IEEE SLT 2021 Alpha-mini Speech Challenge: Open Datasets, Tracks, Rules and Baselines

Yihui Fu[1,], Zhuoyuan Yao[1,*], Weipeng He[2], Jian Wu[1], Xiong Wang[1], Zhanheng Yang[1], Shimin Zhang[1], Lei Xie[1,**], Dongyan Huang[3], Hui Bu[4], Petr Motlicek[2], Jean-Marc Odobez[2]*

[1]Audio, Speech and Language Processing Group (ASLP), School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]Idiap Research Institute, Switzerland
[3]UBTECH Technology Co., Ltd. Shenzhen, China
[4]AISHELL Technology Co., Ltd. Beijing, China

Abstract

The IEEE Spoken Language Technology Workshop (SLT) 2021 Alpha-mini Speech Challenge (ASC) is intended to improve research on keyword spotting (KWS) and sound source location (SSL) on humanoid robots. Many publications report significant improvements in deep learning based KWS and SSL on open source datasets in recent years. For deep learning model training, it is necessary to expand the data coverage to improve the model robustness. Thus, simulating multi-channel noisy and reverberant data from single-channel speech, noise, echo and room impulsive response (RIR) is widely adopted. However, this approach may generate mismatch between simulated data and recorded data in real application scenarios, especially echo data. In this challenge, we open source a sizable speech, keyword, echo and noise corpus for promoting data-driven methods, particularly deep-learning approaches on KWS and SSL. We also choose Alpha-mini, a humanoid robot produced by UBTECH equipped with a built-in four-microphone array on its head, to record development and evaluation sets under the actual Alpha-mini robot application scenario, including environmental noise as well as echo and mechanical noise generated by the robot itself for model evaluation. Furthermore, we illustrate the rules, evaluation methods and baselines for researchers to quickly assess their achievements and optimize their models.

Index Terms— keyword spotting, sound source location, noise and echo, deep learning, datasets

1. Introduction

Robots, as useful assistants and playmates, are becoming more and more popular in people's daily life. As the first chain of human-robot speech interaction (HRSI), the accuracy and efficiency of speech interaction have an important impact on the interaction effectiveness and user experience. In typical HRSI scenarios, robot always work in a very complex acoustic scene, including users' voices, background noise, and the voice of the robot itself (echo and mechanical noise). To activate the speech interactions between users and devices, a standby keyword spotting (KWS) module, also known as wake-up word detection, is particularly important to detect predefined keyword in the audio stream to trigger voice interactions. A good KWS system needs to maintain high robustness with low false rejections and false alarms under the constraint of low computation cost. Meanwhile, accurate sound source location (SSL) can provide essential cues for subsequent beamforming, speech enhancement and speech recognition algorithms. In home environments, the following interferences pose great challenges to HRSI: 1) various types of noises from TV, radio, other electrical appliances and human talking, 2) echoes from the loudspeaker(s) equipped on the robot, 3) room reverberation and 4) noises from the mechanical movements of the robot. These noise interferences complicate KWS and SSL to a great extent. Thus, robust algorithms are highly in demand.

Conventional KWS system has been developed maturely, including large vocabulary continuous speech recognition (LVCSR) based lattice search [1, 2, 3], hidden Markov model (HMM) based keyword-filler method [4, 5, 6], discriminative models based on large-margin formulation or recurrent networks and query-by-example (QbyE) based template matching approaches [7, 8, 9, 10, 11]. Recently, with the development of deep learning and its successful applications, deep KWS frameworks have been introduced [12, 13, 14, 15, 16, 17, 18]. In the deep KWS family, an acoustic model is trained to predict the sub-word of keyword and a posterior handling method is followed to generate a confidence score of the whole keyword. These approaches are highly attractive to deploy on edge-device with small footprint and low latency, as the size of the model can be easily controlled and no complicated graph-search is involved. Besides, attention-based end-to-end method has also been introduced to the KWS task [19] and further performance improvement has been observed which significantly simplifies the model structure and the decoding process. Another trick to boost KWS performance recently is to employ a two-stage strategy, where a first-stage detector provides candidates to the second stage to make the final decision [20, 21].

