INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Domain Aware Training for Far-field Small-footprint Keyword Spotting

Haiwei Wu[1], *Yan Jia*[1], *Yuanfei Nie*[2], *Ming Li*[1]

[1]Data Science Research Center, Duke Kunshan University, Kunshan, China
[2]Montage Technology, Kunshan, China
ming.li369@dukekunshan.edu.cn

Abstract

In this paper, we focus on the task of small-footprint keyword spotting under the far-field scenario. Far-field environments are commonly encountered in real-life speech applications, causing severe degradation of performance due to room reverberation and various kinds of noises. Our baseline system is built on the convolutional neural network trained with pooled data of both far-field and close-talking speech. To cope with the distortions, we develop three domain aware training systems, including the domain embedding system, the deep CORAL system, and the multi-task learning system. These methods incorporate domain knowledge into network training and improve the performance of the keyword classifier on far-field conditions. Experimental results show that our proposed methods manage to maintain the performance on the close-talking speech and achieve significant improvement on the far-field test set.
Index Terms: small footprint keyword spotting, far-field condition, domain aware training, multi-task learning

1. Introduction

Small footprint keyword spotting (KWS), also known as wake-up word detection, is a task to detect the occurrences of a pre-defined keyword in continuous speech signals. With the rapid development of mobile devices, smart speakers, and other applications, which require a hands-free conversational interface, this technology is attracting more and more attention. Different from the traditional keyword spotting task, with the constraints of hardware, real-life wake-up word detection must have a small memory and low computational cost. And simultaneously, it also requires to be highly accurate in detection and robust in different complex environments like noisy and far-field conditions.

Traditional approaches [1, 2] on this task involve Hidden Markov Models (HMMs), which are utilized to construct the keyword model and the filler/background model. The background model is trained with non-keyword speech as well as background noise and silence. The acoustic modeling schemes for speech units include Gaussian Mixture Model (GMM), Deep Neural Network (DNN), and Time-Delayed Neural Network (TDNN) [2], and so on. After training, the Vertibi search is applied to find the optimal path in the decoding graph. Whenever the likelihood ratio of the keyword vs. filler model is larger than the pre-defined threshold, the system triggers.

In recent years, many researchers focus on the DNN based keyword spotting systems, which achieve better performances than traditional methods [3–11]. In these approaches, a DNN model is trained for words instead of phonemes. The output smoothed posterior probabilities are calculated later from the DNN model's output to compute the confidence score. DNN based methods have the advantages of light-weighting and low latency, which is suitable for real-life applications. As for modeling, many structures based on Convolutional Neural Network (CNN) [3], Recurrent Neural Network (RNN), Convolutional Recurrent Neural Network (CRNN) [4], Long Short Time Memory [5] (LSTM) and attention mechanism [8, 9] are explored. Furthermore, [10] adopts the residual network structure to classify the speech command words, and [11] introduces a dilated convolutional structure to model the whole keyword sequence, which also shows good performance.

However, in many real-life applications, like smart speakers, the performance of the KWS system is often degraded under the low Signal-to-Noise Ratio (SNR) and far-field conditions. The room reverberation and different kinds of noises in this scenario impose great challenges on the performance of the DNN model, which is trained mainly by close-talking data due to the zero or limit resource for real data collection. A traditional method to tackle this problem is to train DNN models using pooled speech data either collected or simulated from different environments.

In this paper, we employ three domain aware training mechanisms to improve network performance under far-field conditions. The first method is motivated by the noise-robust training with environmental noise embeddings [12, 13] in the speech recognition area. We pre-train a domain classifier to extract environmental domain embeddings, which are fused to the training of the keyword classifier. And the second method is inspired by the within-sample variability-invariant loss [14] and paralleled data training [15–17] mechanisms successfully applied in speaker verification and automatic speech recognition on complex environments. We propose a training scheme of multi-task learning [18] with the CORAL loss on KWS, which reduces the mismatch of close-talking and far-field conditions in a multi-domain joint learning setup. The third method is based on the multi-task learning [19, 20], which optimizes models to make predictions on both domain types and keywords simultaneously.

This rest of the paper is organized as follows. Section 2 describes the framework of the CNN based KWS system, and in section 3, our proposed domain aware training approaches are introduced. Section 4 discusses the experimental results, and section 5 concludes our work, respectively.

