

REDUCING LANGUAGE CONFUSION FOR CODE-SWITCHING SPEECH RECOGNITION WITH TOKEN-LEVEL LANGUAGE DIARIZATION

Hexin Liu^{1,2}, Haihua Xu¹, Leibny Paola Garcia³, Andy W. H. Khong², Yi He¹, Sanjeev Khudanpur³

¹Bytedance

²School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

³CLSP and HLT-COE, Johns Hopkins University, USA

ABSTRACT

Code-switching (CS) occurs when languages switch within a speech signal and leads to language confusion for automatic speech recognition (ASR). We address the problem of language confusion for improving CS-ASR from two perspectives: incorporating and disentangling language information. We incorporate language information within the CS-ASR model by dynamically biasing the model with token-level language posteriors corresponding to outputs of a sequence-to-sequence auxiliary language diarization (LD) module. In contrast, the disentangling process reduces the difference between languages via adversarial training so as to normalize two languages. We conduct experiments on the SEAME dataset. Compared to the baseline model, both the joint optimization with LD and the language posterior bias achieve performance improvement. Comparison of the proposed methods indicates that incorporating language information is more effective than disentangling for reducing language confusion in CS speech.

Index Terms— code-switching, automatic speech recognition, token, language diarization, language posterior

1. INTRODUCTION

Code-switching (CS) refers to the switching of languages within a spontaneous multilingual recording. Although existing automatic speech recognition (ASR) methods have shown to achieve good performance on monolingual speech [1, 2], CS-ASR is still a challenge due to language confusion arising from code switches and the lack of annotated data.

Language information is often incorporated into CS-ASR models to tackle challenges associated with language confusion. In [3], language identification (LID) serves as an auxiliary task which enriches the shared encoder with language information. A bi-encoder transformer network was proposed in [4], where two encoders are pre-trained on monolingual data independently to decouple the modeling of Mandarin and English for the capture of language-specific information. This dual-encoder CS-ASR approach has shown to be effective and several methods were subsequently proposed

based on this framework [5, 6, 7]. A language-specific attention mechanism has also been proposed to reduce multilingual contextual information for a transformer encoder-decoder CS-ASR model [8, 9]. In this approach, monolingual token embeddings are separated from code-switching token sequences before being fed into their respective self-attention modules within the decoder layers.

It is useful to note that the dual-encoder approach, in general, performs LID at frame-level units—frame-level LID outputs are assigned to the outputs of language-specific encoders before the weighted sum in the mixture-of-experts interpolation process. The frame-level LID, however, is not desirable since the LID performance generally degrades with shorter speech signals [10, 11]. In addition, CS can be regarded as a speaker-dependent phenomenon [12], where languages within a CS speech signal share information such as the accent and discourse markers. Therefore, the language-specific attention mechanism may lead to cross-lingual information loss while learning monolingual information. Due to the nature of languages and their transitions in multilingual recordings, exploiting CS approaches at a lower-granularity token level would be more appropriate for CS-ASR.

Language diarization (LD), as a special case of LID, involves partitioning a code-switching speech signal into homogeneous segments before determining their language identities [13, 14]. In our work, LD is reformulated into a sequence-to-sequence task similar to that of ASR to capture utterance-level language information. Inspired by the success of utterance-level one-hot language vector for multilingual ASR [15, 16], we propose to reduce language confusion within CS speech by supplementing the token embeddings with their respective soft language labels—token-level language posteriors predicted by the LD module—before feeding these embeddings into the ASR decoder. Since two languages in a CS scenario can be auditorially similar due to the accent and tone of the bilingualist, language posteriors are expected to convey more language information than one-hot language label vectors. Moreover, to explore the effect of language information for CS-ASR, we also propose a second technique to disentangle the language information from the encoder mod-

ule via adversarial training. The disentangling process aims to normalize two languages such that language confusion resulting from CS can be decreased. Performance evaluations of the proposed methods allow one to gain insights into how language information reduces language confusion for CS-ASR.

