Aligning Speech to Languages to Enhance Code-switching Speech Recognition

Hexin Liu, Xiangyu Zhang, Leibny Paola Garcia-Perera, Andy W. H. Khong, Eng Siong Chng, Shinji Watanabe

***Abstract*—Code-switching (CS) refers to the switching of languages within a speech signal and results in language confusion for automatic speech recognition (ASR). To address language confusion, we propose the language alignment loss that performs frame-level language identification using pseudo language labels learned from the ASR decoder. This eliminates the need for frame-level language annotations. To further tackle the complex token alternatives for language modeling in bilingual scenarios, we propose to employ large language models via a generative error correction method. A linguistic hint that incorporates language information (derived from the proposed language alignment loss and decoded hypotheses) is introduced to guide the prompting of large language models. The proposed methods are evaluated on the SEAME dataset and data from the ASRU 2019 Mandarin-English code-switching speech recognition challenge. The incorporation of the proposed language alignment loss demonstrates a higher CS-ASR performance with only a negligible increase in the number of parameters on both datasets compared to the baseline model. This work also highlights the efficacy of language alignment loss in balancing primary-language-dominant bilingual data during training, with an 8.6% relative improvement on the ASRU dataset compared to the baseline model. Performance evaluation using large language models reveals the advantage of the linguistic hint by achieving 14.1% and 5.5% relative improvement on test sets of the ASRU and SEAME datasets, respectively.**

***Index Terms*—code-switching, speech recognition, alignment, language, large language model**

I. Introduction

CODE-Switch (CS) refers to the switching of languages within a spontaneous multilingual recording. Intra-sentence code-switching occurs when the language changes within a single sentence, while inter-sentence code-switching involves the switching of languages at the sentence boundaries. Unlike monolingual speech, code-switched speech presents a greater challenge for automatic speech recognition (ASR) due to language confusion and the lack of annotated data.

Although a CS-ASR system can operate similarly to monolingual ASR by combining language-specific vocabularies [1], [2], recent works address challenges associated with language confusion by incorporating language information. One direct approach is to optimize the ASR and language identification (LID) or diarization (LD) tasks jointly [3]–[7]. Here, LID and LD determine the language identity of speech samples during training to enrich the model with language information [6], [8], while only the ASR output is computed during inference. Apart from joint optimization, approaches based on the bi-encoder and the mixture-of-experts method have been proposed [9], [10]. These approaches have been derived from the Transformer architecture [11], [12], where the models incorporate two encoders pre-trained on monolingual data independently to capture language-specific information. Pre-trained language-specific encoders have also been adopted in other architectures due to their effectiveness in distinguishing languages [13], [14]. In contrast to language-specific encoder modules, the language-aware decoder module has been explored to reduce multilingual contextual information via a language-specific self-attention mechanism within the Transformer decoders [15]. In addition, a conditional factorization method factorizes CS-ASR into two monolingual recognition processes before integrating multiple recognized monolingual segments into a single bilingual sequence [16]. As an extension, a conditionally factorized connectionist temporal classification (CTC) module that allows for a zero-shot setting has been proposed [17], [18]. A factored language model integrating syntactic and semantic features in a code-switched language model has also been explored to enhance CS-ASR [19].

Although existing works achieve reasonable CS-ASR performance, constraints that impede model performance persist. In particular, code-switching timestamps are often excluded in code-switching corpora since the annotation process is resource intensive and requires expertise in bilingualism. Such a limitation renders the application of supervised language identification for joint optimization impractical [5]. Furthermore, learning discriminative language information successfully during training relies on language-specific encoders or decoders, resulting in a significant increase in model parameters. For bilingual code-switching speech, the accent of the primary language may also introduce bias to the secondary language, resulting in the two languages being similar auditorially [20], [21]. Achieving language identification on accented speech, therefore, remains a key challenge [22].

We propose a language alignment loss (LAL) to enrich the CS-ASR model with language information. This is achieved by capturing frame-level language information without the need for additional language annotations. While existing multilingual ASR benefits from utterance-level one-hot language vector (since only one language is present in each utter-

Hexin Liu and Eng Siong Chng are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798.

Xiangyu Zhang is with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney NSW 2052, Australia.

Leibny Paola Garcia-Perera is with CLSP & HLT-COE, Johns Hopkins University, USA ◦

Andy W. H. Khong is with the School of Electrical and Electronic Engineering and the Lee Kong Chian School of Medicine, Nanyang Technological University, Singapore 639798.

Shinji Watanabe is with Carnegie Mellon University, Pittsburgh, PA 15213 USA.

ance [23], [24]), a code-switching speech signal may contain two languages. Therefore, detecting language change points at high granularity (in particular at frame level) is preferable as opposed to the use of a one-hot vector for each utterance [7]. Furthermore, incorporating the LAL requires a single linear layer resulting in only a negligible increase in model parameters compared to existing approaches that require additional encoders or decoders. Lastly, as the by-product of LAL, frame-level language predictions can be summarized as an utterance-level hint. This hint aims to facilitate the incorporation of an external large-scale language model (LLM) in ASR via a generative error correction method [25]–[27].

The remainder of this paper is organized as follows: Section II introduces the hybrid CTC/attention model, which serves as a baseline model on which we apply the proposed method. The proposed LAL and the incorporation of the code-switching hint in external language modeling are presented in Sections III and III-D, respectively. Information pertaining to the datasets, model configurations, and experiment setup are provided in Section IV. We present results and analysis in Section V before highlighting the advantages and limitations of our proposed methods in Section VI. Finally, we conclude our work in Section VII.

II. Preliminary

A. Conformer-based hybrid CTC/attention ASR model

We employ a Conformer-based hybrid CTC/attention ASR model as the baseline in this paper. The hybrid CTC/attention model comprises an encoder, a decoder, and a CTC module, where the decoder and CTC modules share the encoder outputs [28]. The encoder and decoder modules comprise Conformer encoder and Transformer decoder layers [11], [29], respectively [2].

Given a speech signal, we define its acoustic features as $\mathbf{X} = (\mathbf{x}_t \in \mathbb{R}^F | t = 1, \ldots, T)$ and paired token sequence as $W = (w_n \in \mathcal{V} | n = 1, \ldots, N)$, where $\mathcal{V}$ is the vocabulary, T and N are the lengths of feature and token sequences, respectively, and F is the dimension of acoustic feature. The encoder generates hidden outputs $\mathbf{H} = (\mathbf{h}_t \in \mathbb{R}^D | t = 1, \ldots, T')$ from $\mathbf{X}$, which are subsequently used as inputs into the decoder and CTC modules. Here, T' is the length of the hidden output sequence, where $T' < T$ due to the subsampling, and D is the dimension of the hidden output. With $\mathbf{H}$, the CTC module computes the token sequences in accordance with the Bayesian decision theory by factorizing $p_{\text{ctc}}(W|\mathbf{X})$ as [18]

$$\begin{aligned} p_{\text{ctc}}(W|\mathbf{X}) &= \sum_{Z} p(W|Z, \mathbf{X})\, p(Z|\mathbf{X}) \\ &\approx \sum_{Z} p(W|Z)\, p(Z|\mathbf{X}), \end{aligned} \tag{1}$$

where $Z = (z_t \in \mathcal{V} \cup \{< blank >\} | t = 1, \ldots, T')$ is a framewise token sequence conditioned on $\mathbf{X}$. The variable

$$p(Z|\mathbf{X}) = \prod_{t=1}^{T} p(z_t | z_1, \ldots, z_{t-1}, \mathbf{X}) \approx \prod_{t=1}^{T} p(z_t|\mathbf{X}) \tag{2}$$

is the acoustic model of CTC, where the probabilistic chain rule and the conditional independence assumption have been invoked. Exploiting the Bayes' rule, the probabilistic chain rule, and the conditional independence assumption, the CTC token model is given by

$$\begin{aligned} p(W|Z) &= \frac{p(W)\,(Z|W)}{p(Z)} \\ &= \prod_{t=1}^{T} p(z_t|z_1, \ldots, z_{t-1}, W)\,\frac{p(W)}{p(Z)} \\ &\approx \prod_{t=1}^{T} p(z_t|z_{t-1}, W)\,\frac{p(W)}{p(Z)}. \end{aligned} \tag{3}$$

