EURASIP Journal on Audio, Speech, and Music Processing

METHODOLOGY

Open Access

Three-stage training and orthogonality regularization for spoken language recognition

Zimu Li[1,2], Yanyan Xu[1,2*], Dengfeng Ke[3] and Kaile Su[4]

Abstract

Spoken language recognition has made significant progress in recent years, for which automatic speech recognition has been used as a parallel branch to extract phonetic features. However, there is still a lack of a better training strategy for such architectures of two individual branches. In this paper, we analyze the mostly used two-stage training strategies and reveal a trade-off between the recognition accuracy and the generalization ability. Based on the analysis, we propose a three-stage training strategy and an orthogonality regularization method. The former adds a multi-task learning stage to the traditional two-stage training strategy to extract hybrid-level and noiseless features, which can improve the recognition accuracy on the basis of maintaining the generalization ability, while the latter constrains the orthogonality of base vectors and introduces prior knowledge to improve the recognition accuracy. Experiments on the Oriental Language Recognition (OLR) dataset indicate that these two proposed methods can improve both the language recognition accuracy and the generalization ability, especially in complex challenge tasks, such as cross-channel or noisy conditions. Also, our model, which combines these two proposed methods, performs better than the top three teams in the OLR20 challenge.

Keywords Spoken language recognition, Automatic speech recognition, Three-stage training, Orthogonality regularization, Multi-task learning

1 Introduction

Spoken language recognition (also called Language Identification, LID) is to identify the language spoken in an utterance [1], which can be used as a pre-processing step in many multilingual applications, such as speech translation [2] and multilingual speech recognition [3].

Classical end-to-end LID methods are composed of a feature extractor that maps variable-length speech segments to fixed-length embeddings [4], and an identification module which makes the decision. Different levels of features are beneficial for LID [5, 6], including acoustic features [7, 8], phonetic features [9], syntactic features [10], and prosodies [11]. Among them, acoustic features are the most commonly used features in classical end-to-end LID systems [4, 12, 13].

Although classical end-to-end LID methods directly utilize acoustic features as input, phonetic features can better reveal the basic difference between languages, that is, the distribution or frequency of phones [5]. Since phonetic features represent information at a higher level than acoustic features, they are more robust for noise and channels [9]. Therefore, recent end-to-end methods mainly focus on phonetic features [14] (also called bottleneck features [15]), which are extracted by an automatic speech recognition (ASR) task.

While the classical end-to-end LID methods using acoustic features can implicitly learn phonetic knowledge

*Correspondence:
Yanyan Xu
xuyanyan@bjfu.edu.cn
[1] School of Information Science and Technology, Beijing Forestry University, 35 Qing-Hua East Road, 100083 Beijing, China
[2] Engineering Research Center for Forestry-oriented Intelligent Information Processing of National Forestry and Grassland Administration, Beijing Forestry University, 35 Qing-Hua East Road, 100083 Beijing, China
[3] School of Information Science, Beijing Language and Culture University, 15 Xueyuan Road, 100083 Beijing, China
[4] Institute for Integrated and Intelligent Systems, Griffith University, Nathan 4111, QLD, Australia

as well, phonetic features are more easily discovered by an ASR task with frame-level phonemic labels, considering utterance-level linguistic labels might be too coarse [16]. ASR tasks can efficiently extract phonetic features, and noise such as speaker information can be filtered. Phonetic features extracted from one of the hidden layers of a pre-trained ASR model can be interpreted as compression of phonetic information [17–19], and they are much richer, that is, they are at frame-level and involve compacted information of all phones [9].

Lately, end-to-end ASR models have achieved outstanding performances and largely simplified multilingual models by learning shared representations directly from data [20, 21]. These studies have greatly simplified the process of extracting phonetic features for LID tasks, making it one of the hot-spots in recent years. Watanabe et al. [22] present a single multilingual model with a unique vocabulary that can recognize speeches of 10 languages. Multi-task learning that jointly learns linguistic and phonetic information is studied in [23, 24]. Ren et al. [25] build a two-stage language identification system that outperforms the baseline multi-task system. Wang et al. [26] analyze different conformer-based architectures and demonstrate the great improvement of a two-stage system, named transfer learning in their experiments. Duroselle et al. [16, 27] study different modeling and training strategies and show that bottleneck features can be greatly improved by using language identification loss during the training of the feature extractor. Alum et al. [28] incorporate a large pre-trained multilingual XLSR-53 wav2vec2.0 model and reveal its excellent modeling abilities, that is, fine-tuning the model with just one utterance per target language already outperforms the baseline model that does not use pre-training but is trained with around 10,000 utterances per language.

Although ASR has been applied as an auxiliary task to LID, how to better optimize these two independent LID and ASR neural networks has not yet been well studied. Most of the existing methods use a two-stage training strategy, which is not optimal and faces a trade-off between the recognition accuracy and the generalization ability.

Therefore, in this paper, we introduce a three-stage training strategy with an orthogonality regularization method to overcome the problem of the trade-off between the recognition accuracy and the generalization ability, which we analyze is caused by a trade-off between hybrid-level knowledge, that is, various mixed-level features including phonetic information and prosodies, and noise embedded in a shared encoder. Existing methods can learn only single-level phonetic features or only noisy hybrid-level features, but the methods proposed in this paper can learn hybrid-level and noiseless features, so they overcome the trade-off mentioned above and outperform the existing methods.

The main contributions of this paper are summarized as follows.

- This paper systematically analyzes the phonetic features and hybrid-level features extracted by the ASR task and discusses the trade-off between the recognition accuracy and the generalization ability for optimizing the LID model with an auxiliary ASR task.
- A novel three-stage training strategy is proposed to learn hybrid-level knowledge by adding a multi-task learning stage to the traditional two-stage strategy. The supervision of the ASR task makes an encoder encode hybrid-level features and exclude most of noise. Meanwhile, a frozen encoder in the final stage ensures that there is almost no new noise being learned. Such hybrid-level and noiseless features can guarantee both the recognition accuracy and the generalization ability.
- The orthogonality regularization method is introduced to improve the performances of both the two-stage and three-stage training strategies. By adding prior knowledge to model a better embedding space, the language classifier can achieve a higher accuracy.

The rest of the paper is organized as follows. An overview of joint speech and language recognition architectures is described in Section 2. Then, our proposed methods are discussed in Section 3. Section 4 details our whole multi-task architecture, and the experiments are illustrated in Section 5. Finally, in Section 6, we conclude this paper.

2 Related works

2.1 The ASR-LID parallel branches architecture

Although the end-to-end LID architecture (as shown in Fig. 1a) has achieved great success in the past few years, it still has the problem of domain mismatch [29, 30]. Domain information about speakers, genders, channels, and other kinds of noise may have different distributions between the training set and the real environment. An ideal LID system should ignore such domain-related noise, which may cause a poor generalization ability. As phonetic features are independent of noise, they can be used to boost the LID performance [9].