SSL has been studied for decades. Conventionally, generalized cross correlation with phase transform (GCC-PHAT) [22], steered-response power with phase transform (SRP-PHAT) [22] and multiple signal classification (MUSIC) [23] are among the most popular approaches. These traditional signal processing based methods are analytically derived with the assumptions about the signal, noise and environment such as the noise is white and the SNR is higher than 0dB, etc. Recently, Lin et al. investigated the reverberation-robust localization approach of using redundant information of multiple microphone pairs and proposed the OnsetMCCC and MCC-PHAT methods [24, 25]. With the rapid development of deep learning based speech enhancement and separation, several methods were shown to achieve promising performance on the SSL task. In [26, 27], the authors estimated the masks of target speech to improve the robustness of conventional cross-correlation-based,

beamforming-based and subspace-based algorithms for SSL estimation in environments with strong noise and reverberation. In [26, 28], the authors utilized the ideal ratio mask (IRM) and its variants and considered direct sound as the target signal, which leads to high localization accuracy.

Although many approaches have addressed the problem of KWS and SSL, there have been only a few studies evaluate the ability of KWS and SSL on humanoid robots with challenging acoustic conditions. On the other hand, large-scale dataset on robot for KWS or SSL is still extremely deficient. He et al. proposed multiple speaker detection and localization dataset recorded by Pepper robot [29]. Lollmann et al. published acoustic source localization and tracking dataset on LOCATA challenge [30]. However, these datasets neither consider the scenario of echo nor have good coverage of different room sizes and reverberation scenarios. Thus it is necessary to release a sizable dataset for KWS and SSL based on humanoid robot and a common platform to better tackle the problem of HSRI in real application scenarios.

In this paper, we address the necessity of solving the problem of KWS and SSL on humanoid robot in noisy and echo scenario. It is expected that researchers from both academia and industry can promote the problem solving through this challenge. The rest of the paper is organized as follows. In Section 2, we give detailed introduction of dataset to release. In Section 3 and Section 4, the details of rules, evaluation method and baselines of KWS and SSL tracks are introduced. Other information about participating the challenge is available in Section 5. A conclusion is drawn in Section 6.

2. Datasets

Our goal of releasing the open source dataset in Table 1 is to ensure the fair training resources and evaluation platform for researchers. The training data includes single channel keyword, speech, noise, echo data and recorded echo and mechanical noise of Alpha-mini. The development and evaluation sets contain keyword, speech, noise, echo and mechanical noise data recorded by Alpha-mini. During recording, we play the clean and noise signals through Hi-Fi loudspeakers and use the built-in microphone array of Alpha-mini to record. As for the echo data, various types of audio played by Alpha-mini built-in dual-loudspeaker is recorded by the built-in microphone array on the head of the Alpha-mini. The mechanical noise is generated by movable joints of Alpha-mini and recorded by the same built-in microphone array. Typical recording scenes are shown in Fig. 1. The robot is equipped with four microphones located on its head and two loudspeakers located on both sides of its waist. The distance between two neighbor microphones and two loudspeakers are 3.7 cm and 6.3 cm, respectively. The vertical distance between the loudspeakers and the plane of microphone array is 13 cm, as shown in Fig. 2. All recorded data are six-channel signal, where the first four channels are recorded signals and the rest two channels are reference signals played by the dual-loudspeakers of Alpha-mini.

Here we give the detailed illustration of subsets in Table 1:

- **Keyword-Train**: 'Wukong Wukong' wake-up word speech data provided by UBTECH recorded in anechoic room including voices from both adults and children, used for KWS model training.
- **Speech-Train**: The open-source AISHELL-1 [31] training set is processed by deep complex convolutional recurrent network (DCCRN) [32] and weighted prediction Error (WPE) algorithm [33], resulting in enhanced and dereverbed 'clean'
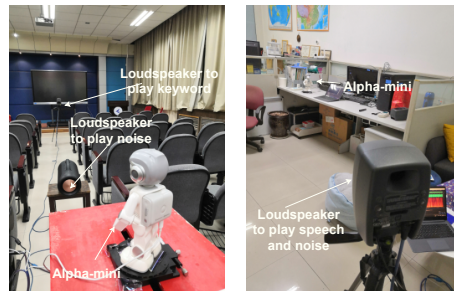