Consequently, the term $p(W)/p(Z)$ is often excluded and hence (1) can be rewritten as

$$p_{\text{ctc}}(W|\mathbf{X}) \approx \sum_{Z} \prod_{t=1}^{T} p(z_t|z_{t-1}, W)\, p(z_t|\mathbf{X}). \tag{4}$$

We note that tokens W are embedded into $\mathbf{W} = (\mathbf{w}_n \in \mathbb{R}^D | n = 1, \ldots, N)$ before being fed into the decoder module along with $\mathbf{H}$. The decoder then predicts the next token w_n based on historical tokens $w_{1:n-1}$ and $\mathbf{H}$ via

$$p(w_n|w_{1:n-1}, \mathbf{X}) = \text{Decoder}(\mathbf{w}_{1:n-1}, \mathbf{H}), \tag{5}$$

where $p(w_n|w_{1:n-1}, \mathbf{X})$ is the posterior of w_n given acoustic features and historical tokens, and $\text{Decoder}(\cdot)$ denotes the Transformer decoder. The encoder-decoder module computes the token sequences by factorizing $p_{\text{att}}(W|\mathbf{X})$ as

$$p_{\text{att}}(W|\mathbf{X}) \approx \prod_{n=1}^{N} p(w_n|w_{1:n-1}|\mathbf{X}). \tag{6}$$

The model is optimized via a multi-task objective function

$$\mathcal{L}_{\text{asr}} = \alpha \mathcal{L}_{\text{ctc}} + (1 - \alpha)\, \mathcal{L}_{\text{att}}, \tag{7}$$

where $\mathcal{L}_{\text{ctc}}$ denotes the CTC loss, $\mathcal{L}_{\text{att}}$ denotes the cross-entropy loss with label smoothing for the encoder-decoder branch [30], and α is a parameter associated with multi-task learning. The decoding process aims to maximize the linear combination of the logarithmic CTC and attention objectives such that the decoded token sequence is given by

$$\widehat{W} = \underset{W}{\text{argmax}} \left\{ \alpha \text{log} p_{\text{ctc}}(W|\mathbf{X}) + (1 - \alpha) \,\text{log} p_{\text{att}}(W|\mathbf{X}) \right\}. \tag{8}$$

Here, $\widehat{W}$ is also referred to as a single hypothesis, and the final hypothesis of the given speech signal is chosen as the one with the highest likelihood among multiple hypotheses generated during the beam search. While the hybrid CTC/attention model has shown to be effective for the CS-ASR task, it performs CS-ASR similarly to monolingual ASR without exploiting any code-switching information, which, as a consequence, limits the CS-ASR performance.

B. Improving ASR via LLM and efficient fine-tuning

Due to the scarcity of code-switching data, general code-switching language models moderately improve the performance of a CS-ASR system when being incorporated through shallow fusion [4], [7]. Although developing an external

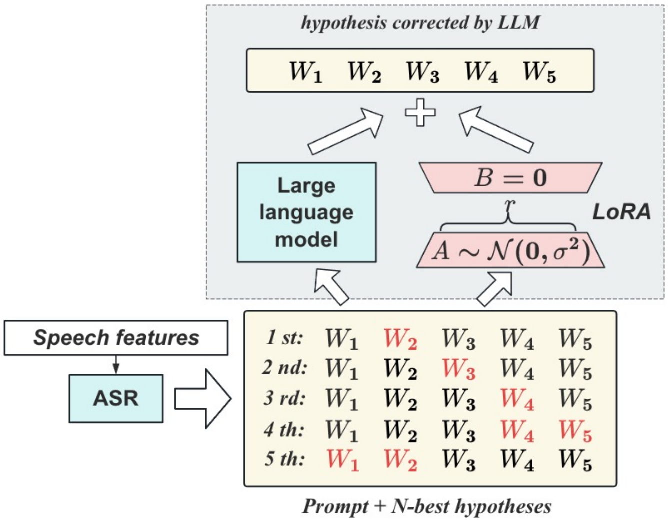

Fig. 1. The in-context learning-based generative error correction method for improving ASR with an external LLM proposed in [27]. Variable r is the low intrinsic rank, and matrices $\mathbf{A}$ and $\mathbf{B}$ are initialized by a random Gaussian distribution and zero, respectively. The red and black tokens within the N-best hypotheses denote the wrongly and correctly predicted tokens, respectively.

language model on monolingual data or synthesized code-switching data has shown to be effective [31]–[33], the performance is limited by the domain mismatch. Since large language models have achieved success in natural language processing and have been extended for computer vision and speech signal processing applications [25], [34], [35], we propose to adopt open-source LLMs, which is robust against diverse domains due to the large-scale training data, to improve CS-ASR by addressing the complex token alternatives in bilingual scenarios.

Recent works have also attempted to improve speech recognition through the use of LLMs. A direct method involves prompting an LLM using paired discrete speech and text embeddings [36]. An in-context learning-based generative error correction method has also been applied to LLMs [27] as presented in Fig. 1, where the final prediction is generated by summarizing and correcting the N-best ASR hypotheses [37]. This approach has shown to be effective in monolingual ASR. However, code-switching leads to more token alternatives that are of similar auditory or syntactic characteristics compared to a monolingual application—direct transference of the generative error correction to CS-ASR may not be desirable.

In addition, to fine-tune an LLM efficiently, low-rank adaptation (LoRA) [38] has been proposed. As shown in Fig. 1, computational complexity is reduced by freezing the pre-trained LLM and injecting trainable rank decomposition matrices $\mathbf{A}$ and $\mathbf{B}$ into its Transformer-based layers. The forward pass is then defined as the linear combination of the pre-trained model $\mathbf{M}_0$ and the trained decomposed matrices $\mathbf{A}$ and $\mathbf{B}$ such that

$$(\mathbf{M}_0 + \mathbf{\Delta M})\,\mathbf{X} = (\mathbf{M}_0 + \mathbf{AB})\,\mathbf{X}, \tag{9}$$

where $\mathbf{\Delta M}$ is the model parameters of the model update. Matrices $\mathbf{A}$ and $\mathbf{B}$ are initialized as by random Gaussian and zero values, respectively, so that $\mathbf{\Delta M} = \mathbf{AB} = 0$ at the beginning of training.

III. Language Alignment Loss

Since multilingual ASR benefits from the supplementary language information offered by utterance-level one-hot language vectors, we propose to enhance CS-ASR performance by incorporating frame-level language information, enabling the detection of code-switching at a high granularity. In this work, we introduce the LAL that is incorporated to the encoder-decoder framework to capture language information as shown in Fig. 2.

A. Frame-level language identification

To capture frame-level language information, we employ a linear layer as a built-in language classifier. This layer takes the hidden output unit of the encoder module as its input and aims to generate a language decision for each hidden output unit.

Due to the lack of frame-level language ground truth, existing works usually perform language identification in an unsupervised manner [7], [9]. Nevertheless, this unsupervised frame-level classification extends beyond language identities and includes elements such as phonemic or domain information. Therefore, incorporating language information becomes important in guiding the unsupervised language identification process.

Although frame-level language annotations are unavailable, token-level language information can readily be inferred from text, particularly in cases where there is a notable contrast in character structure or morphology between the two languages. Past research has investigated the conversion of byte-pair encoding (BPE) tokens into their respective language labels to facilitate language identification or diarization [4], [39], [40]. Here, we employ a similar conversion strategy, where BPE tokens are first transformed into token-level labels corresponding to their respective languages. The pseudo-frame-level language annotations are subsequently extracted by aligning frames to these token-level language labels.

