Let's Focus on Neuron:
Neuron-Level Supervised Fine-tuning for Large Language Model

Haoyun Xu[*♣♠] **Runzhe Zhan**[*♣] **Yingpeng Ma**[♣]
Derek F. Wong[†♣] **Lidia S. Chao**[♣]

[♣]NLP^{2}CT Lab, Department of Computer and Information Science, University of Macau
[♠]Tiger Research, Shanghai
nlp2ct.{haoyun, runzhe, yingpeng}@gmail.com, {derekfw, lidiasc}@um.edu.mo

Abstract

Large Language Models (LLMs) are composed of neurons that exhibit various behaviors and roles, which become increasingly diversified as models scale. Recent studies have revealed that not all neurons are active across different datasets, and this sparsity correlates positively with the task-specific ability, leading to advancements in model pruning and training efficiency. Traditional fine-tuning methods engage all parameters of LLMs, which is computationally expensive and may not be necessary. In contrast, Parameter-Efficient Fine-Tuning (PEFT) approaches aim to minimize the number of trainable parameters, yet they still operate at a relatively macro scale (e.g., layer-level). We introduce Neuron-Level Fine-Tuning (NeFT), a novel approach that refines the granularity of parameter training down to the individual neuron, enabling a more parameter-efficient fine-tuning model. The experimental results show that NeFT not only exceeded the performance of full-parameter fine-tuning and PEFT but also provided insights into the analysis of neurons. Our code and data are available at: https://github.com/NLP2CT/NeFT.

1 Introduction

Neurons, as fundamental components of Large Language Models (LLMs), fulfill diverse roles across model regions. As language models scale, the neurons display varying phenomena. Increase in model size enhances the the capability to generalize from text to basic and even unseen concepts (Patel and Pavlick, 2022). The internal concepts may be distributed across numerous neurons, and a significant proportion of neurons can become inactive, never triggering across diverse datasets (Durrani et al., 2020; Voita et al., 2023). This sparsity has been substantiated by recent studies utilizing it to prune

LLMs or enhance the efficiency of LLM inference, suggesting that not all neurons need to be active, a trait that becomes more pronounced in larger models (Ma et al., 2023; Frantar and Alistarh, 2023; Liu et al., 2023; Kurtic et al., 2023; Song et al., 2023).

LLMs commonly adapt to specific tasks through full-parameter supervised fine-tuning (SFT). Parameter-efficient fine-tuning (PEFT), which operates on a layer-level modular parameter selection basis (Houlsby et al., 2019; Li and Liang, 2021; Lester et al., 2021; Hu et al., 2022), seeks to reduce the trainable parameters during model training. Building on insights from model interpretability research, we propose that the granularity of parameter training can be refined to the neuron level. Consequently, we present a Neuron-Level Fine-Tuning (NeFT) approach, designed to improve model performance by selectively updating neurons identified as sensitive.

In an initial experiment, we identified sensitive neurons for the NLI task on Llama-2-7b-chat (Touvron et al., 2023) and used a trained probe to select a subset of neurons. The preliminary experiment suggest that training on these neurons outperform the full-parameter fine-tuning. To identify sensitive neurons for more complex tasks like translation and summarization, we devised a method that evaluates neuron similarity pre- and post-SFT, treating those with low similarity as sensitive. The NeFT surpasses the full-parameter fine-tuning model in task performance. To understand why it works, we further categorized neurons into three types, namely *strongly affected*, *suppressed*, and *indirectly affected* neurons, and introduced "rank difference" to assess neuron utilization. Our findings indicate that: 1) Neurons exhibit varying degrees of sensitivity during the SFT process. 2) Neurons strongly affected by SFT elicit significant alterations in parameter utilization patterns. 3) Neurons that are important for one task tend to be relevant for others. This consistency implies that neurons identified in

---

one context may be beneficial for transfer learning in similar datasets.

2 Background

Parameter Efficient Fine-tuning PEFT technique aims to enhance the performance of pre-trained models on tasks while minimizing the number of trainable parameters and computational complexity. This approach is particularly beneficial for reducing the training costs associated with large pre-trained models. PEFT can be a empirical choice of specific layers or modules, and some studies have shown that training only a single layer can sometimes outperform full-parameter fine-tuning. For instance, Yuan et al. (2023b) found that for translation tasks, embedding fine-tuning is effective than full-parameter fine-tuning except low-resource settings. However, the empirical choice and layer-wise searching is time-consuming while recent advances focus on leveraging external module to update all the layer-level parameters.

Adapter The goal of adapter (Houlsby et al., 2019; Pfeiffer et al., 2021; Rücklé et al., 2021; Liu et al., 2022) is to insert a small number of parameters into the model, and then train only these parameters when fine-tuning a downstream task, leaving the original parameters of the pre-trained model unchanged. This makes the trainable parameters more efficient and ensures that the original knowledge is not forgotten.

LoRA Low-Rank Adaptation (LoRA; Hu et al. 2022) has emerged as one of the most prevalent methods in both academic research and industry applications. LoRA's principal concept involves decomposing a large weight matrix into two low-rank matrices, significantly reducing the number of trainable parameters. Although the two low-rank matrices introduced by LoRA add to the model's architecture, they do not introduce additional computational costs during inference as they function concurrently with the original structures. For the sake of practical implementation, LoRA generally applied to affecting the computation of linear or multi-head attention mechanisms.

Sparse Fine-Tuning Sparse fine-tuning distinguishes itself from methods that add external modules like adapters or LoRA by introducing an initial step to pinpoint critical parameters. This process leverages various metrics, such as Fisher information (Sung et al., 2021) or L_0 regularization (Guo

et al., 2021a), to determine which parameters are essential (Chen et al., 2020; Prasanna et al., 2020). These identified parameters are then specifically targeted in the subsequent training phase. To date, sparse training (Guo et al., 2021b) techniques have been preliminarily investigated within the context of pre-trained LMs (Guo et al., 2021b; Dao et al., 2022), LLMs (Ansell et al., 2024; Thangarasa et al., 2023). However, they have not been examined from a neuron-level perspective.

3 Preliminary Experiment

Previous interpretability analyses within feedforward networks (FFNs) have revealed notable phenomena at the neuron level. Research by Gurnee and Tegmark (2023) has demonstrated that certain neurons exhibit sensitivity to specific entities and can encapsulate world knowledge. Motivated by this insight, we begin our investigation to select certain neurons using the similar probing approach and examine their effectiveness during SFT.