Fig. 1: *The typical recording scenes.*

and 'dry' version, which can be used for KWS model and SSL model training.
- **Noise-Train**: The noise data comes from 1) songs and pure music, 2) noise set of DNS challenge [34] and 3) various kinds of indoor noise including but not limited to clicking, keyboard, door opening/closing, fan, bubble noise, etc. This set can be used in KWS model and SSL model training.
- **Echo-Train**: The data released for echo simulation, which includes 1) songs, pure music, news broadcasting, people crosstalk and 2) speech generated by the Alpha-mini text-to-speech engine.
- **Echo-Record**: Various types of audio, played by Alpha-mini built-in dual-loudspeakers, recorded (echo) by the Alpha-mini built-in microphone array in a quiet room. The audio types played by Alpha-mini are the same as Echo-Train but the audio data has no overlap.
- **Noise-Mech**: Noise of mechanical movements generated by the movable joints of Alpha-mini, recorded by the Alpha-mini built-in microphone array in a quiet room.
- **KWS-Dev**: Recorded keywords, noise, echo and mechanical noise by Alpha-mini in two rooms. Keywords and noise are played by two Hi-Fi loudspeakers while echo is generated by Alpha-mini at the same time. The mechanical noise is recorded along then added to the recorded noisy and echoic signal. The audio types of played keyword, noise, echo and mechanical noise are the same as corresponding sets in training but audio data has no overlap. The Hi-Fi loudspeakers are randomly placed in each room. This set is used as development set for KWS model optimization.
- **SSL-Dev**: Recorded speech, noise, echo and mechanical noise by Alpha-mini in two rooms. Speech and noise are played by one Hi-Fi loudspeaker separately. Echo and mechanical noise are generated by Alpha-mini separately. Then we mix speech, noise, echo and mechanical noise together. The speech comes from enhanced and dereverbed 'clean' and 'dry' version of AISHELL-1 development and test set. The audio types of played noise, echo and mechanical noise are the same as corresponding sets in training but audio data has no overlap. The angle between Alpha-mini and loudspeakers covers every single degree from $1°$ to $360°$. The angle definition is illustrated in Fig. 2(a) and the straight ahead of the robot is defined as $90°$. This set is used as development set for SSL model optimization.
- **KWS-Test**: Recorded keywords, noise, echo and mechanical noise by Alpha-mini in three rooms. Other setups are the same as KWS-Dev. This is the evaluation set for KWS Track.
- **SSL-Test**: Recorded speech, noise, echo and mechanical noise by Alpha-mini in three rooms. Other setups are the same as SSL-Dev. This is the evaluation set for SSL Track.

Table 1: *Data to release.*

Dataset	Subset	Duration (hrs)	Format	Scenario	Mic-Loudspeaker distance (metres)
Training	Keyword-Train	9.4	16kHz, 16bit, single channel wav	-	-
	Speech-Train	146.1			
	Noise-Train	60.0			
	Echo-Train	28.5			
	Echo-Record	3.0	16kHz, 16bit, six-channel wav		
	Noise-Mech	8.6			
Development	KWS-Dev	7.5	16kHz, 16bit, six-channel wav	Keyword only Keyword+Noise Keyword+Echo Keyworkd+Noise+Echo Keyword+Echo+Mech	[2, 4]
	SSL-Dev	20.0		Speech only Speech+Noise Speech+Echo Speech+Noise+Echo Speech+Echo+Mech	
Evaluation	KWS-Eval	TBA	Same as Development	Same as Development	[2, 5]
	SSL-Eval				

(a) Top view

(b) Front view
(c) Left view

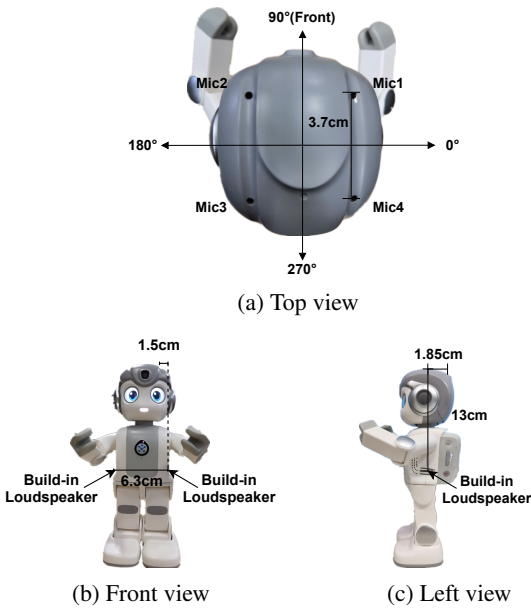**Fig. 2**: *Three views of Alpha-mini robot.*

3. Keyword Spotting (KWS) Track

This track is designed for KWS task. We illustrate the data arrangement, evaluation and ranking method, rules and baseline methods and results in this section.

3.1. Data Arragement

The data can be used in this track is shown in Table 2. Participants can use their own room impulse response (RIR), either collected or simulated, for data augmentation to train the KWS model. Furthermore, Echo-Record and Noise-Mech are provided as the reference of time-delay of echo and mechan-

ical noise of Alpha-mini, respectively. Participants can also use these data sets during training. KWS-Dev, SSL-Dev, KWS-Eval, SSL-Eval are six-channel recorded data. Participants can use KWS-Dev and SSL-Dev directly without any simulation to optimize the model.

