Selective Hearing through Lip-reading

Zexu Pan, Ruijie Tao, Chenglin Xu, Member, IEEE, and Haizhou Li, Fellow, IEEE

Abstract—Speaker extraction algorithm emulates human’s ability of selective attention to extract the target speaker’s speech from a multi-talker scenario. It requires an auxiliary stimulus to form the top-down attention towards the target speaker. It has been well studied to use a reference speech as the auxiliary stimulus. Visual cues also serve as an informative reference for human listening. They are particularly useful in the presence of acoustic noise and interference speakers. We believe that the temporal synchronization between speech and its accompanying lip motion is a direct and dominant audio-visual cue. In this work, we aim to emulate human’s ability of visual attention for speaker extraction based on speech-lip synchronization. We propose a self-supervised pre-training strategy, to exploit the speech-lip synchronization in a multi-talker scenario. We transfer the knowledge from the pre-trained model to a speaker extraction network. We show that the proposed speaker extraction network outperforms various competitive baselines in terms of signal quality and perceptual evaluation, achieving state-of-the-art performance.

Index Terms—Multi-modal, target speaker extraction, time-domain, self-enrollment, speaker embedding, speech-lip synchronization.

I. INTRODUCTION

HUMANS have a remarkable ability to focus attention on a particular speech in the presence of multiple noise sources and competing background speakers [1]. Speaker extraction algorithm mimics human’s selective attention to extract only the target speaker’s speech in such an adverse acoustic environment, which is also referred to as the cocktail party problem [2].

The human brain has a limited capacity to process all sensory stimuli perceived. Instead, human cognition relies on an inherent attention mechanism to focus the neural resources according to the contingencies of the moment [3]. Such attention can be categorized into a top-down and bottom-up process. The top-down attention refers to the voluntary allocation of neural resources based on guidance such as a specific goal or reference. The bottom-up attention is driven by external stimuli that are salient because of their inherent properties relative to the background [4]. In this paper, we aim to mimic human’s visual attention during listening to solve the single-channel target speaker extraction problem. This is a non-trivial, but crucial step for other downstream tasks such as speaker diarization [5], speaker recognition [6], speaker verification [7], and speech recognition [8].

Humans separate a speech mixture by attending to the salient prosody of individual speech tracks. Speech separation represents one direction of research towards machine intelligence of such bottom-up auditory attention, which has recently seen major progress [9]–[17]. The formulation of speech separation usually requires the number of speakers to be known in advance, which limits the scope of real-world applications. The formulation of speaker extraction avoids such limitation, that uses a reference signal to form a top-down voluntary attention, which is also referred to as the attractor in this paper. It proves to be effective in dealing with an unknown number of speakers. Most speaker extraction algorithms make use of auditory stimuli such as a reference speech to characterize the speakers. The reference speech is encoded as a speaker embedding in advance. In this way, the speaker extraction algorithm extracts speech that sounds similar to the reference speech [18]–[28].

As the above-mentioned speaker extraction techniques use the speaker embedding as the attractor in the neural solutions, they require pre-enrollment of a reference speech, which sometimes are not available in practice, for example, when we would like a robot to pay auditory attention to a passer-by. Unlike reference speech, speech-related visual cues can be easily obtained during human-computer interactions. In this work, we aim to use visual auxiliary information for target speaker extraction when visual cues are available. In this way, the pre-enrolled reference speech is not required.

Human’s attention is multi-modal [29], through a variety of sensory systems such as audition, vision, touch, and smell. These multi-modal stimuli are processed in human nervous systems in an interactive manner, which is referred to as reentry [30]. Reentry describes the bidirectional exchange of signals along reciprocal axonal fibers linking two or more brain areas, that are temporally correlated, and can educate each other. The human brain has a cortex to process the audio and visual stimuli together [31]. In a cocktail party, visual cues are not corrupted by background noise, reverberation, and interference speech thus provide a robust attractor for auditory attention. The studies in neuroscience show that speech comprehension is improved by looking at the speaker in adverse conditions [32], [33], leveraging on long-term cross-modal temporal integration. Inspired by these findings, we aim to emulate human’s visual attention in a computational model for speaker extraction, named reentry model after the reentry theory by Edelman [30].

This research is supported by Programmatic Grant No. A18A2b0046 from the Singapore Government’s Research, Innovation and Enterprise 2020 plan (Advanced Manufacturing and Engineering domain); Human-Robot Interaction Phase 1 (Grant No. 192 25 00054), the National Research Foundation, Prime Minister’s Office, Singapore under the National Robotics Programme. This research is also funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy (University Allowance, EXC 2077, University of Bremen).

Zexu Pan is with the Integrative Sciences and Engineering Programme, and the Institute of Data Science, National University of Singapore, 119077 Singapore (e-mail: pan_zexu@u.nus.edu).

Zexu Pan, Ruijie Tao, Chenglin Xu, and Haizhou Li are with the Department of Electrical and Computer Engineering, National University of Singapore, 119077 Singapore (e-mail: ruijie.tao@u.nus.edu; xuchenglin28@gmail.com, haizhou.li@nus.edu.sg).

Haizhou Li is also with the University of Bremen, 28359 Bremen, Germany.
In human attentive listening, there are typically three types of visual cues, namely face-voice similarity [34], viseme-phoneme mapping [37], [38], and speech-lip synchronization [49]. Face-voice similarity explores the general correlation between a human face and its voice signature such as gender, age, and nationality. Viseme-phoneme mapping defines the relationship between the shapes of lips and mouth with respect to the sounds they represent, i.e., phonemes [41]. Among the visual cues, we advocate that the temporal synchronization between speech and lip movements would be the most informative, which will be the focus of this paper.

To emulate human’s visual attention during listening, we explore the speech-lip synchronization in a multi-talker setting with a pre-trained network, named speech-lip synchronization (SLSyn) network. We propose a self-supervised training strategy for the SLSyn network such that the learning of speech-lip synchronization could leverage abundant unlabeled training data that are in the same domain as the speaker extraction task. We transfer the pre-trained knowledge from the SLSyn network to the reentry model, which performs ‘hearing with ears and eyes’. The proposed network outperforms various competitive baselines and achieves state-of-the-art performance on VoxCeleb2 dataset mixtures.

The rest of the paper is organized as follows. In Section II we introduce the related works on target speaker extraction. In Section III we formulate the proposed reentry model. In Section IV we describe the experimental setup. In Section V we report the results. Finally, Section VI concludes the study.

II. Reference Signals for Speaker Extraction

Neural speaker extraction typically requires a reference signal to form the top-down attention towards the voice of the target speaker in a speech mixture. Studies in neuroscience suggest that either auditory [42] or visual stimulus [32], [33] can serve as a reference cue in human attentive listening.

A. Pre-enrolled auditory stimuli

Each speaker has a unique voice signature, which can be characterized by a fixed dimensional vector, referred to as speaker embedding, such as i-vector [43], x-vector [44], d-vector [45], and other similar feature representations [46]. It is common that a speaker embedding derived from the target speaker is employed to form the attractor. VoiceFilter [20] and TseNet [47] are such examples, where they pre-train a speaker encoder, which is then used to derive d-vector or i-vector from the reference speech.

The speaker encoder is typically pre-trained via a speaker verification task. As such speaker encoder is trained independently of the speaker extraction task, the d-vector and i-vector derived may not be the optimal representation for speaker extraction. To address this issue, some suggest jointly train the speaker encoder with the speaker extraction network [22], [48], [49]. The studies of SpEx+ [18] and SpEx+ [19] further the idea by introducing a multi-task learning strategy, to jointly train the speaker encoder with an additional speaker classification loss, thus benefiting from the improved speaker characterization [20], [47] as well as task-oriented optimization [22], [48], [49].

