*APSIPA Transactions on Signal and Information Processing, 2025, 14, e23*

## Original Paper

# Time-domain Separation Priority Pipeline-based Cascaded Multi-task Learning for Monaural Noisy and Reverberant Speech Separation

Shaoxiang Dang[1*], Tetsuya Matsumoto[1], Yoshinori Takeuchi[2] and Hiroaki Kudo[1]

[1] *Graduate School of Informatics, Nagoya University, Japan*
[2] *School of Informatics, Daido University, Japan*

ABSTRACT

Monaural speech separation is a crucial task in speech processing, focused on isolating single-channel audio with multiple speakers into individual streams. This problem is particularly challenging in noisy and reverberant environments where the target information becomes obscured. Cascaded multi-task learning breaks down complex tasks into simpler sub-tasks and leverages additional information for step-by-step learning, serving as an effective approach for integrating multiple objectives. However, its sequential nature often leads to over-suppression, degrading the performance of downstream modules. This article presents three main contributions. First, we propose a separation-priority pipeline to ensure that the critical separation sub-task is preserved against over-suppression. Second, to extract deeper multi-scale features, we design a consistent-stride deep encoder-decoder structure combined with depth-wise multi-receptive field fusion. Third, we advocate a training strategy that pre-trains each sub-task and applies time-varying and time-invariant weighted fine-tuning to further mitigate

[*] Corresponding author: Shaoxiang Dang, dang.shaoxiang.s0@s.mail.nagoya-u.ac.jp

Received 09 April 2025; revised 26 May 2025; accepted 08 June 2025
ISSN 2048-7703; DOI 10.1561/116.20250022

over-suppression. Our methods are evaluated on the open-source Libri2Mix and real-world LibriCSS datasets. Experimental results across diverse metrics demonstrate that all proposed innovations improve overall model performance.

*Keywords:* Speech separation, speech enhancement, cascaded multi-task learning, separation priority pipeline, transfer learning

## 1 Introduction

Speech separation (SS) emerged as a solution to the cocktail party problem [5], which highlights people's ability to effortlessly follow the desired speaker despite the presence of interfering speakers and background noises. Therefore, SS aims to disentangle speech information in scenarios where multiple speakers are conversing simultaneously [54]. Monaural speech separation further narrows this scope to the specific scenario where only one microphone is available to record mixed audio [23]. Deep learning techniques have facilitated the progress of SS at an accelerated pace [3, 24, 25]. In contrast to other domains where deep learning prevails, obtaining clean speech labels for SS is challenging because it is not an easy job to gain high-quality individual speeches from the mixed speech which has already been collected in advance. Therefore, most studies simulate mixed speeches by temporally adding sources, while simultaneously preserving the sources as separation labels [19, 62, 28, 16]. In the context of generating a massive number of samples via this way, models have the option to base their processing on traditional time-frequency features or learned features. Time-frequency domain separation models typically first compute the spectrogram of the mixed speech. Then, using the mixed speech spectrogram as the model's input to output source spectrograms. Typically, there are two learning objectives for the model: the mapping approach involves the model directly outputting the spectrograms of each speaker, whereas the masking approach generates a mask applied to the input mixture spectrogram to produce the target speaker's spectrogram. Finally, the waveform of the sources is reconstructed from their respective spectrograms [3, 25, 19, 62, 28, 16, 60, 15, 59, 57].

It is worth noting that this method typically requires additional consideration of phase information to achieve better results [20, 17]. Moreover, since the optimization objective of this approach is minimizing spectrogram error, frequency-domain methods do not offer any advantage in improving the signal-to-noise ratio. Time-domain separation methods use differentiable 1-dimensional convolutional and transposed 1-dimensional convolutional layers to bridge between the time-domain signal and feature space, hence they are also referred to as end-to-end (e2e) separation methods. Their innovation lies in the unification of both amplitude and phase of traditional handcrafted features, enabling models to learn features tailor-made for separation and directly optimize on signal-to-noise or signal-to-distortion metrics [36, 33, 35, 31, 2, 49, 52]. Because time-domain and frequency-domain models each have different areas of focus, the combination of time-domain and frequency-domain methods is also becoming increasingly popular [56, 55, 63].

With the growing popularity of e2e approaches, the encoder-decoder framework has also been extensively studied. Compared to the original single-layer encoder-decoder model, deep encoders have been found to be more robust in feature extraction [26]. Additionally, dilated convolutions have been employed to increase the receptive field within convolutional layers [43, 64, 42]. Despite thriving with the aid of e2e separation methods, SS in complex environments continues to be a concern for researchers. For instance, the challenge of separating mixed speech in the presence of loud background noise remains a significant issue [58]. The background noise interferes with models' ability to capture the desired information, resulting in dwindled results compared to clean environments. To resolve this dilemma, a natural solution is to introduce speech enhancement (SE) [48, 47] for mixed speech as a front-end [68, 51, 38, 22, 37]. The SE task aims to improve the quality of speech [13], encompassing various aspects such as noise cancellation and echo suppression. This article specifically defines SE as the removal of background noise. This strategy is prevalent in similar tasks, such as in mixed speech recognition tasks, where SS serves as the front-end and automatic speech recognition (ASR) as the back-end [39]; in enhancement tasks involving noise and reverberation, the SE module serves as the front-end, and the dereverberation (DE) module as the back-end [37, 67]. These sub-tasks are jointly trained via cascaded multi-task learning. While the additional utilization of noise-free mixtures as enhancement module labels in multi-task learning leads to an improvement in accuracy, SS with SE encounters a new issue: the over-suppression problem [22, 10]. For the separation module, although its input alters to noise-free mixed speech after the introduction of the SE module, the quality of mixed speech decreases to a certain degree due to the processing of the SE module. One hypothesis regarding gradient conflicts suggests that the divergent optimization directions of the SE task and the SS task impede overall performance. To address this, A learning patch is proposed to adjust the gradient weights of the front-end task (SE task) based on the more critical posterior task (SS task) [22, 10]. Unfortunately, learning patches require adjusting gradients layer by layer, which consumes excessive computational resources. Furthermore, this approach has been shown to compromise the model's generalization ability [10, 9].

This paper is an extended version of the conference paper [10]. In the conference paper, to mitigate the over-suppression problem in multi-task models for noisy speech separation, we proposed SPP, which prioritizes the separation module. SPP ensures the integrity of the input to the SS module, making it more effective. Additionally, no extra gradient modulation is required, and the labels for the swapped SS module are easily accessible. In this paper, we propose the SPP-based cascaded multi-task learning approach to address monaural speech separation in noisy and indoor environments. There are three main innovations. Firstly, we develop two SPP pipelines for time-domain monaural noisy and reverberant speech separation. They are SPP with enhancement-secondary (SPP-ES) and SPP with derevereberation-secondary (SPP-DS). Secondly, to enhance the model's processing ability, we propose a consistent stride deep encoder-decoder structure (CS-DEDS) and depth-wise multi-receptive field fusion (DW-MRFF) blocks. Finally, to further alleviate the over-suppression issue, we conduct pre-training for each sub-task, followed by an overall transfer learning approach with time-varying and time-invariant weights. We validate our theory and model on the publicly available simulated dataset Libri2Mix. The effectiveness of each innovation is confirmed across various metrics.

## 2 Problem Formulation and Related Works

### 2.1 Problem formulation

For a clear narrative, related terms are first introduced in Table 1. Supposing $s^r(t)$ and $n(t)$ denote the spatial image and ambient noise, respectively. Monaural noisy and reverberant mixture $x(t)$ can be formulated in the time domain by:

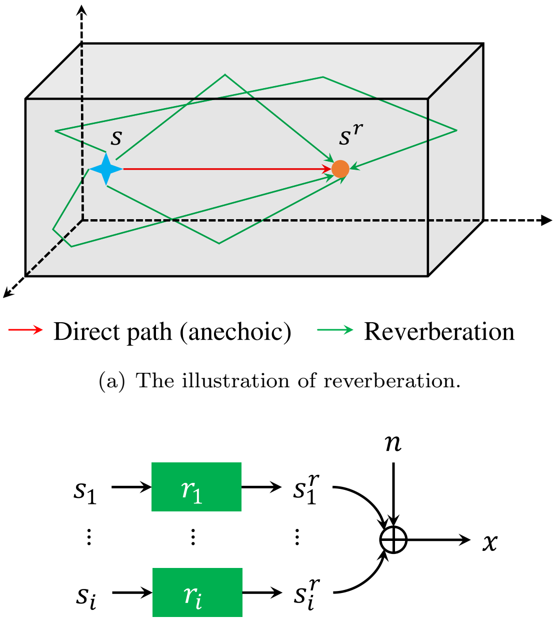
$$x(t) = \sum_{i=1}^{I} s_i^r(t) + n(t) \tag{1}$$

where $t$ and $I$ represent the discrete time and the number of sources, respectively. $i$-th reverberant signal can be decomposed as follows:

$$\begin{align} s_i^r(t) &= s_i(t) * r_i(t) \tag{2} \\ &= s_i(t) * (rd_i(t) + ru_i(t)) \tag{3} \\ &= s_i(t) * rd_i(t) + s_i(t) * ru_i(t) \tag{4} \\ &\triangleq s_i^d(t) + s_i^u(t) \tag{5} \end{align}$$

Table 1: Terminology explanation.

| Terms | | | Explanation | | |
|---|---|---|---|---|---|
| Noisy | **vs.** | Clean | Have ambient noise | **vs.** | Have no ambient noise |
| Anechoic | **vs.** | Reverberant | Direct path signal | **vs.** | Spatial image |
| Mixture | **vs.** | Source | Mixed signal | **vs.** | Individual signal |



(a) The illustration of reverberation.

(b) The simulation of noisy and reverberant mixture.

