Language and Task Arithmetic with Parameter-Efficient Layers for Zero-Shot Summarization

Alexandra Chronopoulou[1*] **Jonas Pfeiffer**[2] **Joshua Maynez**[2]
Xinyi Wang[2] **Sebastian Ruder**[3†] **Priyanka Agrawal**[2]
[1]Google [2]Google DeepMind [3]Cohere

Abstract

Parameter-efficient fine-tuning (PEFT) using labeled task data can significantly improve the performance of large language models (LLMs) on the downstream task. However, there are 7000 languages in the world and many of these languages lack labeled data for real-world language generation tasks. In this paper, we propose to improve zero-shot cross-lingual transfer by composing expert modules trained separately on language or task data. Our method composes *language* and *task* PEFT adapters via element-wise arithmetic operations to leverage unlabeled data and English labeled data. We extend our approach to cases where labeled data from more languages is available and propose to arithmetically compose PEFT adapters trained on languages related to the target. Empirical results on summarization demonstrate that our method is a strategy that obtains consistent gains using minimal training of PEFT parameters.

1 Introduction

Large language models (LLM) have achieved impressive performance on various real world applications in many different human languages (Xue et al., 2021; Brown et al., 2020; Chowdhery et al., 2022; Anil et al., 2023; Jiang et al., 2024). Summarization (Nenkova and McKeown, 2011) is a particularly interesting and useful task because it allows users to quickly aggregate and access relevant information from large amounts of textual data. Developing a competitive text summarization system for a language typically involves fine-tuning a pretrained model on labeled summarization data in the given language. Standard supervised fine-tuning of LLMs can be very expensive due to the large model size. Parameter-efficient tuning (PEFT) is an effective alternative that achieves competitive performance while incurring much less computational and memory cost (Hu et al., 2022; Lester et al., 2021; Zhang et al., 2023b).

Despite the effectiveness of PEFT (Touvron et al., 2023), it also has several limitations if we want to develop competitive multilingual summarization systems. First, current PEFT methods generally require access to labeled task data in a given language. While there are several existing datasets in English to train competitive summarization systems (Hermann et al., 2015; Grusky et al., 2018; Narayan et al., 2018), many languages in the world with millions of speakers do not have such resources (Giannakopoulos et al., 2015; Scialom et al., 2020; Cao et al., 2020). Second, standard PEFT methods optimize a separate set of parameters for each language, resulting in thousands of fine-tuned checkpoints, which need to be stored and deployed individually (Fifty et al., 2021). Finally, as the standard PEFT methods are fine-tuned in isolation, they cannot leverage information from related tasks.

In this paper, we want to improve zero-shot multilingual summarization with PEFT to better support languages that might lack labeled summarization data. To this end, we propose a simple yet effective method that composes language and task information stored in different trained PEFT parameters through element-wise operation. We leverage unlabeled data to train language parameters with PEFT, and perform element-wise arithmetic operations with pretrained *task* and *language* parameters to construct new parameters for a language without labeled summarization data. While several prior works have studied methods that compose PEFT methods for zero-shot cross-lingual transfer (Pfeiffer et al., 2020; Vu et al., 2022), these methods generally incur an additional inference cost. Our method provides a simpler and more flexible framework to leverage many related languages at a fixed inference cost.

[*] Work done during an internship at Google DeepMind. Correspondence to alexandrachron@google.com
[†] Work done while working at Google

Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024), pages 114–126
November 16, 2024 ©2024 Association for Computational Linguistics