3.1 Select Neurons

Given that NLI classification task is well-suited for probing, we choose it as our testbed of preliminary experiment to discover neurons related to the NLI task. We first processed the XNLI dataset[1] through the Llama-2-7b-chat model (Touvron et al., 2023) to obtain the hidden states for each layer. For each sentence in the dataset, we averaged the hidden states across all tokens, resulting in a set of hidden states for each layer. We paired each hidden state data with the corresponding target labels, i.e., Entailment, Neutral, and Contradiction, to train a Ridge classifier as a probe. Next, we use the trained probe to identify the neurons most sensitive to XNLI tasks. Specifically, we selected 100,000 neurons as *sensitive neurons* according to the cosine similarity between the neurons and the hidden states of probe decision space.

3.2 Experimental Results

Upon isolating the sensitive neurons, we implement a gradient masking technique on the non-sensitive neurons throughout SFT process. This approach guarantees that updates are confined exclusively to the previously identified sensitive neurons. Notably, our NeFT strategy does not require the incorporation of additional architectural elements, resulting that a mere 6% of the total parameters

---

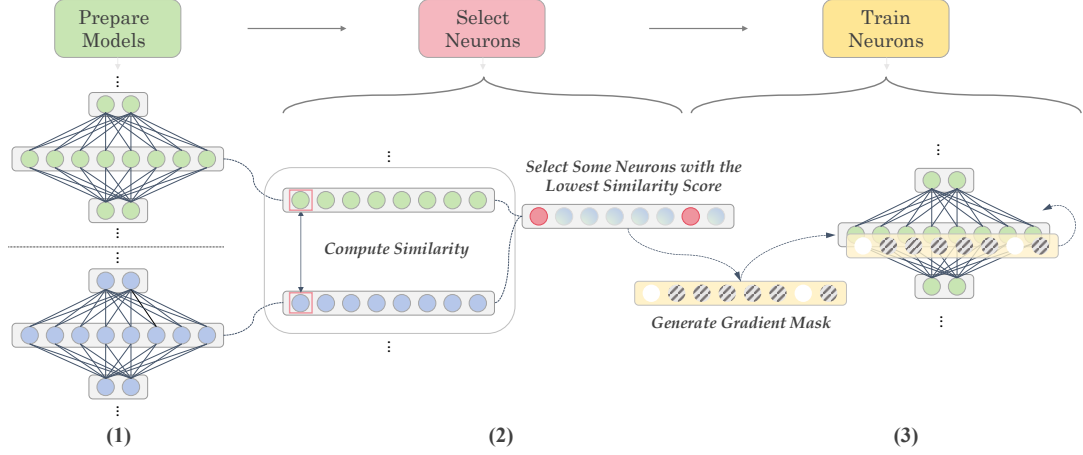[1] https://github.com/facebookresearch/XNLI

Figure 1: This diagram shows the whole process of our proposed Neuron-Level Fine-Tuning method. (1) Prepare two models, one is the original model (M_{Org}) and the other is the model (M_{FT}) trained with full-parameter fine-tuning. (2) Calculate the cosine similarity for each pair of neurons in the corresponding positions of M_{Org} and M_{FT} and select the $x\%$ neurons with the lowest score and refer to these neurons as sensitive neurons. (3) Mask the gradients of non-sensitive neurons during SFT training to ensure that only the selected neurons are updated.

	Accuracy (%)		
	English	**French**	**German**
FT-full	82.8	70.7	67.0
NeFT	**84.9**	**76.9**	**80.2**

Table 1: Preliminary experiment of NeFT effectiveness was conducted on XNLI tasks. NeFT demonstrated superior performance compared to full-parameter training.

were modified during training. All remaining training parameters are maintained in alignment with the full-parameter fine-tuning.

The results presented in Table 1 prove that NeFT is possible to outperform full-parameter fine-tuning by training only on a small number of sensitive neurons. However, this experiment on the NLI task may not fully exemplify the generality of NeFT. In the subsequent sections, we will introduce a methodology designed to identify and train sensitive neurons in more intricate contexts, thereby extending the applicability of NeFT across a broader spectrum of tasks.

4 Methodology

Current probing techniques fall short in addressing complex tasks. To remedy this, we propose a flexible approach as shown in Figure 1 and Algorithm 1. to facilitate the identification of sensitive neurons that are most influential to model performance improvements.

Algorithm 1 Neuron-Level Fine-Tuning (NeFT)

Require: SFT Dataset $\mathcal{D}$, model M_{Org}.
1: Train full parameters of M_{Org} on dataset $\mathcal{D}$ within limited steps K and obtain M_{FT}.
2: Calculate similarity scores $\{S^{(i,n)}\}$ by Eq.1.
3: Select $x\%$ of the neurons $\{w^{(i,n)}\} \leftarrow \arg\min_{S^{(i,n)}} \text{Select}(\mathbf{W}_{Org}, \{i\}, \{n\})$.
4: **for** $1, \ldots, \text{Epochs}$ **do**
5:     **for** $1, \ldots, \text{Batches}$ **do**
6:         Calculate gradient $\mathbf{g}$ of each batch.
7:         $\mathbf{g}(\{\mathbf{W}_{Org} \setminus \{w^{(i,n)}\}\}) \leftarrow 0$
8:         Back propagate the gradient $\mathbf{g}$.
9:     **end for**
10: **end for**

4.1 Prepare Models

In fact, training a model entails the discernment and engagement of sensitive neurons. By evaluating neurons that deviate most from their initial state, we can ascertain which ones the model has prioritized during its training, thereby revealing their sensitivity to the task at hand. Building on this premise, we begin by preparing an initial model alongside its fine-tuned counterpart for the designated task. We employ the SFT dataset $\mathcal{D}$ to fine-tune initial model M_{Org} with all parameters, resulting in the fine-tuned model M_{FT}. In practice, for the sake of efficiency, the fine-tuned model also can be derived within a limited number of training steps. Our

analysis in subsequent sections suggests that this abbreviated training regimen yields a model whose performance is not significantly different from one that has been trained to convergence.

4.2 Select Neurons by Model Itself

We define an individual neuron as a distinct entry within the weight matrix. For example, in a linear layer of an MLP with a weight matrix of dimension $\mathbb{R}^{m \times n}$, a single neuron corresponds to a row in this matrix, which is dimensionally represented as $\mathbb{R}^{1 \times n}$. We then calculate the cosine similarity for each corresponding pair of neurons between the original model and the fine-tuned model. A neuron at a given position (i, n) within the models is denoted by $\mathbf{W}_{\mathrm{Org}}^{(i,n)}$ and $\mathbf{W}_{\mathrm{FT}}^{(i,n)}$, where i represents the layer index and n indicates the n-th neuron within that layer. Following the formula in Equation 1, we calculate the cosine similarity score $\{S^{(i,n)}\}$:

$$S(w_{\mathrm{Org}}, w_{\mathrm{FT}}) = \frac{w_{\mathrm{Org}} \cdot w_{\mathrm{FT}}}{||w_{\mathrm{Org}}|| \times ||w_{\mathrm{FT}}||} \quad (1)$$

where $w_{\mathrm{Org}} \in \mathbf{W}_{\mathrm{Org}}^{(i,n)}$, $w_{\mathrm{FT}} \in \mathbf{W}_{\mathrm{FT}}^{(i,n)}$. The k neurons with the lowest cosine similarity scores are then selected for further training process.