Directly using LID to extract frame-level phonetic information from raw acoustic features is quite inefficient, for the utterance-level language labels are too coarse to provide sufficient supervision [9]. Considering the ASR task runs at the frame level, the output of hidden layers from a well-trained ASR model can incorporate phonetic information and be robust to noise. Such an architecture can provide a dramatic performance

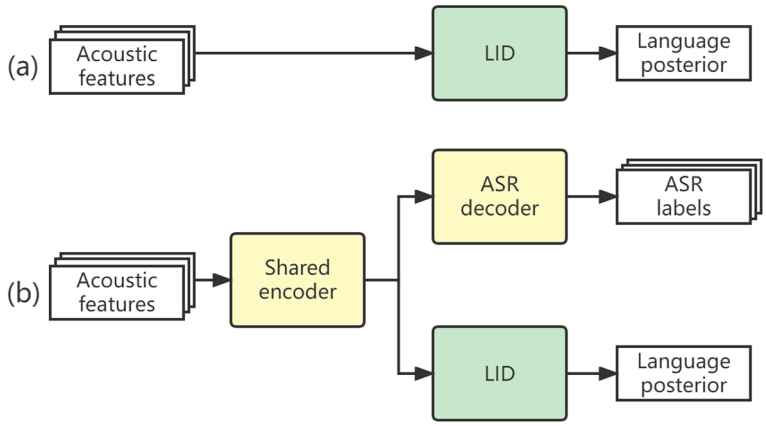

Fig. 1 **a** The end-to-end LID architecture. It directly gets utterance-level LID decisions from frame-level acoustic features. **b** The ASR-LID parallel branches architecture. The shared encoder trained with ASR-loss can produce frame-level phonetic information, which helps to improve the LID performance dramatically

improvement over end-to-end LID systems [9, 23, 26, 31–33].

The architecture of phonetic LID is generally composed of two branches: the one for the LID task and the other for the ASR task, as shown in Fig. 1b. Phonetic features and other high-level knowledge are extracted by the shared encoder and then fed into the two branches. The ASR branch tends to model phonetic features to the encoder, but the LID branch tends to model hybrid-level features with domain noise. How to optimize the two individual tasks is still worth studying.

2.2 Training strategies

Traditional training methods utilize a two-stage training strategy [15], which updates parts of the neural network using ASR-loss and LID-loss one after another. In the first stage, the shared encoder is trained with an ASR decoder using a hybrid CTC/attention loss. Acoustic features are compassed into phonetic features, and other information is treated as noise and filtered out. In the second stage, the LID model, with or without the shared encoder, is trained using a softmax cross-entropy loss.

Both frozen and unfrozen encoders are selected in recent years, as shown in Table 1 ("-F" means the frozen encoder is used, otherwise the unfrozen encoder). If the shared encoder is frozen, it will not be trained in the second stage, and features extracted by the encoder are still single-level phonetic features. Otherwise, if the shared encoder is unfrozen and is trained together with the following LID model, the model can extract other higher-level features, including phonetic features, and all the extracted features are called hybrid-level features. Predictably, noise is naturally included in such end-to-end learning.

Table 1 Previous proposed methods

	Year	encoder	ASR-loss	Strategy
Ren [25]	2019	ResNet	CTC	2-stage-F
Zhao [23]	2019	TDNN	Cross-entropy	Multi-task
Wang [26]	2021	Conformer	Hybrid CTC/attention	2-stage
Li [24]	2021	TDNN	Cross-entropy	Multi-task
Duroselle [27]	2021	Conformer	CTC	2-stage-F
Duroselle [16]	2021	Conformer	CTC	2-stage
Alum [28]	2022	Conformer	Unknown	2-stage

3 Proposed methods

3.1 A trade-off between the recognition accuracy and the generalization ability

Table 1 lists the details of the recent proposed methods. It can be observed that these methods differ in whether the ASR encoder is frozen or not during the second stage of training. It is shown in [16] that the unfrozen encoder is superior in the recognition accuracy. In our preliminary experiments, we tried these two training strategies mentioned above and extracted fixed-length embeddings from some cross channel test data. Visualizations of these embeddings are shown in Fig. 2, with each color representing one language. While the ideal language identification model should ignore channel differences, embeddings in the same color should be gathered into one cluster, even if they are from different channels. From Fig. 2a, it can be observed that the model trained using a frozen encoder is relatively compact within each category, and the classification boundaries are relatively smooth. The unfrozen encoder strategy, as shown in Fig. 2b, exhibits multiple sub-centers within each category, and the distance between cross-channel data is far,

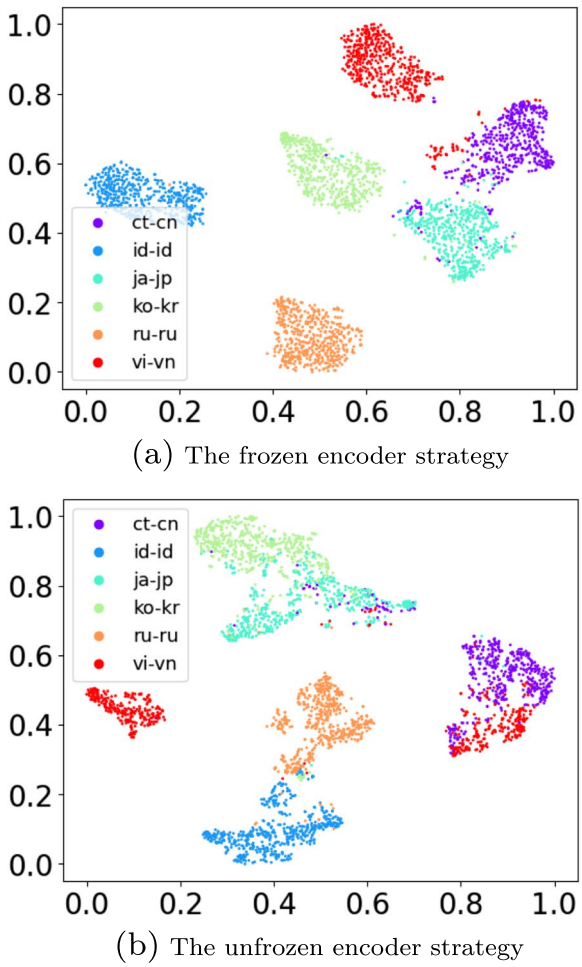

Fig. 2 The visualization results of language space trained with a frozen encoder strategy (**a**) and an unfrozen encoder (**b**) strategy, respectively

indicating a poor generalization performance. However, the method with the poor generalization performance achieved a higher accuracy, which reflects the trade-off between the accuracy and robustness.

Moreover, the methods listed in Table 1 utilize the traditional phonetic features [33]. There is a long-standing hypothesis that phonetic features may be sufficient for LID tasks [9], so that the ASR branch is designed to extract phonetic features and the LID branch is designed to decode these phonetic features, whereas other kinds of features are generally ignored. However, it cannot explain the trade-off mentioned above and why a higher ASR accuracy does not yield a better LID performance [34, 35]. Therefore, in this section, we analyze the reason why the two-stage training strategy can hardly improve both the recognition accuracy and the generalization ability.

Figure 3 lists three types of features fed into LID under different training strategies. When we directly use acoustic features to train an end-to-end LID model, the extracted information can be separated into three parts, as shown in Fig. 3a. Firstly, according to whether the information can be extracted by ASR, two parts separated by a vertical line are phonetic information and other high-level features. Then, these high-level features can further be separated into language discriminative features and noise, by a horizontal line. Specifically, language discriminative features include prosodies and some special phone sequences [5], which can hardly be extracted by an end-to-end ASR task. Different kinds of noise include not only background noise but also the bias of the dataset, such as an unbalanced speaker distribution. While such kind of noise may be useful in a specific dataset, the generalization ability of the LID model may be affected, too.