2. CNN based KWS system

Our baseline is constructed on a CNN based KWS system proposed by [3]. The pipeline has three main components, feature extraction, network prediction, and confidence computation. In the step of feature extraction, we extract 40-dimensional log-Mel filterbank energy (Fbank) with a 25ms window and a 10 ms shift. And we apply a window of 40 frames to generate training samples as the input of the model considering the context.

Our convolutional network structure contains three convolutional layers, each of which is followed by a max-pooling layer. The convolutional kernels have the size of $(3, 3)$ with

http://dx.doi.org/10.21437/Interspeech.2020-1412

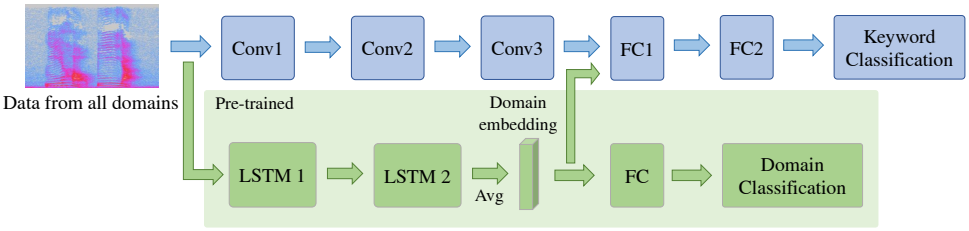

Figure 1: *Framework of the domain embedding system.*

the stride of $(1, 1)$, and the pooling size is set to be $(2, 2)$. Two fully-connected layers accompanied by a final softmax activation layer are then used to predict the target keywords. Rectified linear unit (ReLU) is used as the activation function in hidden layers.

After the training process, the sequence of acoustic features is projected to a posterior probabilities sequence of keywords by the model. In the module of confidence computation, we adopt the method proposed in [6, 21] to make the decisions. In this approach, we define a sliding window with the length of T_s frames which is used to compute scores and denote the input acoustic features in a window as $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_{Ts}\}$. $\boldsymbol{w} = \{w_1, w_2 \cdots w_M\}$ represents the words sequence of the pre-defined wake up word. We smooth the output probabilities at a length of L frames by taking average as

$$s_{w_i}(\boldsymbol{x}_t) = \frac{1}{L} \sum_{j=t-L-1}^{t} p_{w_i}(\boldsymbol{x}_j), \tag{1}$$

where $s_{w_i}(\boldsymbol{x}_t)$ represents the smoothed probablities at time t of word w_i and $p_{w_i}(\boldsymbol{x}_j)$ refers to the network output of j^{th} frame at word w_i. After smoothing, we compute the confidence score as follows:

$$h(\boldsymbol{x}) = \left[\max_{1 \le t_1 < \cdots < t_M \le T_s} \prod_{i=1}^{M} s_{w_i}(\boldsymbol{x}_{ti}) \right]^{\frac{1}{M}}, \tag{2}$$

where $h(\boldsymbol{x})$ refers to the output confidence score. Compared to previous methods [1], it has the advantage of considering the order of words that trigger, and at the same time, the time complexity is $O(MT_s)$, which is suitable for the real-time application. The system triggers whenever the confidence score exceed the pre-defined threshold.

3. Domain Aware Training

The influence of far-field and noisy conditions in speech signal processing is commonly noticed in many areas like speech recognition and speaker verification. In our works, we apply three domain aware training algorithms on the far-field small-footprint keyword spotting to enrich the knowledge on domains of models. The first algorithm introduces environmental domain embeddings to the keyword classifier. The second method applies correlation alignment loss to reduce the distortion of far-field speech. We also employ multi-task learning to predict keywords and domains simultaneously.

3.1. Environmental Domain Embeddings

In this subsection, we illustrate our approach that optimizes models with environmental domain embeddings derived from a pre-trained domain classifier. This method is inspired by [12], which explicitly learns the environmental knowledge with the introduction of noise embeddings to the acoustic model. In this paper, we extend this approach to the far-field word-level modeling task.

Our structure consists of two models: a domain classifier and a keyword classifier. Our domain classifier is optimized with the keyword speech samples recorded from different distances, including 0.25M, 1M, and 3M, which refers to different domain types. The classifier is constructed with a two-layer stacked LSTM structure, followed by an average pooling layer and a final linear layer. Domain embeddings are extracted from the output of the pooling layer. Through this structure, the acoustic features are transformed into a fix-dimensional representation with domain knowledge.

