groups neurons into modules by function, and each module works for its corresponding function. Given the enormous amount of possible structures, we focus on Mixture-of-Experts as a promising candidate, which partitions neurons into experts and usually activates different experts for different inputs. Experimental results show that there are functional experts, where clustered are the neurons specialized in a certain function. Moreover, perturbing the activations of functional experts significantly affects the corresponding function. Finally, we study how modularity emerges during pre-training, and find that the modular structure is stabilized at the early stage, which is faster than neuron stabilization. It suggests that Transformers first construct the modular structure and then learn fine-grained neuron functions. Our code and data are available at <https://github.com/THUNLP/modularity-analysis>.

1 Introduction

Recently, pre-trained Transformers have shown the potential to achieve general intelligence (Brown et al., 2021; Fei et al., 2022; Reed et al., 2022; OpenAI, 2023; Bubeck et al., 2023), which encourages researchers to explore the analogy between

Transformers and human brains (Toneva and Wehbe, 2019; Caucheteux et al., 2021; Caucheteux and King, 2022; Goldstein et al., 2022). These works have shown that the behaviors of Transformers resemble those of human brains. Do the internal structures of Transformers also mirror those of human brains in order to achieve similar behaviors?

Neuroscientists have found that the structure of neuron organization in human brains follows a modular pattern (Bullmore and Sporns, 2009; Meunier et al., 2010), which has two main characteristics. (1) **Functional specialization of neurons**: each neuron is mainly specialized in a certain function. (2) **Function-based neuron grouping**: neurons with the same function are clustered in a local region, and each function relies on a specific region. In this work, we wonder whether Transformers also organize neurons in a modular way.

(Q1) Are neurons functionally specialized?

The functional specialization of neurons is the basis of modularity. To this end, we propose a novel framework to analyze the functionality of neurons in Transformers. In this framework, we study three representative functions by a unified method, including semantic function (Scarlini et al., 2019a; Suau et al., 2020), knowledge function (Jiang et al., 2020; Dai et al., 2022), and task function (Wang et al., 2022b). Experimental results show that the neurons in pre-trained Transformers become much more specialized than those in randomly-initialized ones after self-supervised learning on large-scale corpora. Moreover, in pre-trained Transformers, there are several groups of neurons, each of which excels in a specific function.

(Q2) Is there a modular structure of neurons?

In analogy to human brains, a modular structure should group neurons with the same function together as modules, and each module plays a crucial role in a specific function. Since there are

* Equal contribution

† Corresponding authors

thousands of neurons in a Transformer, it is impractical to iterate over all possible structures of neurons. We consider the grouping of Mixture-of-Experts (MoE) (Jacobs et al., 1991; Fedus et al., 2022) as a promising candidate, which gracefully partitions neurons into experts and is widely used in Transformers. Moreover, most MoE models are sparsely activated, which is similar to human brains. Specifically, we study two types of MoE models, pre-partitioned MoE (pre-MoE) and post-partitioned MoE (post-MoE). Pre-MoE refers to the model architectures that expand feedforward layers by MoE to improve model capacity before pre-training (Fedus et al., 2022). Post-MoE refers to the models that are converted from vanilla Transformers to their MoE version by partitioning feedforward layers into experts after pre-training (Zhang et al., 2022b). It provides a way to group neurons without changes in parameters and forward pass.

By studying the function distribution on experts, we find that both pre-MoE and post-MoE Transformers have a strong tendency to distribute the neurons excelling in a certain function concentratedly into some experts. Moreover, perturbing the activations of 3% of experts by function leads to performance close to random guessing, which is more significant than perturbing the same amount of individual neurons specialized in the function. Therefore, the MoE structure indeed reflects the modularity of pre-trained Transformers.

(Q3) How does modularity emerge during pre-training? By analyzing the pre-training process, we find that the functions of expert networks are stabilized to a large extent at the early stage (around 15% of the total training steps) for both pre-MoE and post-MoE Transformers, which is faster than the neuron stabilization (around 75% of the total steps). Our findings provide evidence for a coarse-to-fine mechanism of pre-training, which first constructs the coarse modular structure and then learns fine-grained neuron functions.

2 Related Work

Interpreting Pre-trained Transformers. Although pre-trained Transformers have achieved great success in the field of NLP (Min et al., 2021; Bommasani et al., 2021), their inner working mechanism is still a black box. Researchers explore interpreting the pre-trained Transformers (Rogers et al., 2020) from different perspectives, such as hidden representations (Liu et al., 2019), attention matri-

ces (Voita et al., 2019; Clark et al., 2019b), and output distributions (Petroni et al., 2019). Among them, studying the behavior of a single neuron is an important branch (Radford et al., 2017; Sajjad et al., 2022; Bills et al., 2023), which is most related to our work. From the neurons in feedforward layers, researchers have found various encoded information such as concepts, facts, and task abilities (Suau et al., 2020; Dai et al., 2022; Wang et al., 2022b). In this work, we study how these neurons are organized to form a modular structure, which is a new interpretation perspective.

Modularity of Neural Networks. Modularity is a widespread property in complex systems, both artificial (Ballard, 1987; Baldwin et al., 2000) and biological (Von Dassow and Munro, 1999; Lorenz et al., 2011; Clune et al., 2013). Previous work mainly focuses on incorporating explicitly designed modules into neural networks (Andreas et al., 2016; Kirsch et al., 2018; Goyal et al., 2021). Recently, Hod et al. (2021); Csordás et al. (2021); Dobs et al. (2022) study whether standard neural networks become modular by themselves, and have discovered some naturally-emerging modular structures of CNNs and LSTMs. Compared to previous work, we extend the modular analysis to pre-trained Transformers, which are expected to be more complex w.r.t. architecture and to contain more knowledge.

Transformers with MoE. Sparse MoE is usually used to enlarge the model capacity of Transformers while keeping the computational efficiency (Lepikhin et al., 2021; Lewis et al., 2021; Fedus et al., 2022). Specifically, for a given input, MoE conditionally selects a subset of experts to process the input and then combines the outputs of these experts to generate the final output. Beyond computation efficiency, MoE is also used to implement modular Transformers (Gururangan et al., 2022; Zhang et al., 2022a; Pfeiffer et al., 2022; Wang et al., 2022a). These works explicitly design extra constraints during pre-training to ensure the modularization of expert networks. However, it is still unclear whether modular experts can emerge naturally in Transformers during pre-training.

3 Functionality Evaluation

In this section, we first introduce the definition of neurons and how to evaluate the functionality of neurons and experts. Then, we briefly introduce the evaluation setups, including the pre-trained models.

Neurons in Transformer. Following Dai et al. (2022); Wang et al. (2022b), we study the neurons in the feedforward networks (FFNs), which account for about two-thirds of Transformer parameters. Previous work has shown that there is rich information in FFNs (Suau et al., 2020; Geva et al., 2021, 2022; Dai et al., 2022; Wang et al., 2022b).

Specifically, the FFN is a two-layer MLP and computes the output by $\text{FFN}(\mathbf{x}) = \mathbf{W}^O \sigma(\mathbf{W}^I \mathbf{x} + \mathbf{b}^I) + \mathbf{b}^O$, where $\mathbf{W}^I \in \mathbb{R}^{d_{ff} \times d}$, $\mathbf{W}^O \in \mathbb{R}^{d \times d_{ff}}$ are the weight matrices, $\mathbf{b}^I \in \mathbb{R}^{d_{ff}}$, $\mathbf{b}^O \in \mathbb{R}^d$ are the bias vectors, d is the dimensions of input and output, d_{ff} is the dimensions of intermediate hidden states, and σ is the activation function. For simplicity, we discard the bias terms in the following part. For fine-grained analysis, we dissect the FFN into neurons and rewrite the FFN equation as

$$\text{FFN}(\mathbf{x}) = \sum_{i=1}^{d_{ff}} \sigma(\mathbf{W}_{i,:}^I \cdot \mathbf{x}) \mathbf{W}_{:,i}^O, \quad (1)$$

where $\mathbf{W}_{i,:}^I$ and $\mathbf{W}_{:,i}^O$ are the i -th row and column of \mathbf{W}^I and \mathbf{W}^O , respectively. The FFN output is the sum of the outputs of all neurons. From this perspective, we define a neuron n_i as a row vector $\mathbf{W}_{i,:}^I$ and a column vector $\mathbf{W}_{:,i}^O$. The neuron activation of n_i is $\sigma(\mathbf{W}_{i,:}^I \cdot \mathbf{x})$. The number of neurons in an FFN is equal to the intermediate hidden dimension d_{ff} .

