LC4SV: A DENOISING FRAMEWORK LEARNING TO COMPENSATE FOR UNSEEN SPEAKER VERIFICATION MODELS

Chi-Chang Lee[1,2]*, Hong-Wei Chen*[1]*, Chu-Song Chen*[1,2]*, Hsin-Min Wang*[2]*, Tsung-Te Liu*[1]*, Yu Tsao*[2]

[1]National Taiwan University, Taipei, Taiwan
[2]Academia Sinica, Taipei, Taiwan

ABSTRACT

The performance of speaker verification (SV) models may drop dramatically in noisy environments. A speech enhancement (SE) module can be used as a front-end strategy. However, existing SE methods may fail to bring performance improvements to downstream SV systems due to artifacts in the predicted signals of SE models. To compensate for artifacts, we propose a generic denoising framework named LC4SV, which can serve as a pre-processor for various unknown downstream SV models. In LC4SV, we employ a learning-based interpolation agent to automatically generate the appropriate coefficients between the enhanced signal and its noisy input to improve SV performance in noisy environments. Our experimental results demonstrate that LC4SV consistently improves the performance of various unseen SV systems. To the best of our knowledge, this work is the first attempt to develop a learning-based interpolation scheme aiming at improving SV performance in noisy environments.

Index Terms— speech enhancement, speaker identification, speaker verification, reinforcement learning

1. INTRODUCTION

A speech enhancement (SE) process involves extracting speech components from a distorted speech signal to create an enhanced signal that exhibits better properties [1]. Recently, SE was formulated as a regression task that converts noisy speech signals into clean ones using deep neural network-based mapping functions [2–11]. Most of the time, signal-level distance measures are used as objective functions for model training (e.g., L1 norm [12], L2 norm [13], SI-SDR [14], or multiple-resolution loss [15]). For the evaluation of SE performance, a standard measure is to conduct a subjective listening test, usually requiring measurement of speech quality and intelligibility. However, achieving unbiased subjective assessment results requires a large-scale listening test (a large number of listeners, each listening to a large number of speech utterances), which is often prohibitive. To overcome this limitation, numerous objective evaluation metrics have been developed [16–19].

On the other hand, the SE unit, as a separately trained pre-processor, can play a key role in speech-related applications, improving the performance without modifying the main application system parameters [20–22]. In particular, speaker identification (SID) and speaker verification (SV) tasks are popular speech-related applications. The model architecture and objective function of SE have been investigated in several studies to improve the performance of SID and SV under noisy conditions [23, 24]. Moreover, some studies explored that SE-induced artifacts and distortions may even degrade SV performance [23]. To deal with this issue, Shon et al. [25], Mošner et al. [26], and Dowerah et al. [27] directly cascade an SV model and use the corresponding SV target errors instead of signal-level distances (e.g., L1 and L2) to train SE models. However, most commercial SV systems are often provided by third parties, which may not be accessible when training SE models. Since different SV models differ in various training settings, an SE unit trained only on the SV objective of one specific SV model may not generalize well to other unseen SV systems. Directly feeding the predicted output of an SE model into a downstream model would introduce severe artifacts and destroy the integrity of critical information required for SID and SV tasks. To prevent artifacts and over-parameterization, we employ a learning-based interpolation agent. The agent automatically determines an appropriate linear combination of the enhanced output and the corresponding noisy input to compensate for the information, thereby mitigating the adverse effects of the SE model and improving SV performance.

In this work, we design a generic denoising framework, LC4SV, as a "universal" pre-processor for SV in noisy environments by combining an SE unit for primary denoising and an interpolation agent as a compensation processor. First, we pre-train the SE model with a signal-level measurement and then fine-tune it with the SV objective generated by a proxy SV model. Next, to prevent the enhanced results from lacking crucial information required by another proxy SV model, we additionally train a reinforcement learning (RL)-based interpolation agent to automatically determine the appropriate interpolation of the enhanced output and the noisy input for compensation. The RL-based interpolation agent is trained to achieve a lower error rate of verification results, so the interpolated output can better generalize to other SV models

979-8-3503-0689-7/23/$31.00 ©2023 IEEE

accordingly. According to the experimental results, LC4SV is able to improve the SV performance of various unseen SV systems, outperforming the SV performance of the same SV systems using noisy or enhanced speech signals of other SE models. Furthermore, LC4SV achieves better performance compared to two baseline methods: (i) using the signal-to-noise ratio (SNR) as a hard threshold to select either enhanced or noisy signals to feed into the SV model and (ii) predetermining the interpolation coefficients. This study makes two main contributions: (1) to the best of our knowledge, this is the first attempt to develop a generic SE preprocessor applicable to a variety of unseen SV systems; (2) we propose a rational interpolation scheme for compensation and relate it to the motivation for better generalization.

