INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Layer-wise Fast Adaptation for End-to-End Multi-Accent Speech Recognition

Xun Gong, Yizhou Lu, Zhikai Zhou, Yanmin Qian†

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{gongxun,luyizhou4,zhikai.zhou,yanminqian}@sjtu.edu.cn

Abstract

Accent variability has posed a huge challenge to automatic speech recognition (ASR) modeling. Although one-hot accent vector based adaptation systems are commonly used, they require prior knowledge about the target accent and cannot handle unseen accents. Furthermore, simply concatenating accent embeddings does not make good use of accent knowledge, which has limited improvements. In this work, we aim to tackle these problems with a novel layer-wise adaptation structure injected into the E2E ASR model encoder. The adapter layer encodes an arbitrary accent in the accent space and assists the ASR model in recognizing accented speech. Given an utterance, the adaptation structure extracts the corresponding accent information and transforms the input acoustic feature into an accent-related feature through the linear combination of all accent bases. We further explore the injection position of the adaptation layer, the number of accent bases, and different types of accent bases to achieve better accent adaptation. Experimental results show that the proposed adaptation structure brings 12% and 10% relative word error rate (WER) reduction on the AESRC2020 accent dataset and the Librispeech dataset, respectively, compared to the baseline.

Index Terms: automatic speech recognition, multi-accent, layer-wise adaptation, end-to-end

1. Introduction

In recent years, end-to-end (E2E) automatic speech recognition (ASR) models, which directly optimize the probability of the output sequence given input acoustic features, have made great progress in a wide range of speech corpora [1]. One of the most pressing needs for ASR today is the support for multiple accents in a single system, which is often referred to as multi-accent speech recognition in the literature. The difficulties of recognizing accented speech, including phonology, vocabulary and grammar, have posed a serious challenge to current ASR systems [2]. A straightforward method is to build a single ASR model from mixed data (accented speech from non-native speakers and standard data from native speakers). However, such models usually suffer from severe performance degradation due to the accent mismatch during training and inference [3, 4, 5, 6, 7]. Previous work has explored different accent adaptation methods for acoustic models. MixNet [6, 8] is based on Mixture of Experts (MoE) architecture, where experts are specialized to segregate accent-specific speech variabilities. Model-agnostic meta-learning (MAML) [9] approach is also explored to learn the rapid adaptation to unseen accents. One-hot accent vectors are well utilized to build multi-basis adaptation [7, 10, 11], where each basis is designed to cover certain types of accents. Recently, with the improvements in accent identification (AID) models [12], several methods have been explored to integrate AID into speech recognition for accent adaptation. For example, [6, 13] proposed to concatenate accent embeddings and acoustic features for adaptation of the acoustic model. [6, 8] proposed a multi-task framework to jointly model both ASR and AID tasks. Meanwhile, these approaches have also been applied to the E2E ASR framework [5, 14].

In this paper, we study a novel approach to rapid adaptation of accented data via the layer-wise transformation of input features. Compared to previous works, the proposed method stimulates the potential of both accent embeddings and hidden representations. Instead of simply concatenating accent embeddings and input features, we adopt a different scheme with scaling and shifting transformations, which has been proven a valuable method to utilize the accent embeddings [7, 15, 11]. Furthermore, we propose the multi-basis adapter layer architecture to represent the accent-dependent features. The adapter basis based approach has shown its potential in various fields, including computer vision [16], natural language processing [17], neural machine translation [18] and multi-lingual ASR [19]. Similarly, multiple bases are also proved to be effective in speaker adaptation [20, 21] and code-switching ASR task [22]. However, the effectiveness of such approaches in multi-accent speech recognition has not been investigated to the best of our knowledge. In this paper, we incorporate the adapter basis based technique into the E2E ASR architecture for multi-accent speech recognition. Furthermore, we downsize the typically massive bases to much smaller modules in each adapter layer. As the proposed method models different accents in the continuous embedding space, it can naturally cope with unseen accents in the inference stage by a linear combination of adapter bases. During adaptation, interpolation coefficients between different adapter bases are predicted from the accent embeddings. With the proposed framework, accent adaptation can be achieved in a parameter-efficient and flexible manner.

