Finding Skill Neurons in Pre-trained Transformer-based Language Models

Xiaozhi Wang[1*], **Kaiyue Wen**[2*], **Zhengyan Zhang**[1],
Lei Hou[1,3†], **Zhiyuan Liu**[1,3†], **Juanzi Li**[1,3]
[1]Department of Computer Science and Technology, BNRist;
[2]Institute for Interdisciplinary Information Sciences;
[3]KIRC, Institute for Artificial Intelligence,
Tsinghua University, Beijing, 100084, China
{wangxz20,wenky20}@mails.tsinghua.edu.cn

Abstract

Transformer-based pre-trained language models have demonstrated superior performance on various natural language processing tasks. However, it remains unclear how the skills required to handle these tasks distribute among model parameters. In this paper, we find that after prompt tuning for specific tasks, the activations of some neurons within pre-trained Transformers[1] are highly predictive of the task labels. We dub these neurons *skill neurons* and confirm they encode task-specific skills by finding that: (1) Skill neurons are crucial for handling tasks. Performances of pre-trained Transformers on a task significantly drop when corresponding skill neurons are perturbed. (2) Skill neurons are task-specific. Similar tasks tend to have similar distributions of skill neurons. Furthermore, we demonstrate the skill neurons are most likely generated in pre-training rather than fine-tuning by showing that the skill neurons found with prompt tuning are also crucial for other fine-tuning methods freezing neuron weights, such as the adapter-based tuning and BitFit. We also explore the applications of skill neurons, including accelerating Transformers with network pruning and building better transferability indicators. These findings may promote further research on understanding Transformers. The source code can be obtained from https://github.com/THU-KEG/Skill-Neuron.

1 Introduction

Pre-trained language models (PLMs), mostly based on Transformer architecture (Vaswani et al., 2017), have achieved remarkable performance on broad and diverse natural language processing (NLP) tasks (Han et al., 2021). However, it remains unclear how the skills required to handle these tasks distribute among model parameters. Are there specific neurons within pre-trained Transformers encoding these skills? Progress on this problem may help to understand the working mechanisms of pre-trained Transformers (Zeiler and Fergus, 2014; Karpathy et al., 2015; Bau et al., 2020; Suau et al., 2020), intervene model behaviors (Bau et al., 2018; Mitchell et al., 2021), and improve model efficiency (Dalvi et al., 2020; Zhang et al., 2021).

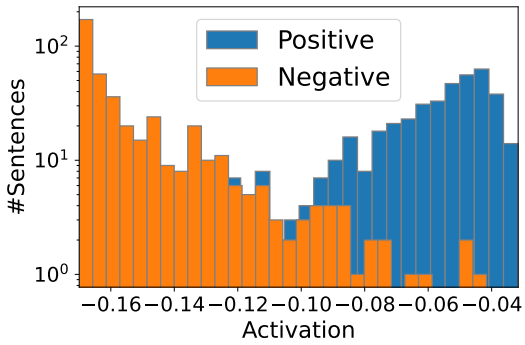


Figure 1: Histogram of activation of a neuron within RoBERTa$_{\text{BASE}}$ on positive-label (blue) and negative-label (orange) sentences in SST-2 validation set.

Prompt tuning (Li and Liang, 2021; Lester et al., 2021) prepends some trainable embeddings, i.e., *soft prompts*, into the inputs and adapts PLMs to handle tasks by only tuning the soft prompts while freezing all the PLM parameters. It has attracted wide attention recently as a promising parameter-efficient fine-tuning methods (Su et al., 2021; Liu et al., 2022). In this paper, we find that after prompt tuning for a task, the activations on soft prompts of some neurons within pre-trained Transformers are **highly predictive** for the task. For instance, Figure 1 shows the activation distribution of a specific neuron within RoBERTa$_{\text{BASE}}$ (Liu et al., 2019b). This neuron's activation is highly predictive of the labels of SST-2 (Socher et al., 2013), an established sentiment analysis dataset. When the input sentences express positive sentiments, the activations on soft prompts of this neuron tend to be much

[*] indicates equal contribution.
[†] Corresponding author: Z.Liu and L.Hou.
[1]For brevity, *Transformer-based language models* are often referred to as *Transformers* in this paper.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11132–11152
December 7-11, 2022

higher than when they express negative sentiments. It suggests that this neuron may encode the skill of distinguishing sentiments.

We dub these special neurons *skill neurons* and develop a simple and effective method to find them for classification tasks via prompt tuning. For a binary classification task, we first calculate the empirical mean activation on a soft prompt token over the training set for each neuron and use it as this neuron's baseline activation. If this neuron's activation for an input sample is higher than the baseline, we regard it as predicting one label and vice versa. We aggregate the prediction accuracies on the validation set of multiple soft prompts as the neuron's predictivity score. The neurons with the highest predictivity scores are identified as skill neurons. For multi-class classification tasks, we decompose them into multiple binary classification subtasks and aggregate the skill neurons of subtasks as the skill neurons of the multi-class task.

We confirm the skill neurons encode task-specific skills with a series of experimental findings: (1) Skill neurons generally and stably emerge. For all the 7 investigated tasks and 5 random trials, we can consistently find skill neurons with high predictivities close to prompt tuning. (2) Skill neurons are crucial for handling tasks. When we perturb skill neurons by adding random noises to their activations, the performances on corresponding tasks drop much more significantly than when random neurons are perturbed. (3) Skill neurons are task-specific. Similar tasks exhibit similar predictivity rankings of skill neurons, and skill neurons of same-type tasks are more important for handling a task than those of different-type tasks. (4) Skill neurons are not from shallow word selectivity. The skill neurons typically do not selectively activate on keywords relating to the task, and their predictivities are not significantly influenced by the label words used in prompt tuning.

After showing that skill neurons encode skills, we further demonstrate that skill neurons are most likely generated in pre-training rather than manufactured by the fine-tuning process of prompt tuning. This is concluded from: (1) Even for randomly generated prompts and untuned hard prompts, the skill neurons still exhibit much better predictivity performance than random guesses. (2) Skill neurons are also crucial for other fine-tuning methods freezing neuron weights. Performance of models trained with adapter-based tuning (Houlsby et al., 2019) and BitFit (Ben-Zaken et al., 2022) significantly drops when the skill neurons found with prompt tuning are perturbed.

Moreover, we explore the practical applications of skill neurons. First, we apply skill neurons to network pruning (Anwar et al., 2017; Dalvi et al., 2020), which aims at removing redundant parameters to reduce memory cost and accelerate inference. Experiments show that by only keeping top skill neurons active, we can reduce the pre-trained Transformer to 66.6% of its original parameters and achieve about 1.4 inference speedup. Then we explore building better prompt transferability indicators following Su et al. (2021). We improve their *overlapping rate of activated neurons* metric by only taking skill neurons into account, and this achieves significantly better performance.

To summarize, our contributions are four-fold: (1) We observe the existence of skill neurons, the special neurons within pre-trained Transformers, which are highly predictive for specific tasks, and develop a method to find them via prompt tuning. (2) We empirically confirm that skill neurons do encode the skills required to handle tasks. (3) We show skill neurons are generated in pre-training rather than fine-tuning. (4) We preliminarily explore the applications of skill neurons. We hope these findings could facilitate future research on understanding the mechanism of PLMs.

2 Preliminary

We introduce the basic knowledge about prompt tuning (§ 2.1), the definition of investigated neurons (§ 2.2), and the investigation setup (§ 2.3).

2.1 Prompt Tuning

Prompt tuning (PT), or soft prompting, is a recently-developed parameter-efficient fine-tuning method, which has attracted wide attention with its capability to effectively adapt PLMs to downstream tasks (Li and Liang, 2021; Lester et al., 2021) and query inner knowledge of PLMs (Qin and Eisner, 2021; Zhong et al., 2021). PT prepends some *soft prompts* into the input sequences to prompt the PLM to decode the desired *label words* of the training task in the same way as the pre-training objective. For each task, a *verbalizer* function (Schick and Schütze, 2021) is used to map the specific label words to the labels of the task. Each soft prompt is a virtual token, which is essentially a trainable embedding. During prompt tuning, only the param-

eters in soft prompts are tuned, and all the PLM's original parameters are frozen.

Formally, given an input sequence with n tokens $X = \{w_1, w_2, \ldots, w_n\}$, prompt tuning prepends l randomly initialized soft prompts $P = \{p_1, p_2, \ldots, p_l\}$ before them, where $p_i \in \mathbb{R}^d$ and d is the input dimension of the PLM. Taking the PLMs pre-trained with the masked language modeling objective (Devlin et al., 2019) as an example, a special `[MASK]` token is prepended, and the prompt tuning objective is to maximize the likelihood of filling desired label word y into it:

$$\mathcal{L} = p(y|[\texttt{MASK}], P, x_1, \ldots, x_n). \quad (1)$$

Some initial prompt tuning works (Qin and Eisner, 2021; Zhong et al., 2021) regard soft prompts as the relaxation of natural language *hard* prompts, which are initially designed to query inner factual knowledge of PLMs (Petroni et al., 2019; Jiang et al., 2020). Su et al. (2021) hypothesize that soft prompts work by stimulating PLMs' inner abilities. Inspired by these, we observe the inner activations of PLMs and find skill neurons.

2.2 Neurons in Transformers

Transformer (Vaswani et al., 2017) is the state-of-the-art NLP model architecture, which is used by the majority of PLMs (Devlin et al., 2019; Liu et al., 2019b; Brown et al., 2020; Raffel et al., 2020). A pre-trained Transformer model is typically stacked with multiple identical Transformer layers. Each Transformer layer consists of a self-attention module and a feed-forward network (FFN), among which the FFN carries two-thirds of the parameters. Previous work has highlighted the importance of FFN (Press et al., 2020; Dong et al., 2021) and found FFN encodes rich information (Suau et al., 2020; Geva et al., 2021; Dai et al., 2021). Inspired by these, we study the neurons and activations within FFN.

Formally, the FFN in a Transformer layer is:

$$\mathrm{FFN}(\mathbf{x}) = f(\mathbf{x}\mathbf{K}^\top + \mathbf{b}_1)\mathbf{V} + \mathbf{b}_2, \quad (2)$$

where $\mathbf{x} \in \mathbb{R}^d$ is the hidden embedding of a token, $f(\cdot)$ is the activation function, $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{d_m \times d}$ are trainable matrices, and $\mathbf{b}_1, \mathbf{b}_2$ are biases.