If the encoder is frozen in the second stage of the LID training, the features fed into LID are still phonetic features, just the same as those after the first stage of the

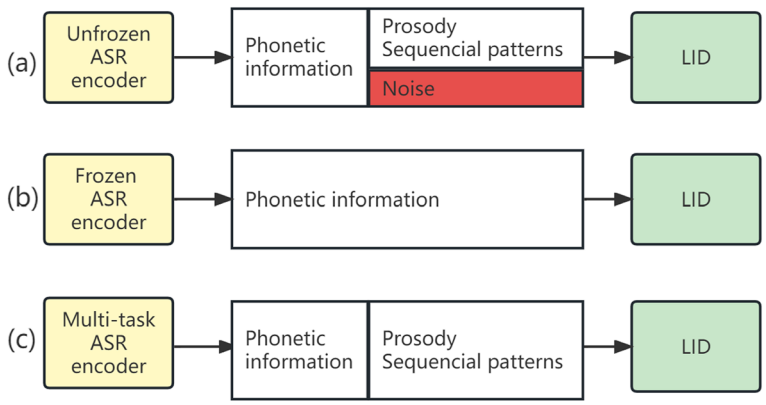

Fig. 3 Three types of features fed into LID. **a** End-to-end LID tends to learn hybrid-level knowledge, which includes noise. **b** A pretrained frozen ASR encoder can only provide phonetic features for LID. **c** Ideally, a pretrained unfrozen ASR encoder should provide noiseless hybrid-level features

ASR training, as shown in Fig. 3b. Because the encoder is frozen, the LID model can only make decisions based on the extracted phonetic features rather than noise, thus ensuring a good generalization ability.

If we train both the encoder and LID in the second stage, as shown in Fig. 3c, the encoder is ideally allowed to extract hybrid-level information suitable for LID but not discriminative for ASR, which explains the superior of the unfrozen encoder. Nonetheless, considering that stage is actually an end-to-end LID training, the encoder will forget the phonetic information and noise will still be learned by the encoder after many steps of training. In this case, overfitting may be caused by the noise, which is discovered by our visualization analysis shown in Fig. 6, that is, although the unfrozen encoder may have better results, the embeddings extracted from the frozen encoder models have smoother classification boundaries.

Therefore, there is a trade-off between the recognition accuracy and the generalization ability for the traditional two-stage training strategy. The features extracted by the shared encoder are either only phonetic information or hybrid-level information with noise, so that they are either too sparse to get a good recognition accuracy or too flexible to get a good generalization ability. Hence, our goal is to improve the recognition accuracy of LID and keep the generalization ability at the same time.

3.2 The proposed three-stage training strategy

In this paper, we propose a three-stage training strategy by adding a multi-task learning stage between the first ASR stage and the final LID stage, as shown in Fig. 4, so that the high-level features suitable for LID can be embedded into the encoder before freezing. Meanwhile, the phonetic features will not be forgotten and noise cannot be learned in the final LID stage after freezing. With the proposed three-stage strategy using the frozen encoder, the final LID model can have both the good performance of the unfrozen encoder and the generalization ability of the frozen encoder.

Firstly, the conformer-based ASR model is trained using a hybrid CTC/attention loss function. Then, the whole end-to-end model is trained in a multi-task manner. The final loss is composed of the LID loss and the ASR loss with an empirical control factor α:

$$\mathcal{L} = \mathcal{L}_{asr} + \alpha \mathcal{L}_{lid}, \tag{1}$$

where $\mathcal{L}_{asr}$ is defined in Eq. 4 in Section 4 and $\mathcal{L}_{lid}$ is the classical softmax cross-entropy loss. Finally, the third stage fine-tunes the model with $\mathcal{L}_{lid}$ only. The shared parameters of two adjacent stages are initialized by the values of the parameters of the previous stage using transfer learning.

3.3 Orthogonality regularization

Orthogonality regularization is used in our work to improve the LID performance. Specifically, it constrains the basis vectors to help model the language space and can be used in both the traditional two-stage strategy and our proposed three-stage strategy.

Empirically, compared with the unfrozen encoder, the frozen encoder has a better generalization ability but is poor at the classification ability. We design an embedding space with a set of orthogonal basis vectors to facilitate

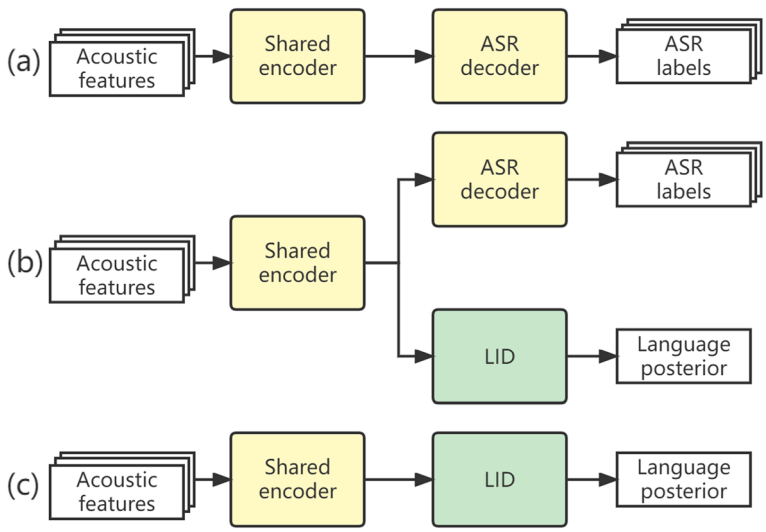

Fig. 4 The proposed three-stage training strategy. **a** The first end-to-end ASR training stage. **b** The second multi-task training stage. **c** The final end-to-end LID training stage

the frozen encoder to get a better classification ability without reducing its generalization ability.

The supervised training of the end-to-end LID method is a progress of finding the most language discriminative embedding space, from a geometric point of view. The centers of languages form a set of basis vectors for the language space. After the embeddings are extracted by the LID model from input utterances, the projection of one embedding onto each basis vector corresponds to the similarity between the utterance and the center of the corresponding language.

The classification loss function used in this paper, that is, softmax cross-entropy loss, is presented as follows:

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}, \tag{2}$$

where $x_i \in \mathbb{R}^d$ denotes the embedding feature of the i-th sample, belonging to the y_i-th language, and n means the number of languages. N represents the batch-size. Formula $e^{W_j^T x_i + b_j}$ is the projection mentioned above, and $W \in \mathbb{R}^{n \times d}$ is n language centers or n basis vectors. The further the distance between two basis vectors is, the larger the between-class variance of their corresponding languages is, and the more separable these two languages are. Therefore, we speculate the basis vectors are orthogonal to each other, which means the value of $W^T W - I$ should be as less as possible. However, in our preliminary experiments, there is not always a reduction of that term, especially when the shared encoder is frozen, so we add orthogonality regularization into the LID loss:

$$\mathcal{L}_{lid} = \mathcal{L}_{CE} + \lambda \sigma_{max}(WW^T - I), \tag{3}$$

where λ is a hyper-parameter, and $\sigma_{max}(\cdot)$ denotes the spectral norm of a matrix.

4 The model architecture

This section presents the details of the phonetic LID model we design, including a shared encoder and two decoders for LID and ASR respectively. The shared encoder can be implemented by various structures, and in this paper, we use the conformer [36], which integrates components from CNNs and transformers, and can efficiently capture both local and global correlations. With the hybrid CTC/attention ASR decoder [37, 38], the shared encoder has strong capabilities to discriminate languages and therefore can improve the performance [26].

For the LID decoder, we do not choose complex structures such as TDNN [4] or LSTM [39]. We implement it in a simple but acceptable manner to present the relationship between LID and ASR more clearly. Frame-level phonetic features extracted by the shared encoder are directly aggregated by a statistic pooling layer without any kind of transform. The aggregated embedding vector has a fixed length and contains linguistic information extracted from phonetic features. Then, two fully connected layers are used to make the final decision. The first fully connected layer maps the fix-length embeddings to a space with the best language discrimination, while the second fully connected layer projects the transformed embeddings to the centers of 10 languages, respectively.

