LANDeRMT: Detecting and Routing Language-Aware Neurons for Selectively Finetuning LLMs to Machine Translation

Shaolin Zhu\textsuperscript{1}, Leiyu Pan\textsuperscript{1}, Bo Li\textsuperscript{2,3}, Deyi Xiong\textsuperscript{1*}

\textsuperscript{1}College of Intelligence and Computing, Tianjin University, Tianjin, China
\textsuperscript{2}School of Software, Tsinghua University, Beijing, China
\textsuperscript{3}Baidu APP Technology and Platform R&D Department, Baidu Inc, Beijing, China
{zhushaolin, lypan, dyxiong}@tju.edu.cn
{li-b19}@mails.tsinghua.edu.cn

Abstract

Recent advancements in large language models (LLMs) have shown promising results in multilingual translation even with limited bilingual supervision. The major challenges are catastrophic forgetting and parameter interference\textsuperscript{1} for finetuning LLMs when provided parallel training data. To address these challenges, we propose LANDeRMT, a Language-Aware Neuron Detecting and Routing framework that selectively finetunes LLMs to Machine Translation with diverse translation training data. In LANDeRMT, we evaluate the awareness of neurons to MT tasks and categorize them into language-general and language-specific neurons. This categorization enables selective parameter updates during finetuning, mitigating parameter interference and catastrophic forgetting issues. For the detected neurons, we further propose a conditional awareness-based routing mechanism to dynamically adjust language-general and language-specific capacity within LLMs, guided by translation signals. Experimental results demonstrate that the proposed LANDeRMT is very effective in learning translation knowledge, significantly improving translation quality over various strong baselines for multiple language pairs.

1 Introduction

Conventional neural machine translation (NMT) usually requires a huge amount of parallel training data (Costa-jussà et al., 2022; Fedus et al., 2022; Zhu et al., 2023, 2024b). In contrast, multilingual LLMs, e.g., BLOOM (Scao et al., 2022), LLama2 (Touvron et al., 2023), in spite of being trained with mainly monolingual data, require only a few examples to demonstrate remarkable prowess in multilingual translation via in-context learning (ICL) (He et al., 2023; Lyu et al., 2023). However, such LLM-based MT exhibits a major drawback that the quality of yielded translations is highly sensitive to the provided examples in ICL (Vilar et al., 2023) and outputs might suffer from overgeneration (Bawden and Yvon, 2023).

To address these issues, existing studies attempt to use various finetuning methods, such as adapter-based method (Alves et al., 2023), instruction-based tuning method (Li et al., 2023a). However, these approaches primarily focus on balancing between the original LLMs and new finetuning translation data They use only incremental data to acquire new knowledge without considering catastrophic forgetting of knowledge originally captured by LLMs (Liu et al., 2021; Shao and Feng, 2022; Huang et al., 2023). Many studies have shown that catastrophic forgetting indeed exists across languages as LLMs are fine-tuned on one language pair and then used to translate another language on which LLMs are not fine-tuned (Li et al., 2023b; Zhu et al., 2024a). Additionally, as LLMs are usually generally developed for multiple tasks (i.e., sharing parameters across different tasks), finetuning LLMs for MT task may cause parameter interference for other tasks (Luo et al., 2023). In Section 4.3, we find that full-parameter finetuning of LLMs cannot always improve translation quality on all language pairs. Therefore, is it possible to design a new finetuning method for LLMs, which can mitigate the issues of catastrophic forgetting and parameter interference during the finetuning process of LLMs to multilingual machine translation?

In multilingual NMT, previous efforts evaluate the importance of model neurons to each language pair and only tune language-specific neurons for the current language pair during training (Xie et al.,
Recent studies on LLM unveil that many neurons in the feed-forward networks (FFN) are only activated for specific tasks and become “dead” for irrelevant tasks (Voita et al., 2023; Conmy et al., 2023). Inspired by these studies, we propose LANDeRMT, a language-aware neuron detecting and routing framework for selectively finetuning LLMs to MT, which aims to mitigate the issues of catastrophic forgetting and parameter interference. First, we evaluate the MT awareness of each neuron in FFNs. For neurons that are related to multilingual MT tasks, we further evaluate the relevance of each neuron to each language pair. According to their MT/language “awareness/relevance”, we divide neurons into the unactivated neurons, language-general neurons and language-specific neurons. After that, we finetune LLMs on multilingual parallel training data. During finetuning, only the parameters of language-general neurons and language-specific neurons for the current language pair are tuned. This selective finetuning process can alleviate the parameter interference issue.

As language-general and language-specific capacity matters for MT (Zhang et al., 2021; Koishekenov et al., 2023), we propose a conditional awareness routing mechanism to dynamically schedule language-general and language-specific capacity across sub-layers in LLMs under the guidance of translation signals. In doing so, we can alleviate the catastrophic forgetting issue and facilitate LLMs to be adapted to MT.

The main contributions of this work are summarized as follows:

- We propose LANDeRMT that aims at mitigating the catastrophic forgetting and parameter interference issues for efficiently finetuning LLMs to MT.
- To well schedule language-general and language-specific capacity across sub-layers in LLMs, we propose a conditional awareness-based routing mechanism.
- Experiments on ten language pairs show that our model achieves the state-of-the-art results compared to previous strong baselines and demonstrate the robustness of the proposed model in various settings.

2 Related Work

LLMs, with a few examples provided via in-context learning, have demonstrated impressive capabilities in machine translation without requiring explicit supervision from parallel training data (Moslem et al., 2023; Ghazvininejad et al., 2023; Sia and Duh, 2023; Han et al., 2022). However, LLMs with ICL for MT suffer from the sensitiveness to the provided examples (Vilar et al., 2023) and yielded translations might be overgenerated (Bawden and Yvon, 2023).

Another line of research on LLMs, known as domain-adaptive pretraining, focuses on finetuning LLMs to downstream tasks (Cheng et al., 2023; Dong et al., 2023). Although these approaches have demonstrated efficacy in adapting various LLMs and result in enhanced performance on downstream tasks (Wu et al., 2023; Gupta et al., 2023; Wu et al., 2024; Zhu and Xiong, 2023), they rarely apply to multilingual generation tasks, e.g., multilingual MT.

In order to efficiently adapt LLMs to MT, recent years have witnessed efforts on finetuning LLMs for MT (Vilar et al., 2023; Alves et al., 2023). Alves et al. (2023) show that adapter-based finetuning with LoRA (Hu et al., 2022) matches the performance of traditional finetuning while reducing the number of training parameters by a factor of 50. Li et al. (2023a) investigate the multilingual generalization when finetuning LLMs. However, they do not explicitly overcome catastrophic forgetting and parameter interference issues. To address these issues, our work starts with analyzing the neurons within the model, and finetunes LLMs by distinguishing neurons.

Research interests in understanding the inner workings of LLMs and NMT models have been growing recently (Räuker et al., 2022; Bills et al., 2023; Garde et al., 2023). Voita et al. (2023) focus on neurons inside FFNs and find that the network is sparse and represents many discrete features in LLMs. They find many of the alive neurons are reserved for discrete features and act as token and n-gram detectors for different languages. In addition, previous NMT efforts evaluate the importance of NMT neurons in each language pair and only finetune language-specific neurons for the current language pair participate in training for conventional multilingual NMT (Xie et al., 2021; Patel et al., 2022). Partially motivated by these studies, we propose a language-aware neuron detecting
and routing framework for selectively finetuning LLMs to MT. In our method, we use awareness-based evaluation of neurons in LLMs and divide the neurons into language-general and language-specific neurons. We only update the parameters of language-general neurons and the corresponding language-specific neurons for the current language pair during training to overcome catastrophic forgetting and parameter interference to enhance the multilingual translation ability of LLMs.

3 Methodology

The proposed LANDeRMT is illustrated in Figure 1. We first propose a method to analyse which FFN layers of LLMs have strong relevance to source-target language pair. This allows us to exclusively concentrate on layers that are related to the MT task, hence reducing the distraction from unrelated parameters. Then, we employ Taylor Expansion (TE) (Xie et al., 2021) to evaluate the strength of awareness (relevance) of neurons at those layers to the given language pairs of the MT task. Finally, we only route and finetune the detected language-aware neurons for the MT task. This can ensure that we only need to update a small number of relevant parameters of LLMs for MT.

3.1 Detecting Language-Pair-Relevant Layers

We introduce a representation analysis (RA) method to detect language-pair-relevant layers, which is based on the difference in activations between FFN layers. RA aims to measure the changes in the response of each FFN layer to the input source sentence during the LLM forward propagation process that “translates” the source sentence into the target sentence, so as to identify FFN alignment layers that are highly “activated” for the source-target language pair. For each consecutive pair of layers \( i \) and \( i + 1 \) within the LLM, we compute the activation difference \( D \), to estimate the degree of change in information representation between these two layers. The estimation is computed as follows:

\[
D_i = \left| \frac{1}{N} \sum_{n=1}^{N} A_{i,n} - \frac{1}{N} \sum_{n=1}^{N} A_{i+1,n} \right| \tag{1}
\]

where \( A_{i,n} \) and \( A_{i+1,n} \) represent the activation values at the \( i \)-th and \( i + 1 \)-th layers during the \( n \)-th forward propagation, and \( N \) is the total number of forward propagations. In this manner, \( D_i \) captures the extent of change in activation values between adjacent layers along the depth of the model when the input source language is translated into the target language. The most significant changes in layer representations indicate the most critical layers that are related to the source-target language pair. Therefore, our layer selection criterion focuses on identifying those layers with the top-\( k \) \( D \) values as follows:
\[ L_{LPR} = \arg \max_{i \in \text{top}k} \{D_1, D_2, ..., D_k \} \]  

where \( L_{LPR} \) denotes the optimal language-pair-relevant layers.

### 3.2 Evaluating the Language Awareness of Neurons

Once we find the language-pair-relevant layer, do we need to finetune all neurons of the layer for the corresponding language pair? Our experiments show that this all-neuron-finetuning strategy is not as expected (see Section 4.4). The main reasons are two fold. First, if all parameters of the detected FFNs are updated for all language pairs, catastrophic forgetting problem still remains (Liu et al., 2021). Second, there is no effective mechanism to overcome the parameter interference issue to preserve the language-general and the language-specific knowledge.

Partially inspired by the studies on the importance-based neuron finetuning for NMT (Xie et al., 2021) and neuron interpretability in LLMs (Voita et al., 2023), we propose to use the TE to evaluate which neurons are essential to all languages and which neurons are responsible for specific languages. We first define the awareness score \( \Phi(i) \) of a neuron to a certain language:

\[
\Phi(i) = |\Delta \mathcal{L}(\mathbf{h}_i)|, \ i \in L_j
\]  

\( L_j \) is the \( j \)-th layer that is the detected language-pair-relevant layer. \( \mathbf{h}_i \) is the output of neuron \( i \). \( \Delta \mathcal{L}(\mathbf{h}_i) \) is the loss change between setting \( \mathbf{h}_i \) to \( \mathbf{0} \) and keeping it at its original value. It can be transformed by TE into the following form:

\[
|\Delta \mathcal{L}(\mathbf{h}_i)| = \left| \frac{\partial \mathcal{L}}{\partial \mathbf{h}_i} \mathbf{h}_i \right|
\]  

We estimate the loss change as the product of the gradient of the loss function with respect to the activation value and the activation value itself. The detailed proof can be found in the appendix A.2, which is similar to that by Xie et al. (2021). Then, we determine which neurons are shared across all language pairs (i.e., language-general neurons) and which neurons are only related to specific language pairs.

We define \( \mathbf{X}_i \) as the vector of awareness scores of the \( i \)-th neuron for each language. For each neuron, we calculate the variance \( \sigma(X_i) \) of the awareness scores across different languages. Within a specific layer, we sort the neuron awareness scores based on their variance from the highest to the lowest. A variance threshold \( \lambda(i) \) is calculated to distinguish language-general neurons from language-specific neurons as follows:

\[
\lambda(i) = \text{sort}(\sigma(\mathbf{X}_i))_{\epsilon \times p}, i \in L_j
\]  

where \( p \) is the number of neurons in the \( L_j \) layer, \( \epsilon \) is a predefined ratio. For neurons with language awareness score variances below the estimated threshold \( \lambda(i) \), we categorize them as language-general neurons, otherwise as language-specific neurons. Each detected language-specific neuron is assigned to the language with the highest awareness score.

The set of neurons that are specific either to the source language or to the target language are aggregated as the neurons exclusive to that language pair.

### 3.3 Routing and Finetuning

In our proposed framework, for a given bilingual dataset of a specific language pair, only the language-general and language-specific neurons of the detected FFNs for this language pair participate in the forward computation and the parameters associated with them are updated during the backward propagation, as illustrated in Figure 2. Nevertheless, it has been empirically shown that the language signals from language indicator tokens alone are not sufficient (Arivazhagan et al., 2019), making modules or mechanisms dedicated to language-general and language-specific modeling a necessity (Zhang et al., 2020, 2021). To address this issue, we propose a conditional awareness-based routing mechanism (CAR) that allows the model to decide and learn what proportion of the outputs
of language-general and language-specific neurons should be allocated for the translation of the language pair. For an input token $x_t$, CAR is evaluated as follows:

$$\text{CAR}(x_t) = \frac{\sum_{i=1}^{N} \phi(i)}{\sum_{i=1}^{N} \phi(i) + \sum_{j=1}^{M} \phi(j)} \quad (6)$$

$$h_G(x_t) = \text{FFN}_G(\text{CAR}(x_t), x_t) \quad (7)$$

$$h_S(x_t) = \text{FFN}_S((1 - \text{CAR}(x_t)), x_t) \quad (8)$$

where $G$ denotes language-general and $S$ language-specific. $\text{FFN}_G$ and $\text{FFN}_S$ are language-general and language-specific neurons, respectively. $N$ is the total number of language-specific neurons in a FFN layers for a language pair. $M$ is the total number of language-general neurons in a FFN layers. We combine $\text{FFN}_G$ and $\text{FFN}_S$ to alleviate the parameter interference. The fusion output $H^f$ is given by:

$$H^f = h_G(x_t) + h_S(x_t) \quad (9)$$

Uppercase $H^f$ is just a notation here for the addition result of $h_G(x_t)$ and $h_S(x_t)$, which is only used to distinguish it from $h_G(x_t)$ and $h_S(x_t)$. During the finetuning stage, we only update the parameters of language-general and language-specific neurons for a specific language pair and freeze other parameters of LLMs.

4 Experiments

We conducted extensive experiments with involving multiple models across various translation directions to evaluate the proposed framework against a set of strong baselines.

4.1 Dataset

During the finetuning stage, we selected 5 language pairs to tune LLMs. All the original training data came from the recent WMT general translation track. All data followed the license that can be freely used for research purposes. In addition, we used the way in (Huang et al., 2023) to clean training data. All datasets originated from the Workshop on Machine Translation (WMT)\textsuperscript{2}. Specifically, we extracted 200,000 sentence pairs for each translation direction. In addition, we employed ten translation instruction finetuning templates sourced from FLAN v2 (Longpre et al., 2023), adopting them to our parallel data. Each sentence pair from parallel corpus was randomly assigned one translation instruction template. We assessed our model’s performance using established test sets like WMT16, WMT14 and OPUS-100.

4.2 Settings and Baselines

**Settings** In the language-pair-relevant layers detection phase, we set $k$ to 4. We executed a freezing operation on the parameters of the remaining layers while exclusively finetuning the parameters within the chosen four layers. In the language-aware neurons evaluation phase, we categorized the parameters within the selected layer into language-general and language-specific parameters, setting $\epsilon$ to 0.9. During the model finetuning stage, we configured the finetuning hyper-parameters as follows: the finetuning epoch was set to 1, the number of language pairs was specified as 10, the number of iterations per epoch for each language pair was set to 12,500, the batch size was set 8, and the AdamW optimizer was employed. Additionally, the learning rate was set to 5e-5. Furthermore, we introduced a gradient accumulation operation, updating the model parameters every 10 iterations to enhance convergence. The LLMs used for our experiments are BLOOM-7b1 (Scao et al., 2022) and Baichuan2-7B-Base (Yang et al., 2023).

**Baselines** We compared our approach to the following strong baselines.

- **0-shot**: This approach uses instructions directly to make the model perform downstream tasks without providing any in-context demonstrations.

- **In-context (Zhang et al., 2023)**: This is a training-free approach that allows the LLMs to perform downstream tasks. In particular, we use 5 random shots as in-context demonstrations.

- **Adapter**: This method facilitates the acquisition of new knowledge by incorporating additional adapter modules following model-specific layers, effectively addressing the issue of catastrophic forgetting.

\textsuperscript{2}https://www.statmt.org/
### Methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>Params</th>
<th>WMT16 en-de</th>
<th>WMT16 de-en</th>
<th>WMT16 en-it</th>
<th>WMT16 it-en</th>
<th>WMT14 en-fr</th>
<th>WMT14 fr-en</th>
<th>OPUS-100 en-ar</th>
<th>OPUS-100 ar-en</th>
<th>OPUS-100 en-zh</th>
<th>OPUS-100 zh-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-shot</td>
<td></td>
<td>11.71</td>
<td>17.82</td>
<td>8.07</td>
<td>16.05</td>
<td>19.58</td>
<td>18.88</td>
<td>9.46</td>
<td>26.37</td>
<td>6.14</td>
<td>22.02</td>
</tr>
<tr>
<td>In-context</td>
<td></td>
<td>11.49</td>
<td>14.57</td>
<td>9.80</td>
<td>13.12</td>
<td>18.74</td>
<td>15.29</td>
<td>10.01</td>
<td>21.24</td>
<td>6.83</td>
<td>17.19</td>
</tr>
<tr>
<td>Adapter</td>
<td>806M</td>
<td>15.61</td>
<td>19.67</td>
<td>15.25</td>
<td>23.63</td>
<td>28.08</td>
<td>27.26</td>
<td>11.28</td>
<td>28.07</td>
<td>15.18</td>
<td>25.05</td>
</tr>
<tr>
<td>LoRA</td>
<td>31M</td>
<td>15.12</td>
<td>19.03</td>
<td>14.82</td>
<td>22.85</td>
<td>27.16</td>
<td>27.72</td>
<td>11.15</td>
<td>27.74</td>
<td>15.72</td>
<td>25.12</td>
</tr>
<tr>
<td>Adapter-LoRA</td>
<td>806M</td>
<td>16.31</td>
<td>20.23</td>
<td>15.83</td>
<td>23.82</td>
<td>28.05</td>
<td>28.22</td>
<td>11.78</td>
<td>28.46</td>
<td>16.32</td>
<td>25.61</td>
</tr>
<tr>
<td>LANDeRMT (Ours)</td>
<td>805M</td>
<td><strong>18.85</strong></td>
<td><strong>22.03</strong></td>
<td><strong>19.82</strong></td>
<td><strong>25.99</strong></td>
<td><strong>31.91</strong></td>
<td><strong>30.55</strong></td>
<td><strong>16.97</strong></td>
<td><strong>31.44</strong></td>
<td><strong>22.47</strong></td>
<td><strong>28.11</strong></td>
</tr>
</tbody>
</table>

Table 1: BLEU scores on the 10 language pairs for xx-to-English and English-to-xx translation. The highest score on each translation direction is highlighted in bold.

- **LoRA (Hu et al., 2022):** This method efficiently finetunes a model for a downstream task by converting certain structures into low-rank matrices and subsequently finetuning them to suit the task.

- **Adapter-LoRA (Alves et al., 2023):** It uses adapter-based finetuning with LoRA, which matches the performance of traditional finetuning while reducing the number of training parameters.

### 4.3 Main Results

For evaluating translation performance, we used two automatic evaluation metrics sacreBLEU3.

**Comparison with ICL** In order to examine the effectiveness of our proposed method, we evaluated LANDeRMT on various test set and multiple language pairs. As shown in Table 1, our method can use new parallel training data to enhance the translation ability of LLMs.

**Comparison with finetuning baselines** Compared to the baselines, our method is the best for all translation directions in Table 1. For relatively low-resource language pairs, such as English-Chinese, our method achieves significant improvements over baselines. Compared to the full parameter finetuning approach, our method has a clear parametric advantage since it only fine-tunes parameters in four layers in the model. Our approach exhibits a notable advantage in terms of the number of parameters to be tuned. In comparison to other efficient finetuning methods, e.g., the adapter baseline approach, our method finetunes basically the same number of parameters as it. However, our method is much better than the adapter-based approach in terms of translation quality.

### 4.4 Ablation Study

In the ablation experiments, we employed four distinct experimental setups, denoted as Layers, LANDeRMT-LS, LANDeRMT-LG, and LANDeRMT. The Layers configuration finetuned all parameters of the language-pair-relevant layer for each language direction. In the LANDeRMT-LS setup, we finetuned only the language-specific parameters of the selected layers, with each language direction adjusting parameters specific to that language direction. The LANDeRMT-LG setup focused on finetuning only the language-general parameters of the selected layers, with all language directions adjusting the same language-general parameters. The LANDeRMT method, proposed in this paper, finetuned both the language-general parameters and language-specific parameters to each language pair.

From Table 2, we observe that LANDeRMT-LS underperforms the Layers method, likely due to its smaller parameter size, which constitutes only 10% of the parameters in Layers. In details, we can observe that LANDeRMT achieves a 4.28 BLEU improvement over Layers in en-fr, a 9.64 BLEU improvement over LANDeRMT-LS, and a 3.76 BLEU improvement over LANDeRMT-LG. These experiments demonstrate the effectiveness of CAR. Surprisingly, LANDeRMT-LG achieves better re-

---

3BLEU+case.mixed+numrefs.1+smooth.none+tok.13a+version.2.2.1
results despite finetuning fewer parameters than Layers. This suggests that our selection of language-general parameters effectively captures language alignment, significantly improving translation performance. However, finetuning the language-general parameters alone, as in LANDeRMT-LG, is insufficient to fully grasp language-specific information.

5 Analysis

5.1 LANDeRMT Improves Transfer Learning across Languages

We examined the transfer learning ability of LANDeRMT in different translation directions. We finetuned the model using only parallel data from a particular language direction. In other words, we finetuned only the language-general and language-specific parameters for that language pair, and then observed the performance of the model in other language directions. The Y-axis of Figure 3 shows the single language direction that we have finetuned, and the X-axis shows the language direction of the test data, which is plotted as the improvement in the model’s translation performance before and after the finetuning. Since BLOOM-7b is a model that is not mainly trained on a parallel corpus, its translation performance before finetuning is poor, which is the reason for the large improvement in the model’s translation performance. We observe that when finetuning one language direction, the results of other language directions can also be significantly improved, which proves that our method is effective in facilitating transfer learning between languages. However, there are some exceptions. For example, when finetuning the de-en direction, the ar-en direction does not improve significantly or even decrease to some extent. We believe that this may be due to the fact that Arabic belongs to a different language family from German, and that the distance between the languages is far, making it difficult for cross-lingual transfer learning.

5.2 Language-Pair-Relevant Layers for Different Language Pairs

Figure 4 illustrates the variation in the average activation values across each layer of the model when inputting translation instructions generated using diverse language pairs. It is noteworthy that the average activation values of various language pairs exhibit a similar trend of change, particularly in the shallower layers of the model. Moreover, when interchanging the source and target languages within language pairs, the average activation values consist-
tently follow a more uniform trend, as evidenced by the translation directions of ar-en and en-ar. This indicate that not only the semantic remains consistent between the source and target languages within the same language pair, but the identical semantic is still existing on across different language pairs.

The absolute value of activation value changes from layer to layer can be calculated by using the average activation values of each layer, as illustrated in Figure 5. We can find that early and late-stage layers in the model harbor information pertaining to language pairs. For instance, layers 6 and 7, as well as the final layers, exhibit higher absolute values of activation value changes. Furthermore, an observation can be made that the early layers of the model encapsulate language pair-related information that is language pair-general, displaying a substantial overlap across different language pairs. Conversely, the layers towards the end of the model contain language pair-related information that is language pair-specific, characterized by a diminished overlap among different language pairs.

5.3 Neuron Awareness for Different Languages: General and Specific

The main idea of our proposed method is to distinguish language-general from language-specific neurons. To verify whether this goal has been achieved, we conducted the following experiments. As mentioned in Section 3.2, we categorize neurons based on their awareness scores, and we observed significant differences in the awareness scores of language-general and language-specific neurons across layers requiring fine-tuning. We illustrate this in Figure 6 and Figure 7. We can find that for almost all language pairs, there are noticeable differences in awareness scores between certain intermediate layers and the final layers. This indicates that our categorization of neurons based on layers accurately reflects the practical scenario. It also suggests that language-general and language-specific neurons exhibit varying levels of importance across different layers of the model, particularly in layers targeted for fine-tuning. Such differences likely stem from the distinct roles that language-general and language-specific neurons play in capturing and processing language-specific and language-general information.

5.4 Neuron Awareness for Different Language Pairs

Figure 8 depicts the average neuron awareness scores for each layer of the model, computed using TE with monolingual data inputs in various
language pairs. The employed multidirectionally aligned monolingual data ensures semantic one-to-one correspondence. The results show consistent trends in neuron awareness scores across different language pairs, particularly in the intermediate layers, indicating the model’s ability to capture semantic information consistently across language pairs. Additionally, related languages such as Spanish, English, and French exhibit more similar trends, supporting our hypothesis.

Furthermore, we observed that language-specific neurons tend to have higher awareness scores in the last layers of the model. This suggests a heightened focus on encoding and retaining language-specific semantic information during the output phase, particularly in deeper layers. Notably, language-specific neurons related to English consistently exhibit high awareness scores across all language pairs. This can be attributed to the prevalence of English data during the model’s pre-training phase, indicating robust representation and preservation of English language-specific information throughout the model.

5.5 Results on Other LLMs
We also finetuned the Baichuan-7B-Base model using the LANDeRMT method and compared it with the adapter-LoRA finetuning approach. Results are shown in Figure 9. We observe that across the 10 language directions selected our proposed method outperforms the adapter-LoRA finetuning method. This demonstrates the applicability of the LANDeRMT method across different models, achieving optimal results not only in the BLOOM-7b models but also in the Baichuan model.

6 Conclusion
In this paper, we have presented a novel approach that not only improves translation quality but also mitigates the risk of forgetting previous knowledge while adapting to new data. We propose a TE to evaluate neuron awareness scores for MT tasks and categorize them into language-general neurons and language-specific neurons. The proposed routing mechanism ensures optimal allocation of resources across language-specific and language-general capacities, further enhancing the adaptability of LLMs. Our experimental results, conducted across ten language pairs, validate the effectiveness of our model, showcasing superior performance compared to existing baselines.

Acknowledgments
The present research was supported by the National Natural Science Foundation of China Youth Foud (Grant No.62306210) and the Key Research and Development Program of Yunnan Province (Grant No. 202203AA080004). We would like to thank the anonymous reviewers for their insightful comments.
Limitations

Although LANDeRMT is a new approach to fine-tune LLMs to enhance the translation ability of LLMs. The finetuning procedure is shorter than training LLMs as the amount of data required during the finetuning stage is much smaller than during the training stage. This significantly reduces the cost of training model from scratch but maybe still totally overcome parameters interference as we not fully update the parameters of LLMs. Additionally, due to computational constraints, we are currently unable to design additional experiments to validate how our method enhances the upper limit of translation capabilities of LLMs when more training data is added.

Ethics Statement

This study adheres to the ethical guidelines set forth by our institution and follows the principles outlined in the ACM Code of Ethics and Professional Conduct. All datasets used in our experiments are publicly available.

References


Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bien, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhao, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiyi Wu. 2023. Baichuan 2: Open large-scale language models. CoRR, abs/2309.10305.


A Appendix

A.1 Language Details

We introduce the characteristics of different languages as shown in Table 3.

<table>
<thead>
<tr>
<th>Code</th>
<th>Language</th>
<th>Genus</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>en</td>
<td>English</td>
<td>Romance</td>
<td>SVO</td>
</tr>
<tr>
<td>ar</td>
<td>Arabic</td>
<td>Semitic</td>
<td>VSO</td>
</tr>
<tr>
<td>fr</td>
<td>French</td>
<td>Romance</td>
<td>SVO</td>
</tr>
<tr>
<td>de</td>
<td>German</td>
<td>Germanic</td>
<td>SVO</td>
</tr>
<tr>
<td>zh</td>
<td>Chinese</td>
<td>Sinitic</td>
<td>SVO</td>
</tr>
<tr>
<td>it</td>
<td>Italian</td>
<td>Romance</td>
<td>SVO</td>
</tr>
</tbody>
</table>

Table 3: The characteristics of languages in our setting.

A.2 Taylor Expansion

We first express $\Delta L(h_i)$ as loss change as shown in the following equation.

$$|\Delta L(h_i)| = |L(H, h_i = 0) - L(H, h_i)|$$

$H$ is the representation produced by a neuron other than $i$ in the same structure as the $i$ neuron. We then perform a first-order Taylor expansion of $L(H, h_i)$ at $h_i = 0$.

$$L(H, h_i) = L(H, h_i = 0) + \frac{\partial L(H, h_i)}{\partial h_i} h_i + R_1(h_i)$$

The term $R_1(h_i)$ can be ignored since the derivatives of the activation function of second order and higher in the model tend to zero. So the above equation can be reduced to the following form.

$$L(H, h_i) \approx L(H, h_i = 0) + \frac{\partial L(H, h_i)}{\partial h_i} h_i$$

Therefore $|\Delta L(h_i)|$ can eventually be simplified to the following form.

$$|\Delta L(h_i)| \approx \left| \frac{\partial L(H, h_i)}{\partial h_i} h_i \right|$$

A.3 Effect of Hyperparameter $k$

When the relation of layers for different languages is determined, the number of language pairs associated with each layer can be adjusted according to $k$. When $k = 30$, the threshold is max, so all layers will be allocated to tune LLMs, and when $k = 0$, the threshold is 0 so none layers will be tuning for all language pairs just like the 0-shot ICL. To better show the overall impact of the hyperparameter $k$, we vary it from 0 to 30 and the results are shown in Figure 10. As we can see, the translation performance of the proposed approach increases with the increment of $k$ and reach the best performance when $k$ equals 4. As $k$ continues to increase, the performance deteriorates, which indicates that the over-specific layers are bad at capturing the common language-pair-relevant alignment and will lead to performance degradation.

A.4 Effect of Hyperparameter $\epsilon$

We set several different sets of $\epsilon$ values to classify language-general neurons and language-specific neurons. The experimental outcomes are depicted in Figure 11. The figure reveals that the average value of all language-specific BLEU scores peaks when language-general neurons constitute 0.9 of the total neuron count. Below this threshold, the
translation efficacy diminishes as the proportion of language-general neurons decreases. Conversely, exceeding the 0.9 threshold results in a decline in performance, with a higher proportion of language-general neurons leading to poorer results.

A.5 Language Cluster

The main idea of our proposed method is to let the language-general and the language-specific knowledge be captured by different neurons. To validate whether the language-general and language-specific neurons of FFNs within LLMs general or specific language knowledge, we plotted the distribution of different neurons across various languages in 10-th layer, as shown in Figure 12 and Figure 13. From these figures, it is evident that for the language-general FFNs neurons, the distributions for various languages intersect without clear boundaries, indicating a shared representation of language knowledge. In contrast, for the language-specific neurons, the boundaries between language distributions are highly distinct, highlighting the independence of language-specific knowledge. This observation underscores our neurons awareness can make neurons capture and integrate language knowledge in a general manner across multiple languages within the language-general FFNs. Conversely, the distinct boundaries in the distributions of language-specific FFNs neurons suggest that these neurons are dedicated to encoding language-specific nuances and characteristics.