Figure 1: Problem formulation. Sub-figure (a) illustrates the generation process of indoor reverberation, where $s$ denotes the signal source and $s^r$ represents the receiver. The red arrow indicates the direct path, while the green arrows represent the reverberation paths. Sub-figure (b) depicts the simulation of noisy and reverberant mixture $x$, where $r_i$ and $n$ denote time-invariant RIR and ambient noise.

where $*$ denotes convolution operation. In this work, the position of source is static, so $r_i(t)$ denotes time-invariant room impulse response (RIR). RIR consists of the direct path $rd_i(t)$ and undesired reverberant path $ru_i(t)$. An illustration of reverberation is depicted in Figure 1(a). The objective of this paper is to separate each individual anechoic signal $s_i^d(t)$ from $x(t)$ by filtering out $s_i^u(t)$ and $n(t)$ [34, 7]. It is also worth noting that there is only a delay difference between the target signal $s_i^d(t)$ and the original signal $s_i(t)$.

### 2.2 End-to-end (e2e) separation methods

Time-domain audio separation network (TasNet) is the first e2e separation approach [36, 32]. This approach's architecture is composed of an encoder layer, a masking network, and a decoder layer. The encoder layer extracts features from the waveform, the masking network separates these features, and the decoder layer upsamples the features and reconstructs the waveform. A fully-convolutional structure-based masking network (Conv-TasNet) is developed in [35]. These blocks are stacked with exponentially increasing dilation factors to obtain multi-scale representation. To capture both local and global dependency, features are segmented into a 3-D tensor and processed by a dual-path recurrent neural network (DPRNN) architecture [31]. Later on, RNNs are replaced with transformer encoders, naming the model SepFormer [49]. In the context of complex environments of this paper, to distinguish the aforementioned methods from cascaded multi-task learning methods, they are also referred to as single-task methods.

### *2.3 Cascaded multi-task learning methods*

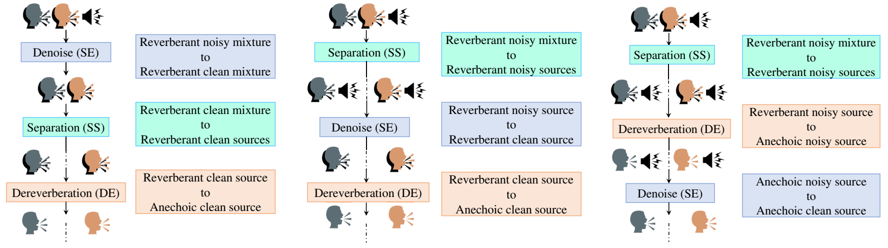
Cascaded multi-task learning involves decomposing complex task into several step-by-step simpler subtasks. Its success is mainly attributed to the leverage of more information compared to e2e separation methods [37]. The processing sequence of enhancement, separation, and dereverberation yields the best results in complex environmental separation. Additionally, two points are noteworthy: first, each module used an e2e architecture, meaning the input and output of each module were waveforms; second, the implementations of the processors used TasNet and Conv-TasNet. Based on this, shared encoder and decoder scheme (SEDS) streamlines the model, with intermediate processing operating in the time-frequency domain [38]. Moreover, an extra channel attention mechanism is added before the DP process. However, the sequential nature of cascaded multi-task learning causes it to suffer from the over-suppression problem, which means the SE module, while denoising, also loses other information that could potentially aid in separation.

To address the over-suppression problem, one hypothesis posits gradient conflicts [22]. In a cascaded multi-task structure of SE and SS, for layers influenced by both SE and SS, they sometimes have conflicting optimization directions. This means that gradient conflict occurs, and to prevent it, an additional step of gradient modulation (GM) is performed before back-propagation. Concretely, for each layer of such a module where gradient conflict occurs, the projection gradient modulation (PGM) approach treated gradients from the enhancement loss as vectors and adjusted them by projecting onto the orthogonal vector of the separation gradient at that layer [22], and negative gradient modulation (NGM) simply took the negative value of the enhancement gradient vector as the adjusted gradient value [10]. Although GM is effective, it has two drawbacks. Firstly, SEDS does not exhibit the same level of generalization as dependent encoder-decoder scheme [37]. Secondly, GM introduces additional computational expenditure, which increases exponentially with the number of sub-tasks.

Another idea for tackling the over-suppression problem challenges the conventional pipeline with enhancement as the front-end. In the task of separating noisy mixed speech, a separation-priority pipeline (SPP) with separation as the front-end is proposed [10]. This approach ensures the integrity of the input to the separation module. Additionally, without the need for specialized GM, the averaged rate of gradient conflict can also be reduced to below 4%.

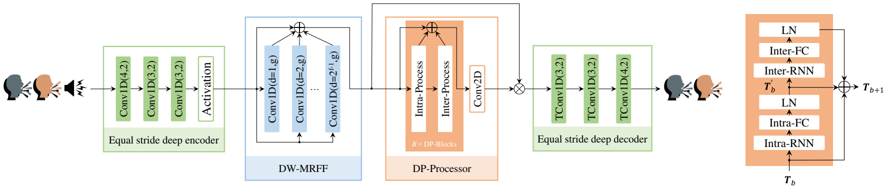
## 3 Proposed Methods

To address the issue of separating mixed speech in the presence of noise and reverberation, we propose two SPP-based cascaded multi-task learning methods: one is SPP with enhancement as the secondary priority (SPP-ES), and the other is SPP with dereverberation as the secondary priority (SPP-DS). Together with the previous EPP, the diagram of these three pipelines are depicted in Figure 2. In each sub-task, the proposed network consists of a deep encoder structure, a depth-wise multi-receptive field fusion (DW-MRFF), a processor, a deep decoder structure. The overview of the proposed network is displayed in Figure 3.


Denoise (SE)
Reverberant noisy mixture to Reverberant clean mixture
Separation (SS)
Reverberant clean mixture to Reverberant clean sources
Dereverberation (DE)
Reverberant clean source to Anechoic clean source
Separation (SS)
Reverberant noisy mixture to Reverberant noisy sources
Denoise (SE)
Reverberant noisy source to Reverberant clean source
Dereverberation (DE)
Reverberant clean source to Anechoic clean source
Separation (SS)
Reverberant noisy mixture to Reverberant noisy sources
Dereverberation (DE)
Reverberant noisy source to Anechoic noisy source
Denoise (SE)
Anechoic noisy source to Anechoic clean source

(a) EPP diagram. (b) SPP-ES diagram. (c) SPP-DS diagram.

Figure 2: The diagrams of EPP, SPP-ES, and SPP-DS are presented in sub-figure (a), (b), and (c) respectively. All SE, SS, and DE modules are colored in blue, aquamarine, and tangerine respectively for distinction. A description of every sub-task is placed on the right side of each pipeline.



(a) SE sub-module of EPP. ⊗ denotes the Hadamard product. (b) DP Block.

Figure 3: The proposed network for implementing each sub-module. Specifically, sub-figure (a) displays the SE sub-module of EPP, where the processor produces only one mask. In contrast, the processor in the SS sub-module generates masks corresponding to the number of speakers. Sub-figure (b) elaborates on the implementation of the DP block.

### 3.1 Consistent stride deep encoder-decoder structure (CS-DEDS)

Inspired by vanilla deep encoder and decoder architecture [26], and to serve the model in processing deeper and finer features, we propose consistent stride deep encoder-decoder structure (CS-DEDS) which also consists of three encoder layers and three decoder layers. Unlike the vanilla approach, which applies a large stride (stride of 8) in the first layer and smaller strides (stride of 1) in the next two layers, we use a consistent stride of 2 across all three layers. This design choice allows for gradual feature extraction and helps prevent excessive information loss after the first encoder layer of the vanilla approach. Additionally, to control the model parameters, the window size of the first layer is reduced from 16 to 4. This design also ensures that the receptive fields of both the deep encoder layers and the single-layer encoder remain consistent, thereby controlling the performance improvements brought by changes in the receptive field. Given the input signal $x \in \mathbb{R}^{1\times T}$, the deep encoded feature $\boldsymbol{H}_e \in \mathbb{R}^{N\times L}$ ($N$ is the feature dimension and $L$ is the number of time steps) is obtained through three 1-dimensional convolutional encoder layers with window sizes and stride sizes of (4,2), (3,2), (3,2), respectively, followed by an exponential linear unit (ELU) activation function [6].

For a masked representation $\boldsymbol{H}_d \in \mathbb{R}^{I\times N\times L}$ which is to be decoded, the estimation of each source $i$ is restored via three 1-dimensional transposed convolutional layers that are mirror-symmetric to the encoder layers in terms of parameters.

### 3.2 Depth-wise multi-receptive field fusion (DW-MRFF)

Inspired by Conv-Tasnet and HiFi-GAN [35, 29], we propose a depth-wise multi-receptive field fusion (DW-MRFF) based on the stacked residual and depth-wise 1-dimensional convolutional layers with a fixed window size and exponentially increasing dilation factors. For each layer $f$ of DW-MRFF with a window size of $W_f$ and dilation factor of $D_f$, features $\boldsymbol{H}_e$ and convolutional kernels are evenly divided into several groups based on depth, and convolution is performed within the corresponding groups:

$$\boldsymbol{H}_e^f = \text{DW-MRFF}_f(\boldsymbol{H}_e, W_f, D_f) \tag{6}$$

where $D_f$ equals $2^{f-1}$.

Finally, we sum up the output of each layer as well as initial input:

$$\boldsymbol{H}_s = \boldsymbol{H}_e + \sum_{f=1}^{F} \boldsymbol{H}_e^f \tag{7}$$

### *3.3 DP-processor*

The goal of DP-processor is to generate masks $\boldsymbol{M} = [\boldsymbol{M}_1, \cdots, \boldsymbol{M}_I] \in \mathbb{R}^{I \times N \times L}$ for each source $i$ from $\boldsymbol{H}_s$, and then $\boldsymbol{H}_d$ is gained by applying each learned feature mask back to $\boldsymbol{H}_s$:

$$\boldsymbol{H}_d[i, :, :] = \boldsymbol{H}_s \otimes \boldsymbol{M}[i, :, :] \tag{8}$$

where $\otimes$ denotes the Hadamard product.