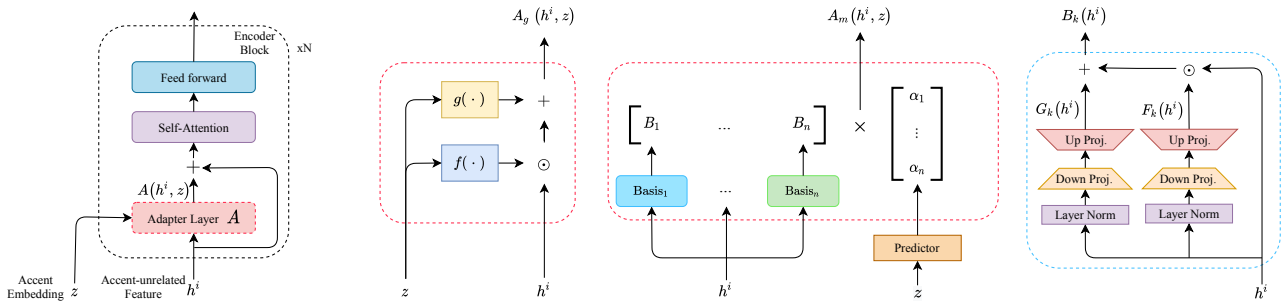
The rest of the paper is organized as below: In Section 2, we present our layer-wise adapter architecture with the multi-task regularization. Experimental results are presented and analyzed in Section 3. Finally, the conclusion is given in Section 4.

2. Layer-wise Fast Adaptation on E2E Multi-Accent ASR

In this section, we first give a brief review of the joint connectionist temporal classification (CTC)-attention based E2E ASR. Then we describe the proposed accent adapter layer and corresponding training strategies. The new approach mainly includes two parts: the adapter layer construction and interpolation coefficients regularization.

†corresponding author

http://dx.doi.org/10.21437/Interspeech.2021-1075



(a) Layer-wise Adaption *(b) Gated Adapter Layer A_g* *(c) Multi-Basis Adapter Layer A_m* *(d) Basis B_k*

Figure 1: *Schematic diagram of the proposed adapter layer. The adapter layer in (a) is optionally inserted in each encoder block, which is discussed in Section 3.3.1. Here, $+$, $\times$, and $\odot$ denote summation, matrix multiplication, and element-wise product, respectively.*

2.1. Pretrained transformer-based E2E ASR

The transformer is a sequence-to-sequence (S2S) structure consisting of a multi-layer encoder and a multi-layer decoder [23]. The encoder takes acoustic features as input to be mapped into high-level representations $\mathbf{h}$. The decoder network utilizes the encoded representation $\mathbf{h}$ with an attention mechanism and outputs the predicted tokens auto-regressively. At each decoding step, the decoder emits the posterior probabilities of the next token given previous outputs. We train the transformer model with the joint CTC-attention framework [24] to exploit the advantages from both CTC and attention-based models. The loss function is defined as below:

$$\mathcal{L}_{jca} = \lambda_{ctc}\mathcal{L}_{ctc} + (1-\lambda_{ctc})\mathcal{L}_{s2s} \tag{1}$$

where $\mathcal{L}_{ctc}$ and $\mathcal{L}_{s2s}$ are the CTC and S2S objective losses, respectively. A tunable parameter $\lambda_{ctc} \in [0, 1]$ is used to control the contribution of each loss.

2.2. Adapter Layer

The E2E ASR model trained on common standard corpora usually lacks generalization on accented data due to the accent mismatch. Adapter layers are injected into the ASR encoder blocks to transform the accent-unrelated features into the accent-related space. The architecture of the new ASR encoder with the proposed adapter layer is illustrated in Figure 1(a). The adapter layer, hereinafter denoted as $\mathcal{A}$, is used as a preprocessing to transform accent-unrelated features into accent-related features. Denote by $\boldsymbol{h^i}$ the input feature before the encoder block, $\boldsymbol{z}$ the accent embedding, and $\mathcal{A}(\boldsymbol{h^i}, \boldsymbol{z})$ the output feature in the accent-related space. The output feature $\mathcal{A}(\boldsymbol{h^i}, \boldsymbol{z})$ is then wrapped into the encoder block by a residual connection $(+)$ as shown in Figure 1(a), to enable the original acoustic information to flow through later encoder layers. Different types of adapter layers $\mathcal{A}$ are explored in the following sections: $\mathcal{A}_g$ in Section 2.2.1 and $\mathcal{A}_m$ in Section 2.2.2.