3.2. Evaluation and Ranking

We use a combination of false reject rate (FRR) and false alarm rate (FAR) on KWS-Eval and SSL-Eval respectively as the criterion of the KWS performance. Suppose the evaluation set has N_{key} examples with keyword and $N_{non-key}$ examples without keyword, we define FRR and FAR as follows:

$$\text{FRR} = \frac{N_{FR}}{N_{key}}, \quad \text{FAR} = \frac{N_{FA}}{N_{non-key}}, \quad (1)$$

where N_{FR} is the number of examples with keyword but the KWS system gives a negative decision and N_{FA} is the number of examples without keyword but the KWS system gives a positive decision. The final score of KWS is defined as:

$$\text{Score}^{KWS} = \text{FRR} + \text{FAR}. \quad (2)$$

FRR and FAR are calculated on all examples in KWS-Eval and SSL-Eval respectively and the final rank is Score^{KWS} calculated by Eq. (2). The system has lower Score^{KWS} will be ranked higher.

KWS-Eval and SSL-Eval will not be released before organizers notify the participants about the results. Participants need to provide the organizers with a docker image of a runnable KWS system. The executable file in the image needs to receive the list of data in KWS-Eval and SSL-Eval and outputs the result of KWS. The output determines whether the sample contains keyword. If keyword exists, the sample is labeled as 1, and 0 otherwise. A detailed technical support of the usage and submission of docker will be provided later.

3.3. Rules

The use of any other data that is not provided by organizers (except for RIR) is strictly prohibited. Furthermore, it is not allowed to use KWS-Dev and SSL-Dev to train the KWS model.

There is no limitation on KWS model structure and model training technology used by participants. And the KWS model can have a maximum of 500 ms look ahead. To infer the current frame T (in ms), the algorithm can access any number of past frames but only 500 ms of future frames ($T + 500$ ms). In case there are submitted systems with the same score, the system with lower time delay will be given a higher ranking.

Table 2: *Data arrangement for KWS Track.*

Train	Development	Evaluation
Keyword-Train		
Speech-Train		
Noise-Train	KWS-Dev	KWS-Eval
Echo-Train	SSL-Dev	SSL-Eval
Echo-Record		
Noise-Mech		

3.4. Baseline

Front-end: We use signal-based front-end for pre-processing. We apply frequency least mean square (FLMS) algorithm for acoustic echo cancellation (AEC) and delay and sum beamforming (DSBF) with SSL estimated by GCC-PHAT on multi-channel signal to generate single-channel signal as the input of KWS system.

Deep KWS: The first baseline is based on deep KWS model. After pre-processing, we extract 40-dimension mel-filterbanks feature by a window of 25ms with a shift of 10ms as the input of deep KWS. An 8-layer dilation time-delay neural network (TDNN) [35] is used as the KWS model shown in Fig. 3(a). The kernel size of the first four layers is 5, and 3 for the rest. Dilation rate of these layers loops among $\{1, 2, 4, 8\}$. There is a batch normalization (BN) layer with rectified linear unit (ReLU) activation function between each TDNN layer. A fully connection layer (FC) is applied to map the output of TDNN into two categories – keyword and filler. Softmax function is used to generate the posterior probability of both keyword and non-keyword.

We use post processing in [36] to generate the keyword confidence score from the posterior probabilities. The system will wake up if the confidence exceeds a predefined threshold. First, we smooth the raw posterior probabilities from the model over a fixed time window of size w_{smooth}. Suppose p_t is the raw posterior probabilities of keyword at frame t, smoothing is done by:

$$p_t^{'} = \frac{1}{t - h_{smooth} + 1} \sum_{k=h_{smooth}}^{t} p_k, \qquad (3)$$

where $h_{smooth} = \max(1, t - w_{smooth} + 1)$ is the index of the first frame within the smooth window. Thus the confidence score at frame t is the smoothed posterior $p_t^{'}$.

For data simulation, the RT60 of RIRs we generate ranges from 0.2 s to 0.8 s with image method. The room size ranges from 3 m $\times$ 3 m to 8 m $\times$ 8 m and the hight is maintained at 3 m. The mic-loudspeaker distance ranges from 1.5 m to 5 m. Both SNR and SER range from -5 dB to 10 dB. During training, cross entropy is used as the loss function. The batch size

and the initial learning rate is set to 128 and 0.001, respectively. We train the model for 50 epochs with Adam optimizer using PyTorch. The result is shown in Table 3. Note that the performance of the baseline KWS system decreases rapidly in noisy and echo scenarios. In particular, echo poses a much bigger challenge to the model than noise, possibly because the source of echo is closer to the microphone thus the SER is relatively low. Compared with noise scenario, the Score$^{\text{SSL}}$ in echo scenario decreases 0.16 on average. In addition, the overall performance of the KWS system is worse in the conference room scene due to larger reverberation and mic-loudspeaker distance.