For simplicity, let $\mathbf{a} = f(\mathbf{x}\mathbf{K}^\top + \mathbf{b}_1) \in \mathbb{R}^{d_m}$. We regard $\mathbf{a}_i$, the i-th element of $\mathbf{a}$, as the activation of the i-th neuron on input $\mathbf{x}$. It represents the importance of $\mathbf{K}_i$ and $\mathbf{V}_i$, the i-th column vectors of $\mathbf{K}$ and $\mathbf{V}$, respectively. Hence we define $\mathbf{K}_i$ and $\mathbf{V}_i$ as the weights of the i-th neuron in this layer.

Although they study essentially the same parameters as us, Dai et al. (2021) and Zhang et al. (2021) use the term neuron to denote activations in our definition. Some other works (Dalvi et al., 2019; Durrani et al., 2020; Hennigen et al., 2020; Antverg and Belinkov, 2022) define a dimension in contextualized representations as a neuron. Since we study how the skills distribute among model parameters rather than input-dependent representations, we study the neurons defined in this section.

2.3 Investigation Setup

To comprehensively investigate the skill neuron phenomenon, we use RoBERTa$_{\text{BASE}}$ (Liu et al., 2019b), a widely-used Transformer model pre-trained with the masked language modeling objective (Devlin et al., 2019), and conduct experiments on 7 tasks of 3 types, including: (1) **Sentiment Analysis**, including `SST-2` (Socher et al., 2013), `IMDB` (Maas et al., 2011), and TweetEval (`Tweet`) (Barbieri et al., 2020); (2) **Natural Language Inference**, including `MNLI` (Williams et al., 2018) and `QNLI` (Wang et al., 2019); (3) **Topic Classification**, including `AG News` and `DBpedia` (Zhang et al., 2015). Details about the tasks and prompt tuning implementations are shown in appendices A and B, respectively.

3 Finding Skill Neurons

We use a simple and effective method to find skill neurons for a given pre-trained Transformer $\mathcal{M}$.

3.1 Binary Classification Task

We first introduce how to find skill neurons for binary classification tasks. Let $\mathcal{T}$ be a binary classification task and its dataset be $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{|D|}, y_{|D|})\}$, which is divided into training set D_{train}, development set D_{dev}, and test set D_{test}. The i-th sample (x_i, y_i) contains an input x_i and its label $y_i \in \{0, 1\}$.

For a specific neuron $\mathcal{N}$ within $\mathcal{M}$, let $a(\mathcal{N}, t, x)$ be the activation of it on token t given the input sentence x. We firstly do prompt tuning on $\mathcal{M}$ with D_{train} and get a group of l soft prompts $P = \{p_1, p_2, \ldots, p_l\}$. Given a soft prompt p_i, we calculate the baseline activation of $\mathcal{N}$ on p_i over the training set as follows:

$$a_{\text{bsl}}(\mathcal{N}, p_i) = \frac{1}{|D_{\text{train}}|} \sum_{x_j, y_j \in D_{\text{train}}} a(\mathcal{N}, p_i, x_j). \quad (3)$$

Intuitively, we can regard that the neuron $\mathcal{N}$ predicts positive label 1 for the input sentence x when $a(\mathcal{N}, p_i, x) > a_{\text{bsl}}(\mathcal{N}, p_i)$. Hence the prediction accuracy over the development set is as follows:

$$\text{Acc}(\mathcal{N}, p_i) = \frac{\sum_{x_j, y_j \in D_{\text{dev}}} \mathbf{1}_{[\mathbf{1}_{[a(\mathcal{N}, p_i, x_j) > a_{\text{bsl}}(\mathcal{N}, p_i)]} = y_j]}}{|D_{\text{dev}}|}, \tag{4}$$

where $\mathbf{1}_{[\text{condition}]} \in \{0, 1\}$ is the indicator function evaluating to 1 iff the condition holds.

The above way only considers the positive correlations between the labels and neuronal activations, which is also the case of previous work (Geva et al., 2021; Dai et al., 2021). However, strong negative correlations also suggest that the information about skills is encoded in this neuron. Conceptually, this is similar to the fact that inhibitory neurons in brains also contribute to certain functions (Rudy et al., 2011). Hence we define the predictivity of $\mathcal{N}$ on soft prompt token p_i as:

$$\text{Pred}(\mathcal{N}, p_i) = \max(\text{Acc}(\mathcal{N}, p_i), 1 - \text{Acc}(\mathcal{N}, p_i)). \tag{5}$$

For each group of soft prompts P, the predictivity of $\mathcal{N}$ on it is defined as the predictivity of the best soft prompt token. Considering the skill neurons shall be consistently predictive, we conduct 5 random trials of prompt tuning and get 5 groups of prompts: $\mathcal{P} = \{P_1, P_2, \ldots, P_5\}$. The overall predictivity of neuron $\mathcal{N}$ is defined as:

$$\text{Pred}(\mathcal{N}) = \frac{1}{|\mathcal{P}|} \sum_{P_i \in \mathcal{P}} \max_{p_j \in P_i}(\text{Pred}(\mathcal{N}, p_j)). \tag{6}$$

Then we sort all the neurons within model $\mathcal{M}$ by the descending order of their predictivities and use the top neurons as the skill neurons in experiments. Appendix G discusses some potential design choices considered in finding skill neurons.

3.2 Multi-class Classification Task

To find skill neurons for a multi-class classification task, we first decompose it into multiple binary classification subtasks. Then we find skill neurons by ranking the neurons with their predictivities of the decomposed subtasks in a similar way as introduced in § 3.1 but use the soft prompts of the original task instead of subtasks. Skill neurons of the multi-class classification task consist of equal numbers of subtask skill neurons. For instance, MNLI (Williams et al., 2018) task requires to classify the relationships between sentence pairs into ENTAILMENT, NEUTRAL and CONTRADICTION. We decompose it into two subtasks: the first one is to classify ENTAILMENT and CONTRADICTION samples, and the second one is to classify NEUTRAL and NON-NEUTRAL samples. If we need top-100 skill neurons of MNLI, we will retrieve top-50 unique skill neurons for the two subtasks, respectively. Figure 2 shows the activation distribution of the two top skill neurons within $\text{RoBERTa}_{\text{BASE}}$ of the two subtasks, respectively. The samples of three labels form three distinguishable clusters, which suggests the effectiveness of this skill-neuron-finding method. More details about how we decompose the investigated tasks are shown in appendix A.

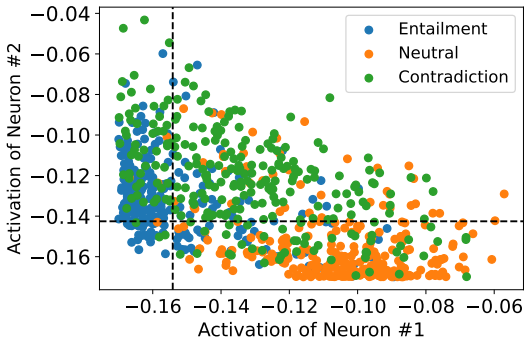


Figure 2: Distribution of activations of two neurons on a soft prompt for samples in MNLI validation set. Dashed lines indicate baseline activations of the two neurons.

Task	Prompt Tuning	Skill Neuron
SST-2	$91.8_{\pm 0.5}$	$91.6_{\pm 0.3}$
IMDB	$91.6_{\pm 0.5}$	$92.0_{\pm 0.3}$
Tweet	$70.0_{\pm 0.2}$	$56.0_{\pm 3.2}$
MNLI	$76.8_{\pm 1.8}$	$74.7_{\pm 2.5}$
QNLI	$85.7_{\pm 0.7}$	$86.0_{\pm 0.4}$
AG News	$98.8_{\pm 0.1}$	$98.9_{\pm 0.1}$
DBpedia	$99.7_{\pm 0.1}$	$99.8_{\pm 0.1}$

Table 1: Accuracies (%) on various tasks of prompt tuning and skill neurons, along with standard deviations over 5 random trials. For the binary classification tasks, the skill neuron performance is the predictivity of the top-1 skill neuron. For multi-class classification tasks, the skill neuron performance is obtained by training a logistic regression model taking only the activations of the top-1 neurons of decomposed subtasks as inputs.

4 Do Skill Neurons Encode Skills?

We explore whether skill neurons really encode task-specific skills with a series of experiments.

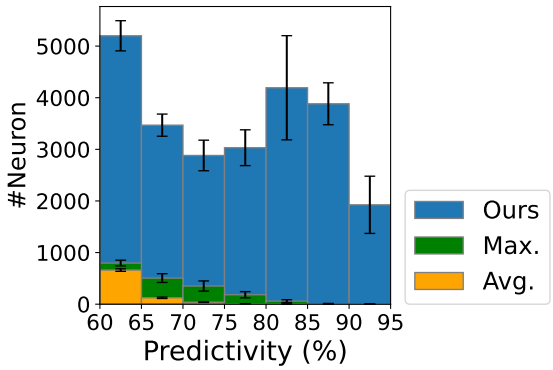


Figure 3: Histogram of neuron's predictivity for IMDB. Error bars indicate ± 1 s.e.m. over 5 random trials.

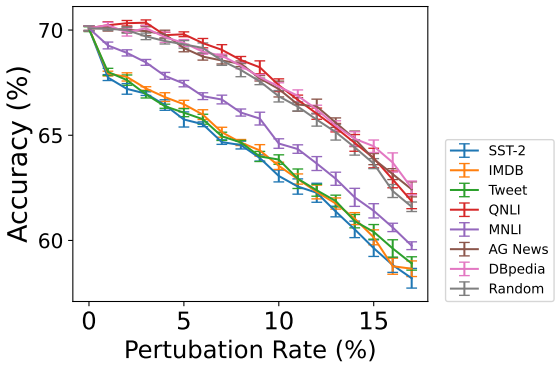


Figure 4: Accuracy on Tweet drops along with the neuron perturbation rate. Error bars indicate ± 1 s.e.m. over 5 random trials. The perturbations are conducted in descending orders of neurons' predictivities for different tasks or in random order (the "Random" curve).

4.1 Skill Neurons Generally and Stably Emerge

We first confirm that the skill neuron phenomenon is general and stable for various NLP tasks.

Generality. To explore whether we can generally find highly-predictive skill neurons for various tasks, we apply the skill-neuron-finding method in § 3 to 7 NLP tasks introduced in § 2.3. The performances of the top-predictivity found skill neurons and prompt tuning are shown in Table 1. For all the tasks, we can find skill neurons achieving comparable performance to prompt tuning, which demonstrates specific skill neurons generally exist in pre-trained Transformers for various tasks.