B. Self-enrolled auditory stimuli

Despite much success, neural speaker extraction with speaker pre-enrollment is not practical in some scenarios. This prompts us to look into solutions without the need for pre-enrolled reference speech. A recent study [50] on audio-visual speaker extraction seeks to use visual reference to replace audio reference. In this study, a visual reference is employed to extract the target speech first, which is subsequently encoded as a speaker embedding to serve as the reference whenever the visual cues are occluded. While such a two-pass mechanism improves in the absence of visual cues, the self-enrolled speaker embedding does not contribute when visual cues are adequately available. This suggests that the self-enrolled speaker embedding is not as informative as the visual reference, partly due to the fact that the two-pass mechanism is not jointly optimized.

MuSE [51] furthers the idea of visual reference in audio-visual speaker extraction. It self-enrolls the speaker’s voice signature from the intermediate extracted target speech on the fly. MuSE leverages the visual reference as well as the self-enrolled speaker embedding in an integrated manner, which shows that both of them actively contribute to the speaker extraction task. AVSE [52] is another method that self-enrolls the speaker’s voice signature directly from the mixture speech. AVSE is a frequency-domain implementation while MuSE takes a time-domain approach. This paper is motivated by the findings in the prior studies on the effective use of visual reference to self-enroll a speaker.

C. Face-voice correlation

It is known that there exist correlations between human face and voice, such as gender dependency, face ethnicity vs accent, body size and composition vs vocal characteristics. Prior studies [34], [35] have explored such general knowledge to effectively use visual stimuli for speaker extraction. They employ a speaker encoder that is pre-trained to learn the cross-modal correspondence, and to encode the face image into a speaker embedding.

By using such speaker embedding to serve as an attractor, the speaker pre-enrollment is not required. It is noted that speaker extraction systems that rely on such general correlation knowledge generally don’t outperform those with pre-enrolled auditory stimuli.

D. Viseme-phoneme correspondence

Phonemes are the smallest units of spoken language, and the visual equivalent are the visemes. Generally speaking, a viseme corresponds to a set of phonemes that have identical appearances on the lips. Therefore, one is able to associate a phoneme with a viseme class, but a viseme may map to multiple phonemes. The viseme-phoneme correspondence is usually trained from a visual speech recognition task, which aligns the input viseme sequence with the output phoneme sequence or word sequence [53], [54]. Some prior studies have effectively made use of such viseme-phoneme correspondence in speaker extraction [37], [38], [55]. They transfer part of the pre-trained visual speech
recognition network as the visual encoder of the speaker extraction network, where the visual encoder encodes the input viseme sequence to a viseme embedding sequence. The viseme embeddings are used to form the attention attractor. As the visual encoder is optimized for the phoneme recognition task, therefore, not the best for speech reconstruction, we are prompted to look into a better way to train the visual encoder for speaker extraction. Furthermore, the training of visual speech speech recognizer requires video data with text transcription, which is scarce. We are motivated to leverage the audio-visual information that exists in abundantly available parallel audio-visual data for self-supervised learning, without the need for text transcription.

Motivated by a study that face embeddings encoded by face recognition models can recover facial expressions [56], ‘looking to listen at the cocktail party’ [57] employs viseme embeddings which are encoded by a face recognition model as the attractor for speaker extraction. However, the face embeddings are trained to discriminate faces. It is neither an optimal representation for viseme-phoneme mapping nor audio-visual synchronization.

E. Audio-visual synchronization

Some studies have implemented the use of audio-visual synchronization cues for speaker extraction [39], [40], where they encode the raw visual frames or facial landmarks into a sequence of visual embeddings as the reference signal.

Others [58], [59] further the idea by pre-training the visual encoder of speech separation network on audio-visual synchronization tasks. [58] pre-trains the visual encoder on the scene synchronization task which finds that scene motion is temporally synchronized with audio events. [59] pre-trains the visual encoder on the object synchronization task which detects the object that is responsible for the sound in a video. All studies point to the direction that the scene and object synchronization features encoded by the visual encoder could be employed as a visual attractor in speech separation networks. However, the audio-visual synchronization for speech to lips in a multitalker scene is not as prominent as that for whistling sound to a train. Furthermore, the audio-object synchronization features are not pre-trained to disentangle overlapped speech.

The success of the effective use of audio-visual synchronization cues prompts us to look into speech-lip synchronization that is a more direct and informative cue for speaker extraction. Humans have the experience to improve listening by lip-reading when someone is talking. In this paper, we would like to focus on the use of speech-lip synchronization cues for speaker extraction.

III. AUDIO-VISUAL SPEAKER EXTRACTION NETWORK

We propose an audio-visual speaker extraction network using speech-lip synchronization cues, as depicted in Fig. 1, that is called reentry model. Let $x(\tau)$ be a multi-speaker speech waveform in the time-domain, which consists of the target’s speech $s(\tau)$ and other interference speech $b_i(\tau)$,

$$x(\tau) = s(\tau) + \sum_{i=1}^{I} b_i(\tau)$$

(1)

We would like to extract $\hat{s}(\tau)$ that approximates $s(\tau)$.

In the reentry model, 1) the speech encoder transforms $x(\tau)$ into a sequence of spectrum-like frame-based embeddings $X(\tau)$, referred to as mixture speech embeddings. In this paper, a variable with $\tau$ as the index represents a sequence of samples in the time-domain, while a variable with $t$ as the index represents a sequence of frame-based embeddings. 2) The attractor encoder encodes $x(\tau)$ and its accompanying video $v(t)$ into a sequence of speech-lip synchronization embeddings $V(t)$ to form the attractor signal. 3) The speech encoder encodes the intermediate estimated speech embeddings $\hat{S}(\tau)$ into speaker embedding $A^\prime$, which represents the attended speaker’s voice signature. 4) The speaker extractor takes in $V(t)$ and $A^\prime$ as top-down attention attractors to estimate a mask $M(t)$ for $X(t)$ which only let pass the target speaker. The extracted speech embeddings $\hat{S}(t)$ is estimated by element-wise multiplication between the estimated mask $M(t)$ and the mixture speech embeddings $X(t)$:

$$\hat{S}(t) = X(t) \otimes M(t)$$

(2)

5) Finally, the speech decoder transforms the extracted speech embeddings $\hat{S}(t)$ into a time-domain waveform $\hat{s}(\tau)$.

This study is a departure from either the time-domain speaker extraction network [38] named TDSE in this paper or MuSe [51] which the attractor encoder is trained in a supervised manner. We propose a self-supervised pre-training strategy for the proposed attractor encoder.

Fig. 1. The proposed audio-visual speaker extraction network, named reentry model. The attractor encoder derives feature representations for speech-lip synchronization; the speaker extractor produces a filtering mask for the target speaker dynamically; the speaker encoders characterize the target speaker and provide the top-down attention to the speaker extractor. The symbol ⊠ represents the concatenation of features along channel dimension in convolutional layers; the symbol ⊙ represents element-wise multiplication.
A. Self-supervised speech-lip synchronization learning

Self-supervised learning has been well studied in audio-visual synchronization learning, where a network learns from positive and negative samples easily derived from found data. The positive samples could be the videos with the original soundtrack, while the negative samples are the videos with a random time-shifted soundtrack [58], [59]. During learning, one common approach is to encode audio and visual signals into embedding sequences separately, and compare the similarity of the two embedding sequences [59]–[61]. It makes the decision on short video segments, and doesn’t make good use of the temporal correlation between modalities. Another idea is to fuse the audio and visual embedding sequences, that are extracted separately, through an audio-visual network such as Convolutional Neural Network (CNN) [58] or long short-term memory (LSTM) [62] to capture their temporal dependencies.

Motivated by the second idea, we propose a speech-lip synchronization (SLSyn) network to detect the presence of synchronization between a multi-talker speech mixture and a video sequence of face bounding boxes. As the SLSyn network is optimized for speech-lip synchronization detection, it is expected to derive speech-lip synchronization embedding that characterizes the audio-visual joint events. The SLSyn network is pre-trained with multiple speakers to be a speaker-independent model.

