AN EFFICIENT ENCODER-DECODER ARCHITECTURE WITH TOP-DOWN SELECTIVE ATTENTION FOR SPEECH SEPARATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks have shown excellent prospects in speech separation tasks. However, finding the balance between model complexity and performance is still challenging in real-world applications. In addition, some existing lightweight models still have a significant performance gap compared to state-of-the-art models, since they usually pursue efficiency at the expense of model performance. In this paper, we provided a bio-inspired efficient encoder-decoder architecture by mimicking the brain's top-down selective attention, called TSANet, with decreased model scale and complexity without sacrificing performance. The model contains three modules: a dense bottom-up connection (DBC) encoder, a selective attention (SA) module and a local feature transform (LFT) decoder. Firstly, the DBC encoder focuses on obtaining multi-scale features through dense connections. Second, the SA module extracts global features at the top layer using multi-scale features, and then the global features are fused with features from different scales by using top-down selective attention. Finally, the adjacent layer features are used as the input into the LFT decoder to reconstruct the acoustic features by modulating the fused features through affine transform. On three benchmark datasets, TSANet consistently achieved state-of-the-art (SOTA) separation performance and faster separation efficiency. Our study suggests that top-down selective attention can be a more efficient strategy for speech separation with less computational costs.

1 INTRODUCTION

In cocktail parties, people's communications are inevitably disturbed by various sounds (Bronkhorst, 2015; Cherry, 1953), such as environmental noise and extraneous audio signals, potentially affecting the quality of communication. Humans can effortlessly perceive the speech signal of a target speaker in a cocktail party to improve the accuracy of speech recognition (Haykin & Chen, 2005). In speech processing field, the corresponding challenge is to separate different speakers' audios from the mixture audio, known as speech separation.

Due to the rapid development of deep neural networks (DNNs), DNN-based speech separation methods have significantly improved (Luo & Mesgarani, 2019; Luo et al., 2020; Tzinis et al., 2020; Chen et al., 2020; Subakan et al., 2021; Hu et al., 2021), especially in the time domain. As in natural language processing, the SOTA speech separation methods are now embracing increasingly complex models to achieve better separation performance, such as DPTNet (Chen et al., 2020) and Sepformer (Subakan et al., 2021). These models typically use multiple transformer layers (Vaswani et al., 2017) to capture longer contextual information, leading to a large number of parameters and high computational costs and having a hard time deploying to edge devices. However, we question whether such complexity is always needed in order to improve the separation performance. Recently, other researchers have proposed some lightweight speech separation models to confirm that computationally small models can also achieve competitive separation performance, such as SuDORM-RF (Tzinis et al., 2020) and A-FRCNN (Hu et al., 2021). These encoder-decoder speech separation networks have shown significant potential. However, these models have their own specific problems, such as poor separation performance of SuDORM-RF and high complexity of A-FRCNN.

Hence, we empirically hope to investigate whether lightweight solutions exist for encoder-decoder architectures that can improve speech separation performance.

Human brain has the ability to process large amounts of sensory information with extremely low energy consumption (Attwell & Laughlin, 2001; Howarth et al., 2012). Encoder-decoder speech separation networks (e.g., SuDORM-RF and A-FRCNN) contain a large number of top-down, bottom-up, and lateral connections, similar to the brain's hierarchical structure for processing sensory information (Park & Friston, 2013). From a neuroscience perspective, mimicking the biological neural network architecture is a promising way to find a simple and efficient speech separation proposal. However, the aforementioned encoder-decoder architectures mainly simulated the interaction between lower (e.g., A1) and higher (e.g., A2) functional areas in primates, neglecting the role of higher cortices in speech signal processing. Previous neuroscience studies (Rauschecker & Scott, 2009; Schreiner & Winer, 2007) have found that the hierarchical structure of cerebral cortex projects information from lower auditory cortices in parallel to higher auditory cortices with complex response properties, which are associated with a large number of bottom-up connections in the cortex. This connectivity pattern is similar to that of DenseNet (Huang et al., 2017) in computer vision, projecting features from the shallow output to the deeper layer at the same time. Additionally, the selective attention plays a critical role in tasks where task-independent signals or background noise are distinguished (Elhilali et al., 2009; Mesgarani & Chang, 2012). It is widely believed that this ability to distinguish between different audios is related to selective attention, a psychological process that allows individuals to focus on a specified target. Auditory selective attention mechanisms can enhance the neural processing of task-relevant sounds while inhibiting task-irrelevant neural activity (Okamoto et al., 2011). However, the way how attentional mechanisms affect the processing of complex auditory scenes is largely unknown (Elhilali et al., 2009).