In different tasks, DP-processors are referred to by different names. In separation, enhancement, and dereverberation models, we respectively call them the separator, enhancer, and dereverberator. Apart from the fact that the separator generates masks equal to the number of sources, while the other two only generate a single mask, there is no distinction in modeling. To capture the local and global representation, DP-processor first splits $\boldsymbol{H}_s$ into chunks with length of $K$ and hop size of $P$, generating a 3-D tensor $\boldsymbol{T} \in \mathbb{R}^{N \times K \times S}$, where $S$ is the number of generated chunks. The DP structure will alternately process $\boldsymbol{T}$ through the time dimension (intra-process) and the chunk dimension (inter-process). In this paper, we mainly consider RNN structure and Transformer structure to implement processor.

We use a total number of $B$ stacked DPRNN blocks, each of which consists of an intra- and inter- process sub-block. For each input $\boldsymbol{T}_b$ of $b$-th DPRNN blocks, inter- sub-block consists of a recurrent neural network (RNN) layer, a fully-connected (FC) layer, and a layer normalization, followed by a residual connection:

$$\boldsymbol{T}_b^{'} = \mathrm{LN}(\mathrm{Intra\text{-}FC_b}(\mathrm{Intra\text{-}RNN_b}(\boldsymbol{T}_b[:, :, s]))) + \boldsymbol{T}_b \tag{9}$$

$b$-th inter- sub-block models similarly:

$$\boldsymbol{T}_{b+1} \leftarrow \mathrm{LN}(\mathrm{Inter\text{-}FC_b}(\mathrm{Inter\text{-}RNN_b}(\boldsymbol{T}_b^{'}[:, k, :]))) + \boldsymbol{T}_b^{'} \tag{10}$$

### *3.4 SPP in noisy and reverberant speech separation*

We classify mainstream cascaded multi-task learning methods that employ the SE module as the front-end as EPP [48, 51, 38, 22, 37]. In contrast to their structure, which better aligns with intuition though, the SPP we propose positions the SS module first: one is with enhancement as the secondary priority (SPP-ES), and the other is with dereverberation as the secondary priority (SPP-DS). For SPP-ES, the task of the SE module is to estimate the reverberant clean source from the input containing reverberation and noise, while the task of the DE module is to take the reverberant clean source as input and output the anechoic clean source. For SPP-DS, DE module first predicts anechoic noisy source from reverberant noisy source, followed by the

SE module that estimate final anechoic clean source. It is worth noting that the number of samples doubles after separation. Hence, the subsequent DE and SE modules process samples one after another.

### 3.5 Training approaches

Training from scratch is a straightforward approach. However, the downstream SE and DE modules remain susceptible to the over-suppression issue. To mitigate this, we first pre-train each of the seven core tasks individually (note that the SS task for both SPP-ES and SPP-DS is identical, as is the DE task for EPP and SPP-ES), as introduced in Figure 2. After pre-training, we assemble the required modules according to different pipelines for fine-tuning.

## 4 Experiments

### 4.1 Datasets

We primarily focus on the case where there are two speech sources. Hence, we use Libri2Mix [8], a derived mixture dataset from LibriSpeech [40], and the noise from WHAM! to synthesize noisy mixtures [48]. The training (*train-100*), validation (*dev*), and testing (*test*) subsets of Libri2Mix process 13900, 3000, and 3000 samples, respectively. Samples are mixed based on the length of the shorter utterance. For each sound source, we follow previous recipes [37] of using Pyroomacoustics [46] to simulate anechoic and reverberant versions according to the random configurations exhibited in Table 2. The resulting signal-to-noise ratios (SNRs) are distributed with a mean of $-2.0$ dB and

Table 2: Reverberation sampling distribution configuration. Length (L), width (W), and height (H) specify the size and position of room and receiver, respectively. The source position is determined by H, distance, and angles ($\theta$). T60 denotes the duration, in seconds, for a sound to decrease by 60 decibels (dB). $\mathcal{U}$ represents the continuous uniform distribution.

| | | |
|---|---|---|
| Room | L (m) | $\mathcal{U}(5{,}10)$ |
| | W (m) | $\mathcal{U}(5{,}10)$ |
| | H (m) | $\mathcal{U}(3{,}4)$ |
| T60 | T (s) | $\mathcal{U}(0.2{,}0.6)$ |
| Receiver | L (m) | $\frac{L_{room}}{2}+\mathcal{U}(-0.2{,}0.2)$ |
| | W (m) | $\frac{W_{room}}{2}+\mathcal{U}(-0.2{,}0.2)$ |
| | H (m) | $\mathcal{U}(0.9{,}1.8)$ |
| Sources | H (m) | $\mathcal{U}(0.9{,}1.8)$ |
| | Dist. (m) | $\mathcal{U}(0.66{,}2)$ |
| | $\theta$ | $\mathcal{U}(0{,}2\pi)$ |

Table 3: Configuration of LibriCSS. Subsets *0L* and *0S* imply the 0% overlapping ratio with long and short inter-utterance silence.

| overlapping ratio (%) | *0L* | *0S* | *10* | *20* | *30* | *40* |
|---|---|---|---|---|---|---|
| No. of utterance | 601 | 813 | 815 | 875 | 972 | 941 |

a standard deviation of 7.0 dB. Furthermore, to meticulously evaluate the model's capability, we also utilize other versions of the Libri2Mix test set that does not contain reverberation or background noise, as well as the single-channel utterance-wise evaluation schemes of LibriCSS to conduct generation tests on real-world environment [4]. LibriCSS encompasses six subsets with varying overlapping ratios, and its configuration is shown in Table 3. All the data is sampled at 8 kHz.

## *4.2 Parameters*

### *4.2.1 Model parameters*

All models, including those reproduced from previous studies, use 64-dimensional features. The single encoder Conv1D layer has a kernel size of 16 samples, a stride size of 8 samples, and 64 channels. As for the deep encoder layers, we use $E = 3$ Conv1D layers. The kernel size and the stride size of the first layer are 4 samples and 2 samples, while the second layer and the third layer both have a kernel size of 3 samples and a stride of 2 samples. All three layers have 64 output channels. This design ensures that the resulting features have the same size as those obtained from a single encoder Conv1D layer with a kernel size of 16 samples and a stride of 8 samples, thereby eliminating the impact of feature size inconsistency on the processor. Similarly, the deep decoder layers use $D = 3$ T-Conv1D layers. Their parameters are mirror-symmetric to those of the deep encoder layers. For the MRFF, we use 8 Conv1D layers with a kernel size of 5 samples and a stride of 1 sample. The dilation factors for these layers increase exponentially from 1 to 8.

We use DPRNN to implement the DP-processor. When employing the DP structure, the 2-D tensor is folded to 3-D shape with a window size of $K = 250$ and a hop size of $P = 125$. For each Transformer layer, we use 8 parallel attention heads and a 1024-dimensional FC layer, while each RNN layer in the DPRNN utilizes a 128-dimensional LSTM. Adhering to the conclusion of prior research [22, 10], we employ 6 processor blocks in the SS module, and 2 processor blocks in the SE module. For the DE module, we simply use 1 processor block. This is because the best results are reported in [10] when the layer number of SS module and SE module are in a 3:1 ratio. Meanwhile, we restrict the total number of layers to 9.

#### 4.2.2 Training parameters

We conduct DPRNN experiments using an NVIDIA GeForce RTX 3080 Ti GPU, and SepFormer and reproduction experiments using an NVIDIA GeForce RTX 4070 Ti SUPER GPU. All code is developed based on the open-source toolkit Asteroid [41]. We use Adam as the optimizer [27].

The training epoch is set to 120, and the batch size is 4 for DPRNN and 2 for SepFormer. For training from scratch, we use an initial learning rate of 0.00015, which is adjusted to 0.0001 for fine-tuning. The initial learning rate is halved if no improvement is observed on the validation set for 5 consecutive epochs, and an early stop strategy is triggered after 30 consecutive epochs without improvement. Gradient clipping is set to 5. The best-performing checkpoint on the validation set is used for testing.

### 4.3 Objective functions

#### 4.3.1 Objective function with time-invariant (TI) weights

Objective function of SPP enhancement secondary (ES) is given by:

$$\mathcal{L}_{\text{SPP-ES}} = \alpha_{\text{SS}}\mathcal{L}_{\text{SS}} + \alpha_{\text{SE}}\mathcal{L}_{\text{SE(ES)}} + \alpha_{\text{DE}}\mathcal{L}_{\text{DE(ES)}} \tag{11}$$

where $\alpha_{\text{SS}}$, $\alpha_{\text{SE}}$, and $\alpha_{\text{DE}}$ represent the weight of each sub-task. Given the estimated noisy and reverberant source $\hat{s}_i^{nr}$, estimated reverberant source $\hat{s}_i^{r}$, and anechoic source $s_i^d$, $\mathcal{L}_{\text{SS}}$, $\mathcal{L}_{\text{SE(ES)}}$, and $\mathcal{L}_{\text{DE(ES)}}$ are represented by:

$$\mathcal{L}_{\text{SS}} = -\max_{\pi\in\mathcal{P}} \frac{1}{I}\sum_i \text{SI-SNR}(\hat{s}^{nr}_{\pi(i)}, s_i^{nr}) \tag{12}$$

$$\mathcal{L}_{\text{SE(ES)}} = -\frac{1}{I}\sum_i \text{SI-SNR}(\hat{s}^{r}_{\pi(i)}, s_i^{r}) \tag{13}$$

$$\mathcal{L}_{\text{DE(ES)}} = -\frac{1}{I}\sum_i \text{SI-SNR}(\hat{s}^{d}_{\pi(i)}, s_i^{d}) \tag{14}$$

In SS task, we employ permutation-invariant training (PIT) to calculate the best permutation mapping set $\pi$ among all possible mapping set $\mathcal{P}$. $\pi$ assists the model in establishing the correspondence between estimated and ground truth speeches [28]. Once $\pi$ is determined in training SS module as shown in Equation 12, SE and DE module in Equations 13 and 14 will adopt the same one. $s_i^{nr}$ is gained by the temporal addition of reverberant image $s_i^r$ and ambient noise $n$.

