EMERGENT MODULARITY IN PRE-TRAINED TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-trained Transformers have shown the potential to realize the dream of general intelligence, encouraging researchers to explore the analogy between Transformers and human brains. These advances raise the question of whether Transformers have a modular structure similar to brain regions, where neurons are closely related and specialized in a certain function. In this work, we analyze the modularity of Transformers by studying the expert networks, which are clusters of neurons, in Mixture-of-Experts (MoE) Transformers.[1] To evaluate the functional specialization of experts, we propose a novel framework to identify the functionality of both neurons and experts. We conduct empirical analyses on two representative pre-trained Transformers and find that (1) Transformer neurons are functionally specialized, which provides the necessary condition of modularity. (2) Transformer experts are modularized. There are functional experts, where clustered are the neurons specialized in a certain function. (3) The modular structure is stabilized at the early stage of pre-training, which is faster than the neuron stabilization. It reveals the coarse-to-fine mechanism of pre-training, which first constructs the coarse modular structure and then improves the fine-grained neuron functions. In summary, we explore the emergent modularity in pre-trained Transformers and hope to help the community better understand the working mechanism of Transformers. Our code and data will be released to facilitate future research.

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), which encourages researchers to explore the analogy between Transformers and human brains (Toneva & Wehbe, 2019; Caucheteux et al., 2021; Caucheteux & King, 2022; Goldstein et al., 2022). Previous work has shown that the **behaviors** of Transformers are similar to those of human brains, which naturally raises a question: do Transformers also have similar internal **structures** to human brains?

To study the internal structure of human brains, neuroscientists have partitioned the brain into several regions, where the neurons in each region are closely connected and work for a certain function (Garey, 1999; Graziano & Aflalo, 2007). In analogy to brain regions, we explore to discover the modular structure of Transformers. Considering that module is a broad concept, in this work, we study modules consisting of neurons and focus on two characteristics of modules: (1) **Clustered**. The neurons in a module should be closely related, e.g., activated simultaneously (Meyes et al., 2020); (2) **Functionally Specialized**. The neurons in a module should be specialized in a specific function (Csordás et al., 2021).

Towards the characteristic of clustering, we analyze the Mixture-of-Experts (MoE) structure (Jacobs et al., 1991) in Transformers because each expert is naturally a cluster of neurons (Lepikhin et al., 2021; Fedus et al., 2022). To study functional specialization, we propose a unified and hierarchical framework to analyze the functionality of both neurons and expert networks. In this framework, we study three diverse functions by a unified method, including semantic function (Scarlini et al., 2019b; Suau et al., 2020), knowledge function (Jiang et al., 2020; Dai et al., 2022), and task function. Based on this framework, we can identify the functionality of neurons and experts.

[1] The MoE partitioning may be done before or after pre-training.

In this work, we study two types of MoE Transformers, 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. Post-MoE refers to the models that are converted from vanilla Transformers to their equivalent MoE version by MoEfication (Zhang et al., 2022b) after pre-training.[2] We conduct extensive experiments on Switch Transformer (Fedus et al., 2022) and T5 (Raffel et al., 2020) for pre-MoE and post-MoE respectively. In summary, we study the following research questions:

(Q1) **Necessary condition of modularity: are neurons functionally specialized?** We first study the functional specialization of neurons, which is the foundation of functional modularity. According to the experiments on neuron functions, we find that the neurons in pre-trained Transformers become more specialized than those in randomly-initialized ones after self-supervised learning on large-scale corpora. In particular, we find in pre-trained Transformers, there are several groups of neurons, each of which excels in a specific function.

(Q2) **Modularity of MoE: are expert networks modularized?** We further study the function distribution among experts. The results suggest 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 expert networks for a certain function will lead to more significant performance degradation of the function than perturbing individual neurons specialized in the function. Therefore, the expert networks indeed have specialized functions and are functionally modularized.

(Q3) **Emergence of modularity: how do modular experts emerge?** 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. It reveals the coarse-to-fine mechanism of pre-training, which first constructs the coarse modular structure and then improves the fine-grained neuron functions.

We hope our observations on the emergent modularity of expert networks in Transformers can provide insights for future research on modular Transformers. Besides, it also provides a new modular perspective to connect biological neural networks and artificial neural networks.

2 Related work

Interpreting Pre-trained Transformers. As large-scale pre-trained Transformers have achieved great success on a wide range of NLP tasks (Min et al., 2021; Bommasani et al., 2021), researchers explore to understand how these models work (Rogers et al., 2020), such as probing the model knowledge (Liu et al., 2019; Hewitt & Manning, 2019; Petroni et al., 2019) and interpreting the model behaviors (Voita et al., 2019; Clark et al., 2019b). Among them, neuron-level analysis is another important branch (Sajjad et al., 2021), which is most related to our work. Some works study the contextualized representations as neurons and find that they capture amounts of linguistic information (Dalvi et al., 2019; Durrani et al., 2020; Antverg & Belinkov, 2022). Other works study the neurons in feedforward layers and find that there is various information encoded by neurons such as concepts and facts (Suau et al., 2020; Dai et al., 2022). In this work, we follow the neuron definition of the second group and try to understand how 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 & 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) 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 capture more language knowledge.