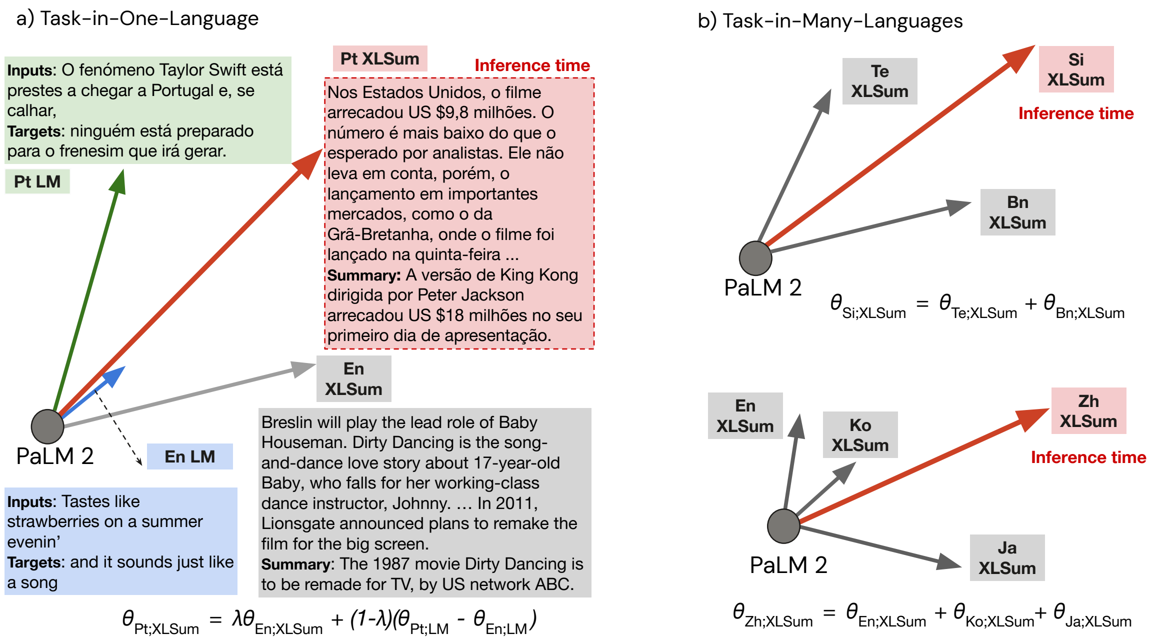

Figure 1: **Illustration of our language and task arithmetic approach for zero-shot cross-lingual transfer using LoRA parameters learned on top of PaLM 2.** (a) We train a task adapter using the summarization objective in En and language adapters using Prefix-LM in En and Pt. At inference time, a summary is generated in Pt, shown with a dotted frame (Subsection 2.1). (b) We add the weights of task adapters trained for summarization in languages similar to the target. We use the resulting vector for zero-shot summarization in the target language (Subsection 2.2).

Our method is inspired by the lottery ticket hypothesis (Frankle and Carbin, 2019), which posits that distinct models fine-tuned on the same dataset follow linear trajectories while maintaining a consistent loss (Frankle et al., 2020; Yunis et al., 2022). This hypothesis implies that element-wise operations on different fine-tuned models can also remove biases of the pretrained model (Ilharco et al., 2023), allowing the accumulation of information from auxiliary tasks (Matena and Raffel, 2021), or improve adaptation to unforeseen textual domains (Li et al., 2022a; Chronopoulou et al., 2023a). Our work is the first to extend this observation to improve cross-lingual transfer by combining pretrained language and task parameters.

Our contributions are the following:

1. Assuming we only have task data in English, we combine PEFT parameters trained on English task data and unlabeled data in other languages through element-wise composition. This setup, termed *Task-in-One-Language*, improves the model's summarization performance across all unseen target languages, as demonstrated on the XLSum benchmark (Hasan et al., 2021).
2. Extending our first approach, we consider scenarios with task data from multiple languages (*Task-in-Many-Languages*). When labeled task data for summarization are available in various languages, we combine representations from languages most related to the target, consistently improving performance over the baselines using the XLSum benchmark.
3. We apply our language and task arithmetic to a different PEFT method, the Kronecker adapter (Edalati et al., 2022) and evaluate its performance on XLSum and TyDi-QA (Clark et al., 2020). We find that our approach is also effective with these other methods and tasks.

2 Language and Task Arithmetic

Prior work has applied element-wise operations to the weights of fine-tuned models (Matena and Raffel, 2021; Wortsman et al., 2022; Ilharco et al., 2023; Ainsworth et al., 2023; Yadav et al., 2023), or PEFT modules (Chronopoulou et al., 2023a; Zhang et al., 2023a). These studies demonstrate that interpolating the weights of fine-tuned models (or specific layers) effectively creates multi-task and multi-domain models. We hypothesize that element-wise operations can also be used to combine knowledge acquired in different languages. Our work is the first to propose the arithmetic composition of language and task PEFT modules for cross-lingual natural language generation. Figure 1 illustrates an overview of our approach. 1.

Our goal is to enable Large Language Models (LLMs) to support summarization in an unseen target language (T) for which we lack labeled data. We assume access to labeled task data in other languages, as well as unlabeled monolingual data in both the source language (S) and the target language (T). In particular, we can use either labeled or unlabeled data to train small PEFT modules that capture the attributes of a given task or language.

Task Adapter: We fine-tune an LLM using LoRA adapters on labeled data from XLSum (Hasan et al., 2021) in the source language S. We refer to the fine-tuned model as Task Adapter.

Language Adapter: We fine-tune LoRA parameters with LLMs on monolingual data in the source or target language (S or T). We refer to the fine-tuned model as language adapter. We use the prefix-LM pretraining objective from T5 (Raffel et al., 2020) with mC4 data to train language adapters.

We propose to compose the *language* and *task* vectors to better support summarization into the target language T. Next, we introduce our method under two different data settings.

2.1 Task-in-One-Language

First, we consider the zero-shot setting where the source language S is English. We have labeled data in S, and some amount of unlabeled data both in the source language S and the target language T.

Composing via Language and Task Addition: We want to encourage the model to generate in the target language T and learn the task from the data available in the source language S.