SI-SNR (scale-invariant signal-to-noise ratio) in Equations 12, 13, and 14 are similarity metrics of two time-domain signals:

$$\text{SI-SNR}(\hat{s}, s) = 10 \log_{10} \frac{\parallel s_{\text{proj}} \parallel^2}{\parallel s_{\text{noise}} \parallel^2} \tag{15}$$

$$s_{\text{proj}} = \frac{\langle \hat{s}, s \rangle s}{\parallel s \parallel^2} \tag{16}$$

$$s_{\text{noise}} = \hat{s} - s_{\text{proj}} \tag{17}$$

where $s_{\text{proj}}$ denotes the projection of $\hat{s}$ on $s$.

In the case of SPP-DS, objective function of dereverberation module $\mathcal{L}_{\text{DE(DS)}}$ still keeps the noise, and noise is removed in the final SE module. Entire objective function is altered to:

$$\mathcal{L}_{\text{SS}} = -\max_{\pi \in \mathcal{P}} \frac{1}{I} \sum_i \text{SI-SNR}(\hat{s}^{nr}_{\pi(i)}, s^{nr}_i) \tag{18}$$

$$\mathcal{L}_{\text{DE(DS)}} = -\frac{1}{I} \sum_i \text{SI-SNR}(\hat{s}^{nd}_{\pi(i)}, s^{nd}_i) \tag{19}$$

$$\mathcal{L}_{\text{SE(DS)}} = -\frac{1}{I} \sum_i \text{SI-SNR}(\hat{s}^{d}_{\pi(i)}, s^{d}_i) \tag{20}$$

where $s^{nd}_i$ is the result of adding anechoic source $s^d_i$ and ambient noise $n$ likewise.

Since each sub-task is assumed to be equally important during training, time-invariant (TI) fine-tuning method means that the objective function weights for sub-tasks are evenly distributed, namely $\alpha_{\text{SS}} = \alpha_{\text{SE}} = \alpha_{\text{DE}} = \frac{1}{3}$.

### *4.3.2 Time-varying (TV) weights objective function for fine-tuning*

Considering that the final output represents the overall model performance, inspired by [12], an alternative time-varying (TV) weights objective function for fine-tuning is proposed for SPP-DS by:

$$\alpha_{\text{SS}}(e) = \alpha_{\text{DE}}(e) = \begin{cases} \frac{1}{3}, & e < 40 \\ \frac{-7e}{2400} + \frac{9}{20}, & e \geq 40 \end{cases} \tag{21}$$

$$\alpha_{\text{SE}}(e) = \begin{cases} \frac{1}{3}, & e < 40 \\ \frac{14e}{2400} + \frac{1}{10}, & e \geq 40 \end{cases} \tag{22}$$

where $e$ is current epoch number. Before the 40th epoch, the weights for the three tasks are all 1/3. By epoch 120, the weights linearly transition to the ratio of 1:1:8.

### 4.4 Evaluation metrics

We first evaluate the model's performance from the category of signal-to-noise ratio, including SI-SNR [30], SI-SNR improvement (SI-SNRi), SDRi (signal-to-distortion ratio improvement), SIRi (signal-to-interference ratio improvement) [53]. An estimated signal $\hat{s}$ of target source $s$ can be assumed to be composed of four components:

$$\hat{s} = s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}} \tag{23}$$

where $s_{\text{target}}$ calculates the projection of target signal on estimated signal, $e_{\text{interf}}$ and $e_{\text{noise}}$ represent the error projection vector of undesired signals and noise on estimated signal respectively. $e_{\text{artif}}$ is the rest error term.

SDR and SIR are thus given by:

$$\text{SDR}(\hat{s}, s) = 10 \log_{10} \frac{\| \, s_{\text{target}} \, \|^2}{\| \, e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}} \, \|^2} \tag{24}$$

and

$$\text{SIR}(\hat{s}, s) = 10 \log_{10} \frac{\| \, s_{\text{target}} \, \|^2}{\| \, e_{\text{interf}} \, \|^2} \tag{25}$$

SDRi and SIRi are used to remove the influence factors of mixed speech:

$$\text{SDRi}(\hat{s}, s) = \text{SDR}(\hat{s}, s) - \text{SDR}(\hat{s}, m) \tag{26}$$

$$\text{SIRi}(\hat{s}, s) = \text{SIR}(\hat{s}, s) - \text{SIR}(\hat{s}, m) \tag{27}$$

In particular, SIRi also directly serves for separation ability.

The second category is perceptual metrics, including perceptual evaluation of speech quality (PESQ) [44] and short-time objective intelligibility (STOI) [50]. The third category is enhancement metrics, including predicted rating of speech distortion (CSIG), predicted rating of background distortion (CBAK), and predicted rating of overall quality (COVL) [21]. For real-world dataset LibriCSS where transcriptions are available instead of clean reference signals, word error rate (WER) is used. As an extra step of ASR, we employ an SSL-based ASR model to recognize estimated signals [1]. In addition to the above evaluation metrics, we report the model size and computational load for 3-second data, measured in Giga Multiply-Accumulate operations (GMACs). The tools we use are publicly available at the public site[1]

[1] https://github.com/sovrasov/flops-counter.pytorch

The mel cepstrum distortion (MCD) measures the similarity of two sequences of mel cepstra based on the mel cepstrum coefficient (MCEP) [18], which is given by:

$$\mathrm{MCD}(\hat{s}, s) = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{D}(\hat{c}_d - c_d)^2} \tag{28}$$

where $\hat{c}_d$ and $c_d$ are $d$-th dimension of estimated MCEP and target MCEP, respectively. $D$ denotes the dimension of MCEP. In the visualization section, the MCD metric is employed to measure the distortion of exhibited samples.

## 5 Results

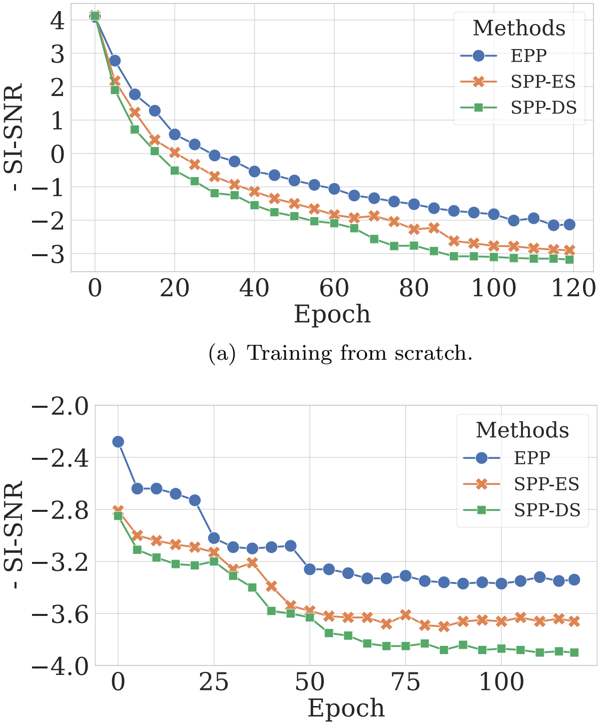
### *5.1 Comparative results with previous methods*

Table 4 compares the SI-SNRi performance and computational complexity of the proposed methods against reproduced results from previous separation models. U-Mamba-Net [11], a much more compact model designed for severely degraded mixture, achieves a competitive SI-SNRi of 8.50 dB. SepFormer and its resource-efficient variant [14] yield 9.04 dB and 3.95 dB, respectively. The DPRNN-based encoder-decoder attractor structure (EDA) [45] also achieves 8.50 dB. MossFormer [65] employs a joint local and global self-attention

Table 4: The results of existing and proposed methods.

| Methods | Fine-tuning? | SI-SNRi (dB) | # Param. (M) | GMACs |
|---|---|---|---|---|
| TasNet | - | 5.70 | 23.2 | 27.8 |
| SuDoRM-RF | - | 2.90 | 2.6 | 3.6 |
| SuDoRM-RF+ | - | 5.33 | 2.7 | 3.0 |
| Conv-TasNet | - | 6.88 | 6.3 | 18.7 |
| DPRNN (6 layers) | - | 7.59 | 3.7 | 23.9 |
| DPRNN (9 layers) | - | 7.15 | 5.4 | 35.6 |
| DPRNN (EPP) | - | 8.08 | 5.6 | 40.2 |
| SepFormer | - | 9.04 | 25.7 | 196.8 |
| ReSepFormer | - | 3.95 | 4.8 | 8.3 |
| EDA | - | 8.99 | 7.5 | - |
| U-mamba-net | - | 8.50 | 4.4 | 2.5 |
| MossFormer | - | 8.11 | 1.1 | - |
| MossFormer2 | - | **10.75** | 9.2 | - |
| Proposal (SPP-ES) | ✗ | 8.77 | 6.2 | 52.7 |
| Proposal (SPP-DS) | ✗ | 8.95 | 6.2 | 52.7 |
| Proposal (SPP-ES) | ✓ | 9.48 | 6.2 | 52.7 |
| Proposal (SPP-DS) | ✓ | 9.71 | 6.2 | 52.7 |

architecture, achieving 8.11 dB in SI-SNR. Its variant, MossFormer2 [66], incorporates an additional feedforward sequential memory network and, despite having significantly more parameters, delivers the highest performance among existing models with 10.75 dB, surpassing our proposed method by approximately 1 dB. The proposed SPP-DS model attains a comparable SI-SNRi score, trailing SepFormer by only 0.09 dB, while requiring merely one-fourth of its GMACs. Moreover, with transfer learning, SPP-DS achieves an additional improvement of 0.77 dB SI-SNRi without increasing model size compared to training from scratch, and surpasses SepFormer by 0.68 dB. Both SPP-based models also substantially outperform the EPP-based model. Furthermore, regardless of whether transfer learning is applied, SPP-DS consistently maintains an average advantage of 0.2 dB SI-SNRi over SPP-ES. Figure 4 presents the learning curves for both training from scratch and transfer learning fine-tuning, highlighting the practical effectiveness of the proposed approaches.



