Waveform-domain Speech Enhancement Using Spectrogram Encoding for Robust Speech Recognition

Hao Shi, *Student Member, IEEE,* Masato Mimura, *Member, IEEE,* Tatsuya Kawahara, *Fellow, IEEE*

***Abstract*—While waveform-domain speech enhancement (SE) has been extensively investigated in recent years and achieves state-of-the-art performance in many datasets, spectrogram-based SE tends to show robust and stable enhancement behavior. In this paper, we propose a waveform-spectrogram hybrid method (WaveSpecEnc) to improve the robustness of waveform-domain SE. WaveSpecEnc refines the corresponding temporal feature map by spectrogram encoding in each encoder layer. Incorporating spectral information provides robust human hearing experience performance. However, it has a minor automatic speech recognition (ASR) improvement. Thus, we improve it for robust ASR by further utilizing spectrogram encoding information (WaveSpecEnc+) to both the SE front-end and ASR back-end. Experimental results using the CHiME-4 dataset show that ASR performance in real evaluation sets is consistently improved with the proposed method, which outperformed others, including DEMUCS and Conv-Tasnet. Refining in the shallow encoder layers is very effective, and the effect is confirmed even with a strong ASR baseline using WavLM.**

***Index Terms*—speech enhancement, robust ASR, time-frequency hybrid model, spectral information refining**

I. Introduction

Speech enhancement (SE) [1], [2] aims to recover speech components from noisy speech signals. Noise has a very adversary effect on human hearing and signal processing [3]. Thus, SE has been one of the important research topics of speech signal processing. Big data-driven, deep learning-based supervised SE methods [4], [5] show more powerful performance than traditional SE methods [6], [7], [8]. While traditional SE methods make some mathematical assumptions [6], [7], [9], which limit the enhancement performance, the deep learning-based SE utilizes the nonlinear mapping capabilities of deep neural networks to mitigate the above issue [4], [5].

Deep learning-based SE [4], [5] can be classified into frequency-domain [4], [5] and waveform-domain models [10], [11]. Frequency-domain SE extracts frequency features from the waveform-domain speech signals using the short-time Fourier transform (STFT). The magnitude of spectrogram [5] is a common frequency-domain feature, but it ignores the phase information and limits the model performance. To address the problem, real and imaginary parts of STFT, also called the complex-domain spectrogram [12], which contains both magnitude and phase information, have been adopted by many SE systems in recent years [13], [14].

Different from frequency-domain SE, waveform-domain SE [10], [11], [15] adopts speech waveform as input and output features. The magnitude and phase information is included in the waveform. With intensive studies, waveform-domain SE methods achieve state-of-the-art performance in many datasets [10], [11], [16]. However, it is often pointed out that the frequency-domain SE systems have more stable enhancement performance than waveform-domain SE systems [17] because of the stability of the magnitude of the spectrogram compared with the phase information [18].

In order to improve the robustness of the waveform-domain SE method, we have proposed a waveform-spectrogram hybrid system (WaveSpecEnc) [19]. The proposed method complements waveform-domain DEMUCS [10] with the magnitude of spectrogram information. The waveform-spectrogram information fusion is done in the encoder. In each encoder layer, temporal and spectral information is first extracted by convolution processing at the utterance level. Then, the temporal feature maps are segmented and aligned with the spectral feature maps. The aligned spectral information is used to refine the segmental temporal information. The Hybrid DEMUCS [20] also integrates information on the temporal and spectrogram domains. The significant difference between the Hybrid DEMUCS [20] and our proposed WaveSpecEnc is that the Hybrid DEMUCS [20] employs shared encoder and decoder layers to process the information from different domains, while the WaveSpecEnc integrates the spectrogram information into the waveform encoding layer by layer.

In addition to human hearing experience, improving automatic speech recognition (ASR) [3], [21], [22] in noisy conditions is crucial for the SE front-end. Previous works [23], [24] have found that information loss caused by the SE front-end affects the performance of ASR. To alleviate the problem, in this study, we improve the WaveSpecEnc by augmenting the encoding information of the ASR back-end with spectral information extracted in the SE module (WaveSpecEnc+). The enhanced spectral feature maps in the last layer in the WaveSpecEnc encoder are used to supplement the filter-bank (FBank) encoding in the ASR back-end. Different from previous works [23], [24], the enhanced waveform-domain and spectrogram-domain encodings are fused in this work instead of fusing the original noisy and enhanced spectrograms. Furthermore, previous work [25] has found that some speech information is highlighted after joint training,

Hao Shi, Masato Mimura, and Taysuya Kawahara are with the Graduate School of Informatics, Kyoto University, Kyoto, Japan (e-mail: shi@sap.ist.i.kyoto-u.ac.jp, mimura@sap.ist.i.kyoto-u.ac.jp, kawahara@i.kyoto-u.ac.jp).

which means that spectral information useful for ASR is emphasized. In this manner, we aim to extract discriminative information from the enhanced spectral feature maps, which helps improve filterbank encoding performance in the ASR encoder. Compared to WaveSpecEnc [19], which only integrates spectrogram encoding information into the front-end's temporal information, WaveSpecEnc+ integrates spectrogram encoding information into both the front-end and the ASR back-end to enhance the performance of ASR.

In the following sections, we will introduce related work in Section II. We will explain the proposed method in Section III. The experimental settings and results are presented in Section V. The conclusion will be given in Section VII.

II. Related Work

A. Supervised Waveform-domain SE

A noisy speech signal y can be expressed as:

$$y = x * r + n \tag{1}$$

where x represents a clean speech signal in the waveform domain, $*$ represents the convolution operator, r represents a room impulse response, and n represents an additive noise. SE aims to recover x from y. In this work, we focus on the effect of the SE for additive noise n, and the dereverberation is not the focus of this work. According to the input and output features, SE can be classified into frequency-domain and waveform-domain methods.

The spectrogram is a common feature of frequency-domain SE. It is extracted via STFT:

$$X = |STFT(x)| \tag{2}$$

$STFT(\bullet)$ denotes the Short-time Fourier Transform. $|\bullet|$ denotes the modulus. X represents the magnitude spectrogram of the clean speech. For the loss function, mean absolute error (MAE) is commonly used:

$$\mathcal{L}_{mae_f} = \frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}|\hat{X}_{t,f} - X_{t,f}|, \tag{3}$$

where $\hat{X}$ represents the enhanced magnitude of spectrogram. T, F are the number of time and frequency points in the magnitude.

For waveform-domain SE, waveform is the input and output of the neural network. For the loss function, it adopts waveform-domain MAE,

$$\mathcal{L}_{mae_t} = \frac{1}{T}\sum_{t=1}^{T}|\hat{x}_t - x_t|, \tag{4}$$

where $\hat{x}$ is the enhanced speech waveform. T is the number of time points in the waveform.

DEMUCS [10] is a powerful waveform-domain SE system based on the U-Net structure. It contains an encoder, a decoder, and two long short-term memory (LSTM) [26] layers between them. Each **"Time Block" (DEMUCS encoder layer)** contains two "Conv_1d" layers:

$$GLU(Conv1d((ReLU(Conv1d(\cdot))))). \tag{5}$$

The first "Conv_1d" layer is followed by the "ReLU" activation function, while the second "Conv_1d" layer is followed by the "GLU" activation function. Each **"DEMUCS Decoder Layer"** contains one "Conv_1d" layer and one "DeConv_1d" layer:

$$ReLU(ConvTranspose1d(GLU(Conv1d(\cdot)))). \tag{6}$$

The "Conv_1d" layer is followed by the "GLU" activation function, while the "DeConv_1d" layer is followed by the "ReLU" activation function. The kernel size of the encoder and decoder layers is 8.

DEMUCS [10] also adopts upsampling [27] and downsampling [27] processing to the original input and enhanced output waveform. For the loss function, it adopts waveform-domain MAE with multi-resolution frequency loss.

$$\begin{aligned}
&\mathcal{L}_{sc}(r) = \frac{||STFT_r(\hat{x})| - |STFT_r(x)||}{|STFT_r(x)|},\\
&\mathcal{L}_{mag}(r) = \frac{1}{T}|\log|STFT_r(\hat{x})| - \log|STFT_r(x)||,\\
&\mathcal{L}_{stft}(r) = \mathcal{L}_{sc}(r) + \mathcal{L}_{mag}(r),\\
&\mathcal{L}_{demucs} = \alpha\mathcal{L}_{mae_t} + (1-\alpha)\sum_{r=1}^{R}(\mathcal{L}_{stft}(r))
\end{aligned} \tag{7}$$

R represents the multi-resolution number, and r represents the specific resolution among {32ms, 64ms, 128ms} used in STFT.