Let $\theta_{\text{LM;T}}$ be the LoRA parameters trained on the monolingual data in the target language T, and $\theta_{\text{task;S}}$ be the LoRA parameters trained on the labeled task data in the source language S, we propose to calculate the zero-shot task module for the target language T as:

$$\theta_{\text{task;T}} = \lambda\theta_{\text{task;S}} + (1-\lambda)(\theta_{\text{LM;T}}) \quad (1)$$

The scaling term λ is determined using held-out validation data. We refer to this approach as *Language and Task; Add.*

Composing via Language and Task Addition and Subtraction: We want to steer the model's ability to generate in the target language T, but avoid generating in the source language S. Previous work showed that subtraction can be a method of "unlearning" information (Ilharco et al., 2023; Zhang et al., 2023a). We propose *subtracting* the source language adapter from the target language adapter. The intuition is that by negating the source language adapter, we control the generation, making the model "forget" the source language.

Our goal in this zero-shot transfer setup is to obtain a model that has a **strong summarization ability** (learned from the task adapter) **in the correct target language** (learned from the target language adapter) **while *not* generating in the source language** (unlearned from the source language adapter).

Formally, let $\theta_{\text{LM;S}}$ be the LoRA parameters trained on the monolingual data in the source language S. We propose to calculate the zero-shot task module for the target language T as:

$$\theta_{\text{task;T}} = \lambda\theta_{\text{task;S}} + (1-\lambda)(\theta_{\text{LM;T}} - \theta_{\text{LM;S}}) \quad (2)$$

where λ is a hyperparameter tuned in the same way as in the previous setting. We refer to it as *Language and Task; Add and Subtract*.

2.2 Task-in-Many-Languages

Subsection 2.1 presents language and task arithmetic when we want to do zero-shot transfer from a single source language S. However, in practice, we sometimes have data in many different source languages. In this subsection, we extend our language and task arithmetic framework to the setting where we utilize data in many different languages.

Composing via Task-only Addition: First, we want to utilize labeled task data in various source languages. Formally, given labeled task data for N languages ($S_1, ..., S_N$), we want to use the LLM to support an unseen target language T, for which we have no task data. To this end, given LoRA parameters ($\theta_{\text{task;S}_1}, ..., \theta_{\text{task;S}_N}$) trained on labeled task data in ($S_1, ..., S_N$), we propose to perform zero-shot generation on the target language T using the average of PEFT modules of its related languages:

$$\theta_{\text{task;T}} = \frac{1}{L}\sum_{i=1}^{L}\theta_{\text{task;S}_i} \quad (3)$$

where $L <= N$. If $L = N$, we essentially add the weights of all available task adapters (we name this method *Task-only; Add all*). To select a subset of L languages that are most related to the target language T, we use the URIEL language vectors (Littell et al., 2017). We retrieve the pre-computed syntactic and geographic distances between T and each of the N languages of the training set using an implementation of the toolkit lang2vec.[1] We refer to this approach as *Task-only; Add related*.

Composing via Language and Task Addition and Subtraction: Similarly, if we have both labeled and unlabeled data in several source languages, we can modify Equation 2 to leverage both types of data in many different languages:

$$\theta_{\text{task;T}} = \lambda\theta'_{\text{task;S}} + (1-\lambda)(\theta_{\text{LM;T}} - \theta'_{\text{LM;S}}) \quad (4)$$

Where $\theta'_{task;S} = \frac{1}{L}\sum_{i=1}^{L}\theta_{\text{task;S}_i}$ (as computed in Equation 3), i.e., it is the average of the related (to the target T) task adapters, and $\theta'_{LM;S} = \frac{1}{L}\sum_{i=1}^{L}\theta_{\text{LM;S}_i}$, i.e., it is the average of the related language adapters according to URIEL. This approach is denoted as *Language and Task; Add and Subtract related*.

3 Experimental Setup

3.1 Tasks and Datasets

Summarization: We use XLSum (Hasan et al., 2021), a news summarization dataset of BBC articles, where each article has a one-sentence summary. While prior work studies the zero-shot learning setting where only English labeled data is available (Vu et al., 2022), we utilize the available multilingual training data for a more realistic setting. Specifically, we use a subset of XLSum as our training set, and specifically the articles and summaries of the languages: Arabic (ar), Bengali (bn), English (en), Japanese (ja), Korean (ko), Indonesian (id), Swahili (sw), Russian (ru), Telugu (te), Thai (th), and Turkish (tr). We refer to this set as XLSum$_{seen}$. Training dataset stats are shown in Table 7 of the Appendix.

For zero-shot evaluation, we select 11 languages from XLSum as unseen languages: Marathi (mr), Gujarati (gu), Chinese simplified (zh), Nepali (ne), Portuguese (pt), Sinhala (si), Somali (so), Vietnamese (vi), Yoruba (yo), Ukrainian (uk), and Persian (fa). We do not use training data from any of these languages. We refer to this set of 11 languages as XLSum$_{unseen}$.

Unlabeled data: We use unlabeled data from mC4 (Xue et al., 2021) with the prefix language modeling objective from T5 (Raffel et al., 2020). This corpus has been created using a Common Crawl-based dataset covering 101 languages. All languages considered in our experiments are covered by mC4. For the language adapters, we fine-tune the LLM using LoRA on prefix-LM for $5k$ steps in each language.

