IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 March 2024, accepted 17 May 2024, date of publication 21 May 2024, date of current version 30 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3403761

RESEARCH ARTICLE

Knowledge Distillation-Based Training of Speech Enhancement for Noise-Robust Automatic Speech Recognition

GEON WOO LEE[1], (Graduate Student Member, IEEE), HONG KOOK KIM[1,2,3], (Senior Member, IEEE), AND DUK-JO KONG[1], (Member, IEEE)

[1]AI Graduate School, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea
[2]School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea
[3]AunionAI Company Ltd., Gwangju 61005, Republic of Korea

Corresponding author: Duk-Jo Kong (dukjokong@gist.ac.kr)

This work was supported in part by Gwangju Institute of Science and Technology (GIST)-Massachusetts Institute of Technology (MIT) Research Collaboration funded by GIST, in 2024; and in part by the ''Project for Science and Technology Opens the Future of the Region'' Program through the Innopolis Foundation funded by Ministry of Science and ICT under Project 2022-DD-UP-0312.

ABSTRACT This paper addresses the training issues associated with neural network-based automatic speech recognition (ASR) under noise conditions. In particular, conventional joint training approaches for a pipeline comprising speech enhancement (SE) and end-to-end ASR model surfer from a conflicting problem and a frame mismatched alignment problem because of different goals and different frame structures for ASR and SE. To mitigate such problems, a knowledge distillation (KD)-based training approach is proposed by interpreting the ASR and SE models in the pipeline as teacher and student models, respectively. In the proposed KD-based training approach, the ASR model is first trained using a training dataset, and then, acoustic tokens are generated via K-means clustering using the latent vectors of the ASR encoder. Thereafter, KD-based training of the SE model is performed using the generated acoustic tokens. The performance of the SE and ASR models is evaluated on two different databases, noisy LibriSpeech and CHiME-4, which correspond to simulated and real-world noise conditions, respectively. The experimental results show that the proposed KD-based training approach yields a lower character error rate (CER) and word error rate (WER) on the two datasets than conventional joint training approaches, including multi-condition training. The results also show that the speech quality scores of the SE model trained using the proposed training approach are higher than those of SE models trained using conventional training approaches. Moreover, the noise reduction scores of the proposed training approach are higher than those of conventional joint training approaches but slightly lower than those of the standalone-SE training approach. Finally, an ablation study is conducted to examine the contribution of different combinations of loss functions in the proposed training approach to SE and ASR performance. The results show that the combination of all loss functions yields the lowest CER and WER and that tokenizer loss contributes more to SE and ASR performance improvement than ASR encoder loss.

INDEX TERMS Noise-robust automatic speech recognition, speech enhancement, knowledge distillation, teacher-student model, acoustic tokenizer, K-means clustering.

I. INTRODUCTION

Recent advancements in neural network architectures and training approaches have demonstrated consistent progress, enhancing capabilities in not only image and natural language processing but also audio and speech signal processing.

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

Speech processing encompasses the comprehensive analysis, synthesis, and recognition of speech, including speaker verification, speech separation, speech enhancement (SE), speech synthesis, and automatic speech recognition (ASR) [1], [2]. ASR has gained considerable attention, particularly in voice-based information retrieval systems, chatbots, and automated transcription systems [3]. Moreover, the interest in ASR, specifically for real-world scenarios, such as closed captioning in social media video content [4] and real-time interaction between users in videoconferencing [5], is increasing.

Conventional ASR systems have a modular structure comprising three distinct components: a feature extractor that captures spoken signal characteristics, an acoustic model that converts extracted features into linguistic units, and a language model that incorporates grammar, lexicon, and related linguistic information [6]. Compared with conventional ASR systems, ASR systems based on end-to-end (E2E) neural network architectures achieve state-of-the-art performance by employing a sequence-to-sequence training approach to transform speech signals into their corresponding text sequences.

However, E2E-based ASR models typically face performance degradation in distant microphone settings or under low signal-to-noise ratio (SNR) conditions because of the distortion of speech signals by real-world ambient noise [7], [16]. To address this issue, many studies have integrated SE models into ASR systems [17], [18], [19], [20], [21]. Conventional SE models, designed primarily for voice communications, have improved ASR robustness against background noise. Notably, deep learning-based SE models, particularly U-Net-based architectures and their evolved versions, such as convolutional recurrent neural networks (RNNs) with long short-term memory layers and a deep complex convolutional recurrent network (DCCRN) that processes complex spectra [17], [19], have outperformed conventional statistical approaches in enhancing speech quality in noisy environments.

Nevertheless, these SE models may introduce unintended artifacts into the enhanced speech signals. These artifacts can create mismatched conditions for ASR, thereby degrading ASR performance [20]. This occurs because the SE model is trained without regard to the ASR model. To address the artifact issue when using an SE model as a preprocessor for an ASR model, a multi-condition training (MCT) approach can be employed to train the ASR model [20], [21]. This involves including both noisy speech signals and the corresponding enhanced speech signals in the training dataset, accounting for the artifacts in the enhanced speech. According to research, this MCT approach improves ASR performance better than training an ASR model solely on noisy speech signals. However, the improvement achieved by the MCT approach is limited because the artifacts in enhanced speech are not predictable [22].

As an alternative, a pipeline integrating SE and ASR models has been explored in a joint optimization framework [22], [23]. In this approach, the SE and ASR models are treated as front- and back-end modules, respectively. When optimizing the entire pipeline, challenges arise because of conflicting gradients, leading to a convergence issue, which is called a conflicting problem [23], [24]. To solve this conflicting problem, several training approaches have been studied, such as the asynchronous subregion optimization (ASO)-based approach [25], [26] and the gradient surgery-based approach [24], [27]. Although these approaches yield promising results, they suffer from frame mismatching between SE and ASR [28], which mainly stems from the different objectives of SE and ASR. In other words, the SE model aims to reconstruct a signal within a short frame length, whereas the ASR model aims to predict a sequence of words or characters that is considerably longer than the frame length used in the SE model. Although the SE and ASR models operate on different frame lengths, the joint loss for training the combined model of SE and ASR can be computed according to the length of the SE frame [17], [27].

Therefore, to mitigate this frame mismatched alignment problem for a pipeline comprising SE and ASR models, this paper proposes an approach based on the student–teacher model. In other words, the ASR and SE models correspond to the teacher and student models, respectively. Therefore, the ASR model, which will serve as a teacher model for the SE model, is first trained using a training dataset and then frozen. Thereafter, a loss function is proposed to train the SE model using the output of the teacher model. In particular, as the ASR model is constructed with an encoder–decoder structure, a frame output of the ASR encoder is clustered via K-means clustering to obtain the pseudo-label for the current frame. Next, using the proposed loss function, the SE model is trained via knowledge distillation (KD). The main contributions of this paper can be summarized as follows:

- A pipeline comprising SE and ASR models is interpreted as a teacher–student model, and the output of the ASR model is used to train the SE model. To the best of our knowledge, this is the first study to present a teacher–student model with two models having different objectives.
- A new loss function is proposed for training the student SE model by introducing acoustic tokens via K-means clustering.
- The proposed training approach improves the performance of both the ASR and SE models compared with conventional joint training approaches.

The remainder of this paper is organized as follows. Section II briefly reviews E2E-based ASR architectures and the methodologies of KD approaches applied to ASR. Section III explains the neural architecture of the proposed pipeline comprising the SE and ASR models, which is

interpreted as a teacher–student model. Section IV proposes an acoustic-tokenizer-based loss function for training the SE model to improve the ASR performance. Section V evaluates the performance of the proposed KD-based training approach on two datasets, noisy LibriSpeech and CHiME-4, which include simulated and real-world noisy speeches, respectively. In addition, the performance of the SE and ASR models trained using the proposed training approach is compared with those trained using conventional joint training approaches. Moreover, an ablation study is conducted to examine SE and ASR performance according to different combinations of loss functions in the proposed training approach. Finally, Section VI concludes this paper.

II. RELATED WORKS ON KD-BASED APPROACH FOR ASR

The proposed KD-based training approach primarily explores the leveraging of linguistic information from the E2E-based ASR model to the SE model. Therefore, this section reviews related works on E2E-based ASR model architectures and KD-based approaches for ASR.

A. E2E-BASED ASR ARCHITECTURES

E2E-based ASR models typically adopt an encoder–decoder structure: the encoder combines the feature extractor and the acoustic model; the decoder corresponds to the language model [7], [8]. The encoder and decoder are connected through an attention mechanism and jointly trained, resulting in better performance than traditional modular-structured ASR systems with individually trained components. To realize an E2E-based ASR model, RNNs were used for both the encoder and decoder [9]. However, the ability of this RNN-based ASR model to handle long-term dependencies and contextual alignments between speech and text was limited because of the global attention mechanism [10]. Thus, instead of using RNNs in the encoder–decoder structure, a transformer-based architecture was adopted to mitigate long-term dependencies; however, it failed to capture local speech contexts [11].

