ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414959

META-ADAPTER: EFFICIENT CROSS-LINGUAL ADAPTATION WITH META-LEARNING

Wenxin Hou, Yidong Wang, Shengzhou Gao, Takahiro Shinozaki

Tokyo Institute of Technology, Tokyo, Japan

ABSTRACT

Transfer learning from a multilingual model has shown favorable results on low-resource automatic speech recognition (ASR). However, full-model fine-tuning generates a separate model for every target language and is not suitable for deploying and maintaining in production. The key challenge lies in how to efficiently extend the pre-trained model with fewer parameters. In this paper, we propose to combine the adapter module with meta-learning algorithms to achieve high recognition performance under low-resource settings and improve the parameter-efficiency of the model. Extensive experiments show that our methods can achieve comparable or even superior recognition rates than the state-of-the-art baselines on low-resource languages, especially under very-low-resource conditions, with a significantly smaller model profile.

Index Terms— speech recognition, low-resource, cross-lingual, efficient adaptation, meta-learning

1. INTRODUCTION

End-to-end (E2E) Automatic Speech Recognition (ASR) systems, due to their simplicity in structure and great potential in performance, have witnessed fast development and gained popularity over the recent years [1]. However, the performance of deep learning models highly depends on the availability of training data. Though there are relatively adequate amount of labeled data for popular languages, ASR performances on many low-resource languages are still suffering from a severe data scarcity.

Different languages, despite seemingly different in many ways, intrinsically share a considerable amount of information. Under the context of ASR systems, multilingual systems [2, 3, 4] are developed to capture and utilize this information in common to facilitate ASR systems for low-resource languages. Recently, several large-scale systems have been introduced for multilingual ASR [5]. Pretap et al. [6] introduced a massive single E2E model with up to 1 billion parameters trained on 50 languages. Nearly at the same time, Hou et al. [7] reported a super language-independent Transformer-based ASR model (LID-42) jointly trained on 6 million training utterances from 42 languages with hybrid CTC-attention multi-task learning [8]. Both of them achieved a significant recognition accuracy improvement on low-resource ASR via transfer learning. However, it was also shown that transfer learning could be less effective under very-low-resource conditions due to the overfitting problem [7].

Recently, the *learning to learn* meta-learning concept has been brought into this field to tackle the above challenge. Hsu et al. [9] presented a meta-learning approach for low-resource ASR. Instead of the conventional transfer-learning approach, which pre-trains the initial model on source languages jointly by multi-task learning and then performs fine-tuning on target low-resource languages, they adopted a meta-learning pre-training approach where the model parameters are meta-learned from the source domain. Winata et al. [10] proposed to apply meta-learning for fast cross-accented adaptation and validated the effectiveness in English.

However, existing research mostly focused on full-model adaptation where all the model parameters are re-trained on the target languages, which is computationally expensive and parameter-inefficient, especially for large-scale systems. [11] introduced the adapter module for parameter-efficient domain adaptation in machine translation, where only few parameters are introduced for each target domain. In [5, 12], the authors used language-specific adapters to enhance the performance on each language for a multilingual ASR model.

In this work, we attempt to find an efficient approach for cross-lingual ASR adaptation. We experiment with our meta-adapter module on several low-resources languages. To summarize, our contributions are as follows:

- We investigate several parameter-efficient adaptation methods where few new parameters are introduced and propose Meta-Adapters to combine the adapter module with meta-learning for efficient cross-lingual adaptation.
- Our proposed Meta-Adapters outperform the other parameter-efficient adaptation methods and achieve comparable or even superior recognition rate than classical fine-tuning strategies on low-resource languages with a significantly smaller model profile.