Stability. To rule out the possibility that the skill neurons are just from randomness and confirm the stability of this phenomenon, we conduct 5 random trails (with different data orders and prompt initializations) to find skill neurons for all the tasks. Figure 3 shows the distributions of neuron predictivities within $\text{RoBERTa}_{\text{BASE}}$ for SST-2 task. Distributions for the other tasks are left in appendix C. We can see that our method can stably find substantial skill neurons with high predictivities via prompts. Previous methods use average (Dai et al., 2021) and maximum (Suau et al., 2020) activations on input tokens instead of activations on prompts to find selective neurons, which are shown as the "Avg." and "Max." results in Figure 3, respectively. The experimental results indicate that previous methods hardly find highly-predictive neurons, which suggests that prompt tuning is crucial for finding skill neurons. We encourage future work to explore the reason why prompt tuning can help in this.

4.2 Skill Neurons are Crucial for Handling Tasks

A natural hypothesis is that if the skill neurons really encode skills, they shall be more important for PLMs to handle various tasks. To verify this, we perturb the skill neurons and see whether PLM's performance drops more than perturbing random neurons. Specifically, the perturbation is to add a Gaussian noise ($\mu = 0$ and $\sigma = 0.1$) into the neurons' activations (Arora et al., 2018), so that the neurons cannot function properly, and then we observe the PLM's prompt tuning performances.

The perturbation results on Tweet task are shown in Figure 4, from which we observe that when we perturb top skill neurons of this task, the PLM's performance drops much more significantly than when we perturb neurons in random order. It indicates that the highly-predictive skill neurons are indeed crucial for handling tasks and supports that skill neurons encode skills. Perturbation results on the other tasks are shown in appendix D.1, and they all exhibit similar phenomena.

4.3 Skill Neurons are Task-specific

We further study whether skill neurons are task-specific, i.e., do skill neurons encode task-specific high-level skills like distinguishing sentiments for sentiment analysis, or do they just encode some task-general low-level skills like recognizing parts of speech, which are also helpful for handling tasks.

First, if skill neurons are task-specific, we shall find similar skill neurons for similar tasks. To verify this, we rank neurons in descending orders of their predictivities for different tasks and see Spear-

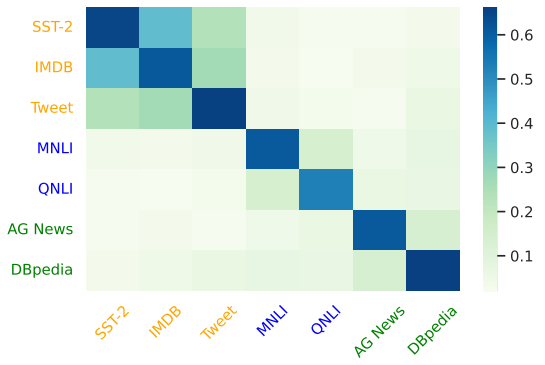


Figure 5: Spearman's rank correlations between the neuron predictivity orders of different tasks. Results are averaged over all the layers.

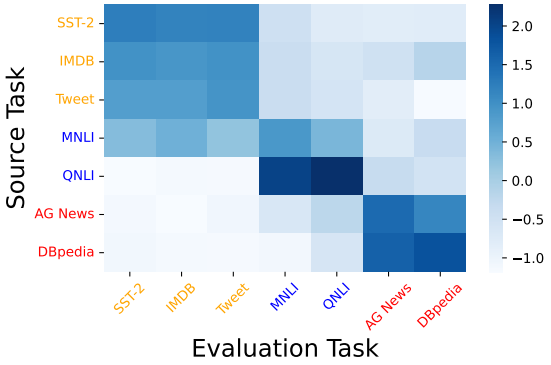


Figure 6: Neuronal importances of different task pairs. Results are averaged over 5 random trials. For an evaluation task, the neuronal importances of different source tasks are normalized as z-scores.

man's rank correlations (Spearman, 1987) between the orders of different tasks. The average results over all the 12 layers of $\text{RoBERTa}_{\text{BASE}}$ are shown in Figure 5. We can see that the correlations between similar tasks of the same type are obviously higher, which confirms that similar tasks have similar skill neurons. The layer-wise correlations are shown in appendix C, from which we can see skill neurons tend to be more task-specific in higher layers, which is consistent with previous probing findings (Liu et al., 2019a).

Moreover, if skill neurons are task-specific, the skill neurons of same-type tasks shall be more important for handling a specific task. This has been supported by Figure 4, which shows that the accuracy on `Tweet` drops much more significantly when we perturb neurons in the predictivity orders of same-type tasks (`SST-2`, `IMDB`). To qualify this effect and comprehensively show this phenomenon in all tasks, we define the *neuronal importance* of a source task to an evaluation task as the area between the accuracy curves obtained by perturbing neurons in the predictivity order of the source task and in random order. For instance, in Figure 4, the neuronal importance of `SST-2` to `Tweet` is the area between the blue curve and the gray curve. The overall neuronal importance is shown in Figure 6, from which we can see the skill neurons of same-type tasks are obviously more important, which strongly supports that the found skill neurons encode task-specific skills again.

4.4 Skill Neurons are not from Word Selectivity

Previous works (Dai et al., 2021; Suau et al., 2020) show that neurons in Transformers may selectively activate on some words or concepts. To confirm that skill neurons encode skills, we show that skill neurons are not from these selectivities.

	Cosine Similarity
Top	`AGES`, `GES`, `ITIES`, `ause`, `UNCH`, `AGE`, `ORK`, `STE`, `TING`, `FE`
Bottom	`sham`, `Nicol`, `bogus`, `Rox`, `Nay`, `contro`, `guy`, `uneven`, `arbitrarily`, `unnatural`
	Average Activation
Top	`starters`, `village`, `oster`, `iddled`, `af`, `mafia`, `aley`, `tired`, `dep`, `ophobic`
Bottom	`official`, `repression`, `illegal`, `called`, `ensible`, `regime`, `abusers`, `should`, `creation`, `refuse`

Table 2: Related words for `SST-2`'s top skill neuron.

We first do case studies on the related words of the top skill neurons, including the words with top and bottom cosine similarities between their input embeddings and the neuron weight vectors (Dai et al., 2021), and the words with top and bottom average activations (Suau et al., 2020). The results of `SST-2` are shown in Table 2. We can see these related words do not convey sentiments, which demonstrates the skill neurons are not from keyword selectivities. Results of the other tasks are shown in appendix F.

Furthermore, considering the prompt tuning method does predictions by decoding label tokens, we need to check whether skill neurons depend on the label words used. If so, it indicates that the skill neurons do not encode the skills for handling tasks but encode the skills for selectively decoding some words. We rule out this possibility by finding that if we use different random words as label words, the

Task	Random Guess	Random Model	Random Prompt	Hard Prompt
SST-2	50.0	$52.8_{\pm 0.4}$	$78.1_{\pm 0.4}$	83.3
IMDB	50.0	$58.0_{\pm 0.7}$	$76.7_{\pm 2.0}$	75.1
Tweet	33.3	$48.3_{\pm 0.0}$	$48.2_{\pm 1.8}$	48.6
MNLI	33.3	$32.2_{\pm 0.4}$	$39.8_{\pm 1.1}$	40.5
QNLI	50.0	$54.3_{\pm 0.8}$	$69.5_{\pm 0.5}$	65.2
AG News	50.0	$62.7_{\pm 0.3}$	$96.0_{\pm 0.3}$	95.9
DBpedia	50.0	$60.9_{\pm 0.4}$	$98.8_{\pm 0.1}$	99.2

Table 3: Accuracies (%) on various tasks of top skill neurons found with random prompts and untuned hard prompts, compared to random guess and random model. We also report standard deviations over 5 random trials.

achieved predictivity orders of neurons are pretty consistent. Specifically, for all the tasks, the average Spearman's correlation between the neuron predictivity orders of 5 random label words is 0.87.

5 Where do Skill Neurons Come from?

In § 4, we confirm that skill neurons do encode task-specific skills. Then a natural question is where skill neurons come from, i.e., do skill neurons acquire these skills in pre-training or prompt tuning? We find that skill neurons are most likely **generated in pre-training** with empirical evidence.

We first try to find skill neurons with tuning-free prompts, including random prompts, which are randomly generated embeddings, and human-written hard prompts. The predictivities of the found neurons are shown in Table 3. We can see that even without tuning, we can still find neurons with non-trivial predictivities. Malach et al. (2020) shows that randomly initialized neural networks may have predictive subnetworks. Hence we also compare with randomly initialized models using random prompts. It can be observed that the neurons in random models are predictive to some extent, but their predictivities are far below the neurons in pre-trained models. These results imply that the skill neurons are generated in pre-training, and prompt tuning only serves as an effective tool to observe the specificity of these neurons.

To provide stronger evidence, we explore whether the skill neurons found with prompt tuning are also important for other fine-tuning methods with different dynamics. We explore two parameter-efficient fine-tuning methods, including adapter-based tuning (Houlsby et al., 2019), which only tunes the additional adapter layers plugged in Transformers, and BitFit (Ben-Zaken et al., 2022), which only tunes the bias vectors. The two tuning

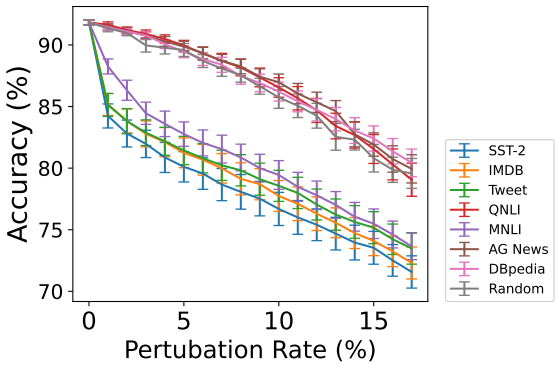


Figure 7: BitFit accuracy on IMDB drops along with the neuron perturbation rate. Error bars indicate ± 1 s.e.m. over 5 random trials. The perturbations are conducted in predictivity orders obtained with prompt tuning.

