Parameter Differentiation based Multilingual Neural Machine Translation

Qian Wang[1,2] , **Jiajun Zhang**[1,2*]
[1]**National Laboratory of Pattern Recognition, CASIA, Beijing, China**
[2]**University of Chinese Academy of Sciences, Beijing, China**
{qian.wang, jjzhang}@nlpr.ia.ac.cn

Abstract

Multilingual neural machine translation (MNMT) aims to translate multiple languages with a single model and has been proved successful thanks to effective knowledge transfer among different languages with shared parameters. However, it is still an open question which parameters should be shared and which ones need to be task-specific. Currently, the common practice is to heuristically design or search language-specific modules, which is difficult to find the optimal configuration. In this paper, we propose a novel parameter differentiation based method that allows the model to determine which parameters should be language-specific during training. Inspired by cellular differentiation, each shared parameter in our method can dynamically differentiate into more specialized types. We further define the differentiation criterion as inter-task gradient similarity. Therefore, parameters with conflicting inter-task gradients are more likely to be language-specific. Extensive experiments on multilingual datasets have demonstrated that our method significantly outperforms various strong baselines with different parameter sharing configurations. Further analyses reveal that the parameter sharing configuration obtained by our method correlates well with the linguistic proximities.

1 Introduction

Neural machine translation (NMT) has achieved great success and drawn much attention in recent years (Sutskever, Vinyals, and Le 2014; Bahdanau, Cho, and Bengio 2015; Vaswani et al. 2017). While conventional NMT can well handle the translation of a single language pair, training an individual model for each language pair is resource-consuming, considering there are thousands of languages in the world. Therefore, multilingual NMT is developed to handle multiple language pairs in one model, greatly reducing the cost of offline training and online deployment (Ha, Niehues, and Waibel 2016; Johnson et al. 2017). Besides, the parameter sharing in multilingual neural machine translation encourages positive knowledge transfer among different languages and benefits low-resource translation (Zhang et al. 2020; Siddhant et al. 2020).
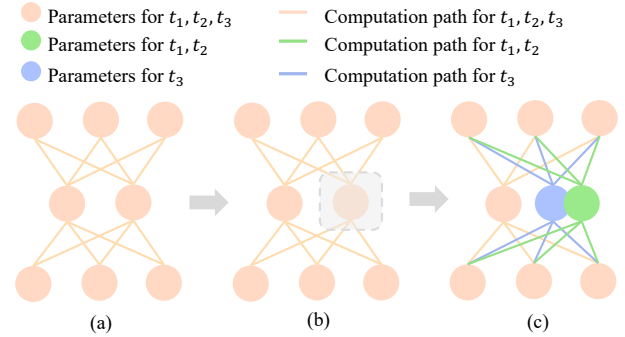
Figure 1: The illustration of parameter differentiation. Each task t_i represents a translation direction, e.g. EN→DE. (a) Initialized as completely shared, (b) the model detects parameters that should be more specialized during training, and (c) the shared parameters differentiated (◯) into more specialized types (◯◯).

Despite the benefits of the joint training with a completely shared model, the MNMT model also suffers from insufficient model capacity (Arivazhagan et al. 2019; Lyu et al. 2020). The shared parameters tend to preserve the general knowledge but ignore language-specific knowledge. Therefore, researchers resort to heuristically design additional language-specific components and build MNMT model with a mix of shared and language-specific parameters to increase the model capacity (Sachan and Neubig 2018; Wang et al. 2019b), such as the language-specific attention (Blackwood, Ballesteros, and Ward 2018), lightweight language adapter (Bapna and Firat 2019) or language-specific routing layer (Zhang et al. 2021). These methods simultaneously model the general knowledge and the language-specific knowledge but require specialized manual design. Another line of works for language-specific modeling aims to automatically search for language-specific sub-networks (Xie et al. 2021; Lin et al. 2021), in which they pretrain an initial large model that covers all translation directions, followed by sub-network pruning and fine-tuning. These methods include multi-stage training and it is non-trivial to determine the initial model size and structure.

In this study, we propose a novel parameter differentiation based method that enables the model to automatically determine which parameters should be shared and which ones should be language-specific during training. Inspired by cellular differentiation, a process in which a cell changes from one general cell type to a more specialized type, our method allows each parameter that shared by multiple tasks to dynamically differentiate into more specialized types. As shown in Figure 1, the model is initialized as completely shared and continuously detects shared parameters that should be language-specific. These parameters are then duplicated and reallocated to different tasks to increase language-specific modeling capacity. The differentiation criterion is defined as inter-task gradient similarity, which represents the consistency of optimization direction across tasks on a shared parameter. Therefore, the parameters facing conflicting inter-task gradients are selected for differentiation while other parameters with more similar inter-task gradients remain shared. In general, the MNMT model in our method can gradually improve its parameter sharing configuration without multi-stage training or manually designed language-specific modules.

We conduct extensive experiments on three widely used multilingual datasets including OPUS, WMT and IWSLT in multiple MNMT scenarios: one-to-many, many-to-one and many-to-many translation. The experimental results prove the effectiveness of the proposed method over various strong baselines. Our main contributions can be summarized as follows:

- We propose a method that can automatically determine which parameters in an MNMT model should be language-specific without manual design, and can dynamically change shared parameters into more specialized types.
- We define the differentiation criterion as the inter-task gradient similarity, which helps to minimizes the inter-task interference on shared parameters.
- We show that the parameter sharing configuration obtained by our method is highly correlated with linguistic features like language families.