Sparse Mixture-of-experts in Transformer is a variant of FFN (Lepikhin et al., 2021), which significantly increases the model capacity by adding more parameters and keeps the computational cost affordable. In MoE layers, each expert is an FFN, and the output of the MoE layer is the weighted sum of the outputs of all experts. We can also define neurons of MoE as Equation 1. The neuron-based form of MoE is provided in Appendix A.1.

Predictivity for Functions. To comprehensively study the functions of neurons and experts, we cover three typical functions. For each function, we construct various sub-functions, each of which is a fine-grained version of the function. There are 576 sub-functions in total. Here we introduce the three functions and their sub-functions.

Semantic Function. Semantic function refers to the ability to understand the meaning of input texts. In this work, we focus on how neurons capture the patterns of word senses. We use a large-scale dataset with word-sense annotations, OneSec (Scarlino et al., 2019b), to construct binary classification data for semantic sub-functions. In OneSec, each

sentence has a keyword whose sense is annotated based on Wikipedia¹. We first filter out the keywords that have more than one sense in the dataset and then randomly select 100 sentences for each sense. For each sense pair of a word, we construct a binary classification dataset by labeling the sentences with one sense as positive and the sentences with the other sense as negative. Finally, we have 529 semantic sub-functions, each of which is a binary classification problem to distinguish the two senses of a word. For experiments on Switch Transformer in Section 6, we randomly sample 100 semantic sub-functions for evaluation.

Knowledge Function. Knowledge function refers to the ability to memorize factual knowledge. In this work, we focus on the factual triples, which are used to construct knowledge graphs. We define a knowledge sub-function as a binary classification to identify whether a triple is correct. Specifically, we sample several triples from Wikidata as positive instances and randomly replace their head or tail entities to construct negative instances. We group these instances according to their relations and each relation has its corresponding knowledge sub-function. There are 39 knowledge sub-functions from T-REx (Elsahar et al., 2018; Elazar et al., 2021) and each sub-function has 400 instances.

Task Function. Task function refers to the ability to perform downstream tasks. Previous work has shown that training a small part of parameters in pre-trained Transformers can achieve comparable performance to full-parameter fine-tuning (Lester et al., 2021; Hu et al., 2022) so that Transformers are supposed to learn amounts of task knowledge from pre-training. In this work, we use several classification datasets. from GLUE (Wang et al., 2018), including SST-2 (Socher et al., 2013), QQP², MNLI (Williams et al., 2018), CoLA (Warstadt et al., 2018), MRPC (Dolan and Brockett, 2005), RTE (Dagan et al., 2006), QNLI (Rajpurkar et al., 2016). There are 8 task sub-functions in total because MNLI is split into two binary classification tasks. To stimulate these sub-functions, we adopt the input templates provided by (Raffel et al., 2020) to improve neuron predictivity. For each label class, we randomly sample 1000 instances for evaluation if the number of instances under this class is larger than 1000.

Admittedly, coarse is our function classifica-

¹<https://en.wikipedia.org/>

²<https://data.quora.com/>

First-Quora-Dataset-Release-Question-Pairs

tion. It does not cover all functions learned by pre-trained Transformers, and there are interactions between each pair of functions so there is unavoidable overlap. However, we focus on a unified framework and concrete evaluation approach, and they can be easily generalized to other ways of function classification, meaning that our contribution is independent of function classification. Using this way of classification is simply due to its typicality.

To evaluate the ability of a neuron to capture the pattern of a sub-function, we compute the predictivity of the neuron activations for the sub-function. Following [Suau et al. \(2020\)](#); [Wang et al. \(2022b\)](#), we focus on the sub-functions that can be formulated as a binary classification problem.

We denote the dataset of a sub-function as $\mathcal{D} = \{(s_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, where s_i is the input sequence and $y_i \in \{0, 1\}$ is the label. Since the computation of FFNs is word-wise, we define the activation of the neuron n_j of a sequence s_i as $a_{ij} = \max_{\mathbf{x}_k \in s_i} \sigma(\mathbf{W}_{j,:}^I \cdot \mathbf{x}_k)$, where $s_i = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l\}$ is the hidden states of s_i and l is the length of s_i . Then, we have the pairs of neuron activations and labels, $\mathcal{A}_j = \{(a_{ij}, y_i)\}_{i=1}^{|\mathcal{D}|}$. Based on \mathcal{A}_j , we compute the average precision (AP) ([Zhu, 2004](#)) of the neuron activations as the predictivity of the neuron n_j . For each expert, we compute the average AP of all neurons in the expert as the predictivity of the expert for the sub-function. Please refer to [Appendix A.2](#) for more details.

Evaluation Setups. We evaluate two kinds of MoE models, pre-partitioned MoE and post-partitioned MoE. (1) **Pre-partitioned MoE** expands feedforward layers by MoE before pre-training. Switch Transformer ([Fedus et al., 2022](#)) is a representative pre-MoE model. The architecture of Switch Transformer is similar to T5 ([Raffel et al., 2020](#)) except that Switch Transformer replaces FFNs in the even Transformer layer with MoE layers. (2) **Post-partitioned MoE** converts a pre-trained vanilla Transformer into its MoE model by partitioning FFNs into experts. Since [Zhang et al. \(2022b\)](#) show that vanilla Transformers have implicit MoE structures by discovering the inner correlation among neurons, we use the same method to MoEfy vanilla Transformers. Note that we only adopt MoEfication to provide a neuron grouping. We do not change any architecture, parameters, forward function (computing all neurons and then summing them up), etc, making the

MoE-fied T5 **identical** to the original T5.

Due to the computational cost, we consider the Switch Transformer with 16 experts and use the same number of experts for the post-MoE models. Since the functionality evaluation does not involve decoding, we only compute the neuron and expert predictivities of the encoders. Besides, we focus on the neurons in MoE layers for Switch Transformer to facilitate the modular analysis in the following sections.

4 Functional Specialization of Neurons

In this section, we analyze the functional specialization of neurons. First, we compare the neuron predictivities of pre-trained models with those of randomly-initialized models at the layer level, which will present a general picture of neuron functions. Second, we study how functions distribute on neurons in each layer.

Neuron predictivities of different layers. At each layer, we compute the best predictivity of neurons for each sub-function and then calculate the average best predictivity among all sub-functions in each function. For comparisons, we also evaluate randomly-initialized models. We report the results in the left part of [Figure 1](#).

From this figure, we have two observations. (1) The average best predictivities of pre-trained neurons are significantly higher than those of randomly-initialized neurons, indicating that the neurons have learned these functions from pre-training and the neurons with top-ranked predictivities indeed excel in corresponding sub-functions. (2) The best predictivity of the task function increases with the layer number while the best predictivities of the semantic and knowledge functions vary little across layers. It suggests that the task function may be more difficult than the semantic and knowledge functions so the higher layers are more suitable for learning the task function. Note that we evaluate the frozen pre-trained models on the tasks without fine-tuning or prompt-tuning and the pre-training data of Switch Transformer do not contain the tasks. Hence, we can exclude the possibility of optimization artifacts ([Durrani et al., 2021](#)), which may make the higher layers more related to the task function.