Consequently, cutting-edge ASR systems transitioned to using conformers [12] or ContextNet [13] rather than conventional transformers for the encoder–decoder structure. In contrast, some ASR models were diversely designed using distinct architectures for the decoder constructed using either connectionist temporal classification (CTC) [14] or neural transducers [15]. Among various decoder structures, the neural transducer exhibited superior performance because of its capacity to overcome the challenges inherent in the conditional independence assumption in other decoders [12]. This performance was further enhanced when the RNN was replaced with a conformer, resulting in a conformer–transducer structure that captured both global and local contextual information well.

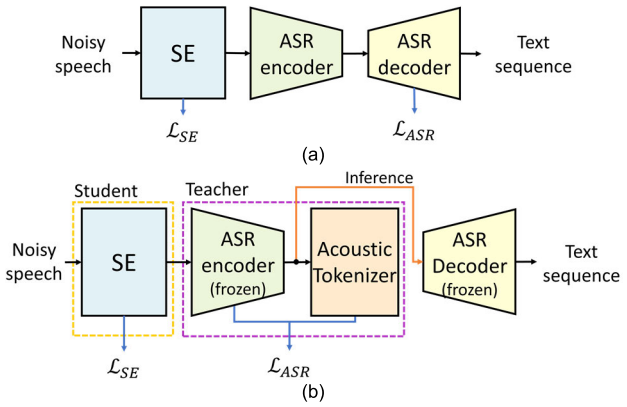

FIGURE 1. Block diagrams of a pipeline comprising SE and ASR models for (a) joint training approach and (b) the proposed KD-based training approach.

B. KD FOR E2E-BASED ASR MODEL

Conventional KD has been used to compress large models into small models or to improve performance by leveraging a teacher model [29], [30]. Moreover, KD has demonstrated successful performance improvement in multitask learning with different objective functions [31], [32]. When applying KD to classification tasks, the gradient landscape can be smoothed because of the temperature parameter, which can help different gradients converge to the global optimum. To transfer knowledge, the distance between the latent vectors of the teacher and student models should be minimized, and the teacher model is guided to converge well through the probability distribution for the target class. In a regression task, KD is performed using only vector similarity, and a successful result is less likely because it is difficult to transfer the probability distribution [33], [34].

Several studies have leveraged linguistic and acoustic information from pretrained ASR models to enhance ASR performance [35], [36], [37]. The KD approach using pretrained ASR models can be categorized into decoder- and encoder-side methods [36]. The decoder-side KD method, which uses the output vector of the ASR decoder, transfers the global context to the student model, because the decoder output vector integrates the latent vectors from the ASR encoder with a large context window to capture the linguistic unit information. Thus, the decoder-side KD method maximizes transition probabilities between linguistic units, resulting in its inability to handle the frame-wise characteristics inherent in the ASR encoder [36], [37]. Conversely, the encoder-side KD method uses the output vector of the ASR encoder optimized in a frame-wise fashion with a fixed-size sliding window. Thus, it focuses on improving the relationship between speech frames and their linguistic information without considering contextual information [35], [36].

Therefore, this paper attempts to benefit from both encoder-side and decoder-side KD methods. To this end, the encoder-side KD method [35] is primarily used. In addition,

an acoustic tokenizer is proposed and incorporated into the encoder-side KD method so that frame-wise information is converted into segment-wise information to capture a certain degree of global information, as in the decoder-side KD method.

III. INTERPRETATION OF PIPELINE COMPRISING SE AND ASR AS TEACHER–STUDENT MODEL

Fig. 1(a) illustrates a conventional pipeline comprising SE and ASR models for joint training [21], [22]. As mentioned in Section I, conventional joint training approaches combine speech quality loss, $\mathcal{L}_{SE}$, and ASR loss, $\mathcal{L}_{ASR}$, and then jointly or asynchronously train the pipeline.

In contrast, this paper interprets the pipeline as a combination of a student (SE) model and a teacher (ASR encoder) model, as depicted in Fig. 1(b). To align the different goals of the SE and ASR models, i.e., predictions of clean speech and text, respectively, the ASR model is first trained using a noisy speech training dataset and then frozen. After the ASR model is frozen, a teacher model is constructed by concatenating the ASR encoder and an acoustic tokenizer, as shown in Fig. 1(b). Because the output of the ASR encoder corresponds to an acoustic token for a given input frame [26], [36], this acoustic tokenizer serves as a surrogate model of the ASR decoder to extract linguistic information at frame-wise granularity. To train the acoustic tokenizer, the output vector of the ASR encoder applied to a clean speech dataset is clustered using K-means clustering to compute pseudo-labels. Compared with conventional training approaches, an acoustic tokenizer loss is proposed and combined with the ASR encoder loss in the proposed training approach, which will be explained in the next section.

In this paper, the DCCRN-based SE model and conformer (encoder)–transducer (decoder)-based ASR model are employed. For a fair comparison, the architecture and hyperparameters of the DCCRN-based SE model and conformer–transducer(s)-based ASR model are set identically to those in [12] and [19], respectively.

IV. PROPOSED KD-BASED SE MODEL TRAINING APPROACH USING ACOUSTIC TOKENIZER

This section presents the procedure of the proposed KD-based training approach for the SE model. As mentioned in Section III, the conformer–transducer-based ASR model is already trained. Next, a teacher model is constructed by adding an acoustic tokenizer, as depicted in Fig. 2(a). This acoustic tokenizer is trained using clean speech signals from the training dataset used for ASR model training. Thereafter, a pipeline is constructed, as shown in Fig. 2(b). The DCCRN-based SE model is randomly initialized using Xavier initialization [38]. For a given set of clean utterances and their noisy versions, a noisy utterance is processed using the SE model, followed by the ASR encoder and acoustic tokenizer, whereas a clean utterance is processed using only the ASR encoder and acoustic tokenizer. During this process, three different losses are computed: negative SNR loss, $\mathcal{L}_{NSNR}$; ASR encoder loss, L_{Enc}; tokenizer loss, L_{Token}. Finally, the SE model is trained via backpropagation using these losses.

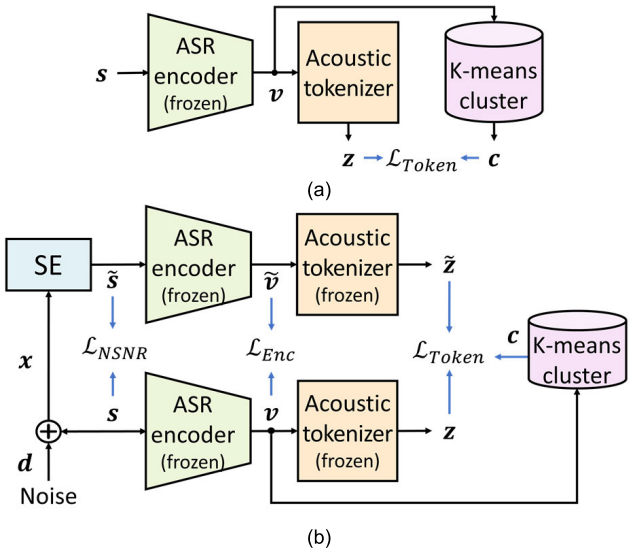

FIGURE 2. Block diagram of the proposed KD-based training procedure for (a) the acoustic tokenizer and (b) the SE model.

A. ACOUSTIC TOKENIZER

A clean speech utterance sample, $\boldsymbol{s}$, sampled at 16 kHz is segmented into consecutive frames every 25 ms with an overlap length of 16 ms, resulting in $\boldsymbol{s} = \{s_1, s_2, \cdots, s_N\}$. Here, $s_n \in \mathbb{R}^{N_s}$ with $N_s = 400$, and N represents the total number of frames in $\boldsymbol{s}$. $\boldsymbol{s}$ is then inputted into the ASR encoder, $Enc(\cdot)$, yielding the output sequence $\boldsymbol{v} = Enc(\boldsymbol{s}) = \{v_1, v_2, \cdots, v_M\}$, where $v_m (\in \mathbb{R}^{N_v})$ denotes the m-th latent vector whose dimension is $N_v = 144$. To accelerate the training and inference, subsampling layers are employed in the ASR encoder to reduce the frame rate by a factor of four so that $M = \lfloor N/4 \rfloor$.