In this paper, we propose an encoder-decoder speech separation network with top-down selective attention called TSANet, inspired by higher cortical areas and selective attention to process sensory information in a brain-like, low-complexity manner. Our network is able to present better performance when compared to state-of-the-art methods (e.g. Sepformer (Subakan et al., 2021)) in terms of accuracy in separated audios, while staying lightweight and efficient at runtime. As shown in Figure 1, we first add dense bottom-up connections to the network encoder, namely the DBC encoder, to obtain local features with different temporal resolutions. Then, in the SA module, global features are extracted from local features via the multi-head self-attention (MHSA) followed by the multi-layer perception (MLP). Third, we add top-down selective attention to different scale features by using the global features, allowing global features to adjust feature details at the same level. Finally, we designed LFT layers for the network decoder, responsible for performing affine transformations on local features to reconstruct fine-grained acoustic features. The experimental results demonstrated that TSANet achieved state-of-the-art separation performance on three datasets (LRS2-2Mix, Libri2Mix (Cosentino et al., 2020), and WHAM! (Wichern et al., 2019)). For separation efficiency results in LRS2-2Mix dataset – when comparing with Sepformer, TSANet saved about 91% of the parameters and 70% of the computational cost, improving inference speed by 64%, achieving reasonable tradeoff between performance and complexity.

Our contributions can be summarized in the following two points:

1. To reduce the performance gap between encoder-decoder models and state-of-the-art methods, we proposed three components that combine the brain's mechanism (higher cortical functions and selective attention) to recover fine-grained acoustic features.
2. We verify the performance and effectiveness of TSANet in speech separation tasks. Compared with the state-of-the-art models, TSANet achieved state-of-the-art performance on all three benchmark datasets with fewer parameters, lower computational cost, and faster inference speed.

2 Related work

In recent years, most of DNN-based speech separation methods has been divided into two main categories: Conv-TasNet (Luo & Mesgarani, 2019) based and DualPathRNN (Luo et al., 2020) based. The reason is twofold. First, both models perform speech separation on time-domain, presenting a significant performance improvement when compared to those operating on time-frequency domain