3.2 Training and Implementation Details

We use PaLM 2-S (Anil et al., 2023), a state-of-the-art, highly multilingual language model, as the base LLM for all our experiments.

We add LoRA parameters of rank 4 to the Key, Query, Value, Projection attention matrices. We do not tune this hyperparameter. This results in adding parameters that account for just 0.2% of the parameters of PaLM 2 (we do not update the weights of the pretrained model). We fine-tune PaLM 2 on prefix-LM, XLSum using LoRA with learning rate $2e-4$.

For XLSum, we report ROUGE-2 (Lin, 2004) as the evaluation metric for En, and SentencePiece-ROUGE-2 for all other languages. This is an extension of ROUGE that handles non-Latin character using a SentencePiece tokenizer; in this work, we use the mT5 tokenizer (Xue et al., 2021).

3.3 Baselines

TASK-IN-ONE-LANGUAGE: The baseline is computed by fine-tuning PaLM 2 on En XLSum data using LoRA parameters. During fine-tuning, only the LoRA parameters are being updated, while the underlying LLM remains frozen.

TASK-IN-MANY-LANGUAGES: The baseline is computed by fine-tuning PaLM 2 on XLSum data of each of the language in XLSum$_{seen}$ independently using LoRA parameters. Then, the best-performing model (per target language) is selected. We denote this as *baseline (best)*.

We also compute a *multilingual baseline*: we simply concatenate the datasets of the different languages of XLSum$_{seen}$ and we train the LLM with LoRA on the entire dataset.[2]

4 Results and Discussion

4.1 Task-in-One-Language

Language and task arithmetic (Add and Subtract) improves zero-shot cross-lingual transfer: We present the main results of our language and

[1] https://github.com/antonisa/lang2vec

[2] We also ran the full fine-tuning baselines and we observed that the gap to the PEFT baselines is small, results are shown in the Appendix.

Method	Mr	Gu	Zh	Ne	Pt	Si	So	Vi	Yo	Uk	Fa	Avg
Task-in-One-Language												
Baseline	20.5	30.3	23.9	29.4	22.3	34.5	21.3	24.5	17.3	17.4	25.1	24.2
Language and Task (Add)	20.6	30.3	24.1	29.4	22.3	34.7	21.5	24.5	17.7	18.1	25.2	24.4
Language and Task (Add and Subtract)	**20.7**	**30.6**	**24.6**	**29.6**	**22.5**	**35.4**	**21.8**	24.6	**18.5**	**20.9**	**25.8**	**25.0**

Table 1: **Language and task arithmetic improves zero-shot cross-lingual transfer on XLSum when we only have task data in En**. We show ROUGE-2 spm scores on XLSum$_{unseen}$. We train the task adapter using En XLSum data and the language adapter using Prefix-LM on mC4 data.

Method	Mr	Gu	Zh	Ne	Pt	Si	So	Vi	Yo	Uk	Fa	Avg
Task-in-Many-Languages												
Baseline (best)	21.2	31.2	25.6	28.4	22.5	35.8	22.1	25.6	21.4	21.6	25.3	25.5
Baseline (multilingual)	**21.4**	31.2	**26.4**	28.8	22.8	35.4	22.4	**25.7**	20.2	21.5	25.5	25.6
Task-only (Add all)	**21.4**	31.3	25.6	28.6	22.8	35.4	22.0	25.5	20.4	21.3	25.5	25.4
Task-only (Add related)	21.1	**31.5**	25.4	**30.2**	**23.1**	**36.3**	**22.9**	25.1	**22.9**	**21.8**	**25.7**	**26.0**
Language and Task (Add and Subtract related)	21.2	**31.5**	25.4	**30.4**	**23.0**	**36.4**	**22.8**	25.0	**22.9**	**21.7**	**25.7**	**26.0**

Table 2: **Addition of task adapters improves zero-shot cross-lingual transfer on XLSum when we have task data in multiple languages.** We show ROUGE-2 spm zero-shot scores on XLSum$_{unseen}$.

task arithmetic approach in cross-lingual summarization in Table 1. In the second row, we show the results by composing the language and task LoRA parameters via addition (*language and task; add*). This approach provides only slight improvements over the task adapter baseline in terms of ROUGE-2. Our language and task arithmetic approach with addition and subtraction (third row) consistently outperforms the baseline as well as the simple addition of source task and target language LoRA parameters. We highlight that the language adapters are trained by fine-tuning PaLM 2 with LoRA on prefix-LM for just $5k$ steps; even with this minimal training, they provide knowledge that is helpful to the pretrained model.

Why is subtracting the source language adapter important? We hypothesize that since the task adapter encodes information on summarizing articles in En (source), it is beneficial to add a language adapter that encourages the LLM to generate in the target language, but at the same time avoid generating in the source. Intuitively, negating the En language adapter parameters likely reduces the bias of the model towards En and enhances the ability of the model to generate in the target language.