[2]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 with parameters frozen and study their expert networks.

Transformers with Mixture-of-Experts. Mixture-of-experts (Jacobs et al., 1991) is usually used to enlarge the model capacity of Transformers while keeping the computational efficiency (Fedus et al., 2022; Lepikhin et al., 2021). 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., 2022). These works explicitly design extra constraints during pre-training to ensure the modularization of expert networks. However, it is still unclear whether standard Transformers can form modular structures by themselves. In this work, we study the emergent modularity of both pre-MoE and post-MoE Transformers to understand their inner working mechanism.

3 FUNCTIONALITY EVALUATION

In this section, we first introduce the definition of neurons, which are the basic units of experts, 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. Transformer is widely used by existing pre-trained language models (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2021), which is mainly composed of attention and feedforward networks (Vaswani et al., 2017). Among them, feedforward networks (FFNs) account for about two-thirds of the parameters and are the main components of Transformer. Moreover, previous work has shown that there is rich information in FFNs (Suau et al., 2020; Dai et al., 2022; Geva et al., 2021). Hence, we focus on FFNs in this work and study how information distributes in FFNs.

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

$$\text{FFN}(\boldsymbol{x}) = \sum_{i=1}^{d_{ff}} \sigma(\boldsymbol{W}^I_{i,:} \cdot \boldsymbol{x})\boldsymbol{W}^O_{:,i}, \tag{1}$$

where $\boldsymbol{W}^I_{i,:}$ and $\boldsymbol{W}^O_{:,i}$ are the i-th row and column of $\boldsymbol{W}^I$ and $\boldsymbol{W}^O$, 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 $\boldsymbol{W}^I_{i,:}$ and a column vector $\boldsymbol{W}^O_{:,i}$. The neuron activation of n_i is $\sigma(\boldsymbol{W}^I_{i,:} \cdot \boldsymbol{x})$. The number of neurons in an FFN is equal to the hidden dimension of the first linear layer d_{ff}.

Mixture-of-experts in Transformer is a variant of FFN (Lepikhin et al., 2021; Fedus et al., 2022), which significantly increases the model capacity by adding more parameters and keeps similar computational costs. 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, $\text{MoE}(\boldsymbol{x}) = \sum_{i=1}^{E} \alpha_i \text{FFN}_i(\boldsymbol{x})$, where α_i is the weight of the i-th expert and FFN_i is the i-th expert e_i. α_i is computed by a gating network. We can also rewrite the MoE layer into a neuron-based form,

$$\text{MoE}(\boldsymbol{x}) = \sum_{i=1}^{E} \alpha_i \sum_j \sigma(\boldsymbol{W}^I_{i,j,:} \cdot \boldsymbol{x})\boldsymbol{W}^O_{i,:,j} = \sum_{i,j} \sigma(\boldsymbol{W}^I_{i,j,:} \cdot \boldsymbol{x})\alpha_i\boldsymbol{W}^O_{i,:,j}, \tag{2}$$

where $\boldsymbol{W}^I_{i,j,:}$ and $\boldsymbol{W}^O_{i,:,j}$ are the j-th row and column of $\boldsymbol{W}^I_i$ and $\boldsymbol{W}^O_i$, respectively. The gating coefficient α_i is non-negative and can be viewed as the scaling factor of $\boldsymbol{W}^O_{i,:,j}$.Correspondingly, we define a neuron $n_{i,j}$ as a row vector $\boldsymbol{W}^I_{i,j,:}$ and a column vector $\boldsymbol{W}^O_{i,:,j}$, and the neuron activation of $n_{i,j}$ is $\sigma(\boldsymbol{W}^I_{i,j,:} \cdot \boldsymbol{x})$.

Predictivity for Functions. To comprehensively study the functions of neurons and experts, we cover three typical functions, including semantic function, knowledge function, and task function. To study these functions finely, we construct sub-functions for each function and there are 576 sub-functions in total. Please refer to Appendix A.6 for the details of the functions.

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. Based on neuron predictivity, we can further evaluate the predictivity of experts. Following Suau et al. (2020), 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 pointwise, we define the activation of the neuron n_i of a sequence s as $a = \max_{\boldsymbol{x} \in \boldsymbol{s}} \sigma(\boldsymbol{W}_{i,:}^I \cdot \boldsymbol{x})$, where $\boldsymbol{s} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_l\}$ is the hidden states of s and l is the length of s. Then, we have the pairs of neuron activations and labels, $\mathcal{A} = \{(a_i, y_i)\}_{i=1}^{|\mathcal{D}|}$. Based on $\mathcal{A}$, we compute the average precision (AP) of the neuron activations as the predictivity of the neuron. For the i-th 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.5 for more details.