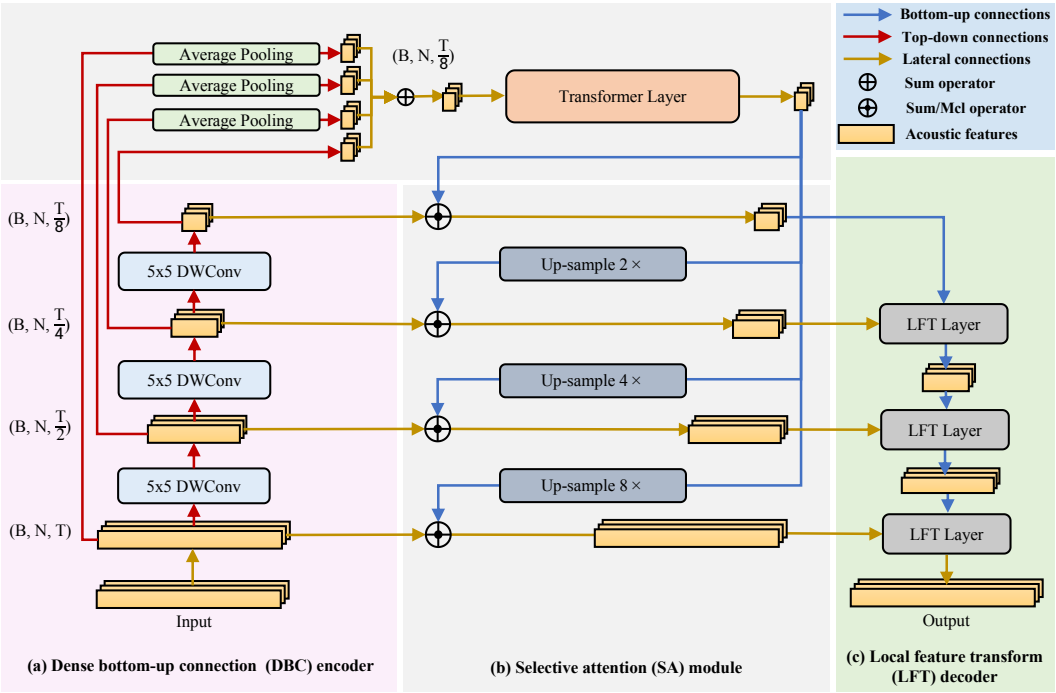

Figure 1: Main architecture of TSANet. B, N, and T denote batch size, number of channels, and length of features, respectively. By downsampling S times, TSANet contains $S + 1$ features with different temporal resolutions. Here, we set S to 3. The red, blue, and orange arrows indicate bottom-up, top-down, and lateral connections, respectively. (a) The structure of the DBC encoder, where "5×5 DWConv" denotes a depthwise convolutional layer with a kernel size of 5 and stride size of 2 followed by global layer normalization (GLN) (Luo & Mesgarani, 2019). (b) The structure of the SA module, where MHSA represents multi-head self-attention layer; MLP indicates a network structure consisting of three convolutional layers; "Sum/Mcl" stands for summation or multiplication operation. Each convolutional layer is followed by GLN. (c) The structure of LFT decoder, where the LFT layer adaptively modulates features from different scales by a set of learnable parameters.

(Hershey et al., 2016; Chen et al., 2017; Yu et al., 2017). Secondly, both models are highly modular networks, allowing more straightforward modifications over network architectures to further improve separation performance. Recently, the DualPathRNN-based methods Sepformer (Subakan et al., 2021) and DPTNet (Chen et al., 2020) replaced LSTM layers with transformer layers to avoid performance degradation due to long-term dependencies and to process data in parallel for efficiency. These methods achieved better separation performance, but the number of model parameters and computational cost became larger due to additional transformer layers. Although large-scale neural networks achieved better performance in various tasks such as BERT (Devlin et al., 2019) and DALL·E 2 (Ramesh et al., 2022), it is also an important topic to design lightweight models that can be deployed to low-resource platforms.

The encoder-decoder speech separation model SuDORM-RF (Tzinis et al., 2020) achieved a certain extent of trade-off between model complexity and runtime efficiency, but its performance was still far from SOTA methods. In addition, other researchers have designed a fully recurrent convolutional neural network A-FRCNN (Hu et al., 2021) with asynchronous update scheme from the perspective of brain processing information, and obtained competitive separation results with a relatively small number of parameters, but the computational complexity of this model still has a lot to improve further. Therefore, we need to consider how to construct a smaller and better speech separation model.

The brain's sensory system is hierarchically organized, allowing sensory information (e.g., auditory information) to rise through a series of brain regions before reaching the higher sensory cortex (e.g.,

higher auditory cortex) (Rauschecker & Scott, 2009). Typically, primary sensory cortices send projections to secondary sensory cortical areas, and also projects to areas outside the sensory pathways (higher cortices) (Jones & Powell, 1969). In some cases, many bottom-up connections are projected to higher cortical areas (Sarter et al., 2001). The interaction between bottom-up and top-down is particularly critical when the target and interfering sources overlap in time and frequency (O'Sullivan et al., 2019). In addition, the higher cortex can receive projections from the primary cortex (Zatorre & Schönwiesner, 2011) and obtain phonological, semantic, and melodic (Homma & Bajo, 2021) information. Also, the higher-order auditory cortex provides an object-based representation of target speech in top-down modulation of the primary auditory cortex in a multi-person auditory scene (Puvvada & Simon, 2017). In addition, selective attention is essential for human to process cocktail party scenes (Power et al., 2012; Rimmele et al., 2015; Golumbic et al., 2013), enhancing neural processing in auditory areas. It remains to be explored how exactly attention affects the different auditory processing stages. However, these evidences only suggested how the neural system works. It is still unclear whether it performs better than existing schemes in engineering tasks. In this paper, we will focus on designing lightweight neural networks for speech separation tasks by mimicking the neural system processing audio signals.

3 METHOD

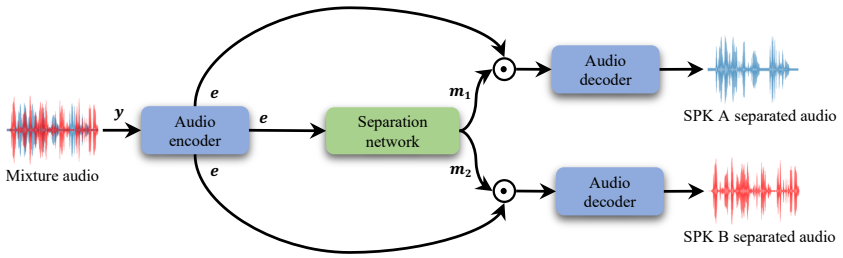

Figure 2: The overall pipeline for the speech separation task. We assume that the mixture audio contains two speakers. The parameters are shared between the two audio decoders.