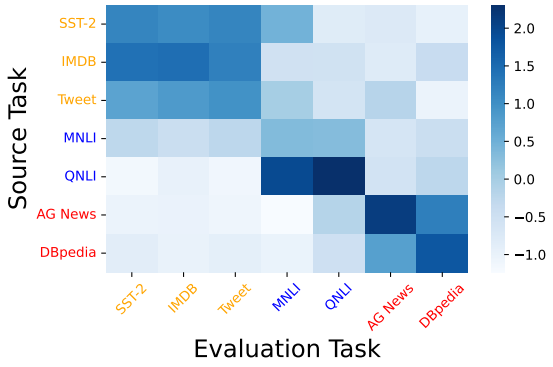


Figure 8: Average neuronal importance over models trained with adapter-based tuning and BitFit.

methods both keep neuron weights fixed, which ensures that the skill neurons are unchanged during tuning. BitFit model's performances on IMDB when neurons are perturbed in the descending orders of predictivities obtained with prompts are shown in Figure 7, and the results for other tasks and adapter models are shown in appendix D. We can see the highly-predictive skill neurons found with prompts are still crucial for models fine-tuned with other methods. To comprehensively show this effect, similar to § 4.3, we visualize the average neuronal importance over models trained with adapter-based tuning and BitFit in Figure 8. The skill neurons found with prompt tuning also exhibit task-specific importance, which again supports that skill neurons are generated in pre-training rather than manufactured by prompt tuning.

6 Application

We further explore the applications of our skill neuron finding. We show two preliminary use cases:

Task	Prompt Tuning	Pruned Model	Speedup
SST-2	$91.8_{\pm 0.5}$	$89.3_{\pm 2.0}$	1.34
IMDB	$91.6_{\pm 0.5}$	$87.6_{\pm 3.0}$	1.34
Tweet	$70.0_{\pm 0.2}$	$69.0_{\pm 0.9}$	1.34
MNLI	$76.8_{\pm 1.8}$	$70.0_{\pm 1.1}$	1.38
QNLI	$85.7_{\pm 0.7}$	$81.0_{\pm 1.0}$	1.36
AG News	$98.8_{\pm 0.1}$	$99.8_{\pm 0.1}$	1.32
DBpedia	$99.7_{\pm 0.1}$	$99.0_{\pm 0.1}$	1.33

Table 4: Accuracies (%) on various tasks of vanilla prompt tuning and prompt tuning on pruned models, along with standard deviations over 5 random trials. We also report the achieved inference speedups on the tasks. Speedups are evaluated on a single CPU since it is widely used for model inference (Mittal et al., 2021).

network pruning and transferability indicator.

6.1 Network Pruning

First, we apply our skill neuron finding to network pruning (Anwar et al., 2017; Dalvi et al., 2020), which is to reduce memory cost and accelerate inference by removing redundant parameters in neural networks. Existing works have explored prune PLMs with weight magnitude (Han et al., 2015; Gordon et al., 2020) and loss attribution (Michel et al., 2019). Here we explore prune PLMs by only keeping the top 2% skill neurons active for each task and set the activations of the 98% frozen neurons always as their baseline activations. Considering that the frozen neurons are fixed, we merge them into bias terms. We apply this pruning method to the top 9 layers of $\text{RoBERTa}_{\text{BASE}}$ and reduce it to 66.6% of its original parameters. The performances of prompt tuning on pruned models and vanilla prompt tuning on the original model are shown in Table 4. Our pruning based on skill neurons generally performs comparably to vanilla prompt tuning and can achieve about 1.4 inference speedup.

6.2 Transferability Indicator

Previous works (Su et al., 2021; Vu et al., 2021) explore improving prompt tuning with cross-task prompt transfer. Su et al. (2021) propose that the *overlapping rate of activated neurons* (ON) between soft prompts can serve as a prompt transferability indicator, which has good correlations with zero-shot prompt transferability and can help to qualify task similarities and improve prompt transfer. Su et al. (2021) take all neurons into ON calculation, but the redundant neurons without task-specific skills may bring noisy signals. Here we only take the top 20% skill neurons of target tasks into the calculation. This improves the average Spearman's correlation between ON and prompt transferability over our tasks from 0.53 to 0.71.

7 Related Work

Selective Neurons in Artificial Neural Networks There have long been findings about selective neurons in artificial neural networks. Many computer vision works (Coates et al., 2012; Le et al., 2013; Zeiler and Fergus, 2014; Agrawal et al., 2014; Zhou et al., 2015; Bau et al., 2020) find that both supervised and unsupervised models can have units selectively respond to specific visual objects and concepts. Radford et al. (2017) also find neurons corresponding to sentiments in unsupervised long short-term memory networks. Interestingly, there are similar selective neurons in human brains (Barlow, 1972; Quiroga et al., 2005). The widespread emergence of these neuronal selectivities implies that there may be common learning mechanisms among intelligent systems, which is extremely worthwhile to explore in the future.

Bau et al. (2017) and Mu and Andreas (2020) find that selective neurons are more important, which is consistent with our findings. However, Morcos et al. (2018) draw opposite conclusions. We discuss this with experiments in appendix H.

Analyzing Pre-trained Transformers After the success of Transformer-based PLMs (Devlin et al., 2019; Yang et al., 2019; Raffel et al., 2020), many efforts have been devoted to analyzing how PLMs work, such as probing the knowledge of PLMs (Liu et al., 2019a; Hewitt and Manning, 2019; Petroni et al., 2019) and understanding the behaviors of PLMs' parameters (Voita et al., 2019; Clark et al., 2019). Among these, some works (Dalvi et al., 2019; Durrani et al., 2020; Antverg and Belinkov, 2022) find that individual neurons capture linguistic properties, but they define neurons as dimensions in contextualized representations. Other works (Suau et al., 2020; Geva et al., 2021; Dai et al., 2021) study the same group of neurons as us and find that some neurons encode specific information like concepts, facts, and word patterns. Inspired by them, we study whether neurons encode high-level skills for handling tasks in this work and demonstrate that we can observe skill neurons with the help of prompts. We believe it is promising to explore whether and how skill neurons collaborate with the neurons encoding information in future works.

8 Conclusion and Future Work

In this paper, we find some special neurons in pre-trained Transformers whose activations on soft prompts are highly predictive of the task labels of inputs. We dub these neurons skill neurons and develop a method to find them via prompt tuning. With extensive experiments, we confirm that skill neurons encode task-specific skills required to handle these tasks and find empirical evidence showing that skill neurons are most likely generated in pre-training rather than fine-tuning. We also demonstrate some practical applications of our skill neuron finding. In the future, we will extend our prompt-based skill neuron finding method to more scenarios, such as covering non-classification tasks and other parameters in Transformers like attention heads. We will also explore more fundamental problems about skill neurons and the working mechanisms of PLMs, including how the skill neurons emerge in pre-training, as well as the relationships between skill neurons and neurons encoding specific information found in previous works.

Limitations

Although we conducted extensive experiments, the exploration scope of this work has some limitations: (1) The experimental analyses are all based on $\text{RoBERTa}_{\text{BASE}}$. Whether the skill neuron phenomenon widely exists for other Transformer-based pre-trained language models is unclear and more explorations are needed to verify it. (2) The datasets used in our experiments are all English, which limits the linguistic features covered in our analyses, and the evaluation tasks are limited to classification tasks. We choose English just because of its rich resource. Although we intuitively believe the observed phenomena are not dependent on the English language, experiments on more diverse languages are needed in future works. (3) Following previous works (Geva et al., 2021; Dai et al., 2021), the analyzed neurons in our work all distribute in the feed-forward layers of Transformers. Deeper analyses may require considering other parameters like the attention heads. We encourage future works to address these limitations and get more comprehensive analysis results.

Acknowledgements

This work is supported by the New Generation Artificial Intelligence of China (2020AAA0106501), the Institute for Guo Qiang, Tsinghua University (2019GQB0003), and Huawei Noah's Ark Lab. We thank anonymous reviewers for their suggestions.

References

Pulkit Agrawal, Ross B. Girshick, and Jitendra Malik. 2014. Analyzing the performance of multilayer neural networks for object recognition. In *Proceedings of ECCV*, pages 329–344.

Omer Antverg and Yonatan Belinkov. 2022. On the pitfalls of analyzing individual neurons in language models. In *Proceedings of ICLR*.

Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. 2017. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):1–18.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. 2018. Stronger generalization bounds for deep nets via a compression approach. In *Proceedings of ICML*, pages 254–263.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *Proceedings of ISWC/ASWC*, pages 722–735.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of EMNLP*, pages 1644–1650.

Horace B Barlow. 1972. Single units and sensation: A neuron doctrine for perceptual psychology? *Perception*, 1(4):371–394.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2018. Identifying and controlling important neurons in neural machine translation. In *Proceedings of ICLR*.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. *Proceedings of CVPR*, pages 3319–3327.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. 2020. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078.

Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of ACL*.

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. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*, pages 1877–1901.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Adam Coates, Andrej Karpathy, and A. Ng. 2012. Emergence of object-selective features in unsupervised feature learning. In *Proceedings of NeurIPS*, pages 2681–2689.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint, arXiv:2104.08696*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. 2019. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of AAAI*, pages 6309–6317.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of EMNLP*, pages 4908–4926.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. 2021. Attention is not all you need: pure attention loses rank doubly exponentially with depth. In *Proceedings of ICML*, pages 2793–2803.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of EMNLP*, pages 4865–4880.

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.

Mitchell A Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing bert: Studying the effects of weight pruning on transfer learning. *arXiv preprint arXiv:2002.08307*.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *Proceedings of NeurIPS*, pages 1135–1143.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, et al. 2021. Pre-trained models: Past, present and future. *AI Open*, pages 225–250.

Lucas Torroba Hennigen, Adina Williams, and Ryan Cotterell. 2020. Intrinsic probing through dimension selection. In *Proceedings of EMNLP*, pages 197–216.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of NACCL-HLT*, pages 4129–4138.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of ICML*, pages 2790–2799.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, pages 818–833.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Gregory S. Corrado, Kai Chen, Jeffrey Dean, and A. Ng. 2013. Building high-level features using large scale unsupervised learning. In *Proceedings of ICASSP*, pages 8595–8598.

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.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of EMNLP*, pages 175–184.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of ACL*, pages 4582–4597.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual

representations. In *Proceedings of NAACL-HLT*, pages 1073–1094.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-Tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. In *Proceedings of ACL*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907,11692*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL-HLT*, pages 142–150.