2. THE HYBRID CTC/ATTENTION ASR MODEL

The hybrid CTC/attention ASR model comprises an encoder module, a decoder module, and a connectionist temporal classification (CTC) module [2, 17]. In our work, the encoder and decoder modules comprise conformer encoder layers and transformer decoder layers [18, 19], respectively.

Given a speech signal, we define its acoustic features $\mathbf{X} = (\mathbf{x}_t \in \mathbb{R}^F | t = 1, \dots, T)$ and token sequence $W = (w_n \in \mathcal{V} | n = 1, \dots, N)$, where \mathcal{V} is a vocabulary of size V , T and N are the lengths of acoustic features and token sequence, respectively. These tokens are transformed into D -dimensional token embeddings $\mathbf{W} = (\mathbf{w}_n \in \mathbb{R}^D | n = 1, \dots, N)$ before being fed into the decoder module. The encoder generates $\mathbf{H} = (\mathbf{h}_t \in \mathbb{R}^D | t = 1, \dots, T_1)$ from inputs \mathbf{X} , which are then used as inputs for the decoder and CTC modules. During training, masked self-attention of the token embeddings and the cross-attention between hidden output and token embeddings are performed within the decoder layer. Here, the masked self-attention process prevents the current tokens from attending to future tokens. The decoder predicts the next token w_n based on historical tokens $w_{1:n-1}$ and \mathbf{H} via

$$p(w_n | w_{1:n-1}, \mathbf{X}) = \text{Decoder}(w_{1:n-1}, \mathbf{H}), \quad (1)$$

where $p(w_n | w_{1:n-1}, \mathbf{X})$ is the posterior of decoding w_n given acoustic features and historical tokens, and $\text{Decoder}(\cdot)$ denotes the transformer-based ASR decoder. The model is optimized via a multi-task objective function

$$\mathcal{L}_{\text{asr}} = \alpha \mathcal{L}_{\text{ctc}} + (1 - \alpha) \mathcal{L}_{\text{att}}, \quad (2)$$

where \mathcal{L}_{ctc} and \mathcal{L}_{att} denote the CTC loss and encoder-decoder cross-entropy loss with label smoothing, respectively, and α is a parameter associated with multi-task learning. The decoding process is defined to maximize the linear combination of the logarithmic CTC and attention objectives

$$\widehat{W} = \underset{W}{\text{argmax}} \{ \alpha \log p_{\text{ctc}}(W | \mathbf{X}) + (1 - \alpha) \log p_{\text{att}}(W | \mathbf{X}) \}. \quad (3)$$

3. REDUCING LANGUAGE CONFUSION

3.1. Sequence-to-sequence language diarization

As shown in Fig. 1, given the encoder outputs \mathbf{H} and the token sequence W with the corresponding token embeddings $\mathbf{W}^{\text{ld}} = (\mathbf{w}_n^{\text{ld}} \in \mathbb{R}^D | n = 1, \dots, N)$, we propose to decode the token-level language labels $L = (l_n \in \mathcal{V}^{\text{ld}} | n = 1, \dots, N)$ of W via a transformer decoder being denoted as

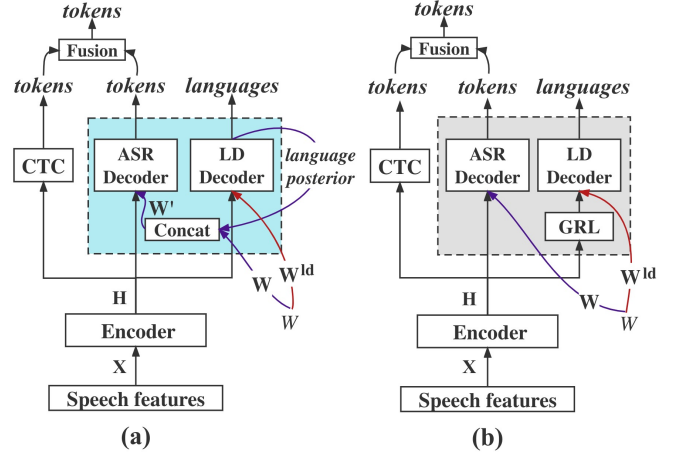


Fig. 1. The hybrid CTC/attention model with (a) incorporating language information using language posterior bias, and (b) disentangling language via adversarial training.