As shown in Fig. 2 (a), the SLSyn network consists of an audio front-end to extract audio features from the speech mixtures, a visual front-end to extract visual features from the target’s face track, and an audio-visual back-end to fuse the concatenated audio and visual features for an utterance level binary classification. We design a visual front-end with a 3D convolutional layer and a ResNet to encode raw images into visemes similar to [63], representing the appearance of the lips. As temporal convolutional neural network (TCN) is effective in capturing long-term dependencies [10] in speech processing [64], [65], we adopt TCN structure as the audio front-end. The audio and visual features are concatenated at the frame level, and taken by the TCN-based audio-visual back-end.

Let \( v(t) \) be the input face video sequence, and \( x(\tau) \) the speech signal. The activations at the intermediate layers of the SLSyn network represent different levels of abstraction. We study three different feature representations as the speech-lip synchronization embeddings, namely \( \text{sync}_1(t) \), \( \text{sync}_2(t) \), and \( \text{sync}_3(t) \), as shown in Fig. 2 (a). The three feature representations have the same time-resolution as the target’s face video sequence.

As illustrated in Fig. 2 (b), the original videos with speech-lip synchronized soundtracks serve as the positive samples, while videos with randomly shifted soundtracks serve as the negative samples. An additional interference speech is added to both positive and negative samples to simulate the cocktail party scenarios.

We minimize the following binary cross-entropy (BCE) loss:

\[
\mathcal{L}_{BCE} = -y \log(\hat{y}) - (1 - y) \log(\hat{y})
\]

where \( y \in [0, 1] \) indicates whether the input video has shifted soundtrack while \( \hat{y} \) is the predicted probability of the video and soundtrack being synchronized at the end of the SLSyn network.

This learning is similar to noise-contrastive estimation [66], where the noisy samples in SLSyn network training are asynchronous videos. As human-annotated labels are not needed, the pre-training can be done on abundantly available videos, also in the same domain as the speaker extraction algorithms to minimize the domain mismatch.

B. Speaker extraction using speech-lip synchronization cues

We now formulate the reentry model, as illustrated in Fig. 1.

The reentry model consists of a speech encoder, an attractor encoder, a speaker extractor, a number of speaker encoders, and a speech decoder. The speaker extractor consists of an interlace structure between a number of speaker encoders and speaker extraction modules.

1) Speech encoder: The speech encoder takes a time-domain approach [10], [12], [18], [38] to encode input speech waveform into a spectrum-like speech embedding sequence. It performs a 1D convolution on \( x(\tau) \), with a channel size \( N \), kernel size \( L \), and stride \( L/2 \).

\[
X(t) = \text{Conv1D}(x(\tau), N, L, L/2)
\]
where \( N \) is also the speech embedding dimension. As \( x(\tau) \) is a time-domain signal, the 1D convolution behaves like a frequency analyzer [10].

2) Speech decoder: The speech decoder is an inverse function of the speech encoder, that takes the extracted speech embeddings \( \hat{S}(t) \) as input and produces a time-domain waveform as output. We formulate a speech decoder as a linear layer followed by an overlap and add \( O n A(\cdot) \) operation to reconstruct \( \hat{s}(\tau) \) from \( \hat{S}(t) \) [67].

\[
\hat{s}(\tau) = O n A(\text{Linear}(\hat{S}(t), N, L), L/2)
\]

where the \text{Linear}(\cdot) layer has input and output feature dimension of \( N \) and \( L \) respectively, the overlap and add operation has a frame shift of \( L/2 \).

3) Attractor encoder: The attractor encoder seeks to encode the synchronized sequence of lip images \( v(t) \) with mixture speech waveform \( x(\tau) \) into a sequence of embeddings \( V(t) \) representing speech-lip synchronization. The SLSyn network is trained just to do that.

We adopt part of the pre-trained SLSyn network, followed by several adaptation layers to form the attractor encoder. As the speech-lip synchronization embeddings extracted from the pre-trained SLSyn network, namely \( \text{sync}_1(t) \), \( \text{sync}_2(t) \), and \( \text{sync}_3(t) \), are not optimized for speaker extraction directly, they have different properties than those from the speaker extractor [68]. The adaptation layers adapt the speech-lip synchronization embeddings towards the speech extractor task. We follow [37], [38] to design a stack of TCNs in the adaptation layers.

4) Speaker extractor: Receptive masks have been well studied in source separation literature to mimic human’s selective attention, such as ideal binary mask [69], ideal ratio mask [70], ideal amplitude mask [71], wiener-filter like mask [72], and phase sensitive mask [72]. We adopt the masking methods [10], [19], [38] to estimate a receptive mask \( M(t) \) for the speech of the target speaker as shown in Fig. 1.

Similar to [19], [18], [38], we use the TCN structure for the speaker extractor to capture the long-range dependency of the speech signal. The speaker extractor consists of \( R \) repeated stacks of TCNs in a speaker extraction pipeline. We illustrate the reentry model with \( R = 4 \) in Fig. 1. An intermediate mask \( M^{\tau}(t) \) is estimated after the \( \tau \)th stack. Each stack consists of \( B \) TCNs with an exponentially growing dilation factor \( 2^b \) (\( b = 0, ..., B - 1 \)). With the pipelined stacks, the receptive field of the network increases exponentially, and better models the long-term temporal dependencies of speech signals [10]. Hence, the masks produced by the TCN stacks are expected to be progressively refined along the speaker extraction pipeline.

The reentry theory suggests that in the human brain, temporally correlated signals educate each other through bidirectional exchanges. Following the reentry theory, we propose to make use of the interactive evidence between the speech and lips on the fly as we decode the target speech.

The speaker extractor takes the mixture speech embeddings \( X(t) \) as input, and top-down attention cues as the condition, to extract clean speech from the target speaker. Besides the speech-lip synchronization cue \( V(t) \), we also consider the self-enrolled speaker embedding \( A^r \) as another top-down attention cue. The top-down attention cues modulate with the input speech mixture to estimate the receptive masks. The study in SpEx [18] shows that concatenating the attention stimulus at the start of every TCN stack effectively guides the mask estimation. Bring the idea forward to the audio-visual context, we concatenate \( A^r \) and \( V(t) \) with \( M^{\tau}(t) \) to form the input of each TCN stack, except for the first TCN stack which we concatenate \( V(t) \) with \( X(t) \) as input. The \( A^r \) and \( V(t) \) are up-sampled along the time dimension to match the temporal resolution of \( M^{\tau}(t) \) before concatenation. We will discuss the self-enrolled speaker encoders next.

5) Speaker encoders: An speaker encoder derives an utterance level speaker embedding \( A^r \) a.k.a self-enrolled speaker embedding from the intermediate estimated speech embeddings \( \hat{S}^r(t) \) after the \( r \)th TCN stack in the speaker extraction pipeline, which

\[
\hat{S}^r(t) = X(t) \otimes M^r(t)
\]

Since \( M^r(t) \) becomes available only after the first TCN stack, we apply the speaker encoders starting from the second TCN stack. We concatenate \( A^r \) with \( M^r(t) \) in the same way as \( V(t) \). As \( \hat{S}^r(t) \) varies along the speaker extraction pipeline, we design \( R - 1 \) speaker encoders of the same architecture, but with distinctive weights that are trained independently of each other, as opposed to weight sharing in SpEx [18].

C. Multi-task learning

To ensure that the speaker embedding \( A^r \) is encoded for the purpose of speaker extraction, we train the \( R - 1 \) speaker encoders together with the speaker extractor network, with two training objectives, i.e., the scale-invariant signal-to-noise ratio (SI-SDR) [73] to measure the signal quality, and the cross-entropy (CE) loss for speaker classification.