2. METHODOLOGY

2.1. Hybrid CTC-Attention Transformer ASR Model

We employ a sequence-to-sequence Transformer-based [13] ASR model as the base model architecture. The input acoustic features composed of 80-dimensional filter banks and

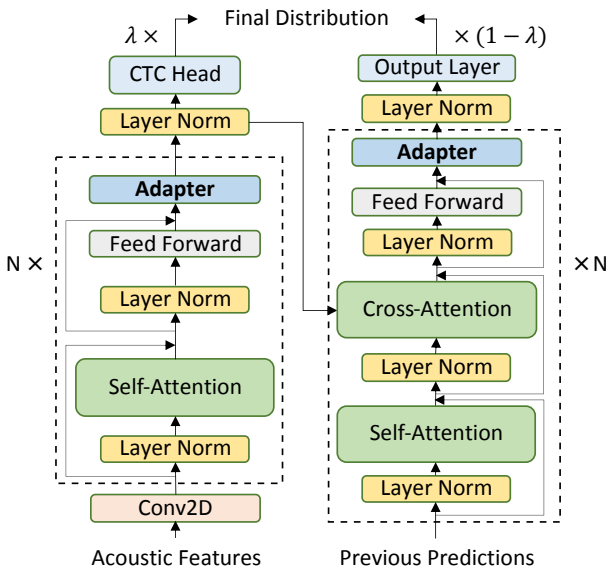

Fig. 1. Overview of the Transformer ASR model based on hybrid CTC-attention architecture with adapters embedded.

3-dimensional pitch features are first fed into several 2D convolutional layers with stride 2 to obtain more expressive representations. Then the Transformer encoder layers process the learned representations to generate encodings by self-attention and feed-forward. In addition to self-attention and feed-forward, the Transformer decoder layers interact with the encoder via cross-attention.

Apart from the attentional Transformer decoder (ATT), we employ a connectionist temporal classification (CTC) [14] head to encourage monotonic alignment in encodings. During training, given speech input X and target labels Y, the multi-objective loss function $\mathcal{L}_{\text{MOL}}$ is given by:

$$\mathcal{L}_{\text{MOL}} = -\alpha \log P_{\text{CTC}}(Y|X) - (1-\alpha) \log P_{\text{ATT}}(Y|X), \quad (1)$$

where α is the weight of CTC loss.

Similarly, during decoding, the final output distribution is also decided by a weighted sum of the CTC head and Transformer decoder predictions:

$$\hat{Y} = \underset{y \in \mathcal{Y}}{\text{argmax}} \{\beta P_{\text{CTC}}(Y|X) + (1-\beta) P_{\text{ATT}}(Y|X)\}, \quad (2)$$

where β is a tunable parameter to balance the two parts.

2.2. Meta-Adapters for Efficient Adaptation

The proposed method utilizes the adapter modules to reduce the adaptation parameters and aims to help the adapter modules find a proper initialization for faster adaptation via meta-learning. Given a pre-trained model, the proposed adaptation approach is composed of two phases: (i) incorporating and meta-training the adapters on a bunch of source tasks; (ii) fine-tuning the pre-trained adapters on unseen target tasks.

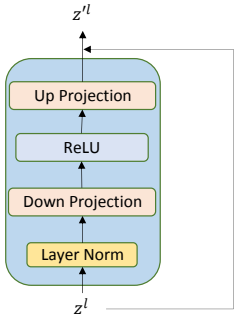

Fig. 2. Architecture of the adapter module.

2.2.1. *Language Adapters*

As shown in Figure 2, an adapter consists of layer normalization, a down-projection layer, a non-linearity function, and an up-projection layer. For the adapter in layer l, the function can be formulated as:

$$\text{Adapter}(\mathbf{z}^l) = \mathbf{z}^l + \mathbf{W}_u^l \text{ReLU}\left(\mathbf{W}_d^l \left(\text{LayerNorm}\left(\mathbf{z}^l\right)\right)\right), \quad (3)$$

where z^l represents the inputs to the adapter in layer l. We incorporate the adapters into the Transformer ASR model as depicted in Figure 1.