Keyword-filler: We provide another baseline based on Kaldi Hi-Mia recipe [1]. The acoustic model accepts the mel-filterbanks feature of front-end output as input and outputs the posterior probabilities of probability density function-identification (pdf-id). We extract 71-dimension mel-filterbanks feature by a window of 25 ms with a shift of 10 ms. A 6-layer dilation TDNN with ReLU activation function is used to get the time domain information. Then, a fully connected layer maps the high-dimensional representation to the posterior probabilities of pdf-id. The model is trained for 2 epochs with 512 batch size. The learning rate degradation algorithm is shown in Eq. 4)

$$lr_j = lr_0 \times \exp(\frac{j}{S - 1} \log(\frac{lr_{S-1}}{lr_0})) \quad j = 0,...,S - 1, \qquad (4)$$

where S denotes the total step of training and lr_j denotes the learning rate at step j. As for language model, a decoding graph is used to calculate confidence score of keyword from the posterior probability of the acoustic model. The decoding graph only accepts the phonemes included in the keyword and computes the score of the keyword. The result is shown in Table 3. It is proved again that echo, rather than noise, poses a greater impact on KWS result. Furthermore, KWS performance degrades to a great extend in larger reverberation and mic-loudspeaker distance scenario. A complete Kaldi based baseline script will be provided later.

4. Sound Source Location (SSL) Track

This track is designed for SSL task. We illustrate the data arrangement, evaluation and ranking method, rules and baseline of SSL task in this section.

4.1. Data

The data that participants can use in this track is shown in Table 5. Participants can also use their own RIR, either collected or simulated, for data augmentation to train the SSL model. Furthermore, Echo-Record and Noise-Mech are provided as the reference of time-delay of echo and mechanical noise of Alpha-mini, respectively. Participants can also use these data sets during training. SSL-Dev and SSL-Eval are six-channel recorded data. Participants can use SSL-Dev directly without any simulation to optimize the model.

4.2. Evaluation and Ranking

We use a combination of Mean Absolute Error (MAE) and accuracy (ACC) as the criterion of the SSL performance. With the list of absolute errors of angle $\{e_i\}, i = 1, ...N$, where N

---

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/hi_mia/w1

Table 3: *Results of KWS baseline.*

Room	Set	Scenario	FRR		Average		Set	Scenario	FAR		Average	
			Deep KWS	Keyword-filler	Deep KWS	Keyword-filler			Deep KWS	Keyword-filler	Deep KWS	Keyword-filler
Office	KWS-Dev	Keyword only	0.10	0.01			SSL-Dev	Speech only	0.02	0.03		
		Keyword+Noise	0.16	0.08				Speech+Noise	0.14	0.08		
		Keyword+Echo	0.26	0.31				Speech+Echo	0.21	0.31		
		Keyword+Noise+Echo	0.41	0.37				Speech+Noise+Echo	0.23	0.24		
		Keyword+Echo+Mech	0.30	0.36	0.32	0.35		Speech+Echo+Mech	0.27	0.16	0.19	0.14
Conference Room		Keyword only	0.13	0.14				Speech only	0.03	0.02		
		Keyword+Noise	0.31	0.36				Speech+Noise	0.21	0.05		
		Keyword+Echo	0.42	0.57				Speech+Echo	0.24	0.20		
		Keyword+Noise+Echo	0.62	0.64				Speech+Noise+Echo	0.24	0.14		
		Keyword+Echo+Mech	0.52	0.68				Speech+Echo+Mech	0.29	0.15		

Table 4: *Results of SSL baseline.*

Set	Room	Scenario	ACC$_{10}$ (%)	Average (%)	ACC$_{7.5}$ (%)	Average (%)	ACC$_5$ (%)	Average (%)	MAE($^\circ$)	Average ($^\circ$)
SSL-Dev	Office	Speech only	61.67		45.49		33.96		9.80	
		Speech+Noise	46.25		33.40		23.68		22.17	
		Speech+Echo	28.96		21.32		16.04		35.64	
		Speech+Noise+Echo	19.65		15.28		10.97		53.37	
		Speech+Echo+Mech	19.72	33.69	14.93	24.23	10.97	17.78	53.88	38.50
	Conference Room	Speech only	57.99		39.37		27.57		11.36	
		Speech+Noise	46.60		33.19		24.65		20.23	
		Speech+Echo	23.61		15.90		11.60		51.45	
		Speech+Noise+Echo	15.83		11.74		8.75		64.07	
		Speech+Echo+Mech	16.67		11.67		9.58		63.06	