We aim to design an efficient speech separation model to reconstruct clearer target speech signals. To alleviate the computational bottleneck, we keep most of model's computation at the top layer, where a small temporal resolution takes place that can significantly reduce computational effort. We first present the overall pipeline for the speech separation task (see Figure 2). Then, we describe the three main components in TSANet: (a) dense bottom-up connection (DBC) encoder , (b) selective attention (SA) module and (c) local feature transform (LFT) decoder. Finally, similar to A-FRCNN Hu et al. (2021), we cycle TSANet several times and use the output features from the last cycle as the final separation output.

3.1 OVERALL PIPELINE

Given an audio containing multiple speakers, the speech separation methods aim to extract different speakers' utterances and route them to different output channels. Let $\boldsymbol{y} \in \mathbb{R}^{1\times T}$ be a multi-speaker time-domain audio signal with the length T:

$$\boldsymbol{y} = \sum_{i}^{C} \boldsymbol{x}_i + \boldsymbol{n}, \tag{1}$$

consisting of C speaker signals $\boldsymbol{x}_i \in \mathbb{R}^{1\times T}$ plus the noise signal $\boldsymbol{n} \in \mathbb{R}^{1\times T}$. We expect the separated speeches $\bar{\boldsymbol{x}}_i \in \mathbb{R}^{1\times T}$ from all speakers to be closer to target speeches $\boldsymbol{x}_i$.

Our proposed method uses the same three-part pipeline as in Conv-TasNet (Luo & Mesgarani, 2019): *an audio encoder*, *a separation network* and *an audio decoder*. The audio encoder transforms $\boldsymbol{y}$ into a frame-based N-dimensional embedding sequence $\boldsymbol{E} \in \mathbb{R}^{N\times T'}$ with length T', called mixture audio embedding, which can be implemented using a 1-D convolutional layer. Similar to existing

separation methods, instead of directly estimating the target speech embeddings, we use a DNN-based separation network $f(\boldsymbol{E};\boldsymbol{\theta})$ to generate a set of masks $\boldsymbol{M} = \{\boldsymbol{M}_i \in \mathbb{R}^{N\times T'} | i = 1, ..., C\}$ associated with the target speakers (see details in Section 3.2, 3.3, 3.4 and 3.5). Each target speech embedding $\bar{\boldsymbol{E}}_i \in \mathbb{R}^{N\times T'}$ is generated by applying the corresponding mask $\boldsymbol{M}_i$ to the mixture audio embedding $\boldsymbol{E}$:

$$\bar{\boldsymbol{E}}_i = \boldsymbol{E} \odot \boldsymbol{M}_i, \tag{2}$$

where $\odot$ denotes element-wise product. Finally, the target waveform $\bar{\boldsymbol{x}}_i$ is reconstructed using the target speech embedding $\bar{\boldsymbol{E}}_i$ through an audio decoder, which can be implemented using a 1-D transposed convolutional layer. Next, we present three components in the separation network TSANet.

3.2 Dense bottom-up connection (DBC) encoder

The DBC encoder is the first component of TSANet, designed for extracting features at different temporal resolutions and project them to the highest-level regions using dense bottom-up connections. Lower layers have a higher temporal resolution, while higher layers have a lower temporal resolution. The DBC encoder works as follows (see Figure 1a).

For an input audio mixture embedding $\boldsymbol{E}$, our proposed DBC encoder processes $\boldsymbol{E}$ in a bottom-up manner step by step, followed by obtaining features $\{\boldsymbol{F}_i \in \mathbb{R}^{N\times \frac{T'}{2^{i-1}}} | i = 1, ..., S+1\}$ with different temporal resolutions, where S denotes the number of down-sampling. The bottom-up connections in the DBC encoder are realized by down-sampling layers, which consist of a 5×5 convolutional layer followed by a global normalization layer (GLN) (Luo & Mesgarani, 2019) and PReLU. To efficiently expand the perceptual field, the convolutional layers in the DBC encoder use dilation convolution instead of standard convolution to aggregate longer contexts.

3.3 Selective attention (SA) module

The SA module is divided into two significant steps:

(1). The SA module receives multi-scale features $\{\boldsymbol{F}_i | i = 1, ..., S+1\}$ as input and tries to obtain attention signals $\boldsymbol{G}^m \in \mathbb{R}^{N\times \frac{T'}{2^S}}$.

