S^4-Tuning: A Simple Cross-lingual Sub-network Tuning Method

Anonymous ACL submission

Abstract

The emergence of multilingual pre-trained language models makes it possible to adapt to target languages with only few labeled examples. However, vanilla fine-tuning tends to achieve degenerated and unstable results, owing to the *Language Interference* among different languages, and *Parameter Overload* under the few-sample transfer learning scenarios. To address two problems elegantly, we propose S^4-Tuning, a Simple Cross-lingual Sub-network Tuning method. S^4-Tuning first detects the most essential sub-network for each target language, and only updates it during fine-tuning. In this way, the language sub-networks lower the scale of trainable parameters, and hence better suit the low-resource scenarios. Meanwhile, the commonality and characteristics across languages are modeled by the overlapping and non-overlapping parts to ease the interference among languages. Simple but effective, S^4-Tuning gains consistent improvements over vanilla fine-tuning on three multilingual tasks involving 37 different languages in total (XNLI, PAWS-X, and Tatoeba).

1 Introduction

Recently, a variety of multilingual pre-trained language models (PLMs) have been proposed, including mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020). Based on these PLMs, it is possible to adapt the model to specific target languages, with only a handful of labeled examples in the downstream tasks, which is called *few-shot cross-lingual transfer learning* (Lauscher et al., 2020; Hedderich et al., 2020; Bari et al., 2021).

However, traditional fine-tuning tends to obtain degenerated and unstable results, due to the following two challenges. (1) **Parameter Overload**: Given only few labeled data for a target language, it is challenging to update all model parameters, and such a mismatch between the scale of data and trainable parameters can cause overfitting (Dodge et al., 2020; Zhao et al., 2021). (2) **Language Interference**: Sharing commonality though, different languages also possess their own characteristics. Hence, the adaption towards a specific target language can interfere with that of other languages (Lin et al., 2021), which also damages the transfer performance.

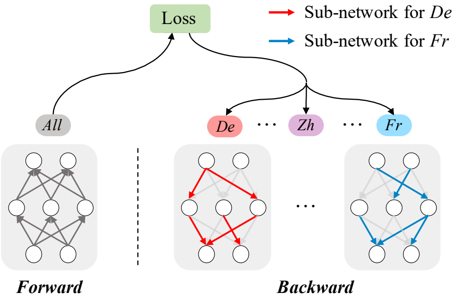


Figure 1: Utilizing full network during the forward process (left), S^4-Tuning only updates a specific sub-network according to the language of the input example (right). Sub-networks are detected based on the importance of model parameters towards different languages.

Therefore, it is natural to ask the question, *How to address the Parameter Overload and Language Interference problem* ***elegantly***? In this paper, we propose a Simple Cross-lingual Sub-network Tuning method, S^4-Tuning, which tries to deal with these two problems jointly. As shown in Figure 1, S^4-Tuning detects the most fundamental language sub-networks (with a simple and intuitive criterion in Sec. 3.2), and only updates the specific sub-network corresponding to the input language during training. For one thing, we update the language sub-network on a matching scale, which better suits the low-resource scenarios and addresses the *Parameter Overload* problem. For another, the commonality across languages is modeled by the overlap among different language sub-networks, while the characteristics are also allowed by the non-overlapping parts. With such a better trade-off, the *Language Interference* problem is alleviated.

Simple to implement, S^4-Tuning also reveals evident effectiveness in the downstream tasks in our experiments. Compared with vanilla fine-tuning, S^4-Tuning consistently offer improvements across different multi-lingual downstream tasks. For example, it improves by 0.9 and 5.6 average points on XNLI and Tatoeba tasks, respectively.

2 Related Work

Towards better few-shot cross-lingual transfer, Zhao et al. (2021) freeze the embedding and encoder layers of the PLM during fine-tuning, which is not effective and flexible enough. Nooralahzadeh et al. (2020) adopt the traditional meta-learning method MAML (Finn et al., 2017), but it is not practical enough, since it requires extra abundant labeled data for meta-training. Differently, we try a more elegant and effective way to handle the *Parameter Overload* and *Language Interference* problem through language sub-networks.

Some works also find a sub-network for each language pair in machine translation (Lin et al., 2021; Xie et al., 2021), or each task in multi-task learning (Sun et al., 2020; Liang et al., 2021). However, their forward and backward are both based on sub-networks, which is more like ***pruning***. Instead, we update parameters within the sub-network during the backward process, but still forward on the whole network to fully utilize the knowledge stored in the entire model. Our work most closely resembles the work of Xu et al. (2021). However, S^4-Tuning deals with multiple sub-networks simultaneously rather than a single sub-network in more challenging few-shot multi-lingual scenarios, and adopts different criteria for language sub-network detection. We empirically show the superiority of S^4-Tuning in Figure 3 in Section 4.5.