Table 5: *Data arrangement for SSL Track.*

Train	Development	Evaluation
Speech-Train		
Noise-Train		
Echo-Train	SSL-Dev	SSL-Eval
Echo-Record		
Noise-Mech		

is the number of examples, we compute the MAE as:

$$\mathrm{MAE} = \frac{1}{N} \sum_{i=1}^{N} e_i. \qquad (5)$$

ACC under different tolerances δ is defined as:

$$\mathrm{ACC}_\delta = \frac{1}{N} \sum_{i=1}^{N} a_i, \quad a_i = \begin{cases} 1 & \text{if } e_i \leqslant \delta \\ 0 & \text{otherwise} \end{cases}, \qquad (6)$$

The final score of SSL is defined as:

$$\begin{aligned} \mathrm{Score}^{\mathrm{SSL}} &= (0.3 \times \mathrm{ACC}_{10} + 0.35 \times \mathrm{ACC}_{7.5} + 0.35 \times \mathrm{ACC}_5) \\ &+ (1 - \mathrm{MAE}/\mathrm{MAE}_{\mathrm{baseline}}). \end{aligned} \qquad (7)$$

The final rank is computed according to ACC under each tolerance and MAE of all examples in SSL-Eval by Eq. (7). The MAE$_{\mathrm{baseline}}$ of SSL-Eval will be released by organizers. The system with higher score will be ranked higher.

SSL-Eval will not be released before organizers notify the participants about the results. Participants need to provide organizers with a docker image of a runnable SSL system. The executable file in the image needs to receive the list of data in SSL-Eval and outputs the result of SSL. The output determines

the direction of speech ranges from 1° to 360°. A detailed technical support of the usage and submission of docker will be provided later.

4.3. Rules

The use of any other data that is not provided by organizers (except for RIR) is strictly prohibited. Furthermore, it is not allowed to use SSL-Eval and Keyword-Train to train the SSL model.

There is no limitation on the system architecture, models, training techniques and time delays. However, we encourage participants to develop models with better performance and lower time delay. In case the submitted systems with the same score, the system with lower time delay will be given higher ranking.

4.4. Baseline

Inspired by [29, 37], we adopt a fully convolutional multi-task framework for SSL task which takes multi-channel signal as input and output the probability distribution of the direction of sound source. We adapt short time Fourier transform (STFT) with 32ms frame length and 16ms frame hop to first five channels of the raw waveform and derive its magnitude and phase. Then we concatenate the magnitude and phase to generate $\mathbf{X} \in \mathbb{R}^{2C \times F \times T}$ as the input of the model, where C denotes channel number, F denotes the number of frequency bins, T denotes the number of frames and 2 denotes magnitude and phase. The 3-layer temporal convolutional networks (TCN) module uses dilated convolution network whose dilation increase exponentially to get wider receptive field and more contextual information. The details of the model is shown in Fig. 3(b). Multi-task is adopted to predict both the SSL and speech/non-speech (SNS) likelihood. The desired SSL output values are the maximum of Gaussian functions centered at the

DOAs of the ground truth source:

$$p_i^{\text{SSL}} = \begin{cases} \max_{\overline{\theta} \in \Theta} \exp(-d(\theta_i, \overline{\theta})^2/\sigma^2) & 1 \leqslant i \leqslant 360 \\ 0 & \text{otherwise} \end{cases}, \quad (8)$$

where $\Theta = \Theta_s \cup \Theta_n$ is the union of ground truth speech and noise DOAs, $\sigma = 45°$ is the parameter to control the width of the Gaussian curves, $d(\cdot, \cdot)$ denotes the distance between two angles. The desired SNS output values are the one-hot value depend on whether the nearest source is speech or noise:

$$p_i^{\text{SNS}} = \begin{cases} 1 & \text{if the nearest source is speech} \\ 0 & \text{otherwise} \end{cases}. \quad (9)$$