The loss \( \mathcal{L}_{SI-SDR} \) measuring signal quality is shown in Eq. 7, which is applied to the output of the reentry model between the extracted speech \( \hat{s}(\tau) \) and clean speech \( s(\tau) \). We omit the subscript \( r \) for brevity.

\[
\mathcal{L}_{SI-SDR} = -10 \log_{10} \left( \frac{||\hat{s}\cdot s > \tau||^2}{||s - \hat{s}\cdot s > \tau||^2} \right)
\]

The loss \( \mathcal{L}_{CE} \) that accounts for speaker classification is shown in Eq. 8, which is applied to every speaker encoder’s output \( A^r \) of the \( R - 1 \) speaker encoders.

\[
\mathcal{L}_{CE} = - \sum_{c=1}^{C} p[c] \log \hat{p}[c]
\]

where \( p[c] = 1 \) for target speaker, and \( p[c] = 0 \) otherwise, \( \hat{p}[c] = \text{softmax}(W^r A^r)[c] \) is the predicted probability for speaker \( c \). \( C \) is the number of speakers in the dataset; \( W^r \) is a learnable weight matrix specific to each encoder for speaker classification.

The overall loss \( \mathcal{L}_{total} \) is defined as:

\[
\mathcal{L}_{total} = \mathcal{L}_{SI-SDR} + \gamma \sum_{r=1}^{R-1} \mathcal{L}_{CE}
\]

where \( \gamma \) is a scaling factor.
Now that each speaker encoder is constrained with the speaker classification loss, it is expected to derive a speaker embedding $A'$ that best represents the target speaker’s voice characteristics, helping the speaker extractor in the next TCN stack. Besides, the gradient of the speaker classification loss is passed from speaker encoders to stacks of TCNs in the multi-task learning framework. Hence, the speaker extractor is also trained to extract the target’s speech containing the target speaker’s voice characteristics, besides the signal extraction quality. Therefore, the speaker self-enrollment is progressively refined with the speaker extractor with the interlaced structure, ensuring the convergence of speaker extraction.

It is worth noting that $A'$ is derived from mixture speech and visual signals without the supervision of speaker identity at run-time inference. Therefore, $A'$ is also called self-enrolled speaker embedding.

D. Relationship with MuSE

MuSE [51] employs a pre-trained visual speech recognition network as the attractor encoder and exploits the viseme-phoneme correspondence to form the focus attention. The main difference between the reentry model and MuSE lies in two aspects. 1) the reentry model employs a self-supervised pre-training scheme for the SLSyn network that doesn’t require text transcription of the soundtrack as the visual speech recognition network does. This brings us multiple advantages. We can now easily adapt the SLSyn network on abundantly available domain data to reduce domain mismatch. 2) the SLSyn network learns the speech-lip synchronization rather than the viseme-phoneme correspondence. The latter is not a direct cue for overlapped speech disentanglement and signal reconstruction.

While the reentry model is different from the MuSE framework, it shares some ideas that are studied in MuSE and other related work, that is to exploit a self-enrollment strategy, and to employ a multi-stage progressive refinement pipeline for speaker extraction. We expect that the progressive refinement architecture will outperform a single-stage extraction [72]. The findings in this paper certainly reinforce this line of thought.

IV. Experimental Setup

A. Experiment data

1) Speaker extraction: We simulate a two-speaker mixture dataset (VoxCeleb2-2mix) and a three-speaker mixture dataset (VoxCeleb2-3mix) from the VoxCeleb2 dataset [75] for a series of system evaluation. VoxCeleb2 dataset is an audio-visual dataset that contains over 1 million utterances for 6,112 celebrities, extracted from YouTube videos, and is pre-processed with face-tracking algorithms.

We remove utterances shorter than 4 seconds, and randomly select 40,000 clean utterances from 800 speakers in the original VoxCeleb2 train set to create our train set (20,000 utterances); we randomly select another 8,000 clean utterances from the same 800 speakers in the original train set to create our validation set (5,000 utterances); we use the 36,237 clean utterances from 118 speakers in the original test set to form our test set (3,000 utterances). The interference speech is mixed with the target speech at a random Signal-to-Noise (SNR) ratio between 10dB to −10dB. The long utterance is truncated to the length of the short utterance when creating a speech mixture. The audios are sampled at 16 kHz, the videos are synchronized with the audios and sampled at 25 FPS.

The train set is used for network training, the validation set is used for optimizing network configurations. The utterances and speakers in the test set do not overlap with those in the train set and validation set, which allows us to perform speaker-independent experiments using the test set.

2) Synchronization detection: To pre-train the speaker-independent speech-lip synchronization network, we create a multi-talker dataset (VoxCeleb2-sync) from the VoxCeleb2 dataset. We create 4 million video clips to form a train set, and 40 thousand video clips to form a validation set. They range from 1 to 4 seconds. A quarter of the video clips are with clean speech and three-quarters of them are with simulated two-speaker speech mixtures. The simulation follows the same protocol as that of the VoxCeleb2-2mix.

The video clips with the original soundtrack are the positive samples that have synchronized speech and lip movements. The negative samples have a randomly shifted soundtrack by 0.2 to 1 seconds relative to the visual sequence. Besides VoxCeleb2-sync, we create a smaller version named VoxCeleb2-sync-s, which contains 10% of the video clips from VoxCeleb2-sync; and a clean dataset named VoxCeleb2-sync-c, which is of the same size as VoxCeleb2-sync, but all are clean speech utterances from VoxCeleb2.

B. Training of reentry model

Both the SLSyn network and reentry model are implemented in PyTorch, and optimized by the Adam optimizer [76], with an initial learning rate of 0.001. Their network architecture and parameters can be found in the Appendix.

For the SLSyn network, the learning rate is decreased by 4% for every epoch, the training stops when the best validation loss does not improve for 4 consecutive epochs. For the reentry model, the learning rate is halved when the best validation loss does not improve for 6 epochs consecutively, the training stops when the best validation loss does not improve for 10 epochs consecutively. All models are trained on 2 Tesla 32GB V100 GPUs. During the reentry model training, the utterances are truncated to 6 seconds to fit into the GPU memory, during inference, the full utterance is evaluated.

The overall training is carried out in three stages. In the first stage, we pre-train the SLSyn network to form the attractor encoder. In the second stage, we train the reentry model, which uses the pre-trained attractor encoder without updating its weights. In the last stage, we find-tune the attractor encoder together with the reentry model using the re-initialized optimizer. We will justify the 3-stage strategy through an ablation experiment.

C. Baselines

Time-domain speaker extraction algorithms usually outperform their frequency-domain counterparts. MuSE [51] is a

1The code is available at https://github.com/zexupan/reentry.
time-domain technique that reports the start-of-the-art performance in audio-visual speaker extraction, and is built on TDSE [38]. We implement the two models as the reference baselines.

1) TDSE: TDSE [38] has a network component and architecture similar to the reentry model, except that the former has neither the speaker encoders for top-down auditory attention nor the speaker classification task, which is designed to train the speaker encoders. In addition, TDSE uses a pre-trained VSR network as the attractor encoder. The implementation of TDSE-O follows the original architecture in the paper. We also implement TDSE-I as an improved version for a fair comparison with the reentry model. The improvements include, a) the TDSE-I replaces the batch normalization in TDSE-O with layer normalization, b) it concatenates the visual attractors to every stack of TCNs, just like what the reentry model does, instead of the second stack only, and c) the attractor encoder is fine-tuned similarly to the reentry model.

2) MuSE: Like TDSE, MuSE [51] uses a VSR pre-trained network as the attractor encoder. We implement the original architecture, denoted as MuSE-O, and an improved version MuSE-I, with layer normalization and attractor encoder fine-tuning. MuSE-O takes the intermediate estimated speech embeddings $\hat{S}(t)$ and visual cues $V(t)$ as input to the speaker encoders, and applies a speaker encoder to the start of every TCN stack, while MuSE-I and reentry model only take $\hat{S}(t)$ as input to the speaker encoders starting from the second TCN stack similarly to the reentry model.