2.2.1. Gated Adapter Layer

The first scheme to attain the transform function follows our previous investigation in [15]. As shown in Figure 1 (b), a scaling factor $f(\boldsymbol{z})$ and a shifting factor $g(\boldsymbol{z})$ can be applied to the input feature for accent adaptation:

$$\mathcal{A}_g(\boldsymbol{h^i}, \boldsymbol{z}) = f(\boldsymbol{z}) \odot \boldsymbol{h^i} + g(\boldsymbol{z}) \tag{2}$$

where $\mathcal{A}_g$ is the gated adapter layer, and $\odot$ denotes the element-wise product. $f(\boldsymbol{z})$ and $g(\boldsymbol{z})$ are separately generated by a single dense layer with $\tanh(\cdot)$ activation.

2.2.2. Multi-basis Adapter Layer

The second scheme is to build a multi-basis adapter layer as in Figure 1 (c). The multi-basis layer concatenates the output $B_k(\boldsymbol{h^i})$ from each basis with the corresponding interpolation coefficient α_k. Similar to Section 2.2.1, the scaling $F_k(\cdot)$ and shifting $G_k(\cdot)$ modules are used to transform the input $\boldsymbol{h^i}$ into the accent-related space as shown in Figure 1 (d), where $k = 1, 2, \ldots, n$ and n is the number of adapter bases.

$$\begin{aligned}\mathcal{A}_m(\boldsymbol{h^i}, \boldsymbol{z}) &= \sum_{k=1}^{n} \alpha_k B_k(\boldsymbol{h^i}) \\ &= \sum_{k=1}^{n} \alpha_k * \{F_k(\boldsymbol{h^i}) \odot \boldsymbol{h^i} + G_k(\boldsymbol{h^i})\}\end{aligned} \tag{3}$$

Note that one can also use scaling-only $\big(G_k(\boldsymbol{h^i}) = \mathbf{0}\big)$ and shifting-only $\big(F_k(\boldsymbol{h^i}) = \mathbf{0}\big)$ operations in the bases, which will be discussed in Section 3.3.3.

Projection Module To make the bases in Figure 1 (d) simple and flexible, we propose a sandglass-style structure for $F(\cdot)$ and $G(\cdot)$ modeling: a down-projection network and a up-projection network with the non-linear activation $\mathrm{ReLU}(\cdot)$. This architecture allows us to easily adjust the modules' capacity, depending on the complexity of the accents. Additionally, we normalize the input of each adapter basis by LayerNorm [25].

Predictor Different from the one-hot accent vector that is commonly used in prior works on accent adaptation [7, 10], here we adopt a soft assignment of bases by interpolating between all adapter bases dynamically. To estimate interpolation coefficients $\boldsymbol{\alpha} \in \mathbb{R}^n$ from accent embedding $\boldsymbol{z}$, a predictor $p(\cdot)$ model is used, and give guidance on the usage of modules.

$$\boldsymbol{\alpha} = \mathrm{SoftMax}(p(\boldsymbol{z})), \text{ where } 1 = \sum_{k=1}^{n} \alpha_k \tag{4}$$

where the interpolation coefficients $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ are probabilities for multiple bases. The predictor $p(\cdot)$ can be composed of several DNN layers.