4.3 Neuron-Level Fine-Tuning

After calculating the similarity scores for each neuron pair, we rank the neurons according to their similarity scores to pinpoint the ones to which the model allocates the most attention during training, characterized by their lower similarity scores. We record the location of these neurons, represented by the positional information (i, n). To train solely the identified neurons, we modify the gradient $\mathbf{g}(\mathbf{W}_{\mathrm{Org}})$. Prior to each update during the training process, we refer to the previously saved positional information (i, n) to decide if the gradient of a specific neuron should be retained or set to zero. This gradient masking policy ensures that only the specific neurons are updated.

5 Experiments

We conducted our main experiments using the Llama-2-7b-chat model, focusing on machine translation and cross-lingual text summarization tasks. Additionally, we investigated the generalizability of NeFT by evaluating its performance across different model architectures and examining its effectiveness on general-purpose instruction-following datasets.

5.1 Experimental Settings

Data For the machine translation task, we sourced our training and development datasets from the News Commentary[2] (Tiedemann, 2012) and Lego-MT (Yuan et al., 2023a). The test set was collated from Flores-101 (Goyal et al., 2022) and Lego-MT, ensuring a diverse range of linguistic challenges. Specifically, we utilized two distinct data sources for our experiments: For the English-Maori (En→Mi) and English-Bosnian (En→Bs) language pairs, both training and test sets were sourced from Lego-MT. For all other language pairs, we sampled training data from the WMT19 News Commentary corpus and used Flores-101 for test sets. To facilitate efficient training, we randomly sampled a subset of the original training corpus. Regarding the cross-lingual text summarization task, our dataset was taken from CrossSum (Bhattacharjee et al., 2023). Due to computational resource limitations, we restricted our training to sentences comprising fewer than 1024 tokens. Resulting in a small number of training samples, especially on the language pairs **Fr→Zh** and **Hi→Zh**. For these datasets, this results in 0.1k sentences, making this a low-resource setting.

To evaluate the model's performance on general-purpose tasks, we employed the Alpaca dataset (Taori et al., 2023) for training. For evaluation, we utilized the IFEval benchmark (Zhou et al., 2023) and the validation split of MMLU (Hendrycks et al., 2021) as our test sets.

Training Setup For the machine translation task and cross-lingual text summarization task, we conducted our experiments on the Llama-2-7b-chat model. We fine-tuned all models with a batch size of 6 for translation tasks and a batch size of 3 for summarization tasks until they reached convergence. The optimal checkpoints were determined based on the lowest validation loss observed on the development datasets. Typically, the checkpoint corresponding to the lowest evaluation loss emerged from the first epoch. However, in the summarization task, particularly when dealing with smaller datasets, this convergence point might occur during the second epoch. For training with LoRA, convergence was typically attained within 2 to 3 epochs. In the case of NeFT, when a certain percentage $x\%$ of the model parameters were trained, we denoted this as $\text{NeFT}_{x\%}$. Similarly, for

---

[2] https://huggingface.co/datasets/wmt/wmt19

Method	%Para.	En→Zh		En→Fr		Fr→Zh	
		20k	**100k**	**20k**	**100k**	**20k**	**100k**
Llama2-7B-Chat	-	41.28/3.15		55.44/6.96		46.38/1.85	
FT-full	100	74.46/22.20	78.31/26.42	79.18/25.98	81.72/30.27	72.07/18.19	76.30/22.22
FT-embed	2	78.64/23.49	79.78/25.38	82.80/29.03	83.80/30.15	78.22/22.77	77.89/20.80
FT-{in | out}	41	77.58/24.34	79.61/27.35	81.97/28.13	83.87/32.96	76.31/20.89	78.22/22.77
LoRA$_{r=256}$	9	80.95/27.13	79.40/26.22	83.38/32.25	83.10/32.78	79.91/22.84	78.34/22.24
NeFT$_{9\%}$	9	**82.27/28.69**	**82.91/30.48**	**84.75/34.14**	**85.53/35.98**	**80.10/24.07**	**80.87/25.61**

Method	%Para.	Hi→Zh	Hi→Fr	En→Mi	En→Bs
		4.5k	**4.5k**	**10k**	**10k**
Llama2-7B-Chat	-	43.15/0.35	48.43/0.75	39.31/0.28	38.68/0.38
FT-full	100	53.93/6.21	57.90/6.09	63.26/13.63	64.21/6.95
FT-embed	2	58.73/6.86	60.94/6.15	49.65/2.21	58.18/4.05
FT-{in | out}	41	60.54/9.03	63.09/7.17	**64.07/14.37**	67.38/7.48
LoRA$_{r=256}$	9	60.47/8.64	61.12/5.36	52.31/4.33	66.85/7.79
NeFT$_{9\%}$	9	**66.23/10.69**	**68.43/9.67**	62.73/13.08	**71.63/9.58**

Table 2: COMET/BLEU score comparison of each method on the translation task. For LoRA and NeFT, we report the performance when tuning 9% of model parameters. In the majority of cases, NeFT outperforms the baseline methods. More detailed results are available in Appendix Table 7.

Method	%Para.	En→Zh			Fr→Zh			Hi→Zh		
		2.3k			**0.1k**			**0.1k**		
		R1	R2	RL	R1	R2	RL	R1	R2	RL
Llama2-7B-Chat	-	2.80	1.79	2.43	5.17	2.81	4.36	1.13	0.78	1.09
FT-full	100	21.88	17.39	20.71	30.24	19.79	25.36	10.16	4.35	9.42
FT-embed	2	1.35	0.67	1.09	10.48	6.39	8.32	0.73	0.19	0.67
FT-{in | out}	41	23.13	18.16	22.07	32.45	26.59	29.24	15.51	8.21	13.60
LoRA$_{r=256}$	9	19.94	15.73	18.74	4.48	2.72	4.48	2.46	0.45	2.16
NeFT$_{9\%}$	9	24.31	19.27	23.29	**33.57**	**26.67**	**31.46**	15.23	**9.70**	14.21
NeFT$_{12\%}$	12	23.38	18.71	22.54	29.96	22.26	25.33	13.56	8.12	12.58
NeFT$_{6\%\text{Union}}$	12	**26.05**	**21.06**	**25.07**	30.85	22.86	26.60	**15.78**	9.32	**14.36**

Table 3: Performance comparison of each method on cross-lingual summarization task. For comparing LoRA with NeFT, we report the performance when tuning 9% of model parameters. The results indicate that NeFT remains effective even in low-resource settings.