3 S^4-Tuning: Simple CroSS-lingual Sub-network Tuning

We formally present the problem formulation (Sec. 3.1). Then we introduce our proposed method, S^4-Tuning, which firstly detects the most important sub-network for each target language (Sec. 3.2), and then only updates the corresponding sub-network during the backward process (Sec. 3.3).

3.1 Problem Formulation

Given a specific task, the original multilingual PLM θ_{pre} is firstly fine-tuned on rich-resource labeled data $\mathcal{D}_s = (\mathcal{X}_s, \mathcal{Y}_s)$ in source language s to obtain θ_s (*source training*) following Lauscher et al. (2020). Then, we aim to better adapt θ_s to multiple target languages $\mathcal{T} = \{t_1, t_2, \dots, t_{\|\mathcal{T}\|}\}$ with target labeled data $\mathcal{D}_\mathcal{T} = \{(\mathcal{X}_\text{t}, \mathcal{Y}_\text{t}) \mid t \in \mathcal{T}\}$ (*target adapting*). Specifically, suppose there are $\mathcal{C}$ different classes, we have K training examples for each class $c \in \mathcal{C}$ in target language t, and K is remarkably small in low-resource scenarios, leading to $|\mathcal{D}_s| \gg |\mathcal{D}_\mathcal{T}|$. In our paper, we use English as source language following Lauscher et al. (2020).

3.2 Language Sub-network Detection

In this section, we aim to identify the most important sub-network for each target language. In detail, for target language t, if parameter h_i is essential to language t, the change of loss would be large once we remove h_i (i.e., $h_i = 0$) (Molchanov et al., 2017), which is shown in Equation 1 and H refers to other parameters excluding h_i.

$$\Omega^t(h_i) = \left|\mathcal{L}^t(H, h_i = 0) - \mathcal{L}^t(H, h_i)\right| \qquad (1)$$

Following Molchanov et al. (2017), we approximate with Taylor Expansion, and obtain Eq. 2.

$$\Omega^t(h_i) = \left|\frac{\partial \mathcal{L}^t(H, h_i)}{\partial h_i} h_i\right| \qquad (2)$$

Though different scoring criteria can be used, we find this one works best. After deriving the importance score of parameters for target language t based on $(\mathcal{X}_t, \mathcal{Y}_t)$, parameters with the highest score are selected as the sub-network for t. It can be indicated by a mask M_t, where $M^t(h_i) = 1$ if h_i belongs to the sub-network, and $M^t(h_i) = 0$ otherwise. With N parameters in total, we can set up sub-network scale by $p_t = \frac{\sum_{i=1}^{N} M^t(h_i)}{N}$. We unify p_t across different languages as p, that is, $p = p_1 = p_2 = \cdots = p_{\|\mathcal{T}\|}$.

3.3 Constrained Language Adaption

According to the distinctive patterns of language sub-networks, we adapt to the target languages with their most essential parameters.

Forward During the forward procedure, we encode instances by the *full network* regardless of its language. In this way, we can better make full use of the knowledge contained in the whole model.

Backward Different from vanilla fine-tuning, we only update the parameters within the significant *language sub-network*. It can be achieved by multiplying the gradients with the mask M^t. By this

Method	ar	bg	de	el	en	es	fr	hi	ru	sw	th	tr	ur	vi	zh	Avg
								K=64								
FC Only	77.43	82.36	82.28	81.51	88.84	83.93	82.48	76.08	79.30	71.55	76.38	78.55	72.37	78.94	78.27	79.35±0.03
FC+Pooler	77.50	82.55	82.44	81.75	88.94	84.20	82.69	76.25	79.75	71.84	76.83	78.96	72.59	79.30	78.64	79.62±0.07
Full Model	78.77	83.73	83.05	81.98	88.32	84.16	83.05	76.67	80.54	72.35	77.42	79.65	73.45	80.10	79.34	80.17±0.53
S^4-Tuning (Ours)	**79.26**	**84.01**	**83.64**	**82.55**	**89.10**	**84.87**	**83.63**	**77.94**	**81.06**	**73.24**	**78.11**	**80.21**	**74.28**	**80.59**	**80.18**	**80.84±0.16**
								K=128								
FC Only	77.97	83.01	82.70	81.99	89.04	84.62	82.99	76.63	80.11	72.49	77.13	79.25	73.23	79.54	79.41	80.01±0.02
FC+Pooler	78.06	83.07	82.78	82.10	89.08	**84.66**	**83.15**	76.70	80.17	72.79	77.44	79.44	73.31	79.85	79.46	80.14±0.11
Full Model	78.80	83.61	83.23	82.31	88.43	83.95	82.91	77.01	80.62	72.66	77.65	79.50	73.58	80.29	80.00	80.30±0.28
S^4-Tuning (Ours)	**79.70**	**84.43**	**84.04**	**82.90**	**89.08**	84.61	83.75	**77.93**	**81.38**	**73.67**	**79.03**	**80.47**	**74.64**	**81.24**	**81.13**	**81.20±0.04**