2.2.3. Multi-task Regularization

During training, we found that, without any constraints, the distribution of interpolation coefficients $\boldsymbol{\alpha}$ would rapidly reduce to a certain basis for all accents, which greatly limits the adapter layer's adaptation capability. Thus, we apply the multi-task

learning (MTL) scheme to utilize the loss from an auxiliary task, i.e. the predictor in Section 2.2.2, to regularize the training of both ASR and predictor models. An auxiliary loss from the predictor is introduced to the ASR loss $\mathcal{L}_{jca}$, and then the final loss $\mathcal{L}_{mtl}$ for the entire system is calculated as:

$$\mathcal{L}_{mtl} = \mathcal{L}_{jca} + \gamma_{mtl}\mathcal{L}_{\text{MSE}}(\boldsymbol{\alpha}^{(\text{ref})}, \boldsymbol{\alpha}) \quad (5)$$

where $\boldsymbol{\alpha}^{(\text{ref})}$ is the target label of the predictor outputs $p(\boldsymbol{z})$, $\boldsymbol{\alpha}$ is the output of the predictor, and γ_{mtl} is a hyperparameter to control the contribution of the predictor loss. The target label $\boldsymbol{\alpha}^{(\text{ref})}$ is obtained via the clustering of accent embeddings extracted from the pretrained AID model. The number of clusters is set to n, and here the K-means algorithm is adopted.

3. Experiments

3.1. Setup

3.1.1. Dataset

Our experiments are conducted on the Accented English Speech Recognition Challenge 2020 (AESRC2020) dataset [2] and the Librispeech corpus [26]. AESRC2020 contains a training set for 8 English accents in England (UK), America (US), China (CHN), Japan (JPN), Russia (RU), India (IND), Portugal (PT) and Korea (KR), with 20-hour data for each accent, however two more accents Canada (CAN) and Spain (ES) are included in test set while cv set has eight accents. Librispeech contains a 960-hour training set, while dev-clean/other (dev c/o) and test-clean/other (test c/o) are used for standard tests. We report the word error rate (WER) on all evaluation sets.

3.1.2. E2E based Baseline

For acoustic feature extraction, 80-dimensional fbank features are extracted with a step size of 10ms and a window size of 25ms, and utterance-level cepstral mean and variance normalization (CMVN) is applied on the fbank features. For language modeling, the 500 English Byte Pair encoding (BPE) [27] subword units are adopted. For E2E ASR, we adopt the transformer with the configuration of a 12-layer encoder and a 6-layer decoder [23, 28], where each self-attention layer has an attention dimension of 512 and 8 heads. SpecAugment [29] is also applied for data augmentation during training. During decoding, the CTC module is used for score interpolation [24] with a weight of 0.3, and a beam width of 10 is applied for beam searching. All models are built using the ESPnet toolkit [30].

3.2. Accent Identification and Embedding Extraction

A pretrained time-delay neural network (TDNN) [31] based accent identification (AID) model is used for extracting 256-dimension accent embeddings. It accepts phone posteriorgram (PPG) features as input and is trained to predict accent categories. The accent embeddings are obtained from the penultimate layer output of the AID model. More details about the AID model can be found in our accent identification system description [12] for the AESRC 2020 challenge [2].

3.3. Exploration of Multi-Basis Adapter Layer

We first investigate the performance of the proposed multi-basis adapter layer architecture in Section 2.2.2 with different injection positions, numbers of bases and types of bases.

3.3.1. Position of Adapter Layer

The performance of the baseline model in Section 2.1 and our proposed models with 4-basis adapter layers are compared in Table 1. Different positions of the adapter layers are evaluated, including $\{1\}, \{6\}, \{12\}, \{1\text{-}6\}$, and $\{1\text{-}12\}$, where $\{m\text{-}n\}$ means injecting adapter layers into the m^{th}~n^{th} encoder blocks.

For models with different positions of a single adapter layer injected at only one encoder block (lines 2~4), the performance becomes slightly worse as the injection position moves towards the last encoder block. However, as the number of adapter layers increases, the WER is only on par with single adapter layer-based models. This indicates that a single adapter layer injected in the first encoder block is already capable of adapting to various accents, while still keeping the parameter efficiency. Therefore, in the following experiments, only one multi-basis adapter layer is injected in the first encoder block.