LoRA configurations, if the LoRA rank was set to r, we represented this as LoRA$_r$.

To evaluate model performance beyond translation tasks, we conducted additional experiments with Llama-3.2-1B (Dubey et al., 2024) and Mistral-7B-v0.1 (Jiang et al., 2023) on instruction-following tasks. All models were fine-tuned for one epoch using a batch size of 8.

Baselines Our method was evaluated against four baselines:1) **FT-full**: This strategy fine-tunes all parameters of the LLM. 2) **FT-{in | out}**: This approach fine-tunes the weights of input and output projection layers of all the MLPs. 3) **FT-embed** (Yuan et al., 2023b): This technique focuses solely on fine-tuning the embedding layer. 4) **LoRA** (Hu et al., 2022): We apply it to fine-tune all linear structures within each layer (excluding the head layer of language model).

To maintain the integrity of the comparison, we adjusted the number of trainable parameters of NeFT to closely align with those employed in the LoRA configurations.

Evaluation We employ the COMET (Rei et al., 2020) and ordinary BLEU score (Papineni et al., 2002) to assess the quality of machine translations. COMET is a neural metric designed for the multilingual evaluation of machine translation systems. Higher COMET or BLEU scores indicate better translation quality. In the context of text summarization, we utilize the ROUGE metric (Lin, 2004) to evaluate the extent to which the machine-generated summary encapsulates the core points of

the reference summary. ROUGE are represented as "R(N)" in the reported results, where N corresponds to the length of the N-gram. For IFEval and MMLU benchmarks, we followed the official evaluation settings and metrics.

5.2 Main Results

Performance and Generalization   Table 2 and 3 present the comparative performance of different PEFT methods. Our results demonstrate that NeFT consistently outperforms other approaches across various tasks and language pairs. For the three language pairs where we compare two different data scales (En→Zh, En→Fr, Fr→Zh), we maintain the same subset of fine-tuned neurons within each pair, demonstrating NeFT's robustness across different training data sizes.

In addition to its superior performance on targeted language pairs, we also find that NeFT have a potential to improve cross-lingual generalization. For summarization task, we merge an equal number of neurons identified from the translation task of a specific language pair with those from the summarization task. For instance, neurons from the $\text{NeFT}_{9\%}$ configuration in the summarization task were combined with those from the $\text{NeFT}_{9\%}$ setting in the translation task. This amalgamated configuration is referred to as $\text{NeFT}_{6\%\text{Union}}$. Our observations indicate that the potential for further improving cross-lingual summarization with neurons identified in translation task is more pronounced when dealing with high-resource language pairs like English-to-Chinese translation. Additionally, we provide a comprehensive comparison in respect to LoRA across language pairs in Appendix A.

For instruction-following tasks, both models were fine-tuned on the Alpaca dataset for one epoch and evaluated on the IFEval benchmark and MMLU validation set. As shown in Table 4, NeFT consistently outperforms full fine-tuning across both model architectures, demonstrating its potential applicability beyond cross-lingual tasks.

6 Analysis

6.1 Utilization of Neurons

Metric   To explore the divergence of neuron utilization, we compared models trained under various NeFT configurations by analyzing neuron utilization. Specifically, during inference process, for each neuron, we computed the Pearson correlation between its weight vector and the hidden state rep-

Model / Method	IFEval (Strict / Loose)	MMLU Validaition
Llama-3.2-1B		
FT-full	0.104 / 0.111	0.253
$\text{NeFT}_{1.2\%}$	**0.131 / 0.144**	**0.254**
Mistral-7B-v0.1		
FT-full	0.089 / 0.100	0.247
$\text{NeFT}_{9\%}$	**0.262 / 0.274**	**0.466**

Table 4: Performance comparison between NeFT and full fine-tuning on general-purpose tasks across different model scales. Evaluation metrics include instruction following capability (using strict and loose matching criteria) and MMLU accuracy. Results show NeFT consistently outperforms full fine-tuning while using only a small fraction of trainable neurons.

resentations from all preceding layers. The neurons were then ranked in descending order based on their maximum correlation scores, yielding a neuron importance ranking denoted as **Rank**. To discern the contrast in neuron utilization between models, we compute the rank disparity for each neuron across the two models, represented as $\Delta\textbf{Rank}$. We then determined the mean of the absolute values of these rank differences, $\text{Avg}(\Delta\textbf{Rank})$, to quantify the overall divergence in utilization of the neurons.
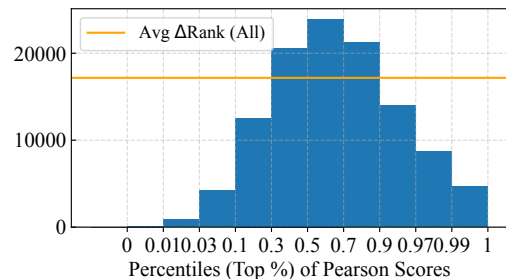

Figure 2: Average rank differences between $\text{NeFT}_{6\%}$ and $\text{NeFT}_{3\%}$ were calculated for neurons. The ranks were sorted based on pairwise Pearson scores in descending order.

Shifts of Neuron Utilization   To analyze how utilization of neurons shift after fine-tuning, we categorize the neurons into different buckets according their Pearson correlation scores and calculate the average rank differences of each bucket between $\text{NeFT}_{6}\%$ and $\text{NeFT}_{3}\%$, denoted as $\text{Avg}(\Delta\textbf{Rank})$. The bucketing strategy relies on the top percentiles of the overall Pearson scores. Figure 2 demonstrates that the shift in utilization for neurons within the top 0% to 3% range is negligible, whereas neurons with intermediate correlation scores ex-
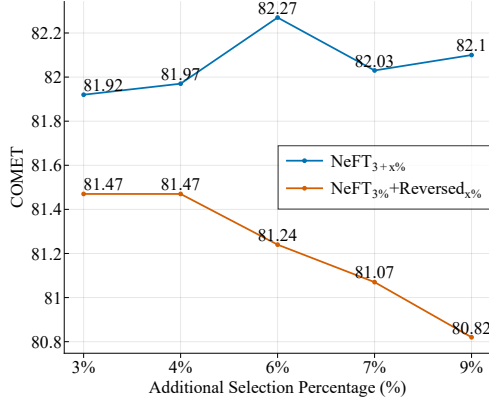
Figure 3: COMET scores of models trained with different NeFT settings when NeFT$_{3\%}$ serve as a foundation. Neurons with different similarity scores were separately incorporated and trained using 20k English-Chinese translation data.

hibit the most significant sensitivity to fine-tuning. These findings suggest that our selection strategy of sensitive neurons is reasonable. Furthermore, we also found that this observation is consistent across various training settings for other NeFT models.