Evaluation Setups. In the experiments, we evaluate two representative pre-trained Transformers, Switch Transformer (Fedus et al., 2022) and T5 (Raffel et al., 2020). The architecture of Switch Transformer is similar to T5 except that Switch Transformer replaces FFNs in the even Transformer layer with MoE layers, and belongs to the pre-MoE Transformers. To evaluate the implicit structure of T5, we convert them into their corresponding MoE version by MoEfication (Zhang et al., 2022b). The vanilla T5 after MoEfication belongs to the post-MoE Transformers. To match the experimental settings, we choose the base size of these two models and set the number of experts in each MoE layer to 16. Since the functionality evaluation does not involve decoding, we only compute the neuron predictivity 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 ARE NEURONS FUNCTIONALLY SPECIALIZED?

In this section, we study the functional specialization of neurons, which is the necessary condition of modularity. If we find that there is a group of neurons that mainly excel in a certain function, we can conclude that the neurons are functionally specialized.

We first study how functions distribute among different Transformer layers. Specifically, 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 presentation consistency, we normalize the best predictivity for each function. Then, we study how functions distribute in each layer. Specifically, we first identify the neurons with the top predictivity ranking for each sub-function as *sub-functional neurons* in each layer and then compute the overlap between 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, it means the two sub-functions share a large portion of neurons. In the experiments, we set $k = 32$ for T5 and $k = 512$ for Switch Transformer. 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. For comparisons, we also evaluate randomly-initialized models.

We report the results in Figure 1. From this figure, we have the following observations. (1) The best predictivity of pre-trained neurons is significantly higher than that of randomly-initialized neurons, indicating that the neurons have learned these functions from pre-training and the neurons with top-ranked predictivity indeed excel in corresponding sub-functions. (2) The best predictivity of the task function increases with the layer number while the best predictivity of the semantic and knowledge functions varies little across layers. It suggests that the difficulty of the task function may be higher than the semantic and knowledge functions so the higher layers are more suitable for learning the task function. (3) In the pre-trained models, the distribution similarity of the same function is significantly larger than that of different functions, which indicates that the sub-functions of the same function share a large portion of neurons. And, it is different from the randomly-initialized models as shown in Appendix A.1. Hence, we conclude that **there are some emergent groups of neurons after pre-training, each of which is corresponding to a certain function.** (4) One neuron may be capable of multiple sub-functions even from different functions. For example, the average

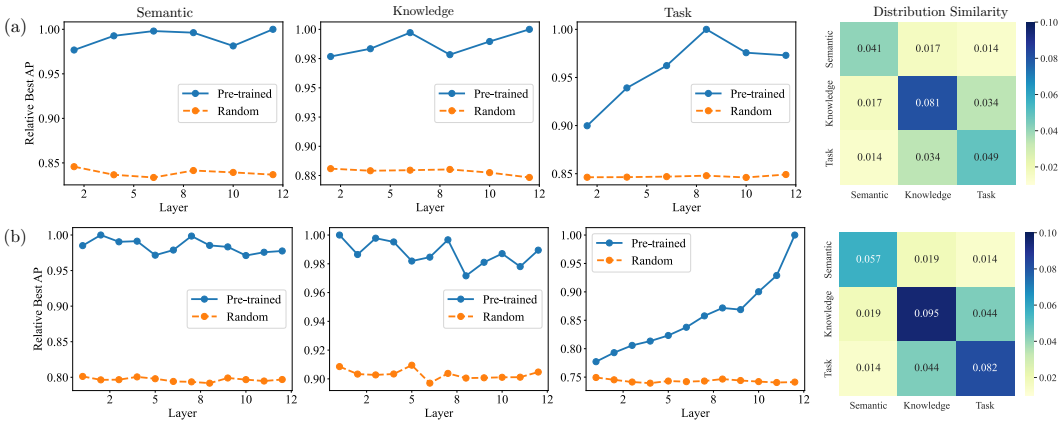


Figure 1: Best neuron predictivity of each layer and distribution similarity between different functions of (a) Switch Transformer and (b) T5. We report the average predictivity of each function and the average distribution similarity of different layers. We consider the pre-trained models and their randomly-initialized counterparts.

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 Are expert networks modularized?

In this section, we study the modularity of expert networks. First, we verify whether the neurons specialized in a certain function are concentrated in some experts. Second, we perturb the experts corresponding to a certain function to evaluate the importance of the experts for model performance.

Neuron distribution among 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 among 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 for different sub-functions are independently and randomly distributed among experts, i.e., the number of sub-functional neurons in each expert follows a hypergeometric distribution with parameters N, K, and n_E. The sum of the numbers of sub-functional neurons for each sub-function 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 our experiments, we treat the neurons with the highest 1% predictivity 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. 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. The modularization degree of a functional expert is defined as the relative ratio of functional neurons in the expert compared to uniform distribution, $\frac{r_i}{n_E}/(\frac{Mk}{N})$, where $\frac{r_i}{n_E}$ is the proportion of functional neurons in the expert and $\frac{Mk}{N}$ is the proportion expectation of functional neurons under the 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 two observations. (1) There are much more functional experts in the pre-MoE partitioning of Switch Transformer than in the random

[3]We 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 in the experiments.

Table 1: Proportion of functional experts and their modularization degree.