On the base of our CNN model, our keyword classifier is optimized with keyword speech samples and their environmental domain embeddings. Specifically, we extract the acoustic features from the speech and project them to embeddings with the pre-trained domain classifier. And then, the embeddings are concatenated to the output of the penultimate fully-connected layer. The concatenated features are finally fed into a linear layer for the keyword prediction. To further investigate where to concatenate, we also join the embeddings to the output of the last convolutional layer. Figure 1 illustrates the overall architecture.

3.2. Correlation Alignment

The mismatch of inner-class feature distributions on different domains contributes to the degradation of prediction performance. Focusing on this scenario, we apply the CORAL loss to constrain the embedding feature distortions from different domains in the manner of multi-task learning. In our case, we define the penultimate layer of the neural network as our feature layer for alignment loss computation.

CORAL is proposed to align the second-order statistics of the source and target distributions. [22] extend this work to DNN approaches by constructing a differentiable loss functions, which can be used to minimize the distance between outputs of embedding feature layer from different domains. Suppose the embedding features from source and target domains as D_S and D_T. And we denote the dimension of the feature layer as d and the covariance matrices of source and target features are C_S and C_T, respectively. The CORAL loss can then be defined as

$$l_{CORAL} = \frac{1}{4d^2} \left\| C_S - C_T \right\|_F^2, \tag{3}$$

where $\|\cdot\|_F^2$ denotes the squared matrix Frobenius norm. The covariance matrices of the source and target features [22] are

$$C_S = \frac{1}{n_S - 1} (D_S^\top D_S - \frac{1}{n_S} (\mathbf{1}^\top D_S)^\top (\mathbf{1}^\top D_S)), \tag{4}$$

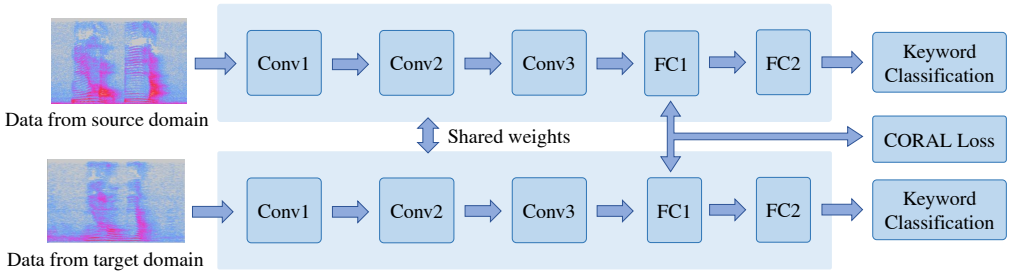

Figure 2: *Framework of the CORAL system.*

$$C_T = \frac{1}{n_T - 1}(D_T^\top D_T - \frac{1}{n_T}(\mathbf{1}^\top D_T)^\top(\mathbf{1}^\top D_T)), \quad (5)$$

where n_S and n_T represent the number of training samples of source and target domains. $\mathbf{1}$ refers to a column vector of all 1 elements.

In our work, we compute alignment loss on the outputs of the penultimate layer of the CNN network. Data from three different domains of 0.25m, 1m, and 3m are pooled together for training, and there are several strategies for the loss computation:

1. $L = L_{ce} + \lambda L_{coral}(E_{0.25M}, E_{1M})$
2. $L = L_{ce} + \lambda L_{coral}(E_{0.25M}, E_{3M})$
3. $L = L_{ce} + \lambda L_{coral}(E_{0.25m}, E_{1M\&3M})$
4. $L = L_{ce} + \lambda(L_{coral}(E_{0.25M}, E_{1M}) + L_{coral}(E_{0.25M}, E_{3M}))/2$
5. $L = L_{ce} + \lambda(L_{coral}(E_{0.25M}, E_{1M}) + L_{coral}(E_{0.25M}, E_{3M}) + L_{coral}(E_{1M}, E_{3M}))/3$

λ is the hyper-parameters representing the weight of alignment loss. The cross-entropy loss L_{ce} is calculated with the logits of data from both the source and target domains. $E_{0.25M}$, E_{1M} and E_{3M} refers to the embedding features used for CORAL calculation. $E_{1M\&3M}$ means that the $1M$ and $3M$ data are regarded as a group. By minimizing the joint loss, the inner-class embedding feature variabilities between close-talking and far-field domains would be reduced. Figure 2 illustrated the whole framework.