2 Background

The Transformer Model A typical Transformer model (Vaswani et al. 2017) consists of an encoder and a decoder. Both the encoder and the decoder are stacked with N identical layers. Each encoder layer contains two modules named multi-head self-attention and feed-forward network. The decoder layer, containing three modules, inserts an additional multi-head cross-attention between the self-attention and feed-forward modules.

Multilingual Neural Machine Translation The standard paradigm of MNMT contains a completely shared model borrowed from bilingual translation for all language pairs. A special language token is appended to the source text to indicate the target language, i.e., $X = \{\text{lang}, x_1, \ldots, x_n\}$ (Johnson et al. 2017). The MNMT is often referred to as multi-task optimization, in which a *task* indicates a translation direction, e.g. EN→DE.

Algorithm 1: Parameter Differentiation

Input : training data $\mathcal{D}$, Tasks $T = \{t_1, t_2, \ldots\}$, models for each task $\mathcal{M} = \{\mathcal{M}_{t_1}, \mathcal{M}_{t_2}, \ldots\}$

```
   // Initialize the shared model
 1 M_t1 = M_t2 = M_t3 = ...
 2 while M not converge do
 3     Train the model M with data D
       // Detect parameters to
          differentiate
 4     flagged = []
 5     for each θ_i in shared parameters of M do
 6         Evaluate θ_i with differentiation criterion
 7         if θ_i should be language-specific then
 8             Add θ_i into flagged
 9         end
10     end
       // Reallocate parameters
11     for each θ_i shared by tasks T_i in flagged do
12         Split T_i into T_i' and T_i''
13         Duplicate θ_i into θ_i', θ_i''
14         Replace θ_i in M_{t|t∈T_i'} with θ_i'
15         Replace θ_i in M_{t|t∈T_i''} with θ_i''
16     end
17 end
```

---

3 Parameter Differentiation based MNMT

Our main idea is to find out shared parameters that should be language-specific in an MNMT model and dynamically change them into more specialized types during training. To achieve this, we propose a novel parameter differentiation based MNMT approach and define the differentiation criterion as inter-task gradient similarity.

3.1 Parameter Differentiation

As we know that cellular differentiation is the process in which a cell changes from one cell type to another, typically from a less specialized type (stem cell) to a more specialized type (organ/tissue-specific cell) (Slack 2007). Inspired by cellular differentiation, we propose parameter differentiation that can dynamically change the task-agnostic parameters in an MNMT model into other task-specific types during training.

Algorithm 1 lists the overall process of our method. We first initialize the completely shared MNMT model following the paradigm in (Johnson et al. 2017). After training for several steps, the model evaluates each shared parameter and flag the parameters that should become more specialized under a certain **differentiation criterion** (Line 4-10). For those flagged parameters, the model then duplicates them and reallocates the replicas for different tasks. After the duplication and reallocation, the model builds new connections for those replicas to construct different computation graphs $\mathcal{M}_{t_j}$ for each task (Line 11-16). In the following training steps, the parameters belonging to $\mathcal{M}_{t_j}$ only update for training data of task t_j. The differentiation happens after every several training steps and the model dynamically becomes more specialized.
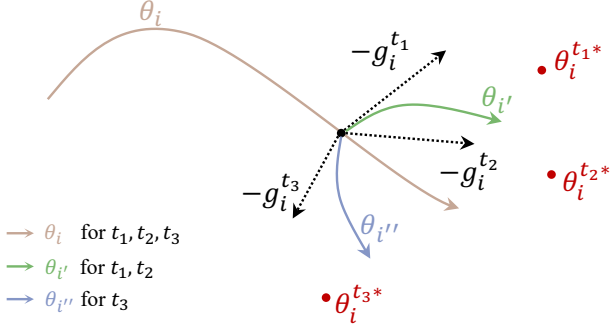
Figure 2: The illustration of parameter differentiation with gradient cosine similarity. The shared parameter θ_i differentiates into $\theta_{i'}$ for tasks $\{t_1, t_2\}$ and $\theta_{i''}$ for $\{t_3\}$ respectively since the gradients $g_i^{t_1}$ and $g_i^{t_2}$ are more similar. $\theta_i^{t_j*}$ denotes the global optimum of θ_i on task t_j.

3.2 The Differentiation Criterion

The key issue in parameter differentiation is the definition of differentiation criterion that helps to detect the shared parameters that should differentiate into more specialized types. We define the differentiation criterion based on inter-task gradient cosine similarity, where the parameters facing conflicting gradients are more likely to be language-specific.

As shown in Figure 2, the parameter θ_i is shared by tasks t_1, t_2, and t_3 at the beginning. To determine whether the shared parameter should be more specialized, we first define the **interference degree** of the parameter shared by the three tasks with the inter-task gradient cosine similarity. More formally, suppose the i-th parameter θ_i in an MNMT model is shared by a set of tasks T_i, the interference degree $\mathcal{I}$ of the parameter θ_i is defined by:

$$\mathcal{I}(\theta_i, T_i) = \max_{t_j, t_k \in T_i} -\frac{g_i^{t_j} \cdot g_i^{t_k}}{\|g_i^{t_j}\|\|g_i^{t_k}\|} \quad (1)$$

where $g_i^{t_j}$ and $g_i^{t_k}$ are the gradients of task t_j and t_k respectively on the parameter θ_i.