(2). The SA module uses the $\boldsymbol{G}^m$ to guide the top-down modulation of $\{\boldsymbol{F}_i | i = 1, ..., S+1\}$.

The details of the first step in the SA module are as follows. First, taking features $\{\boldsymbol{F}_i | i = 1, ..., S+1\}$ with different temporal resolutions as input, we use the average pooling layers $p(\cdot)$ to compress these features by temporal dimension from $\frac{T'}{2^{i-1}}$ to $\frac{T'}{2^S}$ for minimizing computational costs. The features with different temporal resolutions are fused into the global feature $\boldsymbol{G} \in \mathbb{R}^{N\times \frac{T'}{2^S}}$ by a summation operation instead of a concatenation operation, mainly to reduce the computational cost of the subsequent MHSA layer. The calculation of global feature can be written as:

$$\boldsymbol{G} = \sum_{i}^{S} p(\boldsymbol{F}_i). \tag{3}$$

Finally, $\boldsymbol{G}$ will be treated as input to the MHAS and MLP layers to obtain the speaker's acoustic patterns from global contexts.

The MHSA layer is similar to the original structure defined in Transformer (Vaswani et al., 2017), as shown in Figure 1b up, which is able to learn the internal structure of speech and some semantic features for contextual modeling of speech sequences using the self-attention mechanism (Wang et al., 2020; 2021). First, the position encoding in MHSA layer will add the position information ϵ of different frames in the global feature $\boldsymbol{G}$ to get $\boldsymbol{G}^\epsilon = \boldsymbol{G} + \epsilon, \boldsymbol{G}^\epsilon \in \mathbb{R}^{N\times \frac{T'}{2^S}}$. Then, in MHSA, each attention head calculates the dot product attention between all elements of $\boldsymbol{G}^\epsilon$ in different feature spaces to direct model focus to different aspects of information. Finally, the output $\boldsymbol{G}^a \in \mathbb{R}^{N\times \frac{T'}{2^S}}$ of MHSA is fed into GLN and then connected $\boldsymbol{G}$ through the residual connection to obtain $\boldsymbol{G}^t \in \mathbb{R}^{N\times \frac{T'}{2^S}}$.

The MLP layer followed by MHSA layer consists of three convolutional layers. First, $\boldsymbol{G}^t$ is fed into the first 1×1 convolutional layer with GLN and mapped to the $2N$-dimensional representation space. Then, the second convolutional layer is composed of 5×5 depthwise convolution with GLN using a small number of parameters aimed at exploring the contextual relationships among sequences. Finally, similar to the first convolutional layer, the third 1×1 convolutional layer compresses the $2N$-dimensional features into N-dimensions to obtain $\boldsymbol{G}^m \in \mathbb{R}^{N\times\frac{T'}{2^S}}$ and use a residual connection to alleviate the vanishing gradient problem.

In the MHSA and MLP layer, the global feature $\boldsymbol{G}$ is used to extract $\boldsymbol{G}^m$, which contains multi-scale semantics. To better utilize $\boldsymbol{G}^m$ to guide the top-down modulation of features $\{\boldsymbol{F}_i|i = 1, ..., S+1\}$, we added top-down selective attention (see Figure 1b down) to exchange information between $\boldsymbol{G}^m$ and $\boldsymbol{F}_i$. In this way, the multi-scale semantic information of $\boldsymbol{G}^m$ can be used as attention signal to guide the local features $\boldsymbol{F}_i$ to focus more on feature details that may be lost in the bottom-up path in order to improve quality of separated audios.

The details of the second step in SA module are as follows. The $\boldsymbol{G}^m$ and $\boldsymbol{F}_i$ are fused to obtain $\boldsymbol{F}'_i \in \mathbb{R}^{N\times\frac{T'}{2^{i-1}}}$ by element-wise multiplication $\odot$ or summation $\oplus$. The above process is formulated as:

$$\boldsymbol{F}'_i = \sigma(\phi(\boldsymbol{G}^m)) \odot \boldsymbol{F}, \text{or}, \boldsymbol{F}'_i = \sigma(\phi(\boldsymbol{G}^m)) \oplus \boldsymbol{F}, \tag{4}$$

where ϕ indicates nearest neighbor interpolation. The fused features $\boldsymbol{F}'_i$ are used as input to LFT decoder to reconstruct different speakers' features.