Next, each v_m is coded into a one-hot cluster vector, $c_m \in \{0, 1\}^{N_c}$. Here, the number of clusters, N_c, is set to 1.5k because the ASR model is trained to include 1k linguistic units generated using the unigram language model algorithm. In addition to these units, the training dataset contains acoustic noise, such as short pauses, breathing, or cough sounds. To obtain K-means clusters, the mini-batch K-means algorithm in the scikit-learn [39] package is applied to a pool of $\boldsymbol{v}$, which is obtained from all clean speech utterances in the training dataset after removing latent vectors corresponding to silent frames. In this work, a voice activity detection technique is applied to clean utterances with a 40-dB cutoff amplitude level [40] to detect silent frames.

Finally, v_m is tokenized into logits $z_m \in \mathbb{R}^{N_c}$ such that $\boldsymbol{z} = Tokenizer(\boldsymbol{v}) = \{z_1, z_2, \cdots, z_M\}$, where $Tokenizer(\cdot)$ is constructed using a time-distributed layer and N_c denotes

the number of clusters. To train the tokenizer, a tokenizer loss function, $\mathcal{L}_{Token}(\boldsymbol{z}|\boldsymbol{c})$, is proposed between the logit vectors, $\{z_m\}$, and cluster vectors, $\{c_m\}$, and is expressed as follows:

$$\mathcal{L}_{Token}(\boldsymbol{z}|\boldsymbol{c}) = \sum_{m=1}^{M} \log\left(\frac{e^{z_{m,i}/\tau}}{\sum_{j=1}^{N_c} e^{z_{m,j}/\tau}}\right) \tag{1}$$

where $z_{m,i}$ denotes the i-th element of z_m at which $c_{m,i} = 1$, and τ $(= 0.5)$ denotes the temperature parameter.

B. SE MODEL TRAINING

To train the SE model using information about the ASR encoder via KD, noisy utterances are generated by mixing $\boldsymbol{s}$ with a noise signal, $\boldsymbol{d}$, such that $\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{d} = \{x_1, x_2, \cdots, x_N\}$. As depicted in Fig. 2(b), $\boldsymbol{x}$ is inputted into the SE model to predict the estimated clean utterances, $\tilde{\boldsymbol{s}}$. Subsequently, the clean and estimated clean utterances are input into the ASR encoder to obtain two sequences of latent vectors: $\boldsymbol{v} = Enc(\boldsymbol{s})$ and $\tilde{\boldsymbol{v}} = Enc(\tilde{\boldsymbol{s}})$. The latent vectors are then encoded using the tokenizer, such as $\mathbf{z} = Tokenizer(\boldsymbol{v})$ and $\tilde{\mathbf{z}} = Tokenizer(\tilde{v})$. Using the cluster vectors for $\boldsymbol{v}$, a tokenizer loss function conditioned by $\boldsymbol{c}$, $\mathcal{L}_{Token}(\tilde{\mathbf{z}}|\boldsymbol{c})$, is defined as follows:

$$\mathcal{L}_{Token}(\tilde{\mathbf{z}}|\boldsymbol{c}) = \sum_{m=1}^{M} \log\left(\frac{e^{\tilde{\mathbf{z}}_{m,i}/\tau}}{\sum_{j=1}^{N_c} e^{\tilde{\mathbf{z}}_{m,j}/\tau}}\right) \tag{2}$$

where $\tilde{\mathbf{z}}_{m,i}$ denotes the i-th element of $\tilde{\mathbf{z}}_m$ with $c_{m,i} = 1$, as in (1).

In addition to (2), two different loss functions are used to improve speech quality and ASR encoder performance. Speech quality loss is defined as the negative SNR (NSNR) loss by comparing the speech quality between the clean utterance, $\boldsymbol{s}$, and its estimated version, $\tilde{s}$, as follows:

$$\mathcal{L}_{NSNR}(\boldsymbol{s}, \tilde{\boldsymbol{s}}) = -\sum_{n=1}^{N} 10\log_{10}\left(\frac{\|s_n\|^2}{\|s_n - \tilde{s}_n\|^2}\right). \tag{3}$$

The ASR encoder loss is defined as the L_2-norm between two latent vector sequences from $\boldsymbol{s}$ and $\tilde{s}$andisexpressed as follows:

$$\mathcal{L}_{Enc}(\boldsymbol{v}, \tilde{\boldsymbol{v}}) = \sum_{m=1}^{M} \|v_m - \tilde{v}_m\|^2. \tag{4}$$

Finally, the joint loss function for training the SE model is obtained by combining the loss functions in (2)–(4):

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{NSNR}(\boldsymbol{s}, \tilde{\boldsymbol{s}}) + \beta \cdot \mathcal{L}_{Enc}(\boldsymbol{v}, \tilde{\boldsymbol{v}}) + \gamma \cdot \mathcal{L}_{Token}(\tilde{\mathbf{z}}\,|\,\boldsymbol{c}) \tag{5}$$

where α, β, and γ denote the weights of the NSNR, ASR encoder, and tokenizer losses, respectively. Instead of conducting an exhaustive search to determine the values of α, β, and γ, we utilize a grid search-based approach. First, we fix two out of the three weights at 1.0, adjusting the remaining one from 0.1 to 1.0 in steps of 0.1, and measure the word error rate (WER) for each variation. This procedure is replicated for each of the weights, yielding a total of 30 different combinations. Among these, the combination that results in the lowest WER on the validation dataset is selected. Since the selected combination includes two weights fixed at 1.0, we conduct the grid search again to determine one of the two previously undetermined weights. In this second step, we select the weight that demonstrates the lowest WER, leaving the other weight still to be determined. Finally, we perform the grid search again by using the two previously determined weights and varying the remaining weight from 0.1 to 1.0 in steps of 0.1. As a result, α, β, and γ are set to 0.3, 0.7, and 1.0, respectively. Then, the loss function in (5) with these weights is used to train the SE model.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed KD-based training approach for noise-robust ASR and compare it with the MCT and conventional joint training approaches, including ASO-based joint optimization [25] and gradient-remedy-based joint optimization [24]. The ASR and SE performances were measured using two different datasets.

A. DATASETS

We conducted training and evaluation using two different datasets. The first dataset comprised the artificially mixed LibriSpeech speech corpus [41] with the deep noise suppression (DNS) challenge database [42], which includes various types of acoustic noise. The second dataset was the computational hearing in the multisource environment-4 (CHiME-4) database [43], which was curated for speech recognition in real-world scenarios.

1) NOISY LIBRISPEECH DATASET

The ASR model and tokenizer were trained using the train-960 set in the LibriSpeech database. To simulate various noise conditions, the DNS database released in the third DNS challenge was used, which comprises approximately 150 different noise types. The noisy speech utterances were then obtained by mixing the clean speech utterances from the LibriSpeech database with the noise signals from the DNS database. The mixing ratio between the clean speech and noise was controlled such that the SNR ranged from -5 to 5 dB to simulate different SNR conditions.

As a development dataset, the 10-h clean speech utterances from the dev-clean and dev-other sets in the LibriSpeech database were mixed with DNS noise. To evaluate the trained models, the 10-h clean test utterances from the test-clean and test-other sets in the LibriSpeech database were also mixed with DNS noise. Note that the development and test datasets, including dev-clean, dev-other, test-clean, and test-other, were noisy.

2) CHIME-4 DATASET

The CHiME-4 database comprises both real and simulated noisy speech utterances [43]. The real speech data were obtained by recording clean speech data in real-world

environments using a six-microphone array such as a bus, cafe, pedestrian area, and street junction. Conversely, the simulated speech data were generated by convolving six-channel room impulse responses with noisy utterances mixed with clean speech data and real-world noise.

The CHiME-4 database was divided into three different datasets: training, development, and test. The training dataset comprised 1,600 recorded and 7,138 simulated speech utterances, with 4 and 83 speakers for the real recordings and simulated datasets, respectively. The development dataset comprised 1,640 real and 1,640 simulated speech utterances denoted as ''dt05_real'' and ''dt05_simu,'' respectively. Note that the speakers in dt05_real and dt05_simu differed from those in the training dataset. To evaluate the performance of the proposed training approach, a test dataset was used, comprising 1,320 real and 1,320 simulated speech utterances denoted as ''et05_real'' and ''et05_simu,'' respectively. The speakers in the test dataset were excluded from the training and development datasets.

B. HYPERPARAMETERS

1) SE AND ASR MODEL ARCHITECTURES

The architecture and hyperparameters of the DCCRN-based SE model were set identically to those in [19]. Specifically, the input feature was a complex spectrum obtained by applying a 512-point short-time Fourier transform to each noisy speech frame with a frame size of 25 ms and a frame hop size of 16 ms. The number of complex convolutional blocks for the encoder and decoder was set to six each, and these six convolutional blocks had varying numbers of channels, such as [32, 64, 128, 128, 256, 256], with a kernel size of 5×2 and stride size of 2×1.