(a) Training from scratch.

(b) Training using transfer learning.

Figure 4: Comparison of learning curve of training from scratch (a) and fine-tuning phase of using transfer learning (b) on validation dataset.

### *5.2 Detailed comparison of EPP and SPP*

Table 5 provides a detailed comparison between EPP and the proposed SPP-ES and SPP-DS. In experiments without transfer learning, we observe that model performance on specific metrics is influenced by the position of the module associated with each metric. For instance, in EPP, where the SE module is positioned first, optimal results are achieved in CBAK. By prioritizing the SS module, both SPP-ES and SPP-DS deliver exceptional performance in SIRi. Additionally, SPP outperforms in metrics related to signal-to-noise ratio and perceptual quality, which are widely accepted as overall metrics. This indicates that SPP can generate superior overall results.

When we focus on transfer learning methods, we first observe that the initial loss points of SPP that uses TI are lower than that of EPP in Figure 4(b). This suggests that SPP is more rational than EPP. Additionally, compared to training from scratch, transfer learning significantly enhances performance across all three pipelines. This indicates that, through pre-training the downstream model, sub-modules in the latter positions of the pipeline are expected to be better learned compared to joint learning from scratch and over-suppression problem is thus further alleviated. Specifically, the improvements in EPP exceed those observed in SPP, suggesting that alleviating over-suppression in the SS module within EPP contributes to a more substantial performance boost compared to the SE and DE modules in SPP. The narrowing gap of learning curves of EPP and SPP in Figure 4(b) also supports this statement. These findings align with the core argument of this paper, asserting that the SS module plays a more critical role than SE and DE modules. In the last 80 epochs, TV shifts its focus to the later stages, with test set results indicating that this approach is effective, though the improvement is modest. However, when the loss weight of earlier modules is gradually reduced to zero in transfer learning, overall model performance declines sharply. Thus, we conclude that multi-task learning remains essential in transfer learning. Finally, through transfer learning, SPP-DS achieves the best results across all metrics, indicating that this pipeline possesses the strongest separation capability. The best

Table 5: Elaborated results for different pipelines.

| Pipeline | Fine-tuning? | SI-SNRi | SDRi | SIRi | STOI | PESQ | CBAK | COVL |
|---|---|---|---|---|---|---|---|---|
| EPP | ✗ | 8.08 | 8.62 | 16.39 | 73.19 | 1.75 | **2.16** | 2.18 |
| SPP-ES | ✗ | 8.77 | 8.91 | 17.83 | 74.76 | 1.82 | 2.00 | **2.27** |
| SPP-DS | ✗ | **8.95** | **9.10** | **17.97** | **75.21** | **1.82** | 2.00 | 2.26 |
| EPP | TI | 9.23 | 9.42 | 18.70 | 76.01 | 1.86 | 2.03 | 2.30 |
| SPP-ES | TI | 9.48 | 9.64 | 19.05 | 76.67 | 1.90 | 2.07 | 2.35 |
| SPP-DS | TI | 9.71 | 9.65 | 19.51 | 77.14 | 1.92 | 2.10 | 2.38 |
| SPP-DS | TV | **9.72** | **9.66** | **19.62** | **77.23** | **1.93** | **2.11** | **2.39** |

Table 6: The best SI-SNR performance of each sub-task on its corresponding validation dataset. These pre-trained sub-modules are to be assembled for fine-tuning each pipeline.

| Pipeline | SS | SE | DE |
|---|---|---|---|
| EPP | 8.89 | 11.91 | 7.04 |
| SPP-ES | 8.27 | 11.34 | 7.04 |
| SPP-DS | 8.27 | 11.87 | 7.76 |

Table 7: Ablation studies using SPP-DS pipeline. ✗ means that module is not used at all. ✓ in column DW-MRFF means the $G = 1$.

| CS-DEDS | DW-MRFF | SI-SNR | SI-SNRi | SIRi | STOI | PESQ | # Param. | GMACs |
|---|---|---|---|---|---|---|---|---|
| Vanilla | ✓ | 2.60 | 8.43 | 17.00 | 72.68 | 1.77 | 6.2 M | 51.7 |
| ✓ | ✓ | **3.12** | **8.95** | **17.97** | **75.21** | 1.82 | 6.2 M | 52.7 |
| ✓ | ✗ | 2.92 | 8.75 | 17.82 | 74.87 | 1.81 | 5.7 M | 50.2 |
| ✗ | ✓ | 3.08 | 8.91 | 17.96 | 75.17 | **1.83** | 6.1 M | 50.1 |
| ✗ | ✗ | 2.58 | 8.41 | 17.33 | 73.24 | 1.75 | 5.6 M | 48.3 |
| ✓ | $G = 2$ | 2.91 | 8.74 | 17.74 | 74.91 | 1.82 | 6.0 M | 51.5 |
| ✓ | $G = 4$ | 3.12 | 8.95 | 18.10 | 75.33 | 1.83 | 5.8 M | 50.8 |
| ✓ | $G = 8$ | **3.20** | **9.03** | **18.88** | **75.33** | **1.85** | 5.8 M | 50.5 |
| ✓ | $G = 16$ | 3.09 | 8.92 | 18.36 | 74.93 | 1.83 | 5.8 M | 50.4 |

SI-SNR performance of each pre-trained sub-module on the corresponding validation set is shown in Table 6.

### *5.3 Results of ablation studies*

To access the effectiveness of proposed CS-DEDS and DW-MRFF, we conduct ablation studies based on the SPP-DS pipeline. The results are displayed in Table 7. In trials where CS-DEDS is not adopted, a single Conv1D layer with a window size of 16 samples and a stride size of 8 samples is used. Vanilla means that we use the deep encoder and decoder architecture from paper [26]. $G$ in DW-MRFF represents the number of groups into which features are divided. Firstly, the top half of the table illustrates the individual contributions of CS-DEDS and DW-MRFF to the model. Both the CS-DEDS and MRFF modules positively contribute to the model's separation capability. Meanwhile, CS-DEDS achieves an improvement of 0.52 dB SI-SNRi over vanilla DEDS. The lower half of the table examines depthwise convolution controlled by the grouping variable $G$. When the parameter is set to 8, the model achieves optimal performance across all metrics, and both the model's parameters and computational load are reduced compared to when $G = 1$.

### *5.4 Replacing DPRNN block with SepFormer block*

In this section, to examine the generalizability of different pipelines, we replace the DPRNN block in the proposed model with the self-attention block of SepFormer. For an efficient training process and a fair comparison, we compress the size of the SepFormer block to approximately match that of the DPRNN block. Their results are displayed in Table 8. The results, whether using transfer learning or not, demonstrate that the SPP pipeline outperforms the EPP pipeline. Additionally, applying TV in the fine-tuning phase can further enhance the effectiveness of transfer learning.

### *5.5 Merging SE and DE modules in one network*

This section discusses an intriguing idea. The previous results have clearly demonstrated the superiority of SPP. However, regardless of the pipeline, the module in the final position tends to suffer from severe over-suppression problem. For instance, the SE module in SPP-DS. To alleviate this, this section considers merging the SE and DE modules in the SPP, though this approach may be viewed as a degeneration of multi-task learning. When using TV fine-tuning, the weights of the first and second modules are gradually adjusted linearly to achieve a 2:8 ratio over the final 80 epochs. The overall results and comparative results with not merging SE and DE modules are shown in Table 9. Firstly, when training is conducted from scratch, the results after merging outperform those obtained without merging when adopting SPP. We believe this indicates that the overall degree of over-suppression is reduced after merging. Pre-training helps the structure without merging SE and DE better mitigate the over-suppression issue. However, during the fine-tuning phase, TV that shifts focus toward later stages appears to give the merging approach a slight advantage. Observing the training curves in Figure 5, the starting point and overall convergence trend of the merged approach are both below those of the non-merged approach. Additionally, merging offers the

Table 8: Results of using SepFormer block for each module.

| Pipeline | Fine-tuning? | SI-SNRi | SDRi | SIRi | STOI | PESQ | # Param. | GMACs |
|---|---|---|---|---|---|---|---|---|
| EPP | ✗ | 5.02 | 5.53 | 11.38 | 64.47 | 1.52 | 5.5 M | 50.1 |
| SPP-ES | ✗ | 6.25 | 6.45 | **12.53** | 68.14 | 1.61 | 6.2 M | 64.4 |
| SPP-DS | ✗ | **6.40** | **6.77** | 12.18 | **68.95** | **1.61** | 6.2 M | 64.4 |
| EPP | TI | 7.87 | 8.30 | 16.45 | 72.44 | 1.72 | 5.5 M | 50.1 |
| SPP-ES | TI | 7.98 | 8.38 | 16.54 | 72.49 | 1.73 | 6.2 M | 64.4 |
| SPP-DS | TI | 8.37 | 8.40 | 16.99 | 73.12 | 1.75 | 6.2 M | 64.4 |
| SPP-DS | TV | **8.59** | **8.53** | **17.20** | **73.60** | **1.78** | 6.2 M | 64.4 |