Data mismatch is a big issue of supervised SE [28]. Conventionally, supervised SE systems are trained with simulated training data. Using real noisy data for training is difficult because a clean speech waveform is needed for ground truth. However, the data distribution between real and simulated noisy speech often differs significantly. Moreover, the noise conditions are also crucial. SE systems tend to degrade in the presence of unseen noise. It is necessary to evaluate the robustness of SE systems under real data and unseen noise conditions.

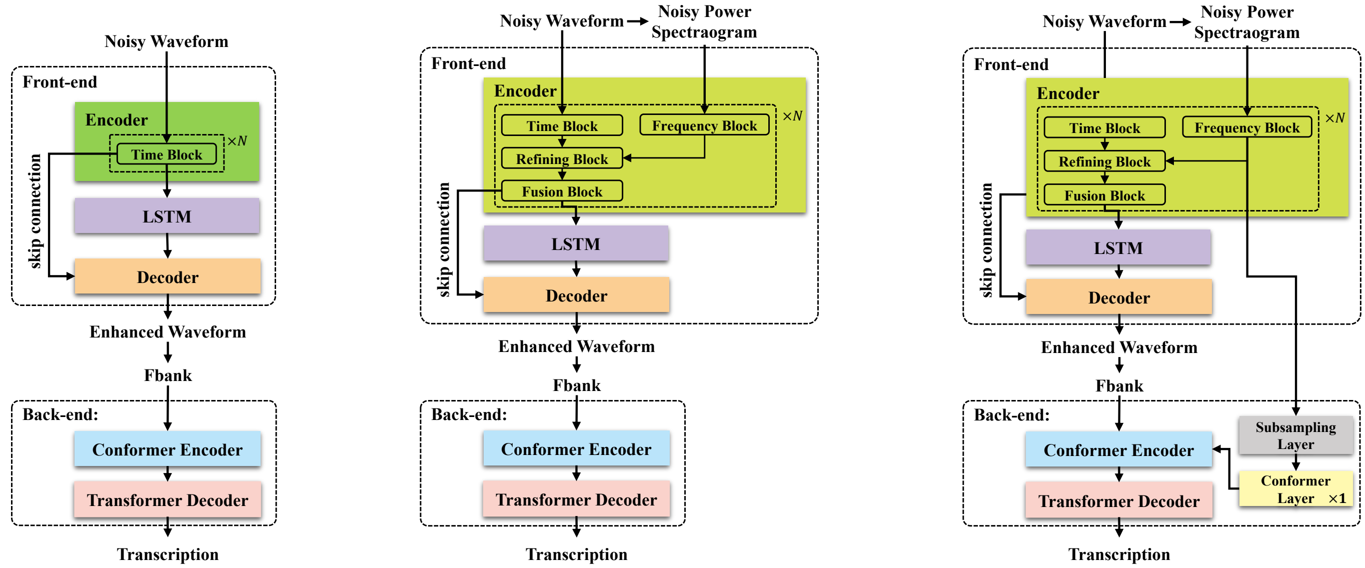
B. Conformer-based ASR System

Conformer-based ASR systems [29] have achieved state-of-the-art performance on many benchmark datasets. It typically consists of two main components: an encoder and a decoder. The encoder is responsible for processing the input speech signal and producing a sequence of feature vectors that capture relevant information about the speech signal. Several Conformer layers hierarchically process the input sequence, with each layer processing the input at a different level of abstraction. The decoder is responsible for converting the sequence of feature vectors produced by the encoder into a sequence of characters or phones that represent the transcription of the input speech signal.

During training, the network is trained to minimize attention-based Transformer decoder loss function [30] augmented with CTC (Connectionist Temporal Classification) [31]:

$$L_{ASR} = \beta * L_{att} + (1-\beta) * L_{CTC} \tag{8}$$

where L_{att} and L_{CTC} are loss functions of the Transformer decoder and CTC, respectively. β is a hyperparameter to control the two losses.



(a) DEMUCS-based ASR system (b) WaveSpecEnc-based ASR system (c) WaveSpecEnc+-based ASR system

Fig. 1. Flowchart of different robust ASR systems.

C. Robust ASR

Robust ASR systems [3], [32], [33] are designed to work well in challenging acoustic environments. We focus on background noise in this paper. Currently, there are two major approaches to improving the robustness of ASR. One straightforward approach called multi-condition ASR is to train the ASR system with a variety of noisy data [3]. Although this approach can boost the robustness of ASR, its performance is limited in very noisy or low SNR conditions. The other approach is to incorporate an SE front-end [23], [34]. Usually, the SE front-end and ASR back-end are pretrained separately before joint training [35]. The loss function of joint training is defined as a sum of the ASR loss (L_{ASR}) and the SE loss ($\mathcal{L}_{demucs}$). However, the SE loss ($\mathcal{L}_{demucs}$) can be computed only when the original clean speech x is available, and thus real noisy data cannot be used. Therefore, we simply use the loss function of ASR (L_{ASR}) for finetuning SE. The DEMUCS-based robust ASR is shown in Fig. 1(a).

III. Proposed Method

A. WaveSpecEnc SE Front-end

Although the performance of waveform-domain SE models has been improved, the instability of the phase information makes the waveform-domain representation less stable than the frequency representations. We propose a waveform-spectrogram hybrid system (WaveSpecEnc) to address this problem. Specifically, we incorporate auxiliary frequency-domain information into waveform-domain features to improve the robustness. The magnitude of the spectrogram is adopted as frequency information.

The waveform-spectrogram information fusion is done in the encoder. Fig. 2(a) shows the encoder layer of the proposed WaveSpecEnc. The waveform-domain feature maps are extracted by the same structure of **"Time Block"** in Eq. (5). The waveform-domain input from the previous WaveSpecEnc encoder layer or original waveform is denoted as $\boldsymbol{y_t}$. To get stable waveform-domain representations, the spectral features are used to refine the extracted feature maps by the "Time Block". Here $\boldsymbol{y_f}$ represents the spectral information from the previous **"Frequency Block"** or the magnitude of the spectrogram. Each "Frequency Block" stacks "Conv_2d" layers:

$$BatchNorm2d(ELU(Conv2d(\cdot))). \tag{9}$$

With different kernel sizes, strides, and convolutional channels, the three "Conv_2d" layers have different purposes. The first "Conv_2d" layer enhances the spectral feature maps and keeps the feature frame the same as the original spectrogram. We denote the output of the first "Conv_2d" layer as $\widetilde{\boldsymbol{y_f}}$, which serves as the input of the next "Frequency Block" in the next encoder layer and the second "Conv_2d" layer simultaneously.

However, it has different convolutional channels and frames from those of the temporal feature maps. Therefore, another two "Conv_2d" layers are introduced to extract deep encoded features with the same convolutional channels and frames as those of the temporal feature maps:

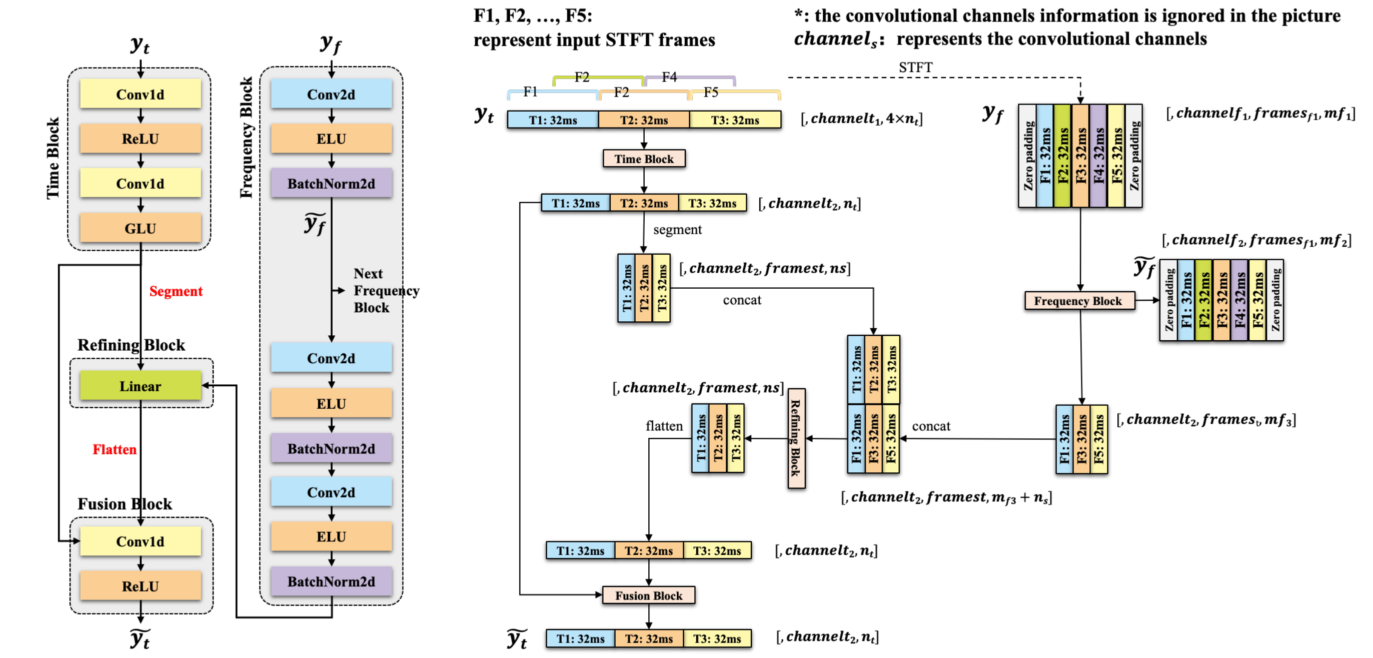
$$BatchNorm2d(ELU(Conv2d(BatchNorm2d(ELU(Conv2d(\cdot)))))). \tag{10}$$