The loss function is defined as the mean square error (MSE) between estimated and ground truth SSL and SNS:

$$\text{Loss} = \left\| \mathbf{p}^{\text{SSL}} - \hat{\mathbf{p}}^{\text{SSL}} \right\|_2^2 + \left\| \mathbf{p}^{\text{SNS}} - \hat{\mathbf{p}}^{\text{SNS}} \right\|_2^2, \quad (10)$$

where $\mathbf{p}^{\text{SSL}}, \hat{\mathbf{p}}^{\text{SSL}}, \mathbf{p}^{\text{SNS}}, \hat{\mathbf{p}}^{\text{SNS}} \in \mathbb{R}^{1 \times 360}$. During evaluation, the result of speaker location is defined by:

$$\hat{\theta} = \underset{1 \leqslant i \leqslant 360}{\text{argmax}} (\hat{\mathbf{p}}_i^{\text{SSL}} \cdot \hat{\mathbf{p}}_i^{\text{SNS}}). \quad (11)$$
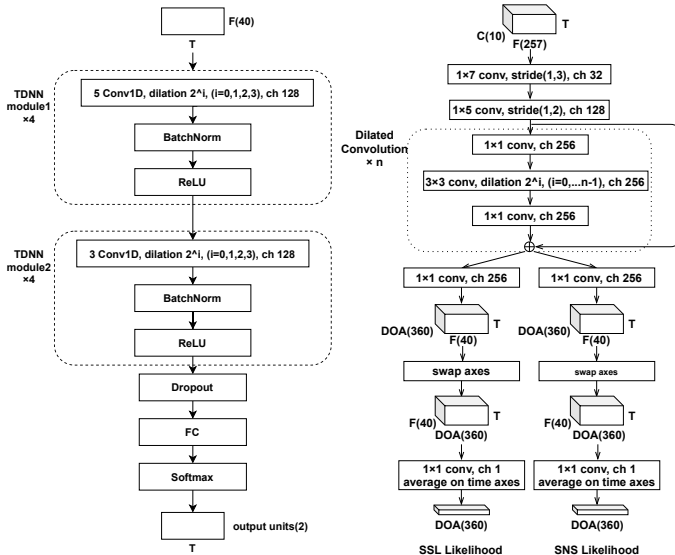
All hyper-parameters for data simulation are the same as KWS Track. We train the model for 20 epochs with Adam optimizer using PyTorch. Initial learning rate is set to 0.001 and will halve if no improvement on SSL-Dev. The result of baseline is shown in Table 4. It is worth noting that compared with the noise interference in far-field, the echo in near-field has a greater impact on the accuracy of SSL. Compared with Speech+Noise scenario, the ACC decreases 15.06 % and MAE increases 44.69° on average in Speech+Echo scenario. Furthermore, all results of Speech+Noise+Echo and Speech+Echo+Mech are very close, which indicates that mechanical noise also poses apparent impact on SSL accuracy.

5. Important dates

- September 27th, 2020: Registration due.
- September 30th, 2020: Release of the training and development set.
- November 22nd, 2020: Deadline for participants to submit docker mirror.
- December 6th, 2020: Organizers will notify the participants about the results.
- December 27th, 2020: Working note report deadline.
- January 19th-22nd, 2021: 2021 IEEE SLT Workshop date.

6. Conclusions

The IEEE SLT 2021 ASC is intended to promote research on KWS and SSL on humanoid robots in noise and echo scenarios. We provide train, development and evaluation datasets for participants to train and evaluate the model, as well as rules, evaluation methods and baselines as reference. It is expected that researchers from both academia and industry can advance the problem solving through this challenge.

(a) Deep KWS baseline.   (b) SSL baseline.

Fig. 3: *Model architecture of Deep KWS and SSL baseline.*

7. References

[1] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 615–622.

[2] P. Motlicek, F. Valente, and I. Szoke, "Improving acoustic based keyword spotting using LVCSR lattices," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4413–4416.

[3] I.-F. Chen, C. Ni, B. P. Lim, N. F. Chen, and C.-H. Lee, "A novel keyword+ LVCSR-filler based grammar network representation for spoken keyword search," in *The 9th International Symposium on Chinese Spoken Language Processing*. IEEE, 2014, pp. 192–196.

[4] B. Yan, R. Guo, X. Zhu, and B. Zhang, "An approach of keyword spotting based on HMM," in *Proceedings of the 3rd World Congress on Intelligent Control and Automation (Cat. No. 00EX393)*, vol. 4. IEEE, 2000, pp. 2757–2759.

[5] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *International Conference on Acoustics, Speech, and Signal Processing,*. IEEE, 1989, pp. 627–630.

[6] C. Choisy, "Dynamic handwritten keyword spotting based on the NSHP-HMM," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 1. IEEE, 2007, pp. 242–246.

[7] J. Hou, L. Xie, and Z. Fu, "Investigating neural network based query-by-example keyword spotting approach for personalized wake-up word detection in Mandarin Chinese," in *ISCSLP*. IEEE, 2016, pp. 1–5.

[8] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *ICASSP*. IEEE, 2015, pp. 5236–5240.