Distribution in each layer. We first identify the neurons with the top predictivity rankings for each sub-function as *sub-functional neurons* in each layer and then compute the overlap between

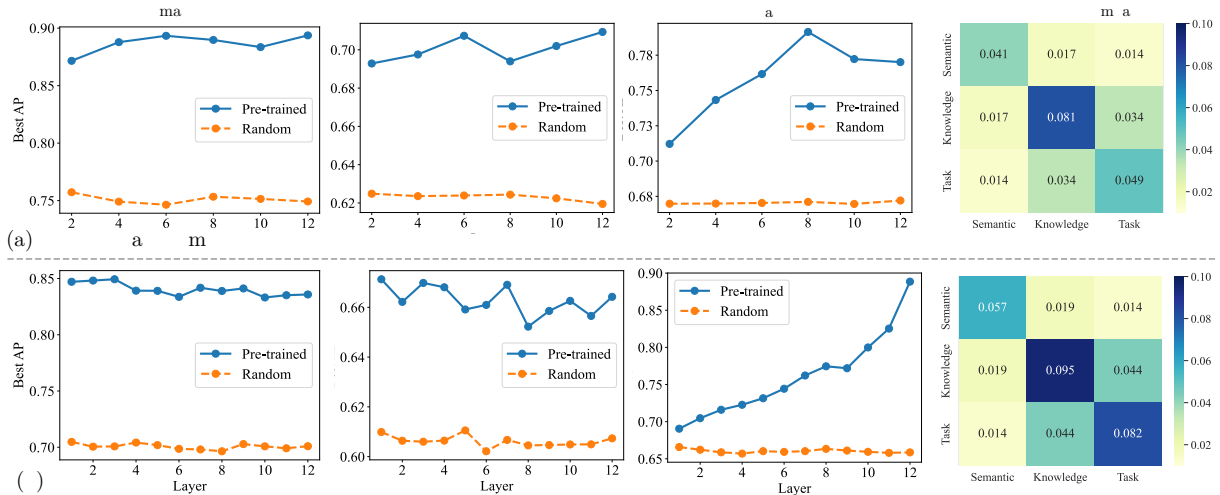


Figure 1: Left part: average predictivity of each function of each layer for the pre-trained models and their randomly-initialized counterparts of (a) Switch Transformer and (b) T5. Right part: average distribution similarity between different functions of different layers for the same pre-trained models. All of the similarities of random-initialized models are around 0.01, including the diagonal elements, which are significantly lower than those of the pre-trained models.

the two sets of sub-functional neurons. Formally, assuming that we identify the top k neurons for each sub-function, the overlap score is defined as $\frac{\sum_{n_i \in \mathcal{N}_1} \mathbb{I}(n_i \in \mathcal{N}_2)}{k}$, where \mathcal{N}_1 and \mathcal{N}_2 are the sets of neurons for the two considered sub-functions. If the overlap score is high, there is a group of neurons that are good at both sub-functions. In the experiments, we consider the neurons with the top 1% predictivities for each sub-function. Since there are hundreds of sub-functions, it is impossible to display all of them in a figure and we compute the average overlap score between two functions to measure the distribution similarity between different functions. Note that we omit the self-overlap scores, which are always equal to 1. The results are reported in the right part of Figure 1.

From this figure, we observe that: (1) In the pre-trained models, the distribution similarity of the same function is significantly larger than that of different functions, which indicates that there are some groups of neurons, each of which is good at a certain function. And, it is not mainly caused by the similarity that naturally exists between sub-functions of the same function because the results of randomly-initialized models do not show such a phenomenon as shown in Appendix A.3. Hence, we conclude that **there are some emergent groups of functionally-specialized neurons after pre-training**. (2) One neuron may be capable of multiple sub-functions even from different functions. For example, the average overlap score between

the knowledge function and task function is also significantly higher than that of random models so there are some neurons good at both knowledge and task sub-functions.

5 Finding Modular Structure of Neurons

Since MoE is a promising candidate for the modular structure of Transformers, we analyze the MoE structure in this section. First, we verify whether the neurons specialized in a certain function are concentrated in some experts, called functional experts. Second, we perturb the activations of experts corresponding to a certain function. Third, we train the model on different datasets, retaining only functional experts by removing non-functional experts. The last two experiments verify the importance of the experts for their corresponding functions.

Distribution on experts. If experts were not functionally specialized, the sub-functional neurons would be randomly distributed among experts. Hence, we conduct statistical hypothesis testing to evaluate whether the neuron distribution on experts is significantly different from random. Assume that there are N neurons in a layer, n_E neurons in each expert, k sub-functional neurons for each sub-function, and M sub-functions in a certain function. The null hypothesis is that the sub-functional neurons of each sub-function are independently and randomly distributed on experts, i.e., the number of sub-functional neurons in each expert follows a hypergeometric distribution with parameters N , K ,

Model	Pre-trained	Partitioning	Semantics		Knowledge		Task	
			Prop.	Degree	Prop.	Degree	Prop.	Degree
Switch Transformer	✗	Random	0.002	0.037	0.001	0.024	0.001	0.018
	✓	Random	0.226	1.038	0.052	0.652	0.003	0.066
	✓	Pre-MoE	0.354	1.490	0.260	1.560	0.219	1.604
T5	✗	Random	0.016	0.257	0.001	0.031	0.001	0.017
	✓	Random	0.252	1.203	0.061	1.031	0.007	0.221
	✓	Post-MoE	0.338	2.000	0.214	2.686	0.120	3.276

Table 1: Proportion of functional experts to all experts and their modularization degree. Higher modularization degrees mean that functional experts have more sub-functional neurons than the expectation of uniform distribution.

and n_E . The sum of the numbers of sub-functional neurons on all sub-functions in an expert is denoted by r_i .³ The alternative hypothesis is that an expert has a larger r_i than expected by chance.

In the experiments, we also treat the neurons with the highest 1% predictivities for each sub-function as its sub-functional neurons. For each function, we compute the p-value of the sum of the hypergeometric distribution for each expert and reject the null hypothesis if the p-value is less than 0.001. We also conduct the same experiment on random partitioning, where the neurons are randomly partitioned into expert-sized clusters, and randomly-initialized counterparts. We regard the experts that reject the null hypothesis as *functional experts* and report the proportion of functional experts to all experts in each function. And, we also consider the modularization degree, which is defined as the relative ratio of functional neurons in the expert compared to a uniform distribution. The proportion of functional experts and the modularization degree are computed by

$$\text{Prop.} = \frac{E_f}{E}, \quad \text{Degree} = \frac{r_i}{n_E} / \left(\frac{Mk}{N} \right), \quad (2)$$

where E_f is the number of functional experts, E is the number of experts, $\frac{r_i}{n_E}$ is the proportion of functional neurons in the expert, and $\frac{Mk}{N}$ is the proportion expectation of functional neurons under a uniform distribution. The overall degree is 0 if no functional expert exists, and otherwise is the average degree among all functional experts.

The results are shown in Table 1. From this table, we have three observations. (1) Since the sub-function distributions of pre-trained models are not independent, the proportion of functional

³We do not find a general form for the distribution of the sum of independent hypergeometric distributions. Since K is significantly smaller than N , we approximate the hypergeometric distribution with a binomial distribution.