2. BACKGROUND AND MOTIVATION

2.1. Noise Robustness Strategies for Speaker Verification

It is well known that acoustic mismatch (often caused by noise interference) limits the applicability of speech-related techniques. To improve noise robustness, most SV models tend to use real or synthetic noisy speech utterances as data augmentation for model training [28]. For example, the well-known freely available speaker recognition dataset Voxceleb [29] provides a large amount of real noisy speech data. Many SV models benefit from its sufficient training data to achieve good robustness [30]. However, most of the SV systems provided by third parties are black-box models whose parameters cannot be fine-tuned to improve performance in specific noisy environments, especially at low SNR. In contrast, an SE unit can be trained separately and used as an advanced front-end processor combined with existing SV models to form a two-stage process for better noise robustness. To further improve the effect of SE units on SV, several attempts have been made to directly update SE model parameters by using the prediction error of a specific SV model [25, 26]. In their scenarios, the SV model used for training and evaluation is the same. However, when the evaluated SV system is different from the trained system, the SV performance with enhanced speech utterances would be sub-optimal due to the remained artifacts and the over-parameterization issue.

2.2. Artifacts Caused by SE Models

In general, SV models exhibit robustness to natural noisy speech signals because they are usually included in the training data as part of the data augmentation scheme. In contrast, artifacts in the output of certain SE models may have unseen effects on SV models. Given the limited model capacity, SE models inevitably introduce artifacts, which can degrade the performance of downstream SV models for reasons that are difficult to trace.

Constrained by the inability to simultaneously modify the front-end and back-end models, it is necessary to devise a post-processing strategy to address the artifacts. An effective and straightforward solution is to employ interpolation techniques instead of directly feeding the enhanced speech signals into the SV model. By combining the enhanced signal with its corresponding noisy input, we impose a constraint to ensure that the resulting signal remains close to the characteristics of natural signal. This approach helps to compensate for lost information and mitigate the negative effects of artifacts. Recently, in the field of Automatic Speech Recognition (ASR), Sato et al. [31] proposed a learning-based method to predict the likelihood of an enhanced signal degrading the performance compared to the original noisy signal. Instead of treating the possibility as a hard threshold, they turned it into an interpolated weight value and empirically improved ASR performance by using this interpolation scheme. However, unlike the ASR task, which calculates the performance metric from individual samples, the performance metric of SV needs to be determined from multiple paired samples. Our framework is proposed to construct an interpolation scheme directly aimed at maximizing SV performance. By pairwisely considering the contrasting performance within a batch, we develop an RL-based method to optimize the SV metric by taking appropriate interpolation coefficients as actions. In Sec. 3, we further illustrate the details of our proposed framework.

3. THE LC4SV SYSTEM

Fig. 1 shows the overall workflow of our proposed LC4SV system. First, referring to VoiceID loss [25], we use the SV objective produced by a proxy SV model to train our base SE unit. The SE unit converts a noisy speech signal S into an enhanced signal $\hat{S}$. Next, we train our interpolation agent in an RL manner using an SV-customized reward scheme. The noisy speech signal and the corresponding enhanced speech signal are concatenated as a two-channel input and fed into the interpolation agent, which determines the appropriate interpolation coefficient α. Finally, we use α to generate a new speech signal $\tilde{S}$ as input to the SV model, where $\tilde{S} = \alpha\hat{S} + (1-\alpha)S$. The details of our SE model and interpolation agent are described in Sec. 3.1 and Sec. 3.2, respectively.

3.1. Training Stages of the SE Model

We use DEMUCS [15] as our SE model, which consists of an encoder-decoder architecture with skip connections.

3.1.1. Pre-training

To provide proper initialization to train the SE model with the SV objective, referring to Alexandre et al. [15], we pre-train our SE model with a multi-resolution STFT loss to capture

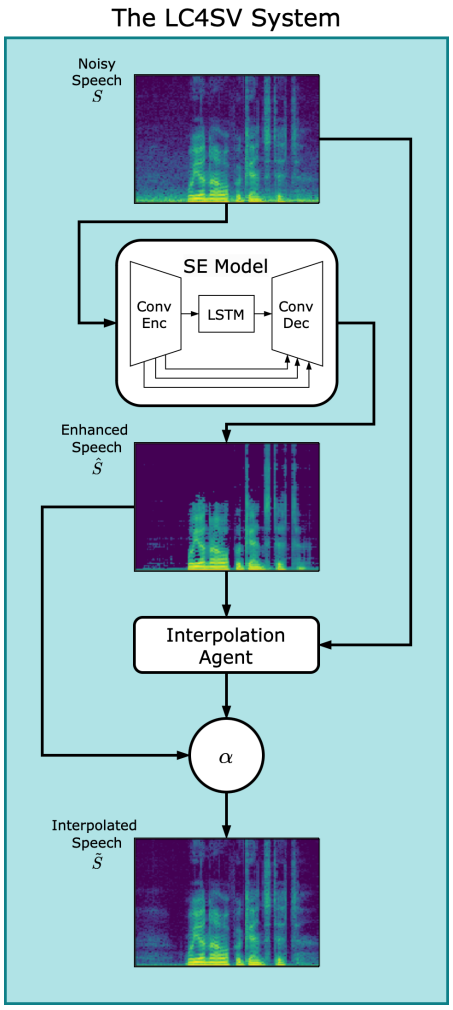


Fig. 1: The overall workflow of the LC4SV system.