As shown in Fig. 5, the architecture for the phonetic feature extraction used in this paper is a hybrid CTC/attention one [38], which consists of three components: a shared encoder, an attention-based decoder, and a CTC decoder. This model can solve the problem caused by the too flexible alignment property of the attention-based method with CTC through a regularization during training and a score correction during decoding [40]. The encoder receives acoustic features of T frames, that is, $X = \{x_t \in \mathbb{R}^{D_1} | t = 1, \ldots, T\}$, and extracts the phonetic features within them, that is, $H = \{h_t \in \mathbb{R}^{D_2} | t = 1, \ldots, T\}$. The following two decoders are used to deal with the T-length phonetic features to an N-length word sequence, that is, $W = \{w_n \in \mathcal{V} | n = 1, \ldots, N\}$. In X, H, and W, x_t is a D_1 -dimensional feature vector (e.g., Mel filterbanks) at frame t, and h_t is a D_2-dimensional feature vector containing phonetic information, and w_n is a word or subword at position n in Vocabulary $\mathcal{V}$.

The loss functions are expressed as:

$$\mathcal{L}_{asr} = (\alpha)\mathcal{L}_{ctc} + (1-\alpha)\mathcal{L}_{att}, \tag{4}$$

$$\mathcal{L}_{ctc} = -logp_{ctc}(W|X), \tag{5}$$

$$\mathcal{L}_{att} = -logp_{att}(W|X), \tag{6}$$

where hyper-parameter α represents the weight of the CTC loss, and p_{ctc} can be computed using dynamic programming [41], and p_{att} is the output of the attention-based decoder. It is shown that the introduction of the CTC module helps to ensure appropriate alignments and fast converge [42].

The building of Vocabulary $\mathcal{V}$ is a tricky problem, especially for multilingual ASR tasks. Considering the out-of-vocabulary problem, the vocabulary should be large enough and its units are typically at character level [22, 38], while the sub-word level units like byte-pair encoding used in [28] can also achieve an acceptable performance. There are 10 languages in our experiments, and we use a byte-pair encoding vocabulary shared over all languages, which is

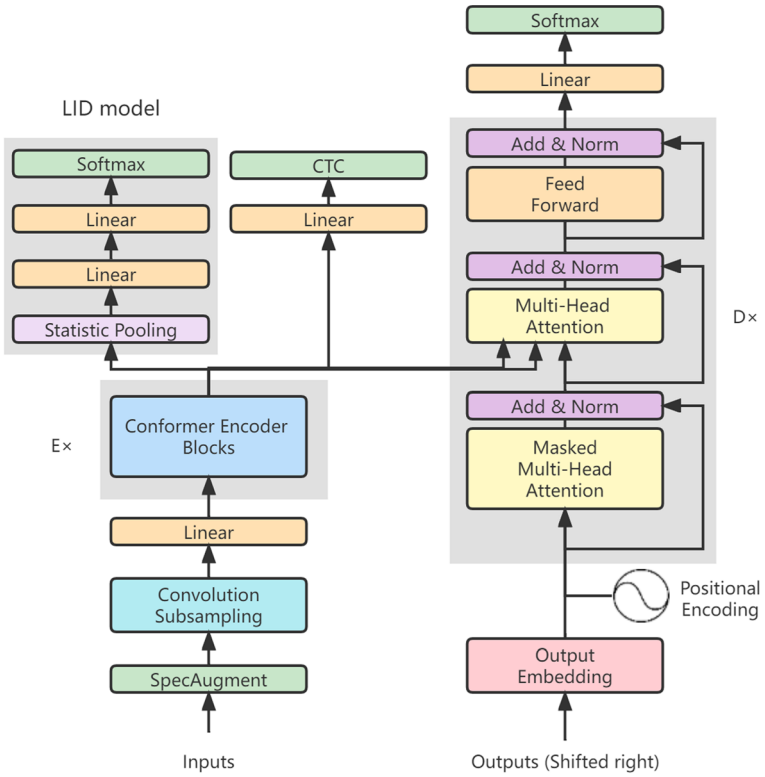

Fig. 5 The architecture of the end-to-end multi-task model

composed of sub-vocabularies, with each corresponding to a language. There are 1200 to 3500 units in each sub-vocabulary.

5 Experiments

5.1 The OLR dataset

We train and evaluate the model on the OLR dataset [43–46]. Table 2 details the contents of the dataset. *ASR-train* refers to the subset used for training the ASR system, composed of AP16-OL7 and AP17-OL3. Other non-transcript subsets in the OLR dataset, including AP17-OLR-test, AP18-OLR-test, and AP19-OLR-test, are added to *ASR-train* and used for LID training, and all of them are collectively named *LID-train* in our experiments. In *ASR-train*, there are 10 languages, that is, Cantonese, Mandarin, Indonesian, Japanese, Russian, Korean, Vietnamese, Kazakh, Tibetan, and Uyghur, consisting of about 71.42 h of speech signals recorded by mobile phones. *channel-test* and *noisy-test* are two standard test sets for OLR20 challenge [43], used for the validation of different recording equipments and environments, and for the validation of noisy environments (low SNR), respectively.

5.2 Training setup

We conduct all experiments on the WeNet platform [47]. Eighty-dimensional FBank is extracted from every speech frame with a 25-ms window and a stride of 10 ms. Data

Table 2 Details of the OLR dataset

Subset	Data composition [43]	Number of utterances	Duration (h)	Channel
ASR-train	AP16-OL7, AP17-OL3	50,071	71.42	Mobile
LID-train	*ASR-train*, AP17-OLR-test, AP18-OLR-test, AP19-OLR-test	176,354	205.25	Mobile
dev	subset of ASR-train	2992	4.53	Mobile
channel-test	AP20-OLR-channel-test	11,848	17.84	Cross-channel
noisy-test	AP20-OLR-noisy-test	9496	13.1	Mobile

augmentation methods, including speed and volume perturbation and specAugment [48], are used before training. For speed perturbation, we apply speed factors of 0.9 and 1.1 respectively to slow down or speed up the original recording. For volume perturbation, the random volume factor is applied. For specAument, a time wrap with the max width of 80, 2 time masks with the max width of 50, and 2 frequency masks with the max width of 10 are applied. No voice activity detection (VAD) is used. The cepstral mean and variance normalization [49] is applied to the input features.

We use the conformer as the ASR encoder and the transformer as the ASR decoder. The parameters are listed in Table 3.

For the LID model, we use a statistic pooling followed by two fully connected layers with ReLu activation and BatchNorm per layer. The dimensions of the input and output of the first fully connected layer are both 256, corresponding to the output dimension of the conformer. For the second fully connected layer, its input dimension is 256, and its output dimension is 10. The dropout rate is set to 0.5 before the cross-entropy loss.

Both ASR and LID models are trained with the ADAM optimizer. ASR models are trained with the warm-up schedule [50] of 25,000 warm-up steps, and the peak learning rate is 1e−2. For the LID model, a relatively low learning rate of 1e−5 is used. We train the models for 240, 80, and 40 epochs for the first, second, and third stages, respectively. 240 is the default number of epochs for training an ASR model in the wenet framework [47], and we also use this number because the model converges well after 240 epochs in our experiments. Eighty and 40 are our empirical values set according to the convergence rate of the training set after the preliminary experiments, which ensures that all systems in this paper can converge under these parameters. We use a dynamic batch size to suit our NVIDIA 1080Ti with 11GB memory. Model average is computed by averaging the 10 best models according to the loss on the dev set, and based on it, the methods in our experiments are fairly compared.

Table 3 Parameters of the conformer ASR model

Conformer encoder	
Number of blocks	12
Linear dimensionality	2048
Output size	256
Number of attention heads	4
Dropout rate	0.1
Type of activation	Swish
Type of the positional encoding layer	Relative
Transformer decoder	
Linear dimensionality	2048
Number of blocks	6
Number of attention heads	4
ASR Training	
CTC weight	0.3
Label smoothing	0.1

5.3 Evaluation metrics

As in the OLR20 challenge [51], we use two metrics to evaluate the language recognition systems: *Cavg* as the principal evaluation metric, and the Equal Error Rate (EER) as the secondary evaluation metric. Considering that language recognition tasks need to keep both the false alarm rate and the false rejection rate as low as possible, *Cavg* and EER comprehensively measure these two rates. Among them, EER does not consider the performance of individual languages, and the scores of all samples are mixed for calculation. However, since the optimal threshold for each language may not be the same for the best metric, *Cavg* calculates the error rate separately for each language and then averages them, resulting in a more precise evaluation.