2.2.2. *Meta-Learning for Adapters*

The meta-learning algorithms aim to pre-train models that easily adapts to new tasks [15]. In our method, different languages are viewed as different tasks. Given n different source languages $\{S_1, S_2, S_3, \cdots, S_n\}$, the meta-learning methods pre-train the meta-adapter module f_{θ_a} to obtain good initialization parameters θ_a for fast adaptation given any unseen target language. Meanwhile, parameters of the pre-trained backbone θ_b are frozen during both the pre-training and the fine-tuning. We consider two meta-learning algorithms: Model-Agnostic Meta-Learning (MAML) [15] and Reptile [16].

MAML first updates the meta-adapter parameters θ_a by using one or more gradient descent on the language S_i^{tra} sampled from the training dataset S^{tra}. For notation simplicity, the update formula for one gradient descent iteration is:

$$\theta'_{a,i} = \theta_a - \epsilon \nabla \mathcal{L}_{S_i^{tra}}(f_{\theta_a}), \quad (4)$$

where $\mathcal{L}$ is a loss function and ϵ is the fast adaptation learning rate. The adapter parameters are then trained by optimizing the performance of $f_{\theta'_{a,i}}$ with respect to θ_a across the languages sampled from the validation dataset S^{val} with probability $p(S_i^{val})$. The meta-optimization objective is:

$$\theta_a = \theta_a - \gamma \sum_{S_i^{val} \sim p(S^{val})} \nabla_{\theta_a} \mathcal{L}_{S_i^{val}}(f_{\theta'_{a,i}}), \quad (5)$$

where γ is the meta step size,

$$\mathcal{L}_{S_i^{val}}(f_{\theta'_{a,i}}) = \mathcal{L}_{S_i^{val}}(f_{\theta_a - \epsilon \nabla_{\theta_a} \mathcal{L}_{S_i^{tra}}(f_{\theta_a})}). \quad (6)$$

Table 1. Statistics of target language data for adaptation

Lang.	Train Dur.(hrs)	#Train Utt.	#Test Utt.
or	0.45	319	84
hsb	1.48	808	379
br	2.84	3684	1953
ga-IE	2.10	2338	497
ro	3.04	2789	1372

Unlike MAML, Reptile simply combines the gradient of multiple inner training steps and updates the meta-adapter parameters in a more natural way. It does not require a training-validation data split during training. For inner training iteration k, the update formula is given by:

$$\theta_{a,i_{k+1}} = \theta_{a,i_k} - \epsilon \nabla \mathcal{L}_{D_i}(f_{\theta_{a,i_k}}), \tag{7}$$

where ϵ is the fast adaptation learning rate and $\theta_{a,i_0} = \theta_a$. After K steps of inner training, the meta-optimization objective is:

$$\theta_a = \theta_a + \gamma \sum_{S_i \sim p(S)} (\theta_{a,i_K} - \theta_a), \tag{8}$$

where γ is the meta step size.

3. EXPERIMENTAL SETUP

3.1. Data Set

We use data from the Mozilla's Common Voice Corpus 5.1 [17]. We select 10 languages as source tasks: Chuvash, Maltese, Hakha Chin, Kyrgyz, Dhivehi, Slovenian, Greek, Latvian, Frisian, and Sakha; and 5 languages as the target tasks: Odia (or), Sorbian, Upper (hsb), Breton (br), Irish (ga-IE) and Romanian (ro). For all languages, we follow the official "dev" and "test" splits for development and testing, respectively. The rest validated data are used for training. Statistics of the low-resource target language data are shown in Table 1.

3.2. Implementation Details

We conducted all the experiments using ESPnet end-to-end speech processing toolkit [18]. The acoustic features are 80-dimensional filter banks with 3-dimensional pitch features computed every 10 ms over a 25 ms sliding window. For every target language, a subword vocabulary of size 100 is generated using the SentencePiece [19] toolkit.