B. Aligning frames with token-level language labels

The encoder-decoder model achieves ASR by mapping speech features to tokens. The alignment between speech frames (i.e., encoder hidden outputs) and text (i.e., tokens) is inherently learned through the cross-attention process illustrated on the left of Fig. 3. Given multiple attention heads within the last Transformer decoder layers, their cross-attention matrices are averaged to generate the weight matrix $\mathbf{Atten} \in \mathbb{R}^{T' \times N}$ of the speech-to-text alignment shown in the top matrix of Fig. 3. The alignment between hidden outputs and languages can then be derived from the frame-to-token weight matrix illustrated by the transition from the top to bottom matrices in Fig 3. Specifically, the averaged cross-attention weight matrix can be decomposed into vectors along T', being $\mathbf{Atten} = (\mathbf{atten}_t \in \mathbb{R}^N | t = 1, \ldots, T')$. Each element in $\mathbf{atten}_t$ denotes the attention weight corresponding

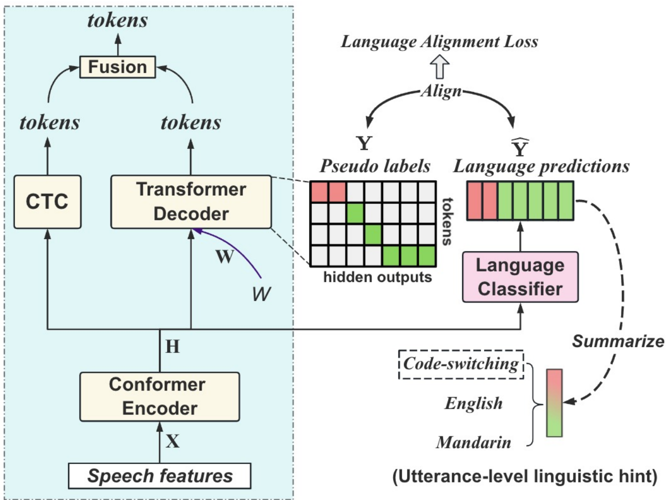

Fig. 2. The hybrid CTC/attention model (in blue) with language alignment process.

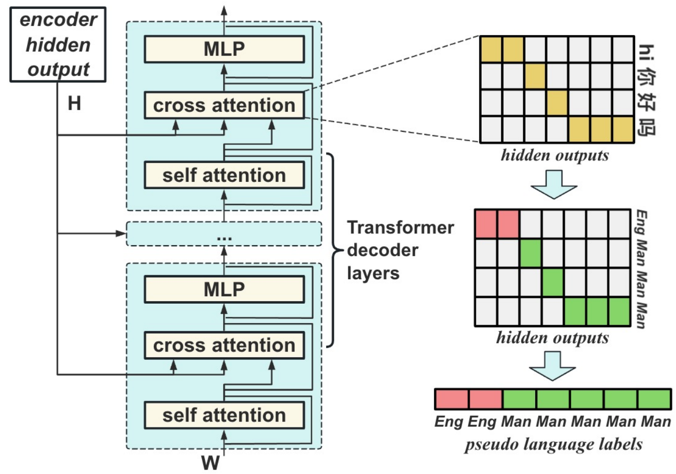

Fig. 3. The pseudo frame-to-language label conversion process. On the right side, the upper matrix represents the cross-attention weight matrix averaged over all attention heads within the last decoder layer. The middle matrix represents the converted frame-to-language weight matrix and the pseudo language labels are illustrated at the bottom. In this illustrative example, red and green cells denote the English and Mandarin tokens, respectively.

to a BPE token within the input token sequence. Each frame can then be assigned a pseudo language label which is the language of the BPE token corresponding to the highest weight. This is achieved via

$$\mathbf{y}_t = \mathrm{T2L}\big(\operatorname*{argmax}_n (\mathbf{atten}_t)\big), \tag{10}$$

where $\mathbf{y}_t$ is the pseudo-frame-level language label and $\mathrm{T2L}(\cdot)$ represents the conversion from the n-th BPE token which has the highest attention weight in $\mathbf{atten}_t$ to its language label.

The pseudo language labels are concatenated to form a sequence $\mathbf{Y} = (\mathbf{y}_t \in \mathbb{R}^C | t = 1, \ldots, T')$. Here, T' is the number of hidden output units and C is the number of languages, where all special tokens are considered as a single language. Each hidden output unit is subsequently projected to a language decision $\mathbf{y}_t$ by a language classifier comprising one linear layer

$$\widehat{\mathbf{y}}_t = \mathrm{Linear}(\mathbf{h}_t), \tag{11}$$

where $\mathrm{Linear}(\cdot)$ denotes computations within a linear layer. Defining $\exp(\cdot)$ as the exponential operation, the language alignment loss for each speech sample is computed via a cross-entropy function

$$\mathcal{L}_{\mathrm{lid}} = \frac{-1}{T'} \sum_{t=1}^{T'} \sum_{c=1}^{C} \log \frac{\exp(\widehat{\mathbf{y}}_{t,c})}{\sum_{i=1}^{C} \exp(\widehat{\mathbf{y}}_{t,i})} \mathbf{y}_{t,c}, \tag{12}$$

where $\widehat{\mathbf{y}}_{t,i}$ denotes the value at the ith dimension of the language prediction vector for the tth hidden output vector. Similarly, $\mathbf{y}_{t,c}$ denotes the value at the cth dimension of the one-hot pseudo language label vector for the tth hidden output vector such that $\mathbf{y}_{t,c} = 1$ for the target language with the remaining elements being zero.

The performance of the CS-ASR model improves during model training as the language information is incorporated, reaching its peak upon being optimized completely. The heightened ASR performance contributes to increasingly accurate language labels. This improvement, in turn, leads to higher language identification performance. Therefore, extracting pseudo language labels and language identification has become integral to the iterative optimization process during training.

C. Balancing training via toke-level language weights

The grammatical structure of code-switching data defines the matrix language as the main language and the embedded language as the secondary language [41]. While code-switching corpora such as SEAME do not exhibit a dominant language [19], [20], the matrix language may prevail in some code-switching corpora, resulting in an imbalanced distribution of tokens and speech frames between the two languages [21]. Therefore, a CS-ASR model trained on such imbalanced data may overfit the matrix language and underfit the embedded language.

While computing the number of speech frames for each language is challenging, the token distribution within a code-switching corpus can be assessed before training. Therefore, we propose to address the imbalance issue by incorporating language weights in (12), being a weighted cross-entropy function as

$$\mathcal{L}_{\mathrm{lal}} = \frac{-1}{T'} \sum_{t=1}^{T'} \sum_{c=1}^{C} w_c^{\mathrm{lang}} \log \frac{\exp(\widehat{\mathbf{y}}_{t,c})}{\sum_{i=1}^{C} \exp(\widehat{\mathbf{y}}_{t,i})} \mathbf{y}_{t,c}, \tag{13}$$

where w_c^{lang} denotes the normalized language weight of language c and can be made inversely proportional to its token count in the training data. However, the token ratio may not align precisely with speech frames due to variations in speech rates across languages. We therefore propose to tune them initially based on the token ratio and accounting for speech rate variations when applying language weights to the LAL.

The relationship between the initial language weights can be illustrated by

$$\frac{w_{\text{Eng}}^{\text{lang}}}{w_{\text{Man}}^{\text{lang}}} \propto \frac{\text{Count}(\text{tokens}_{\text{Man}})}{\text{Count}(\text{tokens}_{\text{Eng}})}, \tag{14}$$

where $\text{Count}(\cdot)$ denotes the number of BPE tokens belonging to the language. In particular, fewer tokens in the training data and a higher speech rate result in a higher language weight.

With the above, the CS-ASR model is optimized via an objective function being the sum of CTC loss, encode-decoder loss, and the proposed LAL such that

$$\mathcal{L}_{\text{asr}} = \alpha \mathcal{L}_{\text{ctc}} + (1 - \alpha)\, \mathcal{L}_{\text{att}} + \beta \mathcal{L}_{\text{lal}}, \tag{15}$$

where β denotes the weight of the language alignment loss during training. The decoding process is similar to (8).