In addition, the architecture and hyperparameters of the conformer–transducer-based ASR model were set identically to those in the conformer(s) described in [12]. As the input feature for the ASR model, an 80-dimensional log-mel spectrum was extracted. The ASR encoder was composed of 16 conformer blocks, each of which provided a latent vector with a dimension of 144 (N_v). As a target feature, the linguistic units for transcribing target texts comprised a special token and 1k linguistic units generated using the unigram language model algorithm [44].

2) PIPELINE TRAINING AND IMPLEMENTATION

As mentioned in the previous section, two different datasets were used to evaluate the performance of the ASR and SE models under simulated and real-world noise conditions. For the simulated noise condition, the ASR and SE models were trained using the noisy LibriSpeech dataset, whereas the tokenizer was trained using a clean LibriSpeech training dataset. In this paper, the Adam optimizer was applied to all model training. To adjust the learning rate, the warmup learning rate scheduler technique with 40,000 warmup steps was used to train the conformer–transducer-based ASR model, whereas a plateau learning rate scheduler with patience of 5 and a factor of 0.5 was used for the acoustic tokenizer and SE model training. In particular, the SpecAugment technique was employed for ASR model training.

Next, for the real-world noise condition, the ASR and SE models were trained using the CHiME-4 dataset. Because of the scarcity of training data in the CHiME-4 dataset, random initialization of the ASR model parameters could not guarantee acceptable ASR performance. Thus, for training using the CHiME-4 dataset, the ASR model was initialized using the parameters from the ASR model trained on the noisy LibriSpeech dataset. Except for the above training, all training parameters for the ASR, SE, and acoustic tokenizer models were identically set to those used for the noisy LibriSpeech dataset.

All experiments were implemented in Python 3.8.10 using TensorFlow 2.11.0 [45] and conducted on an Intel(R) Xeon(R) Gold 6226R workstation with four sets of Nvidia RTX 3090.

C. SPEECH RECOGNITION PERFORMANCE

ASR performance was evaluated by measuring the character error rate (CER) and WER for both the development and test datasets. The CERs and WERs of the ASR model trained using the proposed training approach were compared with those of models trained using six different approaches: (1) an ASR model trained by MCT using clean and noisy training datasets (denoted as MCT-noisy); (2) an SE model trained using clean and noisy speech training datasets, where the enhanced signal was subsequently fed into the MCT-noisy ASR model (denoted as MCT-noisy+standalone-SE); (3) an ASR model trained by MCT using clean and noisy speech, with its enhanced version by the standalone-SE (denoted as MCT-all); (4) a combination of the SE and ASR models trained using a conventional joint optimization approach (denoted as Joint-Straight) [22]; (5) a joint pipeline trained using ASO-based joint optimization (denoted as Joint-ASO) [25]; (6) a joint pipeline trained using gradient-remedy-based joint optimization (denoted as Joint-Grad) [24].

Table 1 compares the average CERs and WERs of the ASR models trained using different training approaches. The performance evaluation was performed using four different noisy LibriSpeech datasets: dev-clean, dev-other, test-clean, and test-other. First, the effect of the SE model on ASR performance was investigated when MCT was applied. As shown in the first three rows of the table, the average CERs and WERs of the MCT-noisy+standalone-SE model increased because the standalone-SE unexpectedly distorted speech, whereas it improved the speech quality, as discussed in the next subsection. However, by adding enhanced speeches to the training dataset, the CERs and WERs of the MCT-all model were marginally reduced for all datasets compared with those of the MCT-noisy model. This was because

TABLE 1. Comparison of the average CERs and WERs of ASR models trained using different training approaches on the noisy LibriSpeech datasets.

Training Approach	CER (%)					WER (%)				
	Development		Test		Avg.	Development		Test		Avg.
	dev-clean	dev-other	test-clean	test-other		dev-clean	dev-other	test-clean	test-other	
MCT-noisy	12.87	12.93	13.04	13.10	12.98	22.77	23.18	22.95	23.41	23.08
+ standalone-SE	16.83	16.83	16.96	16.90	16.88	28.94	29.03	29.38	29.14	29.12
MCT-all	12.71	12.71	13.03	13.06	12.88	22.61	22.74	22.68	22.82	22.71
Joint-Straight [22]	12.51	12.51	12.37	12.43	12.46	22.39	22.51	22.40	22.64	22.49
Joint-ASO [25]	12.33	12.33	12.37	12.31	12.30	22.31	22.42	22.37	22.58	22.42
Joint-Grad [24]	11.18	11.48	11.94	12.01	11.65	20.11	20.89	20.88	20.98	20.72
Proposed	**11.02**	**11.24**	**11.43**	**11.42**	**11.28**	**19.86**	**20.40**	**20.28**	**20.67**	**20.30**

TABLE 2. Comparison of the average CERs and WERs of ASR models trained using different training approaches on the CHiME-4 datasets.

Training Approach	CER (%)					WER (%)				
	Development		Test		Avg.	Development		Test		Avg.
	dt05_simu	dt05_real	et05_simu	et05_real		dt05_simu	dt05_real	et05_simu	et05_real	
MCT-noisy	10.61	10.63	11.27	16.02	12.13	22.60	23.06	23.57	31.00	25.06
+ standalone-SE	15.51	15.80	19.38	21.42	18.03	31.19	32.73	33.58	36.23	33.43
MCT-all	10.49	10.37	11.18	15.98	12.01	21.42	22.93	23.37	30.98	24.68
Joint-Straight [22]	10.33	10.31	10.37	15.66	11.67	21.36	21.81	22.08	29.41	23.67
Joint-ASO [25]	10.25	10.23	10.22	15.48	11.55	21.21	21.67	21.91	29.26	23.54
Joint-Grad [24]	9.53	9.77	9.91	14.81	11.01	20.87	21.03	21.23	28.62	22.94
Proposed	**9.32**	**9.53**	**9.81**	**14.53**	**10.80**	**20.75**	**20.98**	**21.20**	**28.41**	**22.84**

the mismatch between the training and test conditions was mitigated.

Next, the average CERs and WERs of the ASR models were compared on the basis of the different joint training approaches. Note that the training hyperparameters used for Joint-Straight, Joint-ASO, and Joint-Grad were set identically to those in the corresponding papers. As shown in the fourth to sixth rows of the table, the ASR model trained using the Joint-Grad model exhibited the lowest CERs and WERs among the three ASR models. An ASR model was trained using the proposed approach. Compared with the MCT-noisy model, the ASR model trained using the proposed training approach relatively reduced the average CER and WER by 13.15% and 12.03%, respectively, compared with the ASR models trained using the noisy LibriSpeech development and test datasets. Furthermore, the model trained using the proposed training approach achieved the lowest CER and WER among the ASR models in Table 1. Specifically, it relatively reduced the average CER and WER by 3.23% and 2.00%, respectively, compared with the ASR model trained using Joint-Grad, which exhibited the best performance among all models trained using the conventional training approaches.

In addition to performance evaluation on the noisy LibriSpeech datasets, we repeated the experiment using CHiME-4 datasets to examine the effectiveness of the proposed training approach on real-world speech recording. Similar to Table 1, Table 2 compares the average CERs and WERs of the ASR models trained using different training approaches. Compared with the CERs and WERs shown in Table 1, the relative reduction in the average CERs and WERs according to the different training approaches shown in Table 2 showed a similar tendency even though the evaluation was performed using real-world recording data. In other words, the standalone-SE model negatively affected ASR performance when MCT training was used. In addition, Joint-Grad was the best among the conventional joint training approaches. As expected, the proposed training approach also outperformed Joint-Grad. In particular, the ASR model trained using the proposed training approach relatively reduced the average CER and WER by 1.89% and 0.45%, respectively, compared with the ASR model trained using Joint-Grad.

D. SPEECH ENHANCEMENT PERFORMANCE

The speech quality and noise reduction performance of the SE model trained using the proposed training approach were compared with those of the SE models trained using the conventional training approaches. In this regard, five different speech quality metrics were measured: the perceptual evaluation of speech quality (PESQ) [46], short-time objective intelligibility (STOI) [47], and three mean opinion scores, namely, signal distortion (CSIG), background noise intrusiveness (CBAK), and overall signal quality (COVL) [48]. In addition, to evaluate noise reduction quality, five metrics, signal-to-distortion ratio (SDR), signal-to-interference ratio (SIR), signal-to-artifact ratio (SAR) [49], segmental SNR (SSNR), and scale-invariant SNR (SISNR), were evaluated [48].

Table 3 compares the speech and noise reduction quality of the SE models trained using different training approaches and evaluated on test-clean. As shown in the

TABLE 3. Comparison of speech quality and noise reduction quality scores of SE models trained using different training approaches on Test-Clean in the noisy LibriSpeech dataset.