For the number of layers to extract deep encoded features, we have tried to use 1∼3 "Conv_2d" layers: the performance of one "Conv_2d" layer was degraded. The three "Conv_2d" layers perform the same as the two "Conv_2d" layers. Thus, we chose to use two "Conv_2d" layers.

After temporal and spectral information extraction, **"Refining Block"** is adopted to refine the temporal feature. The temporal feature is first segmented into 32ms frames. The spectral feature is extracted with the same frames. The "Refining Block" consists of one fully connected layer:

$$ReLU(linear(\cdot)). \tag{11}$$

Its input feature is a concatenation of 32ms segmental waveform-domain and frequency-domain features. The block converts the waveform-spectrogram hybrid feature maps into the refined temporal feature maps with the same dimensions as the waveform-domain features.



(a) Encoder layer structure of the proposed method **(b) Feature dimensions processed by different blocks in the proposed encoder layer**

Fig. 2. Encoder layer structure of the proposed waveform-spectrogram hybrid system (WaveSpecEnc). $\boldsymbol{y_t}$ represents the waveform-domain input from the previous WaveSpecEnc encoder layer or original waveform; $\boldsymbol{y_f}$ represents the spectral inputs from the previous frequency block or the magnitude of the spectrogram; $\widetilde{\boldsymbol{y_f}}$ represent the spectral output (to the frequency block in the next encoder layer); $\widetilde{\boldsymbol{y_t}}$ represents the final output (to the next encoder or LSTM layer).

Finally, **"Fusion Block"** adopts one "Conv_1d" layer to fuse the original and refined feature maps. The output of each proposed encoder layer is represented as $\widetilde{y_t}$. Fig. 2(b) shows a detailed illustration of these processes with the feature dimensions.

Other parts of the WaveSpecEnc are the same with DEMUCS [10]: two LSTM layers and a decoder with Eq. (6). Both the encoder and decoder contain five layers. The upsampling and downsampling processing are also included. The loss function is the same as Eq. (7).

B. WaveSpecEnc for Robust ASR

It is also important to improve the performance of ASR with the SE front-end. We first try directly using WaveSpecEnc as the SE front-end for robust ASR, as shown in Fig. 1(b). Since the output of WaveSpecEnc is the waveform-domain speech waveform, the log Mel-filterbank (LMFB) is extracted from the enhanced waveform to input to the ASR back-end.

We first pretrain the ASR back-end with a large amount of data. Then, the front-end parameters are finetuned with the ASR (L_{ASR}) loss function. This training scheme is the same as the DEMUCS-based system.

C. WaveSpecEnc+ for Robust ASR

SE front-end often suppresses not only noise but also speech [23], [24]. This is good for human hearing but not for ASR. Some previous works [23], [24] fuse the original noisy spectral feature with the enhanced spectral feature to alleviate this drawback. However, using unprocessed noisy features will make it difficult for network learning. In this study, we exploit or re-use the "Frequency Block" of WaveSpecEnc to augment the features for ASR. Fig. 1(c) shows the flowchart of the WaveSpecEnc+-based robust ASR system.

The subsampling layer is adopted to subsample the output of the final "Frequency Block" in the front-end encoder, which ensures that the spectral information has the same frames as the feature in the ASR back-end. It has the same neural network structure as the subsampling layer in the ASR back-end: two Conv2d layers use a four-time subsampling rate. An additional Conformer layer encodes the spectral information with the attention mechanism. Finally, the LMFB encoding and spectral encoding are fused with a fully connected layer:

$$e_1^f = \mathbb{W}(e_1, s) = ReLU(linear(e_1, s)). \tag{12}$$

s is the extracted spectral encoding information, and e_1 is the output of the first encoder layer of the ASR back-end. e_1^f is the fused feature, which is input to the second encoder layer of the ASR back-end.

IV. Experimental Evaluations of Pretrained Speech Enhancement Front-End

A. Experimental Settings

The experiments were conducted using the CHiME–4 dataset[1], which includes four noise conditions: bus (BUS), cafe (CAF), pedestrian area (PED), and street junction (STR). All data were digitized with 16 kHz sample rate. For SE front-end pretraining, the Channel 1 – Channel 6 simulated data from the training set were used; no development set was used during training. We have adopted the single-channel setting:

[1] https://spandh.dcs.shef.ac.uk/chime_challenge/CHiME4/index.html

TABLE I
(UNPROCESSED NOISY DATA) EVALUATION METRICS ON REAL DEVELOPMENT AND EVALUATION SETS.

Condition	Real Development Set				Real Evaluation Set			
	SIG	OVRL	BAK	dMOS	SIG	OVRL	BAK	dMOS
BUS	1.4	1.2	1.3	2.7	1.6	1.3	1.3	2.4
STR	2.4	1.7	1.6	2.7	2.4	1.7	1.7	2.6
PED	3.0	2.1	2.1	2.9	2.2	1.5	1.5	2.4
CAF	2.5	1.6	1.6	2.7	2.1	1.4	1.4	2.5

the channel 5 data in the development and evaluation sets were used as the test sets. We tested in the following two scenarios.

Seen model: All noise conditions (BUS, CAF, PED, STR) were used in training as seen model.

Unseen model: We held out one noise condition to simulate the case of unseen scenario, that is we trained the model using three different noise scenarios and evaluated it in the remaining unseen noise scenario.

B. Baselines

For the baseline "Bi-LSTM" SE model, the input and output features were the magnitude of the spectrogram. The "Bi-LSTM" contained two Bi-directional LSTM (Bi-LSTM) layers and a fully connected layer. Each Bi-LSTM layer had 896 hidden nodes. For baseline "DEMUCS" and the proposed "WaveSpecEnc", the channels in different depths were {1, 48, 96, 192, 384, 768}. Each LSTM layer contained 768 nodes. In each "Time Block", the kernel size and stride for the two "Conv_1d" layers were {8, 1}, and {4, 1}, respectively. Each decoder layer's kernel size and stride of the "Conv_1d" and "DeConv_1d" layers were {1, 8}, and {1, 4}, respectively. For the first "Conv_2d" layer in each "Alignment Block", the kernel size and stride were 3 and 2, respectively. For the second "Conv_2d" layer in each "Alignment Block", the kernel size and stride were 3 and 1, respectively. The input and output dimensions of "Refining Block" were {640, 191, 63, 23, 9} and {512, 128, 32, 8, 2}. The input and output channels of "Fusion Block" were {96, 192, 384, 768, 1536} and {48, 96, 192, 384, 768}. The kernel size of Conv1d in "Fusion Block" is 1. For extracting the spectrogram, the STFT points were 32ms; the Hanning window was used; the STFT hop length was 16ms. The hyperparameter α in Eq. (7) was set to 0.5. We also compared with "Hybrid DEMUCS (H-DEMUCS)" [20] by following the official source code[2].

All neural networks were implemented with PyTorch. All SE front-ends were trained with 200 epochs.