5.3.1 Cavg

In most language recognition challenges, such as LRE [52] and OLR [43], *Cavg* is chosen as the principle evaluation metric. The pair-wise loss of a specific target/non-target language pair is defined as:

$$\begin{aligned} C(L_t, L_n) =& P_{\text{Target}} P_{\text{Miss}}(L_t) + \\ & \left(1 - P_{\text{Target}}\right) P_{FA}(L_t, L_n), \end{aligned} \tag{7}$$

where L_t and L_n are the target and non-target languages, respectively; P_{Miss} and P_{FA} are the missing and false alarm probabilities, respectively. P_{target} is the prior probability for the target language, which is set to 0.5 in the evaluation.

Cavg is defined as follows:

$$\begin{aligned} C_{avg} =& \frac{1}{N} \Bigg\{ P_{\text{Target}} \cdot \sum_{L_t} P_{\text{Miss}}(L_t) + \\ & \sum_{L_t} \sum_{L_n} P_{\text{Nontarget}} P_{FA}(L_t, L_n) \Bigg\}, \end{aligned} \tag{8}$$

where N is the number of languages and $P_{Nontarget} = (1 - P_{Target})/(N - 1)$.

5.3.2 Equal error rate

EER is widely used in many recognition tasks [24]. In order to define EER, false rejected ratio (FRR) and false

accepted ratio (FAR) need to be considered first, written as:

$$P_{FRR} = \frac{\text{Number of target trials rejected}}{\text{Number of Total target trials}}, \quad (9)$$

$$P_{FAR} = \frac{\text{Number of nontarget trials accepted}}{\text{Number of Total nontarget trials}}, \quad (10)$$

If we use P_{FRR} as the vertical axis and P_{FAR} as the horizontal axis, a continuous curve of FAR corresponding to FRR is obtained, named Detect Error Trade-off (DET).

EER is defined as the value in the DET curve, where $P_{FRR} = P_{FAR}$.

5.4 Experimental results under different training strategies

The experimental results under different training strategies and our proposed orthogonality regularization are listed in Table 4.

We first implement the traditional two-stage strategy with two different encoders: the frozen encoder and the unfrozen encoder, as reported in No. 0 and No. 1 in Table 4. We get the same conclusion as [16], that is, fine-tuning the LID task with the unfrozen encoder outperforms that with the frozen encoder. Note that although No. 1 has better results on *Cavg*, we believe that it may come from overfitting to some languages, and overall, EER is increased.

Then, the orthogonality regularization with $\lambda = 0.1$ is added to the above two experiments, reported as No. 2 and No. 3. From Table 4, we see that our orthogonality regularization brings improvements to the frozen encoder method, while the results of the unfrozen method decrease slightly. The reason may be that the frozen encoder cannot get enough information and our regularization can be deemed as extra prior knowledge, while the unfrozen method already has the flexibility to explore any kinds of knowledge they need, so it cannot benefit from the regularization.

In our proposed three-stage training strategy, a multi-task learning stage is added between the traditional two stages. It can also be trained with the frozen encoder or the unfrozen encoder, as reported in No. 4 and No. 5 respectively in Table 4. Compared with their corresponding two-stage experiments, that is, No. 0 and No. 1, the proposed three-stage strategy obtains obvious better performances, which implies the additional knowledge is beneficial. However, there is one exception. By comparing No. 0 and No. 4 on the *noisy-test* set, we see that the two-stage strategy "asr-lid-F" performs better than the three-stage-stategy "asr-mt-lid-F". The reason may be that noise is not completely filtered by the ASR task, so they can be involved in the end-to-end LID training during the multi-task learning stage and then are fixed by the frozen encoder and finally affect the performance.

The experimental results of the three-stage training strategy with orthogonality regularization are reported in No. 6 and No. 7. It can be observed that the trends are similar to those in the above two-stage training strategies, that is, orthogonality regularization dramatically improves the performance of the frozen method, but it seems useless in the unfrozen method. Moreover, compared with the two-stage training methods, that is, No. 2 and No. 3, the proposed three-stage training methods achieve better performances on both *channel-test* and *noisy-test*.

The experimental results of the two-stage strategy involving multitask learning following LID fine-tuning, that is, mt-lid, are reported in No. 8 and No. 9. Since its multitask learning stage tends to learn hybrid-level and noiseless features, it should theoretically achieve results

Table 4 Experimental results under different training strategies

ID	Strategy	λ	channel-test		noisy-test	
			Cavg%	EER%	*Cavg*%	EER%
0	asr-lid-F	0	8.37	14.83	2.73	5.96
1	asr-lid	0	7.64	18.24	2.13	6.32
2	asr-lid-F	0.1	5.41	14.15	1.92	9.39
3	asr-lid	0.1	7.93	19.64	2.22	6.52
4	asr-mt-lid-F	0	7.25	12.76	4.93	6.19
5	asr-mt-lid	0	5.10	9.58	2.01	5.34
6	asr-mt-lid-F	0.1	4.48	8.18	1.78	3.34
7	asr-mt-lid	0.1	5.30	10.66	2.02	5.74
8	mt-lid-F	0	8.14	15.78	2.67	6.15
9	mt-lid	0	7.92	15.44	2.52	5.74

similar to the three-stage strategy. However, its results are worse, which verify the effectiveness of the three-stage strategy.

5.5 Hyperparameter analysis

To find the optimal strategy of the proposed methods, the hyperparameter λ in Section 3.3 is experimentally and theoretically analyzed.

As described in Section 3.3, λ is the weight of orthogonality regularization in $\mathcal{L}_{lid}$. We hope that the embedding space is a space formed by a set of orthogonal basis vectors, so that the correlation between any two language centers is as small as possible. However, considering the natural similarities between languages, completely orthogonal basis vectors may not directly lead to better classification results. Therefore, we use λ to control the strength of the constraint.

To explore the optimal value of λ, we set it to be 0, 0.01, 0.1, and 1.0, respectively. As shown in Table 5, No. 0 to No. 3 denote the three-stage strategy with the frozen encoder in the final stage. The results are as expected, that is, setting λ to be 0 or 1 cannot bring the best results, and setting λ to be 0.1 is optimal. Also, using orthogonality regularization ($\lambda \neq 0$) is better than without it ($\lambda = 0$), showing the advantages of the introduction of orthogonality regularization.

No. 4 to No. 7 denote the three-stage strategy with the unfrozen encoder in the final stage. The best results on the two test sets appear in No.7 and No.5, respectively. Still, consistent with the results in Table 4, for the unfrozen encoder, the strategies with orthogonality regularization (No. 5 to No. 7) do not show better performances than the strategy without orthogonality regularization (No. 4), as the unfrozen encoder already gives models enough flexibility to learn the optimal embedding space, so orthogonality regularization cannot bring better results. However, it is not worse either. For example, when $\lambda = 1$ (No. 7), the best results on *channel-test* are achieved. Observing that the results of No. 7 on the *noisy-test* set decrease, we suspect that the prior knowledge introduced by the orthogonal constraint still has some impact on modeling the language space, which is worth further analysis in future work.

5.6 Visualizing analysis

As mentioned in Section 3.1, when the pre-trained encoder is unfrozen in the second stage, it may achieve a higher performance but a poor generalization ability. We compare the models' generalization abilities through visualizing analysis of the distributions of language embeddings. The models yielding smoother classification boundaries are considered to have better generalization abilities. Language embeddings are extracted from the penultimate layer of the corresponding LID model and plotted after dimension reduction. T-Stochastic Neighbor Embedding (t-SNE) is used as the non-linear dimension reduction algorithm.