Training Approach	Speech Quality					Noise Reduction Quality				
	PESQ	STOI	CSIG	CBAK	COVL	SDR	SIR	SAR	SSNR	SISNR
MCT-noisy	1.7256	0.6967	1.8547	1.1615	1.3937	-0.1762	-0.1762	-	-2.3140	-0.2280
+ standalone-SE	2.6512	0.8277	2.9671	2.5482	2.3410	**11.1091**	**18.4267**	12.3715	**5.1665**	**10.5811**
Joint-Straight [22]	2.5234	0.8695	2.8330	2.3090	2.2989	9.7628	13.0013	12.1108	4.2454	9.8827
Joint-ASO [25]	2.5439	0.8732	2.8603	2.3241	2.3399	10.6317	16.4198	11.9810	4.5783	10.1932
Joint-Grad [24]	2.5411	0.8709	2.8611	2.3244	2.3406	10.1110	13.1041	12.1499	4.4104	9.9012
Proposed	**2.6653**	**0.8311**	**3.1204**	**2.5684**	**2.4509**	10.8998	17.1902	**12.4433**	4.8182	10.3687

TABLE 4. Comparison of speech quality and noise reduction quality scores of SE models trained using different training approaches on et05_simu in the CHiME-4 test Dataset.

Training Approach	Speech Quality					Noise Reduction Quality				
	PESQ	STOI	CSIG	CBAK	COVL	SDR	SIR	SAR	SSNR	SISNR
MCT-noisy	1.9805	0.8166	2.3944	1.7629	1.7236	5.2784	5.2784	-	-1.4066	5.2399
+ standalone-SE	2.5932	0.8991	2.9186	2.4821	2.3711	12.5115	**17.9900**	12.5413	**10.2023**	**11.5718**
Joint-Straight [22]	2.5234	0.8695	2.8330	2.3090	2.2989	11.8877	16.3142	12.1278	9.9932	10.8723
Joint-ASO [25]	2.5439	0.8732	2.8603	2.3241	2.3399	12.3478	16.5440	12.2389	10.0773	11.1873
Joint-Grad [24]	2.5411	0.8709	2.8611	2.3244	2.3406	12.3252	16.6136	12.2272	10.1423	11.1553
Proposed	**2.6098**	**0.8894**	**3.0731**	**2.5133**	**2.4510**	**12.5144**	17.8202	**12.5554**	10.1864	11.5293

TABLE 5. Ablation Study on the Effectiveness of Different Loss Combinations in the Proposed Training Approach on ASR performance using Test-Clean in the noisy LibriSpeech dataset (√ = applied to the proposed training approach).

Training Approach	Loss Function			CER (%)					WER (%)				
				Development		Test		Avg.	Development		Test		Avg.
	$\mathcal{L}_{NSNR}$	$\mathcal{L}_{Enc}$	$\mathcal{L}_{Token}$	dev-clean	dev-other	test-clean	test-other		dev-clean	dev-other	test-clean	test-other	
MCT-noisy				12.87	12.92	13.04	13.10	12.98	22.77	23.18	22.95	23.41	23.08
Proposed	√			16.83	16.83	16.96	16.90	16.88	28.94	29.03	29.38	29.14	29.12
	√	√		12.74	12.68	12.90	12.92	12.81	22.58	22.78	22.74	23.15	22.81
	√	√	√	**11.02**	**11.24**	**11.43**	**11.42**	**11.28**	**19.86**	**20.40**	**20.28**	**20.67**	**20.30**

table, the standalone-SE model significantly improved both the speech quality and noise reduction quality compared with models trained on noisy speech data. Next, the SE model was excerpted from the pipeline and trained using different training approaches. As shown in the third to fifth rows of the table, the SE models trained using the conventional joint optimization approaches exhibited better SE performance than the model trained using noisy speech data but worse than the standalone-SE model. This is because the conventional training approaches focus on improving only ASR performance without considering SE performance.

In contrast, the SE model trained using the proposed training approach achieved the highest speech quality scores for all measures compared with the SE models trained using the conventional joint training approaches, which is even better than the standalone-SE model. This is because the proposed training approach attempts to improve speech recognition, which results in speech quality improvement. Specifically, the SE model by the proposed training approach improved the CSIG, CBAK, and COVL scores by 0.1533, 0.0202, and 0.1099, respectively, over standalone-SE. However, its noise reduction quality scores were slightly lower than those of the standalone-SE model. This result implies that ASR performance may be more closely associated with speech quality than noise reduction quality.

Next, the speech quality and noise reduction quality measurements were repeated using a part of the CHiME-4 dataset, et05_sumu. Note that simulation data were used in this experiment because ground-truth speech signals for the real recording data were not available. Table 4 compares the speech quality and noise reduction quality scores of the SE models trained using different training approaches. The speech quality and noise reduction quality shown in Table 4 have a similar tendency to the results shown in Table 3. In other words, the SE model trained using the proposed training approach achieved the best scores for all metrics compared with those trained using the conventional joint training approaches. Furthermore, compared with the standalone-SE model, the SE model trained using the proposed training approach improved speech quality scores while maintaining comparable noise reduction quality scores. Specifically, the SE model trained using the proposed training approach increased the CSIG, CBAK, and COVL scores by 0.1545, 0.0312, and 0.0799, respectively, compared with the standalone-SE model.

TABLE 6. Ablation study on the effectiveness of various loss combinations in the proposed training approach on speech and noise reduction quality on Test-Clean in the noisy LibriSpeech dataset (√ = applied to the proposed training approach).

Training Approach	Loss Function			Speech Quality					Noise Reduction Quality				
	$\mathcal{L}_{NSNR}$	$\mathcal{L}_{Enc}$	$\mathcal{L}_{Token}$	PESQ	STOI	CSIG	CBAK	COVL	SDR	SIR	SAR	SSNR	SISNR
MCT-noisy				1.7256	0.6967	1.8547	1.1615	1.3937	-0.1762	-0.1762	-	-2.3140	-0.2280
Proposed	√			2.6512	0.8277	2.9671	2.5482	2.3410	11.1091	18.4267	12.3715	**5.1665**	10.5811
	√	√		2.6500	0.8211	2.9595	2.5410	2.3387	**11.1107**	**18.4303**	**12.3798**	5.1662	**10.5821**
	√	√	√	**2.6653**	**0.8311**	**3.1204**	**2.5684**	**2.4509**	10.8998	17.1902	12.4433	4.8182	10.3687

TABLE 7. Ablation study on the effectiveness of different loss combinations in the proposed training approach on ASR performance on Test-Clean in the CHiME-4 dataset (√ = applied to the proposed training approach).

Training Approach	Loss Function			CER (%)					WER (%)				
				Development		Test		Avg.	Development		Test		Avg.
	$\mathcal{L}_{NSNR}$	$\mathcal{L}_{Enc}$	$\mathcal{L}_{Token}$	dt05_simu	dt05_real	et05_simu	et05_real		dt05_simu	dt05_real	et05_simu	et05_real	
MCT-noisy				10.61	10.63	11.27	16.02	12.13	22.60	23.06	23.57	31.00	25.06
Proposed	√			15.51	15.80	19.38	21.42	18.03	31.19	32.73	33.58	36.23	33.43
	√	√		10.38	10.39	10.44	15.80	11.75	21.47	21.98	22.88	29.82	23.04
	√	√	√	**9.32**	**9.53**	**9.81**	**14.53**	**10.80**	**20.75**	**20.98**	**21.20**	**28.41**	**22.84**

TABLE 8. Ablation study on the effectiveness of various loss combinations in the proposed training approach on speech and noise reduction quality using et05_simu in the CHiME-4 dataset (√ = applied to the proposed training approach).

Training Approach	Loss Function			Speech Quality					Noise Reduction Quality				
	$\mathcal{L}_{NSNR}$	$\mathcal{L}_{Enc}$	$\mathcal{L}_{Token}$	PESQ	STOI	CSIG	CBAK	COVL	SDR	SIR	SAR	SSNR	SISNR
MCT-noisy				1.9805	0.8166	2.3944	1.7629	1.7236	5.2784	5.2784	-	-1.4066	5.2399
Proposed	√			2.5932	0.8891	2.9186	2.4821	2.3711	12.5115	**17.9900**	12.5413	**10.2023**	11.5718
	√	√		2.5938	0.8985	2.8940	2.4375	2.3689	**12.5123**	17.8804	12.5383	10.1783	**11.5872**
	√	√	√	**2.6098**	**0.8894**	**3.0731**	**2.5133**	**2.4510**	12.5144	17.8202	**12.5554**	10.1864	11.5293

E. ABLATION STUDY

1) CONTRIBUTION OF DIFFERENT LOSS FUNCTIONS

This ablation study examines the effect of each loss function in the proposed training approach on SE and ASR performances. Table 5 compares the average CERs and WERs of the ASR models trained with different combinations of loss functions and the noisy LibriSpeech dataset. Note that the ASR model in the first row of the table corresponds to the ASR model trained using MCT and the clean and noisy LibriSpeech training datasets. The second to last rows of the table compare the average CERs and WERs for different combinations of loss functions. The second row presents the ASR performance when only the SE loss function is applied. The ASR performance degraded as though the standalone-SE model was combined with MCT, as shown in Tables 1 and 2.