Intuitively, the gradients determine the optimization directions. For example in Figure 2, the gradient $g_i^{t_j}$ indicates the direction of global optimum for task t_j. The gradients with maximum negative cosine similarity, such as $g_i^{t_1}$ and $g_i^{t_3}$, point to opposite directions, which hinders the optimization and has been proved detrimental for multi-task learning (Yu et al. 2020; Wang et al. 2021).

The gradients of each task on each shared parameter are evaluated on **held-out validation data**. To minimize the gradient variance caused by inconsistent sentence semantics across languages, the validation data is created as multi-way aligned, i.e., each sentence has translations of all languages. With the held-out validation data, we evaluate gradients of each task on each shared parameter for calculating inter-task gradient similarities as well as the interference degree $\mathcal{I}$ for each parameter.

Granularity	Examples of Differentiation Units
Layers	encoder layer, decoder layer
Module	self-attention, feed-forward, cross-attention
Operation	linear projection, layer normalization

Table 1: The examples of differentiation units under different granularities.

The interference degree $\mathcal{I}$ helps the model to find out parameters that face severe interference and the parameters with high interference degrees are flagged for differentiation. Suppose the parameter θ_i shared by tasks T_i is flagged, we cluster the tasks in T_i into two subsets $T_{i'}$ and $T_{i''}$ that minimize the overall interference. The partition P^* is obtained by:

$$P_i^* = \operatorname*{argmin}_{T_{i'}, T_{i''}} [\mathcal{I}(\theta_i, T_{i'}) + \mathcal{I}(\theta_i, T_{i''})] \quad (2)$$

As shown in Figure 2, the gradients of $g_i^{t_1}$ and $g_i^{t_2}$ are similar while $g_i^{t_1}$ and $g_i^{t_3}$ are in conflict with each other. By minimizing the overall interference degree, the tasks are clustered into partition P^* : $T_{i'} = \{t_1, t_2\}, T_{i''} = \{t_3\}$. The parameter θ_i is then duplicated into $\theta_{i'}$ and $\theta_{i''}$ and the replicas are allocated to $T_{i'}$ and $T_{i''}$ respectively.

3.3 The Differentiation Granularity

In theory, each shared parameter can differentiate into more specialized types individually. But in practice, performing differentiation on every single parameter is resource- and time-consuming, considering there are millions to billions of parameters in an MNMT model.

Therefore, we resort to different levels of differentiation granularity, like Layer, Module, or Operation. As shown in Table 1, the Layer granularity indicates different layers in the model, while the Module granularity specifies the individual modules within a layer. The Operation granularity includes the basic transformations in the model that contain trainable parameters. With a certain granularity, the parameters are grouped into different differentiation units. For example, with the Layer level granularity, the parameters within a layer are concatenated into a vector and differentiate together, where the vector is referred to as a differentiation unit. We list the differentiation units in supplementary materials.

3.4 Training

In our method, since the model architecture dynamically changes and results in different computation graphs for each task, we create batches from the multilingual dataset and ensure that each batch contains only samples from one task. This is different from the training of vanilla completely shared MNMT model where each batch may contain sentence pairs from different languages (Johnson et al. 2017). Specifically, we first sample a task t_j, followed by sampling a batch $\mathcal{B}_{t_j}$ from training data of t_j. Then, the model $\mathcal{M}_{t_j}$ which includes a mix of shared and language-specific parameters is trained with the batch $\mathcal{B}_{t_j}$.

We train the model with the Adam optimizer (Kingma and Ba 2015), which computes adaptive learning rates based on the optimizing trajectory of past steps. However, the optimization history becomes inaccurate for the differentiated parameters. For the example in Figure 2, the differentiated parameter $\theta_{i''}$ is only shared by task t_3, while the optimization history of θ_i represents the optimizing trajectory of all the 3 tasks. To stabilize the training of $\theta_{i''}$ on task t_3, we reinitialize the optimizer states by performing a warm-up update for those differentiated parameters:

$$m'_t = \beta_1 m_t + (1 - \beta_1)(g_i^{t_3})$$
$$v'_t = \beta_2 v_t + (1 - \beta_2)(g_i^{t_3})^2 \tag{3}$$

where m_t and v_t are the Adam states of θ_i, and $g_i^{t_3}$ is the gradient of task t_3 on the held-out validation data. Note that we only update the states in the Adam optimizer and the parameters remain unchanged in the warm-up update step.

4 Experiments

4.1 Dataset

We use the public OPUS and WMT multilingual datasets to evaluate our method on many-to-one (M2O) and one-to-many (O2M) translation scenarios, and the IWSLT datasets for the many-to-many (M2M) translation scenario.