“LD decoder” in the proposed model. Here, \mathcal{V}^{ld} denotes the language vocabulary of size V^{ld} .

While tokens within a sequence possess contextual information for the ASR task, they share the language identities in the LD task, i.e., they should either be of the same or different languages. Such information is expected to be available in both self-attention and cross-attention processes. In addition, masking future tokens is required when training the ASR decoder to minimize the mismatch between training and decoding. Masking future tokens during training, however, results in a lower triangular self-attention weight matrix leading to language-related information loss. Therefore, we propose to utilize the complete token sequence to train the LD decoder during the self-attention process if the LD decoder participates in only multi-task optimization but not the decoding process. To this end, we have

$$p(l_n | w_{1:N}, \mathbf{X}) = \text{Decoder}^{\text{ld}}(w_{1:N}^{\text{ld}}, \mathbf{H}), \quad (4)$$

where $\text{Decoder}^{\text{ld}}(\cdot)$ denotes the LD decoder. On the other hand, if the LD decoder participates in the decoding phase, future context should be masked to reduce the mismatch between training and decoding. The LD decoder is then jointly optimized with the hybrid CTC/attention CS-ASR model via

$$\mathcal{L}_{\text{joint}} = \alpha \mathcal{L}_{\text{ctc}} + (1 - \alpha) \mathcal{L}_{\text{att}} + \beta \mathcal{L}_{\text{ld}}, \quad (5)$$

where β is a multi-task learning parameter, and \mathcal{L}_{ld} is a label-smoothed cross-entropy loss between the predicted and ground-truth language labels for the LD decoder.

3.2. Language posterior bias

The utterance-level language vector has shown to be effective for multilingual ASR [15, 16]. Since intra-sentence code-switching occurs at word level, we propose to bias the ASR output using token-level language posteriors predicted by the LD decoder as shown in Fig. 1 (a).

Given historical tokens $w_{1:n-1}$ and hidden output \mathbf{H} , the LD decoder generates a V^{ld} -dimensional language posterior vector $\mathbf{p}(l_{n-1}|w_{1:n-1}, \mathbf{X})$ after a softmax function. Token embedding \mathbf{w}_{n-1} is subsequently concatenated with its language posterior vector \mathbf{p} before being fed into the ASR decoder. The ASR decoder output is next computed via

$$\mathbf{w}'_{n-1} = \text{Concat}(\mathbf{w}_{n-1}, \mathbf{p}(l_{n-1}|w_{1:n-1}, \mathbf{X})), \quad (6)$$

$$p(w_n|w_{1:n-1}, \mathbf{X}) = \text{Decoder}(\mathbf{w}'_{1:n-1}, \mathbf{H}), \quad (7)$$

where $\text{Concat}(\cdot)$ denotes the concatenation operation and $\mathbf{W}' = (\mathbf{w}'_n \in \mathbb{R}^{D+V^{\text{ld}}}|n = 1, \dots, N)$ are input token embeddings of the ASR decoder which are subsequently projected back to D dimensions by a linear layer. The model is optimized via (5) and the decoding process is similar to (3) but with input token embeddings \mathbf{W} of the ASR decoder being replaced with \mathbf{W}' .

3.3. Disentangling with adversarial training

In contrast to incorporating language information, disentangling language information aims to normalize the two languages by reducing the difference between them. To disentangle language information from the encoder, we employ a gradient reversal layer (GRL) between the encoder and LD decoder as shown in Fig. 1(b). The GRL was originally proposed for unsupervised domain adaptation in [20]—it remains in the forward process while the gradient is being reversed during backpropagation. In our CS-ASR context, the model achieves adversarial training that reduces the language information within the encoder while optimizing the LD decoder to be discriminative.