Table 9: The performance of structure of merging SE and DE modules. SPP with "Not Merge" column denotes SPP-DS.

| Pipeline | Fine-tuning? | Merge | | Not Merge | |
|---|---|---|---|---|---|
| | | SI-SNRi | GMACs | SI-SNRi | GMACs |
| EPP | ✗ | 7.93 | 36.9 | 8.08 | 40.2 |
| SPP | ✗ | 9.22 | 49.2 | 8.95 | 52.7 |
| EPP | TI | 9.10 | 36.9 | 9.23 | 40.2 |
| SPP | TI | 9.64 | 49.2 | 9.71 | 52.7 |
| SPP | TV | **9.75** | 49.2 | 9.72 | 52.7 |

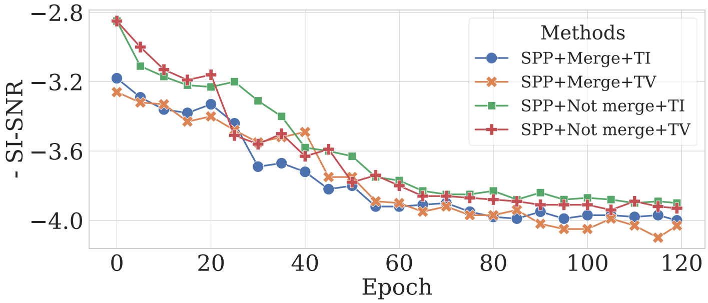


Figure 5: Comparison of learning curve between merging structure and fine-tuning manner.

Table 10: The best SI-SNR performance of each sub-task on its corresponding validation dataset when merging SE and DE modules.

| Pipeline | SS | SE & DE |
|---|---|---|
| EPP | 14.90 | 4.41 |
| SPP | 8.18 | 5.83 |

benefit of reduced computational load. Validation SI-SNR performance during pre-train phase when using transfer learning is displayed in Table 10.

Figure 6 provides a comparison between SPP-ES, SPP (Merge), and SepFormer in terms of inference time (in milliseconds), computational load (in GMACs), and GPU memory usage during training (in GB). SPP demonstrates an advantage over SepFormer in inference time when it exceeds two seconds. When calculating memory usage, the maximum GPU memory consumption is recorded with the model's batch size set to 1.

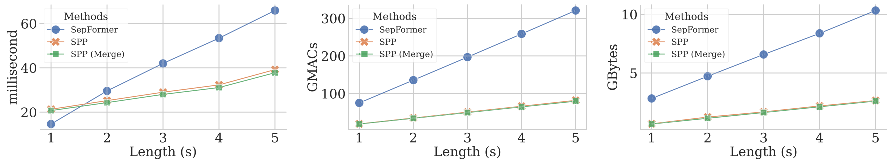


Figure 6: Forward-pass speed (left), GMACs (middle), and GPU memory usage (right) on 1-5 seconds input signals.

## 5.6 Evaluation of separation and enhancement ability

We demonstrate another advantage of cascaded multi-task learning in this section. Since all sub-modules operate directly on waveforms, theoretically, we can selectively choose the required modules for testing based on the specific scenario. Our primary focus is on two scenarios: separating mixed speech without noise or reverberation, separating speech with noise but without reverberation. Lastly, we check the conversation-like case where signals are sparsely overlapped in real-world and indoor environments.

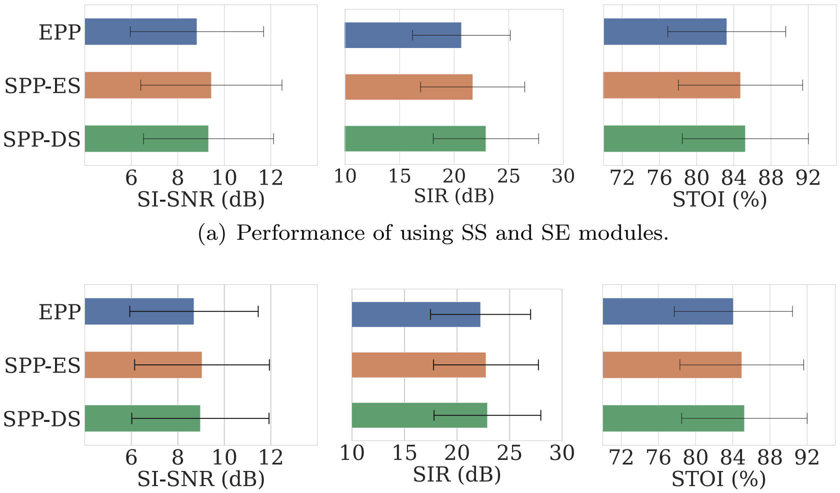
### 5.6.1 Evaluation of separation and enhancement ability

We use a noisy anechoic version of the mixed test set to evaluate the three fine-tuned pipelines. As all pipelines are constructed on cascaded multi-task structures, we conduct two types of inferences. First, we utilize only the SS and SE modules to process the noisy mixture, excluding the DE module. Second, we employ the entire pipeline to obtain the estimated signals, consistent with the evaluations conducted in previous sections. Their performance is presented in Figure 7.

From the displayed results, we can draw two conclusions. First, regardless of whether only the SS and SE modules are used or the entire pipeline is employed, SPP consistently outperforms EPP. Second, the inclusion of the additional DE module does not improve the overall signal-to-noise ratio performance. However, it positively impacts separation and perceptual metrics. This is possibly because the DE modules in EPP and SPP-ES, which are positioned at the end, are trained with ground truth targets, thus serving a corrective function.

### 5.6.2 Evaluation of separation ability

This subsection shows the performance of all transfer-learned pipelines on noise-free and aechoic mixtures. Similarly, besides using SS and SE modules, the performance of solely using SS module is compared in Figure 8.



(a) Performance of using SS and SE modules.

(b) Performance of using SS, SE, and DE modules.

Figure 7: Comparison of three pipelines on noisy aechoic situation. The processing order aligns with the pipeline principle. The error bars represent the standard deviation.

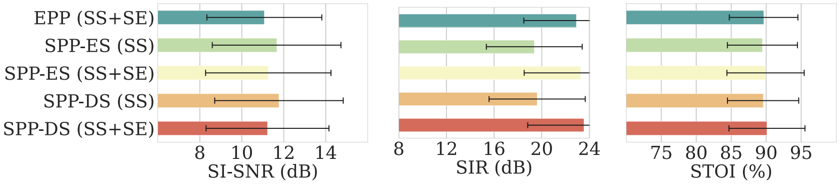


Figure 8: Comparison of three pipelines on clean aechoic situation. The modules in parentheses are manually selected.

The separation outcomes achieved solely from the SS module of EPP are notably poor, with the separated speech signals containing excessive white noise. We speculate that this issue may stem from over-suppression problems during training, leading to degraded input quality. Consequently, when normal speech is fed into the model, it may output a distinct line at a specific frequency, as illustrated by an example in Figure 9. Thus, we have refrained from reporting their performance. Also, this inversely suggests that SPP is more flexible than EPP, capable of handling a broader range of scenarios as required. The remaining results further underscore the effectiveness of SPP. Apart from reaffirming the conclusion drawn in the previous subsection that adding extra modules assist performance on separation and perceptual aspects, it negatively affects signal-to-noise ratio performance.

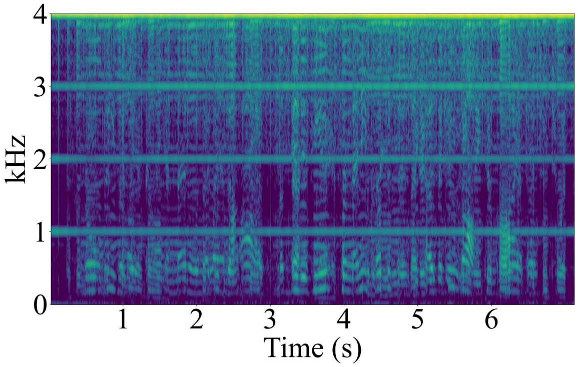


Figure 9: An erroneous instance demonstrating the use of the sole SS module of transfer-learned EPP for separating a noise-free mixture.

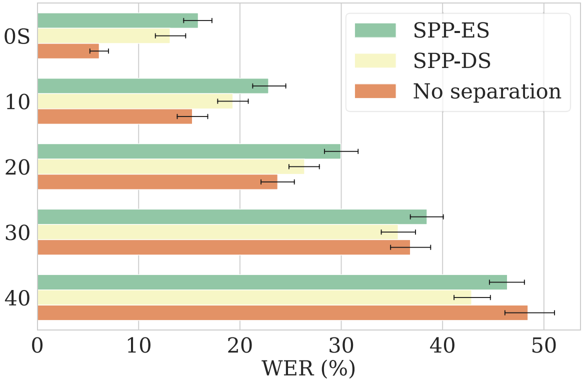


Figure 10: WER (%) performance on LibriCSS. The error bars represent the 95% confidence interval.

### *5.6.3 Evaluation on LibriCSS*

LibriCSS is a dataset created by individuals taking turns reading Librispeech in an indoor environment. Therefore, testing on LibriCSS can be considered as speech separation in a reverberant environment with negligible noise. We process the data using SS and DE modules of fine-tuned models, and then perform recognition using a pre-trained ASR model. The WER performance of each subset is shown in Figure 10. Due to the same reasons as in section 5.6.2, the results of EPP are very poor, so only the results of the SPP-ES and SPP-DS pipelines are compared. First, SPP can perform generalization tests on real data, whereas EPP cannot. Secondly, the results of SPP-DS are better than those of SPP-ES. An intuitive reason for this is that the dereverberation module in SPP-DS is stronger than that in SPP-ES, as it is positioned earlier in the pipeline. Furthermore, we use the same pre-trained ASR model to calculate the WER the mixture signal, and compare it with SPP in Figure 10. Consistent with the conclusions of previous work [4], for data with lower overlap ratios, the separation model often performs worse than when no processing is applied. The threshold for this effect is around 30%, beyond which SPP starts to demonstrate its advantages.