We take the *channel-test* test set as an example and extract language embeddings from the strategies listed in Table 4, as shown in Fig. 6. Each color represents a different language. To make it clear, we slightly adjust the order of the strategies and put those using the frozen encoder together. The two-stage strategies and the three-stage strategies are divided into two lines.

Generally speaking, the classification boundaries of the proposed three-stage strategies (e–h) in Fig. 6 are smoother and clearer than those of the traditional two-stage strategies (a–d). The strategies trained with the frozen encoder (c, d, g, and f) show better generalization abilities, and the strategies trained with orthogonality regularization (b, d, f, and h) have higher between-class variances and lower within-class variances.

Specifically, a and c in Fig. 6 compare the difference in the generalization ability with freezing the encoder or not in the traditional two-stage strategy. Although the former is more accurate, its uneven classification boundary

Table 5 Experimental results with different values of the hyperparameter λ

ID	Strategy	λ	channel-test		noisy-test	
			Cavg%	EER%	*Cavg*%	EER%
0	asr-mt-lid-F	0	7.25	12.76	4.93	6.19
1	asr-mt-lid-F	0.01	5.06	9.23	2.10	3.17
2	asr-mt-lid-F	0.1	4.48	8.18	1.78	3.34
3	asr-mt-lid-F	1	5.10	9.43	2.48	5.07
4	asr-mt-lid	0	5.10	9.58	2.00	5.34
5	asr-mt-lid	0.01	5.50	10.23	1.99	5.67
6	asr-mt-lid	0.1	5.30	10.66	2.02	5.74
7	asr-mt-lid	1	4.03	8.12	2.27	5.68

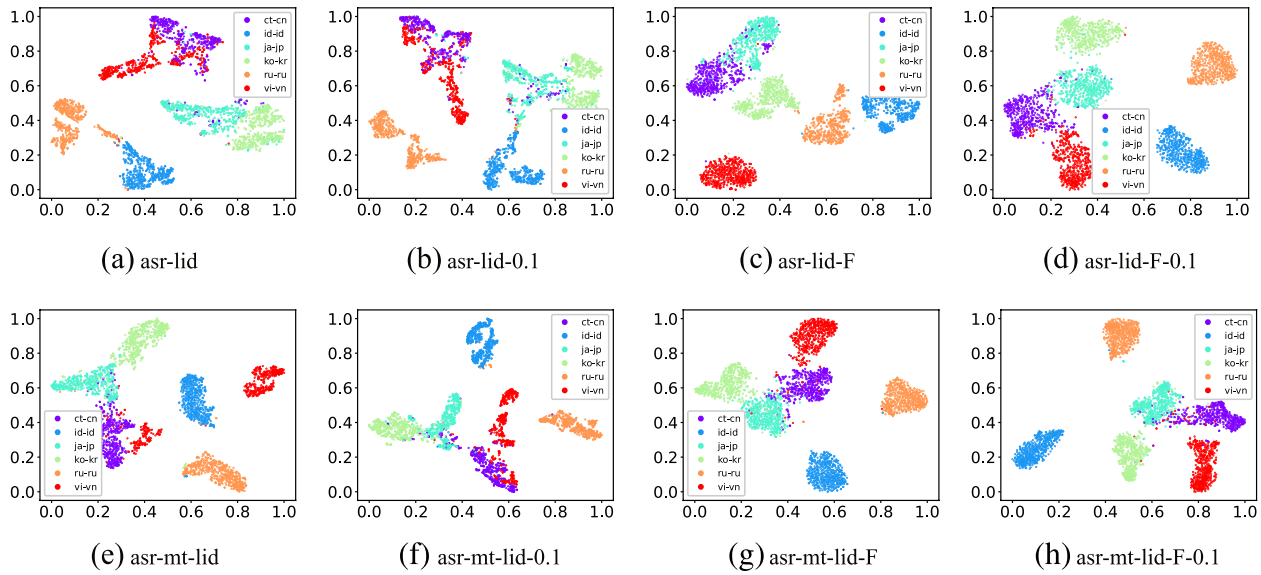

Fig. 6 Language embeddings of different training strategies, plotted by t-SNE on *channel-test*. Each color represents a different language. Models are named by their strategies plus the value of λ and correspond to those in Table 4

brings about a poor generalization ability. The strategies trained with orthogonality regularization reduce the within-class variance, which is quite obvious for Vietnamese (vi-vn) in e and f. Whether or not orthogonality regularization is used, the three-stage strategies show better generalization abilities than the two-stage strategies, especially when using the unfrozen encoder.

5.7 Comparison with state-of-the-arts

Usually, back-end processing can improve the model's performance and hence is widely used in previous OLR challenges [51]. However, back-end processing is computationally complex because it needs to extract the embeddings of the entire training set. Therefore, in our experiments, we only apply back-end processing to our best two-stage and three-stage models to compare with other state-of-the-art models which also do back-end processing.

In our back-end processing, after training, the output of the penultimate layer of the LID model is extracted as language embeddings. Linear discriminant analysis (LDA) is applied using embeddings of the training set and the embeddings are projected to 100 dimensions. After LDA projection and centering, Logistic Regression (LR) is used to generate the scores.

Table 6 gives the comparison of our best two-stage and three-stage models with the top three models reported on the OLR20 Challenge [51]. From the comparison, we see that both the best two-stage and three-stage models are comparable with the top three models on *noisy-test*. It is worth mentioning that our best three-stage model with orthogonality regularization shows obvious advantages on both *channel-test* and *noisy-test*, showing the effectiveness of our proposed three-stage training and orthogonality regularization methods.

6 Conclusions

In this paper, we extended the input features of language recognition from phonetic features to hybrid-level features, and the traditional two-stage training

Table 6 The comparison of our best model with the best two-stage model and the top three models on the OLR20 challenge

ID	Model	λ	channel-test		noisy-test	
			Cavg%	EER%	*Cavg*%	EER%
0	Top 1 on the OLR20 Challenge [51]	-	2.39	2.47	3.47	4.07
1	Top 2 on the OLR20 Challenge [51]	-	4.21	4.51	4.76	4.87
2	Top 3 on the OLR20 Challenge [51]	-	4.77	4.82	5.38	5.60
3	The best two-stage model	0.1	3.90	7.25	2.06	4.20
4	The best three-stage model (ours)	0.1	1.88	3.57	1.86	3.74

strategy was analyzed. Preliminary experiments showed that the traditional two-stage strategy was not enough to ensure both the recognition accuracy and the generalization ability. Therefore, we proposed two methods to solve this problem: the three-stage training strategy and the orthogonality regularization. A multi-task learning stage was added between the traditional two stages, which can extract hybrid-level but noiseless features for the following LID task. Then, we combined orthogonality regularization with the three-stage training strategy to get our end-to-end multi-task architecture. The experimental results demonstrated that our proposed model achieved significant performance improvements, compared with the baseline two-stage model and the top three models on the OLR20 Challenge.

Abbreviations

LID	Language identification
ASR	Automatic speech recognition
OLR	Oriental language recognition
T-SNE	T-Stochastic neighbor embedding
LDA	Linear discriminant analysis (LDA)
LR	Logistic regression
VAD	Voice activity detection
EER	Equal error rate
CTC	Connectionist Temporal Classification

Acknowledgements
Not applicable.

Authors' contributions
The first author mainly performed the experiments and wrote the paper, and the other authors reviewed and edited the manuscript. All of the authors discussed the final results. All of the authors read and approved the final manuscript.

Authors' information
Not applicable.

Funding
This work was supported by the Fundamental Research Funds for the Central Universities (grant number 2021ZY87).