6.2 Effects of Neuron Selection Settings

Performance Comparison To investigate the impact of neuron selection based on similarity score on NeFT training, we incorporated neurons with both high and low cosine similarity scores into the NeFT$_{3\%}$ model and examined the respective performance trends. The study used a dataset comprising 20,000 English-to-Chinese translation training instances. Adding neurons with high similarity scores were labeled Reversed$_{x\%}$ for clarity. Hence, the notation "NeFT$_{3\%}$+Reversed$_{x\%}$" represents a hybrid selection strategy by combining neurons from NeFT$_{3\%}$ and Reversed$_{x\%}$. According to the results presented in Figure 3, the original NeFT neuron selection strategy consistently outperforms the contrasting strategy. Moreover, it also shows a decline in model performance as the proportion of high-similarity neurons increases.

Dynamics of Neuron Utilization Figure 4 provides insights into how neuron utilization is influenced by different NeFT training configurations. In Figure 4 (a), the comparison with the NeFT$_{3\%}$ model which employs the original selection strategy, shows that neuron utilization discrepancies becomes more pronounced when the model is trained with high-similarity neurons. This observation implies that the contrast-
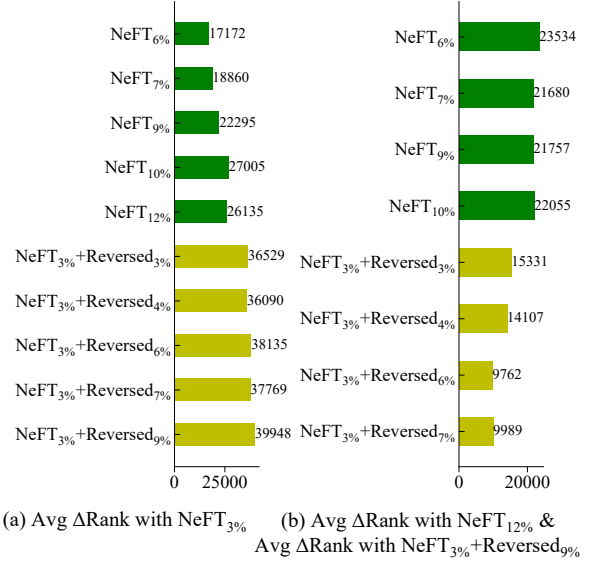

(a) Avg ΔRank with NeFT$_{3\%}$ (b) Avg ΔRank with NeFT$_{12\%}$ & Avg ΔRank with NeFT$_{3\%}$+Reversed$_{9\%}$

Figure 4: Rank difference Avg(Δ**Rank**) is calculated in order to assess the shifts in the utilization of neurons. Overall, the neuron utilization of original neuron selection strategy NeFT$_{x\%}$ is more stable than that of contrasting selection strategy NeFT$_{3\%}$+Reversed$_{x\%}$.

ing strategy, NeFT$_3$%+Reversed$_x$%, tends to induce greater volatility within the neuron utilization patterns. Figure 4(b) analyzes the differences in utilization between models with a larger proportion of trainable parameters and those with a smaller proportion, while employing the same fine-tuning strategy. This comparison is made between both the original selection strategies (e.g., NeFT$_{12}$% vs. NeFT$_x$%) and the contrasting selection strategies (e.g., NeFT$_3$%+Reversed$_9$% vs. NeFT$_3$%+Reversed$_x$%). From the observed data, different from Figure 4 (a), it is apparent that models employing the contrasting neuron selection strategy are less affected by an increase in the number of trainable parameters. This suggests that the original neuron selection strategy has a higher degree of stability with respect to changes in the scale of trainable parameters.

Category of Neurons Based on the observed ranking shifts, we further classify neurons into one of three different categories: "Strongly Affected Neurons", "Suppressed Neurons", and "Indirectly Affected Neurons". The neurons exhibiting a rank difference Δ**Rank** exceeding 100,000 are categorized as *Strongly Affected Neurons*, indicating significant influence from NeFT training. Within the subset of strongly affected neurons, not all exhibited upward movement in their rankings. Conse-

(a) $\text{NeFT}_{x\%}$ ΔRank with $\text{NeFT}_{3\%}$ and $\text{NeFT}_{12\%}$

(b) $\text{NeFT}_{3\%}+\text{Reversed}_{x\%}$ ΔRank with $\text{NeFT}_{3\%}$

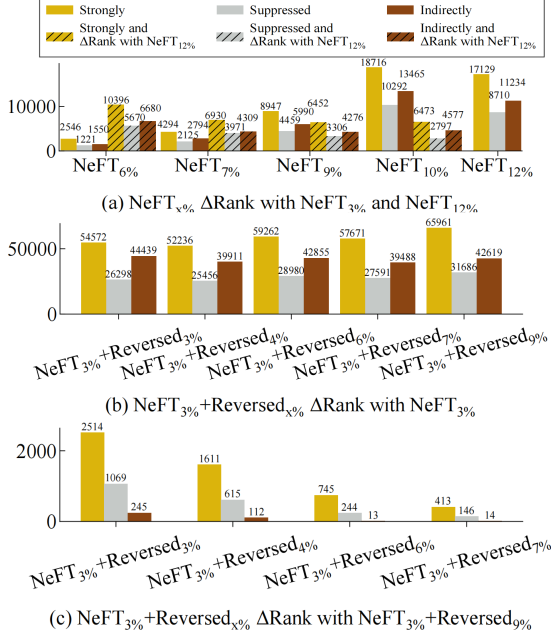(c) $\text{NeFT}_{3\%}+\text{Reversed}_{x\%}$ ΔRank with $\text{NeFT}_{3\%}+\text{Reversed}_{9\%}$

Figure 5: The distribution of three types of neurons (Strongly affected, Suppressed, and Indirectly affected) across models under various training settings.

quently, we classify neurons that decreased in rank as *Suppressed Neurons*. The training of a specific subset of neurons inherently impacts the remaining, untrained neurons during the inference process. Within the subset of strongly affected neurons, we observe that a substantial number of strongly affected neurons were not directly engaged in NeFT training. These neurons are thus designated as *Indirectly Affected Neurons*.