The OPUS dataset consists of English to 12 languages selected from the original OPUS-100 dataset (Zhang et al. 2020). These languages, containing 1M sentences for each, are from 6 distinct language groups: Romance (French, Italian), Baltic (Latvian, Lithuanian), Uralic (Estonian, Finnish), Austronesian (Indonesian, Malay), West-Slavic (Polish, Czech) and East-Slavic (Ukrainian, Russian).

The WMT dataset with unbalanced data distribution is collected from the WMT'14, WMT'16 and WMT'18 benchmarks. We select 5 languages with data sizes ranging from 0.6M to 39M. The training data sizes and sources are shown in Table 2. We report the results on the WMT dataset with the temperature-based sampling in which the temperature is set to $\tau = 5$ (Arivazhagan et al. 2019).

We evaluate our method on the many-to-many scenario with the IWSLT'17 dataset, which includes German, English, Italian, Romanian, and Dutch, and results in 20 translation directions between the 5 languages. Each translation direction contains about 200k sentence pairs.

The held-out multi-way aligned validation data for measuring gradient similarities contains $4,000$ sentences for each language, and are randomly selected and excluded from the training set. We apply the byte-pair encoding (BPE) algorithm (Sennrich, Haddow, and Birch 2016) with vocabulary sizes of 64k for both OPUS and WMT datasets, and 32k for the IWSLT dataset.

4.2 Model Settings

We conduct our experiments with the Transformer architecture and adopt the *transformer_base* setting which includes 6 encoder and decoder layers, 512/2048 hidden dimensions and 8 attention heads. Dropout ($p = 0.1$) and label smoothing ($\epsilon_{ls} = 0.1$) are applied during training but disabled

Language Pair	Data Source	#Samples
English-French (EN-FR)	WMT'14	39.03M
English-Czech (EN-CS)	WMT'14	15.65M
English-German (EN-DE)	WMT'14	4.46M
English-Estonian (EN-ET)	WMT'18	1.94M
English-Romanian (EN-RO)	WMT'16	0.61M

Table 2: Training data sizes and sources for the unbalanced WMT dataset.

during validation and inference. Each mini-batch contains roughly $8,192$ tokens. We accumulate gradients and update the model every 4 steps for OPUS and 8 steps for WMT to simulate multi-GPU training. In inference, we use beam search with the beam size of 4 and the length penalty of 0.6. We measure the translation quality by BLEU score (Papineni et al. 2002) with SacreBLEU[1]. All the models are trained and tested on a single Nvidia V100 GPU.

Our method allows the parameters to differentiate into specialized types by duplication and reallocation, which may results in bilingual models with unlimited parameter differentiation, i.e., each parameter is only shared by one task in the final model. To prevent over-specialization and make a fair comparison, we set a differentiation upper bound defined by the expected final model size $\mathcal{O}$, and let the model control the number of parameters (denoted as k) to differentiate[2]:

$$\mathcal{O} \approx \mathcal{O}_0 + \frac{Q}{N} \times k$$
$$\Rightarrow k \approx \frac{N}{Q} \times (\mathcal{O} - \mathcal{O}_0) \tag{4}$$

where $\mathcal{O}_0$ is the size of the original completely shared model. The total training step Q is set to 400k for all experiments, and the differentiation happens every $N = 8000$ steps of training. We set the expected model size to $\mathcal{O} = 2 \times \mathcal{O}_0$, 2 times of original model. We also analyze the relationship between model size and translation quality by varying $\mathcal{O}$ in the range from 1.5 to 4.

4.3 Baseline Systems

We compare our method with several baseline methods with different paradigms of parameter sharing.

Bilingual trains Transformer model (Vaswani et al. 2017) for each translation direction and results in N individual models for N translation directions.

Multilingual adopts the standard paradigm of MNMT that all parameters are shared across tasks (Johnson et al. 2017).

Random Sharing selects parameters for differentiation randomly (with Operation granularity) instead of using inter-task gradient similarity.

---

[1]https://github.com/mjpost/sacrebleu

[2]Since the parameters are grouped into differentiation units under a certain granularity, the value of k and $\mathcal{O}$ may fluctuate to comply with the granularity.

Table 3: BLEU scores on the OPUS dataset.

| Languages | | FR↔EN | | IT↔EN | | LV↔EN | | LT↔EN | | ET↔EN | |
Direction		←	→	←	→	←	→	←	→	←	→
Baselines	Bilingual (Vaswani et al. 2017)	28.90	28.27	22.55	25.55	31.60	39.75	28.88	36.43	18.65	25.48
	Multilingual (Johnson et al. 2017)	27.33	28.31	21.20	27.09	30.00	40.10	27.69	37.15	20.08	30.09
	Random Sharing	27.48	28.91	21.42	27.18	31.57	41.18	28.94	37.57	20.43	30.15
	Tan et al. (2019)	27.39	29.21	21.97	26.77	31.85	**42.71**	29.27	39.34	21.40	29.79
	Sachan and Neubig (2018)	28.04	29.31	22.86	27.86	32.04	41.43	28.47	38.14	**21.41**	30.30
Ours	PD w. Layer	**29.35**	30.09	22.37	**28.7**	32.31	42.11	29.5	39.04	20.56	30.91
	PD w. Module	29.09	30.09	22.49	28.64	31.86	41.60	29.53	39.04	21.25	31.11
	PD w. Operation	29.26	**30.11**	**23.01**	28.6	**33.06**	42.38	**29.94**	**39.54**	20.89	**31.14**