Eran Malach, Gilad Yehudai, Shai Shalev-shwartz, and Ohad Shamir. 2020. Proving the lottery ticket hypothesis: Pruning is all you need. In *Proceedings of ICML*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Proceedings of NeurIPS*, pages 14014–14024.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. In *Proceedings of ICLR*.

Sparsh Mittal, Poonam Rajput, and Sreenivas Subramoney. 2021. A survey of deep learning on CPUs: opportunities and co-optimizations. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21.

Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. 2018. On the importance of single directions for generalization. In *Proceedings of ICLR*.

Jesse Mu and Jacob Andreas. 2020. Compositional explanations of neurons. In *Proceedings of NeurIPS*, pages 17153–17163.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of EMNLP-IJCNLP*, pages 2463–2473.

Ofir Press, Noah A. Smith, and Omer Levy. 2020. Improving transformer models by reordering their sublayers. In *Proceedings of ACL*, pages 2996–3005.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of NAACL-HLT*, pages 5203–5212.

R Quian Quiroga, Leila Reddy, Gabriel Kreiman, Christof Koch, and Itzhak Fried. 2005. Invariant visual representation by single neurons in the human brain. *Nature*, 435(7045):1102–1107.

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

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. *Journal of Machine Learning Research*, 21:1–67.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of SemEval*, pages 502–518.

Bernardo Rudy, Gordon Fishell, SooHyun Lee, and Jens Hjerling-Leffler. 2011. Three groups of interneurons account for nearly 100% of neocortical gabaergic neurons. *Developmental neurobiology*, 71(1):45–61.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of EACL*, pages 255–269.

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. 1987. The proof and measurement of association between two things. In *Proceedings of AJP*, 3/4, pages 441–471.

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, et al. 2021. On transferability of prompt tuning for natural language understanding. *arXiv preprint arXiv:2111.06719*.

Xavier Suau, Luca Zappella, and Nicholas Apostoloff. 2020. Finding experts in transformer models. *arXiv preprint arXiv:2005.07647*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*, pages 5998–6008.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of NAACL*, pages 5797–5808.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2021. Spot: Better frozen model adaptation through soft prompt transfer. *arXiv preprint arxiv:2110.07904*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of ICLR*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, pages 1112–1122.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP*, pages 38–45.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of NeurIPS*, pages 5754–5764.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Proceedings of ECCV*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of NeurIPS*, pages 649–657.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2021. Moefication: Conditional computation of transformer models for efficient inference. *arXiv preprint arXiv:2110.01786*.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of NAACL*, pages 5017–5033.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. 2015. Object detectors emerge in deep scene cnns. In *Proceedings of ICLR*.

Appendices

A Details about Investigated Tasks

In experiments, we use 7 established public English NLP datasets, which are licensed and intended for research use. These datasets are all created with public texts, and we believe they do not involve personal information and are well anonymized. The details about the datasets are as follows:

A.1 Sentiment Analysis

SST-2 (Socher et al., 2013) requires to classify the sentiments expressed in movie reviews into POSITIVE and NEGATIVE sentiments.

IMDB (Maas et al., 2011) requires to classify the sentiments expressed in reviews from the Internet Movie Database[2] into POSITIVE and NEGATIVE sentiments.

TweetEval (Barbieri et al., 2020) is a collection of 7 Twitter-specific classification tasks. Here we use its sentiment analysis subtask, which is originally from SemEval 2017 Task 4 (Rosenthal et al., 2017). It requires to recognize if a tweet is POSITIVE, NEGATIVE or NEUTRAL. We decompose it to two subtasks: POSITIVE vs. NEGATIVE, and NEURAL vs. NON-NEUTRAL.

A.2 Natural Language Inference

MNLI (Williams et al., 2018) requires to recognize the relationship between sentence pairs as ENTAILMENT, NEUTRAL and CONTRADICTION. We decompose it to two subtasks: ENTAILMENT vs. CONTRADICTION, and NEURAL vs. NON-NEUTRAL.

QNLI (Wang et al., 2019) requires to classify whether a context sentence contains the answer to a question.

A.3 Topic Classification

AG News (Zhang et al., 2015) requires to classify the 4 topics of news articles in the AG's corpus[3].

DBpedia (Zhang et al., 2015) requires to classify the 14 topics of articles in DBpedia (Auer et al., 2007).

Since recognizing different topics requires essentially different skills, we use the only two similar labels of the two tasks. They are BUSINESS and SPORTS in `AG News`, and COMPANY and ATHLETE in `DBpedia`.

[2] https://www.imdb.com

[3] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

Task	Training	Validation	Test
SST-2	53,879	13,470	872
IMDB	20,000	5,000	25,000
Tweet	45,615	2,000	12,284
MNLI	314,161	78,541	9,815
QNLI	83,794	20,949	5,463
AG News	47,966	12,034	3,800
DBpedia	63,899	16,100	9,999

Table 5: Data statistics of the 7 used datasets.

We obtain the datasets from Huggingface's dataset platform (Lhoest et al., 2021). For the datasets included in the GLUE collection (Wang et al., 2019), since we cannot get their test set, we use the released validation set as our test set, 80% random samples from the original training set as our training set, and the other 20% samples as our validation set. The detailed data statistics are shown in Table 5.

B Implementations Details

We implement the prompt tuning method introduced in § 2.1 with $l = 127$ soft prompts. We randomly initialize each soft prompt using a normal distribution with the standard deviation as 0.03. We then train the model using Adam (Kingma and Ba, 2015) as the optimizer. We set the learning rate as 0.001 and the batch size as 8. We do the evaluation on the validation set every 2,000 iterations and early stop the training if the validation accuracy does not rise for 6 times. We use label words `Negative`, `Positive` for binary classification tasks and `Negative, Neutral, Positive` for multi-class classification tasks. For the random label words experiment in § 4.4, we uniformly sample the label words from the vocabulary of RoBERTa (Liu et al., 2019b).

We conduct all experiments on RoBERTa$_{\text{BASE}}$ model, which has 110M parameters, and we use Huggingface's Transformers library (Wolf et al., 2020) to implement the experiments. We run the experiments on NVIDIA GeForce RTX 2080 Ti and NVIDIA GeForce RTX 3090 GPUs, and it takes about 1000 GPU hours.

C More Predictivity Distributions

We report the predictivity distribution for IMDB in § 4.1 and show the distributions for the other 4 binary classification tasks in Figure 9. We can see our method can stably find many highly-predictive skill neurons for all the tasks. For the multi-class

classification tasks, since the predictivities are for decomposed subtasks, we cannot draw distributions for the original tasks and do not include them in the results here.

D More Neuron Perturbation Results

Here we demonstrate more neuron perturbation experimental results.

D.1 Performance Dropping Trends for Prompt Tuning

In Figure 4, we show the performance dropping trend on `Tweet` task. The results on the other tasks are shown in Figure 11.

D.2 Performance Dropping Trends for Adapter-based Tuning

The performance dropping trends of adapter-based tuning models on various tasks are shown in Figure 12.

D.3 Performance Dropping Trends for BitFit

The performance dropping trends of BitFit models on various tasks are shown in Figure 13.

E Layer-wise Correlations between Neuron Predictivity Orders of Different Tasks

Figure 5 shows the overall Spearman's rank correlations between the neuron predictivity orders of different tasks, which is averaged over the 12 layers of $\text{RoBERTa}_{\text{BASE}}$. Here we further present the layer-wise correlations in Figure 14, from which we can see the skill neurons are more and more task-specific from the bottom layer to the top layer, which is consistent with the probing findings (Liu et al., 2019a) showing that PLMs tend to learn general skills in the lower layers and learn specific skills in the higher layers. These results suggest that our neuron-finding method can find both neurons encoding general skills in the lower layers and neurons encoding specific skills in the lower layers, but the found top skill neurons are task-specific in general (Figure 5). In this work, we focus on the task-specific top skill neurons and leave careful study for the neurons encoding general skills in future work.

F More Word Selectivity Results

In Table 2, we show the related words for `SST-2`. Here we further show the results for the other tasks in Table 6. We can see these related words generally do not convey clues about solving the tasks.

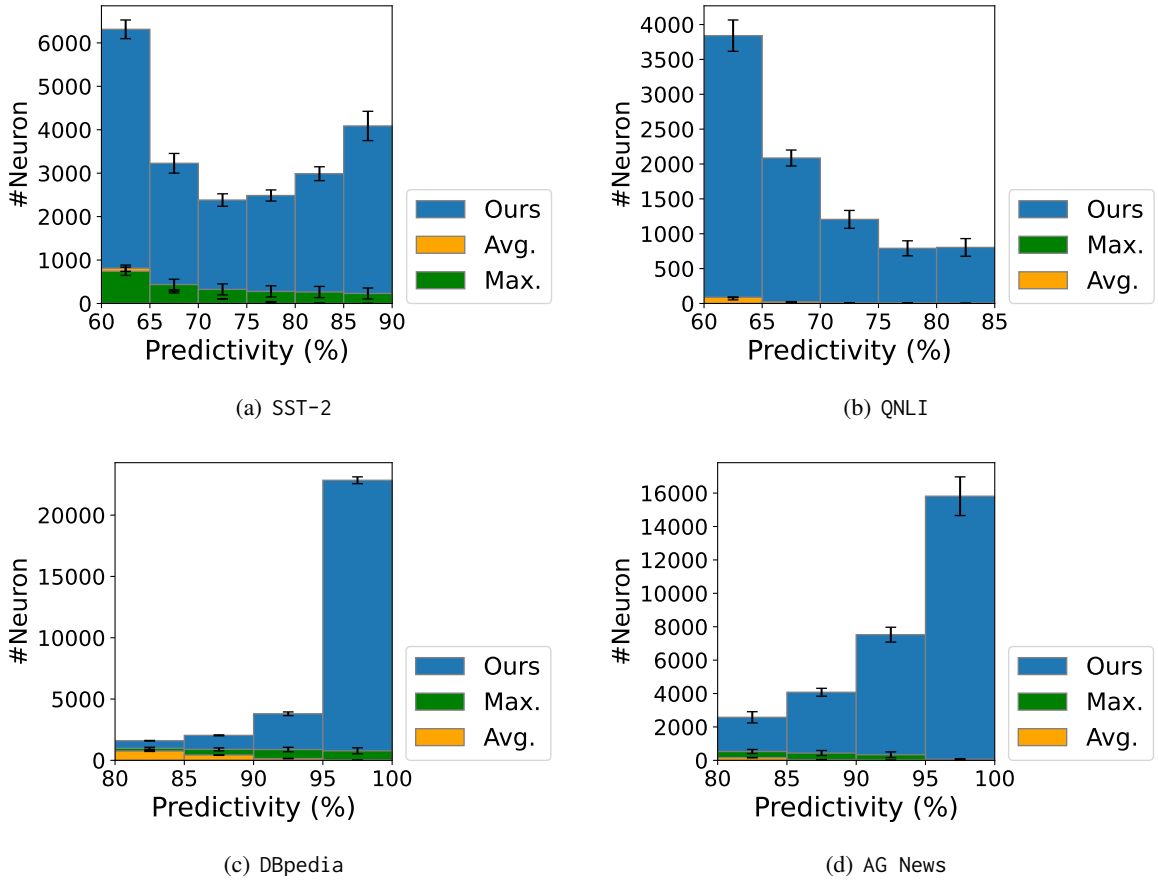
G Discussions on Neuron-Finding Design Choices

In this section, we discuss some potential other design choices that may be used in finding important skill neurons to provide more background about why we choose the method described in § 3 finally and inspire future works.