Categorized Comparison Additionally, we assessed the quantity of different neuron types within the original selection strategy $\text{NeFT}_{x\%}$ and the contrasting strategy $\text{NeFT}_{3\%}+\text{Reversed}_{x\%}$. The categorization includes strongly affected neurons, suppressed neurons, and indirectly affected neurons, as illustrated in Figure 5. The trend indicates a higher count of strongly affected neurons in the contrasting strategy compared to the original selection strategy (Figure 5 (a) and Figure 5 (b)). In the case of suppressed neurons, there is a noticeable trend: as the number of strongly affected neurons rises, the quantity of suppressed neurons increases correspondingly, maintaining a consistent proportion. In Figure 5 (c), the number of strongly affacted neuron found between two $\text{NeFT}_{3\%}+\text{Reversed}_{x\%}$ settings is much lower than that between $\text{NeFT}_{3\%}+\text{Reversed}_{x\%}$ and $\text{NeFT}_{3\%}$. Also, the proportion of indirectly affected neurons is very low which different from Figure 5 (a) and

Original (20k)		Separated (20k)	
FT-full	74.46/22.20	**FT-full**	73.30/22.11
NeFT$_{9\%}$	**82.27/28.69**	**NeFT**$_{6\%}$	82.00/28.45
NeFT$_{9\%\text{Early}}$	81.91/28.50	**NeFT**$_{6\%\text{Trans}}$	**82.02/28.57**

Table 5: COMET/BLEU scores of NeFT-tuned English-Chinese translation model under different settings.

Figure 5 (b).

Based on the above observations discussed in this section, we infer that the inclusion of neurons with high similarity score causes drastic changes in the model's neuronal utilization patterns. However, as the number of neurons increases, the extent of change becomes relatively small. Additionally, the trend observed with the inclusion of more neurons may imply that only a minimal number of *strongly affected neurons* are subject to indirect influence, leading to minimal shifts in the neurons' utilization.

6.3 Ablation Study

Neuron Selection We also focus on assessing the impact of neurons selected by the model prior to its convergence. We selected neurons from a model that had been trained on the English-to-Chinese (En-Zh) translation dataset for only 800 steps, and these neurons are denoted as $\text{NeFT}_{x\%\text{Early}}$ and would be used to select sensitive neurons. As shown in Table 5, the neurons identified by the unconverged model also can delivers satisfactory results, indicating that effective neuron selection does not necessarily require a fully converged model.

Neuron Generalization To evaluate whether the identified critical neurons generalize across different data distributions within the same domain, we conducted experiments with two distinct English-to-Chinese (En-Zh) translation datasets. First, we identified the critical neurons using our primary training dataset. Then, we evaluated the transferability of these neurons by applying $\text{NeFT}_{x\%\text{Trans}}$ to an independent, non-overlapping dataset (denoted as 'Separated' in the tables) from the same language pair. The results in Table 5 indicate that neurons, once identified by the model, can be effectively employed across similar datasets, maintaining consistent performance. This suggest the potential for neurons identified in one domain to be extrapolated to other domains.

Overlap Analysis We conducted further analysis on the overlap of neurons identified under different settings as per the ablation experiments previously

Neurons Overlap	$NeFT_{x\%}$			
	3%	6%	9%	12%
Original (20k)	Separated (20k)	63%	68%	72%	76%
Original (20k)	Original (100k)	61%	65%	68%	71%
Conv (20k)	Non-conv (20k)	63%	68%	72%	75%

Table 6: Overlap proportion of neurons across different data and training settings.

described. The data presented in Table 6 reveal that while the specific neurons selected across different settings are not identical, there is a considerable degree of overlap. Furthermore, the extent of this overlap escalates with an increase in the number of neurons designated for selection. Despite some variability in the specific neurons identified, the overall impact on the performance of models remains largely consistent, as evidenced by the previous results. Table 6 compares the neuron overlap between models trained to full convergence (Conv) and those trained with limited optimization steps (Non-conv).

Taking into account the insights from the preceding sections, we can deduce that the crucial factor regarding neuron selection is not the inclusion of a substantial quantity of non-sensitive $Reversed_x\%$ neurons. Regarding $NeFT_x\%$ neurons, the selection criteria do not require excessive precision, approximate selections are found to be adequate.

Convergence Speed Figures 6 and 7 illustrate the training dynamics of NeFT compared to baseline methods on the English-to-Chinese translation task (20k examples). The convergence curves demonstrate that NeFT consistently achieves lower validation loss across different experimental settings. Notably, when a sufficient number of sensitive neurons is selected, NeFT exhibits significantly faster convergence rates compared to traditional fine-tuning approaches. This accelerated optimization suggests that focusing on sensitive neurons effectively reduces the search space while maintaining model performance.

7 Conclusion

In this study, we introduced Neuron-level Supervised Fine-tuning, NeFT, an innovative approach to supervised fine-tuning. NeFT focuses on identifying task-specific neurons within a model by evaluating the similarity between the neurons of the trained and the original model. By selecting and training on neurons that exhibit low similarity, Our empirical results demonstrate that NeFT gener-
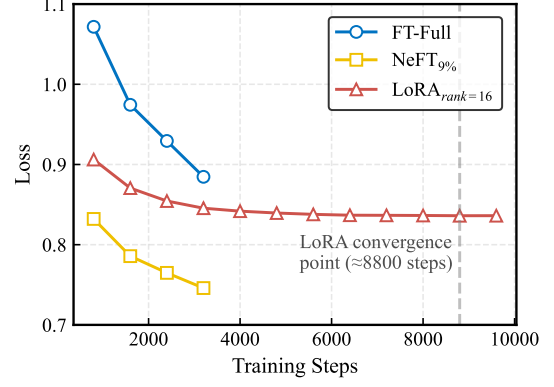

Figure 6: Comparison of loss curves during training among full fine-tuning (FT-full), NeFT with 9% critical neurons ($NeFT_{9\%}$), and LoRA with rank 16 ($LoRA_{r=16}$).
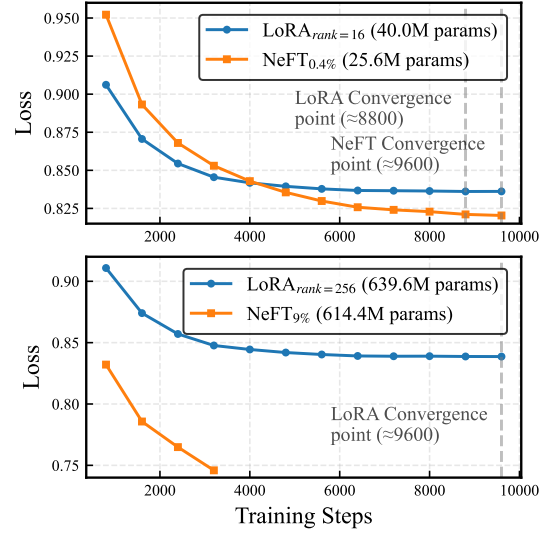

Figure 7: Validation loss comparison between parameter-matched pairs: NeFT achieves better performance with fewer parameters in both settings ($NeFT_{0.4\%}$ vs. $LoRA_{r=16}$ and $NeFT_{9\%}$ vs. $LoRA_{r=256}$, where LoRA uses slightly more parameters).

ally surpasses full-parameter fine-tuning and other fine-tuning methodologies across various settings. When compared with LoRA, NeFT consistently shows superior or comparable results at most levels of parameter counts. Furthermore, our analyses shed light on the functioning of distinct neurons in the NeFT process and the impact of various settings. This understanding contributes to a more nuanced view of the fine-tuning process and opens avenues for further optimization and refinement of neuron-level training strategies. In the future, we will keep applying NeFT method to more generation and reasoning tasks.