D. Linguistic hint for prompting LLM

With reference to Fig. 1, we propose to adopt the LLM-based generative error correction method to improve CS-ASR [27]. To fine-tune the LLM efficiently, LoRA is employed while the LLM is kept frozen during training as illustrated in Section II-B. The prompt originally designed in [27] is used as shown in Fig. 4(a), where the N-best hypotheses are extracted from the ASR output before being inserted in the prompt. The LLM is then optimized to deduce the correct transcript, which is the ground-truth transcription shown in Fig. 4 during training, by leveraging the information provided in these hypotheses. However, code-switching gives rise to more intricate token alternatives with similar auditory or syntactic characteristics compared to a monolingual application. This complexity persists when performing generative error correction for ASR hypotheses. Inspired by the use of chain-of-thought as additional supervision when fine-tuning LLM for downstream tasks [26], [42], we propose to employ an additional linguistic hint during prompting to address the aforementioned challenge in CS-ASR.

To this end, various methods for linguistic hint extraction can be used. An acoustic-biased linguistic hint can be derived from the by-product of the proposed LAL. As shown in Fig. 2, frame-level language predictions are first normalized using the softmax function before generating an utterance-level language decision. In the context of this work, this decision can either be monolingual or multilingual, with the former providing a single language code. In addition, the linguistic hint can be obtained from the decoded hypotheses (i.e., text information), allowing it to be employed in conjunction with the acoustic-biased hint through the weighted voting mechanism.

The linguistic hint is then inserted in the used prompt during fine-tuning as shown in Fig. 4(b). We propose two types of hints—the monolingual hint, where only *<language id>* words are included in the transcription, and the code-switching hint, where speech is multilingual and that *<both language ids>* words are included in the transcription.

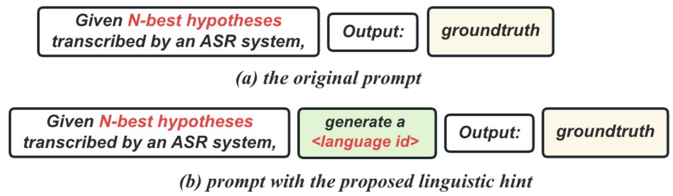

Fig. 4. The original (a) and proposed (b) prompts during fine-tuning LLM and performing generative error correction, *language id* is selected from two languages ("English" and "Mandarin" in this work) and "multilingual".

IV. Dataset, experiment, and model configuration

A. Datasets

We conducted the experiments on data extracted from the ASRU 2019 Mandarin-English code-switching speech recognition challenge [20], [21] and the SEAME dataset. The ASRU 2019 Mandarin-English code-switching speech recognition challenge consists of four datasets, including a 500-hour Mandarin-only training set, a 200-hour intra-sentence English-Mandarin code-switching training set, a 40-hour intra-sentence English-Mandarin code-switching development set, and a 20-hour intra-sentence English-Mandarin code-switching test set. In the experiments, the models were trained on the 200-hour CS training set, validated on the development set, and evaluated on the test set. The SEAME dataset, on the other hand, is a Mandarin-English code-switching corpus containing spontaneous conversational speech [20]. This dataset encompasses both intra- and inter-sentence code-switching speech. We divided the SEAME dataset into a 96.6-hour training set, a 4.9-hour validation set, and two test sets being denoted as $\text{test}_{\texttt{man}}$ and $\text{test}_{\texttt{sge}}$ following the same partitioning method described in [3]. Detailed information about the test sets is provided in Table I, while Table II highlights the duration ratio of each language, where the language labels are annotated at the utterance level.

While both datasets involve English-Mandarin code-switching, the primary distinction between them lies in the accent. The ASRU data was recorded in mainland China and is characterized by a dominant Chinese accent and text. In the training and development sets, each sentence, on average, consists of 8.6 Chinese characters and 1.6 English words. In contrast, the SEAME dataset comprises audio recordings from Singapore and Malaysia, characterized by South-East Asian accents. Additionally, code-switching occurs more frequently within the SEAME dataset compared to the ASRU data due to the bilingual education and language policies in Singapore and Malaysia [43]. This suggests that the SEAME data might pose greater challenges for CS-ASR compared to the ASRU data.

B. Data preprocessing

Since the SEAME dataset contains a small training set of about 98 hrs, we augmented the training data using speed perturbation and SpecAugment [44], [45]. Two training strategies were adopted, where one develops the model on data without speed perturbation while the other trains the model on the

TABLE I
DETAILS OF TWO DATASETS IN TERMS OF DIVISION AND DURATIONS

Corpus	Subset	Duration (hours)
ASRU	train	193.0
	dev	21.3
	test	20.4
SEAME	train	96.6
	dev	4.9
	$\text{test}_{\texttt{man}}$	7.5
	$\text{test}_{\texttt{sge}}$	3.9

TABLE II
DURATION RATIO FOR MANDARIN, ENGLISH, AND CODE-SWITCHING SPEECH OF TEST SETS IN ASRU AND SEAME DATA. THE LANGUAGE ANNOTATIONS ARE AT UTTERANCE LEVEL

Subset	Duration ratio (%)		
	Mandarin	English	Code-Switching
ASRU test	0	0	100
$\text{test}_{\texttt{man}}$	14	7	79
$\text{test}_{\texttt{sge}}$	6	41	53

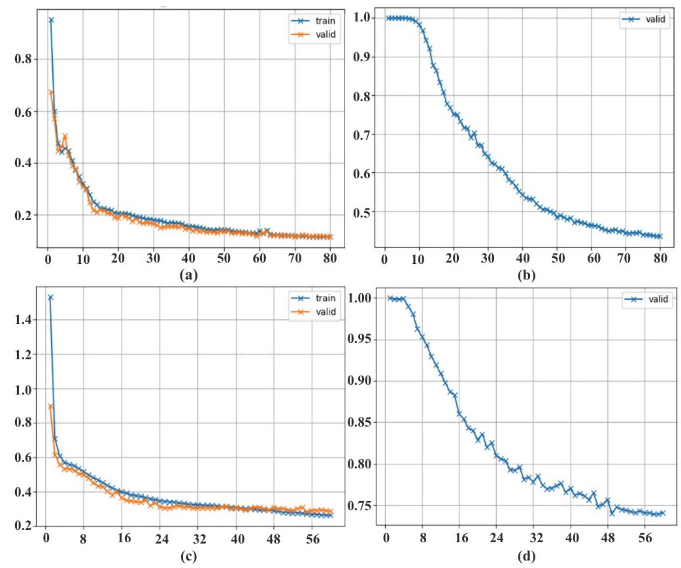

Fig. 5. The convergence of the proposed language alignment loss (y-axis for the left figures) during training and validation on the (a) ASRU data and (c) SEAME data, and the corresponding MER (y-axis for the right figures) on the validation set of (b) the ASRU data and (d) SEAME data, against training epochs (x-axis).

entire augmented data. The speech perturbation was applied with factors 0.9, 1.0, and 1.1. With respect to the ASRU dataset, only SpecAugment was applied for data augmentation. SpecAugment adopted the default setup in ESPnet for two datasets [2]. The time-warp mask size was set to five, and two time and frequency masks were applied, with their lengths uniformly selected from the range of 0 to 40 for time masks and 0 to 30 for frequency masks. Speech samples within both corpora were segmented into durations ranging from 0.1 to 20 s. We extracted $F = 80$ dimensional log-Mel-Fbank features for each speech segment before applying the cepstral mean and variance normalization.

We employed BPE to tokenize the English words in the two English-Mandarin code-switching corpora and split all Mandarin words into individual characters. For the SEAME dataset, this resulted in a total of $V = 5,628$ tokens, including 3,000 English BPE tokens, 2,624 Mandarin characters, and four special tokens for <*unk*>, <*noise*>, <*blank*>, and <*sos/eos*>. For the ASRU data, the same tokenization process yielded a total of $V = 6,923$ tokens, including 3,000 English BPE tokens, 3,920 Mandarin characters, and three special tokens for <*unk*>, <*blank*>, and <*sos/eos*>. The variable $C = 3$ denotes three language classes converted from special, English, and Mandarin tokens.