3) Visual Speech Recognition: TDSE and MuSE both use an attractor encoder pre-trained on the visual speech recognition task [6]. The attractor encoder generates $v\text{sr}_v(t)$, which is the output of the visual front-end of the VSR-LSTM model [63]. The visual front-end consists of a convolutional 3D and a ResNet 18 layer. In a visual speech recognition study [77], this visual front-end is followed by 12 transformer layers to form a VSR-TRAN model as illustrated in Fig. 3, where $v\text{sr}_\text{enc}(t)$ and $v\text{sr}_\text{cpp}(t)$ are the outputs of the 6th and last layer of the transformer. $v\text{sr}_\text{cpp}(t)$ represents the character posterior probability at the frame level. We study the three features, namely $v\text{sr}_v(t)$, $v\text{sr}_\text{enc}(t)$ and $v\text{sr}_\text{cpp}(t)$, which represent different levels of abstraction of the visual cue in our experiments.

D. Evaluation metrics

We use scale-invariant signal-to-noise ratio improvement (SI-SDRi); signal-to-noise ratio improvement (SDRi); perceptual evaluation of speech quality improvement (PESQi) [78].

Fig. 3. Three features derived from a visual speech recognition model.

V. Experiments

A. Visual speech recognition vs speech-lip synchronization

We first compare the two pre-training models, namely visual speech recognition (VSR) [53] and our speech-lip synchronization (SLSyn) network, as reported in Table 1. As VSR is a challenging task with limited training data, VSR-TRAN has a high word error rate of 61.8% on the LRS2 dataset.

For synchronization detection, the accuracy is 86.6% with 270 hours of training data (VoxCeleb2-sync-s). If 2,700 hours of data is used (VoxCeleb2-sync), the accuracy goes up to 94.9%. If the dataset consists of clean speech only (VoxCeleb2-sync-c), the accuracy goes up to 96.8%. The number (no.) of network parameters for the SLSyn network is less than the VSR network due to the simplicity of the task. In VoxCeleb2-sync, negative samples are target shifted speech. To ensure the SLSyn network is not learning shift instead of synchronization, we conduct an additional negative sample test to classify the talking face track with only the interference speaker’s speech. The SLSyn network achieves the classification accuracy of 96.0%, showing that the SLSyn network is learning the speech-lip synchronization information.

We evaluate the reentry model on VoxCeleb2-2mix with different pre-trained attractor encoders in Table 1. In Section 4, we omit the subscript (t) for visual attractors for brevity. With the VSR features, the lower level of abstraction is further away from the decision layer, and represents more the visual content than the phonetic information. We observe that the lower the feature abstraction level, the better the extracted signal quality, with $v\text{sr}_v$ having better results than $v\text{sr}_\text{enc}$ and $v\text{sr}_\text{cpp}$. The results suggest that the viseme-phoneme mapping objective, that optimizes $v\text{sr}_\text{cpp}$, is not the best for speaker extraction.

With the SLSyn features, the higher level of abstraction is near the decision layer, and represents more the speech-lip synchronization/non-synchronization decision than the audio-visual content. We observe that the higher the abstraction level gives the better signal extraction quality, with $\text{sync}_{c3}$ having better results than $\text{sync}_{c2}$ and $\text{sync}_{c1}$. When $\text{sync}_{c3}$ is trained from VoxCeleb2-sync-s with a similar amount of data as VSR, it outperforms the best VSR feature $v\text{sr}_v$ by 0.55dB in terms of SI-SDRi. The best synchronization feature $\text{sync}_{c3}$ outperforms the best VSR feature $v\text{sr}_v$ by 0.69dB in terms of SI-SDRi, showing the advantage of the speech-lip synchronization feature. Unless mentioned otherwise, the reentry model reported in this paper uses $\text{sync}_{c3}$ for the speech-lip synchronization feature representation.

Among the synchronization features, the visual-only feature $\text{sync}_{c1}$ lacks behind audio-visual features $\text{sync}_{c2}$ and $\text{sync}_{c3}$ by about 0.5dB in terms of SI-SDRi, showing the importance of speech in synchronization cue. When $\text{sync}_{c3}$ is trained
We report the evaluation of the pre-trained (PT) model VSR-TRAN in terms of word error rate (WER), and that of the SLSyn network in terms of binary classification accuracy (ACC). We also evaluate the reentry model on VoxCeleb2-2mix with different top-down attention cue (Att). The validation set (VAL.) and test set SI-SDRi are reported with (W) and without (WO) the attractor encoder fine-tuning. The no. of network parameters (Params) is reported in million (M).

| PT model | PT Dataset | Hours | WER (%) | Acc. (%) | Att | Val. SI-SDRi (dB) W/O | Test SI-SDRi (dB) W/O | Val. SI-SDRi (dB) W | Test SI-SDRi (dB) W | Params (M) |
|----------|------------|-------|---------|----------|-----|------------------------|------------------------|----------------------|----------------------|------------|
| VSR-LSTM | LRW [54]   | 160   | -       | -        | -   | 12.29                  | 11.88                  | 12.30                | 11.91                | 20.1       |
| VSR-TRAN | LRS2 [77]  | 200   | 61.8    | -        | -   | 12.07                  | 11.61                  | 12.12                | 11.73                | 75.1       |
|          | VoxCeleb2-sync | 2700 | -       | 94.9     | -   | 12.17                  | 11.70                  | 12.27                | 11.80                | 16.0       |
| SLSyn    |            |       |         |          | sync1 | 12.43                  | 12.14                  | 12.70                | 12.34                | 18.5       |
|          |            |       |         |          | sync2 | 12.58                  | 12.34                  | 13.01                | 12.60                | 18.8       |
| SLSyn    |            |       |         |          | sync3 | 12.27                  | 12.00                  | 12.87                | 12.46                | 18.8       |
| SLSyn    |            |       |         |          | sync3 | 12.33                  | 11.91                  | 12.66                | 12.26                | 18.8       |

**TABLE II**

We study the training strategy for reentry model that involves the attractor encoder in five experiments. In all experiments, we train the reentry model of the same architecture with different variants of the proposed training strategy on VoxCeleb2-2mix, and report the SI-SDRi for the validation and test sets. Init denotes the use of pre-trained attractor decoder; Fix denotes freezing the pre-trained attractor decoder; FT denotes fine-tuning of the attractor encoder.

| Att | Init | Fix | FT | Val. SI-SDRi (dB) | Test SI-SDRi (dB) |
|-----|------|-----|----|-------------------|-------------------|
| Expt 1 | checks | N/A | N/A | 12.27             | 11.92             |
| Expt 2 | checks | N/A | N/A | 12.42             | 12.02             |
| Expt 3 | checks | checks | N/A | 12.58             | 12.34             |
| Expt 4 | checks | checks | checks | 13.01             | 12.60             |
| Expt 5 | face image | checks | checks | checks | 3.29             | 3.00             |

We perform an ablative study to justify the proposed 3-stage training strategy. In Table II, Expt 1 represents a training strategy in which we train the entire reentry model from scratch without any pre-training; Expt 2 represents a training strategy that uses the pre-trained SLSyn network to initialize the attractor encoder, then performs a full reentry model training; Expt 3 represents a variant of Expt 2 that uses the pre-trained SLSyn network to initialize the attractor encoder, then performs a full reentry model training but freezing the attractor encoder; Expt 4 represents the proposed 3-stage training strategy, that is to initialize the attractor encoder with pre-trained SLSyn network, then fix the attractor encoder for model training, finally fine-tune the attractor encoder and the rest of the reentry model together. As the sync3 feature from the SLSyn network has the best representation, we adopt sync3 to justify our training strategy for the reentry model.