Limitation

First of all, this work is limited by experimenting on task-specific benchmarks using Llama-2-7B model. Moreover, this work encounters certain constraints due to the capabilities of current distributed frameworks, particularly regarding gradient operations. Specifically, we are unable to apply gradient operations universally across all structural components of the model. Consequently, our experiments are confined to the **up_proj** and **down_proj** projections within the Llama-2-7b-chat architecture, and we have not extended our investigation to include the **gate_proj** projections. Despite these restrictions, the NeFT methodology has demonstrated impressive results. For future analyses, there are existing neuron-pruning methods that treat neurons as discrete units, as referenced by (Fang et al., 2023; Zhuang et al., 2020; Han et al., 2015; Sun et al., 2023; Ma et al., 2023). These methods could potentially be leveraged in model pruning to eliminate certain neurons. This would enable a more concentrated examination of the neurons that contribute the most value to the model's performance, thereby optimizing the fine-tuning process by focusing on the most impactful neurons.

Acknowledgements

References

Alan Ansell, Ivan Vulic, Hannah Sterz, Anna Korhonen, and Edoardo M. Ponti. 2024. Scaling sparse fine-tuning to large language models. *CoRR*, abs/2401.16405.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Yuan-Fang Li, Yong-Bin Kang, and Rifat Shahriyar. 2023. Crosssum: Beyond english-centric cross-lingual summarization for 1, 500+ language pairs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2541–2564. Association for Computational Linguistics.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pre-trained bert networks. *Advances in neural information processing systems*, 33:15834–15846.

Tri Dao, Beidi Chen, Nimit S Sohoni, Arjun Desai, Michael Poli, Jessica Grogan, Alexander Liu, Aniruddh Rao, Atri Rudra, and Christopher Ré. 2022. Monarch: Expressive structured matrices for efficient and accurate training. In *International Conference on Machine Learning*, pages 4690–4721. PMLR.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neu-

rons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4865–4880. Association for Computational Linguistics.

Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. 2023. Depgraph: Towards any structural pruning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 16091–16101. IEEE.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Trans. Assoc. Comput. Linguistics*, 10:522–538.

Demi Guo, Alexander M. Rush, and Yoon Kim. 2021a. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4884–4896. Association for Computational Linguistics.

Demi Guo, Alexander M Rush, and Yoon Kim. 2021b. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896.

Wes Gurnee and Max Tegmark. 2023. Language models represent space and time. *CoRR*, abs/2310.02207.

Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. Learning both weights and connections for efficient neural networks. *CoRR*, abs/1506.02626.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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 the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Eldar Kurtic, Denis Kuznedelev, Elias Frantar, Michael Goin, and Dan Alistarh. 2023. Sparse fine-tuning for inference acceleration of large language models. *CoRR*, abs/2310.06927.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Ré, and Beidi Chen. 2023. Deja vu: Contextual sparsity for efficient llms at inference time. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 22137–22176. PMLR.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *CoRR*, abs/2305.11627.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Roma Patel and Ellie Pavlick. 2022. Mapping language models to grounded conceptual spaces. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 487–503. Association for Computational Linguistics.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When bert plays the lottery, all tickets are winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3208–3229.

Ricardo Rei, Craig Stewart, Ana C. Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2685–2702. Association for Computational Linguistics.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. Adapterdrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7930–7946. Association for Computational Linguistics.

Yixin Song, Zeyu Mi, Haotong Xie, and Haibo Chen. 2023. Powerinfer: Fast large language model serving with a consumer-grade GPU. *CoRR*, abs/2312.12456.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2023. A simple and effective pruning approach for large language models. *CoRR*, abs/2306.11695.

Yi-Lin Sung, Varun Nair, and Colin Raffel. 2021. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 24193–24205.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Vithursan Thangarasa, Abhay Gupta, William Marshall, Tianda Li, Kevin Leong, Dennis DeCoste, Sean Lie, and Shreyas Saxena. 2023. Spdf: Sparse pre-training and dense fine-tuning for large language models. In *Uncertainty in Artificial Intelligence*, pages 2134–2146. PMLR.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 2214–2218. European Language Resources Association (ELRA).

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2023. Neurons in large language models: Dead, n-gram, positional. *CoRR*, abs/2309.04827.

Fei Yuan, Yinquan Lu, Wenhao Zhu, Lingpeng Kong, Lei Li, Yu Qiao, and Jingjing Xu. 2023a. Lego-mt: Learning detachable models for massively multilingual machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 11518–11533. Association for Computational Linguistics.

Fei Yuan, Shuai Yuan, Zhiyong Wu, and Lei Li. 2023b. How multilingual is multilingual llm? *CoRR*, abs/2311.09071.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *CoRR*, abs/2311.07911.

Tao Zhuang, Zhixuan Zhang, Yuheng Huang, Xiaoyi Zeng, Kai Shuang, and Xiang Li. 2020. Neuron-level structured pruning using polarization regularizer. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

A   Appendix

Compare with LoRA   We conducted a comprehensive comparison with LoRA as shown in Figure 8. We varied LoRA rank between 8 and 256 in order to maintain a comparable number of trainable parameters with NeFT neurons. Our findings reveal that NeFT outperforms LoRA in terms of translation quality, particularly in the X-to-Chinese translation direction. For detailed comparisons and scores, please refer to Appendix Table 7.

Table 7 presents the results of NeFT and LoRA under various training settings for translation of each language pair. It also includes a comparison between NeFT and LoRA at similar trainable parameter scales.
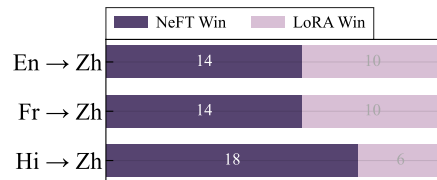
Figure 8: Comparison of NeFT and LoRA across different trainable parameter settings. NeFT consistently utilizes fewer parameters than LoRA at each level. The details are presented in Appendix Table 7. We highlight the best performing method for each language pair (shown in rows) using underlined values. The win statistics are computed vertically, representing the total number of times each method achieves the best performance across different language pairs.

# Lang En→Zh (20k) NeFT | Params	BLEU En→Zh	Fr→Zh	Hi→Zh	LoRA | Params	BLEU En→Zh	Fr→Zh	Hi→Zh
				Rank 8 | 20.0M	81.01/27.12	78.38/23.02	45.33/2.85
NeFT$_{0.4\%}$ | 25.6M	81.22/27.20	78.71/22.71	40.28/2.40	**Rank 16 | 40.0M**	81.24/27.27	78.70/23.03	**46.09/2.88**
NeFT$_{1.5\%}$ | 102.4M	81.86/28.31	79.32/23.29	42.79/2.17	**Rank 64 | 159.9M**	80.99/27.33	77.34/22.20	37.16/0.90
NeFT$_{3\%}$ | 204.8M	81.88/28.34	79.51/23.76	38.23/0.99	**Rank 128 | 319.8M**	81.06/27.25	78.30/22.80	33.19/0.20
NeFT$_{6\%}$ | 409.6M	81.92/28.67	78.99/23.75	37.18/0.80	**Rank 256 | 639.6M**	80.95/27.13	77.81/22.59	31.12/0.09
NeFT$_{9\%}$ | 614.4M	**82.27/28.69**	**79.84/24.01**	34.38/0.45				
NeFT$_{12\%}$ | 819.2M	82.10/28.39	79.75/23.79	36.44/1.30				

# Lang Fr→Zh (20k) NeFT | Params	BLEU En→Zh	Fr→Zh	Hi→Zh	LoRA | Params	BLEU En→Zh	Fr→Zh	Hi→Zh
				Rank 8 | 20.0M	79.12/24.17	79.71/22.77	38.29/1.18
NeFT$_{0.4\%}$ | 25.6M	80.79/24.47	79.63/22.99	**61.46/8.61**	**Rank 16 | 40.0M**	79.32/24.26	79.44/22.54	40.22/1.71
NeFT$_{1.5\%}$ | 102.4M	81.08/24.98	79.85/23.30	60.63/7.94	**Rank 64 | 159.9M**	79.72/24.56	79.59/22.55	35.42/0.62
NeFT$_{3\%}$ | 204.8M	80.66/24.94	79.82/23.66	60.63/8.09	**Rank 128 | 319.8M**	71.67/20.85	80.02/22.81	35.09/0.56
NeFT$_{6\%}$ | 409.6M	**80.96/25.30**	**80.13**/23.64	56.76/6.36	**Rank 256 | 639.6M**	79.59/24.48	79.91/22.84	35.31/0.56
NeFT$_{9\%}$ | 614.4M	80.32/24.72	80.10/**24.07**	57.98/6.96				
NeFT$_{12\%}$ | 819.2M	79.52/23.87	79.92/23.96	51.00/3.34				

# Lang Hi→Zh (4.5k) NeFT | Params	BLEU En→Zh	Fr→Zh	Hi→Zh	LoRA | Params	BLEU En→Zh	Fr→Zh	Hi→Zh
				Rank 8 | 20.0M	58.84/11.98	36.87/0.67	64.48/9.55
NeFT$_{0.4\%}$ | 25.6M	50.77/8.49	37.89/1.83	63.53/8.57	**Rank 16 | 40.0M**	**61.84/13.14**	39.71/1.48	64.71/9.70
NeFT$_{1.5\%}$ | 102.4M	38.69/2.01	35.45/0.90	64.80/9.45	**Rank 64 | 159.9M**	64.61/11.73	50.07/6.11	63.15/10.01
NeFT$_{3\%}$ | 204.8M	38.29/1.38	34.85/0.39	64.53/9.14	**Rank 128 | 319.8M**	64.39/11.89	53.35/7.33	61.90/9.18
NeFT$_{6\%}$ | 409.6M	42.94/2.30	51.11/5.18	65.63/9.68	**Rank 256 | 639.6M**	63.76/10.80	**55.46/8.10**	60.47/8.64
NeFT$_{9\%}$ | 614.4M	39.37/1.22	35.29/0.47	**66.23**/10.69				
NeFT$_{12\%}$ | 819.2M	31.51/0.04	31.16/0.04	65.94/**11.21**				

# Lang En→Fr (20k) NeFT | Params	BLEU en→Fr	Hi→Fr	LoRA | Params	BLEU En→Fr	Hi→Fr
			Rank 8 | 20.0M	84.07/32.40	40.72/1.65
			Rank 16 | 40.0M	84.11/33.03	41.83/1.80
			Rank 64 | 159.9M	83.92/32.49	32.97/0.25
NeFT$_{3\%}$ | 204.8M	84.70/33.51	48.74/3.51	**Rank 128 | 319.8M**	83.69/32.30	32.24/0.17
NeFT$_{6\%}$ | 409.6M	**84.83**/34.12	47.67/3.66	**Rank 256 | 639.6M**	83.38/32.45	30.38/0.06
NeFT$_{9\%}$ | 614.4M	84.75/34.14	47.81/3.34			
NeFT$_{12\%}$ | 819.2M	84.82/**34.33**	**51.73/3.84**			

# Lang Hi→Fr (4.5k) NeFT | Params	BLEU En→Fr	Hi→Fr	LoRA | Params	BLEU En→Fr	Hi→Fr
			Rank 8 | 20.0M	71.61/19.87	66.72/7.66
			Rank 16 | 40.0M	**76.66/20.51**	66.27/7.41
			Rank 64 | 159.9M	77.38/19.39	64.33/6.68
NeFT$_{3\%}$ | 204.8M	52.79/11.04	60.94/6.15	**Rank 128 | 319.8M**	77.16/18.13	62.31/5.74
NeFT$_{6\%}$ | 409.6M	62.18/12.07	67.14/8.16	**Rank 256 | 639.6M**	75.89/16.45	61.12/5.36
NeFT$_{9\%}$ | 614.4M	49.12/8.82	**68.43**/9.67			
NeFT$_{12\%}$ | 819.2M	30.55/0.25	68.37/**9.98**			

# Lang En→Mi (10k) NeFT | Params	BLEU En→Mi	LoRA | Params	BLEU En→Mi
		Rank 8 | 20.0M	57.27/5.92
		Rank 16 | 40.0M	56.52/5.58
		Rank 64 | 159.9M	55.54/5.34
		Rank 128 | 319.8M	53.90/4.69
NeFT$_{9\%}$ | 614.4M	62.73/13.08	**Rank 256 | 639.6M**	52.31/4.33
NeFT$_{12\%}$ | 819.2M	**63.26/13.30**		

# Lang En→Bs (10k) NeFT | Params	BLEU En→Bs	LoRA | Params	BLEU En→Bs
		Rank 8 | 20.0M	68.97/8.19
		Rank 16 | 40.0M	67.68/7.70
		Rank 64 | 159.9M	67.75/7.73
		Rank 128 | 319.8M	68.69/8.34
NeFT$_{9\%}$ | 614.4M	**71.63/9.58**	**Rank 256 | 639.6M**	66.85/7.79

Table 7: The COMET/BLEU scores for all LoRA settings and the comparison with NeFT of the corresponding parameter magnitude in the machine translation experiment.