information at different time-frequency resolutions. The objective function L_{PTN} is defined as:

$$L_{\text{PTN}} = ||\hat{S} - S_{\text{clean}}||_1 + \sum_{i=1}^{M} L_{\text{stft}}^{(i)}(\hat{S}, S_{\text{clean}}), \quad (1)$$

where S_{clean} is the clean target, M is the number of STFT losses with different STFT configurations, i is the configuration index, and $L_{\text{stft}}^{(i)}$ is defined as $L_{\text{sc}}^{(i)}(\hat{S}, S_{\text{clean}}) + L_{\text{mag}}^{(i)}(\hat{S}, S_{\text{clean}})$. $L_{\text{sc}}^{(i)}(\hat{S}, S_{\text{clean}})$ denotes a spectral convergence loss, and $L_{\text{mag}}^{(i)}(\hat{S}, S_{\text{clean}})$ denotes a magnitude loss. $L_{\text{sc}}^{(i)}(\hat{S}, S_{\text{clean}})$ and $L_{\text{mag}}^{(i)}(\hat{S}, S_{\text{clean}})$ are respectively defined as:

$$L_{\text{sc}}^{(i)}(\hat{S}, S_{\text{clean}}) = \frac{|||\text{STFT}^{(i)}(\hat{S})| - |\text{STFT}^{(i)}(S_{\text{clean}})|||_F}{|||\text{STFT}^{(i)}(S_{\text{clean}})|||_F} \quad (2)$$

and

$$L_{\text{mag}}^{(i)}(\hat{S}, S_{\text{clean}}) = \| \log |\text{STFT}^{(i)}(\hat{S})| - \log |\text{STFT}^{(i)}(S_{\text{clean}})| \|_1. \quad (3)$$

3.1.2. Fine-tuning

After pre-training, we fine-tune our SE model using the well-known SV objective - Angular Prototypical (AP) [28] loss. A batch of positive pairs (i.e., two speech signals from the same speaker) is sampled. Assuming there are M signals of the same speaker in a batch, one signal is randomly selected as the query signal, and the remaining $M-1$ signals are used to generate the speaker's centroid embedding. Each speaker's query embedding will be compared to all centroid embeddings to derive similarities for computing the SV objective defined as:

$$L_{\text{AP}} = -\frac{1}{N} \sum_i \log \frac{\exp(S_{i,c_i})}{\frac{1}{N} \sum_j \exp(S_{i,c_j})}, \quad (4)$$

where N is the batch size (i.e., the number of speakers in the batch), i and j are speaker indices, and S_{i,c_j} is calculated as:

$$S_{i,c_j} = w \cdot cos(x_i, c_j) + b, \quad (5)$$

where x_i is the i-th speaker's query embedding, and c_j is the j-th speaker's centroid embedding. S_{i,c_j} is a cosine-based similarity measure with learnable scale w and bias b.

In this work, there are 2 enhanced speech signals for each of the 200 speakers in a batch, i.e., $M = 2$ and $N = 200$. For each speaker, the embedding of one enhanced speech signal is used as the query, and the embedding of the other enhanced speech signal is used as the centroid. When updating our SE model, all parameters of the proxy SV model are frozen. The proxy SV model is a modified version of ResNet-34 (ResNetSE34L) trained with AP loss and provided as the first baseline checkpoint in the GitHub repository of Chung et al. [28], which is publicly available.

3.2. RL-based Interpolation Agent

The building pipeline of our interpolation agent is shown in Fig. 2. Our interpolation agent consists of three components: a waveform embedder, an SNR embedder, and a coefficient-reward predictor. The waveform embedder shares the same feature extraction architecture as the aforementioned proxy SV model, extracting content information from an entire input signal represented as a 256-dimensional embedding. The SNR embedder maps the SNR value of the noisy signal S to the corresponding SNR representation. We divide the SNR values into 6 interval groups: $[-\infty, 0)$, $[0, 3)$, $[3, 6)$, $[6, 9)$, $[9, 12)$, and $[12, \infty]$, each corresponding to a specific 256-dimensional SNR embedding. Note that the 6 SNR embeddings are part of the model parameters of the interpolation agent. We use the WADA SNR estimation algorithm [32] to predict the SNR value of the noisy signal S. The coefficient-reward predictor, which employs a multilayer perceptron module with one hidden layer (128 hidden nodes) and the LeakyReLU activation function, predicts the coefficient action based on the waveform embeddings of S and $\hat{S}$ and the