Availability of data and materials
All data supporting the conclusions of this article are included in the OLR 2020 Challenge [43–46], http://index.cslt.org/mediawiki/index.php/OLR_Challenge_2020.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 14 October 2022 Accepted: 16 March 2023
Published online: 06 April 2023

References

1. E. Ambikairajah, H. Li, L. Wang, B. Yin, V. Sethu, Language identification: a tutorial. Circ. Syst. Mag. IEEE. **11**(2), 82–108 (2011)
2. A. Waibel, P. Geutner, L.M. Tomokiyo, T. Schultz, M. Woszczyna, Multilinguality in speech and spoken language systems. Proc. IEEE. **88**(8), 1297–1313 (2000). https://doi.org/10.1109/5.880085
3. S. Punjabi, H. Arsikere, Z. Raeesy, C. Chandak, N. Bhave, A. Bansal, M. Müller, S. Murillo, A. Rastrow, S. Garimella, R. Maas, M. Hans, A. Mouchtaris, S. Kunzmann, Streaming end-to-end bilingual ASR systems with joint language identification. CoRR. **abs/2007.03900** (2020). arXiv preprint arXiv:2007.03900
4. D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, S. Khudanpur, in *Odyssey 2018: The Speaker and Language Recognition Workshop, 26-29 June 2018, Les Sables d'Olonne, France*, ed. by A. Larcher, J. Bonastre. Spoken language recognition using x-vectors (ISCA, 2018), pp. 105–111. https://doi.org/10.21437/Odyssey.2018-15
5. H. Li, B. Ma, K. Lee, Spoken language recognition: From fundamentals to practice. Proc. IEEE. **101**(5), 1136–1159 (2013). https://doi.org/10.1109/JPROC.2012.2237151
6. R. Tong, B. Ma, D. Zhu, H. Li, E. Chng, in *2006 IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2006, Toulouse, France, May 14-19, 2006*, Integrating acoustic, prosodic and phonotactic features for spoken language identification (IEEE, 2006), pp. 205–208. https://doi.org/10.1109/ICASSP.2006.1659993
7. D.M. González, O. Plchot, L. Burget, O. Glembek, P. Matejka, in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*, Language recognition in ivectors space (ISCA, 2011), pp. 861–864. http://www.isca-speech.org/archive/interspeech_2011/i11_0861.html
8. W. Cai, J. Chen, J. Zhang, M. Li, On-the-fly data loader and utterance-level aggregation for speaker and language recognition. IEEE ACM Trans. Audio Speech Lang. Process. **28**, 1038–1051 (2020). https://doi.org/10.1109/TASLP.2020.2980991
9. Z. Tang, D. Wang, Y. Chen, L. Li, A. Abel, Phonetic temporal neural model for language identification. IEEE ACM Trans. Audio Speech Lang. Process. **26**(1), 134–144 (2018). https://doi.org/10.1109/TASLP.2017.2764271
10. Q. Zhang, H. Boril, J.H.L. Hansen, in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, Supervector pre-processing for prsvm-based chinese and arabic dialect identification (IEEE, 2013), pp. 7363–7367. https://doi.org/10.1109/ICASSP.2013.6639093
11. S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, Y. Bengio, in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, ed. by G. Kubin, Z. Kacic. Learning problem-agnostic speech representations from multiple self-supervised tasks (ISCA, 2019), pp. 161–165. https://doi.org/10.21437/Interspeech.2019-2605
12. M. Jin, Y. Song, I.V. McLoughlin, W. Guo, L. Dai, in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, ed. by F. Lacerda. End-to-end language identification using high-order utterance representation with bilinear pooling (ISCA, 2017), pp. 2571–2575. http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0044.html
13. R. Duroselle, D. Jouvet, I. Illina, in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, ed. by H. Meng, B. Xu, T.F. Zheng. Metric learning loss functions to reduce domain mismatch in the x-vector space for language recognition (ISCA, 2020), pp. 447–451. https://doi.org/10.21437/Interspeech.2020-1708
14. D. Romero, L.F. D'Haro, C. Salamea, in *Fifth International Conference, IberSPEECH 2021, Valladolid, Spain, 24-25 March 2021, Proceedings*, ed. by V. Cardeñoso-Payo, D.E. Mancebo, C.G. Ferreras. Exploring transformer-based language recognition using phonotactic information (ISCA, 2021). http://www.isca-speech.org/archive/IberSPEECH_2021/pdfs/53.pdf
15. R. Fér, P. Matějka, F. Grézl, O. Plchot, K. Veselý, J. Černocký, Multilingually trained bottleneck features in spoken language recognition. Comput. Speech Lang. **46**(nov.), 252–267 (2017)
16. R. Duroselle, M. Sahidullah, D. Jouvet, I. Illina, in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, ed. by H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, P. Motlícek. Modeling