4. DATASET, EXPERIMENT, AND RESULTS

4.1. Dataset and experiment setup

We conducted all experiments on the SEAME dataset—a Mandarin-English code-switching corpus containing spontaneous conversational speech [21]. Both intra- and inter-sentence code-switching speech exist within this dataset. We divide this dataset into a 98.02-hour training set, a 5.09-hour validation set, and two test sets dev_{man} and dev_{sge} in the same manner as [3]. Details of the test sets are presented in Table 1.

We implemented the models and evaluated them on ESPnet¹ [22]. During training and inference, we followed the experiment setup of the espnet2 SEAME recipe with regard to data preprocessing and model configuration. Speech perturbation (with factors 0.9, 1.0, and 1.1) and SpecAugment are applied as data augmentation [23, 24]. Words are transformed into a total of $V = 5,628$ tokens including 3,000 English byte-pair encoding (BPE) tokens, 2,624 Mandarin characters, and 4 special tokens for *unk*, *noise*, *blank*, and *sos/eos*. All tokens are transformed to build \mathcal{V}^{ld} comprising e for English BPEs, m for Mandarin characters, *sos/eos*, and *other* for other

¹Source code: https://github.com/Lhx94As/reducing_language_confusion

Table 1. Details of two test sets divided from SEAME

	Speakers	Hours	Duration ratio (%)		
			Man	Eng	CS
dev_{man}	10	7.49	14	7	79
dev_{sge}	10	3.93	6	41	53

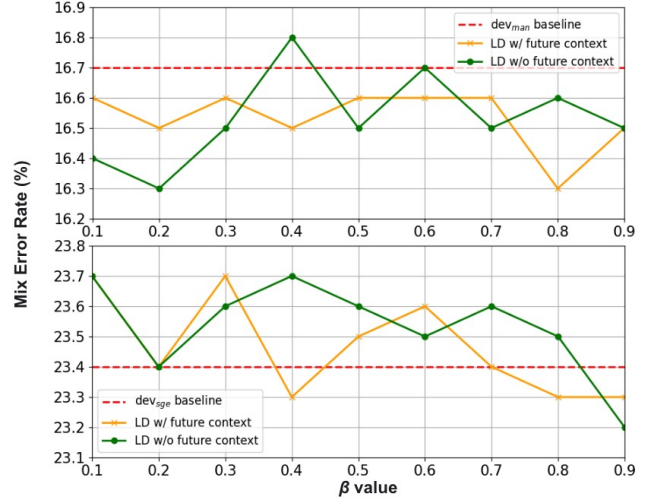


Fig. 2. Performance evaluation of the CS-ASR-LD models with different β values by employing MER (%), where orange and green lines denote training the LD decoders with and without future context in the self-attention process, respectively.

tokens. Tokens in \mathcal{V}^{ld} are used as LD outputs. We extracted $F = 83$ dimensional features comprising 80-dimensional logfbanks and 3-dimensional pitch for each speech segment before applying global mean and variance normalization.

The baseline model is a hybrid CTC/Attention ASR model comprising twelve conformer encoder layers and six transformer decoder layers [18, 19, 25]. The proposed LD decoder employs the same configuration as the ASR decoder. All self-attention encoder and decoder layers have four attention heads with input and output dimensions being $D = 256$, and the inner layer of the position-wise feed-forward network is of dimensionality 2048. The language model (LM) is a sixteen-layer transformer model with each attention layer comprising eight heads. Parameters $\alpha = 0.3$ and a label smoothing factor of 0.1 are used in (2) and (5). All models are trained for 50 epochs on two V100 GPUs. During inference, the 10-best models during validation are averaged. We adopted the 10-best beam search method with $\alpha = 0.4$ in (3). The proposed systems are evaluated by employing mix error rate (MER) comprising word error rate (WER) for English and character error rate (CER) for Mandarin.

4.2. Results of jointly optimizing CS-ASR and LD

We denote the hybrid CTC/attention CS-ASR model which is jointly trained with LD as CS-ASR-LD. The CS-ASR-LD model is evaluated on dev_{man} and dev_{sge} to explore the effect

of the auxiliary LD task with different β values. In addition, we compare the CS-ASR performance with different LD decoders trained on tokens with and without future context, respectively. The above results are presented in Fig. 2.