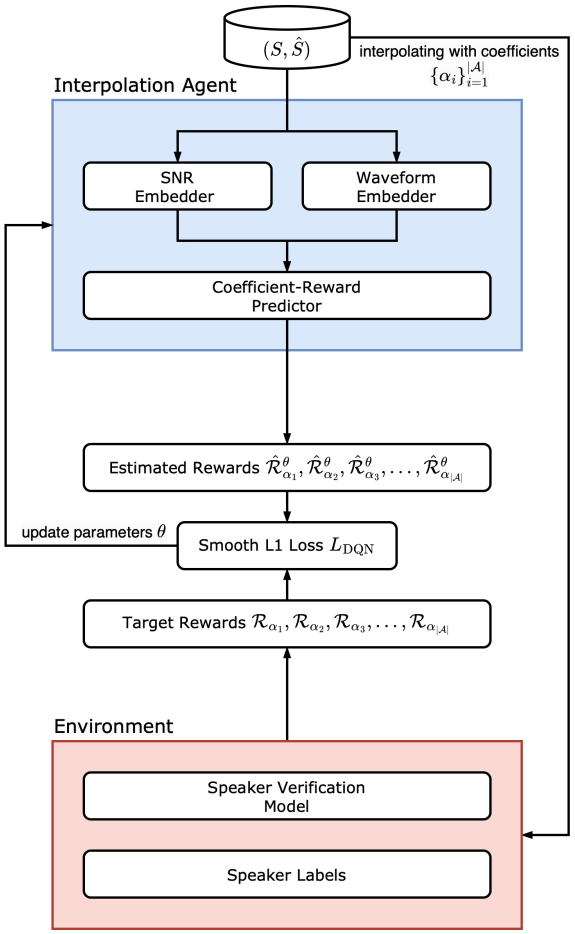


Fig. 2: The building pipeline of our interpolation agent.

SNR embedding. We denote $\mathcal{A}$ as our action set, i as the coefficient index, and α_i as its element, ranging from 0 to 1 with a step size of 0.1 (11 actions in total).

As in the fine-tuning stage of the SE model, a batch of positive pairs is sampled. Any two speech signals from different speakers within the batch will form a negative pair. The second baseline model from the GitHub repository of Chung et al. [28] is used as the SV model to build an environment that yields rewards from each combination of inputs and actions. To maximize the effectiveness of the interpolation, we aim to improve discriminability compared to directly inputting the enhanced signal. Therefore, for the noisy signal S (with speaker embedding x), the enhanced signal $\hat{S}$ (with speaker embedding $\hat{x}$), and the coefficient action α_i, the reward is defined as:

$$\begin{aligned}\mathcal{R}_{\alpha_i}(S, \hat{S}) = \cos(\tilde{x}_{\alpha_i}, \tilde{x}^{+}_{\alpha_i}) - \cos(\hat{x}, \hat{x}^{+}) \\ +\mathbb{E}_{x^{-} \in \mathcal{N}}[\cos(\hat{x}, \hat{x}^{-}) - \cos(\tilde{x}_{\alpha_i}, \tilde{x}^{-}_{\alpha_i})],\end{aligned} \tag{6}$$

where $\tilde{x}_{\alpha_i}$ is the speaker embedding of the interpolated signal $\alpha_i \hat{S} + (1 - \alpha_i) S$; $\hat{x}^{+}$ and $\tilde{x}^{+}_{\alpha_i}$ are the speaker embeddings of the enhanced and interpolated versions of the paired positive sample of S; $\mathcal{N}$ denotes the set of signals of speakers within the batch other than the speaker corresponding to x; x^{-}, $\hat{x}^{-}$, and $\tilde{x}^{-}_{\alpha_i}$ are the speaker embeddings of the noisy, enhanced, and interpolated versions of a negative sample of S. For a positive pair, the similarity between the two interpolated signals (i.e., $\cos(\tilde{x}_{\alpha_i}, \tilde{x}^{+}_{\alpha_i})$ should be higher than the similarity between the two enhanced signals (i.e., $\cos(\hat{x}, \hat{x}^{+})$), i.e., the higher the $\cos(\tilde{x}_{\alpha_i}, \tilde{x}^{+}_{\alpha_i}) - \cos(\hat{x}, \hat{x}^{+})$, the better. For a negative pair, the similarity between the two enhanced signals (i.e., $\cos(\hat{x}, \hat{x}^{-})$) should be higher than the similarity between the two interpolated signals (i.e., $\cos(\tilde{x}_{\alpha_i}, \tilde{x}^{-}_{\alpha_i})$), i.e., the higher the $\cos(\hat{x}, \hat{x}^{-}) - \cos(\tilde{x}_{\alpha_i}, \tilde{x}^{-}_{\alpha_i})$, the better.

To stabilize the training process of the interpolation agent, we choose the Q-Learning algorithm [33] to approximate the action-value function, predicting the improvements made by the selected coefficients and their interpolated signals. Given the agent's model parameters θ and the input $(S, \hat{S})$, the output is expressed as $\hat{\mathcal{R}}^{\theta}_{\alpha_i}(S, \hat{S})$. Here, we employ a smooth L1 loss to compute the distance between the predicted reward and the actual reward. The cost function is expressed as:

$$L_{\text{DQN}} = \mathbb{E}_{S,\hat{S}}[\frac{1}{|\mathcal{A}|} \sum_{i=1}^{|\mathcal{A}|} l_{\alpha_i}(S, \hat{S})], \tag{7}$$

where $l_{\alpha_i}(S, \hat{S})$ is calculated as:

$$l_{\alpha_i}(S, \hat{S}) = \begin{cases} 0.5(e_{\alpha_i}(S, \hat{S}))^2, & \text{if } |e_{\alpha_i}(S, \hat{S})| < 1 \\ |e_{\alpha_i}(S, \hat{S})| - 0.5, & \text{otherwise,} \end{cases} \tag{8}$$

where $e_{\alpha_i}(S, \hat{S})$ is calculated as $\mathcal{R}_{\alpha_i}(S, \hat{S}) - \hat{\mathcal{R}}^{\theta}_{\alpha_i}(S, \hat{S})$.