| Languages | | FI↔EN | | ID↔EN | | MS↔EN | | PL↔EN | | CS↔EN | |
Direction		←	→	←	→	←	→	←	→	←	→
Baselines	Bilingual (Vaswani et al. 2017)	13.92	18.34	21.29	25.61	16.75	21.24	13.46	19.05	16.82	25.27
	Multilingual (Johnson et al. 2017)	15.58	21.43	22.85	28.27	18.12	23.66	14.87	22.24	18.57	28.14
	Random Sharing	16.01	21.30	21.69	27.78	17.13	23.73	15.23	21.97	18.40	28.21
	Tan et al. (2019)	16.15	21.46	22.74	28.00	18.12	23.14	14.86	21.72	18.02	28.08
	Sachan and Neubig (2018)	16.37	21.36	22.39	**29.60**	17.33	23.77	15.75	22.45	**19.70**	28.59
Ours	PD w. Layer	16.42	22.37	22.89	29.28	18.35	24.88	16.07	23.11	19.29	29.31
	PD w. Module	16.44	**22.85**	22.94	28.86	17.62	24.27	16.18	23.12	19.33	29.08
	PD w. Operation	**16.59**	**22.85**	23.09	29.03	**18.61**	**25.27**	**16.45**	**23.34**	19.46	**29.66**

| Languages | | UK↔EN | | RU↔EN | | Average | | Δ Average | | Model Size | |
Direction		←	→	←	→	←	→	←	→	←	→
Baselines	Bilingual (Vaswani et al. 2017)	10.06	18.68	21.63	26.61	20.38	25.86	-0.36	-2.22	12x	12x
	Multilingual (Johnson et al. 2017)	11.59	21.76	20.96	28.76	20.74	28.08	0	0	1x	1x
	Random Sharing	11.57	21.83	21.36	28.91	20.93	28.23	+0.19	+0.15	1.98x	2.00x
	Tan et al. (2019)	11.32	21.74	21.32	28.73	21.20	28.39	+0.46	+0.31	2x	2x
	Sachan and Neubig (2018)	10.96	21.88	22.28	28.80	21.47	28.62	+0.73	+0.54	3.71x	3.25x
Ours	PD w. Layer	12.32	22.68	22.82	30.37	21.85	29.40	+1.11	+1.32	2.14x	1.84x
	PD w. Module	**12.55**	22.44	22.31	30.39	21.80	29.29	+1.06	+1.21	1.82x	1.94x
	PD w. Operation	12.37	**23.05**	**22.98**	**30.60**	**22.14**	**29.63**	+1.40	+1.55	1.96x	1.90x

Table 3: BLEU scores on the OPUS dataset. We compare our method with different levels of parameter sharing in both one-to-many (←) and many-to-one (→) directions. We report our parameter differentiation (PD) method with different granularity: Layer, Module and Operation. **Bold** indicates the best result of all methods.

Sachan and Neubig (2018) uses a partially shared model that proved effective empirically. They share the attention key and query of the decoder, the embedding, and the encoder in a one-to-many model. We extend the settings for the many-to-one model that share the attention key and query of the encoder, the embedding, and the decoder.

Tan et al. (2019) first clusters the languages using the language embedding vectors in the **Multilingual** method and then trains one model for each cluster. To make the model size comparable with our method, we set the number of clusters as 2 and train two distinct models. In our experiment on the OPUS dataset, this method results in two clusters: {FR, IT, ID, MS, PL, CS, UK, RU} and {LV, LT, ET, FI}.

4.4 Results

OPUS Table 3 shows the results of our method and the baseline methods on the OPUS dataset. In both one-to-many (←) and many-to-one (→) directions, our methods consis-tently outperform the Bilingual and Multilingual baselines and gains improvement over the Multilingual baseline by up to +1.40 and +1.55 BLEU on average. Compared to other parameter sharing methods, our method achieves the best results in 20 of 24 translation directions and improves the average BLEU by a large margin. As for the different granularities in our method, we find that the Operation level achieves the best results on average, due to the fine-grained control of parameter differentiation compared to the Layer level and the Module level.

For the model sizes, the method of (Sachan and Neubig 2018) that pre-defines the sharing modules increases linearly with the number of languages involved and results in a larger model size (3.71x). In our method, the model size is unrelated to the number of languages, which provides more scalability and flexibility. Since we use different granularities instead of performing differentiation on every single parameter, the actual sizes of our method range from 1.82x to 2.14x, close but not equal to the predefined 2x.

Languages	FR↔EN		CS↔EN		DE↔EN		ET↔EN		RO↔EN		Average		Sizes	
Direction	←	→	←	→	←	→	←	→	←	→	←	→	←	→
Bilingual	39.87	37.74	27.23	31.43	26.71	31.98	17.55	23.26	23.13	29.23	26.90	30.73	5x	5x
Multilingual	38.07	36.23	25.39	30.77	24.67	31.54	18.90	26.09	26.42	34.85	26.69	31.90	1x	1x
Ours	**40.28**	**37.36**	**26.75**	**32.92**	**27.29**	**32.80**	**19.66**	**27.64**	**27.34**	**35.90**	**28.26**	**33.32**	1.83x	1.87x

Table 4: Results on the WMT dataset. Our method is parameter differentiation with granularity of Operation. **Bold** indicates the best result for multilingual model while the overall best results are underlined.