As shown in Table II, the proposed 3-stage training strategy (Expt 4) provides an SI-SDRi improvement of 0.68dB over that without pre-training (Expt 1), which suggests that the reentry model benefits from the knowledge learned from the pre-training step. The proposed 3-stage training strategy (Expt 4) also outperforms that with a simple initialization (Expt 2) at retaining the transferred knowledge, with an SI-SDRi improvement of 0.58dB. This could be because, during the first few epochs of speaker extractor training, the loss is high. In Expt 2, such high loss back-propagates and adversely affects the integrity of the attractor encoder. We observe in Expt 3, that by fixing the pre-trained attractor encoder, we gain 0.32dB of SI-SDRi, and by fine-tuning the entire reentry model resulting from Expt 3, we gain another 0.26dB of SI-SDRi.

We note that the face-voice association can also be used as the top-down attention cue for speaker extraction [34]. We would like to investigate if the attractor encoder in the reentry model indeed benefits from the face-voice association or the speech-lip synchronization. In Expt 5, we first randomly pick a face image from the video sequence and repeat the same image to form an image sequence. We then use the image sequence in place of the video sequence as the top-down attention cue of the reentry model, and report an SI-SDRi of 3.00dB in Table II which is far behind that of Expt 4. While the attractor encoder seems to take advantage of the face-voice association cue, it mainly benefits from the speech-lip synchronization cue, as most of SI-SDRi gain comes from the latter.

**B. Study of training strategy**

We perform an ablative study to justify the proposed 3-stage training strategy. In Table II, Expt 1 represents a training strategy in which we train the entire reentry model from scratch without any pre-training; Expt 2 represents a training strategy that uses the pre-trained SLSyn network to initialize the attractor encoder, then performs a full reentry model training; Expt 3 represents a variant of Expt 2 that uses the pre-trained SLSyn network to initialize the attractor encoder, then performs a full reentry model training but freezing the attractor encoder; Expt 4 represents the proposed 3-stage training strategy, that is to initialize the attractor encoder with pre-trained SLSyn network, then fix the attractor encoder for model training, finally fine-tune the attractor encoder and the rest of the reentry model together. As the sync3 feature from the SLSyn network has the best representation, we adopt sync3 to justify our training strategy for the reentry model.

As shown in Table II, the proposed 3-stage training strategy (Expt 4) provides an SI-SDRi improvement of 0.68dB over that without pre-training (Expt 1), which suggests that the reentry model benefits from the knowledge learned from the pre-training step. The proposed 3-stage training strategy (Expt 4) also outperforms that with a simple initialization (Expt 2) at retaining the transferred knowledge, with an SI-SDRi improvement of 0.58dB. This could be because, during the first few epochs of speaker extractor training, the loss is high. In Expt 2, such high loss back-propagates and adversely affects the integrity of the attractor encoder. We observe in Expt 3, that by fixing the pre-trained attractor encoder, we gain 0.32dB of SI-SDRi, and by fine-tuning the entire reentry model resulting from Expt 3, we gain another 0.26dB of SI-SDRi.

We note that the face-voice association can also be used as the top-down attention cue for speaker extraction [34]. We would like to investigate if the attractor encoder in the reentry model indeed benefits from the face-voice association or the speech-lip synchronization. In Expt 5, we first randomly pick a face image from the video sequence and repeat the same image to form an image sequence. We then use the image sequence in place of the video sequence as the top-down attention cue of the reentry model, and report an SI-SDRi of 3.00dB in Table II which is far behind that of Expt 4. While the attractor encoder seems to take advantage of the face-voice association cue, it mainly benefits from the speech-lip synchronization cue, as most of SI-SDRi gain comes from the latter.

**C. Study of speaker encoders**

We study the self-enrolled speaker encoders to justify their role through experiments reported in Table III.

The reentry model has the same architecture as the reentry model except that the former has neither the speaker encoders nor the speaker classification task, which is designed to train the speaker encoders. In the speaker extractor in Fig. I, the interlaced architecture between the speaker encoders and the TCN stacks is designed to estimate the mask $M(t)$. With reentry, we would like to examine the contribution of the speaker encoders.

As for the reentry model, we evaluate multiple setups by varying the scaling factor $\gamma$. The $\gamma$ weights the contributions between speaker classification and speaker extraction during the multi-task learning. With $\gamma = 0$, we ignore the speaker classification task during the training of speaker encoders and obtain an SI-SDRi of 12.27dB, which is similar to that of reentry. The results suggest that the $R - 1$ speaker encoders are contributing only when they are trained under a speaker classification supervision in multi-task learning. When $\gamma = 0.005$, the reentry model achieves the best SI-SDRi of
TABLE III
A COMPARISON AMONG THE reentry MODEL, AND ITS VARIANTS ON VoxCeleb2-2mix. R IS THE NO. OF TCN STACKS IN THE SPEAKER EXTRACTOR PIPELINE. ’SHAREd WEIGHTS’ DENOTES A NETWORK ARCHITECTURE WHERE THE WEIGHTS IN R – 1 SPEAKER ENCODERS ARE SHARED. ’WITH A∗’ DENOTES THE USE OF THE SELF-ENROLLED SPEAKER EMBEDDING A∗ AS PART OF THE TOP-DOWN ATTENTION CUE.

| Model   | γ | R  | Shared Weights | With A∗ | Val. SI-SDRi (dB) | Test SI-SDRi (dB) |
|---------|---|----|----------------|---------|-------------------|------------------|
| reentry | 0.5| 4  | N/A            | N/A     | 12.51             | 12.03            |
| reentry | 0.1| 4  | N/A            | N/A     | 12.92             | 12.51            |
| reentry | 0.05| 4 | N/A            | N/A     | 12.89             | 12.45            |
| reentry | 0.05| 1 | N/A            | N/A     | 12.99             | 12.52            |
| reentry† | 0.005| 2 | ×              | ×       | 11.07             | 11.13            |
| reentry‡ | 0.005| 3 | ×              | ×       | 11.87             | 11.90            |
| reentry‡‡ | 0.005| 4 | ×              | ×       | 13.01             | 12.60            |
| reentry* | 0.001| 4 | ×              | ×       | 12.90             | 12.51            |
| reentry** | 0  | 4  | N/A            | N/A     | 12.68             | 12.27            |

12.00dB, which represents 0.33dB improvement over that for γ = 0. The results clearly suggest that the speaker encoders benefit from multi-task learning.

We also present reentry* as an upper-bound model, where the speaker encoders take clean target’s speech as input instead of the intermediate speech. reentry† delivers an SI-SDRi of 13.02dB. The reentry model is only 0.42dB lacking behind the upper-bound.

Furthermore, we build reentry**, which uses a speaker embedding extracted from the clean target’s speech with a pre-trained speaker verification model. The speaker extractor takes such speaker embedding as input instead of speaker embedding estimated on the fly. Despite having clean speech as input, reentry** only attains an SI-SDRi of 12.72dB. This suggests that the independently trained speaker encoders don’t work the best for speaker extraction, thus once again justifies the proposed joint training between the speaker encoders and the speaker extraction network.

D. Study of speaker extractor

We consider that the speaker encoders and speaker classification task may jointly contribute to the speaker extraction in two ways. First, A∗ provides the top-down attention cues for speaker extraction. Second, the speaker classification task supervises the training of speaker encoders and TCN stacks to ensure the output speech carries the target speaker identity. The question is in what way the speaker encoders with speaker classification task contribute to the extracted speech quality.

To answer the question, we implement a model, denoted as reentry†, which we alter the standard reentry model by keeping the speaker encoders during multi-task learning, but removing the top-down attention cue A∗ from the interface. During training, the gradients of the speaker classification loss back-propagate through the speaker encoders to update the TCN stacks, therefore, ensuring the speaker identity of the estimated speech. However, at run-time, only the TCN stacks pipeline participates in the mask estimation without A∗.

The difference between the reentry† model and reentry† is that the former involves the speaker encoders, as well the speaker classification loss, during training, but the latter does not. In Table III, we observe that reentry† has improved SI-SDRi over reentry† from 12.30dB to 12.39dB, that shows the speaker classification task, without explicitly using A∗, offers a small gain of 0.09dB. However, when A∗ is used as the attractor along with V(t) in the reentry model, we obtain a higher SI-SDRi of 12.60dB. The results suggest that the use of self-enrolled A∗ as a top-down attention cue dominates the contribution from the speaker encoders and speaker classification task.