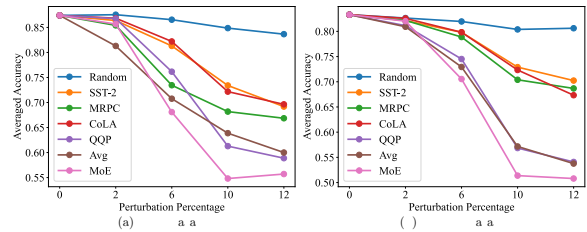


Figure 2: Perturbation performance of T5. For “Random”, we randomly perturb neurons. For “SST-2”, “MRPC”, “CoLA”, “QQP”, we are guided by the neuron predictivities on each dataset and perturb the top-ranked neurons. For “Avg”, we sum the predictivities on all datasets above and also perturb the top-ranked neurons. For “MoE”, we consider the experts with top-ranked sums of the predictivities. The seen datasets are the four datasets above. The unseen datasets include five other datasets.

experts of pre-trained models with random partitioning is higher than that of random-initialized counterparts. (2) However, the proportion of functional experts of pre-trained models with MoE partitioning is still higher than that with random partitioning. Moreover, the modularization degree of the functional experts in MoE structures is significantly higher than that in random partitioning. It indicates that the experts of both pre-MoE and post-MoE are more likely to intensively include neurons excelling in a certain function. (3) We further compare the predictivities of the functional experts and non-functional experts and find that the functional experts have significantly higher predictivities than the non-functional experts in their corresponding functions. It indicates that our quantification for expert predictivities is consistent with the concept of functional experts. More details are in Appendix A.4.

Perturbation analysis. Furthermore, we conduct perturbation experiments, which are widely used to analyze both biological and artificial neural

networks (Cowley et al., 2022; Wang et al., 2022b), to evaluate the causal effect of experts on model performance. Since T5 is a dense model and Switch Transformer is a sparse model, we use different perturbation methods for them. Specifically, the pre-MoE models only select one expert at each layer, so we choose to perturb the selection function. For the post-MoE models, which select all experts at each layer, we perturb the activation values of the targeting experts. Due to the difference in experiment setups, the perturbation performance is also different and their perturbation results are not comparable.

For T5, we perturb all neuron activations of the target experts by adding random noises to them and evaluate the perturbed models on the downstream tasks. We rank experts according to their sum of the predictivities for several downstream datasets and perturb the top-ranked experts. We regard the datasets used in computing the sum of the predictivities as seen datasets, including SST-2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), CoLA (Warstadt et al., 2018), and QQP. To evaluate the generalization ability of the experts, we also perturb them and evaluate the perturbed models on unseen datasets, including MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), CB (De Marneffe et al., 2019), MultiRC (Khashabi et al., 2018), and BoolQ (Clark et al., 2019a). We compute the average performance of the perturbed models on the seen and unseen datasets, respectively. For comparisons, we also conduct neuron-level perturbation and keep the proportion of perturbed neurons equal to that of expert-level perturbation. There are three kinds of neuron-level perturbations: (1) perturb the neurons that have top-ranked predictivities for a certain dataset, (2) perturb the neurons that have top-ranked sums of the predictivities for seen datasets, and (3) perturb the neurons randomly. We perturb the neurons in the last four layers where the task function is mainly located as shown in Figure 1. For fine-grained perturbation percentage, we partition FFNs into 96 experts, which is different from that of the other experiments.

We report the average accuracy of the perturbed models on the downstream tasks in Figure 2. From this figure, we have three observations. (1) The experts with high predictivities are very important for the model performance. For example, perturbing 10% of the experts by function in the

	Original	No Function	Function
7-task Avg.	73.23	72.78	74.45

Table 2: Perturbation performance of Switch Transformer on GLUE tasks.

last four layers (around 3% of the total experts in the model) decreases the average accuracy by nearly 30% and makes the model perform as random guessing. (2) For neuron-level perturbations, ‘‘Avg’’ perturbation achieves a larger performance drop than single-dataset perturbation, which is expected because intuitively it perturbs the neurons with high overall predictivities. (3) **Perturbing experts leads to a more significant performance drop than perturbing individual neurons** on both seen and unseen datasets when the perturbation proportion is higher than 6%. It suggests neurons in the experts cooperate instead of working independently so perturbing them will influence the cooperation and lead to a significant performance drop. We conclude single neurons can not perform a function well in lack of the cooperation with modules despite their high overall predictivity.

For Switch Transformer, in the last four layers, we perturb the selection probabilities of experts. We constraint the model to select from a subset of experts by setting the selection probabilities of the other experts to 0 and fine-tune the perturbed models on the downstream tasks with the template in Gao et al. (2021). There are two kinds of perturbations: (1) No Function: we only select the non-functional experts in a certain task, which is equivalent to setting the activations of the functional experts to 0. (2) Function: we only select the functional experts in a certain task. Since the number of functional experts is smaller than that of non-functional experts, we randomly select a subset of non-functional experts for No Function to make the size of the subset equal to that of the functional experts. Besides, we also report the performance of the original model without any perturbation. More details are in Appendix A.2. From Table 2, we observe that avoiding the functional experts leads to an overall performance drop. Furthermore, only selecting the functional experts even achieves a higher performance than the original model.

In summary, we observe that the specialized neurons tend to be located concentratedly in some experts based on function and the functional experts play an important role when the model performs related functions. Hence, it is reasonable to study

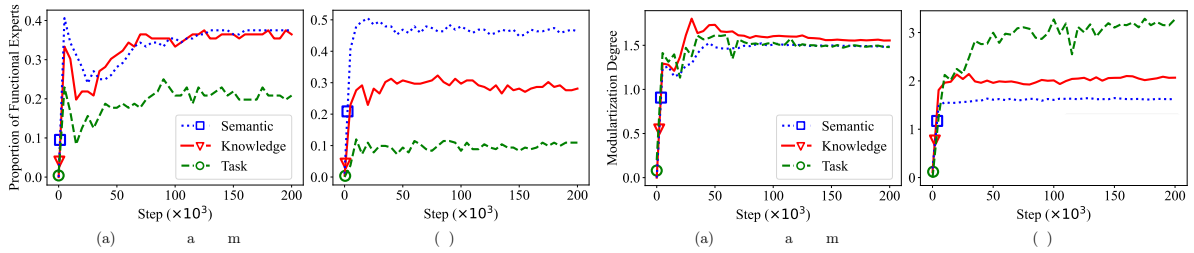


Figure 3: Changing curves of the proportion of functional experts and their modularization degree. We also mark the value of the last checkpoint with random partitioning on the curve by points with different shapes.

the modularity of Transformers by MoE.

6 Emergence of Modularity

In this section, we study the emergence of modularity during pre-training. To this end, we pre-train the base version of T5 and Switch Transformer from scratch. We use the MoEfication partitioning of the last checkpoint as the MoE structure of T5. More details of pre-training are in Appendix A.2.

Emergence Patterns of Functional Experts.

We first study the changing curves of the proportion of functional experts and their modularization degree during pre-training, i.e., we apply the same analysis in Section 5 to each checkpoint. The results are shown in Figure 3. We have the following observations. (1) The proportion of functional experts and their modularization degree quickly achieves a high point, and then keeps relatively stable till the end. It indicates that functional experts emerge at the early stage of pre-training. (2) The proportion of functional experts in the Switch Transformer fluctuates significantly at about 20K steps and its stabilization is slower than that of T5. It suggests that the emergence of the modular structure in Switch Transformer is surprisingly more difficult than T5. The reason may be that Switch Transformer omits the gradients of unselected experts, which causes the optimization to be harder than that of T5 (Du et al., 2022; Zoph et al., 2022).

Stabilization of Experts and Neurons. Even though clear is the changing curve of the number and modularization degree of functional experts from a global perspective, we still do not know how the predictivities of neurons and experts changes.

There are two kinds of predictivity dynamics. The first is the changing of the absolute predictivities, and the second is the relative order changing of predictivities among all experts or neurons. While it is straightforward to study the absolute predictivities, it is difficult to have a consistent analysis standard due to different scales for different func-

tions and layers. Hence, we focus on the relative order changing of predictivities. Intuitively, some experts or neurons may excel in a sub-function at a certain stage, and maintain this relative dominance as pre-training continues. With this in mind, we study the stabilization of predictivity rankings.