Model	Partitioning	Semantics		Knowledge		Task	
		Prop.	Degree	Prop.	Degree	Prop.	Degree
Switch Transformer	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	0.252	1.203	0.061	1.031	0.007	0.221
	Post-MoE	**0.338**	**2.000**	**0.214**	**2.686**	**0.120**	**3.276**

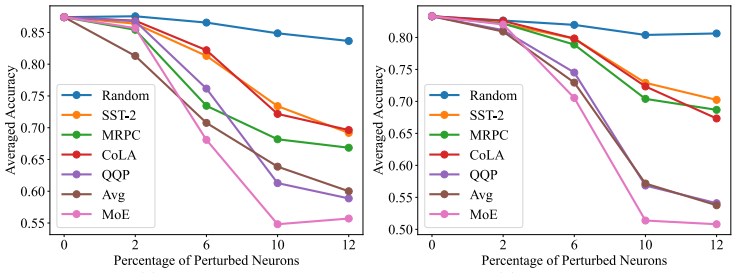


Figure 2: Perturbation performance on different datasets. Each line has a unique perturbation order. For "Random", we randomly perturb neurons. For "SST-2", "MRPC", "CoLA", "QQP", we are guided by the neuron predictivity on each dataset and perturb the neurons with top-ranked predictivity. For "Avg", we sum the predictivity of all datasets above and also perturb the neurons with top-ranked sum of the predictivity. For "MoE", we consider the experts with top-ranked sum of the predictivity. The seen datasets are the four datasets above. The unseen datasets include six other datasets.

partitioning. Moreover, the modularization degree of the functional experts in the pre-MoE is significantly higher than that in the random partitioning. It indicates that pre-trained experts are more likely to intensively include neurons excelling in a certain function. (2) The proportion of functional experts and modularization degree in the post-MoE partitioning of T5 are as good as those in the pre-MoE of Switch Transformer. It suggests that we can also discover modular experts from T5 by MoEfication and achieve similar results to those of Switch Transformer. (3) We further compare the predictivity of the functional experts and non-functional experts and find that the functional experts have significantly higher predictivity than the non-functional experts in their corresponding functions. It indicates that our quantification for expert predictivity is consistent with the concept of functional experts. More details are in Appendix A.2.

Perturbation analysis. Furthermore, we conduct perturbation experiments, which are widely used to analyze both biological and artificial neural networks (Michel et al., 2019; Cowley et al., 2022), to evaluate the effect of functional experts on model performance.

Specifically, we perturb the 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 predictivity for several downstream datasets and perturb the top-ranked experts. We regard the datasets used in computing the sum of the predictivity as seen datasets, including SST-2, MRPC, CoLA, and QQP. To evaluate the generalization ability of the functional experts, we also perturb them and evaluate the perturbed models on unseen datasets, including MNLI, QNLI, CB (De Marneffe et al., 2019), MultiRC (Khashabi et al., 2018), and BoolQ (Clark et al., 2019a). 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 predictivity for a certain dataset, (2) perturb the neurons that have top-ranked sum of the predictivity for seen datasets, and (3) perturb the neurons randomly. The perturbed pre-trained Transformers is T5 and we only perturb the neurons in the last four layers because the task function is mainly located in the last layers as shown in Figure 1.

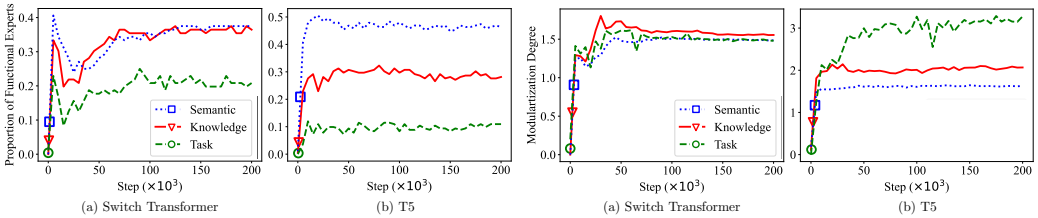


Figure 3: Changing curves of the proportion of functional experts and their modularization degree. We also mark the value for random partitioning on the curve.

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 functional experts are very important for the model performance. For example, perturbing 10% of neurons in the functional experts decreases the average accuracy by nearly 30% and makes the model perform as random guessing. (2) "Avg" perturbation achieves a larger performance drop than single-dataset perturbation, which is expected because intuitively it perturbs the neurons with high overall predictivity. (3) **Perturbing functional experts leads to a more significant performance drop than perturbing individual neurons** on both seen and unseen datasets when the proportion of perturbed neurons is higher than 6%. It suggests neurons in the functional experts cooperate instead of working independently so perturbing them will destroy the cooperation and lead to a more 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.

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

6 How do modular experts emerge during pre-training?

To study the pre-training process, we pre-train the base version of T5 and Switch Transformer from scratch. 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 MoEfication result of the last checkpoint as the MoE structure of T5.

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) Overall, 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 Transform 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 predictivity of neurons and experts changes during pre-training.