Table 1: **Comparison with other fine-tuning methods on XNLI**. S^4-Tuning consistently outperforms other methods under different K settings, and also achieves lower standard deviation compared with *Full Model* tuning. Although with low standard deviation, *FC Only* and *FC+Pooler* yield inferior results.

means, we lower the scale of trainable parameters to address *Parameter Overload*, and maintain the commonality and characteristics across different languages to handle *Language Interference*.

4 Experiments

4.1 Datasets

We conduct experiments on three multilingual tasks. Cross-lingual Natural Language Inference (XNLI) (Conneau et al., 2018) is a natural language inference task involving 15 different languages. Besides, Cross-lingual Paraphrase Adversaries from Word Scrambling (PAWS-X) (Yang et al., 2019) focuses on determining whether two sentences are paraphrases with 7 languages. Tatoeba (Artetxe and Schwenk, 2019) with 37 languages is a cross-lingual sentence retrieval task, which finds the nearest neighbor based on cosine similarity between multilingual representations of sentences.

4.2 Experimental Setups

Experiments are based on XLM-R$_{\text{large}}$ (Conneau et al., 2020). Following Zhao et al. (2021), we firstly fine-tune the PLM for 10 epochs with batch size 32 on full English labeled examples for *source-training*, whose results are comparable to Hu et al. (2020) (details in Appendix A). Then we continue to fine-tune 5 epochs on K-shot data over target languages, and we use $K \in \{64, 128\}$. The translated examples provided by Hu et al. (2020) are used as the training data for target languages. We search learning rate from $\{5\text{e-}6, 8\text{e-}6, 1\text{e-}5, 3\text{e-}5\}$, and p from $\{0.1, 0.3, 0.5\}$. We report the average score on the test set of 5 runs with different seeds.

4.3 Main Results

Besides vanilla ***Full Model*** fine-tuning, we also compare with two strong baselines (Zhao et al., 2021): 1) ***FC Only***: Only update the linear classifier during training. 2) ***FC+Pooler***: Only update the linear classifier and pooler layer during training.

S^4-Tuning helps the model better adapt to target languages with strong and stable performance. As shown in Table 1, S^4-Tuning outperforms other fine-tuning methods on XNLI. For example, compared with *Full Model* tuning, S^4-Tuning yields an improvement of up to 0.90 average points, and the standard deviation of multiple random runs is also lowered, suggesting more stable performance. Although with lower standard deviation, *FC Only* and *FC+Pooler* reveal inferior performance. Similar results are observed on PAWS-X task (shown in Appendix B due to limited space), in which S^4-Tuning also beat other methods on both $K = 64$ and $K = 128$ settings, e.g., outperforms *Full Model* tuning by 0.7 average points when $K = 64$.

S^4-Tuning strengthens the model ability to capture cross-lingual semantics, thanks to more precise and flexible adaption for different target languages. We adopt models fine-tuned on PAWS-X through different methods, and search the best encoder layer to derive multilingual sentence representations for Tatoeba task. The most semantically similar sentence is retrieved directly with cosine similarity between representations. As shown in Table 2, S^4-Tuning yields an improvement of up to 5.64 average points across 36 target languages, in comparison with vanilla *Full Model* tuning.