To study the stabilization of predictivity rankings, we quantify the similarity between a layer of two model checkpoints w.r.t. a particular sub-function, which is either at the expert level or at the neuron level. Specifically, for a sub-function, we define such a similarity as Spearman’s rank correlations (Spearman, 1961) between the predictivities of experts or neurons in the considered layer of the two checkpoints. In this way, we measure to what extent the predictivities of the two checkpoints is positively correlated. We measure the similarity between two adjacent (saved) checkpoints as stabilization score, which reflects the trend toward stabilization. Higher similarity indicates a lower changing pace and thus a higher degree of stabilization. For each function, we show the curve of average stabilization score among all sub-functions in it and across all layers, both at the expert level and neuron level. To facilitate our analysis, we also measure it on random partitioning.

We report the result in Figure 4. From this figure, we have four observations. (1) During the pre-training, both experts and neurons are increasingly stabilized. (2) Experts are stabilized to a large extent at the early stage of pre-training. It takes around 15% of the total training steps for the expert predictivities to achieve a stabilization score of 0.9. (3) Expert stabilization is notably faster than both neuron stabilization (around 75% of the total training steps) and the stabilization for random partitioning⁴. In conclusion, we see strong evidence that **coarse-to-fine is the inner mechanism of pre-training**. Transformer first learns a modular

⁴While the numbers of experts and neurons are different, the correlation scores are comparable (De Winter et al., 2016).

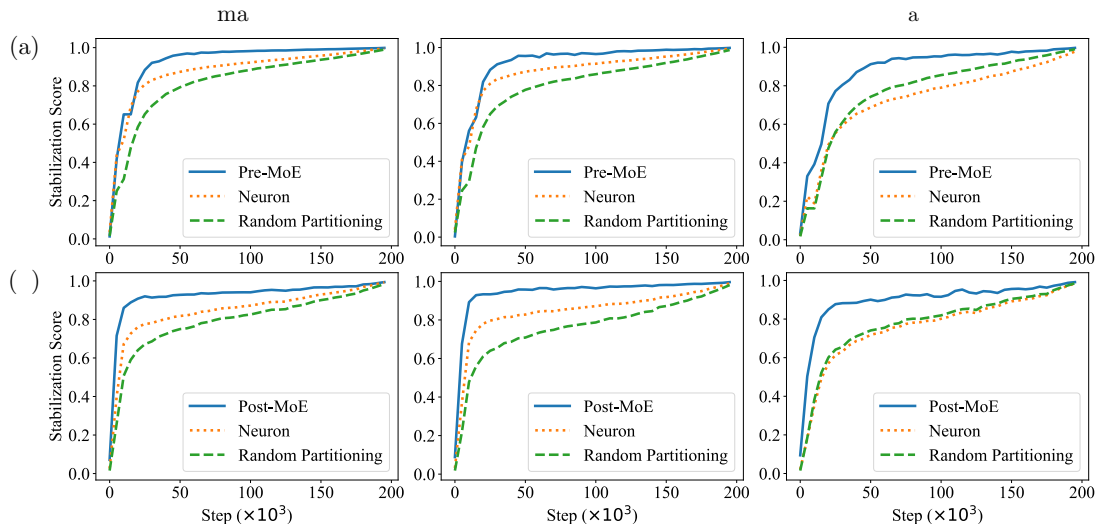


Figure 4: Spearman’s rank correlation between the functionality distributions of two adjacent checkpoints.

structure, where the structure becomes stable at the early stage, and then there is a fine-grained process to learn neuron functions.

7 Discussion

Efficient Pre-training. Large-scale pre-training of Transformers is very expensive (Brown et al., 2021). It is promising to use MoE to reduce the computational cost by activating only a small part of the experts. Our findings have shown the emergent modularity of experts, which demonstrates the reasonableness of the MoE structure. However, as shown in Section 6, Switch Transformer is more unstable than T5 on the modular structure at the beginning. It suggests that we should begin with a dense model and gradually make it sparse instead of directly training a sparse model from scratch, which has been explored in some preliminary works (Nie et al., 2021; Hazimeh et al., 2021).

Model Fusion. Considering there are many pre-trained models on different corpora and even modalities, researchers have started to explore how to fuse them to aggregate different knowledge together. Compared with model ensembling, model fusion has the potential to be more efficient because it does not compute all of the models. Much of the current research in model fusion focuses on weight averaging and achieves some promising results (Li et al., 2022; Matena and Raffel, 2021). However, weight averaging requires two models having the same architecture, which is not always the case. In this work, we discover the modular structure of pre-trained Transformers, which may facilitate the model fusion based on module combinations,

which gets rid of the architecture constraint.

Connection between Brains and Pre-trained Transformers. Building an artificial brain that corresponds to the human brain is an important neuroscience problem, e.g., the Blue Brain project (Markram, 2006). Currently pre-trained Transformers show strong power for predicting brain signals (Toneva and Wehbe, 2019; Caucheteux et al., 2021), but more fine-grained connections between the two are still not clear. In analogy to brain regions, we present the modular structure of pre-trained Transformers. It will be interesting to explore the connection between brain regions and the Transformer modules in the future.

8 Conclusion

In this paper, we study the modularity of pre-trained Transformers and validate the emergence of modularity. We also study the pre-training process to understand the emergence of modularity and find the coarse-to-fine mechanism of pre-training. We expect our evaluation framework and findings will facilitate and inspire future research in this area.

Limitations

The major limitations of our work are as follows: (1) We show that the neuron structure of MoE reveals the presence of modularity in pre-trained Transformers. However, the MoE structure is not the only possible modular structure. To better understand the modular structure of Transformers, we need to explore more types of structures. For example, the number of neurons in each module could be different, and the modular structure could

be hierarchical, where modules are grouped into larger modules. (2) We study three typical functions for language processing: semantic function, knowledge function, and task function. There are many other functions that could be studied, such as the syntactic function, discourse function, etc. Moreover, our categorization of functions may be not suitable for pre-trained Transformers because there are some overlaps between studied functions. A new Transformer-based function categorization may be needed. (3) We transform T5 into its MoE version to study its modular structure while not all dense pre-trained Transformers can be studied in this way because the adopted MoEification technique (Zhang et al., 2022b) can only transform ReLU-based Transformers. Studying the modularity of other dense pre-trained Transformers, such as BERT, is also important for future research.

Acknowledgements

This work is supported by the National Key R&D Program of China (No. 2020AAA0106502) and Institute Guo Qiang at Tsinghua University.

Author Contributions Zhengyan Zhang and Zhiyuan Zeng wrote the code and conducted the experiments. Zhengyan Zhang and Zhiyuan Zeng wrote the initial draft. Yankai Lin, Chaojun Xiao, Xiaozhi Wang, Xu Han, and Zhiyuan Liu, Ruobing Xie significantly edited and improved the paper. Maosong Sun and Jie Zhou provided valuable advice to the research.