During inference, we choose α^* as our reacting action according to $\text{argmax}_{\alpha \in \mathcal{A}}\{\hat{\mathcal{R}}^{\theta}_{\alpha}(S, \hat{S})\}$ to generate the interpolated signal $\tilde{S}$ via $\alpha^* \hat{S} + (1 - \alpha^*) S$.

Table 1: Performance of different SE architectures evaluated on Libri2Mix.

	STOI	PESQ_{WB}	PESQ_{NB}	SI-SDR
DCCRN	0.8857	1.8383	2.4285	**11.9048**
DTLN	0.8577	1.6155	2.1245	10.9175
DEMUCS	**0.8978**	**1.9553**	**2.4486**	11.2817

4. EXPERIMENTS

4.1. Experimental Setup and Implementation Details

The training datasets used in the experiments include: (1) the DNS-Challenge [34] noise dataset, consisting of 65,303 background and foreground noise samples; (2) the Librispeech-960 [35] corpus, consisting of 281,241 utterances; (3) the VoxCeleb2 [29] dataset, consisting of 1,092,009 utterances from 5,994 speakers. The LC4SV framework is divided into three stages: SE model pre-training, SE model fine-tuning, and interpolation agent training. First, for SE model pre-training, the clean speech utterances from Librispeech-960 were contaminated by randomly sampled noise signals from

Table 2: EER (%) and MinDCF results on Libri2Mix, VOiCES, and the Vox1 test list.

(a) Ablation I: tested by RawNet3 (RN3) and ECAPA-TDNN (ECP)

	RN3-Libri2Mix		RN3-VOiCES		ECP-Libri2Mix		ECP-VOiCES	
method	EER	MinDCF	EER	MinDCF	EER	MinDCF	EER	MinDCF
NOISY	9.153	0.5878	2.807	0.1317	7.964	0.4020	2.703	0.1373
SE-PTN	19.631	0.7096	13.793	0.7739	15.890	0.6088	10.571	0.6244
SE-SV	9.271	**0.4843**	2.671	0.1358	7.683	**0.3715**	2.623	0.1384
SE-SV-SNR	8.877	0.4881	2.718	0.1345	7.620	0.3818	2.633	0.1368
LC4SV	**8.755**	0.4873	**2.652**	**0.1345**	**7.616**	0.3758	**2.621**	**0.1335**

(b) Ablation II: tested by baseline (BSLN)

	BSLN-SV-1		BSLN-SV-2	
method	EER	MinDCF	EER	MinDCF
NOISY	2.179	0.1691	1.018	0.0817
SE-PTN	2.731	0.2010	1.235	0.0883
SE-SV	2.132	0.1671	1.023	0.0776
SE-SV-SNR	2.190	0.1700	1.034	0.0824
LC4SV	**2.073**	**0.1652**	**1.018**	**0.0776**

Table 3: EER (%) results for different values of α tested using RawNet3 on Libri2Mix.

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	LC4SV
EER (%)	9.0235	8.9020	8.8978	8.8140	8.8122	8.8151	8.8811	8.9816	9.1070	**8.7550**

DNS-Challenge to form noisy-clean pairs for training, with SNR levels uniformly sampled from -4dB to 6dB. We used the Adam optimizer with β_1 = 0.9 and β_2 = 0.999, a learning rate of 0.0002, and a batch size of 8 with 500,000 steps. Next, with the SV objective derived by the proxy SV model, we fine-tuned the pre-trained SE model using the training data in VoxCeleb2. We used the Adam optimizer with β_1 = 0.9 and β_2 = 0.999, a learning rate of 0.0001, and a batch size of 32 with 10,000 steps. Finally, we froze the parameters of the fine-tuned SE model and combined its enhanced signals with its noisy input signals to train our interpolation agent. The training dataset was also VoxCeleb2. We used the Adam optimizer with β_1 = 0.9 and β_2 = 0.999, a learning rate of 0.0001, and a batch size of 128 with 2,000 steps.