### *5.7 Evaluation of model's performance with reverberation factors*

We conduct an analysis of the model's performance concerning two reverberation factors: T60 and room volume during the reverberation process. T60 represents the time required for sound to decay by 60 decibels, while room volume is determined by multiplying its length, width, and height. In Figure 11, we present their scatter matrix alongside four selected metrics. We observe strong positive correlations among the four metrics, but no clear correlation between each metric and T60 or room volume. This suggests that the proposed model can generally handle reverberations caused by varying room volumes and materials.

Additionally, we report the Pearson coefficient correlations (PCC) between the four metrics and T60, as well as room volume, in Table 11. The model's performance exhibits a weak positive correlation with room volume, indicating

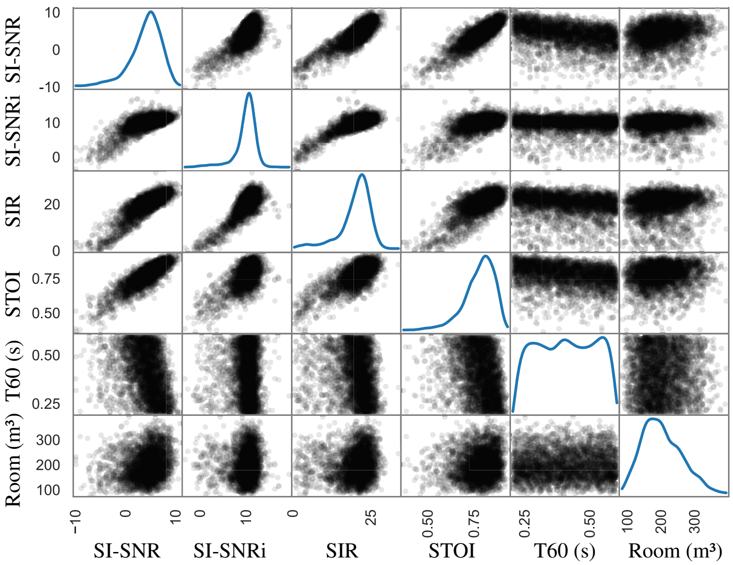


Figure 11: Scatter matrix of four metrics and T60, as well as room volume.

Table 11: The Pearson coefficient correlations (PCC) of four metrics with T60 and room volume.

| | SI-SNR | SI-SNRi | SIR | STOI |
|---|---|---|---|---|
| T60 | -0.327 | -0.024 | -0.150 | -0.249 |
| Room volume | 0.216 | 0.034 | 0.010 | 0.118 |

that larger room volumes lead to weaker reverberations and better results. Conversely, there is a weak negative correlation between performance and T60, suggesting that longer reverberation decay times result in stronger reverberation and poorer results. These findings are consistent with subjective understanding.

### *5.8 Visualization*

We present spectrograms of different model outputs using fine-tuning approach in Figure 12. The first row depicts spectrograms of noisy reverberant mixed speech, while the second row represents spectrograms of ground truth speeches. The third to fifth rows respectively show the spectrograms of estimated signals for EPP, SPP-ES, and SPP-DS. As indicated in the boxes, SPP has two main advantages. Firstly, compared to EPP, SPP exhibits fewer incorrect separation portions, as depicted by the white and yellow boxes. Secondly, the fundamental frequency and harmonics in the red and tangerine boxes of separated speech by SPP are more clearly restored compared to EPP.

Furthermore, we report the MCD score of each separated signal in Table 12. We use MCD to describe the consistency of the Mel cepstrum coefficients between the estimated signal and the target signal within the frequency range of 10 Hz to 800 Hz. The smaller MCD scores of SPP-DS indicate that SPP-DS generates signals of the highest similarity.

### *5.9 Discussion*

In the experiments designed to validate the proposed methods, we first implemented the baseline EPP pipeline and the proposed SPP. These experiments consistently demonstrate the superiority of the SPP pipeline. Subsequent ablation experiments further validate the effectiveness of each newly added component. Pre-training and fine-tuning approaches have also been proven effective in alleviating the over-suppression problem. Although we have demonstrated that sequentially processing modules according to their importance in a cascaded structure, and gradually increasing the weights of the posterior modules during transfer learning, is advantageous for maximizing the alleviation of over-suppression problem and improving the overall quality of speech,

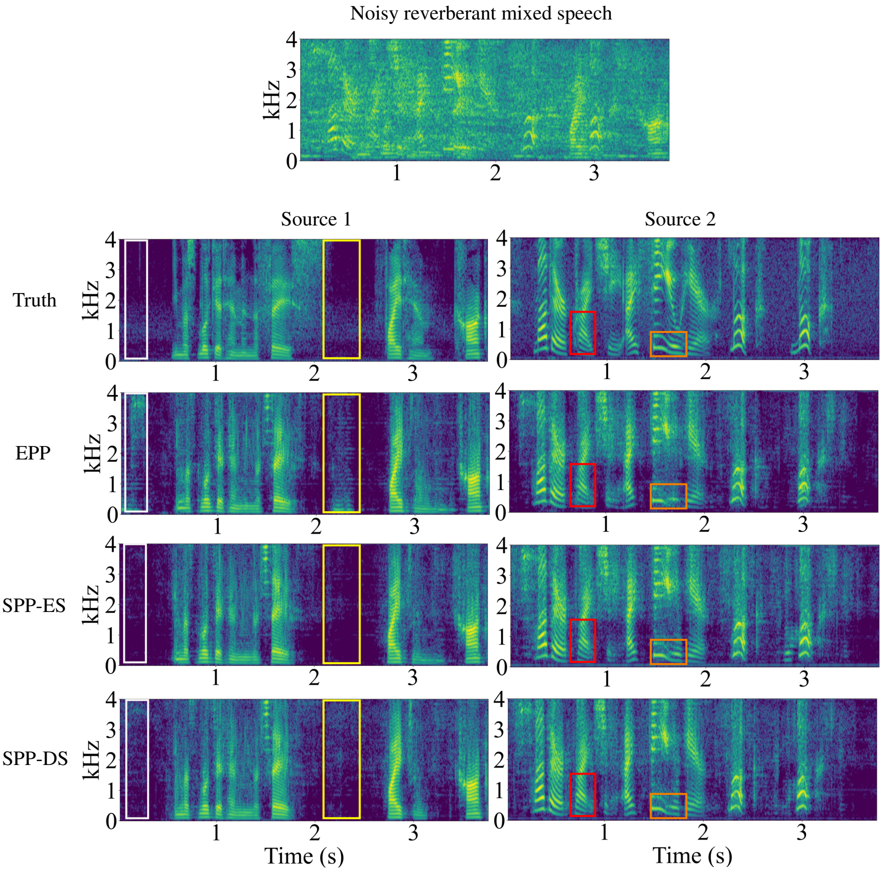


Figure 12: Visualized spectrograms of the outputs of each pipeline use fine-tuning approach.

Table 12: MCD performance of instances in Figure 12.

| Pipeline | Source 1 | Source 2 | Average |
|---|---|---|---|
| EPP | 8.32 | 7.30 | 7.81 |
| SPP-ES | 8.20 | 7.35 | 7.78 |
| SPP-DS | **7.47** | **6.81** | **7.14** |

such efforts merely shift the over-suppression problem to less critical modules. Thus, the over-suppression issue persists. The cascaded structure itself remains the root cause of the over-suppression problem. Therefore, a promising avenue for future research is parallel multi-task learning, followed by the utilization of more powerful feature fusion techniques, such as cross-attention-based feature fusion, to estimate sources. Additionally, the SPP pipeline holds great promise for integration with multimodal approaches, such as incorporating lip movement data to aid in speech separation in complex environments [61].

## 6 Conclusion

In this paper, we introduced an efficient solution to noisy and reverberant speech separation through a separation priority pipeline-based cascaded multi-task learning framework, which challenges the prevailing architectures that typically place the SE module as the front-end. We demonstrated that the EPP pipeline is the root cause of the over-suppression problem affecting the SS module. Within the scope of our current cascaded multi-task learning approach for noisy reverberant speech separation, which encompasses SS, SE, and DE modules, we proposed two variants of the SPP pipeline: one following the SS-SE-DE order and another with SS-DE-SE. These proposed SPP configurations effectively mitigate the over-suppression problem by sequentially handling modules according to their significance, thereby shifting the over-suppression problem to the less critical SE and DE modules. In each sub-module, we introduced the CS-DEDS and the DW-MRFF, built upon the traditional encoder-processor-decoder architecture. Through ablation experiments, we proved that each module contributes positively to enhancing model performance. To further alleviate the over-suppression problem, we implemented pre-training and time-invariant and time-varying fine-tuning approaches on the proposed pipelines. Gradually increasing the weights of modules positioned towards the end of the pipeline resulted in further improvements.

## References

[1] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations”, *Advances in neural information processing systems*, 33, 2020, 12449–60.

[2] J. Chen, Q. Mao, and D. Liu, “Dual-Path Transformer Network: Direct Context-Aware Modeling for End-to-End Monaural Speech Separation”, in *Annual Conference of the International Speech Communication Association*, ed. H. Meng, B. Xu, and T. F. Zheng, ISCA, 2020, 2642–6.

[3] Z. Chen, Y. Luo, and N. Mesgarani, “Deep attractor network for single-microphone speaker separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, 246–50.

[4] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, "Continuous speech separation: Dataset and analysis", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, 7284–8.

[5] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears", *The Journal of the acoustical society of America*, 25(5), 1953, 975–9.

[6] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)", in *International Conference on Learning Representations*, 2016.