We fine-tuned the Chinese LLaMA-2 on a subset of the SEAME training set comprising approximately 60,000 speech segments and the development set of the ASRU data comprising approximately 20,000 speech segments, respectively. N-best lists of the above speech segments were subsequently extracted and incorporated into the prompts. In addition, we removed the <*noise*> and <*unk*> labels from the N-best list since the LLM can hardly address these special tokens.

C. Model configuration and experiment setup

The baseline model is a Conformer-based hybrid CTC/Attention ASR model comprising twelve Conformer encoder layers, six Transformer decoder layers, and a CTC module. The Conformer employs the macaron structure [29], where its convolutional neural network (CNN) module has a kernel size of 15. The swish activation function is applied [46]. A CNN layer first subsamples the input features and projects them into $D = 256$ dimensions before feeding into the macaron modules. All attention layers within the encoder and decoder modules 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 bi-encoder and language posterior bias methods [4], [9] have also been implemented as baselines. In the bi-encoder CS-ASR system, we replicated the encoder module of the baseline model described in the previous paragraph to form the bi-encoder module. The outputs of these two encoders are fed into a linear layer, which serves as a built-in LID module with an output dimensionality of two (corresponding to two languages), before calculating the weighted encoder outputs. It is useful to note that the encoder modules of the bi-encoder method are pre-trained on English and Mandarin corpora separately. Nevertheless, this pre-training was deliberately omitted to ensure a fair comparison. For the language posterior bias-based (LPB) CS-ASR model, the model configuration recommended in [4] was used. The LPB system closely resembles the baseline hybrid CTC/attention model, with an additional language diarization decoder. This language diarization decoder mirrors the structure of the ASR decoder.

We trained the baseline model on the ASRU and SEAME data for seventy and fifty epochs, respectively. The proposed method and reproduced works were trained for an additional ten epochs due to their more complex objective functions or parameters. All models were optimized by an Adam optimizer on two RTX 3090 GPUs. The learning rate increased from 0 to 0.001 over 25,000 update steps, followed by a cosine annealing decay. Parameter $\alpha = 0.3$ and a label smoothing factor of 0.1 were used in (7) and (15). The ten best models during validation were averaged for inference. We adopted the ten-best beam search method with $\alpha = 0.4$ in (8). Evaluation of the proposed systems was quantified via the mix error rate (MER) comprising word error rate (WER) for English and character error rate (CER) for Mandarin.

Experiments associated with the proposed linguistic hint utilized a Chinese LLaMA2-7B model [1]. The 5-best hypotheses were first extracted from the ASR output before being used in the prompt to fine-tune the LLM. Fine-tuning was performed via LoRA with rank $r = 4$ and the AdamW optimizer for ten epochs with a batch size of 128 on the SEAME data, where LoRA is applied to the query and value modules within the self-attention modules. The learning rate was increased from 0 to 0.0002 over 100 update steps, followed by a linear decay. During inference, a temperature of 0.7 was applied to allow for the creativity of the LLM, while other hyper-parameters were fixed at their default settings. We evaluated all methods by employing the MER comprising WER for English and CER for Mandarin.

V. Results and analysis

A. Impact of β values

To assess the impact of β on the CS-ASR performance for the proposed LAL, we first compare the performance of the hybrid CTC/attention model optimized with LAL via (15) for various β values on the ASRU data. Results summarized in Table III highlight that with the incorporation of LAL during training, the hybrid CTC/attention model consistently outperforms the vanilla model. Notably, the best MER of 11.9% is observed when $\beta = 1.5$, achieving a relative improvement of 7.03% compared to the vanilla model. In addition, this performance improvement remains consistent across various LAL parameters, particularly within the range of $1.0 \leq \beta \leq 3.0$.

We next conduct experiments on the SEAME dataset without using speed perturbation during training and results are presented in Table IV. Similar to the ASRU dataset, the hybrid CTC/attention model that incorporates the proposed LAL outperforms the vanilla configuration. In addition, the hybrid CTC/attention model with LAL exhibits consistently high performance for $1.0 \leq \beta \leq 3.0$. This robustness on various datasets and β values indicates the effectiveness of the proposed LAL. As opposed to results achieved on the ASRU dataset, the highest overall performance on the SEAME data is achieved when $\beta = 0.1, 0.5$, and 3.0. Due to the different duration ratios of languages of the two test sets, the above implies that a lower β value leads to a higher performance on the dataset containing predominantly monolingual data.

[1]https://huggingface.co/hfl/chinese-llama-2-7b

TABLE III
Performance evaluation of the proposed method on the ASRU data with different β values in terms of substitutions, deletions, insertions, and the total MER (%)

Method	β for LAL	Sub ↓	Del ↓	Ins ↓	MER ↓
Hybrid CTC/atten	0	11.5	0.6	0.7	12.8
+ LAL	0.5	11.2	0.6	0.7	12.4
	1.0	10.9	0.6	0.7	12.1
	1.5	10.7	0.5	0.6	**11.9**
	2.0	10.8	0.5	0.6	12.0
	2.5	10.9	0.5	0.6	12.0
	3.0	10.9	0.5	0.6	12.0
	4.0	11.4	0.6	0.6	12.6
	5.0	11.0	0.6	0.6	12.2

TABLE IV
Performance evaluation of the proposed method trained without speed perturbation on the SEAME dataset with different β values in terms of substitutions, deletions, insertions, and the total MER (%)

Method	β for LAL	Subset	Sub ↓	Del ↓	Ins ↓	MER ↓
Hybrid CTC/atten	0	test_{man}	12.0	3.0	2.2	17.2
		test_{sge}	17.3	4.1	3.1	24.5
+ LAL	0.1	test_{man}	11.8	2.9	2.2	16.8
		test_{sge}	17.0	4.0	3.0	**23.9**
	0.5	test_{man}	11.8	2.9	2.2	16.8
		test_{sge}	17.0	4.0	3.0	**23.9**
	1.0	test_{man}	11.7	3.0	2.1	16.8
		test_{sge}	17.1	4.0	3.1	24.1
	2.0	test_{man}	11.9	3.0	2.0	16.9
		test_{sge}	17.0	4.0	3.1	24.1
	3.0	test_{man}	11.7	3.0	2.0	**16.7**
		test_{sge}	17.0	4.0	3.0	24.0
	4.0	test_{man}	11.9	3.0	2.0	16.9
		test_{sge}	17.1	4.2	3.0	24.2
	5.0	test_{man}	11.8	3.0	2.2	17.0
		test_{sge}	17.4	4.1	3.1	24.5

B. Balancing the ASRU dataset during training

The effect of language weights employed in (13) for balancing the Mandarin-dominant ASRU data is shown in Table V. It is worth noting that, similar to the number of English frames, the number of frames for the class "other" is also significantly lower than that for the Mandarin frames. Since the class "other" does not contribute to the language identities in the CS-ASR task, this class is not balanced during training and the weight is always set to $w_other^{\text{lang}} = 1$ except for the learnable language weight. In this work, $w_c^{\text{lang}} = 1$ for all classes is the default setup.