Next, the ASR encoder loss was incorporated with the SE loss in the proposed training approach. As shown in the third row of the table, the incorporation of the ASR encoder loss improved ASR performance compared with MCT-noisy, but the improvement was marginal. Finally, the loss function in (5) was used in the proposed training approach. According to the results, the combination of all loss functions yielded the lowest CER and WER among the different loss combinations. Moreover, the contribution of tokenizer loss to ASR performance was greater than that of the ASR encoder, resulting in significant reductions in CER and WER.

The contribution of each loss function to speech quality and noise reduction quality was evaluated. Table 6 shows the SE performance obtained using the test-clean set in the noisy LibriSpeech dataset. Similar to ASR performance in Table 5, the speech quality scores were best when all losses were combined in the proposed training approach. In addition, the tokenizer loss more contributed to improving all the speech quality scores including CSIG, CBAK, and COVL, compared to the ASR encoder loss. However, the noise reduction quality scores did not increase, as discussed in Section V-D., implying that more attention should be paid to speech quality than noise reduction quality for better speech recognition.

In addition to the noisy LibriSpeech datasets, we repeated the ablation study using the CHiME-4 dataset to investigate the effect of different combinations of loss functions in the proposed training approach. Table 7 compares the average CERs and WERs of the ASR models trained using the CHiME-4 dataset. The relative reduction in the average CERs and WERs according to different combinations of loss functions exhibited a similar tendency on both simulated and real recording data to the CERs and WERs shown in Table 5. Table 8 shows the SE performance obtained on the et05_simu set in the CHiME-4 dataset. Similar to the speech quality results in Table 6, the speech quality scores were best

TABLE 9. Comparison of the processing time per one epoch according to four different training approaches.

	Joint-Straight	Joint-ASO	JOINT-GRAD	Proposed
Time (hours)	2.4	3.2	2.8	**1.5**

TABLE 10. Distributions of conflicting gradients (%) of four different training approaches on the noisy LibriSpeech dataset.

$\cos\theta$	Joint-Straight	Joint-ASO	Joint-Grad	Proposed
[1.0, 0.0)	67.81	71.22	80.61	**85.58**
[0.0, -0.01)	1.84	1.75	1.59	**2.19**
[-0.01, -1.0]	30.35	27.03	17.80	**12.23**

when all losses were used in the proposed training approach. Furthermore, the speech quality and noise reduction quality exhibited similar trend patterns, which were influenced by the different combinations of loss functions. In other words, the inclusion of tokenizer loss significantly contributed to ASR and SE performance improvement on the CHiME-4 dataset, surpassing the contributions of other loss functions.

2) COMPARISON OF CONVERGENCE AND PROCESSING TIME

The investigate of a training loss curve provides an intuition on how much the conflicting problem [24]. Fig. 3 shows the training loss curves for Joint-Straight, Joint-ASO, Joint-Grad, and the proposed training approach. Here, the loss for the three conventional training approaches sums NSNR and ASR losses, whereas the proposed training approach computes the sum of NSNR, ASR encoder, and tokenizer losses, resulting in different loss scales. Thus, the left y-axis of the figure represents loss for the conventional training approaches, and the right y-axis denotes loss for the proposed training approach. Among the conventional training approaches, Joint-Grad showed a lower training loss than Joint-Straight and Joint-ASO. This implies that Joint-Grad exhibits a slower convergence speed than Joint-Straight but faster than Joint-ASO. In contrast, the proposed training approach converges at around 10th epoch, demonstrating a faster convergence speed than the conventional training approaches.

In addition, Table 9 compares the time required per epoch. The conventional training approaches require the time to train whole ASR including encoder and decoder, resulting in longer durations than the proposed training approach. The Joint-Grad approach involves gradient projection and rescaling, thereby requiring a longer processing time than Joint-Straight. Similarly, Joint-ASO requires two updates per epoch, requiring a longer processing time than Joint-Straight. Consequently, the proposed training approach showed the shortest processing time per epoch among all training approaches, even though it includes a tokenizer consisting of a time-distributed layer.

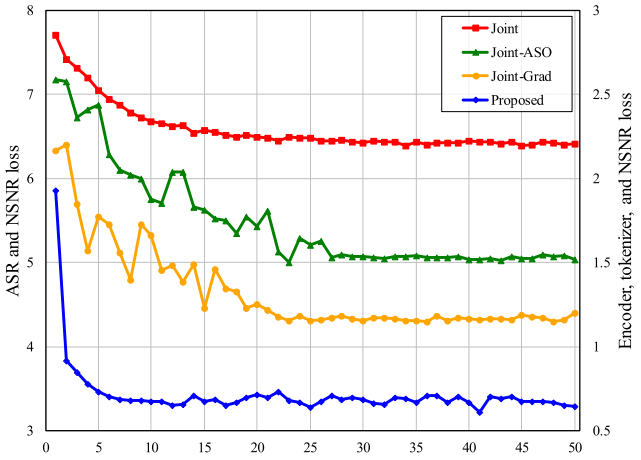

FIGURE 3. Training loss curves on four different training approaches: the left y-axis corresponds to the loss values for Joint, Joint-ASO, Joint-Grad, while the right y-axis to the proposed training approach.

3) ANALYSIS OF ALIGNMENT MISMATCH

To examine the mitigation of the alignment mismatch problem between the SE and ASR losses, conflicting gradients were measured between the SE-related gradient $\boldsymbol{G}_{SE}$ and ASR-related gradient $\boldsymbol{G}_{ASR}$. A conflicting gradient can be measured as $\cos\theta = \boldsymbol{G}_{SE} \cdot \boldsymbol{G}_{ASR} / \|\boldsymbol{G}_{SE}\| \, \|\boldsymbol{G}_{ASR}\|$ [24], [51]. To calculate the gradients, each gradient was computed from the final layer of the DCCRN, specifically a complex two-dimensional transpose convolution layer, using the noisy LibriSpeech training dataset. The $\boldsymbol{G}_{SE}$ and $\boldsymbol{G}_{ASR}$ for each of the three conventional training approaches, Joint-Straight, Joint-ASO, and Joint-Grad, were calculated from the NSNR and ASR loss, respectively. In contrast, the $\boldsymbol{G}_{ASR}$ for the proposed training approach was calculated from both the ASR encoder and tokenizer losses. Note that since the SE-related gradient loss was calculated on a frame-wise basis, the ASR-related gradient loss calculated on a segment-wise basis in the conventional training approaches was converted into a frame-wise basis.

Table 10 presents the distributions of conflicting gradients for the four different training approaches on non-silent frames in the noisy LibriSpeech dataset. In this table, negative conflicting gradients ranging [-0.01, -1.0] indicate several conflicts [52]. As shown in the table, among the three conventional training approaches, Joint-Grad achieved the lowest percentage of severe conflicts. The proposed training approach showed lower percentage of severe conflicts than the three conventional training approaches. This result can be interpreted that the proposed training approach contributes to mitigating the alignment mismatch problem, resulting in higher ASR performance than the conventional training approaches.

4) EXPERIMENT WITH ANOTHER SE MODEL

To verify the effectiveness of the proposed training approach on a pipeline comprising a different SE model, the

TABLE 11. Comparison of average CERs and WERs of ASR models with FullSubNet+ trained using different training approaches on the noisy LibriSpeech datasets.

Training Approach	CER (%)					WER (%)				
	Development		Test		Avg.	Development		Test		Avg.
	dev-clean	dev-other	test-clean	test-other		dev-clean	dev-other	test-clean	test-other	
Joint-Straight [22]	13.11	13.81	13.18	13.96	13.52	23.60	25.14	23.09	24.42	24.06
Joint-ASO [25]	13.04	13.71	13.04	13.88	13.42	23.50	23.98	22.83	24.28	23.90
Joint-Grad [24]	11.72	12.28	11.65	12.41	12.02	21.12	22.39	20.42	21.72	21.41
Proposed	**10.97**	**11.06**	**11.30**	**11.36**	**11.17**	**19.42**	**19.85**	**20.43**	**20.58**	**20.07**

TABLE 12. Comparison of average CERs and WERs of ASR models with FullSubNet+ trained using different training approaches on the CHiME-4 dataset.