4.4 Similarity Between Sub-networks

In this section, we aim to understand the intrinsic relations among different language sub-networks. Specifically, we explore the similarity using the Jaccard similarity coefficient to quantify the overlapping ratio between two sub-networks. Figure 2

Method	**ar**	**he**	**vi**	**id**	**jv**	**tl**	**eu**	**ml**	**ta**	**te**	**af**	**nl**	**de**	**el**	**bn**	**hi**	**mr**	**ur**	**Avg**
K=64																			
FC Only/FC+Pooler*	46.7	63.8	73.0	79.2	16.1	36.3	36.4	65.9	26.7	38.5	61.0	82.1	89.0	60.5	43.8	71.5	53.5	25.3	58.5
Full Model	48.8	65.6	76.4	79.8	17.7	38.5	39.5	66.4	31.1	43.5	61	82.6	89.9	61.4	44.4	72.7	55.2	30.8	60.5
S^4-Tuning (Ours)	**55.6**	**69.0**	**81.8**	**82.6**	**20.3**	**44.0**	**46.8**	**71.8**	**43.3**	**55.0**	**67.0**	**84.7**	**92.4**	**66.7**	**52.5**	**76.6**	**59.2**	**49.6**	**66.1**
K=128																			
FC Only/FC+Pooler*	46.7	63.8	73.0	79.2	16.1	36.3	36.4	65.9	26.7	38.5	61.0	82.1	89.0	60.5	43.8	71.5	53.5	25.3	58.5
Full Model	55.5	69.0	82.6	83.6	21.4	42.3	44.9	76.1	38.1	51.9	67.2	85.7	92.6	67.2	51.7	79.6	63.2	43.6	66.2
S^4-Tuning (Ours)	**58.2**	**71.4**	**85.1**	**86.1**	**23.0**	**47.8**	**50.4**	**74.9**	**46.5**	**58.3**	**70.0**	**87.8**	**93.6**	**70.4**	**56.3**	**81.4**	**65.5**	**51.3**	**69.5**

Table 2: **Comparison with other fine-tuning methods on cross-lingual retrieval task Tatoeba** across 36 languages. We only list 18 languages due to limited space, and the complete results are provided in Appendix D. S^4-Tuning consistently achieves the best performance across different target languages. *: Same as the result of the model after source training (θ_s), since these two methods do not update the encoder layers of the model.

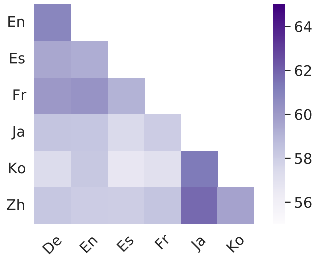
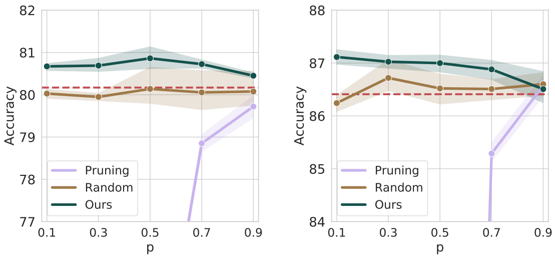


Figure 2: The overlapping ratio between sub-networks of different languages.



(a) XNLI

(b) PAWS-X

Figure 3: Compare S^4-Tuning with **Pruning** and **Random** sub-network across various sub-network ratio p. The red horizontal line denotes the result of vanilla full model tuning. S^4-Tuning reveals superior performance over other strategies.

illustrates the results based on PAWS-X experiments with $K = 128$ and $p = 0.5$ settings, It can be observed that the eastern languages (*Ja*, *Ko*, *Zh*) are similar to each other, while different from the western languages (*De*, *En*, *Es*, *Fr*). For example, the sub-network of Japanese (*Ja*) is much more similar to that of Korean (*Ko*) and Chinese (*Zh*) than others. It suggests that the detected sub-networks potentially capture the inductive bias of language similarity, and model their commonality and characteristics through overlapping and non-overlapping parts flexibly.

4.5 Comparison with Different Sub-network Strategies: Pruning and Random

To further understand the effect of S^4-Tuning, we compare with two sub-network strategies in XNLI and PAWS-X with $K = 64$: 1) **Pruning** (Lin et al., 2021; Xie et al., 2021): both forward and backward are through a pruned sub-network (while S^4-Tuning uses the full network for forward). We adopt Equation 2 as the criterion to prune the model for all target languages. 2) **Random**: the sub-networks are detected randomly for S^4-Tuning rather than following a specific criterion.

As shown in Figure 3, for pruning, the model would collapse if $p < 0.7$, and the best score achieved in $p = 0.9$ is still lower than the vanilla fine-tuning in XNLI. The performance of random sub-network is slightly lower than vanilla fine-tuning in XNLI, while slightly higher in PAWS-X. Compared with these two strategies, S^4-Tuning achieves the best scores in an overwhelming majority of cases, which suggests the superiority of S^4-Tuning in few-shot cross-lingual transfer.