Results presented show that the CS-ASR model gains moderate performance improvement on both English and Mandarin data from high English weights, where the weights of "other" and "Mandarin" were set to 1. This is consistent with our assumption that a high English weight can achieve balance for the secondary language during training, which consequently improves the model performance. Moreover, the highest per-

TABLE V
PERFORMANCE OF THE PROPOSED METHOD WITH $\beta = 1.5$ AND DIFFERENT LANGUAGE WEIGHTS TO BALANCE THE MANDARIN-DOMINANT ASRU DATA DURING TRAINING IN TERMS OF ENGLISH WER, MANDARIN CER, AND MER (%)

Language weights ($\beta = 1.5$)			ASRU test		
Other	**Eng**	**Man**	**Eng ↓**	**Man ↓**	**MER ↓**
vanilla (all 1)			35.4	9.3	11.9
1	10	1	35.2	9.3	11.8
1	50	1	35.3	9.3	11.8
1	100	1	**35.1**	**9.2**	**11.7**
1	100	100	35.8	9.5	12.1
1	1000	1	35.4	9.3	11.8

TABLE VI
RESULTS OF THE PROPOSED METHOD ON SEAME DATASET AFTER EMPLOYING SPEED PERTURBATION WITH FACTOR 0.9, 1.0, AND 1.1 DURING TRAINING IN TERMS OF SUBSTITUTIONS, DELETIONS, INSERTIONS, AND THE TOTAL MER (%)

Method	β for LAL	Subset	Sub ↓	Del ↓	Ins ↓	MER ↓
Hybrid CTC/atten	0	$\text{test}_{\texttt{man}}$	11.5	3.0	2.0	16.6
		$\text{test}_{\texttt{sge}}$	16.4	2.9	3.0	**23.3**
+ LAL	0.5	$\text{test}_{\texttt{man}}$	11.6	3.0	2.1	16.7
		$\text{test}_{\texttt{sge}}$	16.4	4.1	3.0	23.5
	1.0	$\text{test}_{\texttt{man}}$	11.6	2.9	2.2	16.7
		$\text{test}_{\texttt{sge}}$	16.4	4.0	3.1	23.6
	2.0	$\text{test}_{\texttt{man}}$	11.5	3.0	2.0	16.5
		$\text{test}_{\texttt{sge}}$	16.5	4.1	3.0	23.5
	3.0	$\text{test}_{\texttt{man}}$	11.3	3.0	2.1	**16.4**
		$\text{test}_{\texttt{sge}}$	16.2	4.0	2.9	**23.3**
	4.0	$\text{test}_{\texttt{man}}$	11.7	3.1	2.1	16.8
		$\text{test}_{\texttt{sge}}$	16.6	3.9	3.0	23.6
	5.0	$\text{test}_{\texttt{man}}$	11.6	3.0	2.1	16.7
		$\text{test}_{\texttt{sge}}$	16.6	4.0	3.1	23.6

formance is achieved when the weight of English is set to 100. It is also worth noting that the ratio between English to Mandarin weights is higher than the token ratio due to the difference between their speech rates.

C. Impact of speed perturbation

As discussed in Section V-B, the frame-level language identification can be affected by speech rate. We, therefore, investigate the impact of the speed perturbation on the CS-ASR performance when LAL is incorporated. The experiments were conducted on the SEAME dataset and in contrast to the experiments shown in Table IV, we utilize speed perturbation to augment the training data with results being summarized in Table VI.

Results presented show that the hybrid CTC/attention model achieves higher performance after employing speed perturbation as data augmentation. However, due to speech rate variations, one token can be aligned with a different number of acoustic frames when speed perturbation is applied. Therefore, further incorporation of the proposed LAL may lead to lower performance. In addition, the hybrid CTC/attention model with

TABLE VII
PERFORMANCE EVALUATION OF THE PROPOSED METHOD AND STATE-OF-THE-ART APPROACHES ON DATA FROM THE ASRU 2019 CHALLENGE IN TERMS OF ENGLISH WER, MANDARIN CER, AND MER (%)

Method	Num. of Params.	Error Rate (%)		
		Eng↓	**Man↓**	**Mixed↓**
Hybrid CTC/atten	48.27 M	37.1	10.2	12.8
+ LAL (ours)	48.27 M	**35.1**	**9.2**	**11.7**
Bi-encoder [9]	79.90 M	36.0	9.8	12.4
LPB [4]	59.58 M	35.3	**9.2**	11.8

TABLE VIII
PERFORMANCE EVALUATION OF THE PROPOSED METHOD AND STATE-OF-THE-ART APPROACHES ON THE SEAME DATASET IN TERMS OF ENGLISH WER, MANDARIN CER, AND MER (%)

Method	Num. of Params.	$\text{test}_{\texttt{man}}$			$\text{test}_{\texttt{sge}}$		
		Eng ↓	**Man ↓**	**Mixed ↓**	**Eng ↓**	**Man ↓**	**Mixed ↓**
Hybrid CTC/atten	48.27 M	29.2	15.0	16.6	28.2	22.0	23.3
+ LAL (ours)	48.27 M	29.1	14.8	16.4	28.3	21.7	23.2
Bi-encoder [9]	79.90 M	29.2	15.0	16.5	28.2	21.7	23.2
LPB [4]	59.58 M	-	-	**16.3**	-	-	**22.9**

LAL being incorporated achieves the highest performance with $\beta = 3$. This is consistent with the performance of its counterpart on the SEAME dataset without speed perturbation. It is useful to note that the proposed method achieves the highest performance on the ASRU dataset when $\beta = 1.5$. Since a higher β value is required for the SEAME data to achieve the highest performance, this also underpins that the SEAME dataset presents greater challenges in language discrimination than the ASRU dataset.

D. Comparison with state-of-the-art methods

We compared the performance of our proposed method with state-of-the-art approaches. These methods are evaluated on both model parameters and the CS-ASR performance are outlined in Tables VII and VIII. For the ASRU dataset, the hybrid CTC/attention model incorporating LAL with language weights achieves the best MER of 11.7% among all approaches under consideration. While the bi-encoder and LPB methods exhibit higher performance compared to the baseline model, these approaches require an additional Conformer encoder and a Transformer decoder, respectively, resulting in a notable increase in model parameters.

On the SEAME dataset, the hybrid CTC/attention model with the proposed LAL demonstrates a modest performance improvement compared to the vanilla model. Although the LPB method outperforms the proposed method, it requires a higher number of model parameters. In light of the ASRU dataset results, the above findings suggest that integrating the proposed LAL is a lightweight and efficient approach to attain high performance compared to other approaches.

TABLE IX
PERFORMANCE COMPARISON OF DIFFERENT TYPES OF LINGUISTIC HINT USED FOR PROMPTING LLM VIA GENERATIVE ERROR CORRECTION BY EMPLOYING MER (%) AND UTTERANCE-LEVEL LANGUAGE IDENTIFICATION ACCURACY (ACC %) FOR ENGLISH, MANDARIN, AND CODE-SWITCHING. THE N-BEST LIST IS DECODED FROM THE HYBRID CTC/ATTENTION MODEL WITH LAL

Ling. hint type	$\text{test}_{\texttt{man}}$		$\text{test}_{\texttt{sge}}$		ASRU test	
	MER ↓	Acc. ↑	MER ↓	Acc. ↑	MER ↓	Acc. ↑
no hint	16.4	-	23.3	-	11.0	-
LID output (LAL)	17.0	80.3	24.4	84.3	11.0	95.2
1st hypo.	16.5	93.3	23.1	92.5	11.0	99.7
hypos. vote	16.6	93.2	23.0	93.0	11.0	99.8
LAL & hypos. vote	16.6	92.9	23.1	92.5	11.0	99.9
groundtruth	**15.7**	-	**22.0**	-	**11.0**	-

TABLE X
PERFORMANCE EVALUATION OF PROMPTING LLM WITH GENERATIVE ERROR CORRECTION AFTER INCORPORATING THE LINGUISTIC HINT BY EMPLOYING MER (%). "ft" DENOTES FINE-TUNING, "gt" DENOTES THE LINGUISTIC HINT USING GROUND-TRUTH UTTERANCE-LEVEL LANGUAGE LABEL, AND "$pred$" DENOTES THE PREDICTED LINGUISTIC HINT

Method	SEAME		ASRU
	$\text{test}_{\texttt{man}}$	$\text{test}_{\texttt{sge}}$	test
Hybrid CTC/atten	16.6	23.3	12.8
+ LM $latefusion$	16.4	23.0	12.6
Hybrid CTC/atten w/ LAL	16.4	23.2	11.7
+ LM $latefusion$	16.4	23.1	11.9
+ LLM LoRA ft	16.4	23.3	**11.0**
+ LLM LoRA ft + ling. hint ($pred$)	16.5	23.1	**11.0**
+ LLM LoRA ft + ling. hint (gt)	**15.7**	**22.0**	**11.0**

E. Incorporating LLM with linguistic hint

To further enhance the CS-ASR performance, we integrate the LLM through prompting to perform generative error correction on the decoded N-best list. We compare the performance of the prompt in terms of MER and utterance-level LID accuracy before and after incorporating our proposed linguistic hints. The results on SEAME $\text{test}_{\texttt{man}}$, $\text{test}_{\texttt{man}}$, and the ASRU test set are presented in Table IX. Here, the LID accuracy for the hypothesis within the N-best list is determined by summarizing the languages of tokens and that "LID output" denotes the by-product of the proposed LAL.