[7] T. Cord-Landwehr, C. Boeddeker, T. Von Neumann, C. Zorilă, R. Doddipatla, and R. Haeb-Umbach, "Monaural source separation: From anechoic to reverberant environments", in *International Workshop on Acoustic Signal Enhancement (IWAENC)*, IEEE, 2022, 1–5.

[8] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "LibriMix: An open-source dataset for generalizable speech separation", *arXiv preprint arXiv:2005.11262*, 2020.

[9] S. Dang, T. Matsumoto, Y. Takeuchi, and H. Kudo, "A comprehensive study on supervised single-channel noisy speech separation with multi-task learning", *Speech Communication*, 167, 2025, 103162.

[10] S. Dang, T. Matsumoto, Y. Takeuchi, and H. Kudo, "A Separation Priority Pipeline for Single-Channel Speech Separation in Noisy Environments", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2024, 12511–5.

[11] S. Dang, T. Matsumoto, Y. Takeuchi, and H. Kudo, "U-Mamba-Net: A highly efficient Mamba-based U-net style network for noisy and reverberant speech separation", in *2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2024, 1–5.

[12] S. Dang, T. Matsumoto, Y. Takeuchi, and H. Kudo, "Using Semi-supervised Learning for Monaural Time-domain Speech Separation with a Self-supervised Learning-based SI-SNR Estimator", in *Annual Conference of the International Speech Communication Association*, ISCA, 2023, 3759–63.

[13] N. Das, S. Chakraborty, J. Chaki, N. Padhy, and N. Dey, "Fundamentals, present and future perspectives of speech enhancement", *International Journal of Speech Technology*, 24, 2021, 883–901.

[14] L. Della Libera, C. Subakan, M. Ravanelli, S. Cornell, F. Lepoutre, and F. Grondin, "Resource-efficient separation transformer", in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2024, 761–5.

[15] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2015, 708–12.

[16] C. Fan, B. Liu, J. Tao, J. Yi, and Z. Wen, "Discriminative Learning for Monaural Speech Separation Using Deep Embedding Features", in *Annual Conference of the International Speech Communication Association*, ISCA, 2019, 4599–603.

[17] Y. Fu, Y. Liu, J. Li, D. Luo, S. Lv, Y. Jv, and L. Xie, "Uformer: A unet based dilated complex & real dual-path conformer network for simultaneous speech enhancement and dereverberation", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, 7417–21.

[18] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "Espnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, 7654–8.

[19] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, 31–5.

[20] 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 *Annual Conference of the International Speech Communication Association*, ISCA, 2020, 2472–6.

[21] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement", *IEEE Transactions on audio, speech, and language processing*, 16(1), 2007, 229–38.

[22] Y. Hu, C. Chen, H. Zou, X. Zhong, and E. S. Chng, "Unifying speech enhancement and separation with gradient modulation for end-to-end noise-robust speech separation", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, 1–5.

[23] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, 1562–6.

[24] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(12), 2015, 2136–47.

[25] L. Hui, M. Cai, C. Guo, L. He, W.-Q. Zhang, and J. Liu, "Convolutional maxout neural networks for speech separation", in *2015 IEEE*

*international symposium on signal processing and information technology (ISSPIT)*, IEEE, 2015, 24–7.

[26] B. Kadıoğlu, M. Horgan, X. Liu, J. Pons, D. Darcy, and V. Kumar, “An empirical study of Conv-TasNet”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, 7264–8.

[27] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization”, in *3rd International Conference on Learning Representations ICLR*, 2015.

[28] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, “Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(10), 2017, 1901–13.

[29] J. Kong, J. Kim, and J. Bae, “HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis”, *Advances in neural information processing systems*, 33, 2020, 17022–33.

[30] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “SDR–half-baked or well done?”, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, 626–30.

[31] C. Li, Y. Luo, C. Han, J. Li, T. Yoshioka, T. Zhou, M. Delcroix, K. Kinoshita, C. Boeddeker, Y. Qian, *et al.*, “Dual-path RNN for long recording speech separation”, in *IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2021, 865–72.

[32] Y. Luo, *End-to-end speech separation with neural networks*, Columbia University, 2021.

[33] Y. Luo, Z. Chen, and T. Yoshioka, “Dual-Path RNN: Efficient long sequence modeling for time-domain single-channel speech separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, 46–50.

[34] Y. Luo, C. Han, and N. Mesgarani, “Distortion-controlled training for end-to-end reverberant speech separation with auxiliary autoencoding loss”, in *Spoken Language Technology Workshop (SLT)*, IEEE, 2021, 825–32.

[35] Y. Luo and N. Mesgarani, “Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8), 2019, 1256–66.

[36] Y. Luo and N. Mesgarani, “TasNet: time-domain audio separation network for real-time, single-channel speech separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, 696–700.

[37] M. Maciejewski, G. Wichern, E. McQuinn, and J. Le Roux, “WHAMR!: Noisy and reverberant single-channel speech separation”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, 696–700.

[38] Z. Mu, X. Yang, X. Yang, and W. Zhu, "A multi-stage triple-path method for speech separation in noisy and reverberant environments", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, 1–5.

[39] A. Narayanan and D. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4), 2014, 826–35.

[40] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2015, 5206–10.

[41] M. Pariente, S. Cornell, J. Cosentino, S. Sivasankaran, E. Tzinis, J. Heitkaemper, M. Olvera, F.-R. Stöter, M. Hu, J. M. Martín-Doñas, *et al.*, "Asteroid: the PyTorch-based audio source separation toolkit for researchers", *arXiv preprint arXiv:2005.04132*, 2020.

[42] W. Ravenscroft, S. Goetze, and T. Hain, "Receptive field analysis of temporal convolutional networks for monaural speech dereverberation", in *30th European Signal Processing Conference (EUSIPCO)*, IEEE, 2022, 80–4.

[43] W. Ravenscroft, S. Goetze, and T. Hain, "Utterance weighted multi-dilation temporal convolutional networks for monaural speech dereverberation", in *International Workshop on Acoustic Signal Enhancement (IWAENC)*, IEEE, 2022, 1–5.

[44] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 2, IEEE, 2001, 749–52.

[45] K. Saijo, W. Zhang, Z.-Q. Wang, S. Watanabe, T. Kobayashi, and T. Ogawa, "A Single Speech Enhancement Model Unifying Dereverberation, Denoising, Speaker Counting, Separation, and Extraction", in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2023, 1–6.

[46] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, 351–5.

[47] H. Shi, M. Mimura, and T. Kawahara, "Waveform-Domain Speech Enhancement Using Spectrogram Encoding for Robust Speech Recognition", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32, 2024, 3049–60.

[48] H. Shi, K. Shimada, M. Hirano, T. Shibuya, Y. Koyama, Z. Zhong, S. Takahashi, T. Kawahara, and Y. Mitsufuji, "Diffusion-Based Speech Enhancement with Joint Generative and Predictive Decoders", in *International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2024, 12951–5.

[49] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, 21–5.

[50] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech", *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7), 2011, 2125–36.

[51] K. Tan and D. Wang, "A Two-Stage Approach to Noisy Cochannel Speech Separation with Gated Residual Networks.", in *Annual Conference of the International Speech Communication Association*, ISCA, 2018, 3484–8.

[52] E. Tzinis, Z. Wang, and P. Smaragdis, "Sudo RM -RF: Efficient networks for universal audio source separation", in *International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2020, 1–6.

[53] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 14(4), 2006, 1462–9.

[54] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10), 2018, 1702–26.

[55] H. Wang, "Robust Speech Enhancement and Super-resolution Across Multiple Signal Domains", *PhD thesis*, The Ohio State University, 2024.

[56] H. Wang and D. Wang, "Neural cascade architecture with triple-domain loss for speech enhancement", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30, 2021, 734–43.

[57] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, "TF-GridNet: Making time-frequency domain models great again for monaural speaker separation", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, 1–5.

[58] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "WHAM!: Extending Speech Separation to Noisy Environments", in *Annual Conference of the International Speech Communication Association*, ISCA, 2019, 1368–72.

[59] G. Wichern and J. Le Roux, "Phase reconstruction with learned time-frequency representations for single-channel speech separation", in *16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, IEEE, 2018, 396–400.

[60] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(3), 2015, 483–92.

[61] J. Wu, Y. Xu, S.-X. Zhang, L.-W. Chen, M. Yu, L. Xie, and D. Yu, "Time domain audio visual speech separation", in *IEEE automatic speech recognition and understanding workshop (ASRU)*, IEEE, 2019, 667–73.

[62] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, 241–5.

[63] H. Zhang and D. Wang, "Neural cascade architecture for joint acoustic echo and noise suppression", in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, 671–5.

[64] L. Zhang, Z. Shi, J. Han, A. Shi, and D. Ma, "FurcaNeXt: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks", in *MultiMedia Modeling: 26th International Conference, MMM*, Springer, 2020, 653–65.

[65] S. Zhao and B. Ma, "Mossformer: Pushing the performance limit of monaural speech separation using gated single-head transformer with convolution-augmented joint self-attentions", in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, 1–5.

[66] S. Zhao, Y. Ma, C. Ni, C. Zhang, H. Wang, T. H. Nguyen, K. Zhou, J. Q. Yip, D. Ng, and B. Ma, "Mossformer2: Combining transformer and rnn-free recurrent network for enhanced time-domain monaural speech separation", in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2024, 10356–60.

[67] Y. Zhao, Z.-Q. Wang, and D. Wang, "Two-stage deep learning for noisy-reverberant speech enhancement", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(1), 2018, 53–62.

[68] Q.-S. Zhu, J. Zhang, Z.-Q. Zhang, and L.-R. Dai, "Joint training of speech enhancement and self-supervised model for noise-robust ASR", *arXiv preprint arXiv:2205.13293*, 2022.