References

- Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. [Neural module networks](#). In *Proceedings of CVPR*, pages 39–48.
- Carliss Young Baldwin, Kim B Clark, Kim B Clark, et al. 2000. *Design rules: The power of modularity*, volume 1. MIT press.
- Dana H. Ballard. 1987. [Modular learning in neural networks](#). In *Proceedings of AAAI*, pages 279–284.
- Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. [Language models can explain neurons in language models](#). <https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html>.
- Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, et al. 2021. [On the opportunities and risks of foundation models](#). *arxiv preprint arXiv:2108.07258*.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2021. [Language models are Few-Shot learners](#). In *Proceedings of NeurIPS*, pages 1877–1901.
- Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. 2023. [Sparks of artificial general intelligence: Early experiments with GPT-4](#). *arxiv preprint arXiv:2303.12712*.
- Ed Bullmore and Olaf Sporns. 2009. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3):186–198.
- Charlotte Caucheteux, Alexandre Gramfort, and Jean-Remi King. 2021. [Disentangling syntax and semantics in the brain with deep networks](#). In *Proceedings of ICML*, volume 139, pages 1336–1348.
- Charlotte Caucheteux and Jean-Rémi King. 2022. Brains and algorithms partially converge in natural language processing. *Communications biology*, 5(1):1–10.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019a. [BoolQ: Exploring the surprising difficulty of natural yes/no questions](#). In *Proceedings of NAACL-HLT 2019*.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019b. [What does BERT look at? an analysis of bert’s attention](#). In *Proceedings of BlackboxNLP*, pages 276–286.
- Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson. 2013. The evolutionary origins of modularity. *Proceedings of the Royal Society b: Biological sciences*, 280(1755):20122863.
- Benjamin R. Cowley, Adam J. Calhoun, Nivedita Rangarajan, Jonathan W. Pillow, and Mala Murthy. 2022. [One-to-one mapping between deep network units and real neurons uncovers a visual population code for social behavior](#). *bioRxiv*.
- Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. 2021. [Are neural nets modular? inspecting functional modularity through differentiable weight masks](#). In *Proceedings of ICLR*.

- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising textual entailment*, pages 177–190. Springer.
- Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. [Knowledge neurons in pretrained transformers](#). In *Proceedings of ACL*, pages 8493–8502.
- Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The CommitmentBank: Investigating projection in naturally occurring discourse.
- Joost CF De Winter, Samuel D Gosling, and Jeff Potter. 2016. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological methods*, 21(3):273.
- Katharina Dobs, Julio Martinez, Alexander JE Kell, and Nancy Kanwisher. 2022. Brain-like functional specialization emerges spontaneously in deep neural networks. *Science advances*, 8(11):eab18913.
- William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing*.
- Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P. Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. [Glam: Efficient scaling of language models with mixture-of-experts](#). In *Proceedings of ICML*, pages 5547–5569.
- Nadir Durrani, Hassan Sajjad, and Fahim Dalvi. 2021. [How transfer learning impacts linguistic knowledge in deep NLP models?](#) In *Findings of ACL/IJCNLP 2021*, pages 4947–4957.
- Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard H. Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. [Measuring and improving consistency in pretrained language models](#). *Trans. Assoc. Comput. Linguistics*, 9:1012–1031.
- Hady Elsahar, Pavlos Vougiouklis, Arslan Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. [T-REx: A large scale alignment of natural language with knowledge base triples](#). In *Proceedings LREC*.
- William Fedus, Barret Zoph, and Noam Shazeer. 2022. [Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity](#). *Journal of Machine Learning Research*, 23(120):1–39.
- Nanyi Fei, Zhiwu Lu, Yizhao Gao, Guoxing Yang, Yuqi Huo, Jingyuan Wen, Haoyu Lu, Ruihua Song, Xin Gao, Tao Xiang, et al. 2022. Towards artificial general intelligence via a multimodal foundation model. *Nature Communications*, 13(1):1–13.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. [Making pre-trained language models better few-shot learners](#). In *Proceedings of ACL/IJCNLP*, pages 3816–3830.
- Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. [Transformer feed-forward layers are key-value memories](#). In *Proceedings of EMNLP*, pages 5484–5495.
- Ariel Goldstein, Avigail Dabush, Bobbi Aubrey, Mariano Schain, Samuel A Nastase, Zaid Zada, Eric Ham, Zhuoqiao Hong, Amir Feder, Harshvardhan Gazula, et al. 2022. Brain embeddings with shared geometry to artificial contextual embeddings, as a code for representing language in the human brain. *bioRxiv*.
- Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. 2021. [Recurrent independent mechanisms](#). In *Proceedings of ICLR*.
- Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah Smith, and Luke Zettlemoyer. 2022. [Demix layers: Disentangling domains for modular language modeling](#). In *Proceedings of NAACL-HLT*, pages 557–5576.
- Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed H. Chi. 2021. [Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning](#). In *Proceedings of NeurIPS*, pages 29335–29347.
- Shlomi Hod, Stephen Casper, Daniel Filan, Cody Wild, Andrew Critch, and Stuart Russell. 2021. [Detecting modularity in deep neural networks](#). *arXiv preprint arXiv:2110.08058*.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [Lora: Low-rank adaptation of large language models](#). In *Proceedings of ICLR*.
- Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. [Adaptive mixtures of local experts](#). *Neural Comput.*, 3(1):79–87.

- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. [How can we know what language models know](#). *Trans. Assoc. Comput. Linguistics*, 8:423–438.
- Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *Proceedings of ICLR*.
- Louis Kirsch, Julius Kunze, and David Barber. 2018. [Modular networks: Learning to decompose neural computation](#). In *Proceedings of NeurIPS*, pages 2414–2423.
- Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. [Gshard: Scaling giant models with conditional computation and automatic sharding](#). In *Proceedings of ICLR*.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of EMNLP*, pages 3045–3059.
- Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. 2021. [BASE layers: Simplifying training of large, sparse models](#). In *Proceedings of ICM*, volume 139, pages 6265–6274.
- Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2022. [Branch-train-merge: Embarrassingly parallel training of expert language models](#). *arXiv preprint arXiv:2208.03306*.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. [Linguistic knowledge and transferability of contextual representations](#). In *Proceedings of NAACL-HLT*, pages 1073–1094.
- Dirk M Lorenz, Alice Jeng, and Michael W Deem. 2011. The emergence of modularity in biological systems. *Physics of life reviews*, 8(2):129–160.
- Henry Markram. 2006. [The blue brain project](#). In *Proceedings of SC*.
- Michael Matena and Colin Raffel. 2021. [Merging models with fisher-weighted averaging](#). *arXiv preprint arXiv:2111.09832*.
- David Meunier, Renaud Lambiotte, and Edward T Bullmore. 2010. Modular and hierarchically modular organization of brain networks. *Frontiers in neuroscience*, 4:200.
- Bonan Min, Hayley Ross, Elicor Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2021. [Recent advances in natural language processing via large pre-trained language models: A survey](#). *arXiv preprint arXiv:2111.01243*, abs/2111.01243.
- Xiaonan Nie, Shijie Cao, Xupeng Miao, Lingxiao Ma, Jilong Xue, Youshan Miao, Zichao Yang, Zhi Yang, and Bin Cui. 2021. [Dense-to-sparse gate for mixture-of-experts](#). *arXiv preprint arXiv:2112.14397*.
- OpenAI. 2023. [GPT-4 technical report](#). *arXiv preprint arXiv:2303.08774*.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. [Language models as knowledge bases?](#) In *Proceedings of EMNLP-IJCNLP*, pages 2463–2473.
- Jonas Pfeiffer, Naman Goyal, Xi Victoria Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. [Lifting the curse of multilinguality by pre-training modular transformers](#). In *Proceedings of the NAACL-HLT*, pages 3479–3495.
- Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2017. [Learning to generate reviews and discovering sentiment](#). *arXiv preprint arXiv:1704.01444*.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. [Exploring the limits of transfer learning with a unified Text-to-Text transformer](#). *J. Mach. Learn. Res.*, 21:140:1–140:67.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*, pages 2383–2392. Association for Computational Linguistics.
- Scott E. Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. 2022. [A generalist agent](#). *arXiv preprint arXiv:2205.06175*.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. [A primer in bertology: What we know about how BERT works](#). *Trans. Assoc. Comput. Linguistics*, 8:842–866.
- Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2022. [Neuron-level interpretation of deep NLP models: A survey](#). *TACL*, 10:1285–1303.

- Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2019a. [Just “OneSeC” for producing multilingual sense-annotated data](#). In *Proceedings of ACL*, pages 699–709.
- Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2019b. [Just “onesec” for producing multilingual sense-annotated data](#). In *Proceedings of ACL*, pages 699–709.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642.
- Charles Spearman. 1961. The proof and measurement of association between two things.
- Xavier Suau, Luca Zappella, and Nicholas Apostoloff. 2020. [Finding experts in transformer models](#). *arXiv preprint arXiv:2005.07647*.
- Mariya Toneva and Leila Wehbe. 2019. [Interpreting and improving natural-language processing \(in machines\) with natural language-processing \(in the brain\)](#). In *Proceedings of NeurIPS*, pages 14928–14938.
- Elena Voita, David Talbot, Fedor Moiseev, Rico Senrich, and Ivan Titov. 2019. [Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned](#). In *Proceedings of ACL*, pages 5797–5808.
- George Von Dassow and Ed Munro. 1999. Modularity in animal development and evolution: elements of a conceptual framework for evodevo. *Journal of Experimental Zoology*, 285(4):307–325.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. [Glue: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of EMNLP*, pages 353–355.
- Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. 2022a. [Image as a foreign language: Beit pretraining for all vision and vision-language tasks](#). *arXiv preprint arXiv:2208.10442*.
- Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022b. [Finding skill neurons in pre-trained transformer-based language models](#). In *Proceedings of EMNLP*.
- Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2018. Neural network acceptability judgments. *arXiv preprint 1805.12471*.
- Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*.
- Fan Zhang, Duyu Tang, Yong Dai, Cong Zhou, Shuangzhi Wu, and Shuming Shi. 2022a. [Skillnet-nlu: A sparsely activated model for general-purpose natural language understanding](#). *arXiv preprint 2203.03312*.
- Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022b. [MoEfication: Transformer feed-forward layers are mixtures of experts](#). In *Findings of ACL*.
- Mu Zhu. 2004. Recall, precision and average precision.
- Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. [Designing effective sparse expert models](#). *arxiv preprint arXiv:2202.08906*.

A Appendix

A.1 Neurons of MoE

In MoE layers, each expert is an FFN, and the output of the MoE layer is the weighted sum of the outputs of all experts, $\text{MoE}(\mathbf{x}) = \sum_{i=1}^E \alpha_i \text{FFN}_i(\mathbf{x})$, where α_i is the weight of the i -th expert and FFN_i is the i -th expert. α_i is computed by a gating network. Note that the weights of unselected experts are zero, which makes the MoE layer sparse. We can also rewrite the MoE layer into a neuron-based form,

$$\begin{aligned} \text{MoE}(\mathbf{x}) &= \sum_{i=1}^E \alpha_i \sum_j \sigma(\mathbf{W}_{i,j}^I \cdot \mathbf{x}) \mathbf{W}_{i,j}^O \\ &= \sum_{i,j} \sigma(\mathbf{W}_{i,j}^I \cdot \mathbf{x}) \alpha_i \mathbf{W}_{i,j}^O, \end{aligned} \quad (3)$$

where $\mathbf{W}_{i,j}^I$ and $\mathbf{W}_{i,j}^O$ are the j -th row and column of \mathbf{W}_i^I and \mathbf{W}_i^O in the i -th expert, respectively. The gating weight α_i is non-negative and can be viewed as the scaling factor of $\mathbf{W}_{i,j}^O$. Correspondingly, we define a neuron $n_{i,j}$ as a row vector $\mathbf{W}_{i,j}^I$ and a column vector $\mathbf{W}_{i,j}^O$, and the neuron activation of $n_{i,j}$ is $\sigma(\mathbf{W}_{i,j}^I \cdot \mathbf{x})$.

A.2 Experimental Details

Calculation of AP. AP is the weighted average of precision at different recall levels, which is a common metric for evaluating the performance of binary classification models. Since AP only represents the positive correlation, we compute the APs of both neuron activations and their opposite values, $-a_i$, and take the maximum as the final AP. The final AP ranges from 0.5 to 1, where 0.5 means the neuron is useless for the sub-function and 1 means the neuron is perfect for the sub-function.

Randomly-initialized models. In Section 4, the evaluation of randomly-initialized models is conducted 3 times and we report the average results.

Perturbation analysis on T5. To match the magnitude of neuron activations of T5, we set the variance of the Gaussian noise to be 4. The perturbation analysis is conducted 5 times and we report the average results.

Perturbation analysis on Switch Transformer. In this experiment, the predictivity and thus the functional experts are calculated based on the training set. Since Switch Transformer has not been trained on the considered datasets during pre-training, we train it on 128 randomly-picked training instances. We tune \mathbf{W}^O in the last four years

	Original	No Function	Function
CoLA	72.11 (± 0.01)	72.96 (± 0.01)	73.14 (± 0.01)
MNLI	53.94 (± 0.03)	50.94 (± 0.02)	56.24 (± 0.03)
MRPC	75.83 (± 0.02)	75.18 (± 0.02)	77.24 (± 0.02)
RTE	65.66 (± 0.04)	65.30 (± 0.04)	65.43 (± 0.03)
QNLI	75.16 (± 0.02)	76.19 (± 0.02)	78.34 (± 0.02)
QQP	77.14 (± 0.01)	76.80 (± 0.01)	78.26 (± 0.01)
SST-2	92.79 (± 0.01)	92.10 (± 0.00)	92.53 (± 0.01)
7-task Avg.	73.23	72.78	74.45

Table 3: Perturbation performance of Switch Transformer on GLUE tasks. For each dataset, we report the mean and standard deviation (std) accuracy over eight random seeds, using the mean \pm std.

as it takes the neuron activations as inputs, plus the routers in MoE layers. The learning rate and batch size are always 2E-4 and 16 respectively and we take Adam (Kingma and Ba, 2015) as the optimizer. The epoch is 100, 300, 50, 100, 300, 150, and 200 for CoLA, MNLI, MRPC, RTE, QNLI, QQP, and SST-2. We run each experiment with 8 different seeds and report the average best performance on the validation set. The result for each dataset is reported in Table 3.

Hyper-parameters of pre-training. The pre-training corpus is OpenWebText (Radford et al., 2019), which contains 40GB of web text. We use the same pre-training task as the official T5 and Switch Transformer, which is masked language modeling. The total number of training steps is 200K and we save the model every 5K steps. We use the same hyper-parameters for pre-training to avoid the effect of hyper-parameters on our analysis, and it is also a common practice when comparing dense and sparse T5s (Zoph et al., 2022). The learning rate is 1e-4. The batch size is 512. The max lengths of encoder inputs and decoder inputs are 512 and 256, respectively. We use 8 NVIDIA A100 GPUs for pre-training. The total pre-training time is around 3 days.

Experiments on random partitioning. In Section 5 and Section 6, we do the hypothesis testing on random partitioning, and we also calculate Spearman’s rank correlation between adjacent checkpoints on random partitioning. These random experiments are done 1000 times and we report the average results.

A.3 Function Distribution in Each Layer

Following Section 4, we report the distribution similarity of the randomly-initialized models in Figure 5, which is significantly different from that of the pre-trained models.

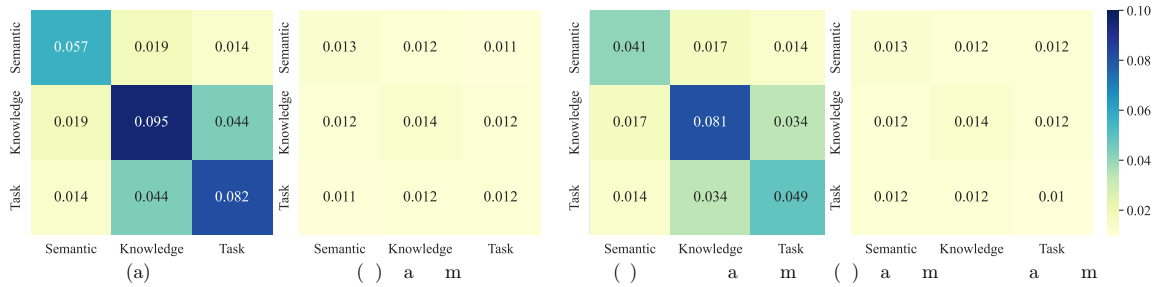


Figure 5: Distribution similarity between different sub-functions. We report the average similarity between functions. We consider the pre-trained models and their randomly-initialized counterparts.