3.3. Multi-task learning

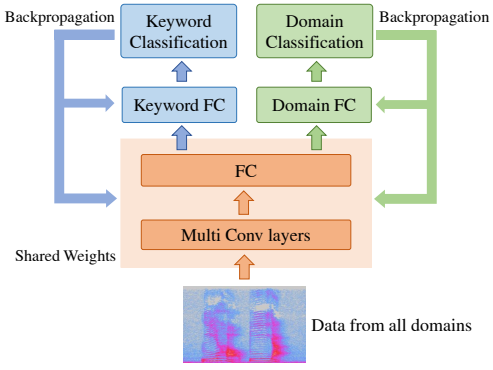

Figure 3: *Framework of the MTL system.*

Multi-task learning (MTL) is a mechanism that simultaneously optimizes the models to learn more than one task with a joint loss function. This method has been successfully applied in many speech-related tasks. In [12], it is implemented to classify the phonemes and the noise environments to improve the robustness of models toward noisy conditions. Inspired by this work, we perform the MTL algorithm to classify the domains and keywords simultaneously.

Figure 3 illustrate our MTL approach. On the base of the baseline CNN structure, an additional fully-connected layer is designed to predict the domain types. The output of the penultimate linear layer serves as a compressed representation with both word and domain information. The previous layers share the weights and are optimized jointly. In the training phase, we calculate the joint cross-entropy with the logits of both the domain and keyword classification. While decoding, only the prediction of keywords is computed.

Table 1: *MOS of data from different domains*

	0.25m	1m	3m
MOS	2.698	2.022	1.375

4. Experimental results

4.1. Data

Our proposed work is evaluated on a subset of the DMASH dataset [23], which is first proposed for the INTERSPEECH 2020 Far-Field Speaker Verification Challenge [23]. It contains audio of a wake-up word consisting of four Chinese characters, "ni hao, mi ya" ("Hello, Mia" in English) and other sentences that can be utilized as negative data. This dataset includes the speech data recorded by iPhone, Android, microphones, and microphone arrays from various distances. We utilize the recordings of the iPhone from a distance of 0.25m, 1m, and 3m, covering 222 speakers in the training set and 41 speakers in the test set. In our experiment, the 0.25m environment is regarded as close-talking (source domain), and 1m and 3m conditions, are viewed as far-field (target domain). See Table 2 for more details of dataset statistics. To objectively measure the data quality, we employ the P.563 algorithm [24] on the audio of different distances. Table 1 illustrate the mean opinion score (MOS) results.

4.2. Experiment setup

We determine target word labels by force-alignment with an LVCSR system trained with the AISHELL-2 dataset [25]. Here, for keyword "ni hao, mi ya", we find out the ending time of "ni", "hao", and "mi", and include its previous 20 frames and next 20 frames to construct a window of 40 frames. Log fbank

Table 2: *Dataset statistics.*

		utterances	positive	negative
Train	0.25m	178k	19k	159k
	1m	146k	15k	131k
	3m	143k	15k	128k
Evaluation	0.25m	37k	4k	33k
	1m	32k	4k	28k
	3m	31k	4k	27k

is adopted as our input acoustic features. The baseline system is trained with cross-entropy loss. Stochastic gradient descent with Nesterov momentum is selected as the optimizer. The learning rate is first initialized as 0.01 and decreases by a factor of 0.1 when the training loss plateau. We train the CNN model for 100 epochs with a batch size of 128 and employ early stopping when the training loss is not decreasing. In the evaluation period, we use a sliding window of 100 frames to compute the confidence score.

As the baseline system, we pool data from both close-talking and far-field conditions for training. In our experiments, for deep CORAL training, we set the weight λ to 0.2, 0.4, 0.6, 0.8, and 1.0, respectively. In our preliminary experiments, we find out that 0.8 is a suitable parameter, so our experiments on the CORAL loss are done under this weight. For MTL training, after the preliminary experiments with $\lambda = \{0.1, 0.2, 0.3, 0.4, 0.5\}$, we observe that the system achieves the best overall performance when $\lambda = 0.2$.