4.2 Task-in-Many-Languages

We present the results of our approach when task data is available in different languages in Table 2. We compare the baselines with *task-only; Add all*, which fine-tunes PaLM 2 with LoRA on each language of the training set, and then computes the weight average of all fine-tuned models.

Task-only (Add all) on par with multilingual baseline: We observe that simply averaging all task adapters is on par with the multilingual baseline. This is intriguing, as it suggests that model merging can be used to iteratively add new task data to a petrained model. As soon as new task data (for a previously unsupported language) become available, one can simply train the corresponding task vector on this data and add it to the model by performing weight averaging. This alleviates the need of training a new multilingual model for every new batch of data.

Adding only related task adapters gives better results for most languages: Our approach *(task-only; Add related)* is presented in row 4. This selective composition of task adapters clearly surpasses the baselines. Our hypothesis is that not all task adapters are as important for a target language T and the final model should only incorporate task adapters trained in languages similar to the target. To select the models that will be averaged, we do not use any test data, but rely on linguistic information. We query the URIEL database and use the languages with the smallest distance to each held-out language T. Our approach outperforms the uniform weight average *(task-only; Add all)*, likely because our model avoids negative transfer between task adapters learned on distant languages, and leverages task information learned from similar languages.

Arithmetically composing language and task

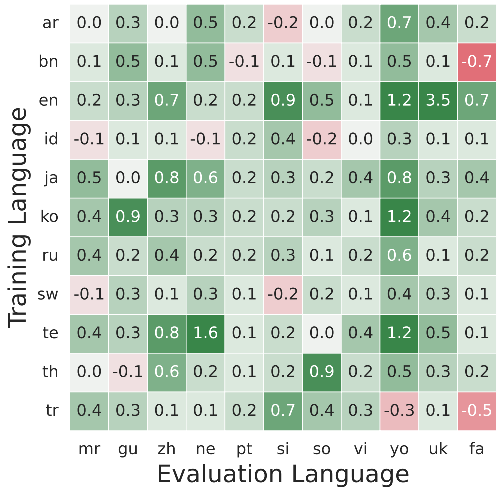

Figure 2: Relative ROUGE-2 improvement of our **language & task arithmetic** over the baseline (task adapter only). Our approach yields consistent improvements for most source-target language pairs.

adapters when task data is available in multiple languages is not helpful: We present the results we computed using *Language and Task; Add and Subtract related* which leverages unlabeled data as well as task data in the final row of Table 2. This approach performs on par with the *task-only; Add related* approach that uses only labeled data. Composing language and task knowledge is beneficial in the absence of enough task data. However, when task data is available in multiple languages, combining information from similar languages yields strong results and unlabeled data does not provide an additional benefit. Therefore, merging the two methods does not provide improvements.

5 Analysis

5.1 Using task adapter in different languages has consistent improvements

For our main language and task arithmetic results with *Task-in-One-Language*, we trained the task adapter on En labeled data and evaluated its performance on XLSum$_{unseen}$. For a more fine-grained assessment of our model, we present its relative performance when the task adapter is trained in each language in XLSum$_{seen}$ (as opposed to just En) against the corresponding baseline. The results are shown in Figure 2. The third row (En) shows the performance difference of *Language and Task (Add and Subtract)* from the baseline (Table 1).

We observe consistent improvements using our approach compared to the baseline across all language pairs. Low-resource languages, such as Yo, benefit more from the cross-lingual transfer setup we propose. In addition, while learning the En task adapter seems to provide higher gains for most evaluation languages, Te, Ja and Ko task adapters also lead to a large performance boost.

While PaLM 2 has been trained on vast multilingual data, providing each language with individual capacity using language modeling yields across-the-board improvements. This suggests that learning language-specific knowledge using PEFT parameters has the potential to strengthen the zero-shot cross-lingual transfer abilities of LLMs at a very small computational cost.

5.2 Our method also works with other PEFT parameters

We showed that composing task and language LoRA weights by element-wise arithmetic brings significant gains to cross-lingual transfer. In this section, we examine whether our findings also generalize to parameter-efficient fine-tuning methods other than LoRA.

One particularly interesting PEFT method is Kronecker adapter (Edalati et al., 2022). While LoRA is based on the multiplication of two low-rank matrices, Kronecker adapter is a matrix decomposition method which does not rely on the low-rank assumption. Instead, it replaces the low-rank decomposition in LoRA with the Kronecker product decomposition. It has been shown that this PEFT method achieves large improvements over LoRA and full fine-tuning on the GLUE benchmark (Wang et al., 2018). We conduct language and task arithmetic using Kronecker adapters as the PEFT modules.[3]

Kronecker adapter: Formally, the Kronecker product is defined as follows:

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}$$

where matrices $\mathbf{A} \in^{m\times n}$ and $\mathbf{B} \in^{\frac{k}{m}\times\frac{d}{n}}$ are the input matrices, and $\mathbf{W} \in^{k\times d}$, k is the model dimension and d is the dimension per attention head is the output matrix. We can tune hyperparameters m and n while keeping the number of additional parameters fixed, which is more flexible than LoRA.