We employ the aforementioned subword-based LID-42 model presented in [7] as the pre-trained multilingual ASR model, which consists of 12 encoder layers and 6 decoder layers with a model dimension of 256. The number of multi-head attention heads is 4 and the inner-dimension of the feed-forward network is 2048.

The bottleneck dimension of adapters is set to 32. We train the meta-adapters for 100 epochs on 10 source languages. CTC loss weight α is set to 0.3. Following [16], we use Adam optimizer [20] with $\beta_1 = 0$ in the inner training loop and vanilla stochastic gradient descent (SGD) in the outer loop. The fast adaptation learning rate ϵ and initial meta step size γ are 0.001 and 1.0, respectively. The meta step size linearly annealed to 0 over the course of training. The number of inner training steps K of Reptile is 4. For MAML implementation, we ignore the second-order term following previous works [9, 10] and the equation 5 becomes:

$$\theta_a = \theta_a - \gamma \sum_{S_i^{val} \sim p(S^{val})} \nabla_{\theta'_{a,i}} \mathcal{L}_{S_i^{val}}(f_{\theta'_{a,i}}). \tag{9}$$

During adaptation, the meta-adapter is fine-tuned for 1000 iterations with a batch size of 8. We then evaluate the model performance on the test set using beam search with a beam size of 10 and a CTC decoding weight β of 0.5.

For each target language, we consider the following approaches as the baselines: (i) Head-FT: train the language-specific CTC head and output layer without injecting adapters; (ii) Vanilla-Adapter: train randomly-initialized adapter modules; (iii) MOL-Adapter: train the adapters which are pre-trained on 10 source languages by multi-objective learning.

Table 2. Word error rates (WER) on test sets. The first three methods are baselines and the last two are proposed methods combining meta-learning and adapter.

Method	or	hsb	ga-IE	br	ro
Head-FT	95.1	100.5	82.6	91.8	86.4
Vanilla-Adapter	71.3	93.3	73.1	83.2	73.3
MOL-Adapter	77.3	89.7	68.2	82.2	67.5
MAML-Adapter	64.8	**75.6**	68.1	80.7	66.0
Reptile-Adapter	**64.1**	75.7	**67.0**	**79.9**	**64.3**

4. RESULTS

Quantitative Analysis. Table 2 shows the word error rate (WER) results of different methods on target low-resource language adaptation. The adapter-based methods outperform the non-adapter-based Head-FT method by a significant margin, which proves the effectiveness of the adapter module. We can observe that the MOL-Adapter shows better performance than the Vanilla-Adapter in 4 out of 5 languages except for Odia (or), indicating that it might have encountered the over-fitting problem when the target training data is particularly limited. It is also found that the meta-learning methods including MAML-Adapter and Reptile-Adapter are doing better than the non-meta-learning counterparts.

Impact of Trainable Parameters. We compare the proposed method with several classical training/fine-tuning strategies including: (i) training a randomly-initialized full model of identical architecture on target languages from scratch (Full-Base); (ii) fine-tuning the decoder only (Decoder-FT); (iii)

Table 3. Trainable parameter sizes of different approaches.

Method	#Parameters
Full-Base & Full-FT	27,235K
Decoder-FT	9,550K
Adapters	381K

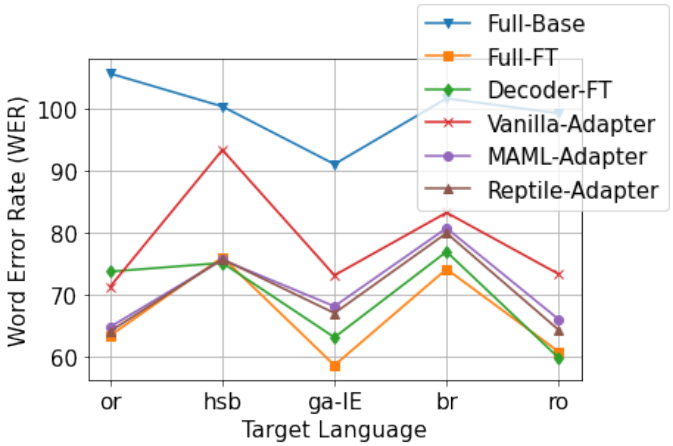

Fig. 3. Word error rate (WER) performance comparison under various model settings. MAML-Adapter and Reptile-Adapter are proposed methods.