C. Evaluation Metrics

We used multiple linear regression analysis to form the following composite measures: signal distortion (SIG) [36], background intrusiveness (BAK) [36], overall quality (OVL) [36], and the subjective Mean Opinion Score (dMOS). All of them are evaluated by the open-source toolkit DNSMOS [37], [38], which is widely used in Deep Noise Suppression (DNS) challenge[3]. Table I shows values of these metrics of the unprocessed noisy development and evaluation sets.

D. Comparison of SE Systems in Different Domains

Fig. 3 and Fig. 4 show the SIG, OVRL, and BAK values of different SE systems in real development and evaluation sets in seen and unseen scenarios, respectively. "DEMUCS" outperforms "Bi-LSTM" on almost all noise conditions. It achieved better speech signal recovery, overall quality recovery, and noise suppression. SIG and OVRL were affected by noise conditions, especially on the evaluation set. The performance of most evaluation metrics is degraded under unseen conditions: the BUS noise condition was the most challenging.

Although "DEMUCS" achieves better speech signal recovery (SIG), overall quality recovery (OVRL), and noise suppression (BAK) than "Bi-LSTM", it is not as good at the dMOS values. Fig. 5 and Fig. 6 show the dMOS values in development and evaluation sets in seen and unseen scenarios, respectively. The waveform-domain SE system is sensitive to noise conditions. This trend is not obvious in the simulated data sets, but is evident in the real data sets. For the simulated noisy sets, "DEMUCS" outperforms "Bi-LSTM" in almost all noise conditions. Its superiority is, however, diminished for the real noisy sets. "DEMUCS" shows a significant degradation for all PED and CAF noise conditions. This may be due to the large difference in data distribution between the training set and the evaluation sets in these two noise conditions. The frequency-domain model showed robustness against unseen noise conditions.

The dMOS degradation of "DEMUCS" may be due to the introduction of artifacts. Fig. 7 shows the magnitude of the spectrogram enhanced by different SE systems. The spectrogram enhanced by "Bi-LSTM" still contains much noise. Although the low-frequency speech signal recovery quality of the "DEMUCS" is higher than the "Bi-LSTM", the high-frequency part introduces noticeable artificial noise.

E. Effect of Spectrogram Encoding

The proposed WaveSpecEnc system combines the advantages of waveform-domain and frequency-domain SE systems. In Fig. 3 and Fig. 4, "WaveSpecEnc" outperforms "Bi-LSTM" on all SIG, OVRL, and BAK evaluation metrics. Compared to "DEMUCS", the proposed system further improves SIG and OVRL by introducing spectral information. The proposed system had a slight improvement on BAK compared to "DEMUCS". For the dMOS value in Fig. 5 and Fig. 6, the proposed "WaveSpecEnc" performed best in all simulated noise conditions compared to "Bi-LSTM" and "DEMUCS". For real noisy conditions in development and evaluation sets, although the method proposed had slightly worse than "Bi-LSTM" in the PED noise condition, there were still large improvements from the waveform-domain "DEMUCS". This result shows that incorporating spectral information into the

[2] https://github.com/facebookresearch/demucs/blob/main/demucs/hdemucs.py

[3] https://www.microsoft.com/en-us/research/academic-program/deep-noise-suppression-challenge-icassp-2023/

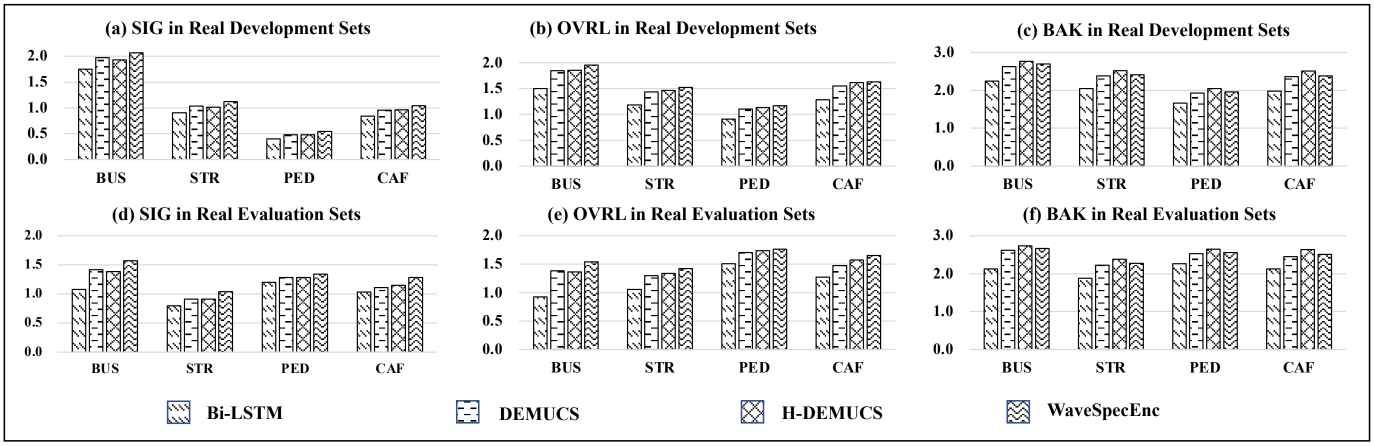


Fig. 3. (Seen) Relatively improvement of SIG / OVRL / BAK values (↑) compared with non-enhanced signals (Table I) in real development and evaluation sets. All noise conditions are SEEN to the model.

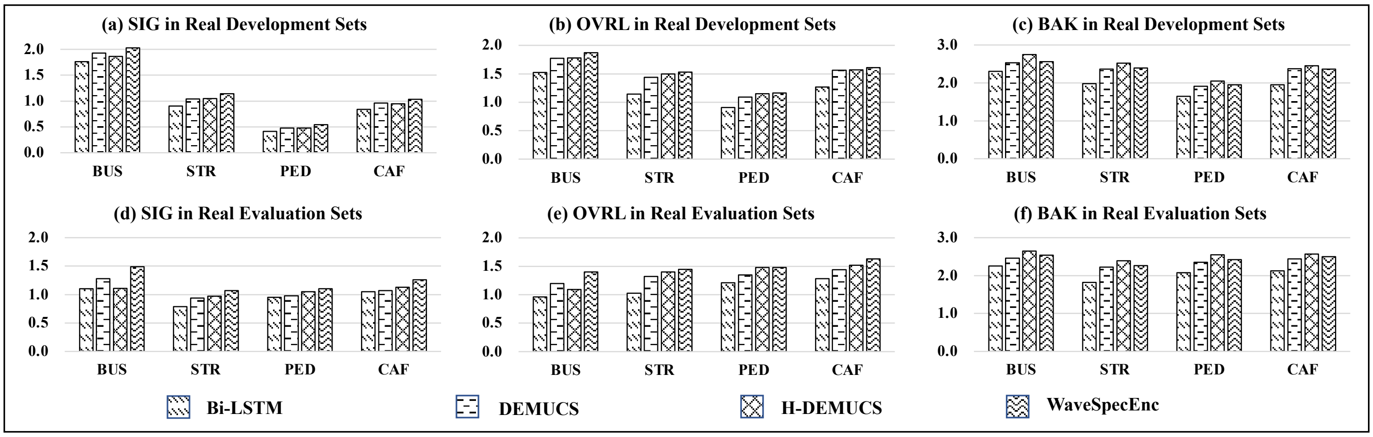


Fig. 4. (Unseen) Relatively improvement of SIG / OVRL / BAK values (↑) compared with non-enhanced signals (Table I) in real development and evaluation sets. The test noise conditions are UNSEEN to the model.

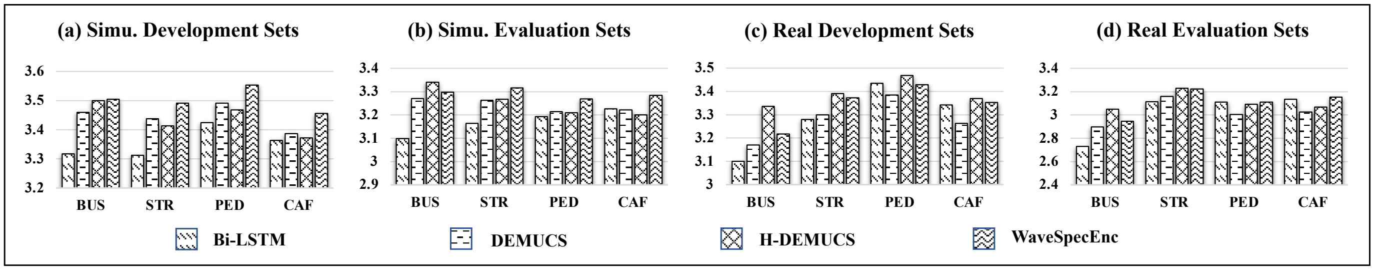


Fig. 5. (Seen) dMOS values (↑) in simulated and real sets. All noise conditions are SEEN to the model.