Table 1: *Performance (WER) (%) comparison of the multi-basis adapter layer positions and numbers.*

Position	Accent		Libri	
	cv	test	dev c/o	test c/o
-	6.54	7.61	5.64/11.43	6.31/11.68
{1}	**5.91**	**6.82**	**5.17/10.37**	**5.48/10.65**
{6}	5.91	6.89	5.23/10.41	5.51/10.73
{12}	6.08	7.08	5.20/10.67	5.74/10.97
{1-6}	5.85	6.82	**5.21/10.39**	5.65/10.79
{1-12}	**5.82**	**6.78**	5.23/10.27	**5.61/10.69**

3.3.2. The Number of Bases

We then explore the impact of different bases numbers (ranging from 2 to 8) on the ASR performance. As shown in Table 2, the WER is gradually decreased as the number of bases increases from 2 to 8. However, the performance gain is very limited when more than 4 bases are used, but more basis will result in more parameters. Considering the tradeoff between performance and model size, we adopt the 4-basis adapter layer in our following experiments.

Table 2: *Performance (WER) (%) comparison on different numbers of bases in one adapter layer.*

# Bases	Accent		Libri	
	cv	test	dev c/o	test c/o
-	6.54	7.61	5.64/11.43	6.31/11.68
2	6.23	7.34	5.36/11.06	6.01/11.32
4	5.91	6.82	5.17/**10.37**	**5.48/10.65**
6	**5.89**	**6.81**	**5.14**/10.41	5.50/10.66
8	5.78	7.01	5.20/10.43	5.52/10.71

3.3.3. Different Types of Bases

Table 3 shows the performance of different bases types, including different connection modes (scale, shift, or both scale and shift) in Section 2.2.2 and different projection module types in the bases. DNN-based basis uses 'Linear' whose encoded dimension is set to 128, while CNN-based basis uses 'Conv2d' with a 5×5 kernel via 16 channels. The best performance is achieved when both scaling and shifting are used. This indicates that the shifting and scaling modes can benefit each other complementarily. We further test different network types (DNN or CNN) in the bases implementation. It is observed that CNN-based modules has insufficient ability to extract accent-related

Table 4: *Performance (WER) (%) comparison of baseline system and different adaptation methods. $\mathcal{A}_g$ denotes the proposed single-basis accent embedding layer adaptation model in Section 2.2.1, and $\mathcal{A}_m$ denotes the proposed multi-basis adaptation model introduced in Section 2.2.2, with injection only in the first encoder block.*

Model	Accent Test Set										Accent		Libri	
	US	UK	IND	CHN	JPN	PT	RU	KR	CAN	ES	cv	test	dev c/o	test c/o
Baseline	5.75	3.17	9.32	13.49	6.66	6.38	11.04	6.80	5.21	9.88	6.54	7.61	5.64/11.43	6.31/11.68
Finetune	4.92	2.82	8.34	12.00	5.92	5.66	9.78	5.82	4.25	9.18	5.85	6.83	7.84/13.03	8.67/13.94
$\mathcal{A}_g$	5.33	3.11	8.80	12.29	6.15	6.09	9.99	6.30	4.70	9.06	5.89	6.89	5.27/10.39	5.76/10.79
$\mathcal{A}_m$	5.29	**2.54**	8.91	11.87	6.04	5.79	9.71	6.00	4.51	8.91	5.91	6.82	5.17/10.37	5.48/10.65
$\mathcal{A}_g$ + $\mathcal{A}_m$	**4.88**	2.57	**8.38**	**11.54**	**5.73**	**5.60**	**9.71**	**5.70**	**4.21**	**8.51**	**5.77**	**6.68**	**4.73/10.22**	**5.32/10.61**

Table 3: *Performance (WER) (%) comparison of different projection module types and connections in the basis.*

Network Type	Connection Mode	Accent		Libri	
		cv	test	dev c/o	test c/o
-	-	6.54	7.61	5.64/11.43	6.31/11.68
DNN	shifting-only	5.96	6.91	5.22/10.41	5.51/10.78
DNN	scaling-only	5.95	6.99	5.23/10.44	**5.46**/10.70
DNN	both	**5.91**	**6.82**	**5.17/10.37**	5.48/**10.65**
CNN	shifting-only	6.12	7.11	5.16/10.61	5.67/11.10

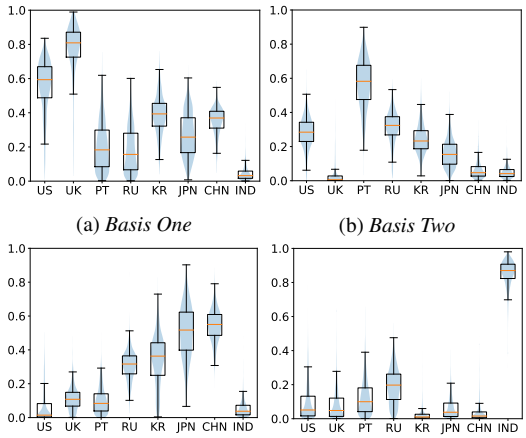
information. In our final system, the DNN-based bases are used for consistency.