There are two kinds of predictivity dynamics during pre-training. The first is the changing of the absolute predictivity, and the second is the relative order changing of predictivity among all experts or neurons in a layer. Although it is straightforward to study the absolute predictivity, the absolute predictivity has different scales for different functions and different layers, and thus it is difficult to have a uniform analysis standard. Hence, we focus on the relative order changing of predictivity. Intuitively, for a sub-function, some experts or neurons excel in it compared to other ones at some

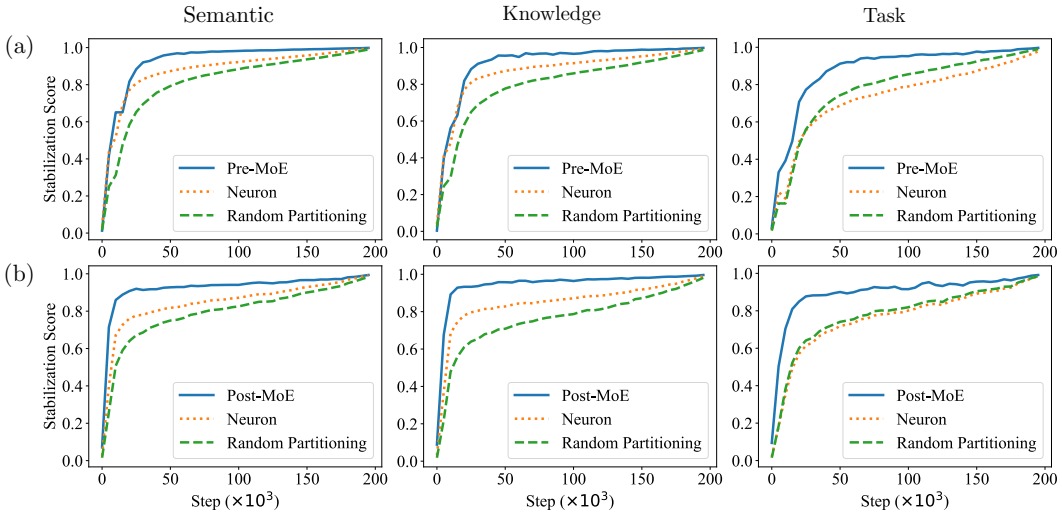


Figure 4: Spearman's rank correlation between the functionality distributions of two adjacent checkpoints during pre-training.

stage, after which they keep such relative dominance as the pre-training is continuously going. From this intuitive perspective, we study the stabilization of predictivity rankings in a layer.

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 predictivity of experts or neurons in the considered layer of the two checkpoints. In this way, we measure to what extent the predictivity 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 predictivity to achieve a stabilization score of 0.9. (3) Expert stabilization is notably faster than both neuron stabilization and the stabilization for random partitioning. In conclusion, we see strong evidence that **coarse-to-fine is the inner mechanism of pre-training**. Transformer first learns a modular structure, where the structure becomes stable at the early stage, and then there is a fine-grained process to improve the predictivity of neurons.

Organization of Sub-Functions. We further study how the model organizes sub-functions into their functional experts[4] 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 $w_1, w_2, \ldots, w_M$. The similarity score between each pair of sub-functions can be seen as a matrix $\boldsymbol{S}$, where $\boldsymbol{S}_{i,j}$ is the similarity score between w_i and w_j for all $1 \leq i, j \leq M$. For a given k, we denote $\boldsymbol{O}_{i,j}^{(k)}$ as the top k expert overlap for w_i and w_j. When Spearman's rank correlation between $\boldsymbol{S}_{i,:}$ and $\boldsymbol{O}_{i,:}^{(k)}$ (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

[4]Strictly 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.

take $\boldsymbol{S}$ as the word similarity matrix calculated by spaCy (Honnibal & 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 5. We also report the result for random partitioning in Appendix A.4. 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-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.

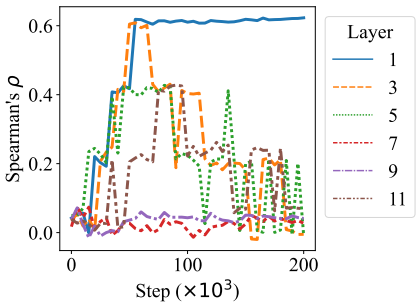


Figure 5: Changing curve of the average clustering scores. We plot the curve for each MoE layer in Switch Transformer.

7 Discussion

Efficient Pre-training. Large-scale pre-training of Transformers requires a large amount of computation resources (Brown et al., 2021; Chowdhery et al., 2022). Mixture-of-Experts is a promising solution to reduce the computational cost of pre-training by activating only a small part of the experts for a certain input. Our findings have shown the emergent modularity of experts during pre-training, which demonstrates the reasonableness of the MoE structure. However, we also find that Switch Transformer is more unstable than T5 on the modular structure at the beginning of pre-training. It suggests that we should gradually sparsify the experts during pre-training, which has been explored in some preliminary works (Nie et al., 2021; Hazimeh et al., 2021).