[3]Similar to LoRA tuning, we add Kronecker adapters for the Key, Query, Value, Projection attention matrices of the Transformer model while keeping the weights fixed.

Method	Mr	Gu	Zh	Ne	Pt	Si	So	Vi	Yo	Uk	Fa	Avg
Task-in-Many-Languages												
Baseline (best)	21.3	31.4	25.6	30.0	22.6	36.0	22.9	25.4	21.8	22.0	25.7	25.9
Baseline (multilingual)	21.2	31.5	26.1	30.8	23.2	36.7	23.1	25.5	21.5	22.0	25.9	26.1
Task-only (Add all)	20.9	31.3	25.6	30.5	22.8	35.9	22.7	25.2	20.8	21.9	25.7	25.7
Task-only (Add related)	21.1	**32.2**	26.2	**31.4**	**24.0**	36.6	22.9	**25.7**	21.9	**22.3**	**26.6**	**26.4**

Table 3: **Adding related task adapters outperforms monolingual and multilingual baselines on XLSum using Kronecker adapter.** Rouge (ROUGE-2 spm) zero-shot scores on the XLSum$_{unseen}$ test set.

Experimental setting: We use PaLM 2 S model as the pretrained LLM. We add a Kronecker adapter with $(m, n) = (32, 16)$. Similar to LoRA, this PEFT method does not decrease inference speed because the additional parameters are added back to the original model weights.

Results: We run the *task-only; Add* experiments using Kronecker adapter and show the results in Table 3. We observe that the results follow a similar pattern as with the LoRA adapter. Our method (*task-only; Add related*) outperforms monolingual and multilingual baselines. This demonstrates that a selective combination of PEFT parameters at the weight level improves the generalization ability of a LLM to languages for which no task data is available. This confirms our intuition that it is possible to compose information learned about a task in different languages by simply performing point-wise operations.

5.3 Module subtraction is particularly helpful for summarization

We proposed two composition approaches for language and task arithmetic: *Add* or *Add and Subtract*. To understand the different impact of these two approaches, we compare their performance on two datasets, TyDi QA and XLSum.

Experimental setting: Besides XLSum, we also evaluate our language and task arithmetic approach on TyDi QA (Clark et al., 2020), a multilingual extractive question answering dataset of 8 typologically diverse languages, based on Wikipedia articles in Bengali (bn), English (en), Finnish (fi), Indonesian (id), Korean (ko), Russian (ru), Swahili (sw), and Telugu (te). We train our model on En task data an evaluate on each of the other languages in the dataset, simulating a zero-shot setup.

Results: We show the results in Table 4. We find that using both addition and subtraction is more beneficial than addition only for XLSum (+0.6 gains in ROUGE). However, we observe that for the QA task, using addition and subtraction performs on par with addition only. We hypothesize that this is likely because TyDi QA is an extractive QA task where the model simply needs to copy a segment of correct answer from the context, while XLSum requires more free-form language generation. Because of this inherent difference between the tasks, discouraging the model from generating in the source language (by negating the source language adapter) is less essential to QA compared to summarization.

Method	TyDi QA	XLSum
Baseline	83.0	24.2
Language and task arithmetic		
- Add	83.3	24.4
- Add and Subtract	83.2	25.0

Table 4: **Language and task arithmetic via addition or addition and subtraction for TyDi QA and XLSum** using LoRA parameters. These are the average results over the unseen languages. For TyDi QA, F1 is shown, while for XLSum, we show ROUGE-2 spm.

5.4 Task adapters selected by *lang2vec*

When we have labeled data available in multiple languages, our proposed *task-only; Add related* approach averages the weights of PEFT parameters that are related to the target language. The relatedness is defined by *lang2vec*, a tool that queries URIEL. To shed light on where the improved performance of our model comes from, we present in Table 5 the source languages that are selected for each of the target languages based on linguistic knowledge.

We witness that a different number of languages is selected for each target language. We do not explicitly control the number of models averaged, we simply sort them using the syntactic and geographic distance. For a given target language T, we average the weights of the source languages

Mr	Gu	Zh	Ne	Pt	Si	So	Vi	Yo	Uk	Fa
Bn	Bn	En	Te	En	Te	Ar	Id	En	Ru	Tr
Te	Te	Ko	Ja	Ru	Bn	Sw	Th	Ar	En	En
Tr		Ja	Tr	Ar		En			Sw	Ar
		Id	Ko							
		Th	Ru							
			Bn							

Table 5: Most similar languages to each of the evaluation languages (based on lang2vec) selected by our *task-only (Add related)* approach.

$S_1, S_2, .., S_N$ that have a syntactic distance < 0.7 and a geographic distance < 0.3. We leave a more fine-grained selection process to future work.