Training Approach	CER (%)					WER (%)				
	Development		Test		Avg.	Development		Test		Avg.
	dt05_simu	dt05_real	et05_simu	et05_real		dt05_simu	dt05_real	et05_simu	et05_real	
Joint-Straight [22]	10.52	10.54	10.54	15.81	11.85	21.87	22.19	22.94	29.97	24.24
Joint-ASO [25]	10.41	10.51	19.47	15.65	11.76	21.72	22.01	22.37	29.62	23.93
Joint-Grad [24]	9.64	9.89	10.05	14.96	11.14	21.23	21.39	21.55	29.01	23.30
Proposed	**9.28**	**9.48**	**9.73**	**14.38**	**10.72**	**20.54**	**20.71**	**21.02**	**28.05**	**22.58**

TABLE 13. Comparison of speech quality and noise reduction quality scores of FullSubNet+-based SE Model trained using different training approaches on Test-Clean in the noisy LibriSpeech dataset.

Training Approach	Speech Quality					Noise Reduction Quality				
	PESQ	STOI	CSIG	CBAK	COVL	SDR	SIR	SAR	SSNR	SISNR
Standalone-SE	2.6897	0.8259	3.0247	2.4507	2.3884	**11.2988**	**12.6931**	17.6119	4.3043	12.6931
Joint-Straight [22]	2.5733	0.8004	2.8596	2.3237	2.2769	10.5887	11.1004	17.0801	4.0122	11.7991
Joint-ASO [25]	2.5933	0.8074	2.8664	2.3203	2.2897	10.6412	11.1832	17.1141	4.0183	11.8391
Joint-Grad [24]	2.5752	0.7988	2.8554	2.3389	2.3389	10.1399	10.5020	16.9185	3.9011	11.5093
Proposed	**2.6922**	**0.8281**	**3.1048**	**2.5616**	**2.5616**	11.1297	12.4006	**17.6290**	**4.5342**	**12.7000**

TABLE 14. Comparison of speech quality and noise reduction quality scores of SE models trained using different training approaches using FullSubNet+ on et05_simu in the CHiME-4 test dataset.

Training Approach	Speech Quality					Noise Reduction Quality				
	PESQ	STOI	CSIG	CBAK	COVL	SDR	SIR	SAR	SSNR	SISNR
Standalone-SE	2.6316	0.8975	2.9769	2.3846	2.4181	12.7013	**17.1754**	**12.8641**	**9.3412**	**13.6830**
Joint-Straight [22]	2.5015	0.8714	2.8565	1.5745	2.3094	11.6016	15.6757	12.0647	9.1213	12.2830
Joint-ASO [25]	2.5746	0.8768	2.8968	1.8513	2.3083	12.1012	16.0755	12.4213	9.2287	12.8292
Joint-Grad [24]	2.5723	0.8753	2.8761	1.4940	2.2848	11.9014	15.7753	12.4448	9.1685	12.7701
Proposed	**2.6669**	**0.8988**	**3.0651**	**2.4144**	**2.4608**	**12.8019**	16.5752	12.7649	9.2511	13.5730

FullSubNet+ SE model [53], which exhibited state-of-the-art SE performance, was replaced with DCCRN SE model. In this work, the authors' source code[1] using the PyTorch framework was transformed into the TensorFlow framework.

Tables 11 and 12 compare the average CERs and WERs of the ASR models trained using four different training approaches on the noisy LibriSpeech dataset and CHiME-4 dataset, respectively, when FullSubNet+-based SE model was employed. Compared with the ASR performance of the DCCRN-based SE model shown in Tables 1 and 2, ASR performances according to the training approach had similar tendency for both datasets: the proposed training approach exhibited the lowest CER and WER among all the training approaches. As shown in Table 11, the ASR model with FullSubNet+ trained using the proposed training approach on the noisy LibriSpeech dataset relatively reduced the average CER and WER by 7.01% and 6.27%, respectively, compared with that trained using Joint-Grad, which was the best-performing conventional training approach. Similarly, the ASR model trained using the proposed training approach on CHiME-4 dataset showed relative reduction of CER and WER by 3.75% and 3.07%, respectively, compared with that using Joint-Grad.

Next, Tables 13 and 14 compare the speech quality and noise reduction quality scores of FullSubNet+-based SE model according to different training approaches, applied to the test-clean set in the noisy LibriSpeech dataset as et05_sumu in the CHiME-4 dataset, respectively. Compared with the results in Tables 3 and 4, the performance trend of FullSubNet+-based SE model according to different training

[1] https://github.com/RookieJunChen/FullSubNet-plus

approaches was similar. In other words, the conventional training approaches yielded lower SE performance than standalone-SE. In addition, the proposed training approach also achieved increased speech quality scores including CSIG, CBAK, and COVL, compared with standalone-SE.

Throughout these experiments on using the different SE model, it could be concluded that the proposed training approach did not limit to any specific SE model, and any kind of SE model could be a component of a pipeline for noise-robust speech recognition.

VI. CONCLUSION

This paper proposed a KD-based joint training approach to address issues associated with a noise-robust E2E ASR model. To this end, a pipeline comprising SE and ASR models was constructed, where the ASR and SE models were interpreted as teacher and student models, respectively. In this pipeline, an acoustic tokenizer acted as an information converter from the frame-wise information of the SE output into the segment-wise information of the ASR output to accommodate different goals of SE and ASR models, i.e., enhancing speech quality and extracting linguistic information, respectively. To train the acoustic tokenizer, a tokenizer loss function was proposed. Furthermore, this loss was combined with SE loss and ASR encoder loss to perform KD-based training of SE models.

The effectiveness of the proposed training approach was evaluated through exhaustive experiments on noisy LibriSpeech datasets. First, the CER and WER of the ASR models trained using the proposed training approach, conventional training approaches, including MCT and MCT-noisy+standalone-SE, and three different joint training approaches were compared. Consequently, the ASR model trained using the proposed training approach yielded the lowest CER and WER among all models on two different datasets. Specifically, the proposed approach relatively improved the CER and WER by 13.15% and 12.03%, respectively, compared with the MCT-noisy training approach. Compared with Joint-Grad, the best-performing conventional joint training approach investigated in this paper, the proposed training approach relatively reduced the CER and WER by 3.24% and 2.00%, respectively.

The proposed training approach was applied to the CHiME-4 dataset to investigate its effectiveness on a real-world recorded speech dataset. According to the results, the proposed training approach achieved lower CER and WER than the conventional joint training approaches.

Next, the speech quality and noise reduction quality of SE models trained using different training approaches were compared. According to the results, the proposed training approach yielded the highest speech quality scores among the comparative approaches, whereas it did not improve the noise reduction quality compared with the standalone-SE training approach. These results imply that ASR performance is more associated with speech quality scores than noise reduction scores.

An ablation study was conducted to examine the contribution of different combinations of loss functions to SE and ASR performance. The combination of all loss functions, such as SE, ASR encoder, and tokenizer losses, yielded the lowest CER and WER. Furthermore, the contribution of tokenizer loss to both SE and ASR performance was greater than that of ASR encoder loss. Then, to investigate the conflicting problem on different goals, the proposed training approach compared its training loss curve with those of conventional training approaches. Additionally, the frame-wise conflicting gradients according to each training approach were examined to analysis the alignment mismatch issue. Finally, additional SE model was applied to the pipeline of the proposed training approach to verify its effectiveness across different SE architecture.

Nevertheless, this study has two limitations. First, we have to determine whether the acoustic tokenizer trained with a cross-entropy loss function using noisy labels from K-means clustering can degrade performance because of overfitting on difficult samples [54]. Furthermore, this study was conducted using only one SE architecture and one ASR architecture, which limits the generality of the results. To address these issues, overfitting on difficult samples will be explored in the future using metric learning, which is effective in feature representation and uses pairwise distances [55]. Moreover, experiments will be conducted using different ASR architectures, e.g., attention-based encoder–decoder-based and CTC-based architectures.

REFERENCES

[1] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 10, pp. 1702–1726, Oct. 2018.

[2] S. A. Nossier, J. Wall, M. Moniri, C. Glackin, and N. Cannings, "An experimental analysis of deep learning architectures for supervised speech enhancement," *Electronics*, vol. 10, no. 1, p. 17, Dec. 2020.

[3] J. M. Perero-Codosero, F. M. Espinoza-Cuadros, and L. A. Hernández-Gómez, "A comparison of hybrid and end-to-end ASR systems for the IberSpeech-RTVE 2020 speech-to-text transcription challenge," *Appl. Sci.*, vol. 12, no. 2, p. 903, Jan. 2022.

[4] G. Caldarini, S. Jaf, and K. McGarry, "A literature survey of recent advances in chatbots," *Information*, vol. 13, no. 1, p. 41, Jan. 2022.

[5] S. E. Eskimez, T. Yoshioka, H. Wang, X. Wang, Z. Chen, and X. Huang, "Personalized speech enhancement: New models and comprehensive evaluation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shenzhen, China, May 2022, pp. 356–360.

[6] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Honolulu, HI, USA, Jul. 2023, pp. 28492–28518.