Model Fusion. Considering there are amounts of pre-trained models on different corpora, researchers have started to explore how to fuse them to aggregate different model knowledge together. Compared with model ensembling, model fusion is expected to be more efficient because it does not compute all of the models. Existing work focuses on weight averaging and achieves some promising results (Li et al., 2022; Matena & 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 & 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 find that the experts in the MoE structure are modularized after pre-training. 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.

REFERENCES

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of CVPR*, pp. 39–48, 2016. URL https://doi.org/10.1109/CVPR.2016.12.

Omer Antverg and Yonatan Belinkov. On the pitfalls of analyzing individual neurons in language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=8uz0EWPQIMu.

Carliss Young Baldwin, Kim B Clark, Kim B Clark, et al. *Design rules: The power of modularity*, volume 1. MIT press, 2000.

Dana H. Ballard. Modular learning in neural networks. In *Proceedings of AAAI*, pp. 279–284, 1987. URL http://www.aaai.org/Library/AAAI/1987/aaai87-050.php.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *arxiv preprint arXiv:2108.07258*, 2021. URL https://arxiv.org/abs/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. Language models are Few-Shot learners. In *Proceedings of NeurIPS*, pp. 1877–1901, 2021. URL http://arxiv.org/abs/2005.14165.

Charlotte Caucheteux and Jean-Rémi King. Brains and algorithms partially converge in natural language processing. *Communications biology*, 5(1):1–10, 2022.

Charlotte Caucheteux, Alexandre Gramfort, and Jean-Remi King. Disentangling syntax and semantics in the brain with deep networks. In *Proceedings of ICML*, volume 139, pp. 1336–1348, 2021. URL http://proceedings.mlr.press/v139/caucheteux21a.html.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. URL https://doi.org/10.48550/arXiv.2204.02311.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL-HLT 2019*, 2019a.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of bert's attention. In *Proceedings of BlackboxNLP*, pp. 276–286, 2019b. URL https://doi.org/10.18653/v1/W19-4828.

Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson. The evolutionary origins of modularity. *Proceedings of the Royal Society b: Biological sciences*, 280(1755):20122863, 2013.

Benjamin R. Cowley, Adam J. Calhoun, Nivedita Rangarajan, Jonathan W. Pillow, and Mala Murthy. One-to-one mapping between deep network units and real neurons uncovers a visual population code for social behavior. *bioRxiv*, 2022. doi: 10.1101/2022.07.18.500505. URL https://www.biorxiv.org/content/early/2022/07/20/2022.07.18.500505.

Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Are neural nets modular? inspecting functional modularity through differentiable weight masks. In *Proceedings of ICLR*, 2021. URL https://openreview.net/forum?id=7uVcpu-gMD.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pp. 177–190. Springer, 2006.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of ACL*, pp. 8493–8502, 2022. URL https://doi.org/10.18653/v1/2022.acl-long.581.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James R. Glass. What is one grain of sand in the desert? analyzing individual neurons in deep NLP models. In *Proceedings of AAAI*, pp. 6309–6317, 2019. URL https://doi.org/10.1609/aaai.v33i01.33016309.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The CommitmentBank: Investigating projection in naturally occurring discourse. 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pp. 4171–4186, 2019. URL https://aclanthology.org/N19-1423.

William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing*, 2005.

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. Glam: Efficient scaling of language models with mixture-of-experts. In *Proceedings of ICML*, pp. 5547–5569, 2022. URL https://proceedings.mlr.press/v162/du22c.html.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. Analyzing individual neurons in pre-trained language models. In *Proceedings of EMNLP*, pp. 4865–4880, 2020. URL https://aclanthology.org/2020.emnlp-main.395.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL http://jmlr.org/papers/v23/21-0998.html.

Nanyi Fei, Zhiwu Lu, Yizhao Gao, Guoxing Yang, Yuqi Huo, Jingyuan Wen, Haoyu Lu, Ruihua Song, Xin Gao, Tao Xiang, et al. Towards artificial general intelligence via a multimodal foundation model. *Nature Communications*, 13(1):1–13, 2022.

Laurence J Garey. *Brodmann's' localisation in the cerebral cortex'*. 1999.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of EMNLP*, pp. 5484–5495, 2021. URL https://aclanthology.org/2021.emnlp-main.446.

Ariel Goldstein, Avigail Dabush, Bobbi Aubrey, Mariano Schain, Samuel A Nastase, Zaid Zada, Eric Ham, Zhuoqiao Hong, Amir Feder, Harshvardhan Gazula, et al. Brain embeddings with shared geometry to artificial contextual embeddings, as a code for representing language in the human brain. *bioRxiv*, 2022.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. In *Proceedings of ICLR*, 2021. URL https://openreview.net/forum?id=mLcmdlEUxy-.

Michael SA Graziano and Tyson N Aflalo. Mapping behavioral repertoire onto the cortex. *Neuron*, 56(2):239–251, 2007.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah Smith, and Luke Zettlemoyer. Demix layers: Disentangling domains for modular language modeling. In *Proceedings of NAACL-HLT*, pp. 5557–5576, 2022. URL https://doi.org/10.18653/v1/2022.naacl-main.407.

Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed H. Chi. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. In *Proceedings of NeurIPS*, pp. 29335–29347, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f5ac21cd0ef1b88e9848571aeb53551a-Abstract.html.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of NAACL-HLT*, pp. 4129–4138, 2019. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419.

Shlomi Hod, Stephen Casper, Daniel Filan, Cody Wild, Andrew Critch, and Stuart Russell. Detecting modularity in deep neural networks. *arXiv preprint arXiv:2110.08058*, 2021. URL https://arxiv.org/abs/2110.08058.

Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87, 1991. URL http://www.cs.toronto.edu/~fritz/absps/jjnh91.pdf.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know. *Trans. Assoc. Comput. Linguistics*, 8:423–438, 2020. URL https://doi.org/10.1162/tacl_a_00324.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 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)*, pp. 252–262, 2018.

Louis Kirsch, Julius Kunze, and David Barber. Modular networks: Learning to decompose neural computation. In *Proceedings of NeurIPS*, pp. 2414–2423, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/310ce61c90f3a46e340ee8257bc70e93-Abstract.html.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *Proceedings of ICLR*, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of EMNLP*, pp. 3045–3059, 2021. URL https://doi.org/10.18653/v1/2021.emnlp-main.243.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models. *arxiv preprint arXiv:2208.03306*, 2022. URL https://doi.org/10.48550/arXiv.2208.03306.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of NAACL-HLT*, pp. 1073–1094, 2019. URL https://doi.org/10.18653/v1/n19-1112.

Dirk M Lorenz, Alice Jeng, and Michael W Deem. The emergence of modularity in biological systems. *Physics of life reviews*, 8(2):129–160, 2011.

Henry Markram. The blue brain project. In *Proceedings of SC*, 2006. URL https://doi.org/10.1145/1188455.1188511.

Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging. *arxiv preprint arXiv:2111.09832*, 2021. URL https://arxiv.org/abs/2111.09832.

Richard Meyes, Constantin Waubert de Puiseau, Andres Posada-Moreno, and Tobias Meisen. Under the hood of neural networks: Characterizing learned representations by functional neuron populations and network ablations. *arXiv preprint arXiv:2004.01254*, 2020.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Proceedings of NeurIPS*, pp. 14014–14024, 2019. URL https://papers.nips.cc/paper/2019/hash/2c601ad9d2ff9bc8b282670cdd54f69f-Abstract.html.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *arxiv preprint arXiv:2111.01243*, abs/2111.01243, 2021. URL https://arxiv.org/abs/2111.01243.

Xiaonan Nie, Shijie Cao, Xupeng Miao, Lingxiao Ma, Jilong Xue, Youshan Miao, Zichao Yang, Zhi Yang, and Bin Cui. Dense-to-sparse gate for mixture-of-experts. *arxiv preprint arXiv:2112.14397*, 2021. URL https://arxiv.org/abs/2112.14397.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. Language models as knowledge bases? In *Proceedings of EMNLP-IJCNLP*, pp. 2463–2473, 2019. URL https://doi.org/10.18653/v1/D19-1250.

Jonas Pfeiffer, Naman Goyal, Xi Victoria Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the NAACL-HLT*, pp. 3479–3495, 2022. URL https://doi.org/10.18653/v1/2022.naacl-main.255.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified Text-to-Text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://arxiv.org/abs/1910.10683.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*, pp. 2383–2392. Association for Computational Linguistics, 2016.

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. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. URL https://doi.org/10.48550/arXiv.2205.06175.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how BERT works. *Trans. Assoc. Comput. Linguistics*, 8:842–866, 2020. URL https://doi.org/10.1162/tacl_a_00349.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. Neuron-level interpretation of deep NLP models: A survey. *arXiv preprint arXiv:2108.13138*, 2021. URL https://arxiv.org/abs/2108.13138.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. Just "onesec" for producing multilingual sense-annotated data. In *Proceedings of ACL*, pp. 699–709, 2019a. URL https://doi.org/10.18653/v1/p19-1069.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. Just "OneSeC" for producing multilingual sense-annotated data. In *Proceedings of ACL*, pp. 699–709, 2019b. URL https://aclanthology.org/P19-1069.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pp. 1631–1642, 2013.

Charles Spearman. The proof and measurement of association between two things. 1961.

Xavier Suau, Luca Zappella, and Nicholas Apostoloff. Finding experts in transformer models. *arXiv preprint arXiv:2005.07647*, 2020. URL https://arxiv.org/abs/2005.07647.

Mariya Toneva and Leila Wehbe. Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain). In *Proceedings of NeurIPS*, pp. 14928–14938, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/749a8e6c231831ef7756db230b4359c8-Abstract.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, and Jakob Uszkoreit. Attention is all you need. In *Proceedings of NeurIPS*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of ACL*, pp. 5797–5808, 2019. URL https://aclanthology.org/P19-1580.

George Von Dassow and Ed Munro. Modularity in animal development and evolution: elements of a conceptual framework for evodevo. *Journal of Experimental Zoology*, 285(4):307–325, 1999.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of EMNLP*, pp. 353–355, 2018. URL https://www.aclweb.org/anthology/W18-5446.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. URL https://doi.org/10.48550/arXiv.2208.10442.