and training strategies for language recognition systems (ISCA, 2021), pp. 1494–1498. https://doi.org/10.21437/Interspeech.2021-277
17. P. Matejka, L. Zhang, T. Ng, O. Glembek, J.Z. Ma, B. Zhang, S.H. Mallidi, in *Odyssey 2014: The Speaker and Language Recognition Workshop, Joensuu, Finland, June 16-19, 2014*, Neural network bottleneck features for language identification (ISCA, 2014). https://isca-speech.org/archive/odyssey_2014/abstracts.html#abs35
18. R. Fér, P. Matejka, F. Grézl, O. Plchot, J. Cernocký, in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, Multilingual bottleneck features for language recognition (ISCA, 2015), pp. 389–393. http://www.isca-speech.org/archive/interspeech_2015/i15_0389.html
19. R. Fér, P. Matejka, F. Grézl, O. Plchot, K. Veselý, J.H. Cernocký, Multilingually trained bottleneck features in spoken language recognition. Comput. Speech Lang. **46**, 252–267 (2017). https://doi.org/10.1016/j.csl.2017.06.008
20. B. Li, R. Pang, T.N. Sainath, A. Gulati, Y. Zhang, J. Qin, P. Haghani, W.R. Huang, M. Ma, J. Bai, in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021, Cartagena, Colombia, December 13-17, 2021*, Scaling end-to-end models for large-scale multilingual ASR (IEEE, 2021), pp. 1011–1018. https://doi.org/10.1109/ASRU51503.2021.9687871
21. N. Pham, T. Nguyen, J. Niehues, M. Müller, A. Waibel, in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, ed. by G. Kubin, Z. Kacic. Very deep self-attention networks for end-to-end speech recognition (ISCA, 2019), pp. 66–70. https://doi.org/10.21437/Interspeech.2019-2702
22. S. Watanabe, T. Hori, J.R. Hershey, in *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017*, Language independent end-to-end architecture for joint language identification and speech recognition (IEEE, 2017), pp. 265–271. https://doi.org/10.1109/ASRU.2017.8268945
23. M. Zhao, R. Li, S. Yan, Z. Li, H. Lu, S. Xia, Q. Hong, L. Li, in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2019, Lanzhou, China, November 18-21, 2019*, Phone-aware multi-task learning and length expanding for short-duration language recognition (IEEE, 2019), pp. 433–437. https://doi.org/10.1109/APSIPAASC47483.2019.9023014
24. L. Li, Z. Li, Y. Liu, Q. Hong, Deep joint learning for language recognition. Neural Netw. **141**, 72–86 (2021). https://doi.org/10.1016/j.neunet.2021.03.026
25. Z. Ren, G. Yang, S. Xu, in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, ed. by G. Kubin, Z. Kacic. Two-stage training for chinese dialect recognition (ISCA, 2019), pp. 4050–4054. https://doi.org/10.21437/Interspeech.2019-1522
26. D. Wang, S. Ye, X. Hu, S. Li, X. Xu, in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, ed. by H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, P. Motlícek. An end-to-end dialect identification system with transfer learning from a multilingual automatic speech recognition model (ISCA, 2021), pp. 3266–3270. https://doi.org/10.21437/Interspeech.2021-374
27. R. Duroselle, M. Sahidullah, D. Jouvet, I. Illina, in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, ed. by H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, P. Motlícek. Language recognition on unknown conditions: The loria-inria-multispeech system for AP20-OLR challenge (ISCA, 2021), pp. 3256–3260. https://doi.org/10.21437/Interspeech.2021-276
28. T. Alumäe, K. Kukk, Pretraining approaches for spoken language recognition: Taltech submission to the OLR 2021 challenge. CoRR **abs/2205.07083** (2022). arXiv preprint arXiv:2205.07083
29. B.M. Abdullah, T. Avgustinova, B. Möbius, D. Klakow, in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, ed. by H. Meng, B. Xu, T.F. Zheng. Cross-domain adaptation of spoken language identification for related languages: The curious case of slavic languages (ISCA, 2020), pp. 477–481. https://doi.org/10.21437/Interspeech.2020-2930
30. M. McLaren, D. Castán, L. Ferrer, in *Odyssey 2016: The Speaker and Language Recognition Workshop, Bilbao, Spain, June 21-24, 2016*, ed. by L.J. Rodríguez-Fuentes, E. Lleida. Analyzing the effect of channel mismatch on the SRI language recognition evaluation 2015 system (ISCA, 2016), pp. 188–195. https://doi.org/10.21437/Odyssey.2016-27
31. S. Ling, J. Salazar, Y. Liu, K. Kirchhoff, in *Odyssey 2020: The Speaker and Language Recognition Workshop, 1-5 November 2020, Tokyo, Japan*, ed. by K. Lee, T. Koshinaka, K. Shinoda. Bertphone: Phonetically-aware encoder representations for utterance-level speaker and language recognition (ISCA, 2020), pp. 9–16. https://doi.org/10.21437/Odyssey.2020-2
32. Z. Li, Y. Liu, L. Li, Q. Hong, in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, ed. by H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, P. Motlícek. Additive phoneme-aware margin softmax loss for language recognition (ISCA, 2021), pp. 3276–3280. https://doi.org/10.21437/Interspeech.2021-1167
33. M. McLaren, L. Ferrer, A. Lawson, in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, Exploring the role of phonetic bottleneck features for speaker and language recognition (IEEE, 2016), pp. 5575–5579. https://doi.org/10.1109/ICASSP.2016.7472744
34. O. Adams, M. Wiesner, S. Watanabe, D. Yarowsky, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, ed. by J. Burstein, C. Doran, T. Solorio. Massively multilingual adversarial speech recognition (Association for Computational Linguistics, 2019), pp. 96–108. https://doi.org/10.18653/v1/n19-1009
35. L.D. Alicia, Z. Ruben, D.T. Toledano, G.R. Joaquin, J. Tu, An analysis of the influence of deep neural network (dnn) topology in bottleneck feature based language recognition. PLoS ONE **12**(8), e0182580 (2017)
36. A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, R. Pang, in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, ed. by H. Meng, B. Xu, T.F. Zheng. Conformer: Convolution-augmented transformer for speech recognition (ISCA, 2020), pp. 5036–5040. https://doi.org/10.21437/Interspeech.2020-3015
37. M. Karafiát, M.K. Baskar, S. Watanabe, T. Hori, M. Wiesner, J. Cernocký, in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, ed. by G. Kubin, Z. Kacic. Analysis of multilingual sequence-to-sequence speech recognition systems (ISCA, 2019), pp. 2220–2224. https://doi.org/10.21437/Interspeech.2019-2355
38. S. Watanabe, T. Hori, S. Kim, J.R. Hershey, T. Hayashi, Hybrid ctc/attention architecture for end-to-end speech recognition. IEEE J. Sel. Top. Signal Process. **11**(8), 1240–1253 (2017). https://doi.org/10.1109/JSTSP.2017.2763455
39. G. Gelly, J. Gauvain, V.B. Le, A. Messaoudi, in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, ed. by N. Morgan. A divide-and-conquer approach for language identification based on recurrent neural networks (ISCA, 2016), pp. 3231–3235. https://doi.org/10.21437/Interspeech.2016-180
40. T. Hori, S. Watanabe, J.R. Hershey, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, ed. by R. Barzilay, M. Kan. Joint ctc/attention decoding for end-to-end speech recognition (Association for Computational Linguistics, 2017), pp. 518–529. https://doi.org/10.18653/v1/P17-1048
41. A. Graves, S. Fernández, F.J. Gomez, J. Schmidhuber, in *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, *ACM International Conference Proceeding Series*, vol. 148, ed. by W.W. Cohen, A.W. Moore. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks (ACM, 2006), pp. 369–376. https://doi.org/10.1145/1143844.1143891
42. S. Karita, N.E.Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, T. Nakatani, in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, ed. by G. Kubin, Z. Kacic. Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration (ISCA, 2019), pp. 1408–1412. https://doi.org/10.21437/Interspeech.2019-1938

43. Z. Li, M. Zhao, Q. Hong, L. Li, Z. Tang, D. Wang, L. Song, C. Yang, in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2020, Auckland, New Zealand, December 7-10, 2020*, AP20-OLR challenge: Three tasks and their baselines (IEEE, 2020), pp. 550–555. https://ieeexplore.ieee.org/document/9306442
44. Z. Tang, D. Wang, L. Song, in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2019, Lanzhou, China, November 18-21, 2019*, AP19-OLR challenge: Three tasks and their baselines (IEEE, 2019), pp. 1917–1921. https://doi.org/10.1109/APSIPAASC47483.2019.9023321
45. Z. Tang, D. Wang, Q. Chen, in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2018, Honolulu, HI, USA, November 12-15, 2018*, AP18-OLR challenge: Three tasks and their baselines (IEEE, 2018), pp. 596–600. https://doi.org/10.23919/APSIPA.2018.8659714
46. Z. Tang, D. Wang, Y. Chen, Q. Chen, in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2017, Kuala Lumpur, Malaysia, December 12-15, 2017*, AP17-OLR challenge: Data, plan, and baseline (IEEE, 2017), pp. 749–753. https://doi.org/10.1109/APSIPA.2017.8282134
47. B. Zhang, D. Wu, Z. Peng, X. Song, Z. Yao, H. Lv, L. Xie, C. Yang, F. Pan, J. Niu, Wenet 2.0: More productive end-to-end speech recognition toolkit. CoRR **abs/2203.15455** (2022). arXiv preprint arXiv:2203.15455
48. D.S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E.D. Cubuk, Q.V. Le, in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, ed. by G. Kubin, Z. Kacic. Specaugment: A simple data augmentation method for automatic speech recognition (ISCA, 2019), pp. 2613–2617. https://doi.org/10.21437/Interspeech.2019-2680
49. S. Molau, F. Hilger, H. Ney, in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '03, Hong Kong, April 6-10, 2003*, Feature space normalization in adverse acoustic conditions (IEEE, 2003), pp. 656–659. https://doi.org/10.1109/ICASSP.2003.1198866
50. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, ed. by I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett. Attention is all you need (2017), pp. 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
51. J. Li, B. Wang, Y. Zhi, Z. Li, L. Li, Q. Hong, D. Wang, in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, ed. by H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, P. Motlícek. Oriental language recognition (OLR) 2020: Summary and analysis (ISCA, 2021), pp. 3251–3255. https://doi.org/10.21437/Interspeech.2021-2171
52. A.F. Martin, C.S. Greenberg, J.M. Howard, D. Bansé, G.R. Doddington, J. Hernandez-Cordero, L.P. Mason, in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, NIST language recognition evaluation - plans for 2015 (ISCA, 2015), pp. 3046–3050. http://www.isca-speech.org/archive/interspeech_2015/i15_3046.html

Publisher's Note