As opposed to the CS-ASR-LD model with an LD decoder that masks future context during training, those with the LD decoder trained on full context consistently exhibit lower MER on dev_{man} than the baseline. In addition, the CS-ASR-LD model with an LD decoder that does not mask future context during training achieves moderate performance improvement on dev_{sge} with high β values compared to the baseline. The best overall performance of 16.3% and 23.3% on dev_{man} and dev_{sge} , respectively, is achieved when training the LD decoder on full context with $\beta=0.8$. Since dev_{man} contains mostly CS speech, the above results indicate that incorporating discriminative language information benefits ASR performance on CS speech.

4.3. Results of biasing CS-ASR with language posterior

We evaluate the hybrid CTC/attention model with language posterior bias (LPB), which is the LD decoder output, on dev_{man} and dev_{sge} . These results and ablation studies are shown in Table 2 with $\beta = 0.8$. Here, employing LPB without joint optimization with LD is achieved by intercepting the backpropagation from the LD decoder to the encoder.

Compared with the baseline, employing either LD or LPB exhibits higher performance on dev_{man} which contains mainly CS test speech. Systems S2.4 and S2.5 suffer from moderate performance degradation on dev_{sge} . Since dev_{sge} comprises predominantly monolingual speech with less language confusion compared to dev_{man} , the above results imply that utilizing token-level language information, while beneficial for CS-ASR, may not contribute effectively to monolingual ASR.

System S2.8 with both LD and LPB achieves the best performance of 16.3% and 23.0% on dev_{man} and dev_{sge} , respectively, among all systems that do not employ LM. This indicates the efficacy of our proposed method. In addition, while employing LM improves the CS-ASR performance, the proposed language-aware methods do not benefit from the use of LM as much compared to the baseline model. This also implies that a language-aware LM would be more appropriate to CS-ASR compared to the general LM [26].

4.4. Results of disentangling language information

Language information is disentangled from the CS-ASR model via adversarial training with a GRL. We evaluated this approach with $\beta = 0.2, 0.5$, and 0.8 and results are shown in Table 3.

Disentangling language information from the CS-ASR model (while maintaining the MER on dev_{man}) results in performance degradation on dev_{sge} compared to the baseline. Since dev_{sge} contains more monolingual speech than dev_{man} , these results suggest that language information contributes to

Table 2. Performance comparison between baseline model and those with language posterior bias (LPB) by employing MER (%), “Future context” denotes whether the future tokens are used or not for the LD decoder during training

Index	Model	β	Future context	LM	MER	
					dev_{man}	dev_{sge}
S0	baseline	-	-	No	16.7	23.4
S1		-	-	Yes	16.5	23.0
S2.1	+LD	0.8	Yes	No	16.3	23.3
S2.2		0.8	No	No	16.6	23.5
S2.3		0.8	Yes	Yes	16.2	23.1
S2.4	+LPB	0.8	Yes	No	16.4	23.5
S2.5		0.8	No	No	16.6	23.5
S2.6		0.8	No	Yes	16.4	23.2
S2.7		0.8	Yes	No	16.6	23.4
S2.8	+LD +LPB	0.8	No	No	16.3	23.0
S2.9		0.8	Yes	Yes	16.5	22.9
S2.10		0.8	No	Yes	16.1	22.8

Table 3. Performance comparison between baseline model and those after disentangling language information by employing MER (%), “Future context” denotes whether the future tokens are used or not for the LD decoder during training

Index	Model	β	Future context	MER	
				dev_{man}	dev_{sge}
S0	baseline	-	-	16.7	23.4
S3.1	+LD+GRL	0.2	Yes	16.8	23.6
S3.2		0.5	Yes	16.7	23.7
S3.3		0.8	Yes	16.7	23.7

the CS-ASR performance even for monolingual speech. This is because the process of ASR intrinsically encodes language-related information such as phonotactics and syntax into the model [27]. Reducing the language information, therefore, may degrade the ASR performance on monolingual speech.