The speaker extractor in the reentry model consists of R TCN stacks that interface with R – 1 speaker encoders. We conduct further experiments to show the interlaced architecture supports a progressively refined speaker self-enrollment process, and report in Table III. We vary the number of TCN stacks R from 2 to 4, thus the speaker encoders R – 1 from 1 to 3, in the speaker extractor. It is seen that, as R increases, the SI-SDRi of reentry increases from 11.13dB to 12.60dB at γ = 0.005, showing that the progressive refinement architecture outperforms a single-stage extraction.

We finally analyze the weight sharing among the speaker encoders. The reentry†† model shares the weights across R – 1 speaker encoders, while the reentry model has individual weights for each speaker encoder. The results suggest that the individual weights scheme outperforms the shared weights counterpart by 0.14dB in terms of SI-SDRi, which confirms our intuition to adopt individual weights in the reentry model.

E. Benchmark against baselines

We now compare the reentry model with two recent models TDSE and MuSE on speech mixture of two-speaker (VoxCeleb2-2mix) and three-speaker (VoxCeleb2-3mix) datasets as summarized in Table IV. We observe that the improved versions of TDSE and MuSE, namely TDSE-I and MuSE-I outperform their original model, i.e., TDSE-O and MuSE-O, thus serving as competitive baselines. Since MuSE-I employs the same speaker encoders architecture and multi-task learning framework as the reentry model, we used the γ = 0.005 for MuSE-I training as it proves to learn the best speaker representation according to Table III. It is seen that the reentry model outperforms all the baselines in terms of SI-SDRi, SDRi, PESQi, and STOIi on VoxCeleb2-2mix and VoxCeleb2-3mix.

We also re-implement two speech separation models, namely ConvTasnet [10] and DPRNN [12] following the original paper, and report their results on VoxCeleb2-2mix and VoxCeleb2-3mix for comparison. Since ConvTasnet and DPRNN are audio-only speech separation algorithms, they don’t use the visual information from the VoxCeleb2-2mix dataset mixture. Hence we expect that the audio-visual speaker extraction outperforms the audio-only models.

In Fig. 4 we show the histogram of SI-SDRi for VoxCeleb2-2mix test set samples by ConvTasnet, TDSE-I, MuSE-I, and
### TABLE IV

**Performance of the reentry model and baselines on the test set of simulated VoxCeleb dataset mixtures.**

| Dataset Mixtures | Task | Model | AE feature | SI-SDRi (dB) | SDRi (dB) | PESQi | STOIi | Params (m) |
|------------------|------|-------|------------|--------------|-----------|------|-------|-----------|
| VoxCeleb2-2mix   | Speech separation | ConvTasnet [10] | N/A | 10.94 | 11.27 | 0.835 | 0.230 | 8.7 |
|                  |      | DPRNN | N/A | 11.57 | 11.88 | 0.902 | 0.235 | 2.6 |
|                  | Speaker extraction | TDSE-O [38] | N/A | 10.64 | 10.93 | 0.823 | 0.226 | 20.2 |
|                  |      | TDSE-I | 0.348 | 11.62 | 11.96 | 0.958 | 0.240 | 20.1 |
|                  |      | MuSE-O [51] | 0.7 | 11.67 | 12.04 | 0.969 | 0.241 | 25.0 |
|                  |      | MuSE-I | 0.7 | 11.91 | 12.22 | 1.007 | 0.245 | 24.5 |
|                  | voxel | reentry | sync | 12.60 | 12.92 | 1.103 | 0.256 | 18.8 |
| VoxCeleb2-3mix   | Speech separation | ConvTasnet [10] | N/A | 9.31 | 9.74 | 0.264 | 0.230 | 8.8 |
|                  |      | DPRNN | N/A | 9.83 | 10.23 | 0.283 | 0.239 | 2.7 |
|                  | Speaker extraction | TDSE-O [38] | N/A | 9.78 | 10.21 | 0.36 | 0.262 | 20.2 |
|                  |      | TDSE-I | 0.348 | 11.54 | 12.04 | 0.500 | 0.303 | 20.1 |
|                  |      | MuSE-O [51] | 0.7 | 11.64 | 12.18 | 0.513 | 0.306 | 25.0 |
|                  |      | MuSE-I | 0.7 | 12.21 | 12.68 | 0.559 | 0.318 | 24.5 |
|                  | voxel | reentry | sync | 12.63 | 13.08 | 0.613 | 0.327 | 18.8 |

### TABLE V

**A comparison across various audio-visual speaker extraction implementations on VoxCeleb2 2-speaker mixture speech.**

| Model               | SDRi (dB) | SDR (dB) |
|---------------------|-----------|----------|
| AVS [79]            | -         | 5.9      |
| AV-BLSTM [77]       | -         | 3.25     |
| FaceFilter [34]     | 2.5       | -        |
| AV-U-Net [35]       | -         | 7.6      |
| VisualVoice [36]    | -         | 10.2     |
| Triantafyllos et al. [37] | 12.1 | 11.8 |
| Li et al. [55]      | -         | 6.7      |
| AVSE [52]           | -         | 6.2      |
| TDSE-O              | 10.9      | 10.9     |
| MuSE-O              | 12.0      | 12.0     |
| reentry             | 12.9      | 12.9     |

### TABLE VI

**The results on VoxCeleb2-2mix-occl test set for models that trained with and without occlusion data.**

| Model               | Train data | SI-SDRi (dB) | PESQi |
|---------------------|------------|--------------|-------|
| TDSE-I              | VoxCeleb2-2mix | 4.98 | 0.671 |
| MuSE-I              | reentry    | 4.02 | 0.625 |
|                     | 6.68       | 0.801 |
| MuSE-I              | VoxCeleb2-2mix-occl | 9.21 | 0.856 |
|                     | reentry    | 9.29 | 0.875 |
|                     | 10.27      | 0.909 |

The reentry model. It shows that the reentry model has more extracted speech samples with higher SI-SDRi gain than the baselines. In Fig. 3 we show the distribution of SI-SDRi gain for mixture with various SNR between the target and interference speakers. It is seen that the reentry model consistently achieves a higher SI-SDRi than the baselines. As the input mixture becomes less noisy (high SNR), the SI-SDRi gain becomes smaller.

**F. Benchmark against related audio-visual speaker extraction methods**

We compare the reentry model with related audio-visual speaker extraction studies in Table V. The TDSE-O, MuSE-O, and reentry models are reported on the test set of VoxCeleb2-2mix, while the reference models are reported on other simulated speech mixtures from VoxCeleb2. The reentry model outperforms the state-of-the-art method despite the fact that we only use a subset of the VoxCeleb2 dataset to create the speech mixtures for training. We understand that the SDR and SDRi results are not directly comparable across the studies as they are not trained and tested on the same dataset. Nonetheless, the compilation of the results serves as a good reference to the state-of-the-art.

**G. Speaker extraction in face of visual occlusion**

As the proposed reentry model relies on speech-lip synchronization cue, any visual occlusion will adversely impact the performance. We evaluate three systems, namely the reentry model, TDSE-I, and MuSE-I to observe how systems perform in face of visual occlusions.

We create a two-speaker visual occlusion dataset, named VoxCeleb2-2mix-occl, based on VoxCeleb2-2mix, but blackout the visual signals randomly and for a random duration while keeping the audio signals intact. This simulates the scenarios where the face-tracking algorithm fails to detect the presence of the target speaker.

In Table VI it is observed that none of the models generalizes well if not trained on occlusion data. Nonetheless, the reentry model outperforms the best baselines with 1.7dB in SI-SDRi and 0.13 in PESQi. When trained on occlusion data, all models improve, with the reentry model still showing the best SI-SDRi and PESQi.