3.4. Results Comparison of Different Adaptation Methods

In this section, we present the detailed performance comparison of all proposed models and the baselines in Table 4. Fine-tuning the baseline model for accented data is an intuitive way to perform adaptation on accented data, which is shown in the second line of Table 4. However, this is not feasible for some unseen accents like Spain (ES), which is unavoidable during inference. On the other aspect, it degrades the performance on standard data, i.e. Librispeech evaluation sets. The gated adapter layer in Section 2.2.1 is denoted as $\mathcal{A}_g$ in the table, which shows significant improvement on both Librispeech and accent datasets. Denote by $\mathcal{A}_m$ the proposed multi-basis adapter layer introduced in Section 2.2.2, adapter layer $\mathcal{A}_m$ is injected only in the first encoder block, which consists of 4 bases that are structured by DNN-based projection modules. Furthermore, we combine $\mathcal{A}_g$ and $\mathcal{A}_m$ by computing the output as $h^i + \mathcal{A}_m(h^i + \mathcal{A}_g(h^i, z), z)$. We observe that the final proposed method $\mathcal{A}_g$ + $\mathcal{A}_m$ consistently outperforms the baseline model which shows that the proposed method can learn accent-related information effectively and improve the robustness of speech recognition against accent variation.

3.5. Visualization of Multi-Basis Adapter Layer

Figure 2 shows the coefficient distributions on each basis from the 4-basis adapter layer model. Accents with large coefficients in each basis are assumed to be more correlated to that basis. It can be clearly seen that different bases capture a different set of highly correlated accents. For example, *Basis Two* focuses mostly on extracting information about the Portuguese (PT) accent, and then the American (US) and Russian (RU) accents. The inherent correlation between different accents can be also revealed from this figure. For example, American (US) and British (UK) accents have consistently high correlations with *Basis One*, and much lower correlations with other bases. Meanwhile, Indian (IND) and Japanese (JPN) accents have distinct preferences for bases: IND accent prefers *Basis Four* while JPN accent prefers *Basis Three*. Results demonstrate that



(a) *Basis One* (b) *Basis Two*

(c) *Basis Three* (d) *Basis Four*

Figure 2: *Boxplot and violinplot visualization of interpolation coefficent distributions for each basis. The vertical axis shows the interpolation coefficent α_i, where i is the basis index. The horizontal axis is the accent categories.*

our proposed multi-basis adapter layer approaches can well-capture the accent-dependent information with the guidance of accent embeddings, thus improving the multi-accent ASR performance.

4. Conclusions

In this paper, we explore a layer-wise adapter architecture to improve E2E-based multi-accent speech recognition models. The proposed models transform accent-unrelated input into an accent-related space by injecting small adapter layers in the ASR encoder blocks. The models use a pretrained accent identification network for accent embeddings estimation, a shared predictor for learning interpolation coefficients of different adapter bases, and several accent-related bases for accent adaptation. Experimental results reveal that we outperform the baseline model up to 12% relative WER reduction on AESRC2020 cv/test sets and 10% relative WER reduction on Librispeech dev/test sets as well. In future work, we would like to investigate different combination methods between the accent embedding and acoustic features, i.e. the internal structures of the adapter basis.

5. Acknowledgements

This work was supported by the China NSFC projects (No. 62071288 and No. U1736202). Experiments have been carried out on the PI supercomputers at Shang-hai Jiao Tong University.

6. References

[1] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs RNN in speech applications," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 449–456.