[9] J. Hou, Y. Shi, M. Ostendorf, M.-Y. Hwang, and L. Xie, "Mining effective negative training samples for keyword spotting," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7444–7448.

[10] Y. Yuan, Z. Lv, S. Huang, and L. Xie, "Verifying deep keyword spotting detection with acoustic word embeddings," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 613–620.

[11] Y. Yuan, L. Xie, C.-C. Leung, H. Chen, and B. Ma, "Fast query-by-example speech search using attention-based deep binary embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[12] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4087–4091.

[13] Z. Chen, Y. Qian, and K. Yu, "Sequence discriminative training for deep learning based acoustic keyword spotting," *Speech Communication*, vol. 102, pp. 100–111, 2018.

[14] G. Retsinas, G. Sfikas, N. Stamatopoulos, G. Louloudis, and B. Gatos, "Exploring critical aspects of CNN-based keyword spotting. a PHOCNet study," in *IAPR*, 2018, pp. 13–18.

[15] S. Fernández, A. Graves, and J. Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *International Conference on Artificial Neural Networks*. Springer, 2007, pp. 220–229.

[16] T. Higuchi, M. Ghasemzadeh, K. You, and C. Dhir, "Stacked 1D convolutional networks for end-to-end small footprint voice trigger detection," *arXiv preprint arXiv:2008.03405*, 2020.

[17] S. Adya, V. Garg, S. Sigtia, P. Simha, and C. Dhir, "Hybrid Transformer/CTC networks for hardware efficient voice triggering," *arXiv preprint arXiv:2008.02323*, 2020.

[18] X. Wang, S. Sun, and L. Xie, "Virtual adversarial training for ds-cnn based small-footprint keyword spotting," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 607–612.

[19] C. Shan, J. Zhang, Y. Wang, and L. Xie, "Attention-based end-to-end models for small-footprint keyword spotting," *arXiv preprint arXiv:1803.10916*, 2018.

[20] M. Wu, S. Panchapagesan, M. Sun, J. Gu, R. Thomas, S. N. P. Vitaladevuni, B. Hoffmeister, and A. Mandal, "Monophone-based background modeling for two-stage on-device wake word detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5494–5498.

[21] S. Sigtia, E. Marchi, S. Kajarekar, D. Naik, and J. Bridle, "Multi-task learning for speaker verification and voice trigger detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6844–6848.

[22] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE transactions on acoustics, speech, and signal processing*, vol. 24, no. 4, pp. 320–327, 1976.

[23] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[24] S. Lin, "Reverberation-robust localization of speakers using distinct speech onsets and multichannel cross correlations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2098–2111, 2018.

[25] ——, "Jointly tracking and separating speech sources using multiple features and the generalized labeled multi-bernoulli framework," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 3211–3215.

[26] P. Pertilä and E. Cakir, "Robust direction estimation with convolutional neural networks based steered response power," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 6125–6129.

[27] C. Xu, X. Xiong, S. Sun, R. Wei, and H. Li, "Weighted spatial covariance matrix estimation for music based TDOA estimation of speech source," in *Interspeech*, 2017, pp. 1894–1898.

[28] Z. Wang, X. Zhang, and D. Wang, "Robust speaker localization guided by deep learning-based time-frequency masking," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 178–188, 2018.

[29] W. He, P. Motlicek, and J.-M. Odobez, "Deep neural networks for multiple speaker detection and localization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 74–79.

[30] H. W. Löllmann, C. Evers, A. Schmidt, H. Mellmann, and W. Kellermann, "The LOCATA challenge data corpus for acoustic source localization and tracking," in *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2018, pp. 410–414.

[31] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–5.

[32] Y. Hu, Y. Liu, S. Lv, M. Xing, S. Zhang, Y. Fu, J. Wu, B. Zhang, and L. Xie, "DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement," *arXiv preprint arXiv:2008.00264*, 2020.

[33] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B. H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio Speech Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[34] C. K. Reddy, E. Beyrami, H. Dubey, V. Gopal, R. Cheng, R. Cutler, S. Matusevych, R. Aichner, A. Aazami, S. Braun *et al.*, "The interspeech 2020 deep noise suppression challenge: Datasets, subjective speech quality and testing framework," *arXiv preprint arXiv:2001.08662*, 2020.

[35] A. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *Readings in Speech Recognition*, vol. 1, no. 3, pp. 393–404, 1990.

[36] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4087–4091.

[37] W. He, P. Motlicek, and J.-M. Odobez, "Joint localization and classification of multiple sound sources using a multi-task neural network," in *Interspeech 2018*, 2018.