We analyze the performance with different visual present duration after the visual occlusion in Fig. 4. We plot the SI-SDRi of the reentry model and baselines with different visual present duration. It is seen that when visual is present for about 1 second, the reentry model is able to achieve near 8dB SI-SDRi. When visual is present for 2 seconds, the reentry model is able to extract the visual occluded segments according to the prosody of the extracted speech when visual is present.

**H. Cross-dataset evaluation**

We are interested in how the reentry model trained on VoxCeleb2 performs on other datasets. We evaluate the reentry model with two competing models on Grid [80], TCD-TIMIT [81], LRS2 [77], and LRS3 [82] datasets. Grid and TCD-TIMIT are ‘studio’ videos while LRS2 and LRS3 are
reentry particularly useful in situations where pre-enrolled face or speech reentry cocktail party. The study is motivated by the tasks such as speech recognition and speaker verification. reentry human auditory attention. The proposed reentry by Edelman which shows cross-modal temporal integration in the above datasets, following the VoxCeleb2-2mix protocol. We generate as the VoxCeleb2 dataset to minimize the visual mismatch. ‘wild’ videos from BBC and TED. These datasets are preprocessed with feature detection and tracking algorithms same as the VoxCeleb2 dataset to minimize the visual mismatch. We generate 3,000 utterances to form a test set for each of the above datasets, following the VoxCeleb2-2mix protocol. As shown in Table VII on LRS2 and LRS3 datasets, the reentry model shows higher SI-SDRi and PESQI, which are consistent with the improvement on the VoxCeleb2 dataset. On the Grid and TCD-TIMIT datasets, which belong to another domain, the reentry model still shows a relative improvement over the competing models, with the exception that MuSE-I outperforms the reentry model on the Grid dataset.

VI. CONCLUSION

We propose an end-to-end audio-visual speaker extraction network to emulate human’s selective auditory attention in the cocktail party. The study is motivated by the reentry theory by Edelman which shows cross-modal temporal integration in human auditory attention. The proposed reentry model is particularly useful in situations where pre-enrolled face or speech reference is not available. In summary, the reentry model presents a step forward in solving the cocktail party problem using a computational model. It will potentially enable applications, such as intelligent hearing aids, or downstream tasks such as speech recognition and speaker verification.

REFERENCES

[1] E. C. Cherry, “Some experiments on the recognition of speech, with one and with two ears,” The Journal of the acoustical society of America, vol. 25, no. 5, pp. 975–979, 1953.

[2] A. W. Bronkhorst, “The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions,” Acta Acustica united with Acustica, vol. 86, no. 1, pp. 117–128, 2000.

[3] F. Katsuki and C. Constantiudis, “Bottom-up and top-down attention: different processes and overlapping neural systems,” The Neuropsychologist, vol. 20, no. 5, pp. 509–521, 2014.

[4] Y. Pinto, A. van den Leij, E. G. Sligte, V. A. Lamme, and H. S. Scholte, “Bottom-up and top-down attention are independent,” Journal of vision, vol. 13, no. 3, pp. 16-16, 2013.

[5] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dekah, D. Povey, S. Watanabe, and S. Khu- danpur, “Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge,” in Proc. INTERSPEECH, 2018, pp. 2808–2812.

[6] R. Tao, R. K. Das, and H. Li, “Audio-Visual Speaker Recognition with a Cross-Modal Discriminative Network,” in Proc. INTERSPEECH, 2020, pp. 2242–2246.

[7] W. Yao, C. Xu, E. S. Chng, and H. Li, “Target Speaker Extraction for Multi-Talker Speaker Verification,” in Proc. INTERSPEECH, 2019, pp. 1273–1277.

[8] X. Zhou, G. Lee, E. Yilmaz, Y. Long, J. Liang, and H. Li, “Self-and-Mixed Attention Decoder with Deep Acoustic Structure for Transformer-Based LVCSR,” in Proc. INTERSPEECH, 2020, pp. 5016–5020.

[9] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2016, pp. 31–35.

[10] Y. Luo and N. Mesgarani, “Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 27, no. 8, pp. 1256–1266, 2019.

[11] C. Xu, W. Rao, X. Xiao, E. S. Chng, and H. Li, “Single channel speech separation with constrained utterance level permutation invariant training using grid lstm,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2018, pp. 6–10.

[12] Y. Luo, Z. Chen, and T. Yoshioka, “Dual-path rnn: Efficient long sequence modeling for time-domain single-channel speech separation,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2020, pp. 46–50.

[13] Z. Chen, Y. Luo, and N. Mesgarani, “Deep attractor network for single-microphone speaker separation,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2017, pp. 246–250.

[14] N. Zeghidour and D. Grangier, “Wavesplit: End-to-end speech separation by speaker clustering,” preprint arXiv:2002.08933, 2020.

[15] D. Stoller, S. Ewert, and S. Dixon, “Wave-u-net: A multi-scale neural network for end-to-end audio source separation,” in Proceedings of 19th International Society for Music Information Retrieval Conference, 2018.

[16] M. Kollbäck, D. Yu, Z.-H. Tan, and J. Jensen, “Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 25, no. 10, pp. 1901–1913, 2017.

[17] Y. Liu and D. Wang, “Divide and conquer: A deep casa approach to talker-independent monaural speaker separation,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 27, no. 12, pp. 2092–2102, 2019.

[18] C. Xu, W. Rao, E. S. Chng, and H. Li, “Spex: Multi-scale time domain speaker extraction network,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 28, pp. 1379–1384, 2020.

[19] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, “SpEx+: A Complete Time Domain Speaker Extraction Network,” in Proc. INTERSPEECH, 2020, pp. 1406–1410.
APPENDIX

A. Speech-lip synchronization network

The speech-lip synchronization network (SLSyn) as shown in Fig. 2(a) consists of an audio front-end, a visual front-end, and an audio-visual back-end, which are detailed in Fig. 7.

The last linear layer of the SLSyn network has an input channel size of 256 and an output channel size of 2, a sigmoid activation is used at the end for binary classification. The TCN and ResNetBlock structures are detailed in Fig. 8.

Through out the paper, Conv1D, Conv2D, and Conv3D are 1, 2, and 3 dimensional convolutional layer respectively, and the parameters are represent as (in_channels/ out_channels) [kernel_size/stride]; the Conv1d* represents a group convolution with groups=in_channels; LN refers to layer normalization; ReLU is rectified linear activation; AvgPool1D and AvgPool3D are 1 and 3 dimensional average pooling layer respectively, and the parameters are represented by [kernel_size/stride].

Fig. 7. The configuration of the speech-lip synchronization network (SLSyn).

(a) TCN
(b) ResNetBlock
(c) Speaker encoder

Fig. 8. (a) The configuration of a temporal convolutional network, TCN [d], d is the stride in the Conv1D* layer, also referred to as the dilation factor. (b) The configuration of a ResNetBlock ([c/in/out]) [d/c/p]. Add refers to element-wise addition. (c) The configuration of the speaker encoder.

B. reentry model

The reentry model as shown in Fig. 11 consists of a speech encoder, an attractor encoder, a speaker extractor, R−1 speaker encoders, and a speech decoder. In the speech encoder and speech decoder, L and N are set to 40 and 256 respectively. The attractor encoder is followed by an adaptation layer of a stack of 5 TCN [0], the TCN structure is shown in Fig. 8(a). The R − 1 speaker encoders shares the same architecture, which is shown in Fig. 8(c). In the speaker encoders, the Dropout layer has a dropout probability of 0.9. PReLU is parametric ReLU. Each speaker encoder is preceded by a speech decoder and a speech encoder shared weights with the speech encoder and speech decoder depicted in Fig. 11. The speaker extractor has R and B set to 4 and 8 respectively.

We use layer normalization instead of batch normalization for all models, such that a small batch size in the fine-tuning stage will not affect the normalization. The network parameters for the speech encoder, speaker encoder, speaker extractor, and speech decoder are selected following [38, 51].