Perturbation-based neuron finding. A natural way to define the importance of a neuron (to a task) is to perturb the neurons and see how they influence the predictions. The perturbation-based method has been used in previous analysis works (Michel et al., 2019), and we also adopt them in our analytical experiments. But we and many other neuron-level analysis works (Dalvi et al., 2019; Durrani et al., 2020; Antverg and Belinkov, 2022; Suau et al., 2020; Geva et al., 2021; Dai et al., 2021) cannot directly use this method to locate important neurons. This is because of the efficiency issue. Perturbing every individual neuron is unaffordable.

Is prompt tuning necessary? This work starts from an interesting empirical finding, i.e., the skill neuron phenomenon. This finding is based on prompt tuning. In § 4 and Figure 3, we show that previous methods without prompt tuning cannot well locate the skill neurons. Since we focus on confirming the finding and exploring the properties of skill neurons, we conduct all the experiments based on prompt tuning and do not explore whether it is necessary. Intuitively, as our experiments suggest that the emergence of skill neurons does not depend on prompt tuning but is mostly an intrinsic property for pre-trained Transformer-based language models, we believe prompt tuning may not be the only way to locate skill neurons. We will explore other methods without prompt tuning in future works, which may bring some benefits, like improving overall efficiency.

Other ways to define neuron's predictivity. In § 3.1, we define the predictivity of a neuron (1) using the maximum over prompt tokens and (2) considering both the positive and negative correlations. These two choices are made with preliminary experiments. Figure 10 shows an example, from which we can see that when defining neuron's predictivity using the mean values over prompt tokens



(a) SST-2 (b) QNLI

(c) DBpedia (d) AG News

Figure 9: Histograms of predictivity for various tasks on neurons within RoBERTa$_{\text{BASE}}$. Error bars indicate ± 1 s.e.m. over 5 random trials.

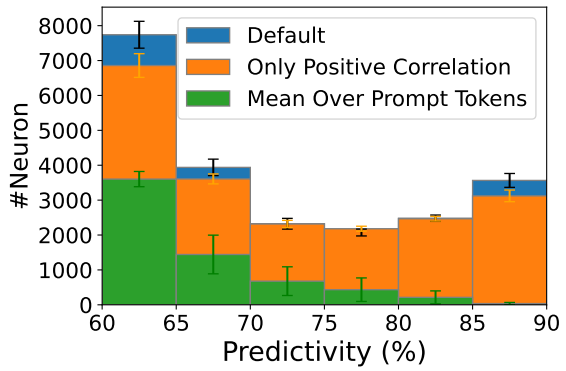


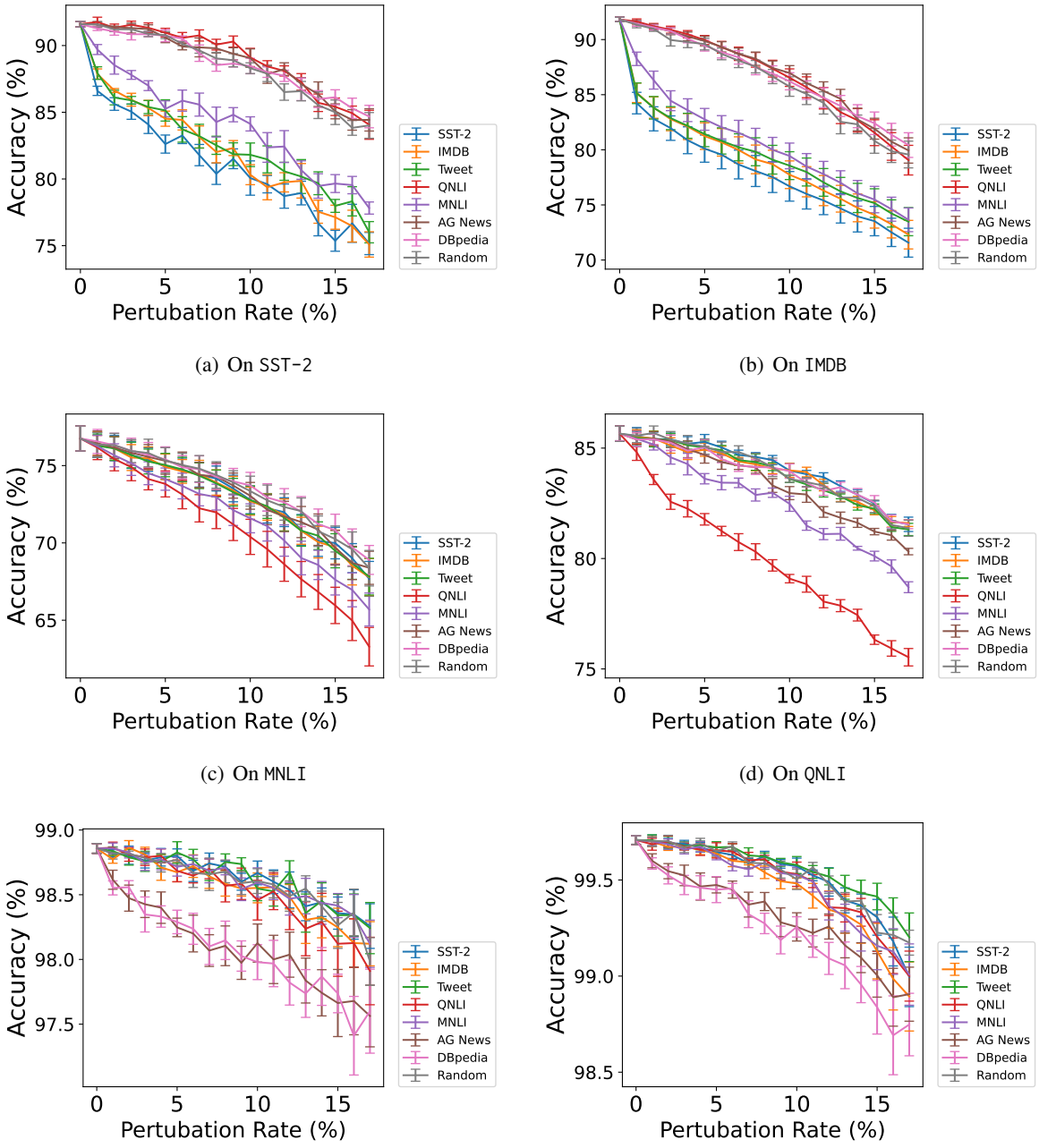
Figure 10: Histogram of neuron's predictivity in different definitions for SST-2. Error bars indicate ± 1 s.e.m. over 5 random trials.

or only considering the positive correlations, the predictivities will be significantly under-estimated than the default definition in § 3.1.

H Experiments following Morcos et al. (2018)

Some previous works (Bau et al., 2017; Mu and Andreas, 2020) suggest that selective neurons contribute more to model accuracies. In § 4, we also find that perturbing selective skill neurons leads to more performance drop. However, Morcos et al. (2018) draw opposite conclusions and find that selective and non-selective neurons are similarly important. These pose questions about why these conclusions are inconsistent.

We find that except for experimental setups, the main difference between Morcos et al. (2018) and ours lies in the definition of neuronal selectivity. Morcos et al. (2018) define a "selectivity index" and we use the predictivity score introduced in § 3. To check whether these different definitions lead to inconsistent results, we do experiments under our setup and also try to perturb neurons in descending orders of their "selectivity index". The results are shown in Figure 15. We can see that when using the "selectivity index", the found neurons are surely not



(a) On SST-2 (b) On IMDB

(c) On MNLI (d) On QNLI

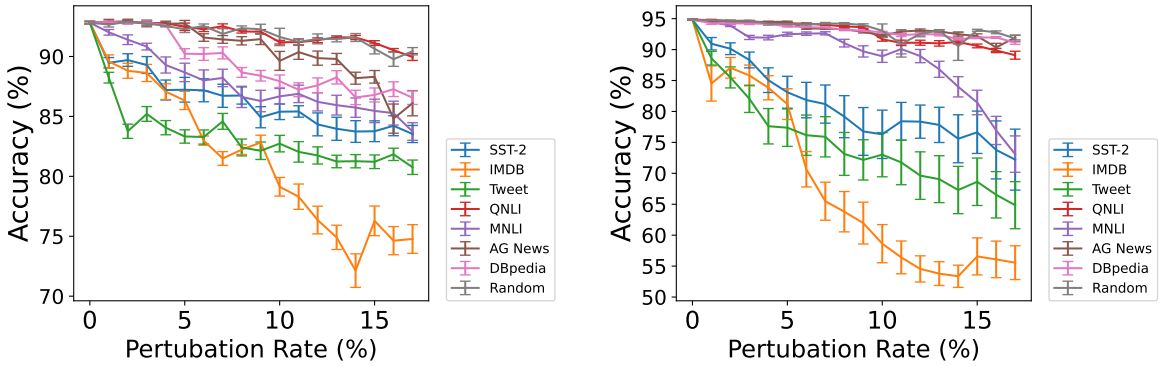
(e) On AG News (f) On DBpedia

Figure 11: Accuracies on various tasks drop along with the neuron perturbation rates. Error bars indicate ± 1 s.e.m. over 5 random trials. The perturbations are conducted in descending orders of neurons' predictivities for different tasks or in random order (the "Random" curve).

more important than random neurons as reported by Morcos et al. (2018). But our predictivity metric can find significantly more important neurons for all the tasks.