	→DE	→EN	→IT	→NL	→RO	Average
DE→	-	33.09 / **34.07**	20.10 / **21.05**	22.18 / **22.35**	18.20 / **19.00**	23.39 / **24.12**
EN→	26.55 / **27.86**	-	27.74 / **28.69**	28.15 / **28.61**	25.34 / **26.58**	26.95 / **27.94**
IT→	19.32 / **20.37**	32.14 / **32.99**	-	20.05 / **20.47**	19.28 / **20.10**	22.70 / **23.48**
NL→	21.06 / **22.09**	32.54 / **33.45**	19.81 / **20.64**	-	17.93 / **18.81**	22.84 / **23.75**
RO→	20.74 / **21.39**	34.78 / **35.76**	22.96 / **23.53**	20.87 / **21.04**	-	24.84 / **25.43**
Average	21.92 / **22.93**	33.14 / **34.07**	22.65 / **23.48**	22.81 / **23.12**	20.19 / **21.12**	30.08 / **31.18**

Table 5: The many-to-many translation results on the IWSLT dataset. Our parameter differentiation method is based on the granularity of Operation. We compare our method with the Multilingual method and report the result with format of Multilingual/Ours. **Bold** indicates the better result.

WMT We further investigate the generalization performance with experiments on the unbalanced WMT dataset. As shown in Table 4, the Multilingual model benefits lower-resource languages (ET, RO) translation but hurts the performances of higher-resource languages (FR, CS, DE). In contrast, our method gains more improvements in higher-resource language (+2.21 for FR←EN) than lower-resource language (+1.05 for RO→EN). Our method can also outperform the Bilingual method in 8 of 10 translation directions.

IWSLT The results on the many-to-many translation scenario with the IWSLT dataset are shown in Table 5. Our method based on Operation level granularity outperforms the Multilingual baseline in all 20 translation directions, but the improvement (+1.10 BLEU on average) is less significant than those on the other two datasets. The reason is that the 5 languages in the IWSLT dataset belong to the same Indo-European language family and thus the shared parameters may be sufficient for modeling all translation directions.

4.5 Analyses

Parameter Differentiation Across Layers Using a shared encoder for one-to-many translation and a shared decoder for many-to-one translation has been proved effective and is widely used (Zoph and Knight 2016; Dong et al. 2015; Sachan and Neubig 2018). However, there lack of analyses on different sharing strategies across layers. The parameter differentiation method provides a more fine-grained control of parameter sharing, making it possible to offer such analyses. To investigate the parameter sharing across layers, we calculate the number of differentiation units within each layer of the final model trained with Operation level granularity. For comparison, the completely shared model has 8 differentiation units in each encoder layer and 13 in each decoder layer (see details in supplementary materials).
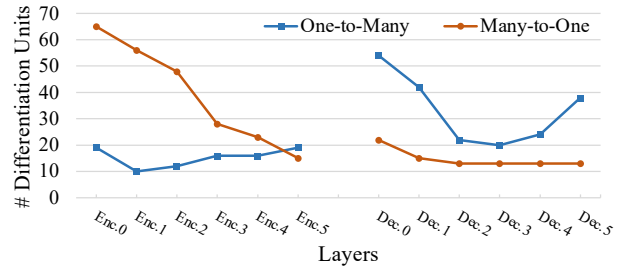

Figure 3: The number of differentiation units within each layer of the final model.

The results are shown in Figure 3. For many-to-one translation, the task-specific parameters are mainly distributed in shallower layers of the encoder and the parameters in the decoder tend to stay shared. On contrary, for one-to-many translation, the decoder has more task-specific parameters than the encoder. Different from the encoder in which shallower layers are slightly more task-specific, both the shallower and the deeper layers are more specific than the middle layers in the decoder. The reason is that the shallower layers in the decoder take tokens from multiple languages as input and the deeper layers are responsible for generating tokens in multiple languages.

Parameter Differentiation and Language Family We investigate the correlation between the parameter sharing obtained by differentiation and the language families. Intuitively, linguistically similar languages are more likely to have shared parameters. To verify this, we first select encoder.layer-0.self-attention.value-projection, which differentiate for the most times and is the most specialized, and then analyze its differentiation process during training.
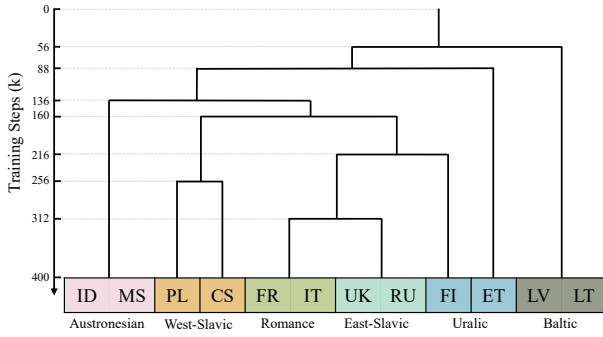
Figure 4: The differentiation process of the parameter group encoder.layer-0.self-attention.value-projection. Parameters are shared across languages in a square and the colors represent linguistic proximities.
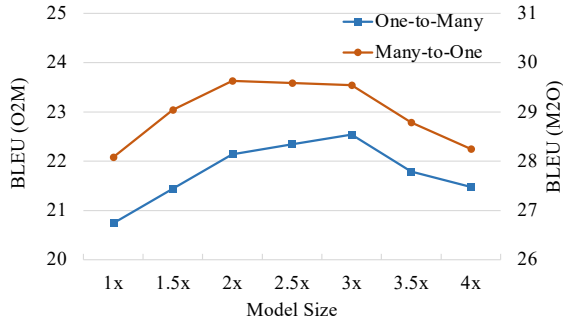