Similar to existing works [8], [47], it is not surprising that the LID accuracy achieved with the incorporation of textual information is higher than the by-product of the proposed LAL. This can be attributed to the fact that an ASR system models both acoustic and language characteristics, whereas an LID system generally focuses solely on acoustic information. Furthermore, a decrease in LID accuracy often results in a corresponding degradation in CS-ASR performance of the generated linguistic hints. The prompt incorporating the linguistic hint, which serves as the ground-truth language label, therefore exhibits significantly higher performance on the SEAME and ASRU datasets compared to other prompts.

Notwithstanding the above, prompts with other linguistic hints achieve comparable performance to the prompt without

TABLE XI
PERFORMANCE COMPARISON OF PROMPTING LLM WITH THE PROPOSED LINGUISTIC HINT ON $\text{TEST}_{\texttt{MAN}}$ SET WITH (THE ORIGINAL DATA) AND WITHOUT (REMOVING OR NORMALIZING) INTERJECTIONS OF THE SEAME DATASET BY EMPLOYING MER (%) AND UTTERANCE-LEVEL LANGUAGE IDENTIFICATION ACCURACY (ACC %). THE N-BEST LIST IS DECODED FROM THE HYBRID CTC/ATTENTION MODEL WITH LAL

Ling. hint type	$\text{test}_{\texttt{man}}$		$w/o\ interjections$	
	MER ↓	Acc. ↑	MER ↓	Acc. ↑
Hybrid CTC/atten w/ LAL	16.4	-	13.6	-
no hint	16.4	-	13.7	-
LID output (LAL)	17.0	80.3	14.5	81.1
1st hypo.	16.5	**93.3**	13.6	**95.4**
hypos. vote	16.8	93.2	13.7	95.3
LAL & hypos. vote	16.8	92.9	13.7	95.3
groundtruth	**15.7**	-	**13.4**	-

the hint, albeit with a moderately lower overall performance in terms of MER. This observation suggests that linguistic hints may introduce a substantial bias into the generative error correction process. In particular, the utilization of a correct linguistic hint in a prompt enhances CS-ASR performance. Conversely, the misclassification of a linguistic hint results in an increased error rate. This implies that the proposed linguistic hint can potentially improve CS-ASR performance, particularly when an accurate language label is available.

The performance of CS-ASR systems with external language modeling are summarized in Table X. We observe that incorporating an LM via shallow fusion improves the CS-ASR performance. The use of LLM does not lead to performance improvement on the SEAME dataset unless accompanied by the ground-truth linguistic hint. In contrast, the LLM can improve the performance on the ASRU dataset, achieving an MER of 11.0% for all types of prompts. We analyzed the performance by considering the data distribution as shown in Table II. Since all utterances in the ASRU test set are code-switched, these predicted linguistic hints all exhibit high LID accuracy, and thus show comparable CS-ASR performance.

F. Impact of interjections

As described in Section IV-A, the SEAME dataset was collected from Singapore and Malaysia, where code-switching is more frequent than the ASRU dataset—interjections such as "lah", "lor", and "ya" that often occur in the former dataset. Annotating interjections is a challenging task even for individuals with bilingual expertise. Hence, interjections can introduce confusion during language modeling, leading to a degradation in the CS-ASR performance for LLMs that have not been trained on them. We therefore removed interjections from utterances in the SEAME training and $\text{test}_{\texttt{man}}$ without changing their semantic information. The CS-ASR performance and the LID accuracy of the linguistic hint prediction before and after removing interjections when incorporating LLM are shown in Table XI. In addition, the first row shows the MER of the hybrid CTC/attention model with LAL. This is computed by comparing the decoded hypotheses and ground-truth transcriptions after removing interjections within them.

TABLE XII
COMPARISON OF THE ORIGINAL ASR WITH LAL OUTPUTS, THE LM LATE FUSION OUTPUTS, THE LLM-GENERATED OUTPUTS, AND THE GROUND-TRUTH TRANSCRIPTS FOR THE SEAME DATASET, ERRORS AND CORRECTED WORDS ARE MARKED IN RED AND BLUE, RESPECTIVELY.

Method	Output
ASR w/ LAL	唉 呀 ah yeah close already the yeah what happen to him ah but 你 先 熬 一 年 先 啦
+ LM $latefusion$	唉 呀 ah yah close already the yeah what happen to him ah but 你 先 熬 一 年 先 啦
+ LLM LoRA ft + ling. hint (gt)	ah yah ah you are close already the yeah what happened to him ah but 你 现 在 熬 一 年 先 啦
Groundtruth	ah yeah ah yeah close with me the yeah what happened to him hah but 你 先 熬 一 年 先 啦

The results indicate that removing interjections improves the CS-ASR performance significantly due to the reduction of language confusion. However, incorporating the LLM does not benefit the CS-ASR performance except for further use of the linguistic hint with ground-truth language labels. This is consistent with results presented in Tables IX and X, suggesting that a correct linguistic hint can lead to performance improvement.

VI. DISCUSSION

A. *Frame-to-token alignment from ASR decoder but not CTC*

In the hybrid CTC/attention ASR model, the frame-to-token alignment can be computed from both CTC and the cross-attention process within the ASR decoder. However, it is worth noting that CTC predictions contain <*blank*> token that results in peaky behaviour [48], [49]. Since the <*blank*> token lacks a language attribute and is unable to be converted into a language label, this peaky behaviour leads to fewer language labels in the pseudo label sequences compared to those derived through the ASR decoder. Therefore, we use the frame-to-token alignment computed from the cross-attention attention weight matrix within the ASR decoder but not the CTC to generate pseudo language labels.

B. *Where the errors happen*

The errors have been analyzed to gain insights into factors affecting the ASR performance. Compared to deletion and insertion rates, a significantly higher substitution rate is observed from the aforementioned results. The high substitution rate impose a challenge in language modeling due to a larger vocabulary and language confusion arising from a code-switching scenario. Given that code-switched text may be more readily achieved than speech, developing a robust code-switching language model is desirable to address language confusion effectively.

In addition to the model performance in CS-ASR, the performance in terms of token-level language identification is worth highlighting. Compared to the CS-ASR model trained on the SEAME dataset, the model trained on the ASRU dataset can generally identify the token-level language change points. This underpins that the SEAME dataset is more challenging than the ASRU dataset since the two languages are less discriminative in the SEAME dataset.

To gain further insights into the CS-ASR performance on the SEAME dataset, we compared the decoded outputs of the ASR model incorporating LAL, and its counterparts enhanced by LM and LLM respectively. The results are juxtaposed with the ground-truth transcription in Table XII. Here, errors are marked in red and the tokens corrected by LM and LLM are marked in blue. One factor that contributes to the errors in the CS-ASR task is language confusion. For instance, the expression “ah yeah” shares both the pronunciation and semantic characteristics with its Chinese equivalent “唉 呀”. Another notable factor that contributes to errors is the liaison, where two Mandarin characters or English words can erroneously be classified as a single entity. The liaison effect can be particularly pronounced in a spontaneous code-switching speech signal.