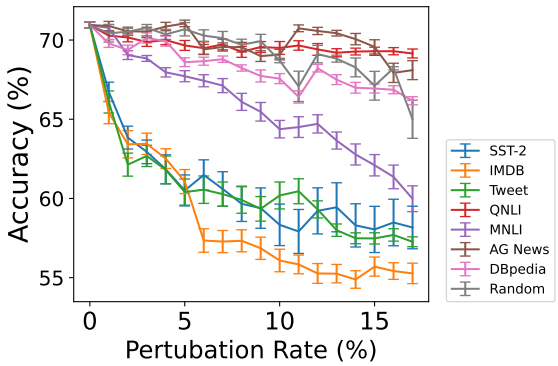
IMDB		Cosine Similarity
	Top	legged, turnout, ladder, heid, flexible, Quite, contrary, runs, Reference, enqu
	Bottom	qq, qa, Capture, Import, Tripoli, hereby, eus, ,, rip, Lima
		Average Activation
	Top	success, Kund, Sanctuary, Lim, Wave, dele, Crystal, flung, Kerala, ...
	Bottom	vation, goodbye, concludes, bye, Congratulations, Congratulations, Fare, farewell, BY, ceremony,
Tweet		Cosine Similarity
	Top	atican, uras, isman, anan, Luck, Merit, Character, alth, atching, character,
	Bottom	Register, enzymes, elsen, Registrar, tasting, regist, soils, µ, Chambers, LINE,
		Average Activation
	Top	dh, Titan, utable, exited, iOS, chel, loophole, acious, 520, Harmony,
	Bottom	spike, unbelievably, Toxic, prov, RIS, resulting, risks, rising, ues, reapp,
MNLI		Cosine Similarity
	Top	trigger, Pis, deadlines, Launch, mares, PROGRAM, Congratulations, Success, Congratulations, Gig,
	Bottom	minim, xt, spoof, dism, avoid, asive, WN, offset, inter, antiqu,
		Average Activation
	Top	nickel, grun, cluded, 91, handled, secure, very, dairy, gent, Roses,
	Bottom	ayed, disl, ect, wipes, screwed, resistance, aw, ruin, shrinking, spite,
QNLI		Cosine Similarity
	Top	otyp, disemb, sidel, melanch, unint, outwe, umbnails, precedence, unfl, Sym,
	Bottom	314, 223, 313, 234, ,, 316, 341, 463, 238, 261,
		Average Activation
	Top	eds, adding, apocalypse, strawberry, apopt, Kid, leaf, Silent, technical,
	Bottom	entrepreneurial, Econom, Columb, prime, roleum, Trade, rounded, isner, enz, 158,
AG News		Cosine Similarity
	Top	aukee, erity, lambda, ropolitan, roxy, LAN, ylon, incinn, oslav, coni,
	Bottom	Gross, Villa, Uri, ende, Summary, Gallup, Temp, Rog, RP, Ram,
		Average Activation
	Top	fight, desert, Merge, Mail, Mid, Rankings, istic, **, berries, Pen,
	Bottom	ETS, 107, Line, 106, observers, Ranked, EB, ido, Bass, alf,
DBpedia		Cosine Similarity
	Top	ming, umbered, hind, utter, pepper, scr, increment, usher, empt, atmospheric,
	Bottom	Chron, kan, Div, Case, Thread, Role, Crash, Mode, Tank, Apps,
		Average Activation
	Top	Bubble, mailed, Ari, razen, Perspective, ogical, Gin, Disney, icons, Huang,
	Bottom	Jacob, Boss, Dad, trough, Shiny, carn, Gravity, toolbar, Sword, temple,

Table 6: Related words for various tasks' top skill neurons.

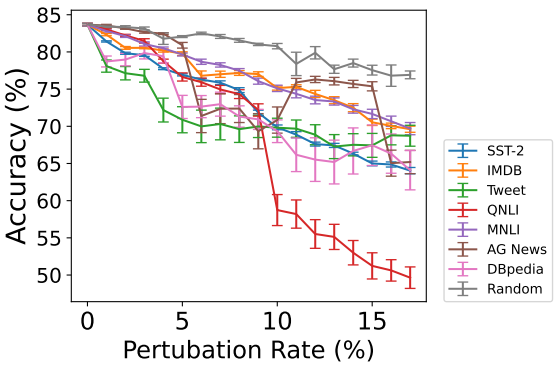


(a) On SST-2

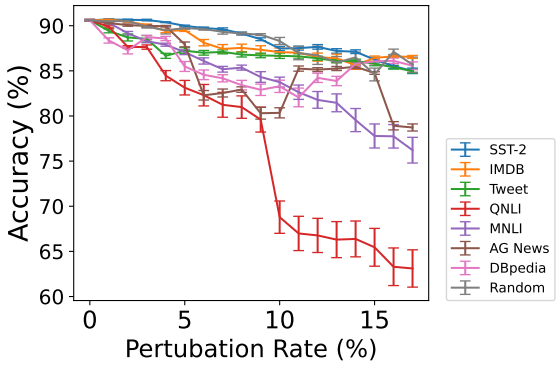
(b) On IMDB



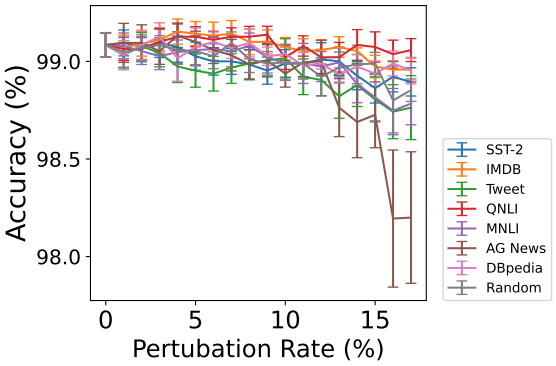
(c) On Tweet



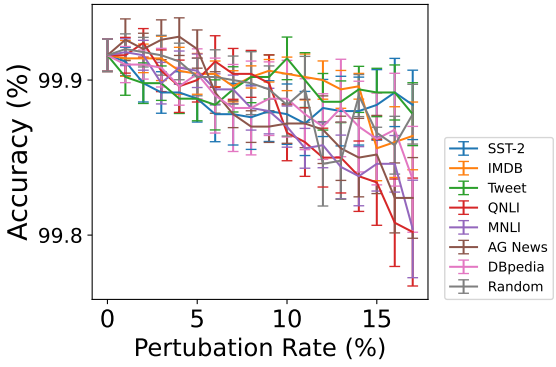
(d) On MNLI



(e) On QNLI

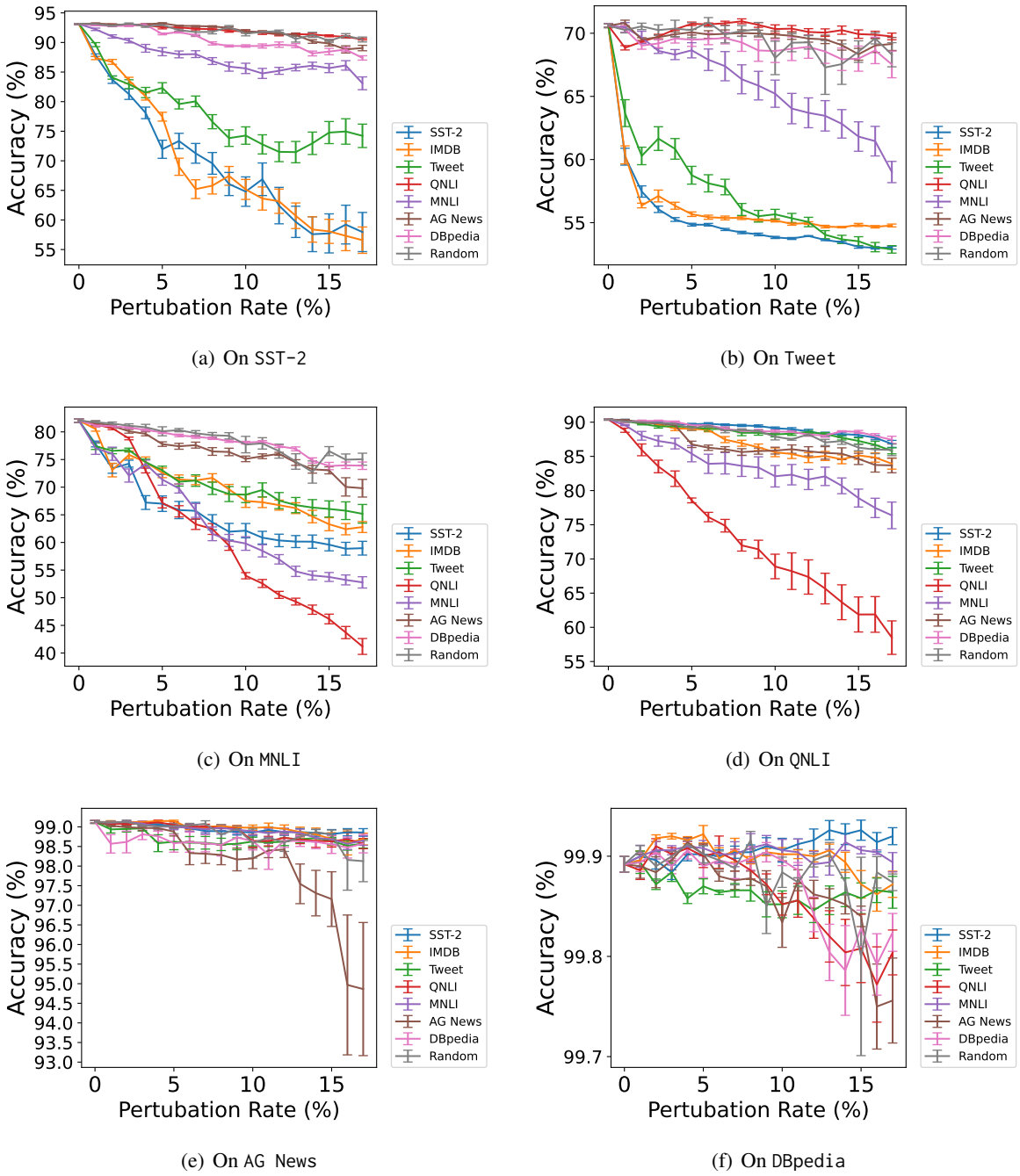


(f) On AG News



(g) On DBpedia

Figure 12: Adapter-based tuning accuracies on various tasks drop along with the neuron perturbation rates. Error bars indicate ± 1 s.e.m. over 5 random trials. The perturbations are conducted in predictivity orders obtained with prompt tuning.