For evaluation, we used two public datasets to verify the effectiveness of our system: LibriMix [36] and noisy VOiCES [37, 38]. LibriMix took speech utterances from LibriSpeech [35] and the noise signals from WHAM! [39]. During mixing, the SNR values were sampled from a normal distribution with a mean of 0dB and a standard deviation of 4.1dB. We tested our LC4SV system on the single-speaker dev and test sets of the Libri2Mix subset, which contains 6,000 utterances. Noisy VOiCES took speech utterances from the original VOiCES corpus, and the noisy version was produced by MultiSV [38]. During mixing, the SNR values were sampled uniformly from 3dB to 20dB. For each evaluation dataset, we further sampled all speakers to generate pair lists with the same number of positive and negative pairs. Two unseen SV models were used to evaluate our processing results, RawNet3 [40] and ECAPA-TDNN [41]. Both SV models are publicly available and trained on the full VoxCeleb training dataset with additive noise and reverberation augmentation. More details about them can be found in the GitHub repository of Chung et al. [28] and the SpeechBrain toolkit [42]. Additionally, we compared the proposed LC4SV system with several baseline SV systems using the following settings: (i) directly using the unprocessed noisy speech (termed **NOISY**); (ii) using the enhanced speech from the SE model pre-trained with the reconstruction loss (termed **SE-PTN**); (iii) using the enhanced speech from the SE model fine-tuned with the SV objective (termed **SE-SV**); (iv) selecting the noisy speech or the enhanced speech from the SE model fine-tuned with the SV objective according to a predetermined SNR threshold (termed **SE-SV-SNR**). For (iv), the SNR threshold was set to 4dB based on the results on the validation set. The SNR values in our experiments were estimated by WADA [32]. Two standardized evaluation metrics were used to evaluate the performance: Equal Error Rate (EER [%]) and Minimum Detection Cost Function (MinDCF). In MinDCF, the prior probability of a target trial P_{target} was set to 0.05.

4.2. Comparison of Different SE Models

We first validated the choice of DEMUCS as the SE model. In addition to DEMUCS, we also trained two widely used SE architectures, DTLN [43] and DCCRN [9], using the same training setup (training data and optimization pipeline) in the pretraining stage. The performance in terms of common evaluation scores [14, 16, 17] was evaluated on the LibriMix [36] test set and shown in Table 1. Three standardized evaluation metrics were used to measure SE performance, namely wide/narrow band perceptual evaluation of speech quality (PESQ_{wb}/PESQ_{nb}) [16], short-time objective intelligibility measure (STOI) [17], and scale-invariant signal-to-distortion ratio (SI-SDR) [14]. PESQ_{nb} was designed to evaluate the quality of the processed speech on a scale from -0.5 to 4.5. STOI was designed to calculate the speech intelligibility on a scale from 0 to 1. SI-SDR was designed to measure the energy ratio between speech and non-speech components. For all three metrics, higher scores indicate better performance.

The results in Table 1 indicate that while the performance of the three SE models is roughly comparable, DEMUCS exhibits superior performance. These results validate the adequacy of designating DEMUCS as the primary SE architecture for ablation studies of the proposed LC4SV method.

4.3. Ablation I: Comparison with Baselines

In the first set of ablation experiments, we compared the performance of SV systems operating on different types of

speech signals, including noisy (**NOISY**), enhanced (**SE-PTN** or **SE-SV**), either noisy or enhanced according to SNR (**SE-SV-SNR**), and interpolated (**LC4SV)** speech. From Table 2a, several observations can be drawn. First, **SE-PTN** dramatically degrades SV performance, indicating that the issue of signal artifacts caused by SE is serious. Second, **SE-SV** outperforms **NOISY** and **SE-PTN**, indicating the effectiveness of fine-tuning the SE model with the SV objective. Third, **SE-SV-SNR** and **SE-SV** both win and lose in different cases, suggesting that there are advantages and disadvantages to using noisy speech or enhanced speech in SV systems. Fourth, **LC4SV** is better than **SE-SV-SNR** in most cases, indicating that hard thresholding on the SNR value is not the best strategy for handling distortion compensation. Fifth, **LC4SV** is the best method among all compared methods. Overall, the results validate that our interpolation agent can provide appropriate coefficients to combine enhanced and noisy speech for better SV performance under noisy conditions.

4.4. Ablation II: Effectiveness of Interpolation Agent

We also evaluated the effectiveness of the interpolation agent in our LC4SV system in an in-domain setting. Here, the test set was VoxCeleb1 [29]. The SV systems used for testing were the first and second baseline models from the GitHub repository of Chung et al. [28], which represent the "seen" SV models of our LC4SV system, as they were used to fine-tune the SE model and train the interpolation agent, respectively. The results are shown in Table 2b.

From Table 2b, several observations can be drawn. First, all the SE methods except **LC4SV** degrade the SV performance of BSLN-SV-2. For **SE-PTN**, **SE-SV**, and **SE-SV-SNR**, BSLN-SV-2 is unseen because it only involves interpolation agent training. In contrast, **SE-SV** brings improvements to BSLN-SV-1 over **NOISY** since BSLN-SV-1 was seen in **SE-SV** training. Second, we again observe that **SE-PTN** degrades the performance of BSLN-SV-1 and BSLN-SV-2 due to severe artifacts in its enhanced speech. Third, **LC4SV** is the only method that consistently brings improvements over **NOISY** to both BSLN-SV-1 and BSLN-SV-2 on this in-domain SV task. Overall, the results in the in-domain experimental setup again demonstrate the effectiveness of our LC4SV approach, whose interpolation agent effectively determines appropriate coefficients to combine enhanced and noisy speech for better SV performance.