Figure 5: The correlation between model size and the average BLEU over all language pairs on the OPUS dataset in both one-to-many and many-to-one directions.

Figure 4 shows the differentiation process of the most specialized parameter. From the training steps, we can find that the differentiation happens aperiodically for this parameter. As for the differentiation results, it is obvious that the parameter sharing strategy is highly correlated with the linguistic proximity like language family or language branch. For example, ID and MS belong to the Austronesian language and share the parameters while ID and FR belonging to the Austronesian language and the Romance language respectively have task-specific parameters. Another interesting observation is that the Baltic languages (LV and LT) become specialized at the early stage of training. We examine the OPUS dataset and find out that the training data of LV and LT are mainly from the political domain, while other languages are mainly from the spoken domain.

The Effect of Model Size We notice that the model size is not completely correlated with performance according to the results in Table 3. Our method initialize the model as completely shared with the model size of 1x, and may differentiate into bilingual models in extreme cases. The completely shared model tends to preserve the general knowledge, while the bilingual models only capture language-specific knowledge. To investigate the effect of the differentiation level, we evaluate the relationship between model size and translation quality.

As shown in Figure 5, the performance first increases with a higher differentiation level (larger model size) and then decreases when the model grows over a certain threshold. The best results are obtained with 3x and 2x model sizes for one-to-many and many-to-one directions respectively, which indicates that the model needs more parameters for handling multiple target languages (one-to-many) than multiple source languages (many-to-one).

5 Related Work

Multilingual neural machine translation (MNMT) aims at handling translation between multiple languages with a single model (Dabre, Chu, and Kunchukuttan 2020). In the early stage, researchers share different modules like encoder (Dong et al. 2015), decoder (Zoph and Knight 2016), or attention mechanism (Firat, Cho, and Bengio 2016) to reduce the parameter scales in bilingual models. The success in sharing modules motivates a more aggressive parameter sharing that handles all languages with a completely shared model (Johnson et al. 2017; Ha, Niehues, and Waibel 2016).

Despite its simplicity, the completely shared model faces capacity bottlenecks for retaining specific knowledge of each language (Aharoni, Johnson, and Firat 2019). Researchers resort to language specific modeling with various parameter sharing strategies (Sachan and Neubig 2018; Wang et al. 2019b, 2018), such as the attention module (Wang et al. 2019a; Blackwood, Ballesteros, and Ward 2018; He et al. 2021), decoupling encoder or decoder (Escolano et al. 2021), additional adapters (Bapna and Firat 2019), and language clustering (Tan et al. 2019).

Instead of augmenting the model with manually designed language-specific modules, researchers attempt to search for a language-specific sub-space of the model, such as generating the language-specific parameters from global ones (Platanios et al. 2018), language-aware model depth (Li et al. 2020), language-specific routing path (Zhang et al. 2021) and language-specific sub-networks (Xie et al. 2021; Lin et al. 2021). These methods start from a large model that covers all translation directions, where the size and structure of the initial model are non-trivial to determine. While our method initializes a simple shared model and lets the model to automatically grows into a more complicated one, which provides more scalability and flexibility.

6 Conclusion and Future Work

In this paper, we propose a novel parameter differentiation based method that can automatically determine which parameters should be shared and which ones should be language-specific. The shared parameters can dynamically differentiate into more specialized types during training. The extensive experiments on three multilingual machine translation datasets verify the effectiveness of our method. The analyses reveal that the parameter sharing configurations obtained by our method are highly correlated with the linguistic proximities. In the future, we want to let the model learn when to stop differentiation and explore other differentiation criteria for more multilingual scenarios like the zero-shot translation and the incremental multilingual translation.

7 Acknowledgement

References

Aharoni, R.; Johnson, M.; and Firat, O. 2019. Massively Multilingual Neural Machine Translation. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 3874–3884. Association for Computational Linguistics.

Arivazhagan, N.; Bapna, A.; Firat, O.; Lepikhin, D.; Johnson, M.; Krikun, M.; Chen, M. X.; Cao, Y.; Foster, G. F.; Cherry, C.; Macherey, W.; Chen, Z.; and Wu, Y. 2019. Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges. *CoRR*, abs/1907.05019.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bapna, A.; and Firat, O. 2019. Simple, Scalable Adaptation for Neural Machine Translation. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 1538–1548. Association for Computational Linguistics.

Blackwood, G. W.; Ballesteros, M.; and Ward, T. 2018. Multilingual Neural Machine Translation with Task-Specific Attention. In Bender, E. M.; Derczynski, L.; and Isabelle, P., eds., *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, 3112–3122. Association for Computational Linguistics.