6 Related Work

LLMs have shown impressive performance in various natural language processing tasks (Radford et al., 2019; Brown et al., 2020; Chung et al., 2022; Touvron et al., 2023), often requiring no extra training to adapt to downstream tasks.

Numerous parameter-efficient methods have been proposed, each addressing the challenge of enhancing efficiency . These methods can be categorized as input composition, function composition, and parameter composition (Pfeiffer et al., 2023). *Input composition* methods, such as prompt tuning, incorporate soft prompts into the input layers to guide the model's behavior (Li and Liang, 2021; Lester et al., 2021). *Function composition* strategies, like adapters (Rebuffi et al., 2017; Houlsby et al., 2019), introduce non-linear functions within pretrained layers to adapt the intermediate representations of the model. *Parameter composition* is exemplified by methods like LoRA (Hu et al., 2022), which introduces a limited number of learnable low-rank matrices into each pretrained layer.

Recent work which is based on the linear mode connectivity (Frankle et al., 2020) suggests averaging the weights of pretrained models fine-tuned on the same dataset with different hyperparameters to improve downstream performance (Izmailov et al., 2018; Gupta et al., 2020; Wortsman et al., 2022). It has also been shown that averaging the weights of models fine-tuned on different tasks improves out-of-domain generalization without leaking information about potentially private labeled datasets (Jin et al., 2023). Composing weights of models fine-tuned on tasks related to the target task is also beneficial (Matena and Raffel, 2021). Ainsworth et al. (2023); Ilharco et al. (2023); Yadav et al. (2023); Huang et al. (2023); Ortiz-Jimenez et al. (2023) show that a model can acquire multi-task learning abilities using model merging, while Daheim et al. (2024) propose model merging by reducing gradient mismatch. There is also work on averaging domain-specific adapter layers (Chronopoulou et al., 2023a) or domain-expert LMs (Li et al., 2022b) with large gains for unseen domains. However, there is no work on PEFT cross-lingual transfer using language and task arithmetic.

In a similar line of thought and to mitigate interference of different tasks during training, Pfeiffer et al. (2021) train task PEFT modules and learn attention parameters to select the most useful of them, while Karimi Mahabadi et al. (2021) learn adapters with hypernetworks. Asai et al. (2022) efficiently integrate knowledge from multiple tasks with a mix of trainable soft prompts. Ponti et al. (2023) propose Polytropon, which learns both adapters and a binary task–module routing matrix, determining which module should be active for each task; Caccia et al. (2023) extend it to a more granular level by mixing subsets of adapter dimensions.

Another research direction considers training PEFT parameters and combining them for cross-lingual transfer. MAD-X (Pfeiffer et al., 2020) stacks task bottleneck adapters with language adapters and using them for cross-lingual transfer. Ansell et al. (2022) identify the parameters that are most useful for a task and a language, and compose them; this work is based on the lottery ticket hypothesis (Frankle et al., 2020). Vu et al. (2022) propose factorizing a prompt into a language and task and training each part while keeping the other frozen. Newly learned knowledge is combined with the existing model using PEFT modules to permit cross-lingual transfer in multiple recent works (Bapna and Firat, 2019; Üstün et al., 2020; Vidoni et al., 2020; Cooper Stickland et al., 2021; Chronopoulou et al., 2023b). To the best of our knowledge, our work is the first to propose improving cross-lingual transfer of a LLM via a combination of weights of PEFT parameters.

7 Conclusion

We present a new method to compose knowledge from parameter-efficient modules using arithmetic operations in order to improve zero-shot cross-lingual transfer. Our experiments in summarization on a wide set of languages using PaLM 2 as the pretrained model show that our *language and task* *arithmetic* achieves consistent improvements over the baselines and introduces a modular approach that can be leveraged for improved generalization of a LLM in languages that lack labeled data.

References

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. 2023. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Alan Ansell, Edoardo Ponti, Anna Korhonen, and Ivan Vulić. 2022. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796, Dublin, Ireland. Association for Computational Linguistics.

Akari Asai, Mohammadreza Salehi, Matthew Peters, and Hannaneh Hajishirzi. 2022. ATTEMPT: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6655–6672, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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, Hong Kong, China. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Lucas Caccia, Edoardo Ponti, Zhan Su, Matheus Pereira, Nicolas Le Roux, and Alessandro Sordoni. 2023. Multi-head adapter routing for cross-task generalization.

Yue Cao, Xiaojun Wan, Jinge Yao, and Dian Yu. 2020. Multisumm: Towards a unified model for multilingual abstractive summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):11–18.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Alexandra Chronopoulou, Matthew Peters, Alexander Fraser, and Jesse Dodge. 2023a. AdapterSoup: Weight averaging to improve generalization of pretrained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2054–2063, Dubrovnik, Croatia. Association for Computational Linguistics.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2023b. 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, Dubrovnik, Croatia. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Asa Cooper Stickland, Xian Li, and Marjan Ghazvininejad. 2021. Recipes for adapting pre-trained monolingual and multilingual models to machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3440–3453, Online. Association for Computational Linguistics.