[2] X. Shi, F. Yu, Y. Lu, Y. Liang, Q. Feng, D. Wang, Y. Qian, and L. Xie, "The accented English speech recognition challenge 2020: open datasets, tracks, baselines, results and methods," *arXiv preprint arXiv:2102.10233*, 2021.

[3] M. Chen, Z. Yang, J. Liang, Y. Li, and W. Liu, "Improving deep neural networks based multi-accent mandarin speech recognition using i-vectors and accent-specific top layer," in *Proc. Interspeech*, 2015.

[4] M. Elfeky, M. Bastani, X. Velez, P. Moreno, and A. Waters, "Towards acoustic model unification across dialects," in *IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 624–628.

[5] B. Li, T. N. Sainath, K. C. Sim, M. Bacchiani, E. Weinstein, P. Nguyen, Z. Chen, Y. Wu, and K. Rao, "Multi-dialect speech recognition with a single sequence-to-sequence model," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4749–4753.

[6] A. Jain, M. Upreti, and P. Jyothi, "Improved accented speech recognition using accent embeddings and multi-task learning," in *Proc. Interspeech*, 2018, pp. 2454–2458.

[7] M. Grace, M. Bastani, and E. Weinstein, "Occam's adaptation: A comparison of interpolation of bases adaptation methods for multi-dialect acoustic modeling with LSTMs," in *IEEE Spoken Language Technology Workshop (SLT)*, 2018, p. 174–181.

[8] A. Jain, V. P. Singh, and S. P. Rath, "A multi-accent acoustic model using mixture of experts for speech recognition," in *Proc. Interspeech*, 2019, pp. 779–783.

[9] G. I. Winata, S. Cahyawijaya, Z. Liu, Z. Lin, A. Madotto, P. Xu, and P. Fung, "Learning fast adaptation on cross-accented speech recognition," in *Proc. Interspeech 2020*, 2020, pp. 1276–1280.

[10] X. Yang, K. Audhkhasi, A. Rosenberg, S. Thomas, B. Ramabhadran, and M. Hasegawa-Johnson, "Joint modeling of accents and acoustics for multi-accent speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1–5.

[11] S. Yoo, I. Song, and Y. Bengio, "A highly adaptive acoustic model for accurate multi-dialect speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5716–5720.

[12] H. Huang, X. Xiang, Y. Yang, R. Ma, and Y. Qian, "AISPEECH-SJTU accent identification system for the accented English speech recognition challenge," *arXiv preprint arXiv:2102.09828*, 2021.

[13] M. T. Turan, E. Vincent, and D. Jouvet, "Achieving multi-accent asr via unsupervised acoustic model adaptation," in *Proc. Interspeech*, 2020.

[14] T. Viglino, P. Motlicek, and M. Cernak, "End-to-end accented speech recognition," in *Proc. Interspeech*, 2019, pp. 2140–2144.

[15] T. Tan, Y. Lu, R. Ma, S. Zhu, J. Guo, and Y. Qian, "AISPEECH-SJTU ASR system for the accented English speech recognition challenge," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[16] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[17] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proc. ICML*. PMLR, 2019, pp. 2790–2799.

[18] A. Bapna and O. 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.

[19] A. Kannan, A. Datta, T. N. Sainath, E. Weinstein, B. Ramabhadran, Y. Wu, A. Bapna, Z. Chen, and S. Lee, "Large-scale multilingual speech recognition with a streaming end-to-end model," in *Proc. Interspeech*, 2019, pp. 2130–2134.

[20] T. Tan, Y. Qian, M. Yin, Y. Zhuang, and K. Yu, "Cluster adaptive training for deep neural network," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4325–4329.

[21] C. Wu, P. Karanasou, and M. J. Gales, "Combining i-vector representation and structured neural networks for rapid adaptation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5000–5004.

[22] Y. Lu, M. Huang, H. Li, J. Guo, and Y. Qian, "Bi-encoder transformer network for Mandarin-English code-switching speech recognition using mixture of experts," in *Proc. Interspeech*, 2020, pp. 4766–4770.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[24] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2017, pp. 4835–4839.

[25] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," in *NeurIPS Deep Learning Symposium*, 2016.

[26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[27] T. 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.

[28] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[29] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[30] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[31] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015, pp. 3214–3218.