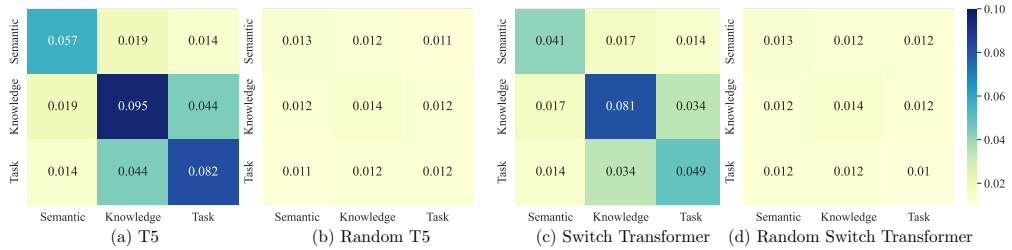
Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *arXiv preprint 1805.12471*, 2018.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, 2018.

Fan Zhang, Duyu Tang, Yong Dai, Cong Zhou, Shuangzhi Wu, and Shuming Shi. Skillnet-nlu: A sparsely activated model for general-purpose natural language understanding. *arXiv preprint 2203.03312*, 2022a. URL https://arxiv.org/abs/2203.03312.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. MoEfication: Transformer feed-forward layers are mixtures of experts. In *Findings of ACL*, 2022b. URL https://aclanthology.org/2022.findings-acl.71.pdf.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. Designing effective sparse expert models. *arxiv preprint arXiv:2202.08906*, 2022. URL https://arxiv.org/abs/2202.08906.



(a) T5 (b) Random T5 (c) Switch Transformer (d) Random Switch Transformer

Figure 6: Distribution similarity between different sub-functions. We report the average similarity between functions. We consider the pre-trained models and their randomly-initialized counterparts.

Table 2: 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.

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

A Appendix

A.1 Function distribution among neurons

Following Section 4, we report the distribution similarity of the randomly-initialized models in Figure 6, which is significantly different from that of the pre-trained models.

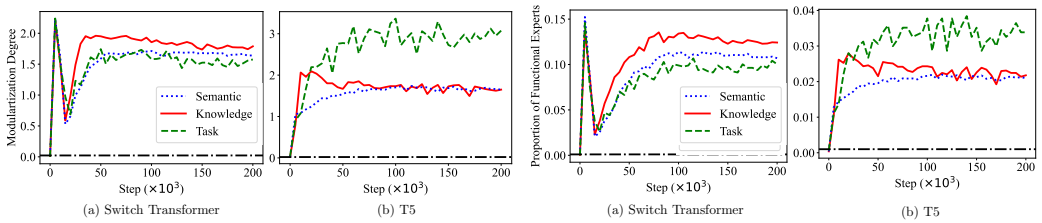
A.2 Predictivity of functional experts

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 2.

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.3 Sub-Functioanl 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



(a) Switch Transformer (b) T5 (a) Switch Transformer (b) T5

Figure 7: Changing curves of the proportion of sub-functional experts and their modularization degree. The horizontal line is the value for random partitioning.

Table 3: Proportion of sub-functional experts identified by hypothesis testing and their modularization degree. The result is averaged within each function.

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**

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 3. The changing curve of the Switch Transformer and T5 trained by us is reported in Figure 7.

A.4 Organization of Sub-Functions on Random Partitioning

We conduct the same experiment for evaluating the organization of sub-functions on random partitioning. The experiment is conducted 200 times and we report curves of the average results in Figure 8. This figure shows that the clustering score is always close to 0 on random partitioning. K is set as 5 in both Figure 5 and Figure 8.

A.5 Details of the experiments

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 on randomly-initialized models is conducted 3 times and we report the average results.

Perturbation analysis. 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.

Hyper-parameters of pre-training. 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 partioning. 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.6 Details of 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 (Scarlini et al., 2019a), to construct binary classification data for semantic sub-functions. In OneSec, each sentence has a keyword whose sense is annotated based on Wikipedia[5]. 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

[5] https://en.wikipedia.org/

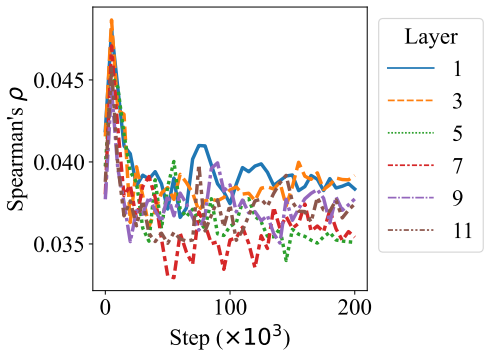


Figure 8: Changing curves of the average clustering scores on random partitioning. We plot the curve for each MoE layer in Switch Transformer.

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.

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 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), QQP[6], MNLI (Williams et al., 2018), CoLA (Warstadt et al., 2018), MRPC (Dolan & 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.

Admittedly, coarse is our function classification. 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.

[6] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs