Neuron Specialization: Leveraging Intrinsic Task Modularity for Multilingual Machine Translation

Shaomu Tan **Di Wu** **Christof Monz**
Language Technology Lab
University of Amsterdam
{s.tan, d.wu, c.monz}@uva.nl

Abstract

Training a unified multilingual model promotes knowledge transfer but inevitably introduces *negative interference*. Language-specific modeling methods show promise in reducing interference. However, they often rely on heuristics to distribute capacity and struggle to foster cross-lingual transfer via isolated modules. In this paper, we explore intrinsic task modularity within multilingual networks and leverage these observations to circumvent interference under multilingual translation. We show that neurons in the feed-forward layers tend to be activated in a language-specific manner. Meanwhile, these specialized neurons exhibit structural overlaps that reflect language proximity, which progress across layers. Based on these findings, we propose *Neuron Specialization*, an approach that identifies specialized neurons to modularize feed-forward layers and then continuously updates them through sparse networks. Extensive experiments show that our approach achieves consistent performance gains over strong baselines with additional analyses demonstrating reduced interference and increased knowledge transfer.[1]

1 Introduction

Jointly training multilingual data in a unified model with a shared architecture for different languages has been a trend (Conneau et al., 2020; Le Scao et al., 2022) encouraging knowledge transfer across languages, especially for low-resource languages (Johnson et al., 2017; Pires et al., 2019). However, such a training paradigm also leads to *negative interference* due to conflicting optimization demands (Wang et al., 2020). This interference often causes performance degradation for high-resource languages (Li and Gong, 2021; Pfeiffer et al., 2022) and can be further exacerbated by limited model capacity (Shaham et al., 2023).

Modular-based methods, such as Language-specific modeling (Zhang et al., 2020b) and adapters (Bapna and Firat, 2019), aim to mitigate interference by balancing full parameter sharing with isolated or partially shared modules (Pfeiffer et al., 2023). However, they heavily depend on heuristics for allocating task-specific capacity and face challenges in enabling knowledge transfer between modules (Zhang et al., 2020a). Specifically, such methods rely on prior knowledge for managing parameter sharing such as language-family adapters (Chronopoulou et al., 2023) or directly isolate parameters per language, which impedes transfer (Pires et al., 2023).

Research in vision and cognitive science has shown that unified multi-task models may spontaneously develop task-specific functional specializations for distinct tasks (Yang et al., 2019; Dobs et al., 2022), a phenomenon also observed in mixture of experts Transformer systems (Zhang et al., 2023). These findings suggest that through multi-task training, networks naturally evolve towards specialized modularity to effectively manage diverse tasks, with the ablation of these specialized modules adversely affecting task performance (Pfeiffer et al., 2023). Despite these insights, exploiting the inherent structural signals for multi-task optimization remains largely unexplored.

In this work, we explore the intrinsic task-specific modularity within multi-task networks in Multilingual Machine Translation (MMT), treating each language pair as a separate task. We focus on analyzing the intermediate activations in the Feed-Forward Networks (FFN) where most model parameters reside. To our knowledge, our study is the first to show that neurons activate in a language-specific way, yet they present structural overlaps that indicate language proximity in general. Moreover, this pattern evolves across layers in the model, suggesting that neurons consistently transition from language-specific to language-agnostic.

[1] We release code at https://github.com/Smu-Tan/Neuron-Specialization.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 6506–6527
November 12-16, 2024 ©2024 Association for Computational Linguistics

Building on these observations, we introduce *Neuron Specialization*, a novel method that leverages intrinsic task modularity to reduce interference and enhance knowledge transfer. In general, our approach selectively updates the FFN parameters during back-propagation for different tasks to enhance task specificity. Specifically, we first identify task-specific neurons from pre-trained unified translation models, using standard forward-pass validation processes without decoding. We then specifically modularize FFN layers using these specialized neurons and continuously update FFNs via sparse networks.

Extensive experiments on small- (IWSLT) and large-scale EC30 (Tan and Monz, 2023) translation datasets show that our method consistently achieves performance gains over strong baselines with various configs. Moreover, we conduct in-depth analyses to show that our method effectively mitigates interference and enhances knowledge transfer in high and low-resource languages, respectively. Our main contributions are summarized as follows:

- We identify inherent multilingual modularity by showing that neurons activate in a language-specific manner and their overlapping patterns reflect language proximity.
- Building on these findings, we enhance task specificity through sparse FFNs, achieving consistent improvements in translation quality over strong baselines.
- We employ analyses to show that our method effectively reduces interference in high-resource languages and boosts knowledge transfer in low-resource languages.

2 Related Work

Multilingual Interference. Multilingual training enables knowledge transfer but also causes *interference*, largely due to optimization conflicts among various tasks (Wang and Zhang, 2022). Methods alleviating task conflicts hold promise to reduce interference (Wang et al., 2020), yet they show limited effectiveness in practice (Xin et al., 2022). Scaling up model size may reduce interference but leads to overly large models (Chang et al., 2023), with risks of overfitting (Aharoni et al., 2019).

Language-Specific Modeling. Recent methods enhance the unified model by utilizing language-specific (LS) modules such as adapters (Bapna and Firat, 2019), LS layers (Zhang et al., 2020b; Pires et al., 2023) and LS hidden states (Xie et al., 2021). Although the unified model serves as a common foundation, these methods strictly isolate modules per language. Such designs present no knowledge sharing among modules and thus offer fewer benefits to low-resource languages. Alternatively, approaches like language family adapters Chronopoulou et al. (2023) seek to facilitate sharing among language-specific modules, however, they heavily depend on heuristics such as using priori linguistic knowledge to enable more flexible parameter sharing.

Additionally, these modular-based methods exhibit parameter inefficiency when handling numerous languages, resulting in increased memory requirements and extended inference times (Liao et al., 2023a,b). Similarly, techniques such as parameter differentiation (Wang and Zhang, 2022) and language clustering training (Tan et al., 2019) alleviate interference by expanding the unified model with substantial extra parameters.

Sub-networks in Multi-task Models. The lottery ticket hypothesis (Frankle and Carbin, 2018) states that within dense neural networks, sparse subnetworks can be found with iterative pruning to achieve the original network's performance. Following this premise, recent studies attempt to isolate sub-networks of a pre-trained unified model that captures task-specific features (Choenni et al., 2023a; Lin et al., 2021; He et al., 2023). Nonetheless, unlike our method that identifies intrinsic modularity within the model, these approaches depend on fine-tuning to extract the task-specific sub-networks. This process may not reflect the original model modularity and also can be particularly resource-consuming for multiple tasks.

Specifically, these methods extract the task-specific sub-networks by fine-tuning the original unified multi-task model on specific tasks, followed by employing pruning to retain only the most changed parameters. We argue that this process faces several issues: 1) The sub-network might be an artifact of fine-tuning, suggesting the original model may not inherently possess such modularity. 2) This is further supported by the observation that different random seeds during fine-tuning lead to varied sub-networks and performance instability (Choenni et al., 2023a). 3) The process is highly inefficient for models covering multiple tasks, as it necessitates separate fine-tuning for each task.

3 Neuron Structural Analysis

Recent work aims to identify a subset of parameters within pre-trained multi-task networks that are sensitive to distinct tasks. This exploration is done by either 1) selecting hidden states that greatly influence task performance (Dobs et al., 2022) or possess high magnitude values (Xie et al., 2021); or 2) fine-tuning the unified model on task-specific data to extract sub-networks (Lin et al., 2021; He et al., 2023; Choenni et al., 2023b). These approaches, however, raise a fundamental question, namely whether the modularity is inherent to the original model, or simply an artifact introduced by network modifications.

In this paper, we perform a thorough identification of task-specific modularity through the lens of neuron behaviors, without altering the original parameters or architectures. We focus on the neurons — the intermediate activations inside the Feed-Forward Networks (FFN) — to investigate if they indicate task-specific modularity features.

As FFN neurons are active (>0) or inactive (=0) due to the $ReLU$ activation function[2], this binary activation state offers a clear view of their contributions to the network's output. Intuitively, neurons that remain inactive for one task but show significant activation for another may be indicative of specialization for the latter. More importantly, this approach ensures that both parameters and hidden states remain unchanged, affirming the observed modularity is inherent to the original model.

3.1 Identifying Specialized Neurons

We choose multilingual translation as a testbed, treating each translation direction as a distinct task throughout the paper. We start with a pre-trained multilingual model with d_{ff} as its dimension of the FFN layer. We hypothesize the existence of neuron subsets specialized for each task and describe the identification process of an FFN layer as follows.

Activation Recording. Given a validation dataset D_t for the t-th task, we measure activation frequencies in an FFN layer during validation. For each sample $x_i \in D_t$, we record the state of each neuron after the activation function $\sigma(\cdot)$, reflecting whether the neuron is active or inactive to the sample. We use a binary vector $a_i^t \in \mathbb{R}^{d_{ff}}$ to store this neuron state information. Note that this vector aggregates neuron activations for all tokens in the sample by taking the neuron union of them. By further merging all of the binary vectors for all samples in D_t, an accumulated vector $a^t = \sum_{x_i \in D_t} a_i^t$ can be derived, which denotes the frequency of each neuron being activated during a forward pass given a task-specific dataset D_t.

Neuron Selection. We identify specialized neurons for each task t based on their activation frequency a^t. A subset of neurons S_k^t is progressively selected based on the highest a^t values until reaching a predefined threshold k, where

$$\sum_{i \in S_k^t} a_{(i)}^t >= k \sum_{i=1}^{d_{ff}} a_{(i)}^t \tag{1}$$

Here, the value $a_{(i)}^t$ is the frequency of the activation at dimension i, and $\sum_{i=1}^{d_{ff}} a_{(i)}^t$ is the total activation of all neurons for an FFN layer. k is a threshold factor, varying from 0% to 100%, indicating the extent of neuron activation deemed necessary for specialization. A lower k value results in higher sparsity in specialized neurons; $k = 0$ means no neuron will be involved, while $k = 100$ fully engages all neurons, the same as utilizing the full capacity of the original model. This dynamic approach emphasizes the collective significance of neuron activations up to a factor of k. In the end, we repeat these processes to obtain the specialized neurons of all FFN layers for each task.

3.2 Neuron Analysis on EC30

In this section, we describe how we identify specialized neurons on the EC30 dataset (Tan and Monz, 2023), where we train an MMT model covering all directions. EC30 is a multilingual translation benchmark that is carefully designed to consider diverse linguistic properties and real-world data distributions. It collects high to low-resource languages, resulting in 30 diverse languages from 5 language families, allowing us to connect our observations with linguistic properties easily. See Sections 5 for details on data and models.

3.2.1 Neuron Overlaps Reflect Language Proximity

We identified specialized neurons following Section 3.1, while setting the cumulative activation threshold k at 95%. This implies that the set of specialized neurons covers approximately 95% of the total activations. Intuitively, two similar tasks

[2] For activation functions like GeLU, we consider neurons inactive when their values are ≤ 0, as discussed in Section 6.2.

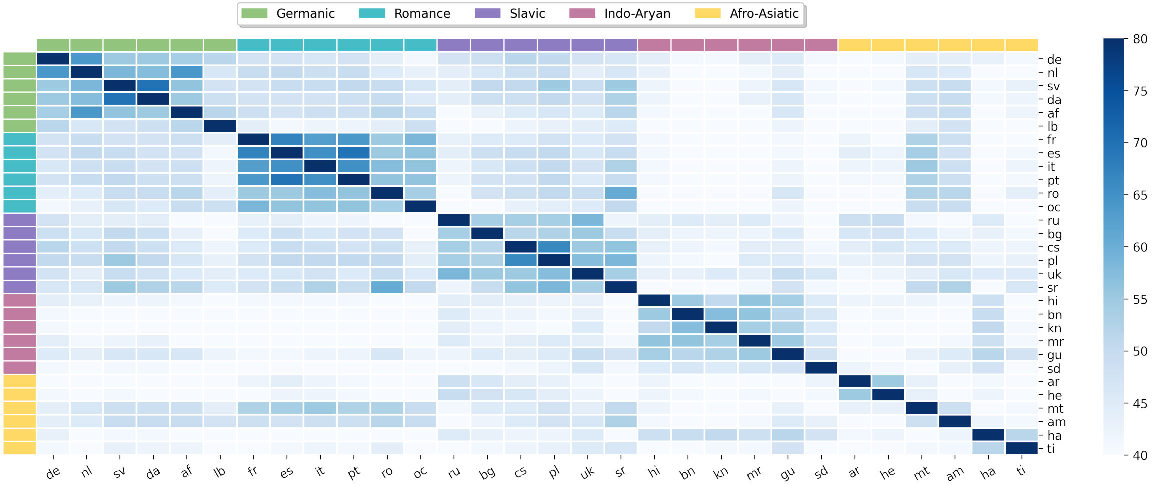

Figure 1: Pairwise Intersection over Union (IoU) scores for specialized neurons extracted from the first decoder FFN layer across all out-of-English translation directions to measure the degree of overlap. Darker cells indicate stronger overlaps, with the color threshold set from 40 to 80 to improve visibility.

should have a high overlap between their specialized neuron sets. Therefore, we examined the overlaps among specialized neurons across different tasks by calculating the Intersection over Union (IoU) scores: For task t_i and t_j, with specialized neurons denoted as sets S^i and S^j, their overlap is quantified by Eq. 2.

$$\mathrm{IoU}(S^i, S^j) = \frac{|S^i \cap S^j|}{|S^i \cup S^j|} \tag{2}$$

Figure 1 shows the IoU scores for specialized neurons across tasks in the first decoder layer. Figures of other layers are in A.9. We observe a structural separation of neuron overlaps, indicating a preference for language specificity. Notably, neuron overlap across language families is relatively low, a trend more pronounced in encoder layers (Figure 8). In addition, this structural distinction generally correlates with language proximity as indicated by the clustering pattern in Figure 1. This implies that target languages from the same family are more likely to activate similar neurons in the decoder, even when they use different writing systems, e.g., Arabic (ar) and Hebrew (he).

We also provide a phylogenetic tree analysis quantifying the correlation between neuron overlaps and linguistic distances in A.9. Moreover, we show that neuron overlaps show linguistic traits beyond family ties, exemplified by notable overlaps between Maltese (mt) and languages in the Romance family due to vocabulary borrowing.

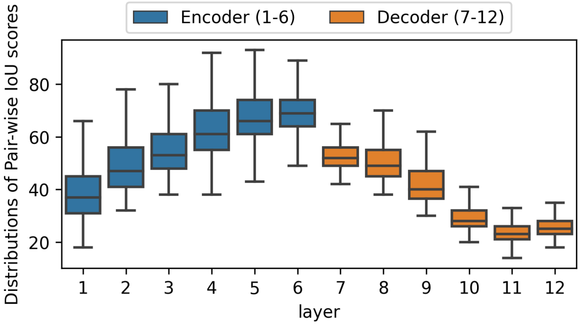

Figure 2: Progression of distribution of IoU scores for specialized neurons across layers on the EC30 dataset. The scores are measured for different source and target languages in the Encoder and Decoder, respectively.

3.2.2 The Progression of Neuron Overlaps

To analyze how specialized neuron overlaps across tasks evolve within the model, we visualize the IoU score distribution across layers in Figure 2. For each layer, we compute the pair-wise IoU scores between all possible tasks and then show them in a distribution. Overall, we observe that from shallow to deeper layers, structural distinctions intensify in the decoder (decreasing IoU scores) and weaken in the encoder (increasing IoU scores).

Furthermore, all neuron overlaps increase as we move up the encoder, regardless of whether these tasks are similar or not. This observation may suggest that the neurons in the encoder become more language-agnostic, as they attempt to map different scripts into semantic concepts. As for the Decoder, the model presents intensified modularity in terms of overlaps of specialized neurons. This can be seen by all overlaps becoming much smaller, indicating that neurons behave more separately.

Our findings align with the common assumption of the transformation process in Seq2Seq models. Similarly, Kudugunta et al. (2019) observed that multilingual embeddings gradually, though not perfectly, align within the encoder. However, our research diverges as it focuses on binary neuron activation patterns, rather than high-dimensional embeddings. Moreover, unlike them, we show that our findings can be leveraged to improve MMT.

4 Neuron Specialization Training

Our neuron structural analysis showed the presence of specialized neurons within the Feed-Forward Network (FFN) layers of a multilingual network. We hypothesize that continuously training the model, while leveraging these specialized neurons' intrinsic modular features, can further enhance task-specific performance. Building on this hypothesis, we propose *Neuron Specialization*, an approach that leverages specialized neurons to modularize the FFN layers in a task-specific manner.

4.1 Vanilla Feed-Forward Network

We first revisit the Feed-Forward Network (FFN) in Transformer (Vaswani et al., 2017). The FFN, crucial to our analysis, consists of two linear layers (fc1 and fc2) with an activation function $\sigma(\cdot)$. Specifically, the FFN block first processes the hidden state $H \in \mathbb{R}^{n\times d}$ (n denotes number of tokens in a batch) through fc1 layer $W_1 \in \mathbb{R}^{d\times d_{ff}}$. Then the output is passed to $\sigma(\cdot)$ and the fc2 layer W_2, as formalized in Eq 3, with bias terms omitted.

$$\mathrm{FFN}(H) = \sigma(HW_1)\, W_2. \tag{3}$$

4.2 Specializing Task-Specific FFN

Next, we investigate continuous training upon a subset of specialized parameters within FFN for each task. Given a pre-trained vanilla multilingual Transformer model with tags to identify the language pairs, e.g., Johnson et al. (2017), we can derive specialized neuron set S_k^t for each layer of a task[3] t and threshold k following the method outlined in Section 3.1. Then, we derive a boolean mask vector $m_k^t \in \{0,1\}^{d_{ff}}$ from S_k^t, where the i-th element in m_k^t is set to 1 only when $i \in S_k^t$, and apply it to control parameter updates. Specifically, we broadcast m_k^t and perform Hadamard Product with W_1 in each FFN layer as follows:

$$\mathit{FFN}(H) = \sigma(H(m_k^t \odot W_1))W_2. \tag{4}$$

m_k^t plays the role of controlling parameter update, where the boolean value of i-th element in m_k^t denotes if the i-th row of parameters in W_1 can be updated or not for each layer[4] during continues training. Broadly speaking, our approach selectively updates the first FFN (fc1) weights during back-propagation, tailoring the model more closely towards specific translation tasks and reinforcing neuron separation.

Note that while fc1 is selectively updated for specific tasks, other parameters are universally updated to maintain stability, and the same masking is applied to inference to ensure consistency. In addition, applying a mask to W_1 will nullify the contribution of the corresponding row in W_2 to the final output, thus, there is no need to apply task-specific masks to W_2, as the masking of sufficiently controls the influence on the output. Our pseudocode is in Appendix A.10.

Relevant studies like Xie et al. (2021), selectively pruning output hidden states for modules like attention and FFNs during training and inference. In contrast, we utilize sparse sub-networks (fc1 weights) since we found FFN neurons are already specialized.

5 Experimental Setup

We describe the experimental setups in this section. Note that we utilize the same training data for both pre-training and methods involving fine-tuning or continual training. More details of the datasets are in Appendix A.1.

5.1 Datasets

IWSLT. Following Lin et al. (2021), we constructed an IWSLT dataset with eight languages. We learned a 30k SentencePiece unigram (Kudo and Richardson, 2018) shared vocabulary and applied temperature sampling with $\tau = 2$. Note that we evaluate on Flores-200 (Costa-jussà et al., 2022) by merging *devtest* and *test*, as our test set.

EC30. We further validate our methods on EC30 dataset (Tan and Monz, 2023), which features 61 million parallel training sentences across 60

[3] We treat each translation direction as a distinct task.

[4] Note that m_k^t is layer-specified, we drop layer indexes hereon for simplicity of notation.

English-centric directions, representing five language families and various writing systems. We classify language pairs into low-resource (=100k), medium-resource (=1M), and high-resource (=5M) categories. We build a 128k size shared unigram vocabulary. Aligning with the original EC30 setups, we use Ntrex-128 (Federmann et al., 2022) as the validation set. Also, we use Flores-200 (merging *devtest* and *test*) as the test set for evaluation.

5.2 Systems

We compare our method with strong open-source baselines that share similar motivations in reducing interference for multilingual translation tasks.

mT-small. For IWSLT, we train an mT-small baseline model on Many-to-Many directions as per Lin et al. (2021): a 6-layer Transformer with 4 attention heads, $d = 512$, $d_{ff} = 1{,}024$.

mT-big For EC30, we train a mT-big baseline model on Many-to-Many directions following Wu and Monz (2023). It has 6 layers, with 16 attention heads, $d = 1{,}024$, and $d_{ff} = 4{,}096$.

Fine-Tune. We finetune baselines with the same routine as our Neuron Specialization Training.

Adapters. We employ two adapter methods: 1) Language Pair Adapter (**Adapter$_{LP}$**) and 2) Language Family Adapter (**Adapter$_{Fam}$**). We omit Adapter$_{Fam}$ for IWSLT due to its limited languages. Adapter$_{LP}$ inserts adapter modules based on language pairs, demonstrating strong effects in reducing interference while presenting no parameter sharing (Bapna and Firat, 2019). In contrast, Adapter$_{Fam}$ (Chronopoulou et al., 2023) facilitates parameter sharing across similar languages by training modules for each language family. Their bottleneck dimensions are 128 and 512 respectively. See Appendix A.2 for more training details.

LaSS. Lin et al. (2021) proposed LaSS to locate language-specific sub-networks following the lottery ticket hypothesis, i.e., finetuning all translation directions from a pre-trained model and then pruning based on magnitude. They then continually train the pre-trained model by only updating the sub-networks for each direction. We adopt the strongest LaSS configuration by applying sub-networks for both attention and FFNs.

Parameter Differentiation (PD). PD dynamically differentiates shared parameters into task-specialized ones to reduce interference (Wang and Zhang, 2022). To avoid over-differentiation, we set upper-bound (ub=100%) to limit the final model size (see more experiments in A.2.6). Due to the lack of multi-GPU support in the official implementation, we only employ PD in IWSLT experiments.

5.3 Implementation and Evaluation

To ensure fair comparisons, we use the fixed training routine for all compared methods, see detailed training and model specifications in Appendix A.2. For evaluation, We adopt the tokenized BLEU (Papineni et al., 2002) for the IWSLT and detokenized SacreBLEU[5] (Post, 2018) for the EC30. In addition, we report ChrF++ (Popović, 2017) and COMET (Rei et al., 2020) in Appendix A.6.

6 Results and Analyses

6.1 Small-Scale Results on IWSLT

We show results on IWSLT in Table 1. For Many-to-One (M2O) directions, our method receives an average +1.7 BLEU gain over the baseline, achieving the best performance among all approaches. The Adapter$_{LP}$, with a 67% increase in parameters over the baseline model, shows weaker improvements (+0.8) than our method. As for One-to-Many (O2M) directions, we observed weaker performance gains for all methods. While the gains are modest (averaging +0.3 BLEU), our method demonstrates consistent improvements across various languages in general. Finally, we show that fine-tuning the baseline with the same setting as our approach does not bring performance gains.

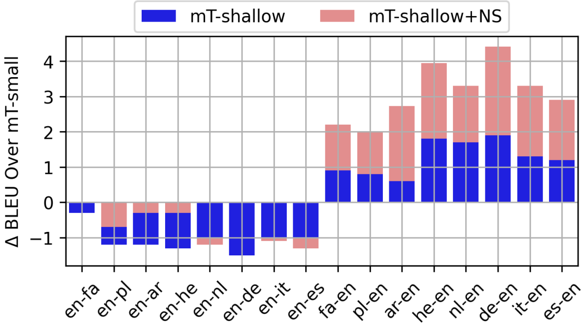

Figure 3: BLEU gains of shallower models over mT-small on IWSLT show improved X-En performance at the expense of En-X. Applying Neuron Specialization reduces EN-X degradation and amplifies X-En gains.

Scaling up does not always reduce interference. Shaham et al. (2023); Chang et al. (2023) have

[5] nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1

Language	$\Delta\theta$	Fa	Pl	Ar	He	Nl	De	It	Es	Avg
Size		89k	128k	139k	144k	153k	160k	167k	169k	
One-to-Many (O2M / En-X)										
mT-small	-	14.5	9.9	12.0	13.1	17.0	20.6	17.3	18.3	15.4
Fine-Tune	0%	+0.1	-0.2	+0.2	+0.4	-0.4	-0.1	-0.3	-0.5	-0.1
Adapter$_{LP}$	+67%	+0.1	-0.1	+0.4	**+1.4**	+0.2	+0.6	+0.1	+0.4	**+0.4**
LaSS	0%	-2.6	0	+0.6	+0.7	-0.2	+0.7	-0.2	-0.4	-0.2
Param-Diff	+100%	-0.8	0	**+1.0**	+1.1	**+0.2**	**+0.9**	**+0.4**	**+0.7**	**+0.4**
Ours	0%	**+0.7**	**+0.1**	+0.9	+0.6	+0.1	+0.1	+0.2	-0.3	+0.3
Many-to-One (M2O / X-En)										
mT-small	-	10.0	8.65	11.1	12.5	11.4	13.6	10.7	11.3	11.2
Fine-Tune	0%	+0.3	-0.2	+0.1	+0.8	+0.7	+0.3	-0.2	0	+0.2
Adapter$_{LP}$	+67%	+0.9	+0.6	+0.9	+1.0	+0.8	+1.0	+0.9	+0.3	+0.8
LaSS	0%	+1.2	+0.6	+0.9	+1.4	+1.1	+1.6	+1.6	+0.8	+1.2
Param-Diff	+100%	+1.7	**+1.3**	+1.3	+1.9	+1.6	**+2.2**	+1.7	+1.3	+1.6
Ours	0%	**+1.6**	+1.2	**+1.7**	**+2.0**	**+1.9**	+2.1	**+1.8**	**+1.4**	**+1.7**

Table 1: BLEU improvements over the baseline (mT-small) on IWSLT. $\Delta\theta$ denotes the relative parameter increase over the baseline, and 'Fine-Tune' signifies finetuning mT-small with the same setting as 'Ours'.

found scaling up the model capacity reduces interference, even under low-resource settings. We then investigate the trade-off between performance and model capacity by employing mT-shallow, a shallower version of mT-small with three fewer layers (with $\Delta\theta = -39\%$ for parameters, see Table 8 for details). Surprisingly, in Figure 3, we show that reducing parameters improved Many-to-One (X-En) performance but weakened One-to-Many (En-X) results. This result indicates that scaling up the model capacity does not always reduce interference, but may show overfitting to have performance degradation. Furthermore, we show that implementing Neuron Specialization with mT-shallow enhances X-En performance in all directions while lessening the decline in En-X translation quality.

6.2 Large-Scale Results on EC-30

Similar to what we observed in the small-scale setting, we find notable improvements when we scale up on the EC30 dataset. Table 2 shows consistent improvements across high-, medium-, and low-resource languages, with an average gain of +1.3 SacreBLEU over the baseline. LaSS, while effective in high-resource O2M pairs, presents limitations with negative impacts (-1.0 score) on low-resource languages, highlighting difficulties in subnetwork extraction for low-resource languages.

In contrast, our method achieves stable and consistent gains and passes statistical significance tests in A.4. The Adapter$_{LP}$, despite increasing parameters by 87% compared to the baseline, falls short of our method in boosting performance. Similar to experiments on IWSLT, we found fine-tuning the baseline on EC30 also brings worse/unchanged performance, suggesting the effectiveness of our method. Additionally, we show that applying Neuron Specialization in the encoder or decoder delivers similar gains, with both combined offering stronger performance.

Random Mask. We applied Neuron Specialization Training using random masks that masked 30% fc1 weights to validate the effectiveness of our method in locating task-specific neurons. Table 2 shows that such random strategy sacrifices performance, especially for low-resource languages.

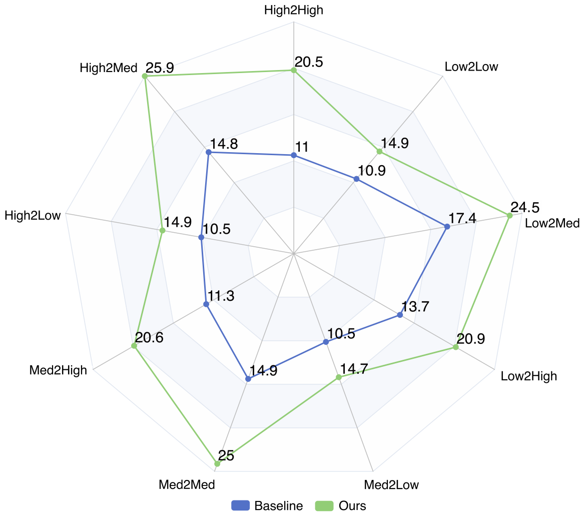

Figure 4: ChrF performance on 870 zero-shot directions across High, Medium, and Low-resource languages.

Methods	$\Delta\theta$	High (5M)			Med (1M)			Low (100K)			All (61M)		
		O2M	M2O	Avg	O2M	M2O	Avg	O2M	M2O	Avg	O2M	M2O	Avg
mT-big	-	28.1	31.6	29.9	29.7	31.6	30.6	18.9	26.0	22.4	25.5	29.7	27.7
Fine-Tune	0%	+0.3	+0.2	+0.3	+0.3	+0.2	+0.3	+0.1	-0.4	-0.2	+0.2	0	+0.1
Adapter$_{Fam}$	+70%	+0.7	+0.3	+0.5	+0.7	+0.3	+0.5	+1.1	+0.5	+0.8	+0.8	+0.4	+0.6
Adapter$_{LP}$	+87%	+1.6	+0.6	+1.1	+1.6	+0.4	+1.0	+0.4	+0.4	+0.4	+1.2	+0.5	+0.8
LaSS	0%	**+2.3**	+0.8	+1.5	**+1.7**	+0.2	+1.0	-0.1	-1.8	-1.0	+1.3	-0.3	+0.5
Random	0%	+0.9	-0.5	+0.2	+0.5	-0.7	-0.2	-0.3	-1.5	-0.9	+0.5	-0.9	-0.2
Ours^{Enc}	0%	+1.2	+1.1	+1.1	+1.0	+1.0	+1.0	+0.7	+0.8	+0.8	+1.0	+1.0	+1.0
Ours^{Dec}	0%	+1.2	+1.1	+1.1	+0.9	**+1.1**	+1.0	+0.7	**+1.1**	+0.9	+0.9	**+1.1**	+1.0
Ours	0%	+1.8	**+1.4**	**+1.6**	+1.4	**+1.1**	**+1.3**	**+1.4**	+0.9	**+1.2**	**+1.5**	**+1.1**	**+1.3**

Table 2: Average SacreBLEU improvements on the EC30 dataset over the baseline (mT-big), categorized by High, Medium, and Low-resource translation directions. 'Random' denotes continually updating the model with randomly selected task-specific neurons. 'Ours^{Enc}' and 'Ours^{Dec}' indicate Neuron Specialization applied solely to the Encoder and Decoder, respectively, while 'Ours' signifies the method applied to both components.

Zero-Shot Translation. We further evaluated our method on 870 zero-shot translation directions. For an unseen zero-shot direction (Src-Tgt), we construct its mask during inference using the Encoder mask from Source-to-English (Src-En) and the Decoder mask from English-to-Target (En-Tgt). We show an average gain of +7.4 and +3.1 on ChrF and SacreBLEU scores over the baseline (mT-big). Of these, 847 directions improved, while 23 have minor declines. We present the comparison in Figure 4, and more details can be found in A.7.

Wider and Deeper Models. We experiment with larger models by scaling up the width and depth (see details in A.5). Table 3 shows we achieve consistent performance gains, confirming the effectiveness of our approach for larger configurations.

Methods	SacreBLEU			COMET		
	Big	Wide	Deep	Big	Wide	Deep
Baseline	27.7	28.3	28.8	79.1	79.7	80.0
Ours	**29.0**	**29.4**	**29.7**	**80.0**	**80.5**	**80.7**

Table 3: Performance comparison between baseline models and our methods on three configurations.

The role of threshold factor. We explore the impact of our sole hyper-parameter k (neuron selection threshold factor) on performance. As mentioned in Section 3.1, a smaller k results in more sparse specialized neuron selection and fc1 weights. Figure 5 shows that our method delivers consistent and positive gains without extensive tuning. More explanations can be found in A.8.

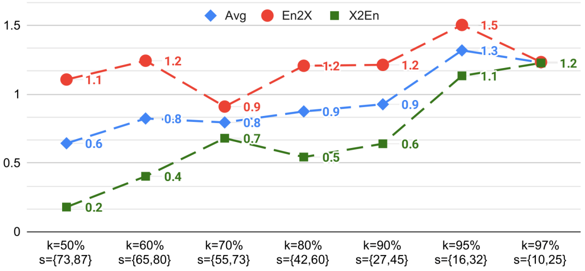

Figure 5: SacreBLEU gains of our method over the mT-large baseline on EC30. The x-axis represents the factor k and the dynamic sparsity of fc1 layers, with values ranging from minimum to maximum achieved sparsity.

Efficiency Comparisons. We compare efficiency across three aspects (Table 5). First, adding lightweight language pair adapters results in an +87% increase in trainable parameters over the baseline. Second, our method, which locates specialized neurons in just 5 minutes, is significantly faster than LaSS, which takes 33 hours with 4 Nvidia A6000 GPUs. Finally, regarding memory costs essential for handling multiple languages in deployment, our method is more economical, requiring only 1-bit masks for the FFN neurons.

Model	$\triangle\theta$	$\triangle T_{subnet}$	$\triangle$ Memory
Adapter$_{LP}$	+87%	n/a	1.42 GB
LaSS	0%	+33 hours	9.84 GB
Ours	0%	+5 minutes	3e-3 GB

Table 5: Efficiency comparison on EC30 dataset regarding extra trainable parameters ($\triangle\theta$: relative increase over the baseline), extra processing time for subnet extraction ($\triangle T_{subnet}$), and extra memory ($\triangle$ Memory).

Lang Size	De 5m	Es 5m	Cs 5m	Hi 5m	Ar 5m	Lb 100k	Ro 100k	Sr 100k	Gu 100k	Am 100k	High Avg	Low Avg
One-to-Many												
Bilingual	36.3	24.6	28.7	43.9	23.7	5.5	16.2	17.8	12.8	4.1	31.8	11.3
mT-big	-4.7	-1.5	-3.6	-4.4	-4.7	+9.0	+8.9	+6.2	+13.9	+3.1	-3.7	+8.2
Ours	-2.0	-0.2	-1.7	-2.4	-3.0	+10.8	+10.0	+8.2	+16.4	+3.7	-1.9	+9.8
Many-to-One												
Bilingual	39.1	24.5	32.6	35.5	30.8	8.7	19.5	21.3	7.0	8.7	32.7	13.0
mT-big	-1.5	+0.9	+0.2	-1.8	-2.3	+13.7	+11.9	+10.3	+18.2	+12.5	-1.1	+13.3
Ours	-0.3	+1.7	+1.8	-0.2	-0.3	+15.3	+12.4	+11.3	+19.6	+14.1	+0.3	+14.5

Table 4: SacreBLEU score comparisons for Multilingual baseline and Neuron Specialization models against Bilingual ones on the EC30 dataset, limited to 5 high- and low-resource languages due to computational constraints. Red signifies negative interference, Blue denotes positive synergy, with darker shades indicating better effects.

Neuron Specialization Beyond ReLU. We validate the adaptability of our method with the GeLU activation function, as it produces negative activation values. We first train an mT-big baseline model with GeLU, defining non-active neurons as ≤ 0, keeping all other settings unchanged. The results (Table 6) show that our method also works with GeLU, yielding consistent improvements (see details in Table 14). We leave exploring other thresholds for defining inactive FFN neurons to future work.

Methods	All (61M)		
	SacreBLEU	ChrF	Comet
mT-big^{relu}	27.7	52.2	79.1
Ours^{relu}	29.0	53.3	80.0
mT-big^{gelu}	27.9	52.3	79.2
Ours^{gelu}	28.9	53.2	80.1

Table 6: Performance comparison between the relu and gelu backbone models and our method.

6.3 The Impact of Reducing Interference

In this section, we measure to what extent our method mitigates interference and enhances knowledge transfer. Similar to Wang et al. (2020), we train bilingual models that do not contain interference or transfers, then compare results between bilingual models, the multilingual baseline model (mT-big), and our method (ours). We train Transformer-big and Transformer-based models for high- and low-resource tasks, see details in A.2.

In Table 4, we show that the multilingual model (mT-big) facilitates clear positive transfer for low-resource languages versus bilingual setups, leading to +8.2 (O2M) and +13.3 (M2O) score gains. However, training a unified multilingual model incurs negative interference, causing performance degradation for high-resource languages (averaged -3.7 and -1.1 drops).

In contrast, our Neuron Specialization method reduces interference for high-resource settings, leading to +1.8 and +1.4 SacreBLEU gains over mT-big in O2M and M2O directions. Moreover, our method enhances low-resource task performance with average gains of +1.6 (O2M) and +1.2 (M2O) SacreBLEU over the mT-big, demonstrating its ability to foster cross-lingual knowledge transfer.

7 Conclusions

In this paper, we have identified and leveraged *intrinsic task-specific modularity* within multilingual networks to mitigate interference. We showed that FFN neurons activate in a language-specific way, and they present structural overlaps that reflect language proximity, which progress across layers. We then introduced *Neuron Specialization Training* to leverage these natural modularity signals to structure the network, enhancing task specificity and improving knowledge transfer. Our experimental results, spanning various resource levels, show that our method consistently outperforms strong baseline systems, with additional analyses demonstrating reduced interference and increased knowledge transfer. Our work deepens the understanding of multilingual models by revealing their intrinsic modularity, offering insights into how multi-task models can be optimized without extensive modifications.

Limitations

This study primarily focuses on Multilingual Machine Translation, a key approach in multi-task learning, which serves as our primary testbed. However, the exploration of multi-task capabilities can extend beyond translation to a wider range of Natural Language Processing tasks. Recent studies have begun investigating the specific roles of FFN neurons in multilingual processing (Tang et al., 2024; Kojima et al., 2024), safety (Chen et al., 2024), and information aggregation (Voita et al., 2023) within Large Language Models. These areas remain unexplored in our research and present promising directions for future work.

Recent research has shown that MLP modules recall and store knowledge, while Attention modules are more likely to aggregate information (Meng et al., 2022; Geva et al., 2021; Elhage et al., 2021; Wang et al., 2023). In addition, extensive research, including Mixture-of-Experts and language-specific modules, studies and modifies FFN blocks. Therefore, our focus on FFN specialization aligns with these insights and complements existing research. We leave investigations of other Transformer components, such as the layer normalization modules, to future work.

Furthermore, we validated our method primarily on models using the ReLU activation function, though we also conducted experiments with GeLU, which showed smaller performance improvements. Exploring alternative thresholds for defining inactive FFN neurons (beyond simply ≤ 0) may lead to further gains, which we leave for future work.

Broader Impact

Recognizing the inherent risks of mistranslation in machine translation data, we have made efforts to prioritize the incorporation of high-quality data, such as two open-sourced Multilingual Machine Translation datasets: IWSLT and EC30. Additionally, issues of fairness emerge, meaning that the capacity to generate content may not be equitably distributed across different languages or demographic groups. This can lead to the perpetuation and amplification of existing societal prejudices, such as biases related to gender, embedded in the data.

Acknowledgments

This research was funded in part by the Netherlands Organization for Scientific Research (NWO) under project number VI.C.192.080. We would like to thank David Stap, Evgeniia Tokarchuk, Vlad Niculae, and Denis Paperno for their useful comments and discussions. We would also like to thank the reviewers for their feedback.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884.

Ali Araabi and Christof Monz. 2020. Optimizing transformer for low-resource neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3429–3435.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548.

Ziv Bar-Joseph, David K Gifford, and Tommi S Jaakkola. 2001. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17(Suppl. 1):S22–S22.

Tyler A Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K Bergen. 2023. When is multilinguality a curse? language modeling for 250 high-and low-resource languages. *arXiv preprint arXiv:2311.09205*.

Jianhui Chen, Xiaozhi Wang, Zijun Yao, Yushi Bai, Lei Hou, and Juanzi Li. 2024. Finding safety neurons in large language models. *arXiv preprint arXiv:2406.14144*.

Liang Chen, Shuming Ma, Dongdong Zhang, Furu Wei, and Baobao Chang. 2023. On the off-target problem of zero-shot multilingual neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9542–9558.

Rochelle Choenni, Dan Garrette, and Ekaterina Shutova. 2023a. Cross-lingual transfer with language-specific subnetworks for low-resource dependency parsing. *Computational Linguistics*, 49(3):613–641.

Rochelle Choenni, Ekaterina Shutova, and Dan Garrette. 2023b. Examining modularity in multilingual lms via language-specialized subnetworks. *arXiv preprint arXiv:2311.08273*.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2023. Language-family adapters for low-resource multilingual neural machine translation. In *Proceedings of the The Sixth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2023)*, pages 59–72.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Katharina Dobs, Julio Martinez, Alexander JE Kell, and Nancy Kanwisher. 2022. Brain-like functional specialization emerges spontaneously in deep neural networks. *Science advances*, 8(11):eabl8913.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *The Journal of Machine Learning Research*, 22(1):4839–4886.

Christian Federmann, Tom Kocmi, and Ying Xin. 2022. Ntrex-128–news test references for mt evaluation of 128 languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 21–24.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André FT Martins. 2022. Results of wmt22 metrics shared task: Stop using bleu–neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Dan He, Minh Quang Pham, Thanh-Le Ha, and Marco Turchi. 2023. Gradient-based gradual pruning for language-specific multilingual neural machine translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 654–670.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6912–6964.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual nmt representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model.

Xian Li and Hongyu Gong. 2021. Robust optimization for multilingual translation with imbalanced data. *Advances in Neural Information Processing Systems*, 34:25086–25099.

Baohao Liao, Yan Meng, and Christof Monz. 2023a. Parameter-efficient fine-tuning without introducing new latency. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4260, Toronto, Canada. Association for Computational Linguistics.

Baohao Liao, Shaomu Tan, and Christof Monz. 2023b. Make pre-trained model reversible: From parameter to memory efficient fine-tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. Learning language specific sub-network for multilingual machine translation. 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 293–305.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Daniel Müllner. 2011. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. 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 244–258.

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*, pages 311–318.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Ponti. 2023. Modular deep learning. *Transactions on Machine Learning Research*. Survey Certification.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

Telmo Pires, Robin Schmidt, Yi-Hsiu Liao, and Stephan Peitz. 2023. Learning language-specific layers for multilingual machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14767–14783.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. 2022. Comet-22: Unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

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)*, pages 2685–2702.

Stefan Riezler and John T Maxwell III. 2005. On some pitfalls in automatic evaluation and significance testing for mt. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 57–64.

Uri Shaham, Maha Elbayad, Vedanuj Goswami, Omer Levy, and Shruti Bhosale. 2023. Causes and cures for interference in multilingual translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada. Association for Computational Linguistics.

Shaomu Tan and Christof Monz. 2023. Towards a better understanding of variations in zero-shot neural machine translation performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13553–13568.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5701–5715, Bangkok, Thailand. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855.

Qian Wang and Jiajun Zhang. 2022. Parameter differentiation based multilingual neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11440–11448.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. 2020. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*.

Di Wu and Christof Monz. 2023. Beyond shared vocabulary: Increasing representational word similarities across languages for multilingual machine translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore. Association for Computational Linguistics.

Di Wu, Shaomu Tan, Yan Meng, David Stap, and Christof Monz. 2024. How far can 100 samples go? unlocking overall zero-shot multilingual translation via tiny multi-parallel data. *arXiv preprint arXiv:2401.12413*.

Di Wu, Shaomu Tan, David Stap, Ali Araabi, and Christof Monz. 2023. Uva-mt's participation in the wmt 2023 general translation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 175–180.

Wanying Xie, Yang Feng, Shuhao Gu, and Dong Yu. 2021. Importance-based neuron allocation for multilingual neural machine translation. 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 5725–5737.

Derrick Xin, Behrooz Ghorbani, Justin Gilmer, Ankush Garg, and Orhan Firat. 2022. Do current multi-task optimization methods in deep learning even help? *Advances in neural information processing systems*, 35:13597–13609.

Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. 2019. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2020a. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020b. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639.

Zhengyan Zhang, Zhiyuan Zeng, Yankai Lin, Chaojun Xiao, Xiaozhi Wang, Xu Han, Zhiyuan Liu, Ruobing Xie, Maosong Sun, and Jie Zhou. 2023. Emergent modularity in pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4066–4083, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Dataset details

Due to the difficulties of mining non-English-centric Translation data, recent research (Johnson et al., 2017; Zhang et al., 2020b,a; Tan and Monz, 2023; Wu and Monz, 2023; Shaham et al., 2023; Pires et al., 2023) has increasingly focused on utilizing English-centric datasets to explore Multilingual Neural Machine Translation (MNMT). Furthermore, Fan et al. (2021) have observed that training in M2M settings does not necessarily enhance performance in supervised directions. Therefore, our approach prioritizes English-centric datasets to remain computationally feasible while still providing valuable insights into MNMT dynamics.

IWSLT We collect and pre-processes the IWSLT-14 dataset following Lin et al. (2021). We refer readers to Lin et al. (2021) for more details.

EC30 We utilize the EC30, a subset of the EC40 dataset (Tan and Monz, 2023) (with 10 extremely low-resource languages removed in our experiments) as our main dataset for most experiments and analyses. We list the Languages with their ISO and scripts in Table 7, along with their number of sentences.

In general, EC30 is an English-centric Multilingual Machine Translation dataset containing 61 million sentences covering 30 languages (excluding English). It collected data from 5 representative language families with multiple writing scripts. In addition, EC30 is well balanced at each resource level, for example, for all high-resource languages, the number of training sentences is 5 million. Note that the EC30 is already pre-processed and tokenized (with Moses tokenizer), thus we directly use it for our study.

A.2 Model and Training Details

We list the configurations and hyper-parameter settings of all systems for the main training setting

	Germanic			Romance			Slavic			Indo-Aryan			Afro-Asiatic		
	ISO	Language	Script	ISO	Language	Script	ISO	Language	Script	ISO	Language	Script	ISO	Language	Script
High	de	German	Latin	fr	French	Latin	ru	Russian	Cyrillic	hi	Hindi	Devanagari	ar	Arabic	Arabic
(5m)	nl	Dutch	Latin	es	Spanish	Latin	cs	Czech	Latin	bn	Bengali	Bengali	he	Hebrew	Hebrew
Med	sv	Swedish	Latin	it	Italian	Latin	pl	Polish	Latin	kn	Kannada	Devanagari	mt	Maltese	Latin
(1m)	da	Danish	Latin	pt	Portuguese	Latin	bg	Bulgarian	Cyrillic	mr	Marathi	Devanagari	ha	Hausa*	Latin
Low	af	Afrikaans	Latin	ro	Romanian	Latin	uk	Ukrainian	Cyrillic	sd	Sindhi	Arabic	ti	Tigrinya	Ethiopic
(100k)	lb	Luxembourgish	Latin	oc	Occitan	Latin	sr	Serbian	Latin	gu	Gujarati	Devanagari	am	Amharic	Ethiopic

Table 7: Details of EC30 Training Dataset. Numbers in the table represent the number of sentences, for example, 5m denotes exactly 5,000,000 number of sentences. The only exception is Hausa, where its size is 334k (334,000).

Models	Dataset	Num. trainable params	Num. Layer	Num. Attn Head	dim	d_{ff}	max tokens	update freq	dropout
mT-shallow	IWSLT	47M	3	8	512	1,024	2,560	4	0.1
Ours-shallow	IWSLT	47M	3	8	512	1,024	2,560	4	0.1
PD-shallow (ub=51%)	IWSLT	71M	3	8	512	1,024	2,560	4	0.1
PD-shallow (ub=None)	IWSLT	118M	3	8	512	1,024	2,560	4	0.1
mT-small	IWSLT	76M	6	8	512	1,024	2,560	4	0.1
LaSS-small	IWSLT	76M	6	8	512	1,024	2,560	4	0.1
Ours-small	IWSLT	76M	6	8	512	1,024	2,560	4	0.1
bilingual-low	EC30	52M	6	2	512	1,024	2,560	1	0.3
bilingual-high	EC30	439M	6	16	1,024	4096	2,560	10	0.1
mT-big	EC30	439M	6	16	1,024	4,096	7,680	21	0.1
LaSS-big	EC30	439M	6	16	1,024	4,096	7,680	21	0.1
Ours-big	EC30	439M	6	16	1,024	4,096	7,680	21	0.1
mT-wide	EC30	540M	6	16	1,024	8,192	7,680	21	0.1
Ours-wide	EC30	540M	6	16	1,024	8,192	7,680	21	0.1
mT-large	EC30	615M	12	16	1,024	4,096	7,680	21	0.1
Ours-large	EC30	615M	12	16	1,024	4,096	7,680	21	0.1

Table 8: Configuration and hyper-parameter settings for all models in this paper. Num. Layer and Attn Head denote the number of layers and attention heads, respectively. dim represents the dimension of the Transformer model, d_{ff} means the dimension of the feed-forward layer. bilingual-low and -high represent the bilingual models for low and high-resource languages.

(EC30) in Table 8. To maintain consistency and comparability across all experiments, we employed the same early stopping settings rather than fixing the training duration for all experiments. We use 4 NVIDIA A6000 (48G) GPUs to conduct most experiments and implement them based on Fairseq (Ott et al., 2019) with FP16. Lastly, we utilize the same training data for both pre-training and methods involving fine-tuning or continual training.

A.2.1 Global training settings

For all systems on both datasets, we adopt the pre-norm and share the decoder input output embedding. In addition, we use the Adam optimizer ($\beta 1 = 0.9$, $\beta 2 = 0.98$, $\epsilon = 10^{-9}$) with 5e-4 learning rate and 4k warmup steps in all methods. Furthermore, we use cross entropy with label smoothing to avoid overfitting (smoothing factor=0.1) and set early stopping to 20. Similar to Fan et al. (2021), we prepend language tags to the source and target sentences to indicate the translation directions for all multilingual translation systems.

More importantly, we applied the same fixed routine across all experiments to ensure a fair comparison among all multilingual systems. Other global settings are the same for all systems to make fair comparisons, such as learning rate, warm-up steps, and batch size.

A.2.2 Bilingual models

We train bilingual models in section 6.3 to study how much our method can reduce interference and foster knowledge transfer. For bilingual models of low-resource languages, we adopt the suggested hyper-parameter settings from Araabi and Monz (2020), such as $d_{ff} = 512$, number of attention head as 2, and dropout as 0.3. Furthermore, We train separate dictionaries for low-resource bilingual models to avoid potential overfitting instead of using the large 128k shared multilingual dictionary.

For bilingual models of high-resource languages, we adopt the 128k shared multilingual dictionary and train models with the Transformer-big architecture as the multilingual baseline (mT-big). The detailed configurations can be found in Table 8.

A.2.3 Language Pair Adapters

We implement Language Pair Adapters (Bapna and Firat, 2019) by ourselves based on Fairseq. The Language Pair Adapter is learned depending on each pair, e.g., we learn two modules for en-de, namely an english adapter on the Encoder side and a German adapter on the Decoder side. Note that, except for the unified pre-trained model, language pair adapters do not share any parameters with each other, preventing potential knowledge transfers. We set its bottleneck dimension as 128 for all experiments of IWSLT and EC30.

IWSLT. For the IWSLT dataset that contains 8 languages with 16 translation directions, the mT-small base model size is 76M. Adapter$_{LP}$ insert 3.2M extra trainable parameters for one direction, thus resulting in 51.2M added parameters for all, leading to 67% relative parameter increase over the baseline model.

EC30. For the EC30 dataset that contains 30 languages with 60 translation directions, the mT-big base model size is 439M. Adapter$_{LP}$ inserts 6.4M extra trainable parameters for one direction, thus resulting in 384M added parameters for all directions, leading to 87% relative parameter increase over the baseline model. When training Adapter$_{LP}$ for low-resource languages, we increased dropout (0.1 -> 0.3) and decreased batch size (max-token: 7680 -> 2560) to avoid overfitting as suggested by Bapna and Firat (2019).

A.2.4 Language Family Adapters

The Language Family Adapter (Chronopoulou et al., 2023) is learned depending on each language family, e.g., for all 6 Germanic languages in the EC30, we learn two modules for en-Germanic, namely the en adapter on the Encoder side and the Germanic adapter on the Decoder side. We set its bottleneck dimension as 512 for all experiments for the EC30.

EC30. For the EC30 dataset that contains 30 languages with 60 translation directions, the mT-big base model size is 439M. Adapter$_{Fam}$ insert 25.3M additional trainable parameters for one family (on EN-X directions), thus resulting in 303.6M added parameters for all families on both EN-X and X-En directions, leading to 69% relative parameter increase over the baseline model.

A.2.5 LaSS

When reproducing LaSS (Lin et al., 2021), we adopt the code from their official Github page[6] with the same hyper-parameter setting as they suggested in their paper. For IWSLT, we finetune the mT-small for each translation direction with dropout=0.3, and we set dropout=0.1 for large-scale EC30. We then identify the language-specific parameters for attention and feed-forward modules (the setting with the strongest gains in their paper) with a pruning rate of 70%. We continue to train the sparse networks while keeping the same setting as the pre-training phase as they suggested.

Note that we observed different results as they reported in the paper, even though we used the same code, hyper-parameter settings, and corresponding Python environment and package version. We also found that He et al. (2023) reproduced LaSS results in their paper, which shows similar improvements (around +0.6 BLUE gains) over the baseline of our reproductions. As for an improved method over LaSS proposed by He et al. (2023), we do not reproduce since no open-source code has been released.

A.2.6 Parameter Differentiation

Wang and Zhang (2022) suggest that parameters with conflicting inter-task gradients might lead to optimization conflicts among tasks. To alleviate such parameter interference, they propose PD that dynamically differentiates shared parameters into task-specialized ones. PD includes a crucial hyper-parameter: the differentiation upper bound (ub). This parameter limits the model's size relative to the original model size. For example, ub = 1.5 restricts the model size to 150% of the original, while ub = None allows the model to grow without restriction, which can result in bilingual models with unlimited parameter differentiation, i.e., each

[6] https://github.com/NLP-Playground/LaSS

parameter is only shared by one task in the final model. We report two configurations of the Parameter Differentiation in Table 9.

A.3 Comparisons with M2M-100 Models

We choose multilingual Transformer architecture as our baseline backbone, which has been commonly used as a strong baseline in many MNMT studies (Pires et al., 2023; Shaham et al., 2023; Arivazhagan et al., 2019; Wu et al., 2024), and is widely recognized as a strong baseline within the community (Chen et al., 2023; Wu et al., 2023; Pan et al., 2021; Wu and Monz, 2023).

We further establish the strength of our baseline models by comparing them to the M2M-100 models, which are state-of-the-art systems trained on an extensive corpus of 7.5 billion parallel sentences. In specific, we directly evaluated the trained M2M-100 models provided in Fairseq [7]. The results, presented in Table 10, demonstrate that both our baseline model (mT-big) and our proposed method (Ours) achieve performance that is comparable to, or even surpasses, the M2M-100 models.

A.4 Robustness tests

To show that the improvements of our method are not due to random variance, we implemented our method with different random seeds for the below experiments and conducted statistical significance tests for our main EC30 results.

A.4.1 Testing with Different Random Seeds

We run our method with different seeds and show robust performance gains for both IWSLT and EC30 datasets (see Table 12 and Table 13).

Seed	O2M	M2O
ΔBLEU over mT-shallow		
seed=222	+0.3	+1.8
seed=111	+0.3	+1.4
ΔBLEU over mT-small		
seed=222	+0.3	+1.7
seed=111	+0.6	+1.2

Table 12: Average BLEU improvements of our Neuron Specialization method (Ours) over baselines (mT-shallow and mT-small) on the IWSLT dataset.

Seed	O2M	M2O	M2M
ΔSacreBLEU over mT-big			
seed=222	+1.5	+1.1	+1.3
seed=111	+1.3	+1.1	+1.2
seed=42	+1.4	+1.2	+1.3

Table 13: Average SacreBLEU improvements of our Neuron Specialization method (Ours) over the baseline (mT-big) on the EC30 dataset.

A.4.2 Statistical Significance Test

We conducted Paired approximate randomization (Riezler and Maxwell III, 2005) paired significance test to show that the improvements of our method over the baseline (mT-big) on EC30 are statistically significant regarding SacreBleu and ChrF++ metrics. In sum, for both metrics, 59/60 directions passed the test (p-value < 0.05) except en-ha. The test is performed with the SacreBleu Python package's paired significance testing feature (–paired-ar).

A.5 Experiments on wider and deeper models

We conducted further experiments to determine if our method retains its effectiveness with larger models. We expanded the baseline model, mT-big, in two key dimensions: a) the feed-forward network (FFN) size, indicating the 'width' of the network; b) the number of layers, representing the 'depth' of the network. Specifically, we introduced mT-wide, which features an expanded FFN dimensionality (from 4,096 to 8,192), and mT-large, which has increased layer count (from 6-6 to 12-12). See model config details in Table 8.

Following these modifications, we applied our neuron specialization approach to these models. The results, as shown in Table 11, demonstrate consistent performance gains across both configurations, further validating the efficacy of our method.

A.6 EC30 result using ChrF++ and COMET

Recent studies (Rei et al., 2020; Costa-jussà et al., 2022) show that ChrF and COMET present high levels of correlation with human judgments, and automatic metrics based on pre-trained embeddings can outperform human crowd workers (Freitag et al., 2021). Notably, Costa-jussà et al. (2022) found an increase of +0.5 in ChrF++ has been correlated with statistically significant improvements in human evaluations, with a change of +1.0 in

[7] `https://github.com/facebookresearch/fairseq/tree/main/examples/m2m_100`

Language Size	$\Delta\theta$	Fa 89k	Pl 128k	Ar 139k	He 144k	Nl 153k	De 160k	It 167k	Es 169k	Avg
One-to-Many (O2M / En-X)										
mT-shallow	-	14.3	8.7	10.8	11.8	16.0	19.1	16.3	17.3	14.3
Param-Diff	+51%	-0.1	+0.5	+0.2	+0.4	+0.3	+0.1	+0.3	+0.2	+0.2
Param-Diff	+252%	+0.8	+0.8	+0.6	+1.4	+0.5	+0.9	+0.6	+0.3	+0.6
Ours	0%	+0.2	+0.6	+1.0	+1.1	-0.1	0	-0.1	-0.3	+0.3
Many-to-One (M2O / X-En)										
mT-shallow	-	10.9	9.4	11.7	14.3	13.1	15.5	11.9	12.5	12.4
Param-Diff	+51%	+0.3	+0.6	+0.9	+0.8	+0.8	+1.8	+1.2	+1.0	+0.9
Param-Diff	+252%	+1.7	+1.7	+1.5	+1.3	+2.1	+2.9	+2.5	+1.5	+1.9
Ours	0%	+1.3	+1.3	+2.1	+2.1	+1.6	+2.5	+2.0	+1.7	+1.8

Table 9: Experiment results (BLEU) using **mT-shallow** backbone models on IWSLT. $\Delta\theta$ denotes the relative parameter increase over the mT-shallow. For Parameter Differentiation, we run the differentiation upper bound (ub) by ub=0.51, ub=1.51, and ub=None, limiting the model size to 51%, 151% of the original one and allowing the model to grow without restriction.

Methods	θ	High (5M)			Med (1M)			Low (100K)		
		O2M	M2O	Avg	O2M	M2O	Avg	O2M	M2O	Avg
mT-big	438m	27.7	32.0	29.9	30.6	34.2	32.4	26.9	32.9	29.9
M2M-100	418m	23.3	28.0	25.7	30.8	32.9	31.9	24.6	32.0	28.3
M2M-100	1.2b	28.3	34.3	31.3	36.3	38.9	37.6	31.7	41.1	36.4
Ours-big	438m	29.6	33.3	31.5	32.0	35.5	33.8	28.1	33.7	30.9

Table 10: Performance comparisons on the EC30 test set using SacreBLEU. θ represents the number of parameters, and 'Ours-big' denotes our neuron specialization method applied to the mT-big. We excluded directions where the M2M-100 models scored <=10 BLEU to ensure fair comparisons, resulting in 51 translation directions.

ChrF++ almost always perceptible to human evaluators, which is studied on the FLORES test set.

To ensure a comprehensive evaluation, we report various automatic metrics in this paper: ChrF++(character level), SacreBleu (detokenized word level), and COMET(representation level) scores as extra results, as shown in Table 14, respectively. We opted for the "wmt22-comet-da" model (Rei et al., 2022), a widely used version from Unbabel's collection of models that serves as the default choice. This model presents SOTA performance in WMT Metrics Shared Task (Freitag et al., 2022). Similar to what we observed in Section 6.2, our Neuron Specialization presents consistent performance improvements over the baseline model while outperforming other methods such as LaSS and Adapters.

Our method, applied to the same FLORES-200 test set, outperformed the baseline with an average increase of +1.1 ChrF++ scores, where most gains were greater than +1.0 ChrF++. This improvement emphasizes the effectiveness of our approach, suggesting a significant alignment with human evaluative standards.

A.7 Reults in Zero-shot translations

Zero-shot neural machine translation (ZS-NMT) represents a pivotal challenge in multilingual machine translation, aiming to handle language pairs never seen during training. Although training unified MMT systems enables zero-shot translations(Johnson et al., 2017), their performance falls short of that seen in supervised directions. Recent findings by Zhang et al. (2020b) suggest that larger model sizes enhance ZS performance. Additionally, Tan and Monz (2023) indicates that vocabulary overlap and linguistic similarities contribute to variations in ZS performance, and that stronger En-centric capabilities might improve ZS results.

ZS-NMT Setups To further investigate whether our method could bring benefits to zero-shot translations, we tested our method across 870 zero-shot directions involving 30 languages. To do that, we created masks using the Encoder mask from

Methods	$\Delta\theta$	High (5M)			Med (1M)			Low (100K)			All (61M)		
		O2M	M2O	Avg	O2M	M2O	Avg	O2M	M2O	Avg	O2M	M2O	Avg
						SacreBLEU							
mT-big	-	28.1	31.6	29.9	29.7	31.6	30.6	18.9	26.0	22.4	25.5	29.7	27.7
Ours-big	0%	+1.8	+1.4	+1.6	+1.4	+1.1	+1.3	+1.4	+0.9	+1.2	+1.5	+1.1	+1.3
mT-wide	+23%	+0.8	+0.6	+0.7	+0.7	+0.6	+0.6	+0.6	+0.6	+0.6	+0.6	+0.6	+0.6
Ours-wide	+23%	+2.2	+1.9	+2.1	+1.8	+1.7	+1.8	+1.4	+1.1	+1.3	+1.8	+1.5	+1.7
mT-large	+40%	+1.2	+1.2	+1.2	+1.0	+1.4	+1.2	+0.8	+1.6	+1.2	+1.0	+1.2	+1.1
Ours-large	+40%	+2.6	+2.3	+2.5	+1.9	+2.0	+2.0	+1.4	+2.2	+1.8	+2.0	+2.1	+2.0
						ChrF++							
mT-big	-	52.4	57.6	55.0	54.0	56.6	55.3	42.5	50.0	46.3	49.6	54.7	52.2
Ours-big	0%	+1.4	+1.1	+1.3	+1.1	+0.9	+1.0	+1.2	+0.8	+1.0	+1.2	+0.9	+1.1
mT-wide	+23%	+0.7	+0.7	+0.7	+0.7	+0.6	+0.7	+0.6	+0.7	+0.7	+0.7	+0.6	+0.7
Ours-wide	+23%	+1.8	+1.6	+1.7	+1.5	+1.4	+1.5	+1.3	+1.0	+1.2	+1.6	+1.3	+1.4
mT-large	+40%	+0.9	+0.9	+0.9	+0.9	+1.1	+1.0	+0.8	+1.4	+1.1	+0.9	+1.1	+1.0
Ours-large	+40%	+2.0	+1.8	+1.9	+1.5	+1.7	+1.6	+1.3	+1.8	+1.6	+1.6	+1.8	+1.7
						COMET							
mT-big	-	82.4	83.9	83.2	81.1	80.1	80.6	73.8	73.4	73.6	79.1	79.1	79.1
Ours-big	0%	+1.4	+1.0	+1.2	+0.9	+0.7	+0.8	+0.8	+0.7	+0.8	+1.0	+0.8	+0.9
mT-wide	+23%	+0.8	+0.6	+0.7	+0.6	+0.6	+0.6	+0.6	+0.6	+0.6	+0.7	+0.6	+0.6
Ours-wide	+23%	+1.8	+1.4	+1.6	+1.3	+1.3	+1.3	+1.3	+1.2	+1.3	+1.5	+1.3	+1.4
mT-large	+40%	+1.0	+0.8	+0.9	+0.7	+1.0	+0.9	+0.9	+1.2	+1.1	+0.9	+1.0	+0.9
Ours-large	+40%	+2.1	+1.6	+1.9	+1.3	+1.6	+1.5	+1.3	+1.9	+1.6	+1.6	+1.7	+1.6

Table 11: The effectiveness of our method on different model configurations. The table shows the averaged improvements on the EC30 dataset over the baseline (mT-big). 'Ours-big', 'Ours-wide', and 'Ours-large' indicate Neuron Specialization applied to the mT-big, mT-wide, and mT-large baselines respectively.

Source-to-English (Src-En) and the Decoder mask from English-to-Target (En-Tgt).

ZS-NMT Results Overall, we observed an averaged +3.1 SacreBLEU improvement on zero-shot directions, with 847 out of 870 directions showing improvements, and 23 directions experiencing minor declines, averaging -0.3 SacreBLEU. Detailed results for high, medium, and low-resource languages (denoted as H, M, and L) are presented in Table 15, along with comparisons of directions achieving baseline scores of 5 and 10 SacreBLEU using both a baseline model (mT-big) and our method are shown in Table 16.

Model	H2H	H2M	H2L	M2H	M2M	M2L	L2H	L2M	L2L
mT-big	1.5	2.2	1.3	1.8	2.4	1.3	2.6	3.1	1.3
Ours	+4.2	+4.7	+1.6	+4.1	+4.3	+1.5	+2.7	+2.8	+1.2

Table 15: SacreBLEU improvements of Neuron Specialization method (Ours) over the mT-big baseline on zero-shot translations.

Model	Num. ≥ 5	Num. ≥ 10
mT-big	37	2
Ours	381	95

Table 16: Number of directions that exceed 5 and 10 SacreBLEU scores for the baseline (mT-big) and our method (Ours).

A.8 Sparsity versus Performance

For the Neuron Specialization, we dynamically select specialized neurons via a cumulative activation threshold k in Equation 1, which is the only hyper-parameter of our method. Here, we discuss the impact of k on the final performance and its relationship to the sparsity. As mentioned in Section 3.1, a smaller factor k results in more sparse specialized neuron selection, which makes the fc1 weight more sparse as well in the Neuron Specialization Training process. In Figure 5, we show that our method consistently outperforms the baseline across a range of k values, from 50 to 97. This

Methods	$\Delta\theta$	High (5M)			Med (1M)			Low (100K)			All (61M)		
		O2M	M2O	Avg	O2M	M2O	Avg	O2M	M2O	Avg	O2M	M2O	Avg
SacreBLEU													
mT-big	-	28.1	31.6	29.9	29.7	31.6	30.6	18.9	26.0	22.4	25.5	29.7	27.7
Fine-Tune	0%	+0.3	+0.2	+0.3	+0.3	+0.2	+0.3	-0.3	-0.4	-0.4	+0.3	0	+0.1
Adapter$_{Fam}$	+70%	+0.7	+0.3	+0.5	+0.7	+0.3	+0.5	+1.1	+0.5	+0.8	+0.8	+0.4	+0.6
Adapter$_{LP}$	+87%	+1.6	+0.6	+1.1	+1.6	+0.4	+1.0	+0.4	+0.4	+0.4	+1.2	+0.5	+0.8
LaSS	0%	**+2.3**	+0.8	+1.5	**+1.7**	+0.2	+1.0	-0.1	-1.8	-1.0	+1.3	-0.3	+0.5
Random	0%	+0.9	-0.5	+0.2	+0.5	-0.7	-0.2	-0.3	-1.5	-0.9	+0.5	-0.9	-0.2
Ours^{Enc}	0%	+1.2	+1.1	+1.1	+1.0	+1.0	+1.0	+0.7	+0.8	+0.8	+1.0	+1.0	+1.0
Ours^{Dec}	0%	+1.2	+1.1	+1.1	+0.9	**+1.1**	+1.0	+0.7	**+1.1**	+0.9	+0.9	**+1.1**	+1.0
Ours	0%	+1.8	**+1.4**	**+1.6**	+1.4	**+1.1**	**+1.3**	**+1.4**	+0.9	**+1.2**	**+1.5**	**+1.1**	**+1.3**
mT-big^{gelu}	0%	+0.1	+0.2	+0.2	+0.1	+0.2	+0.2	+0.1	+0.2	+0.2	+0.1	+0.2	+0.2
Ours^{gelu}	0%	+1.5	**+1.4**	+1.5	+1.3	**+1.1**	+1.2	+1.1	+0.8	+1.0	+1.3	**+1.1**	+1.2
ChrF++													
mT-big	-	52.4	57.6	55.0	54.0	56.6	55.3	42.5	50.0	46.3	49.6	54.7	52.2
Adapter$_{LP}$	+87%	+1.3	+0.2	+0.8	+1.1	+0.1	+0.6	+0.3	+0.3	+0.3	+0.9	+0.2	+0.5
Adapter$_{Fam}$	+70%	+0.6	+0.2	+0.4	+0.7	+0.3	+0.5	+1.1	+0.4	+0.8	+0.8	+0.3	+0.5
LaSS	0%	**+1.7**	+0.8	+1.2	**+1.3**	+0.3	+0.8	-0.3	-1.5	-0.9	+0.9	-0.2	+0.5
Random	0%	+0.7	-0.4	+0.2	+0.4	-0.5	-0.1	-0.5	-1.2	-0.9	+0.2	-0.7	-0.3
Ours^{Enc}	0%	+1.0	+0.9	+1.0	+0.7	+0.9	+0.8	+0.6	+0.9	+0.8	+0.8	+0.9	+0.8
Ours^{Dec}	0%	+0.9	+0.9	+0.9	+0.6	**+1.0**	+0.8	+0.5	**+1.2**	+0.9	+0.7	**+1.0**	+0.9
Ours	0%	+1.4	**+1.1**	**+1.3**	+1.1	+0.9	**+1.0**	**+1.2**	+0.8	**+1.0**	**+1.2**	+0.9	**+1.1**
mT-big^{gelu}	0%	+0.1	+0.1	+0.1	0	+0.1	+0.1	+0.1	+0.2	+0.2	+0.1	+0.1	+0.1
Ours^{gelu}	0%	+1.2	+1.0	+1.1	+1.0	**+1.0**	**+1.0**	+1.0	+0.6	+0.8	+1.1	+0.9	+1.0
COMET													
mT-big	-	82.4	83.9	83.2	81.1	80.1	80.6	73.8	73.4	73.6	79.1	79.1	79.1
Adapter$_{LP}$	+87%	+0.9	+0.2	+0.5	+0.6	+0.2	+0.4	0	+0.1	0	+0.5	+0.2	+0.4
Adapter$_{Fam}$	+70%	+0.4	+0.1	+0.3	+0.4	+0.2	+0.3	+0.7	+0.3	+0.5	+0.5	+0.2	+0.4
LaSS	0%	**+1.5**	+0.8	**+1.2**	**+0.9**	+0.6	+0.8	-0.2	-1.0	-0.6	+0.7	+0.1	+0.4
Random	0%	+0.2	-0.1	+0.1	-0.1	-0.2	-0.2	-0.8	-0.9	-0.9	-0.2	-0.4	-0.3
Ours^{Enc}	0%	+1.0	+0.8	+0.9	+0.5	+0.9	+0.7	+0.3	**+0.9**	+0.6	+0.6	+0.8	+0.7
Ours^{Dec}	0%	+0.9	+0.8	+0.9	+0.5	+1.0	+0.8	+0.3	**+0.9**	+0.6	+0.6	**+1.0**	+0.8
Ours	0%	+1.4	**+1.0**	**+1.2**	**+0.9**	+0.7	+0.8	+0.8	+0.7	**+0.8**	**+1.0**	+0.8	+0.9
mT-big^{gelu}	0%	+0.1	+0.1	+0.1	0	0	0	0	+0.1	+0.1	0	+0.1	+0.1
Ours^{gelu}	0%	+1.2	+0.9	+1.1	**+0.9**	**+1.1**	**+1.0**	**+0.9**	+0.6	**+0.8**	**+1.0**	+0.9	**+1.0**

Table 14: Average improvements on the EC30 dataset over the baseline (mT-big). 'Ours^{Enc}' and 'Ours^{Dec}' indicate neuron specialization applied solely to the Encoder and Decoder, respectively, while 'Ours' signifies the method applied to both components. The best results are highlighted in **bold**.

demonstrates robust positive gains, suggesting that our method is stable across various k settings.

In addition, we show that increasing k leads to higher improvements in general, and the optimal performance is about when k=95%. Such observation follows the intuition since when k is too low, model capacity will be largely reduced. Moreover, we find that when the FFN capacity is significantly reduced (k being very small), we still observe performance gains. Notably, even when 70%-83% of FFN weights are zeroed out (as shown in Figure 5), our method still achieves an increase of +0.6 SacreBLEU. These results indicate that our method can deliver consistent and positive gains without extensive hyperparameter tuning.

Furthermore, in Figure 6, we show that the sparsity of the network presents an intuitive structure: the sparsity decreases in the Encoder and increases in the Decoder. This implies the natural signal within the pre-trained multilingual model that neurons progress from language-specific to language-agnostic in the Encoder, and vice versa in the De-

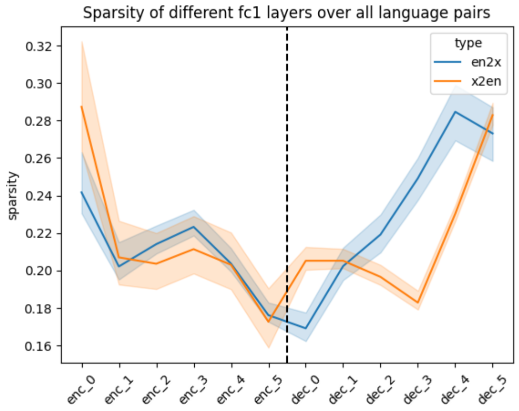

Figure 6: Sparsity progression of Neuron Specialization when $k = 95$ on the EC30. We observe that the sparsity becomes smaller in the Encoder and then goes up in the Decoder. Note that this figure is based on the natural signals extracted from the untouched pre-trained model, and will be leveraged later in the process of Neuron Specialization Training. This intrinsic pattern naturally follows our intuition that specialized neurons progress from language specific to agnostic the in Encoder, and vice versa in the Decoder.

coder. Such observation is natural because it is reflected by the untouched network, similar to what we observed in the Progression of Neuron overlaps in Section 3.2.2.

A.9 Neuron Overlaps Visualization and Phylogenetic Tree.

A.9.1 Phylogenetic Tree

In section 3.2, we show that neuron overlaps reflect language proximity, supported by visualizing the IoU matrix for specialized neurons (Figure 1). Here, we provide a more rigorous analysis quantifying the correlation between neuron overlaps and linguistic distances to provide stronger evidence of how specialized neuron overlaps correlated with language similarity.

To do that, we conduct the Complete Linkage Clustering (Müllner, 2011; Bar-Joseph et al., 2001) to build a phylogenetic tree[8] using the IoU scores for specialized neurons in the first decode FFN layer. As Figure 7 shows, the result presents strong evidence that neuron overlaps mirror the pattern of language proximity.

A.9.2 Additional IoU Visualizations

We provide additional Pairwise Intersection over Union (IoU) scores for specialized neurons in the first Encoder layer (Figure 8), last Encoder layer (Figure 9), and last Decoder layer (Figure 10). The figures show that the Neurons gradually changed from language-specific to language-agnostic in the Encoder, and vice versa in the Decoder.

A.10 Pseudocode of Neuron Specialization

We provide the pseudocode of our proposed method, *Neuron Specialization*. We present the process of Specialized Neuron Identification in Algorithm. 1 and Neuron Specialization Training in Algorithm. 2.

[8] The implementation is done using scipy.cluster.hierarchy.linkage from the SciPy package.

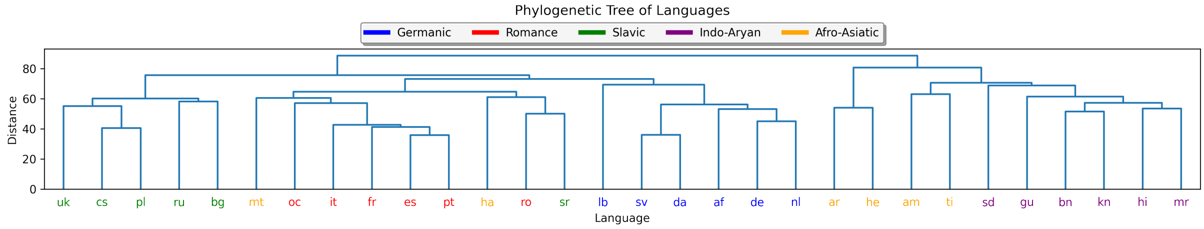

Figure 7: Phylogenetic Tree of Languages using neuron overlap IoU scores in the **first decoder layer** (the IoU values correspond to Figure 1). This figure presents strong evidence that neuron overlaps mirror the pattern of language proximity.

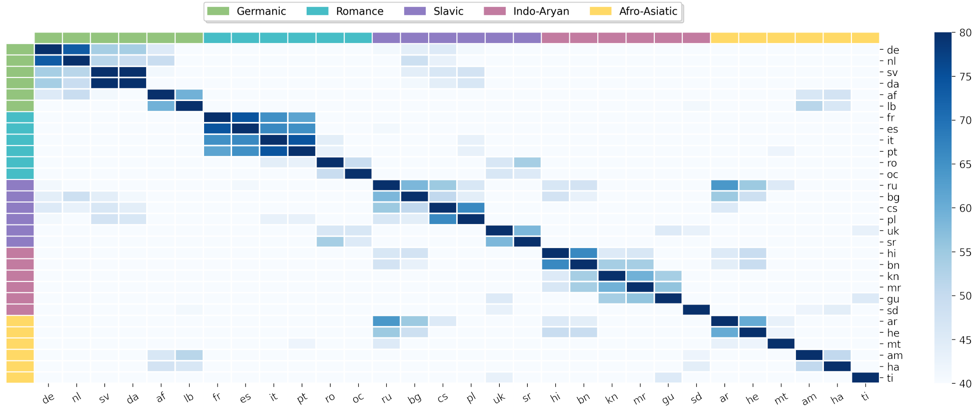

Figure 8: Pairwise Intersection over Union (IoU) scores for specialized neurons extracted from the **first encoder** FFN layer across all X-En language pairs to measure the degree of overlap between language pairs. Darker cells indicate stronger overlap, with the color threshold set from 40 to 80 to improve visibility.

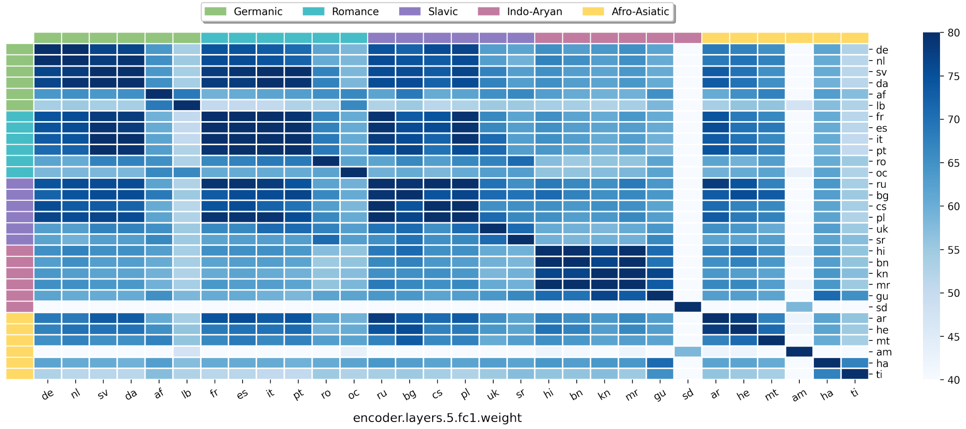

Figure 9: Pairwise Intersection over Union (IoU) scores for specialized neurons extracted from the **last encoder** FFN layer across all One-to-Many language pairs to measure the degree of overlap between language pairs. Darker cells indicate stronger overlap, with the color threshold set from 40 to 80 to improve visibility.

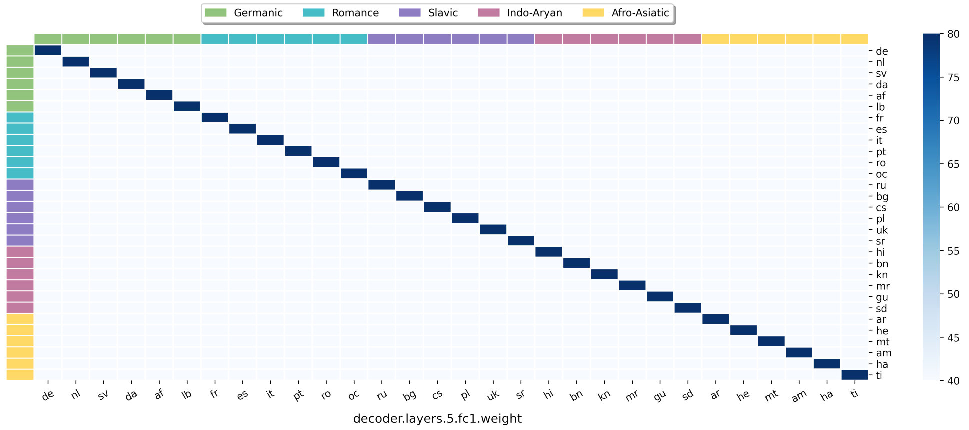

Figure 10: Pairwise Intersection over Union (IoU) scores for specialized neurons extracted from the **last decoder** FFN layer across all X-En language pairs to measure the degree of overlap between language pairs. Darker cells indicate stronger overlap, with the color threshold set from 40 to 80 to improve visibility.

Algorithm 1 Specialized Neuron Identification

```
Input: A pre-trained multi-task model $\theta$ with dimensions $d$ and $d_{ff}$; a validation dataset $D$ with $T$ tasks, where $D = \{D_1, ..., D_T\}$; and an accumulation threshold factor $k \in [0\%, 100\%]$ as the only hyper-parameter.
Output: A set of selected specialized neurons $S_k^t$ for each task $t$.
for task $t$ in $T$ do
    Step 1: Activation Recording
    Initialize activation vector $A_t = \mathbf{0} \in \mathbb{R}^{d_{ff}}$
    for sample $x_i$ in $D_t$ do
        Record activation state $a_i^t \in \mathbb{R}^{d_{ff}}$
        $A_t = A_t + a_i^t$                                  ▷ Accumulate activation states
    end for
    $a^t = \frac{A_t}{|D_t|}$                  ▷ Compute average activation state for task $t$
    Step 2: Neuron Selection
    Initialize selected neurons set $S_k^t = \emptyset$
    while selection condition not met do                 ▷ Refer to Eq. 1 for condition
        Select neurons based on $a^t$ and add them to $S_k^t$
    end while
end for
```

Algorithm 2 Neuron Specialization Training

```
Input: A pre-trained multi-task model $\theta$ with dimensions $d$ and $d_{ff}$. Corpora data $C$ with $T$ tasks that contain both training and validation data. A set of selected specialized neurons $S_k^t$ for each task $t$.
Output: A new specialized network $\theta^{new}$. Note that only the fc1 weight matrix will be trained task-specifically, the other parameters are shared across tasks. In addition, $\theta^{new}$ does not contain more trainable parameters than $\theta$ due to the sparse network feature.
Derive boolean mask $m^t \in \{0, 1\}^{d_{ff}}$ from $S_k^t$ for each layer
while $\theta^{new}$ not converge do
    for task $t$ in $T$ do
        $W_1^T = m^t \cdot W_1^\theta$                ▷ We perform this for all layers, refer to EQ. 4
        Train $\theta^{new}$ using $C^t$    ▷ All parameters will be updated, yet fc1 layers are task specific
    end for
end while
```