4.5. Ablation III: Comparison with Predetermined Constant Coefficients

The key feature of LC4SV is that it automatically and effectively determines the interpolation coefficient α to improve SV performance. To verify the effectiveness of the sample-by-sample estimation, we compared the performance of determining α using LC4SV with that of using a predetermined value of α ranging from 0.1 to 0.9 with a step size of 0.1. The results of testing with the RawNet3 SV model on Libri2Mix are shown in Table 3. From the table, we can notice that **LC4SV** clearly outperforms other constant coefficients, confirming that **LC4SV** can effectively estimate the interpolation coefficient.

Table 4: EER (%) and MinDCF results of RawNet3/ECAPA-TDNN tested on the LibriSpeech dev-clean/test-clean sets.

	RawNet3		ECAPA-TDNN	
method	EER	MinDCF	EER	MinDCF
CLEAN	1.496	0.0509	1.601	0.0614
SE-PTN	1.408	0.0518	1.617	0.0647
SE-SV	1.425	0.0533	1.621	0.0633
LC4SV	1.383	0.0499	1.604	0.0623

4.6. Ablation IV: Oracle Performance Analysis

In addition to evaluating the performance of our proposed approach in noisy environments, we also conducted experiments using clean speech signals to specifically examine the ability of LC4SV to mitigate artifacts without interference from natural background noise. To generate a list of verification pairs for testing purposes, we combined the dev-clean and test-clean sets of LibriSpeech. The resulting testing list contained 24,000 positive pairs and 24,000 negative pairs. The results are shown in Table 4.

Comparing the results in Table 4 and Table 2a, both SVs perform much better in clean environments than in noisy environments. The performance drops using the three type of processed signals is marginal compared to clean speech. In particular, among the three model settings, LC4SV stands out as the most effective at reducing artifacts and produces results comparable to feeding the clean speech.

5. CONCLUDING REMARKS

The over-parameterization issue limits speech enhancement models to generate optimal enhanced speech in unknown speaker verification systems. In this study, we propose a generic denoising framework named LC4SV that learns to compensate for enhanced speech for better generalization. We use a learning-based interpolation agent to automatically generate appropriate interpolation coefficients between a noisy speech input and its enhanced speech. Our interpolation agent is trained to maximize the performance of a proxy speaker verification model, preventing information loss in the predictions of speech enhancement models. Our experimental results demonstrate the effectiveness of LC4SV in mitigating the artifact-induced degradation in speaker verification performance .

6. REFERENCES

[1] Philipos C. Loizou, *Speech Enhancement: Theory and Practice*, CRC Press, Inc., USA, 2nd edition, 2013.

[2] Y. Xu, J. Du, L. Dai, and C. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2015.

[3] M. Kolbæk, Z. Tan, and J. Jensen, "Speech intelligibility potential of general and specialized deep neural network based speech enhancement systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[4] Zhuo Chen, Shinji Watanabe, Hakan Erdogan, and John R. Hershey, "Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks," in *Proc. Interspeech*, 2015.

[5] Bingyin Xia and Changchun Bao, "Wiener filtering based speech enhancement with weighted denoising auto-encoder and noise classification," *Speech Communication*, 2014.

[6] Ding Liu, Paris Smaragdis, and Minje Kim, "Experiments on deep learning for speech denoising," in *Proc. Interspeech*, 2014.

[7] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, "Speech enhancement based on deep denoising autoencoder," in *Proc. Interspeech*, 2013.

[8] Xiao-Lei Zhang and DeLiang Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 967–977, 2016.

[9] Yanxin Hu, Yun Liu, Shubo Lv, Mengtao Xing, Shimin Zhang, Yihui Fu, Jian Wu, Bihong Zhang, and Lei Xie, "Dccrn: Deep complex convolution recurrent network for phase-aware speech enhancement," in *Proc. Interspeech*, 2020.

[10] Chi-Chang Lee, Yu-Chen Lin, Hsuan-Tien Lin, Hsin-Min Wang, and Yu Tsao, "SERIL: noise adaptive speech enhancement using regularization-based incremental learning," in *Proc. Interspeech*, 2020.

[11] Chi-Chang Lee, Cheng-Hung Hu, Yu-Chen Lin, Chu-Song Chen, Hsin-min Wang, and Yu Tsao, "Nastar: Noise adaptive speech enhancement with target-conditional resampling," in *Proc. Interspeech*, 2022.

[12] Ashutosh Pandey and DeLiang Wang, "A new framework for supervised speech enhancement in the time domain.," in *Proc. Interspeech*, 2018.

[13] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1109–1121, 1984.

[14] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey, "SDR-half-baked or well done?," in *Proc. ICASSP*, 2019.

[15] Alexandre Défossez, Gabriel Synnaeve, and Yossi Adi, "Real time speech enhancement in the waveform domain," in *Proc. Interspeech*, 2020.

[16] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. ICASSP*, 2001.