Nico Daheim, Thomas Möllenhoff, Edoardo Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. 2024. Model merging by uncertainty-based gradient matching. In *The Twelfth International Conference on Learning Representations*.

Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J. Clark, and Mehdi Rezagholizadeh. 2022. Krona: Parameter efficient tuning with kronecker adapter.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. 2021. Efficiently identifying task groupings for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 27503–27516. Curran Associates, Inc.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. 2020. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3259–3269. PMLR.

George Giannakopoulos, Jeff Kubina, John Conroy, Josef Steinberger, Benoit Favre, Mijail Kabadjov, Udo Kruschwitz, and Massimo Poesio. 2015. MultiLing 2015: Multilingual summarization of single and multi-documents, on-line fora, and call-center conversations. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 270–274, Prague, Czech Republic. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Vipul Gupta, Santiago Akle Serrano, and Dennis DeCoste. 2020. Stochastic weight averaging in parallel: Large-batch training that generalizes well. In *ICLR*.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 2790–2799.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. Conference on Uncertainty in Artificial Intelligence (UAI), 2018.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. 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 565–576, Online. Association for Computational Linguistics.

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

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2022a. Branch-train-merge: Embarrassingly parallel training of expert language models.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2022b. Branch-train-merge: Embarrassingly parallel training of expert language models.

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

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

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Michael Matena and Colin Raffel. 2021. Merging models with fisher-weighted averaging.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*, pages 103–233.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. 2023. Task arithmetic in the tangent space: Improved editing of pre-trained models.

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

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulic, and Edoardo Maria Ponti. 2023. Modular deep learning. *arXiv preprint*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Edoardo Maria Ponti, Alessandro Sordoni, Yoshua Bengio, and Siva Reddy. 2023. Combining parameter-efficient modules for task-level generalisation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 687–702, Dubrovnik, Croatia. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Marko Vidoni, Ivan Vulić, and Goran Glavaš. 2020. Orthogonal language and task adapters in zero-shot cross-lingual transfer.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. 2022. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9279–9300, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*.

David Yunis, Kumar Kshitij Patel, Pedro Henrique Pamplona Savarese, Gal Vardi, Jonathan Frankle, Matthew Walter, Karen Livescu, and Michael Maire. 2022. On convexity and linear mode connectivity in neural networks. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. 2023a. Composing parameter-efficient modules with arithmetic operations. In *Advances in Neural Information Processing Systems*.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. 2023b. Llama-adapter: Efficient fine-tuning of language models with zero-init attention.

Method	Ar	Bn	En	Id	Ja	Ko	Ru	Sw	Te	Th	Tr	Avg
LoRA	23.4	27.6	23.5	25.0	33.6	30.4	21.3	27.1	26.9	24.7	25.3	26.2
Multi-LoRA	23.0	27.8	22.5	24.6	34.0	30.4	20.8	27.1	27.8	25.1	24.9	26.2
Kronecker	23.4	27.7	23.1	24.8	34.6	31.2	21.6	27.1	27.4	24.8	25.2	26.4
Multi-Kronecker	22.8	27.5	22.5	24.9	34.7	31.2	20.8	27.5	27.6	24.8	25.2	26.3
Full fine-tuning	23.9	28.1	22.6	25.3	34.8	30.4	21.8	27.0	28.2	24.6	25.4	26.6

Table 6: **Parameter-efficient fine-tuning vs Full fine-tuning**. Rouge (ROUGE-2 spm) in-domain scores on the XLSum$_{seen}$ test set.

A Appendix

A.1 Are PEFT methods competitive to full fine-tuning of PaLM 2?

We present the performance of LoRA and Kronecker, two PEFT methods, when used to fine-tune PaLM 2 on summarization in 11 languages of XL-Sum in Table 6. We compare their performance to full fine-tuning of PaLM 2.

Fine-tuning the model with LoRA results in summarization scores that are only 0.4 ROUGE points below full fine-tuning, while fine-tuning with Kronecker provides a performance similar to full fine-tuning (i.e., just 0.2 points worse than full fine-tuning). Based on this finding, we conclude that using PEFT methods to fine-tuning PaLM 2, a state-of-the-art LLM, is largely impactful, as in our experiments LoRA for example trains only 0.2% of the model's parameters whereas fully tuning the LLM requires updates on 100% of the model's parameters.

A.2 XLSum$_{\text{seen}}$ Dataset

We are showing the dataset sizes of XLSum$_{\text{seen}}$ in Table 7.

Language	Lang code	Dataset size
Arabic	ar	38k
Bengali	bn	8k
English	en	306k
Indonesian	id	38k
Japanese	ja	7k
Korean	ko	4k
Russian	ru	62k
Swahili	sw	8k
Telugu	te	10k
Thai	th	7k
Turkish	tr	27k

Table 7: Languages in XLSum seen and dataset sizes (training).