[7] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[8] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2020, pp. 12449–12460.

[9] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 7398–7402.

[10] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 4960–4964.

[11] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on transformer vs RNN in speech applications," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 449–456.

[12] A. Gulati, J. Qin, C. C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 5036–5040.

[13] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "ContextNet: Improving convolutional neural networks for automatic speech recognition with global context," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 3610–3614.

[14] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, and J. Chen, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, Jun. 2016, pp. 173–182.

[15] A. Graves, "Sequence transduction with recurrent neural networks," 2012, *arXiv:1211.3711*.

[16] J. Droppo and A. Acero, "Joint discriminative front end and back end training for improved speech recognition accuracy," in *Proc. IEEE Int. Conf. Acoust. Speed Signal Process.*, Toulouse, France, May 2006, pp. 281–284.

[17] K. Kinoshita, T. Ochiai, M. Delcroix, and T. Nakatani, "Improving noise robust automatic speech recognition with single-channel time-domain enhancement network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 7009–7013.

[18] Z.-Q. Wang, P. Wang, and D. Wang, "Complex spectral mapping for sngle- and multi-channel speech enhancement and robust ASR," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 1778–1787, 2020.

[19] 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," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 2472–2476.

[20] K. Shimada, Y. Bando, M. Mimura, K. Itoyama, K. Yoshii, and T. Kawahara, "Unsupervised speech enhancement based on multichannel NMF-informed beamforming for noise-robust automatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 5, pp. 960–971, May 2019.

[21] B. Schuller, F. Weninger, M. Wöllmer, Y. Sun, and G. Rigoll, "Non-negative matrix factorization as noise-robust feature extractor for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Dallas, TX, USA, Mar. 2010, pp. 4562–4565.

[22] L. Li, Y. Kang, Y. Shi, L. Kürzinger, T. Watzel, and G. Rigoll, "Adversarial joint training with self-attention mechanism for robust end-to-end speech recognition," *EURASIP J. Audio, Speech, Music Process.*, vol. 2021, no. 1, pp. 1–16, Jul. 2021.

[23] D. Ma, N. Hou, V. T. Pham, H. Xu, and E. S. Chng, "Multitask-based joint learning approach to robust ASR for radio communication speech," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Dec. 2021, pp. 497–502.

[24] Y. Hu, C. Chen, R. Li, Q. Zhu, and E. S. Chng, "Gradient remedy for multi-task learning in end-to-end noise-robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Rhodes Island, Greece, Jun. 2023, pp. 1–5.

[25] G. W. Lee and H. K. Kim, "Two-step joint optimization with auxiliary loss function for noise-robust speech recognition," *Sensors*, vol. 22, no. 14, p. 5381, Jul. 2022.

[26] A. Pandey, C. Liu, Y. Wang, and Y. Saraf, "Dual application of speech enhancement for automatic speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Shenzhen, China, Jan. 2021, pp. 223–228.

[27] C. C. Lee, Y. Tsao, H. M. Wang, and C. S. Chen, "D4AM: A general denoising framework for downstream acoustic models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Kigali, Rwanda, May 2023, pp. 1–17.

[28] M. Yang, J. Konan, D. Bick, Y. Zeng, S. Han, A. Kumar, S. Watanabe, and B. Raj, "Paaploss: A phonetic-aligned acoustic parameter loss for speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Phodes Island, Greece, Jun. 2023, pp. 1–5.

[29] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[30] A. Romero, N. Ballas, S. Ebrahimi Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*.

[31] W. H. Li and H. Bilen, "Knowledge distillation for multi-task learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 163–176.

[32] G. M. Jacob, V. Agarwal, and B. Stenger, "Online knowledge distillation for multi-task learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Waikoloa, HI, USA, Jan. 2023, pp. 2358–2367.

[33] K. Clark, M.-T. Luong, U. Khandelwal, C. D. Manning, and Q. V. Le, "BAM! Born-again multi-task networks for natural language understanding," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, 2019, pp. 5931–5937.

[34] M. Takamoto, Y. Morishita, and H. Imaoka, "An efficient method of training small models for regression problems with knowledge distillation," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Shenzhen, China, Aug. 2020, pp. 67–72.

[35] S. Khurana, A. Laurent, and J. Glass, "SAMU-XLSR: Semantically-aligned multimodal utterance-level cross-lingual speech representation," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1493–1504, Oct. 2022.

[36] D. Liang, Z. Huang, and Z. C. Lipton, "Learning noise-invariant representations for robust speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Athens, Greece, Dec. 2018, pp. 56–63.

[37] H. Inaguma and T. Kawahara, "Alignment knowledge distillation for online streaming attention-based speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 31, pp. 1371–1385, Dec. 2023.

[38] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, Sardinia, Italy, Mar. 2010, pp. 249–256.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderpla, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, Oct. 2011.

[40] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python," in *Proc. 14th Python Sci. Conf.*, Austin, TX, USA, 2015, pp. 18–25.

[41] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

[42] C. K. A. Reddy, V. Gopal, R. Cutler, E. Beyrami, R. Cheng, H. Dubey, S. Matusevych, R. Aichner, A. Aazami, S. Braun, and P. Rana, "The INTERSPEECH 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 2492–2496.

[43] S.-J. Chen, A. S. Subramanian, H. Xu, and S. Watanabe, "Building state-of-the-art distant speech recognition using the CHiME-4 challenge with a setup of speech enhancement baseline," 2018, *arXiv:1803.10109*.

[44] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, Brussels, Belgium, 2018, pp. 66–71.

[45] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, "TensorFlow: A system for large-scale machine learning," in *Proc. USENIX Symp. Operating Syst. Design Implement.*, Savannah, GA, USA, Nov. 2016, pp. 265–283.

[46] (2005). *ITU-T Recommendation P.862. Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs*. [Online]. Available: https://www.itu.int/rec/T-REC-P.862

[47] 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. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Dallas, TX, USA, Mar. 2010, pp. 4214–4217.

[48] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 1, pp. 229–238, Jan. 2008.

[49] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

[50] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 3291–3295.
[51] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 5824–5836.
[52] G. Shi, Q. Li, W. Zhang, J. Chen, and X. M. Wu, "Recon: Reducing conflicting gradients from the root for multi-task learning," in *Proc. Int. Conf. Learn. Represent.*, Apr. 2022, pp. 1–20.
[53] J. Chen, Z. Wang, D. Tuo, Z. Wu, S. Kang, and H. Meng, "FullSubNet+: Channel attention fullsubnet with complex spectrograms for speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shenzhen, China, May 2022, pp. 7857–7861.
[54] M. Boudiaf, J. Rony, I. M. Ziko, E. Granger, M. Pedersoli, P. Piantanida, and I. B. Ayed, "A unifying mutual information view of metric learning: Cross-entropy vs. pairwise losses," in *Proc. Eur. Conf. Comp. Vis. (ECCV), Virtual Conf.*, Aug. 2020, pp. 548–564.
[55] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2020, pp. 18661–18673.

GEON WOO LEE (Graduate Student Member, IEEE) received the B.S. degree in electronics and computer engineering from Chonnam National University, South Korea, in 2017, and the M.S. degree in electrical engineering and computer science from Gwangju Institute of Science and Technology (GIST), South Korea, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include deep learning approaches on speech/audio enhancement, source separation, and noise-robust automatic speech recognition.

HONG KOOK KIM (Senior Member, IEEE) received the B.S. degree in control and instrumentation engineering from Seoul National University, South Korea, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 1990 and 1994, respectively. From 1990 to 1998, he was a Senior Researcher with the Samsung Advanced Institute of Technology (SAIT), South Korea. From 1998 to 2003, he was a Senior Technical Staff Member with the Voice-Enabled Services Research Laboratory, AT&T Laboratories-Research, Florham Park, NJ, USA. Since August 2003, he has been a Professor with the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), South Korea, as a Professor. He is also jointly affiliated with the AI Graduate School, GIST. From 2014 to 2015, he was a Visiting Professor with the City University of New York, New York City, USA. His research interests include statistical and deep learning approaches on speech recognition, sound event detection, unsupervised anomaly detection, speech/audio enhancement, and sound source separation. He had served as an Editorial Committee Member and an Area Editor for *Digital Signal Processing* and is serving as a member for APSIPA Speech, Language, and Audio Technical Committee.

DUK-JO KONG (Member, IEEE) received the B.S. degree in electronic engineering from Chungnam National University, South Korea, in 2010, and the M.S. and Ph.D. degrees in electrical engineering and computer science from Gwangju Institute of Science and Technology (GIST), South Korea, in 2012 and 2016, respectively. From 2016 to 2021, he was a Senior Research Scientist with GIST, where he has been a Principal Research Scientist, since 2021, and an Adjunct Professor with the AI Graduate School.