[17] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *Proc. ICASSP*, 2010.

[18] Chandan KA Reddy, Vishak Gopal, and Ross Cutler, "Dnsmos: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *Proc. ICASSP*, 2021.

[19] Erica Cooper, Wen-Chin Huang, Tomoki Toda, and Junichi Yamagishi, "Generalization ability of mos prediction networks," in *Proc. ICASSP*, 2022.

[20] Chi-Chang Lee, Yu Tsao, Hsin-Min Wang, and Chu-Song Chen, "D4AM: A general denoising framework for downstream acoustic models," in *Proc. ICLR*, 2023.

[21] Yen-Ju Lu, Xuankai Chang, Chenda Li, Wangyou Zhang, Samuele Cornell, Zhaoheng Ni, Yoshiki Masuyama, Brian Yan, Robin Scheibler, Zhong-Qiu Wang, et al., "Espnet-se++: Speech enhancement for robust speech recognition, translation, and understanding," in *Proc. Interspeech*, 2022.

[22] Yen-Ju Lu, Samuele Cornell, Xuankai Chang, Wangyou Zhang, Chenda Li, Zhaoheng Ni, Zhong-Qiu Wang, and Shinji Watanabe, "Towards low-distortion multi-channel speech enhancement: The espnet-se submission to the l3das22 challenge," in *Proc. ICASSP*, 2022.

[23] Seyed Sadjadi and John Hansen, "Assessment of single-channel speech enhancement techniques for speaker identification under mismatched conditions," in *Proc. Interspeech*, 2010.

[24] Hassan Taherian, Zhong-Qiu Wang, Jorge Chang, and DeLiang Wang, "Robust speaker recognition based on single-channel and multi-channel speech enhancement,"

IEEE/ACM Transactions on Audio, Speech, and Language Processing, pp. 1293–1302, 2020.

[25] Suwon Shon, Hao Tang, and James R. Glass, "Voiceid loss: Speech enhancement for speaker verification," in *Proc. Interspeech*, 2019.

[26] Ladislav Mošner, Oldřich Plchot, Lukáš Burget, and Jan Honza Černocký, "Multi-channel speaker verification with conv-tasnet based beamformer," in *Proc. ICASSP*, 2022.

[27] Sandipana Dowerah, Romain Serizel, Denis Jouvet, Mohammad Mohammadamini, and Driss Matrouf, "Joint optimization of diffusion probabilistic-based multichannel speech enhancement with far-field speaker verification," in *Proc. SLT*, 2023.

[28] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han, "In defence of metric learning for speaker recognition," in *Proc. Interspeech*, 2020.

[29] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, p. 101027, 2020.

[30] Chang Zeng, Xiaoxiao Miao, Xin Wang, Erica Cooper, and Junichi Yamagishi, "Joint speaker encoder and neural back-end model for fully end-to-end automatic speaker verification with multiple enrollment utterances," *arXiv preprint arXiv:2209.00485*, 2022.

[31] Hiroshi Sato, Tsubasa Ochiai, Marc Delcroix, Keisuke Kinoshita, Naoyuki Kamo, and Takafumi Moriya, "Learning to enhance or not: Neural network-based switching of enhanced and observed signals for overlapping speech recognition," in *Proc. ICASSP*, 2022.

[32] Chanwoo Kim and Richard M. Stern, "Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis," in *Proc. Interspeech*, 2008.

[33] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei Rusu, Joel Veness, Marc Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis, "Human-level control through deep reinforcement learning," *Nature*, pp. 529–33, 2015.

[34] Chandan KA Reddy, Vishak Gopal, Ross Cutler, Ebrahim Beyrami, Roger Cheng, Harishchandra Dubey, Sergiy Matusevych, Robert Aichner, Ashkan Aazami, Sebastian Braun, et al., "The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results," in *Proc. Interspeech*, 2020.

[35] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. ICASSP*, 2015.

[36] Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent, "Librimix: An open-source dataset for generalizable speech separation," in *Proc. Interspeech*, 2020.

[37] Mahesh Kumar Nandwana, Julien van Hout, Colleen Richey, Mitchell McLaren, Maria Alejandra Barrios, and Aaron D. Lawson, "The voices from a distance challenge 2019," in *Proc. Interspeech*, 2019.

[38] Ladislav Mošner, Oldřich Plchot, Lukáš Burget, and Jan Černocký, "Multisv: Dataset for far-field multi-channel speaker verification," in *Proc. ICASSP*, 2022.

[39] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux, "Wham!: Extending speech separation to noisy environments," *arXiv preprint arXiv:1907.01160*, 2019.

[40] Jee-weon Jung, You Jin Kim, Hee-Soo Heo, Bong-Jin Lee, Youngki Kwon, and Joon Son Chung, "Pushing the limits of raw waveform speaker recognition," in *Proc. Interspeech*, 2022.

[41] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Interspeech*, 2020.

[42] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio, "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[43] Nils Westhausen and Bernd Meyer, "Dual-signal transformation lstm network for real-time noise suppression," in *Proc. Interspeech*, 2020.