The performance is measured with the false reject (FR) rate under one false alarm (FA) per hour.

4.3. Results

Table 3: *Performance of the baseline system (the false reject (FR) rate (%) under one false alarm (FA) per hour)*

Training set	0.25M	1M	3M
Only 0.25M	1.29	2.91	11.6
Only 1M	2.03	1.58	7.77
Only 3M	10.9	8.00	10.6
Mix of 0.25M and 1M	**0.91**	**1.38**	6.06
Mix of 0.25M and 3M	1.54	1.97	**5.60**
Mix of all distances	1.41	1.64	6.33

The performance of the baseline system is illustrated in Table 3. From the results, we can obtain the following observations. First, with the increase of recording distance, the distortion becomes severer, and the performance of the baseline system degrades. Second, for the 0.25M and 1M datasets, when the training set and test set are from the same domain, the system performs better than the scenarios of domain mismatch. The network trained with only 3M datasets shows poor performance in every test set. Third, pooling the close-talking domain and target domain training data helps improve the performance on the target domain's test set. And the performance of the close-talking condition can still be maintained. The system trained with data from all fields has a balanced performance, while it is worse than the models trained with its target domains.

The results of our proposed system are shown in Table 4. The EMB1 system represents the system that concatenates domain embeddings to the output of the penultimate linear layer. And the EMB2 system concatenates to the output of the last convolutional layer. The CORAL1 to CORAL5 systems

Table 4: *Performances of models trained with different methods on the test sets*

Model name	0.25M	1M	3M
EMB1	1.11	1.59	4.99
EMB2	1.21	**1.02**	**4.11**
CORAL1	1.37	1.05	4.69
CORAL2	1.19	1.41	5.02
CORAL3	**1.09**	1.52	5.97
CORAL4	1.27	1.47	5.21
CORAL5	1.21	1.41	4.78
MTL	1.70	1.44	5.15

denotes the five different CORAL calculating approaches described in section 3. From the table, we can have the following findings. 1) In the domain embedding approaches, the EMB2 system outperforms the EMB1 system on the far-field conditions. The concatenation of embeddings in an early stage of the network helps the network better learn the domain information from the embeddings. 2) Among the CORAL systems, the CORAL1 system produces the best results on the far-field conditions. The CORAL2 system obtains worse scores than the CORAL1 system, which indicates that this method is sensitive to the domain types. From the results of the CORAL3 and CORAL4, we can see that regarding 1M and 3M datasets as a group are unhelpful to classification. The CORAL5 system calculates the CORAL loss for each pair of domains and achieves balanced results. 3) The MTL system obtains satisfying improvement on far-field speech while it has a relatively large degradation on the close-talking set.

Comparing different algorithms, we can find that systems based on domain embeddings achieve the best improvement, and the CORAL systems also outperform the baseline system on the far-field conditions. The CORAL system has the advantage that it does not require any extra network structures. The domain embedding system has an additional domain classifier, which increases the number of network parameters and the complexity of decoding computation. The MTL method is not as effective as the other two approaches.

5. Conclusions

In this paper, we concentrate on the task of small-footprint keyword spotting under the far-field environment. Far-field environments are commonly noticed in real-life speech applications, and it causes serve degradation of performance due to room reverberation and various kinds of noises. To cope with the distortions, we employ three domain aware training schemes, including learning with domain embeddings, with the CORAL loss, and MTL with inputs from different domains of data. Experimental results show that our methods manage to maintain the performance on the close-talking test dataset and achieve significant improvement in far-field conditions. Approaches with domain embeddings deliver the best performance while increasing the model size and computing cost. The CORAL systems also outperform the baseline system without changing the model structure, while it is sensitive to the domain types of data. The MTL approach is less effective than the other two methods.

6. Acknowledgment

This research was funded by Kunshan Government Research (KGR) Funding in AY 2019/2020.

7. References

[1] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. ICASSP*, 2014, pp. 4087–4091.

[2] M. Sun, D. Snyder, Y. Gao, V. K. Nagaraja, M. Rodehorst, S. Panchapagesan, N. Strom, S. Matsoukas, and S. Vitaladevuni, "Compressed time delay neural network for small-footprint keyword spotting," in *Proc. INTERSPEECH*, 2017, pp. 3607–3611.