5 Conclusion

Towards better few-shot cross-lingual transfer learning, we propose S^4-Tuning. S^4-Tuning detects the most essential sub-network for each target language, and only updates these parameters during the backward process, while still utilizing the full model for the forward process. In this way, we reduce the scale of trainable parameters that better suits low-resource scenarios to address overfitting, and better deal with the interference across languages. Our experiments show that S^4-Tuning consistently outperforms other fine-tuning methods in different downstream tasks.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics (TACL)*.

M. Saiful Bari, Batool Haider, and Saab Mansour. 2021. Nearest neighbour few-shot learning for cross-lingual classification. *arXiv*, arXiv:2109.02221.

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

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv*, arXiv:2002.06305.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Michael A. Hedderich, David Adelani, Dawei Zhu, Jesujoba Alabi, Udia Markus, and Dietrich Klakow. 2020. Transfer learning and distant supervision for multilingual transformer models: A study on African languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chen Liang, Simiao Zuo, Minshuo Chen, Haoming Jiang, Xiaodong Liu, Pengcheng He, Tuo Zhao, and Weizhu Chen. 2021. Super tickets in pre-trained language models: From model compression to improving generalization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. Learning language specific sub-network for multilingual machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistic (ACL)*.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations (ICLR)*.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Tianxiang Sun, Yunfan Shao, Xiaonan Li, Pengfei Liu, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. Learning sparse sharing architectures for multiple tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Wanying Xie, Yang Feng, Shuhao Gu, and Dong Yu. 2021. Importance-based neuron allocation for multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. 2021. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze. 2021. A closer look at few-shot crosslingual transfer: The choice of shots matters. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

A Results on Source Training

Since our work focuses on the *target adapting*, we ensure the results on *source training* are comparable to others. As shown in Table 3, the obtained results based on our implementation is comparable or even better than those of Hu et al. (2020) in three multi-lingual tasks.

	PAWS-X	XNLI	Tatoeba
Hu et al. (2020)	86.4	79.2	57.3
Ours	86.4	79.6	58.5

Table 3: Align initial results after source training.

B Results on PAWS-X

Table 4 illustrates the results of different fine-tuning methods on PAWS-X task. Compared with vanilla full model tuning, S^4-Tuning achieves better performance with lower standard deviation, which suggests that S^4-Tuning helps the model better adapt to target languages and obtain more stable results.

C Detailed Results on Tatoeba

Table 5 demonstrates the results on the cross-lingual retrieval task, Tatoeba, across 36 different target languages in total. Since *FC Only* and *FC+Pooler* do not update the intermediate encoder layers, their results are both the same as that of the model after source training. It can be observed that S^4-Tuning outperform other methods by 5.6 $\sim$ 7.6 average points under $K = 64$ setting, and 3.2 $\sim$ 11.0 average points under $K = 128$ setting.

D Results on XQuAD

We also explore S^4-Tuning in multilingual question answering task, XQuAD (Artetxe et al., 2020). As shown in Table 6, S^4-Tuning provides improvements on both $K = 64$ and $K = 128$ settings, along with lower standard deviation.

Method	de	en	es	fr	ja	ko	zh	Avg
K=64								
FC Only	89.07	94.07	90.26	89.70	**80.33**	79.03	82.81	86.47±0.05
FC + Pooler	89.34	93.90	90.05	89.41	80.12	79.42	82.77	86.43±0.07
Full Model	88.80	93.88	89.52	89.35	79.50	**80.78**	**83.04**	86.41±0.70
S^4-Tuning (Ours)	**90.13**	**94.53**	**90.69**	**90.41**	79.96	80.86	83.22	**87.11±0.16**
K=128								
FC Only	89.46	94.37	90.38	89.90	80.73	79.31	82.93	86.73±0.07
FC + Pooler	89.54	94.19	90.29	89.72	80.32	79.67	82.96	86.67±0.06
Full Model	89.19	94.54	90.85	90.43	80.21	80.93	83.23	87.05±0.41
S^4-Tuning (Ours)	**90.19**	**95.01**	**91.13**	**90.75**	**80.85**	**81.71**	**83.56**	**87.60±0.20**

Table 4: **Comparison with other fine-tuning methods on PAWS-X**. S^4-Tuning achieves the best average score across different languages, and also lower the standard deviation compared with *Full Model* tuning.