Dabre, R.; Chu, C.; and Kunchukuttan, A. 2020. A Survey of Multilingual Neural Machine Translation. *ACM Comput. Surv.*, 53(5): 99:1–99:38.

Dong, D.; Wu, H.; He, W.; Yu, D.; and Wang, H. 2015. Multi-Task Learning for Multiple Language Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, 1723–1732.

Escolano, C.; Costa-jussà, M. R.; Fonollosa, J. A. R.; and Artetxe, M. 2021. Multilingual Machine Translation: Closing the Gap between Shared and Language-specific Encoder-Decoders. In Merlo, P.; Tiedemann, J.; and Tsarfaty, R., eds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Lin-*

guistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, 944–948. Association for Computational Linguistics.

Firat, O.; Cho, K.; and Bengio, Y. 2016. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 866–875.

Ha, T.; Niehues, J.; and Waibel, A. H. 2016. Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder. In *13th International Workshop on Spoken Language Translation 2016*, 5.

He, H.; Wang, Q.; Yu, Z.; Zhao, Y.; Zhang, J.; and Zong, C. 2021. Synchronous Interactive Decoding for Multilingual Neural Machine Translation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 12981–12988. AAAI Press.

Johnson, M.; Schuster, M.; Le, Q. V.; Krikun, M.; Wu, Y.; Chen, Z.; Thorat, N.; Viégas, F. B.; Wattenberg, M.; Corrado, G.; Hughes, M.; and Dean, J. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Trans. Assoc. Comput. Linguistics*, 5: 339–351.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Li, X.; Stickland, A. C.; Tang, Y.; and Kong, X. 2020. Deep Transformers with Latent Depth. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Lin, Z.; Wu, L.; Wang, M.; and Li, L. 2021. Learning Language Specific Sub-network for Multilingual Machine Translation. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 293–305. Association for Computational Linguistics.

Lyu, S.; Son, B.; Yang, K.; and Bae, J. 2020. Revisiting Modularized Multilingual NMT to Meet Industrial Demands. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 5905–5918. Association for Computational Linguistics.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on asso-*

ciation for computational linguistics, 311–318. Association for Computational Linguistics.

Platanios, E. A.; Sachan, M.; Neubig, G.; and Mitchell, T. M. 2018. Contextual Parameter Generation for Universal Neural Machine Translation. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 425–435. Association for Computational Linguistics.

Sachan, D. S.; and Neubig, G. 2018. Parameter Sharing Methods for Multilingual Self-Attentional Translation Models. In Bojar, O.; Chatterjee, R.; Federmann, C.; Fishel, M.; Graham, Y.; Haddow, B.; Huck, M.; Jimeno-Yepes, A.; Koehn, P.; Monz, C.; Negri, M.; Névéol, A.; Neves, M. L.; Post, M.; Specia, L.; Turchi, M.; and Verspoor, K., eds., *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, 261–271. Association for Computational Linguistics.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Siddhant, A.; Johnson, M.; Tsai, H.; Ari, N.; Riesa, J.; Bapna, A.; Firat, O.; and Raman, K. 2020. Evaluating the Cross-Lingual Effectiveness of Massively Multilingual Neural Machine Translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 8854–8861. AAAI Press.

Slack, J. M. 2007. Metaplasia and transdifferentiation: from pure biology to the clinic. *Nature reviews Molecular cell biology*, 8(5): 369–378.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 3104–3112.

Tan, X.; Chen, J.; He, D.; Xia, Y.; Qin, T.; and Liu, T. 2019. Multilingual Neural Machine Translation with Language Clustering. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 963–973. Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wang, Y.; Zhang, J.; Zhai, F.; Xu, J.; and Zong, C. 2018. Three Strategies to Improve One-to-Many Multilingual Translation. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 2955–2960. Association for Computational Linguistics.

Wang, Y.; Zhang, J.; Zhou, L.; Liu, Y.; and Zong, C. 2019a. Synchronously Generating Two Languages with Interactive Decoding. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 3348–3353. Association for Computational Linguistics.

Wang, Y.; Zhou, L.; Zhang, J.; Zhai, F.; Xu, J.; and Zong, C. 2019b. A Compact and Language-Sensitive Multilingual Translation Method. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 1213–1223.

Wang, Z.; Tsvetkov, Y.; Firat, O.; and Cao, Y. 2021. Gradient Vaccine: Investigating and Improving Multi-task Optimization in Massively Multilingual Models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Xie, W.; Feng, Y.; Gu, S.; and Yu, D. 2021. Importance-based Neuron Allocation for Multilingual Neural Machine Translation. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 5725–5737. Association for Computational Linguistics.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020. Gradient Surgery for Multi-Task Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhang, B.; Bapna, A.; Sennrich, R.; and Firat, O. 2021. Share or Not? Learning to Schedule Language-Specific Capacity for Multilingual Translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Zhang, B.; Williams, P.; Titov, I.; and Sennrich, R. 2020. Improving Massively Multilingual Neural Machine Translation and Zero-Shot Translation. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 1628–1639. Association for Computational Linguistics.

Zoph, B.; and Knight, K. 2016. Multi-Source Neural Translation. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 30–34.