[3] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. INTERSPEECH*, 2015, pp. 1478–1482.

[4] S. Ö. Arık, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger, and A. Coates, "Convolutional recurrent neural networks for small-footprint keyword spotting," 2017, pp. 1606–1610.

[5] M. Sun, A. Raju, G. Tucker, S. Panchapagesan, G. Fu, A. Mandal, S. Matsoukas, N. Strom, and S. Vitaladevuni, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in *Proc. SLT*, 2016, pp. 474–480.

[6] B. Liu, S. Nie, Y. Zhang, S. Liang, Z. Yang, and W. Liu, "Focal loss and double-edge-triggered detector for robust small-footprint keyword spotting," in *Proc. ICASSP*, 2019, pp. 6361–6365.

[7] Z. Wang, X. Li, and J. Zhou, "Small-footprint keyword spotting using deep neural network and connectionist temporal classifier," in *arXiv preprint arXiv:1709.03665*, 2017.

[8] C. Shan, J. Zhang, Y. Wang, and L. Xie, "Attention-based end-to-end models for small-footprint keyword spotting," in *Proc. INTERSPEECH*, 2018, pp. 2037–2041.

[9] X. Wang, S. Sun, C. Shan, J. Hou, L. Xie, S. Li, and X. Lei, "Adversarial examples for improving end-to-end attention-based small-footprint keyword spotting," in *Proc. ICASSP*, 2019, pp. 6366–6370.

[10] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in *Proc. ICASSP*, 2018, pp. 5484–5488.

[11] A. Coucke, M. Chlieh, T. Gisselbrecht, D. Leroy, M. Poumeyrol, and T. Lavril, "Efficient keyword spotting using dilated convolutions and gating," in *Proc. ICASSP*, 2019, pp. 6351–6355.

[12] S. Kim, B. Raj, and I. Lane, "Environmental noise embeddings for robust speech recognition," *arXiv preprint arXiv:1601.02553*, 2016.

[13] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7398–7402.

[14] D. Cai, W. Cai, and M. Li, "Within-sample variability-invariant loss for robust speaker recognition under noisy environments," in *Proc. ICASSP*, 2020, pp. 6469–6473.

[15] Y. Qian, T. Tan, and D. Yu, "An investigation into using parallel data for far-field speech recognition," in *Proc. ICASSP*, 2016, pp. 5725–5729.

[16] V. Peddinti, V. Manohar, Y. Wang, D. Povey, and S. Khudanpur, "Far-field asr without parallel data." in *Proc. INTERSPEECH*, 2016, pp. 1996–2000.

[17] J. Li, R. Zhao, Z. Chen, C. Liu, X. Xiao, G. Ye, and Y. Gong, "Developing far-field speaker system via teacher-student learning," in *Proc. ICASSP*, 2018, pp. 5699–5703.

[18] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, pp. 41–75, 1997.

[19] S. Panchapagesan, M. Sun, A. Khare, S. Matsoukas, A. Mandal, B. Hoffmeister, and S. Vitaladevuni, "Multi-task learning and weighted cross-entropy for dnn-based keyword spotting." in *Proc. INTERSPEECH*, vol. 9, 2016, pp. 760–764.

[20] R. Giri, M. L. Seltzer, J. Droppo, and D. Yu, "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning," in *Proc. ICASSP*. IEEE, 2015, pp. 5014–5018.

[21] R. Prabhavalkar, R. Alvarez, C. Parada, P. Nakkiran, and T. N. Sainath, "Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks," in *Proc. ICASSP*, 2015, pp. 4704–4708.

[22] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. ECCV*, 2016, pp. 443–450.

[23] X. Qin, M. Li, H. Bu, W. Rao, R. K. Das, S. Narayanan, and H. Li, "The interspeech 2020 far-field speaker verification challenge," in *Proc. INTERSPEECH*, 2020.

[24] L. Malfait, J. Berger, and M. Kastner, "P. 563—the itu-t standard for single-ended speech quality assessment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 1924–1934, 2006.

[25] J. Du, X. Na, X. Liu, and H. Bu, "Aishell-2: transforming mandarin asr research into industrial scale," *arXiv preprint arXiv:1808.10583*, 2018.