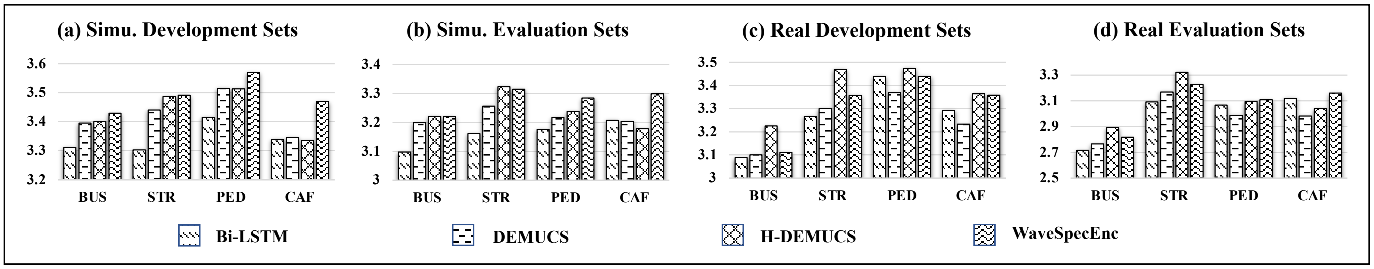
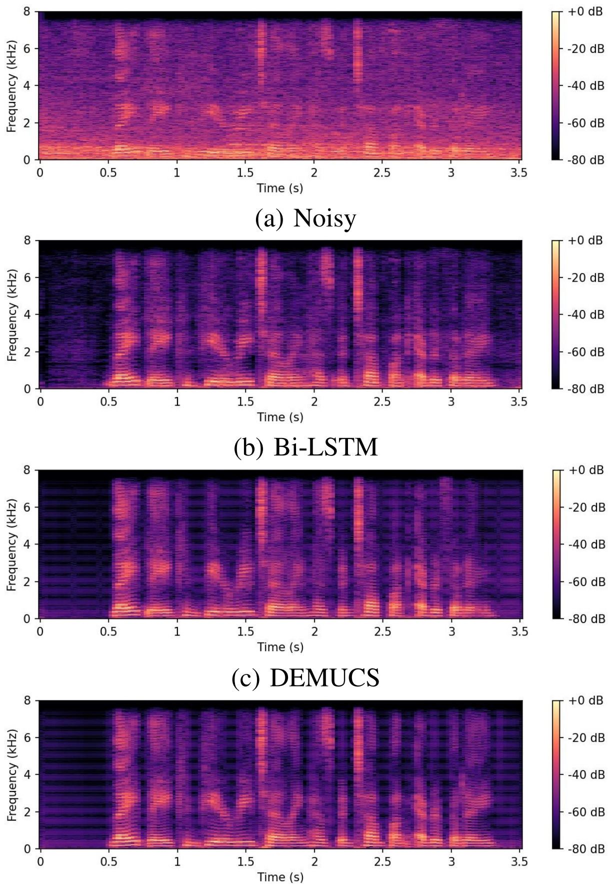


Fig. 6. (Unseen) dMOS values (↑) in simulated and real sets. The test noise conditions are UNSEEN to the model.



(a) Noisy

(b) Bi-LSTM

(c) DEMUCS

(d) WaveSpecEnc

Fig. 7. Enhanced magnitude spectrograms of the pretrained SE front-end. The clip is a real noisy speech under PED noise condition: (a) Noisy, (b) Bi-LSTM enhanced, (c) DEMUCS enhanced, and (d) WaveSpecEnc enhanced.

waveform-domain SE system can improve the stability of the waveform-domain SE system. Furthermore, the proposed method had a similar low-frequency restoration ability with "DEMUCS", which is shown in Fig. 7. The difference between the middle and high frequencies was more noticeable, especially the high-frequency artificial noise introduced by "DEMUCS" was significantly suppressed. This indicates that spectrogram encoding helps to reduce the introduction of artificial noise.

Different ways of combining spectral and temporal information show varying performances on different evaluation metrics. Compared to "H-DEMUCS", "WaveSpecEnc" exhibited better speech signal restoration and overall quality restoration (SIG, OVRL), while "H-DEMUCS" showed better noise reduction ability (BAK). These two methods have their respective strengths and weaknesses in terms of dMOS improvement.

V. Experimental Evaluations of SE-based Adaptation for Noise-mismatched ASR Back-end

Compared with the SE front-end, the ASR back-end has much more parameters, and thus the necessary amount of the training data for the ASR back-end is usually far more than that of the SE front-end. Moreover, in many practical applications, it is often not allowed to finetune the ASR back-end but only possible to tune the SE front-end. In this Section, we first investigate an effective adaptation way to finetune the SE front-end only by freezing the ASR back-end when encountering a new noise scene. ASR performance can be improved by finetuning the SE front-end by propagating the ASR loss [39].

A. Experimental Settings

For the Conformer-based ASR back-end, the number of encoder layers was 6. In each encoder layer, the positional encoding layer type was the relative positional encoding module; the subsampling rate was 4 with 2 Conv2d layers; the dimension of multi-head attention was 512; the number of attention heads was 4; the number of units of position-wise feedforward was 2048; the activation function was swish; the dimension of the input LMFB was 80. The decoder was based on Transformer [40]. The number of decoder layers was 6. In each decoder layer, the dimension of multi-head attention was 512; the number of attention heads was 4; the hidden units number of position-wise feedforward was 2048. We used all transcripts of CHiME-4, WSJ0 [4] and WSJ1 [5] to define a dictionary. The size of BPE vocabulary was 1014 including the $< blank >$, $< unk >$, and $< sos/eos >$.

Pretraining ASR back-end: WSJ0, WSJ1, and Librispeech (960 hours) [41] were used for ASR back-end pretraining. When pretraining, noises from the MUSAN dataset [42] were mixed with clean speech. The signal-to-noise ratio (SNR) was randomly selected between 0 and 20 dB. The ASR back-end was trained with 100 epochs. SpecAug [43] was applied during training. Ten checkpoints that performed well on the development set were averaged as the final pretrained model. The hyperparameter β in Eq. (8) was set to 0.7 according to the default setting of the ESPnet toolkit.

Finetuning SE front-end by freezing ASR back-end: Then, SE front-end was fintuned using the ASR loss. In this Section, the ASR backend was not changed. The finetuning was performed with 70 epochs under the same training condition with CHiME–4 training data of front-end training. The development sets of the corresponding noise conditions of CHiME–4 were used to select the model according to the minimum loss.

Word error rate (WER) was used to evaluate the ASR performance.

B. Evaluation in ASR

Table II and Table III show the WER in real development and evaluation sets. Directly using a cascade system (upper half of Table II) built with the pretrained SE front-end and ASR back-end significantly degraded the recognition performance, because the test noise in CHiME–4 dataset significantly differs from the noise used for pretraining SE front-end and the ASR back-end. The joint training (lower half of Table II) significantly improved recognition performance. "DEMUCS"-based front-end performs better than the "Bi-LSTM"-based front-end in almost all noise conditions in ASR,

[4] https://catalog.ldc.upenn.edu/LDC93s6a

[5] https://catalog.ldc.upenn.edu/LDC94S13A

TABLE II

(Seen) Word Error Rate (%, ↓) in real development and evaluation sets. All noise conditions are seen to the model. FT represents whether the front-end has been finetuned. The back-end is not finetuned in this experiment.