Method	ar	he	vi	id	jv	tl	eu	ml	ta	te	af	nl	de	el	bn	hi	mr	ur	fa
K=64																			
FC Only/FC+Pooler*	46.7	63.8	73.0	79.2	16.1	36.3	36.4	65.9	26.7	38.5	61.0	82.1	89.0	60.5	43.8	71.5	53.5	25.3	71.9
Full Model	48.8	65.6	76.4	79.8	17.7	38.5	39.5	66.4	31.1	43.5	61.0	82.6	89.9	61.4	44.4	72.7	55.2	30.8	73.4
S^4-Tuning (Ours)	**55.6**	**69.0**	**81.8**	**82.6**	**20.3**	**44.0**	**46.8**	**71.8**	**43.3**	**55.0**	**67.0**	**84.7**	**92.4**	**66.7**	**52.5**	**76.6**	**59.2**	**49.6**	**77.7**
K=128																			
FC Only/FC+Pooler*	46.7	63.8	73.0	79.2	16.1	36.3	36.4	65.9	26.7	38.5	61.0	82.1	89.0	60.5	43.8	71.5	53.5	25.3	71.9
Full Model	55.5	69.0	82.6	83.6	21.4	42.3	44.9	**76.1**	38.1	51.9	67.2	85.7	92.6	67.2	51.7	79.6	63.2	43.6	78.8
S^4-Tuning (Ours)	**58.2**	**71.4**	**85.1**	**86.1**	**23.0**	**47.8**	**50.4**	74.9	**46.5**	**58.3**	**70.0**	**87.8**	**93.6**	**70.4**	**56.3**	**81.4**	**65.5**	**51.3**	**80.7**

Method	fr	it	pt	es	bg	ru	ja	ka	ko	th	sw	zh	kk	tr	et	fi	hu	Avg
K=64																		
FC Only/FC+Pooler*	75.9	69.3	83.0	77.4	72.1	74.4	63.5	53.1	60.6	35.0	21.5	68.9	49.6	69.3	52.9	70.3	66.7	58.5
Full Model	77.0	71.6	82.9	79.5	73.0	76.3	65.7	53.8	64.9	39.9	24.0	70.3	48.7	71.8	56.9	74.1	68.7	60.5
S^4-Tuning (Ours)	**79.3**	**73.7**	**83.8**	**82.0**	**76.5**	**80.0**	**74.3**	**56.0**	**69.4**	**59.7**	**25.7**	**76.4**	**53.7**	**75.8**	**62.3**	**80.3**	**75.1**	**66.1**
K=128																		
FC Only/FC+Pooler*	75.9	69.3	83.0	77.4	72.1	74.4	63.5	53.1	60.6	35.0	21.5	68.9	49.6	69.3	52.9	70.3	66.7	58.5
Full Model	80.9	75.0	86.4	83.4	77.3	80.7	73.7	56.9	70.7	54.1	25.2	78.4	54.5	77.6	61.7	79.9	75.2	66.3
S^4-Tuning (Ours)	**83.3**	**77.6**	**87.1**	**85.6**	**81.3**	**83.3**	**76.0**	**63.5**	**73.3**	**61.0**	**28.4**	**80.6**	**58.7**	**80.3**	**66.2**	**82.0**	**76.8**	**69.5**

Table 5: **Detailed results on cross-lingual retrieval task Tatoeba** across 36 languages. S^4-Tuning outperforms vanilla *Full Model* tuning under a overwhelming majority of cases. *: Same as the result of the model after source training (θ_s), since these two methods do not update the encoder layers of the model.

Method	en	es	de	el	ru	tr	ar	vi	th	zh	hi	Avg
K=64												
Full Model	**72.40**	59.14	**60.91**	56.45	**60.30**	**56.27**	53.53	56.79	68.2	**56.22**	57.82	59.82±0.33
S^4-Tuning (Ours)	72.13	**60.30**	60.89	**57.45**	59.87	55.93	**53.92**	**56.92**	**68.44**	55.09	**57.87**	**59.89±0.10**
K=128												
Full Model	72.42	**59.71**	60.34	**57.70**	**60.54**	**56.18**	53.88	57.18	68.40	56.32	58.30	60.09±0.40
S^4-Tuning (Ours)	**72.48**	59.35	**60.54**	57.68	60.47	56.03	**54.13**	**57.98**	**68.79**	**57.24**	**58.62**	**60.30±0.20**

Table 6: **Comparison with Full Model tuning on XQuAD**. S^4-Tuning outperforms Full Model tuning on both $K = 64$ and $K = 128$ settings, with lower standard deviation.