fine-tuning the full model (Full-FT). Table 3 shows the trainable parameters during fine-tuning. We can observe that for each target language, Meta-Adapters introduce only 1.4% and 4.0% new parameters compared with full-model and decoder-only strategies, respectively. On the other hand, the adaptation results of WER are presented in Figure 3. All the fine-tuning methods significantly outperform the Full-Base. Although Full-FT and Decoder-FT generally perform better than adapters due to their larger parameter sizes, we notice that Meta-Adapters perform better than the Decoder-FT on Odia (or) and the Full-FT on Sorbian, Upper (hsb), which could be because of their limited amount of training data.

Impact of Adaptation Data Size. Then we look into the relation between the adaptation performance with respect to a step-by-step reduction in the amount of training data used. We randomly sample a portion of training utterances from Romanian (ro) for adaptation, the results are shown in Table 4. It is found that the gap between Meta-Adapters and other adapters is enlarged, while the gap between Meta-Adapters and Full-FT and Decoder-FT becomes smaller as fewer adaptation data is used. On the contrary, the MOL-Adapter degrades quickly probably due to the overfitting problem. Moreover, we notice that MAML is more robust than Reptile when the target data size is extremely small and its performance surpasses all the other approaches on 5% and 10%-shot subsets.

Impact of Pre-training Epochs. Finally, we analyze the effects of varying the number of pre-training epochs for the

Table 4. Word error rates (WER) w.r.t. Romanian (ro) adaptation data size under k%-shot setting.

Method	5%	10%	15%	30%	100%
Decoder-FT	87.9	70.5	**64.7**	**60.7**	**59.8**
Full-FT	77.3	73.2	67.8	65.7	60.8
Vanilla-Adapter	84.2	78.3	76.7	73.9	73.3
MOL-Adapter	86.2	78.4	72.6	69.1	67.5
MAML-Adapter	**75.7**	**69.9**	66.8	65.1	66.0
Reptile-Adapter	79.7	71.0	67.9	65.2	64.3

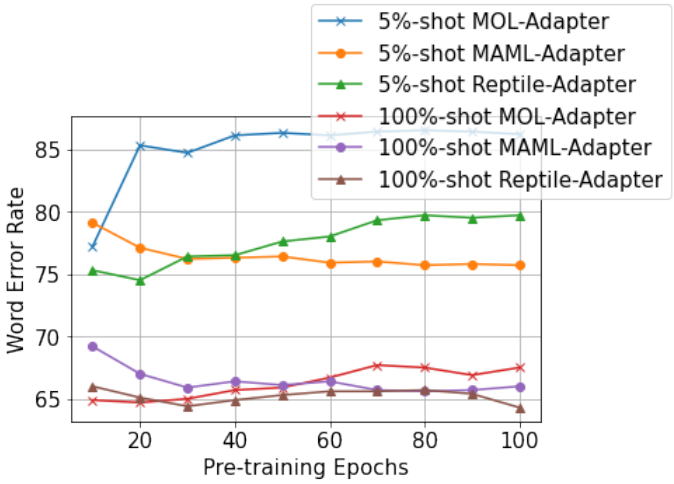

Fig. 4. Word error rate (WER) curve of adapter pre-training epochs on k%-shot Romanian (ro) adaptation.