Systems	FT	Real Development					Real Evaluation				
		BUS	STR	PED	CAF	AVG	BUS	STR	PED	CAF	AVG
Conformer (pretrained, fixed back-end)	✗	18.3	10.8	9.1	12.7	12.7	27.0	14.1	19.6	22.4	20.8
Bi-LSTM	✗	27.9	19.6	15.5	22.1	21.3	61.9	26.3	34.9	40.6	40.9
DEMUCS	✗	24.4	18.4	14.0	20.1	19.2	44.8	24.1	36.1	41.8	36.7
H-DEMUCS	✗	28.8	20.7	16.6	19.8	21.5	51.9	26.7	35.0	38.2	37.9
WaveSpecEnc	✗	21.9	14.0	12.4	17.1	16.3	45.6	23.7	37.4	37.7	36.1
Bi-LSTM	✓	13.3	8.1	6.8	8.4	9.2	21.4	10.1	13.5	16.1	15.3
DEMUCS	✓	11.7	7.7	6.3	6.9	8.2	19.2	9.2	12.8	14.9	14.0
H-DEMUCS	✓	12.2	7.7	6.7	7.3	8.5	20.9	10.7	13.9	15.2	15.2
WaveSpecEnc	✓	10.8	7.3	6.3	6.7	7.8	18.1	9.4	12.6	13.9	13.5
WaveSpecEnc+	✓	10.4	7.1	6.1	6.3	7.5	17.0	8.6	11.8	13.3	12.7

TABLE III

(Unseen) Word Error Rate (%, ↓) in real development and evaluation sets. The test noise conditions are unseen to the model. Compared with the seen results in Table II, the relative decrease percentage of WER under the unseen testing (Decrease). FT represents whether the front-end has been finetuned. The back-end is not finetuned in this experiment.

Systems	FT	Real Development						Real Evaluation					
		BUS	STR	PED	CAF	AVG	Decrease	BUS	STR	PED	CAF	AVG	Decrease
Conformer (pretrained)	✗	18.3	10.8	9.1	12.7	12.7	-	27.0	14.1	19.6	22.4	20.8	-
Bi-LSTM	✓	14.0	8.5	7.3	8.8	9.7	5.4%	24.7	10.7	13.7	15.9	16.2	5.9%
DEMUCS	✓	14.4	7.9	6.6	7.1	9.0	9.8%	26.0	10.2	13.3	15.4	16.2	15.7%
H-DEMUCS	✓	15.0	7.7	6.8	7.4	9.2	8.2%	30.3	10.1	13.8	14.6	17.2	13.2%
WaveSpecEnc	✓	14.0	7.6	6.2	7.4	8.8	12.8%	29.7	9.9	13.0	14.2	16.7	23.7%
WaveSpecEnc+	✓	13.3	7.0	6.0	6.6	8.2	9.3%	23.0	9.0	11.8	13.7	14.4	13.4%

although it was not good at human hearing experiences under the PED and CAF noise conditions, which is evident in real evaluation sets.

The performance of the ASR system is degraded largely when tested under unseen conditions, as shown in Table III. It shows that the degradation of "DEMUCS" is much larger than "Bi-LSTM". In particular, when the BUS noise data is not involved in the training, the "DEMUCS" had a significant performance degradation. This is the case with the proposed "WaveSpecEnc", although spectrogram encoding gives a considerable performance improvement in other noise conditions. This may be because the BUS noise condition was the most adversary as shown in Table I. While "H-DEMUCS" greatly enhanced the dMOS, it did not bring good performance for ASR.

Incorporating spectrogram encoding information into the ASR back-end, "WaveSpecEnc+", significantly and consistently improved ASR performance under all noise conditions. It is also effective in the most challenging BUS condition. This result confirms that incorporating spectrogram encoding not only in the SE front-end but also ASR back-end is crucial. This proposed method significantly outperformed all other methods (p-value < 0.01), although the baseline "Bi-LSTM" shows better robustness (least decrease from the seen condition).

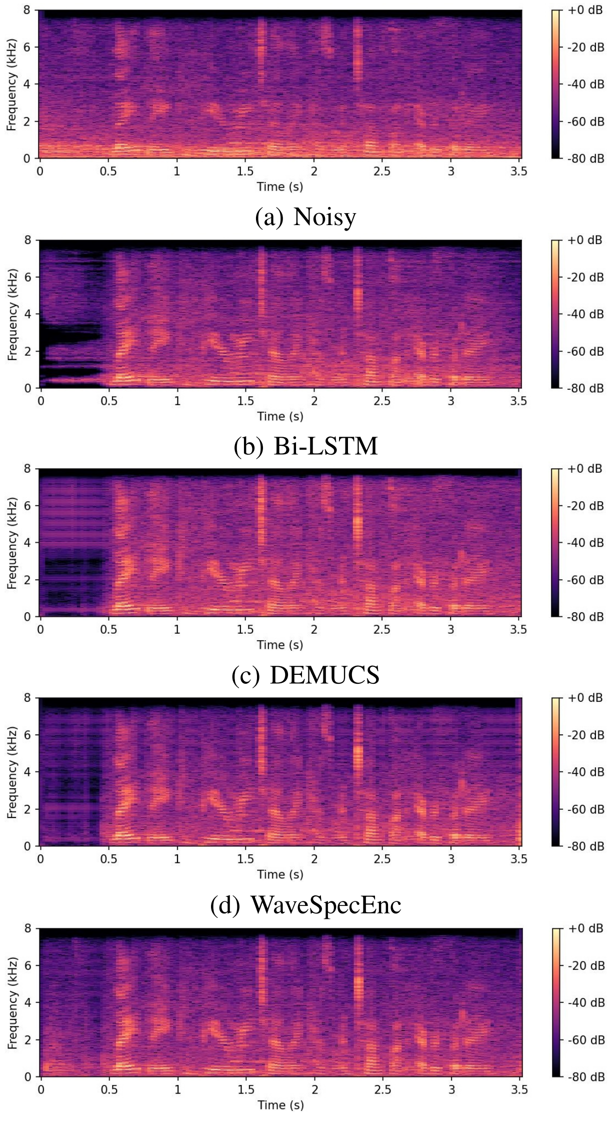
Fig. 8 shows the enhanced magnitude spectrograms of different SE front-ends after finetuning. The front-end output after finetuning is similar to the noisy spectrogram in the speech parts, but the energy of some speech information is more prominent in the enhanced features. This shows that the front-end SE with finetuning preserves the speech signal as much as possible while highlighting the effective ASR-related speech components. Compared with the other enhancement front-ends, the speech components are not highlighted in the "Bi-LSTM" spectrogram. Moreover, some high-frequency information is blurred. The spectrogram of "DEMUCS" introduces artificial noise in the high-frequency parts. "WaveSpecEnc" has some noise reduction effect, but "WaveSpecEnc+" has better noise reduction. This is because the SE finetuning keeps the information of the spectrogram encoding intact and removes adversary artificial noise. We also conducted the dMOS evaluation for the finetuned SE front-end. The evaluation showed that the finetuned SE front-end considerably decreased performance compared with the pretrained front-end model.

C. *Effect of Fusion Layers in ASR Back-end*

"WaveSpecEnc+" incorporates the spectrogram encoding in the first encoder layers of the ASR back-end. The layer-by-layer fusion was compared in Table IV. Fusion in many layers is not so effective for improving ASR performance. Despite ASR performance improvements observed in all models, deep-level incorporation of the spectrogram encoding did not yield noticeable gains. Instead, the most significant improvement was obtained when incorporating the spectrogram encoding at the shallow layer. As the encoder layers in the ASR model become deeper, information within these layers tends to be close to linguistic. In contrast, the shallow layers contain mostly environmental and noise-related details; thus, fusing in the shallow layers shows more effective.

TABLE IV
NUMBER OF CONFORMER LAYERS WITH SPECTROGRAM ENCODING (NUM.). ALL NOISE CONDITIONS ARE SEEN TO THE MODEL.

Num.	Fusion Layers of Freezeed Conformer Encoder Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Development BUS	STR	PED	CAF	AVG	Evaluation BUS	STR	PED	CAF	AVG
1	✓						10.4	7.1	6.1	6.3	7.5	17.0	8.6	11.8	13.3	12.7
2	✓	✓					10.6	6.8	6.0	6.2	7.4	17.7	9.0	12.0	13.3	13.0
3	✓	✓	✓				10.8	6.8	6.1	6.5	7.5	18.2	8.9	11.9	13.5	13.1
4	✓	✓	✓	✓			10.3	6.9	6.2	6.1	7.4	17.8	8.7	11.8	13.4	12.9
5	✓	✓	✓	✓	✓		10.3	6.8	6.1	6.1	7.3	17.7	8.8	11.8	13.2	12.9
6	✓	✓	✓	✓	✓	✓	10.8	7.0	6.4	6.7	7.7	18.0	9.0	12.2	13.7	13.2



(a) Noisy

(b) Bi-LSTM

(c) DEMUCS

(d) WaveSpecEnc

(e) WaveSpecEnc+

Fig. 8. Enhanced magnitude spectrograms of SE front-end after joint training. The clip is a real noisy speech under PED noise condition: (a) Noisy, (b) Bi-LSTM enhanced, (c) DEMUCS enhanced, (d) WaveSpecEnc enhanced, (e) WaveSpecEnc+ enhanced.