Model	Partitioning	Semantics		Knowledge		Task	
		Prop.	Degree	Prop.	Degree	Prop.	Degree
Switch Transformer	Random	0.001	0.022	0.001	0.022	0.001	0.022
	Pre-MoE	0.138	1.968	0.101	2.028	0.124	2.029
T5	Random	0.001	0.021	0.001	0.021	0.001	0.021
	Post-MoE	0.030	2.330	0.038	2.985	0.039	3.522

Table 4: Proportion of sub-functional experts identified by hypothesis testing and their modularization degree. The result is averaged within each function.

Model	Semantics	Knowledge	Task
Switch Transformer	0.957	0.795	0.734
T5	0.912	0.894	0.931

Table 5: Average AP calculated based on $\mathcal{B} = \{(b_i, f_i)\}_{i=1}^E$, where b_i is the average predictivity across all sub-functions for expert e_i , and f_i is whether e_i is a functional expert or not.

A.4 Predictivity of Functional Experts

We quantify the predictivity of the expert for sub-functions in Section 3 and define functional experts in Section 5, and different are the technical details of quantification and definition. Hence, we conduct an experiment to check their consistency. For a function, we calculate b_i as the average predictivity across all sub-functions for each expert e_i , and we also denote $f_i \in \{0, 1\}$ as whether e_i is a functional expert or not. Now we have $\mathcal{B} = \{(b_i, f_i)\}_{i=1}^E$, based on which we compute the AP. A high AP indicates a high consistency. We report the average AP across all layers in Table 5.

The average AP is quite high. Therefore, we are confident that the quantification for expert predictivity is consistent with the concept of functional experts.

A.5 Sub-Function Experts

Similar to the concept of functional experts discussed in Section 4, we can also define so-called sub-functional experts. Basically, for each sub-function, we conduct statistical hypothesis testing on its sub-functional neurons. We similarly calculate the proportion of sub-functional experts and their modularization degree. We report the average result within each function. The result of the Switch Transformer and T5 used in Section 4 is reported in Table 4. The changing curve of the Switch Transformer and T5 trained by us is reported in Figure 6.

A.6 Organization of Sub-Functions

We further study how the model organizes sub-functions into their functional experts⁵ and how the organization changes during the pre-training. From the perspective of sub-functions, it is basically how a sub-function shares functional experts with others.

For a function, we can list all the sub-functions within it denoted as w_1, w_2, \dots, w_M . The similarity score between each pair of sub-functions can be seen as a matrix \mathcal{S} , where $\mathcal{S}_{i,j}$ is the similarity score between w_i and w_j for all $1 \leq i, j \leq M$.

⁵Strictly speaking, we did not define functional experts for a sub-function. In this context, the concept of “functional experts” is used to refer to the experts that have high predictivity for a sub-function.

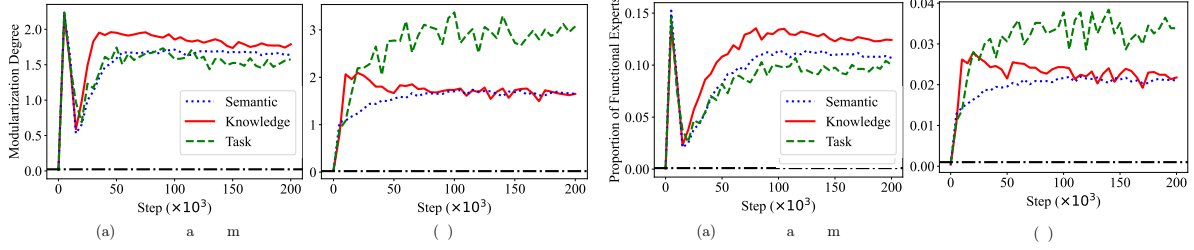


Figure 6: Changing curves of the proportion of sub-functional experts and their modularization degree. The horizontal line is the value for random partitioning.

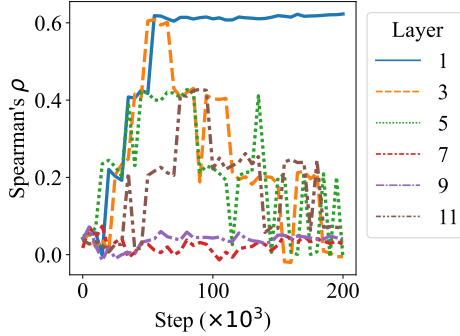


Figure 7: Changing curve of the average clustering scores. We plot the curve for each MoE layer in Switch Transformer.

For a given k , we denote $O_{i,j}^{(k)}$ as the top k expert overlap for w_i and w_j . When Spearman's rank correlation between $S_{i,:}$ and $O_{i,:}^{(k)}$ (denoted as $V_i^{(k)}$) is high, it indicates that the sub-functions similar to w_i share more functional experts than sub-functions dissimilar to w_i do, and vice versa. According to the meaning, we call $V_i^{(k)}$ clustering score.

We do a case study on the semantic function of the Switch Transformer, which focuses on understanding word meanings. Since relatively mature is the method of quantifying word similarity, we take S as the word similarity matrix calculated by spaCy (Honnibal and Montani, 2017). Note that the features at the word level are the lowest level of semantic information, so the word similarity reflects the lowest level of similarity between two semantic sub-functions.

We report the curve of $\frac{\sum_{k=1}^K \sum_{i=1}^M V_i^{(k)}}{KM}$ for each layer in Figure 7. We also report the result for random partitioning in Figure 8. From this figure, we have the following observations. (1) During pre-training, the clustering score of the lowest MoE layer (Layer 1) quickly achieves 0.6 and then keeps stable till the end. It proves that the pre-training tends to organize sub-functions sharing similar low-

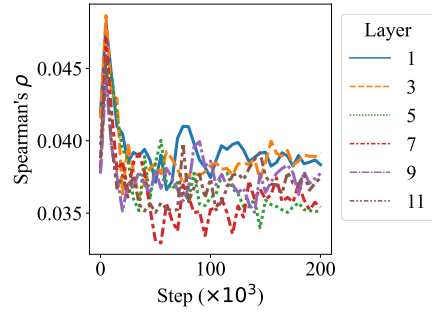


Figure 8: Changing curves of the average clustering scores on random partitioning. We plot the curve for each MoE layer in Switch Transformer.

level information into the same functional experts in the low layer. However, the final clustering score of higher MoE layers is close to 0, indicating that high layers do not organize sub-functions based on word similarity. We guess that the reason is that the high layers may process high-level semantic information, which is not related to the word similarity. (3) We see three interesting curves of layers 3, 5, and 11. Their clustering scores achieve a high point when the clustering score of layer 1 first achieves its highest point, and then they continuously decrease to 0. The trend that high layers become increasingly responsible for high-level features may grow faster when the low-layer organization has been established than when the organization is forming.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
After the conclusion.
- A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.
- A3. Do the abstract and introduction summarize the paper’s main claims?
Abstract and Section 1
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

5,A.2, A.3

- B1. Did you cite the creators of artifacts you used?
5,A.2
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
These datasets are publicly available and free of use for research purposes.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The use is consistent with their intended use.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We use established public datasets, which should not cause privacy issues.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
A.2
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
A.2,A.3

C Did you run computational experiments?

4,5,6

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
A.3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

A.3

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

A.3

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

A.7

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

No response.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

No response.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

No response.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

No response.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

No response.