Reptile-Adapter by fine-tuning adapters at different stages on Romanian (ro) 100%-shot and 5%-shot subset. For comparison, the resultant curve of fine-tuning MOL-Adapters is also presented. As shown in Figure 4, the performance of MOL-Adapter quickly degrades as the number of pre-training epochs increases, which could result from the increasing severity of the overfitting on the source tasks and in turn failing to generalize on the new tasks. This problem gets even more serious when the adaptation data is very small. The performance of the MAML-Adapter appears to be stable on both 5%-shot and 100%-shot curves with no apparent overfitting trend, while some form of overfitting is observed on the 5%-shot curve for the Reptile-Adapter, but nevertheless, it is much better than the MOL-Adapter.

5. CONCLUSIONS

In this work, we propose Meta-Adapter, a fast and parameter-efficient approach for cross-lingual ASR adaptation. Our experiments show the effectiveness of the proposed method on low-resource languages. Our future work may include investigating some adapter-enhancing techniques, e.g., AdapterFusion [21], to further improve the adaptation performance.

6. REFERENCES

[1] Dong Wang, Xiaodong Wang, and Shaohe Lv, "An overview of end-to-end automatic speech recognition," *Symmetry*, vol. 11, no. 8, pp. 1018, 2019.

[2] Ngoc Thang Vu, David Imseng, Daniel Povey, Petr Motlicek, Tanja Schultz, and Hervé Bourlard, "Multilingual deep neural network based acoustic modeling for rapid language adaptation," in *2014 IEEE international Conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 7639–7643.

[3] Sibo Tong, Philip N Garner, and Hervé Bourlard, "An investigation of deep neural networks for multilingual speech recognition training and adaptation," *Proc. Interspeech 2017*, pp. 714–718, 2017.

[4] Oliver Adams, Matthew Wiesner, Shinji Watanabe, and David Yarowsky, "Massively multilingual adversarial speech recognition," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 96–108.

[5] Anjuli Kannan, Arindrima Datta, Tara N. Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, Zhifeng Chen, and Seungji Lee, "Large-Scale Multilingual Speech Recognition with a Streaming End-to-End Model," in *Proc. Interspeech 2019*, 2019, pp. 2130–2134.

[6] Vineel Pratap, Anuroop Sriram, Paden Tomasello, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert, "Massively Multilingual ASR: 50 Languages, 1 Model, 1 Billion Parameters," in *Proc. Interspeech 2020*, 2020, pp. 4751–4755.

[7] Wenxin Hou, Yue Dong, Bairong Zhuang, Longfei Yang, Jiatong Shi, and Takahiro Shinozaki, "Large-Scale End-to-End Multilingual Speech Recognition and Language Identification with Multi-Task Learning," in *Proc. Interspeech 2020*, 2020, pp. 1037–1041.

[8] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[9] Jui-Yang Hsu, Yuan-Jui Chen, and Hung-yi Lee, "Meta learning for end-to-end low-resource speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7844–7848.

[10] Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, Peng Xu, and Pascale Fung, "Learning Fast Adaptation on Cross-Accented Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 1276–1280.

[11] Ankur Bapna and Orhan Firat, "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)*, 2019, pp. 1538–1548.

[12] Genta Indra Winata, Guangsen Wang, Caiming Xiong, and Steven Hoi, "Adapt-and-adjust: Overcoming the long-tail problem of multilingual speech recognition," *arXiv preprint arXiv:2012.01687*, 2020.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[14] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, 2017, pp. 1126–1135.

[16] Alex Nichol and John Schulman, "Reptile: a scalable metalearning algorithm," *arXiv preprint arXiv:1803.02999*, vol. 2, no. 3, pp. 4, 2018.

[17] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020, pp. 4218–4222.

[18] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson-Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., "Espnet: End-to-end speech processing toolkit," *Proc. Interspeech 2018*, pp. 2207–2211, 2018.

[19] Taku Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 66–75.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[21] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych, "Adapterfusion: Non-destructive task composition for transfer learning," *arXiv preprint arXiv:2005.00247*, 2020.