5. CONCLUSION

We proposed two approaches to reduce language confusion for CS-ASR. The first method incorporates language information using the language posterior bias, while the second method disentangles language information via adversarial training. Compared to the baseline model, our proposed LPB method in conjunction with multi-task learning exhibits higher performance on the SEAME dataset. Comparison between the proposed LPB method and the adversarial training for CS-ASR-LD highlights the efficacy of incorporating language information to improve CS-ASR. In addition, the proposed language-aware CS-ASR models achieve lower performance improvement after employing LM during decoding compared to the baseline model. This indicates that general LM may not be suitable for language-aware CS-ASR models due to the lack of language information.

6. REFERENCES

- [1] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2011.
- [2] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 8, pp. 1240–1253, 2017.
- [3] Z. Zeng, Y. Khassanov, V. T. Pham, H. Xu, E. S. Chng, and H. Li, "On the end-to-end solution to mandarin-english code-switching speech recognition," in *Proc. Interspeech*, 2019, pp. 2165–2169.
- [4] 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.
- [5] M. S. Mary N J, V. M. Shetty, and S. Umesh, "Investigation of methods to improve the recognition performance of tamil-english code-switched data in transformer framework," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 7889–7893.
- [6] T. Song, Q. Xu, M. Ge, L. Wang, H. Shi, Y. Lv, Y. Lin, and J. Dang, "Language-specific characteristic assistance for code-switching speech recognition," in *Proc. Interspeech*, 2022, pp. 3924–3928.
- [7] S. Dalmia, Y. Liu, S. Ronanki, and K. Kirchhoff, "Transformer-transducers for code-switched speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 5859–5863.
- [8] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A non-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5884–5888.
- [9] S. Zhang, J. Yi, Z. Tian, J. Tao, Y. T. Yeung, and L. Deng, "Reducing multilingual context confusion for end-to-end code-switching automatic speech recognition," in *Proc. Interspeech*, 2022, pp. 3894–3898.
- [10] S. O. Sadjadi, T. Kheyrkhan, C. S. Greenberg, E. Singer, D. A. Reynolds, L. P. Mason, and J. Hernandez-Cordero, "Performance analysis of the 2017 NIST language recognition evaluation," in *Proc. Interspeech*, 2018, pp. 1798–1802.
- [11] H. Liu, L. P. G. Perera, A. W. H. Khong, S. J. Styles, and S. Khudanpur, "PHO-LID: A unified model incorporating acoustic-phonetic and phonotactic information for language identification," in *Proc. Interspeech*, 2022, pp. 2233–2237.
- [12] N. T. Vu, H. Adel, and T. Schultz, "An investigation of code-switching attitude dependent language modeling," in *Proc. SLSP*, 2013, pp. 297–308.
- [13] D. Lyu, C. E. Siong, and H. Li, "Language diarization for code-switch conversational speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7314–7318.
- [14] H. Liu, L. P. G. Perera, X. Zhang, J. Dauwels, A. W. H. Khong, S. Khudanpur, and S. J. Styles, "End-to-end language diarization for bilingual code-switching speech," in *Proc. Interspeech*, 2021, pp. 1489–1493.
- [15] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 4904–4908.
- [16] 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.
- [17] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 369–376.
- [18] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020.
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [20] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.
- [21] D.-C. Lyu, T.-P. Tan, E. S. Chng, and H. Li, "SEAME: a Mandarin-English code-switching speech corpus in South-East Asia," in *Proc. Interspeech*, 2010, pp. 1986–1989.
- [22] 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.
- [23] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015, pp. 3586–3589.
- [24] 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.
- [25] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," in *Proc. Interspeech*, 2019, pp. 1408–1412.
- [26] H. Adel, N. T. Vu, K. Kirchhoff, D. Telaar, and T. Schultz, "Syntactic and semantic features for code-switching factored language models," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 431–440, 2015.
- [27] H. Liu, L. P. G. Perera, A. W. H. Khong, E. S. Chng, S. J. Styles, and S. Khudanpur, "Efficient self-supervised learning representations for spoken language identification," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1296–1307, 2022.