The LLM fine-tuned on the SEAME training set demonstrated effectiveness in correcting grammatical errors. Words such as “happened” are consequently adjusted to the correct tense. However, the model is less adept at accommodating colloquial expressions. Therefore, the third and fourth cases illustrated in Table XII have been modified to a more formal expression, leading to a higher mixed error rate.

VII. CONCLUSION

We proposed a language alignment loss to enhance CS-ASR with language information. This is achieved by performing frame-level language identification using pseudo labels derived from the ASR decoder. The hybrid CTC/attention model with the language alignment loss exhibits higher performance on the SEAME and ASRU datasets than the vanilla configuration, with only a negligible increase in model parameters during training. In addition, after incorporating language weights in LAL to achieve balance for the secondary language during training, the proposed method obtained further performance improvement and outperforms other approaches on the ASRU data. We then employ an external LLM to improve the CS-ASR performance via generative error correction. Here, a linguistic hint, which can be computed from the LAL output and decoded hypotheses, is subsequently proposed to guide the prompting. Experiments conducted suggest that an accurate linguistic hint can significantly improve the CS-ASR performance in scenarios involving both monolingual and code-switching utterances. Finally, the errors within the hypotheses are analyzed. The LLM fine-tuned on the SEAME data has shown to be effective in correcting grammatical errors, which, in contrast, leads to a lower ASR performance for spontaneous and colloquial speech.

REFERENCES

[1] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-end speech recognition: A survey," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 32, pp. 325–351, 2024.

[2] 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. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 2207–2211.

[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. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 2165–2169.

[4] H. Liu, H. Xu, L. P. Garcia, A. W. H. Khong, Y. He, and S. Khudanpur, "Reducing language confusion for code-switching speech recognition with token-level language diarization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, pp. 1–5.

[5] L.-H. Tseng, Y.-K. Fu, H.-J. Chang, and H.-y. Lee, "Mandarin-English code-switching speech recognition with self-supervised speech representation models," *arXiv preprint arXiv:2110.03504*, 2021.

[6] H. Liu, L. P. Garcia, 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. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 1489–1493.

[7] H. Liu, L. P. Garcia, X. Zhang, A. W. Khong, and S. Khudanpur, "Enhancing code-switching speech recognition with interactive language biases," *arXiv preprint arXiv:2309.16953*, 2023.

[8] H. Liu, L. P. Garcia, 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. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 2233–2237.

[9] 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. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 4766–4770.

[10] 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.

[11] 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.

[12] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5884–5888.

[13] 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.

[14] 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. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 3924–3928.

[15] 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. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 3894–3898.

[16] B. Yan, C. Zhang, M. Yu, S.-X. Zhang, S. Dalmia, D. Berrebbi, C. Weng, S. Watanabe, and D. Yu, "Joint modeling of code-switched and monolingual asr via conditional factorization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 6412–6416.

[17] B. Yan, M. Wiesner, O. Klejch, P. Jyothi, and S. Watanabe, "Towards zero-shot code-switched speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, pp. 1–5.

[18] 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.

[19] H. Adel, N. T. Vu, K. Kirchhoff, D. Telaar, and T. Schultz, "Syntactic and semantic features for code-switching factored language models," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 23, no. 3, pp. 431–440, 2015.

[20] 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. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2010, pp. 1986–1989.

[21] X. Shi, Q. Feng, and L. Xie, "The ASRU 2019 Mandarin-English code-switching speech recognition challenge: Open datasets, tracks, methods and results," *arXiv preprint arXiv:2007.05916*, 2020.

[22] R. Wanneroy, E. Bilinski, C. Barras, M. Adda-Decker, and E. Geoffrois, "Acoustic-phonetic modeling of non-native speech for language identification," in *Multilingual Interoperability in Speech Technology*, 1999.

[23] L. Zhou, J. Li, E. Sun, and S. Liu, "A configurable multilingual model is all you need to recognize all languages," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 6422–6426.

[24] C. Zhang, B. Li, T. N. Sainath, T. Strohman, and S.-Y. Chang, "UML: A universal monolingual output layer for multilingual asr," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, pp. 1–5.

[25] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[26] C.-Y. Hsieh, C.-L. Li, C.-k. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, and T. Pfister, "Distilling step-by-step! Outperforming larger language models with less training data and smaller model sizes," in *Findings of the Association for Computational Linguistics: ACL 2023*, Jul. 2023, pp. 8003–8017.

[27] C. Chen, Y. Hu, C.-H. H. Yang, S. M. Siniscalchi, P.-Y. Chen, and E. S. Chng, "Hyporadise: An open baseline for generative speech recognition with large language models," *arXiv preprint arXiv:2309.15701*, 2023.

[28] 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.

[29] 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. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 5036–5040.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE/CVF Conf. Comput. Vis. and Pattern Recognit.*, 2016, pp. 2818–2826.

[31] Y. Peng, Y. Liu, J. Zhang, H. Xu, Y. He, H. Huang, and E. S. Chng, "Internal language model estimation based language model fusion for cross-domain code-switching speech recognition," *arXiv preprint arXiv:2207.04176*, 2022.

[32] A. Hussein, S. A. Chowdhury, A. Abdelali, N. Dehak, A. Ali, and S. Khudanpur, "Textual data augmentation for Arabic-English code-switching speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2023, pp. 777–784.

[33] H. Yu, Y. Hu, Y. Qian, M. Jin, L. Liu, S. Liu, Y. Shi, Y. Qian, E. Lin, and M. Zeng, "Code-switching text generation and injection in Mandarin-English ASR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, pp. 1–5.

[34] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1877–1901.

[35] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. Int. Conf. Machine Learn.*, 2023, pp. 28 492–28 518.

[36] Y. Fathullah, C. Wu, E. Lakomkin, J. Jia, Y. Shangguan, K. Li, J. Guo, W. Xiong, J. Mahadeokar, O. Kalinli, C. Fuegen, and M. Seltzer, "Prompting large language models with speech recognition abilities," *arXiv preprint arXiv:2307.11795*, 2023.

[37] Y. Wang, Y. Wang, K. Dang, J. Liu, and Z. Liu, "A comprehensive survey of grammatical error correction," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 12, no. 5, pp. 1–51, 2021.

[38] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proc. Int. Conf. Learn. Representations*, 2022.

[39] S. Punjabi, H. Arsikere, Z. Raeesy, C. Chandak, N. Bhave, A. Bansal, M. Müller, S. Murillo, A. Rastrow, A. Stolcke, J. Droppo, S. Garimella, R. Maas, M. Hans, A. Mouchtaris, and S. Kunzmann, "Joint ASR and language identification using RNN-T: An efficient approach to dynamic language switching," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 7218–7222.

[40] Q. Wang and H. Li, "Text-derived language identity incorporation for end-to-end code-switching speech recognition," in *Proc. Conf. on Empirical Methods in Natural Language Processing, Sixth Workshop on Computational Approaches to Linguistic Code-Switching*, 2023.

[41] C. Myers-Scotton, *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press, 1997.

[42] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large

language models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 24 824–24 837.
[43] L. Q. Dixon, "Bilingual education policy in Singapore: An analysis of its sociohistorical roots and current academic outcomes," *International Journal of Bilingual Education and Bilingualism*, vol. 8, no. 1, pp. 25–47, 2005.
[44] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 3586–3589.
[45] 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. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 2613–2617.
[46] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.
[47] K. Praveen, B. Radhakrishnan, K. Sabu, A. Pandey, and M. A. B. Shaik, "Language identification networks for multilingual everyday recordings," in *Proc. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2023, pp. 4124–4128.
[48] A. Zeyer, R. Schlüter, and H. Ney, "Why does CTC result in peaky behavior?" *arXiv preprint arXiv:2105.14849*, 2021.
[49] X. Chen, Y. Y. Lin, K. Wang, Y. He, and Z. Ma, "Improving frame-level classifier for word timings with non-peaky CTC in end-to-end automatic speech recognition," in *Proc. ISCA Annu. Conf. Int. Speech Commun. Assoc.*, 2023, pp. 2908–2912.