VI. EXPERIMENTAL EVALUATIONS OF FINETUNING BOTH SE FRONT-END AND ASR BACK-END

Finetuning the ASR back-end using data from a new noise environment is a direct and effective adaptation method. In this Section, we simultaneously finetune both the SE front-end and ASR back-end using the CHiME-4 dataset. It should be noted that this is possible when a large dataset is available. "Conv-Tasnet" [44] is also compared as the SE front-end, which was pretrained with the simulated CHiME–4 data with 100 epochs. The hyperparameter settings were the same as those of ESPnet [6].

A. Experimental Settings

We tried two types of acoustic models. The first one was the same as the pretrained model in Section V: the Conformer pretrained with Librispeech-960 and MUSAN noise. The second one is WavLM [45].

Finetuning Conformer ASR and SE: We conducted joint training, in which the SE front-end and the ASR back-end parameters were finetuned using the CHiME–4 dataset. All simulated and real data from the training set were used.

Moreover, we also incorporated the language model (LM) to further improve the ASR performance. We utilized the transformer-based LM. It contained 16 encoder layers. In each encoder layer, there was no positional encoding layer; the dimension of multi-head attention was 512; the number of attention heads was 8; the number of units of position-wise feedforward was 2048. The training text consisted of two parts: the first part included text data extracted from the CHiME-4 training set; the second part was obtained from the "wsj1_lng", totaling approximately 1.7 million text samples. Shallow fusion was adopted to integrate the LM and acoustic model with a fusion weight of 0.6 and 0.4.

Finetuning WavLM and SE: We used the pretrained model checkpoint available on HuggingFace [7], which consists of 24 layers of Transformer architecture. We used the character-based dictionary. All parameters were finetuned with the CTC loss. The WaveSpecEnc+ spectrogram encoding is approximately 1.25 times the number of feature frames extracted by the WavLM feature extraction module. Thus, the features need to be time-aligned. We simply drop the sixth frame after every five consecutive frames of the spectrogram encoding.

B. Evaluation in ASR

Table V shows the ASR performance for real sets. Finetuning the ASR back-end using the CHiME–4 dataset significantly enhanced ASR performance. Additionally, incorporating an additional LM further improved the performance.

[6] https://github.com/espnet/espnet/blob/master/egs2/chime4/enh1/conf/tuning/train_enh_conv_tasnet.yaml

[7] https://huggingface.co/microsoft/wavlm-large

TABLE V
WORD ERROR RATE (%, ↓) IN REAL DEVELOPMENT AND EVALUATION SETS. ALL NOISE CONDITIONS ARE SEEN TO THE MODEL. LM DENOTES WHETHER TO USE THE EXTERNAL LANGUAGE MODEL.

Systems	**LM**	**Real Development**					**Real Evaluation**				
		BUS	**STR**	**PED**	**CAF**	**AVG**	**BUS**	**STR**	**PED**	**CAF**	**AVG**
Conformer (pretrained)	-	18.3	10.8	9.1	12.7	12.7	27.0	14.1	19.6	22.4	20.8
Finetuned	-	12.0	10.0	9.0	10.8	10.5	18.5	10.9	14.4	15.8	14.9
Finetuned	✓	9.3	7.8	7.4	8.7	8.3	13.9	9.2	11.2	12.6	11.7
Bi-LSTM	✓	8.6	8.2	7.8	9.3	8.5	12.6	8.8	9.9	10.7	10.5
Conv-Tasnet	✓	7.4	6.9	6.7	7.0	7.0	11.0	8.3	8.5	10.1	9.5
DEMUCS	✓	7.8	7.4	6.7	7.3	7.3	11.3	8.0	8.6	10.4	9.6
H-DEMUCS	✓	7.9	7.2	6.7	7.4	7.3	11.6	8.0	8.6	9.7	9.5
WaveSpecEnc	✓	7.6	7.3	6.7	7.4	7.3	10.5	8.2	8.6	9.6	9.2
WaveSpecEnc+	✓	7.6	7.4	7.1	7.5	7.4	10.3	7.5	7.9	9.2	8.7

TABLE VI
WORD ERROR RATE (%, ↓) IN REAL DEVELOPMENT AND EVALUATION SETS. ALL NOISE CONDITIONS ARE SEEN TO THE MODEL. WAVLM IS ADOPTED AS THE ACOUSTIC MODEL.

Systems	**Real Development**					**Real Evaluation**				
	BUS	**STR**	**PED**	**CAF**	**AVG**	**BUS**	**STR**	**PED**	**CAF**	**AVG**
WavLM	6.3	5.4	4.6	5.2	5.4	8.2	5.8	6.7	6.6	6.8
Bi-LSTM	5.8	4.3	3.6	4.5	4.5	7.7	4.5	5.9	6.1	6.0
Conv-Tasnet	6.1	4.3	3.9	4.2	4.6	8.0	5.0	6.0	6.0	6.2
DEMUCS	5.6	4.8	3.6	4.2	4.5	7.8	4.6	5.7	5.8	6.0
H-DEMUCS	5.8	4.2	3.6	4.0	4.4	7.7	4.8	5.9	5.9	6.1
WaveSpecEnc	5.5	4.8	3.5	4.1	4.5	7.4	4.8	6.2	5.6	6.0
WaveSpecEnc+	5.6	4.5	3.5	4.0	4.4	7.1	4.5	5.8	5.8	5.8

Therefore, in subsequent experiments, we employed LM during decoding.

From this new baseline, the "Bi-LSTM"-based system did not show significant improvement for the evaluation set. "DEMUCS" showed a notable improvement. Compared to finetuning the SE front-end only, jointly optimizing the ASR back-end and the SE front-end led to large performance improvements for "H-DEMUCS". While "WaveSpecEnc" slightly performs better than these, "WaveSpecEnc+" significantly outperformed "DEMUCS" and "H-DEMUCS" in the real evaluation set (p-value < 0.01) since it integrates effective information into the ASR back-end. In addition, we also compared "Conv-Tasnet"-based ASR system. The "Conv-Tasnet" system showed slightly better performance in real development sets (p-value > 0.05), but note that the ASR model was selected according to the checkpoints in the development set. On the other hand, the proposed "WaveSpecEnc+" significantly outperformed "Conv-Tasnet" in real evaluation sets (p-value < 0.01).

Table VI shows the ASR performance with the WavLM-based acoustic model. Pretraining based on self-supervised learning significantly improved the ASR performance. The SE front-ends still significantly improved the performance of ASR. Although the performance difference among the SE methods is small, the proposed "WaveSpecEnc+" resulted in the best performance in the real evaluation set.

C. Comparison Between Different ASR Systems

Table VII lists the performance of different ASR systems under single-channel conditions for the CHiME–4 evaluations. DNN-HMM Hybrid ASR systems perform better with a small amount of data than end-to-end ASR systems. We expect the end-to-end model to perform better with a large amount of training data. Pretraining based on self-supervised learning solves this problem. Particularly, noise-aware pretrained model, such as WavLM, is effective. We have demonstrated the effectiveness of the proposed method (WaveSpecEnc+) in this setting as well. While our ASR back-end is based on simple CTC in Table VI, "IRIS" and "Transformer - WavLM" used WavLM as a feature extractor and incorporated additional Transformer layers for ASR. They also adopted an external

TABLE VII
COMPARISON BETWEEN DIFFERENT SINGLE-CHANNEL AUTOMATIC SPEECH RECOGNITION SYSTEMS (WORD ERROR RATE, %, ↓).

Systems	**SSL**	**Dev. Set**		**Eval. Set**	
		Simu.	**Real**	**Simu.**	**Real**
DNN-HMM Hybrid ASR					
Kaldi [46]	✗	6.8	5.6	12.2	11.4
Yang *et al.* [47]	✗	5.0	3.4	8.6	6.3
Wang *et al.* [33]	✗	5.0	3.5	9.4	6.8
End-to-End ASR					
ESPnet (Conformer)	✗	11.3	9.2	16.8	15.9
IFF-Net [24]	✗	7.9	6.4	13.4	12.4
DPSL-ASR [48]	✗	7.2	5.9	12.2	11.3
WaveSpecEnc+ (this study)	✗	7.3	7.4	12.0	8.7
Transformer - HuBERT [49]	✓	11.6	9.1	18.0	20.4
Transformer - WavLM [49]	✓	5.9	4.0	8.3	4.5
IRIS [49]	✓	3.2	2.0	6.1	3.9
WaveSpecEnc+ - WavLM (this study)	✓	3.0	4.4	5.7	5.8
WaveSpecEnc+ - WavLM - Transformer - LM (this study)	✓	3.3	2.1	6.3	3.7

LM. For fair comparison, we also conducted an experiment based on the IRIS pipeline and replaced IRIS's ConvTasnet with the proposed front-end. Experimental results confirm that the system based on the proposed WaveSpecEnc+ performs better for the real evaluation set, though there is no significant difference for all test sets. Although the ASR performance is almost saturated with the strong ASR back-end, the effect of the proposed front-end was more clearly observed with the lightweight ASR back-end in Table V and VI.