(a) On SST-2

(b) On Tweet

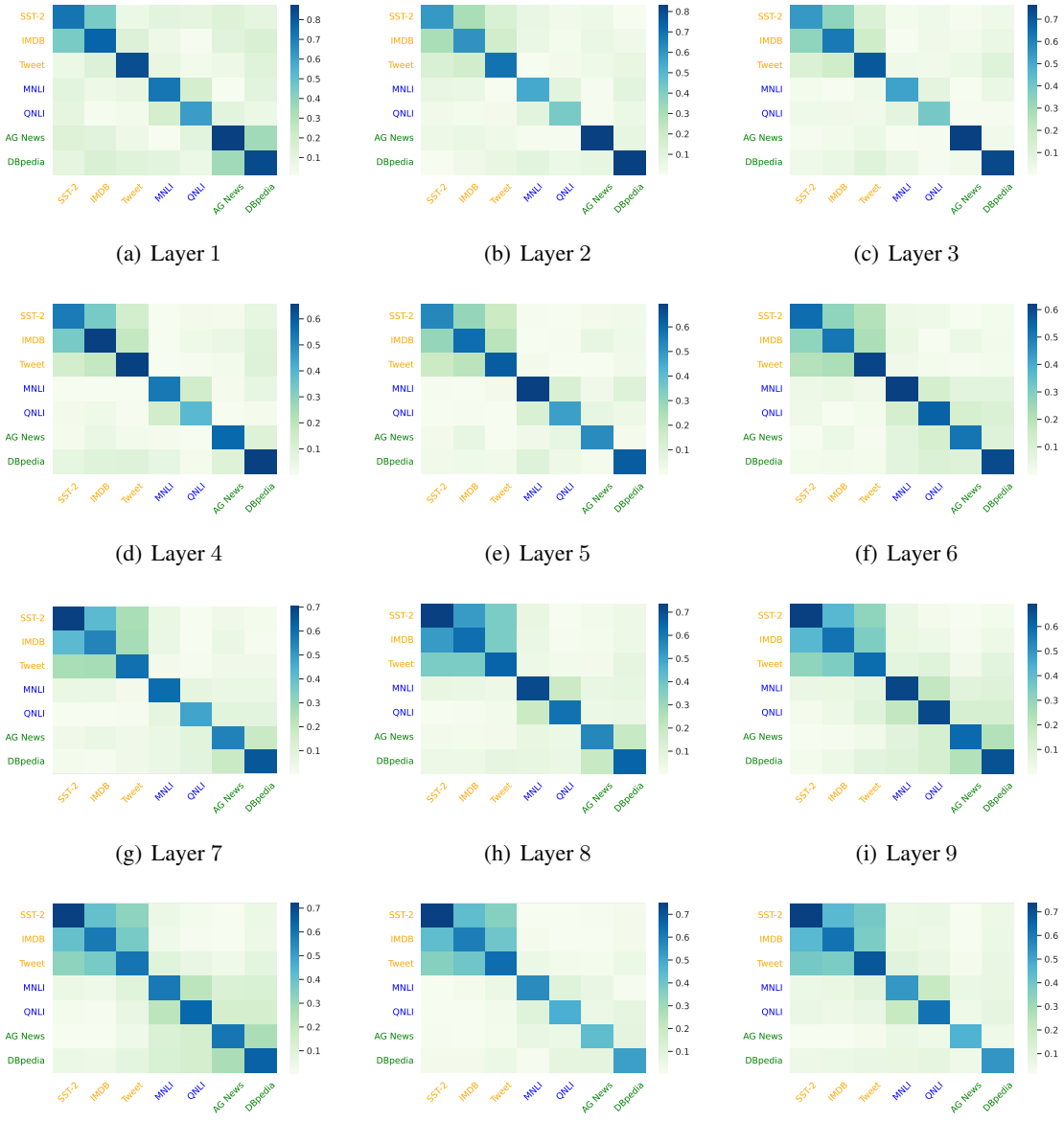
(c) On MNLI

(d) On QNLI

(e) On AG News

(f) On DBpedia

Figure 13: BitFit accuracies on various tasks drop along with the neuron perturbation rates. Error bars indicate ± 1 s.e.m. over 5 random trials. The perturbations are conducted in predictivity orders obtained with prompt tuning.



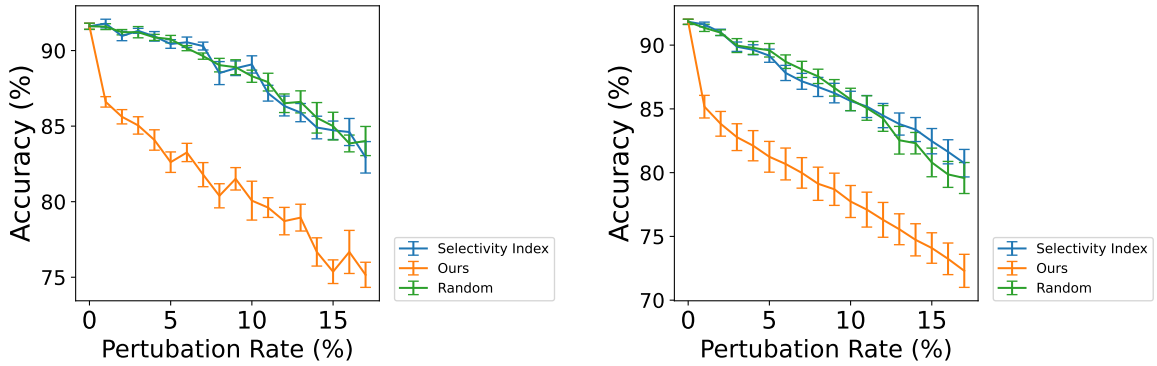
(a) Layer 1 (b) Layer 2 (c) Layer 3

(d) Layer 4 (e) Layer 5 (f) Layer 6

(g) Layer 7 (h) Layer 8 (i) Layer 9

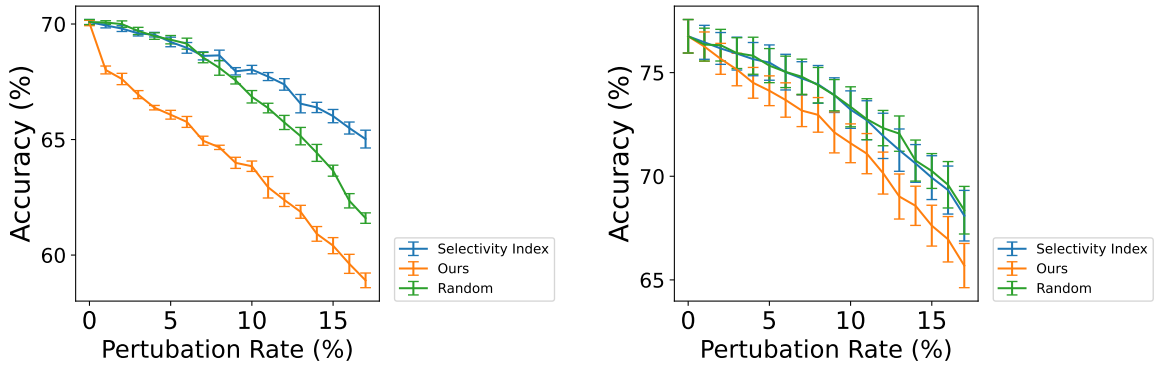
(j) Layer 10 (k) Layer 11 (l) Layer 12

Figure 14: Spearman's rank correlations between the neuron predictivity orders of different tasks on different layers. Layer 1 is the bottom layer near the inputs, and layer 12 is the top layer near the outputs.



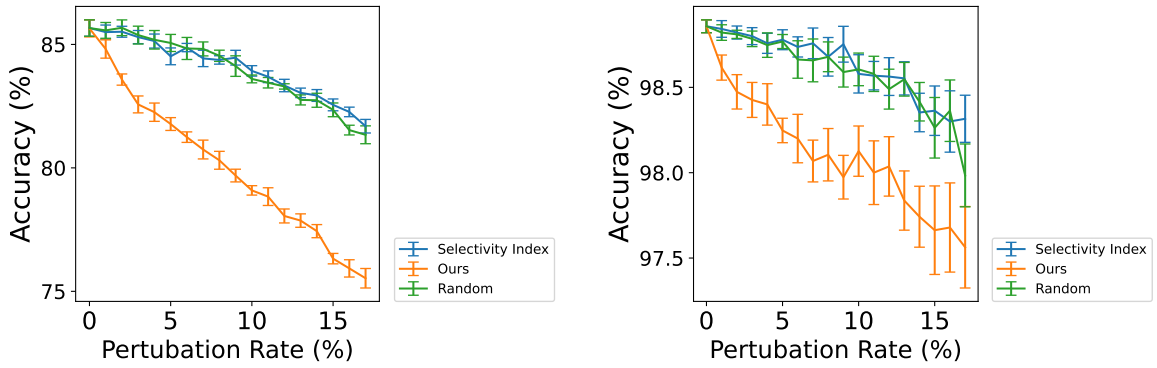
(a) On SST-2

(b) On IMDB



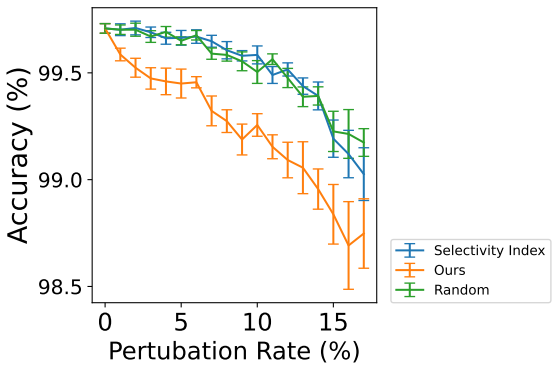
(c) On Tweet

(d) On MNLI



(e) On QNLI

(f) On AG News



(g) On DBpedia

Figure 15: Prompt tuning accuracies on various tasks drop along with the neuron perturbation rates. Error bars indicate ± 1 s.e.m. over 5 random trials. The perturbations are conducted in descending predictivity orders (*Ours*), random orders (*Random*) and descending "selectivity index" (Morcos et al., 2018) orders (*Selectivity Index*).