VII. Conclusions

In this paper, we improve the robustness of waveform-domain speech enhancement SE with spectrogram encoding ("WaveSpecEnc"). The temporal feature maps at each encoder layer in the SE front-end are refined by spectral information. The proposed time-spectrogram hybrid system improved the dMOS score. Artificial noise introduced by the waveform-domain SE front-end can be reduced by the using of spectrogram-domain information. However, "WaveSpecEnc"-based ASR system had minor improvement over the "DEMUCS"-based ASR system. Thus, we incorporate the spectral information of the encoder layer into the ASR back-end ("WaveSpecEnc+"). Compared with "DEMUCS", "WaveSpecEnc+" significantly improved ASR performance in all noise conditions on CHiME–4 evaluation sets. Several acoustic models were used to evaluate the effectiveness of "WaveSpecEnc+"-based ASR systems. Firstly, the experimental results with a frozen pre-trained acoustic model showed that incorporating spectrogram encoding in the ASR back-end is crucial. It is effective to fuse features in only shallow encoder layers of the Conformer-based ASR system. Secondly, the SE front-end and the pre-trained acoustic model were jointly fine-tuned with the CHiME–4 training set. The experimental results showed that integrating the spectral encoding into the ASR back-end is still effective. Thirdly, we also tried WavLM as the acoustic model. The experimental results showed that the SE front-ends still improved the ASR performance, although the performance differences among the SE front-ends were small. Finally, we replaced the "Conv-Tasnet" in the IRIS system with our proposed "WaveSpecEnc+". Experimental results confirm that the system based on the "WaveSpecEnc+" performs better for the real evaluation set.

References

[1] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2007.

[2] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 4, pp. 745–777, 2014.

[3] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech Enhancement with LSTM Recurrent Neural Networks and its Application to Noise-Robust ASR," in *LVA/ICA*, 2015, pp. 91–99.

[4] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 1, pp. 7–19, 2015.

[5] H. Shi, L. Wang, M. Ge, S. Li, and J. Dang, "Spectrograms fusion with minimum difference masks estimation for monaural speech dereverberation," in *Proc. ICASSP*, 2020, pp. 7544–7548.

[6] S. Kamath and P. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise," in *Proc. ICASSP*, vol. 4, 2002, pp. IV–4164–IV–4164.

[7] Y. Ephraim and H. Van Trees, "A signal subspace approach for speech enhancement," *IEEE Speech Audio Process.*, vol. 3, no. 4, pp. 251–266, 1995.

[8] R. Martin, "Speech enhancement using MMSE short time spectral estimation with gamma distributed speech priors," in *Proc. ICASSP*, vol. 1, 2002, pp. I–253–I–256.

[9] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder." in *Proc. Interspeech*, vol. 2013, 2013, pp. 436–440.

[10] A. Défossez, G. Synnaeve, and Y. Adi, "Real Time Speech Enhancement in the Waveform Domain," in *Proc. Interspeech*, 2020, pp. 3291–3295.

[11] A. Pandey and D. Wang, "A new framework for CNN-based speech enhancement in the time domain," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 7, pp. 1179–1188, 2019.

[12] 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*, 2020, pp. 2472–2476.

[13] K. Tan and D. Wang, "Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement," in *Proc. ICASSP*, 2019, pp. 6865–6869.

[14] T. Gerkmann, M. Krawczyk-Becker, and J. Le Roux, "Phase processing for single-channel speech enhancement: History and recent advances," *IEEE IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 55–66, 2015.

[15] K. Tan and D. Wang, "A convolutional recurrent neural network for real-time speech enhancement," in *Proc. Interspeech*, 2018, pp. 3229–3233.

[16] M. Kolbæk, Z.-H. Tan, S. H. Jensen, and J. Jensen, "On loss functions for supervised monaural time-domain speech enhancement," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 825–838, 2020.

[17] Y. Zhao and D. Wang, "Noisy-Reverberant Speech Enhancement Using DenseUNet with Time-Frequency Attention," in *Proc. Interspeech*, 2020, pp. 3261–3265.

[18] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 65–68, 2014.

[19] H. Shi, M. Mimura, L. Wang, J. Dang, and T. Kawahara, "Time-domain speech enhancement assisted by multi-resolution frequency encoder and decoder," in *Proc. ICASSP*, 2023.

[20] A. Défossez, "Hybrid spectrogram and waveform source separation," in *Proc. ISMIR*, 2021.

[21] H. Shi and T. Kawahara, "Exploration of adapter for noise robust automatic speech recognition," *arXiv preprint arXiv:2402.18275*, 2024.

[22] C. Chen, N. Hou, Y. Hu, S. Shirol, and E. S. Chng, "Noise-robust speech recognition with 10 minutes unparalleled in-domain data," in *Proc. ICASSP*, 2022, pp. 4298–4302.

[23] C. Fan, J. Yi, J. Tao, Z. Tian, B. Liu, and Z. Wen, "Gated recurrent fusion with joint training framework for robust end-to-end speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 198–209, 2021.

[24] Y. Hu, N. Hou, C. Chen, and E. Siong Chng, "Interactive feature fusion for end-to-end noise-robust speech recognition," in *Proc. ICASSP*, 2022, pp. 6292–6296.

[25] H. Shi, L. Wang, S. Li, C. Fan, J. Dang, and T. Kawahara, "Spectrograms fusion-based end-to-end robust automatic speech recognition," in *Proc. APSIPA ASC*, 2021, pp. 438–442.

[26] A. Graves, *Long Short-Term Memory*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 37–45.

[27] J. Smith and P. Gossett, "A flexible sampling-rate conversion method," in *Proc. ICASSP*, vol. 9, 1984, pp. 112–115.

[28] Y.-H. Tu, J. Du, and C.-H. Lee, "Speech enhancement based on teacher–student deep learning using improved speech presence probability for noise-robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 12, pp. 2080–2091, 2019.

[29] 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*, 2020, pp. 5036–5040.

[30] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 8, pp. 1240–1253, 2017.

[31] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration," in *Proc. Interspeech*, 2019, pp. 1408–1412.

[32] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust asr," in *Proc. ICASSP*, 2012, pp. 4085–4088.

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

[34] S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda, and T. Nakatani, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *Proc. ICASSP*, 2015, pp. 116–120.

[35] C. Li, J. Shi, W. Zhang, A. S. Subramanian, X. Chang, N. Kamo, M. Hira, T. Hayashi, C. Boeddeker, Z. Chen, and S. Watanabe, "Espnet-se: End-to-end speech enhancement and separation toolkit designed for asr integration," in *Proc. SLT*, 2021, pp. 785–792.

[36] 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, 2008.

[37] C. K. Reddy, V. Gopal, and R. Cutler, "DNSMOS: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *Proc. ICASSP*, 2021, pp. 6493–6497.

[38] ——, "DNSMOS P.835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *Proc. ICASSP*, 2022, pp. 886–890.

[39] J. Woo, M. Mimura, K. Yoshii, and T. Kawahara, "End-to-end music-mixed speech recognition," in *Proc. APSIPA ASC*, 2020, pp. 800–804.

[40] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. ICASSP*, 2018, pp. 5884–5888.

[41] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[42] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," *arXiv:1510.08484*, 2015.

[43] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

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

[45] S. Chen, Z. Wang, C.and Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1505–1518, 2022.

[46] 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," *arXiv:1803.10109*, 2018.

[47] Y. Yang, P. Wang, and D. Wang, "A Conformer Based Acoustic Model for Robust Automatic Speech Recognition," *arXiv:2203.00725*, 2022.

[48] Y. Hu, N. Hou, C. Chen, and E. S. Chng, "Dual-Path Style Learning for End-to-End Noise-Robust Speech Recognition," *arXiv:2203.14838*, 2023.

[49] X. Chang, T. Maekaku, Y. Fujita, and S. Watanabe, "End-to-End Integration of Speech Recognition, Speech Enhancement, and Self-Supervised Learning Representation," *arXiv:2204.00540*, 2022.