3.4 Local feature transform (LFT) decoder

The LFT layer is the core component of the LFT decoder, as shown in Figure 1c. It uses fewer parameters ($\sim$ 0.01M) to generate adaptive parameters ρ and τ from higher-level features to guide the reconstruction of finer-grained speech features. Figure 3 shows detailed architecture of the LFT layer. First, LFT layer learns a set of learnable parameters (ρ, τ) using the features $\boldsymbol{F}'_{i+1}$ from adjacent layers. ρ and τ are obtained by using learnable convolutional layers followed by the nonlinear activation function. Consequently,

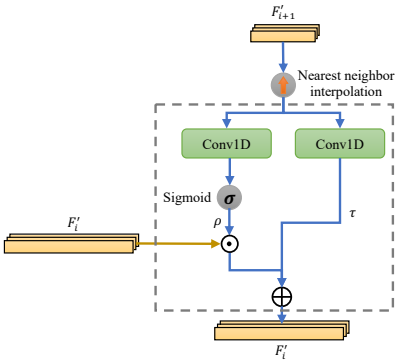

Figure 3: The structure of LFT layer, where "Conv1D" denotes the 5×5 depthwise convolutional layer followed by GLN.

$$\rho = \sigma(h_1(\phi(\boldsymbol{F}'_{i+1}))), \tag{5}$$

$$\tau = h_2(\phi(\boldsymbol{F}'_{i+1})), \tag{6}$$

where h_1 and h_2 denote two different parameters of 1-D convolutional layers, and σ stands for the nonlinear activation function Sigmoid. $\boldsymbol{F}'_i$, ρ and τ have the same feature map dimensions. Then, learnable parameters are used to adaptively modulate $\boldsymbol{F}'_i$' via an affine transform. This process is formulated by

$$\boldsymbol{F}'_i = \rho \odot \boldsymbol{F}'_i + \tau. \tag{7}$$

After the feature transformation operation from the highest layer to the lowest layer is completed, the output features of the LFT decoder use a convolutional layer followed by an activation function (ReLU) to obtain masks $\boldsymbol{M}$.

3.5 Unfolding method of TSANet

To further save the number of parameters for convenient deployment in low-resource devices, we adopt the same unfolding scheme as A-FRCNN (Hu et al., 2021). Figure 1 shows a single block in the whole unfolding scheme. The unfolding scheme of A-FRCNN is to repeat a single block B times over time (weight sharing), such that the model's input adds up with each block's output as the next block's input.

3.6 TRAINING METHOD

In the training process, we apply the permutation-invariant training (PIT) method (Yu et al., 2017), widely used to solve the permutation problem (Hershey et al., 2016) in order to maximize the scale-invariant signal-to-noise ratio (SI-SNR) (Le Roux et al., 2019). The SI-SNR for each speaker is defined as:

$$\text{SI-SNR}_i = 20 \times log_{10} \frac{||\boldsymbol{A}_{target}||}{||\boldsymbol{e}_{noise}||}, \tag{8}$$

where,

$$\boldsymbol{A}_{target} = \frac{\langle \bar{\boldsymbol{x}}_i, \boldsymbol{x}_i \rangle}{||\boldsymbol{x}_i||_2^2}, \boldsymbol{e}_{noise} = \bar{\boldsymbol{x}}_i - \boldsymbol{A}_{target}. \tag{9}$$

In the above equation, $\langle \cdot, \cdot \rangle$ denotes the inner product, and $|| \cdot ||_2^2$ indicates the L2-norm.

4 EXPERIMENT CONFIGURATIONS

4.1 DATASET SIMULATIONS

We evaluated TSANet and other existing methods on three datasets (LRS2 (Afouras et al., 2018), LibriMix (Cosentino et al., 2020) and WHAM! (Wichern et al., 2019)) used for speech separation tasks. The speaker identities of `training/validation` set and `test` set are non-intersecting, meaning unseen speaker data from the training phase were used during testing.

LRS2-2Mix. The LRS2 dataset (Afouras et al., 2018) contains thousands of video clips acquired through BBC. It contains a large amount of noise and reverberation interference, which is more challenging and closer to the actual environment than the WSJ0 dataset (Garofolo et al., 1993). We have created a new speech separation dataset, LRS2-2Mix, using the LRS2 dataset, where the training set, validation set and test set contain 20000, 5000 and 3000 utterances, respectively. The two different speaker audios with 16 kHz sample rate were randomly selected from the LRS2 corpus and were mixed with signal-to-noise ratios sampled between -5 dB and 5 dB. The data were simulated using a script consistent with WSJ0-2Mix[1]. The length of the mixture audios are 2 seconds.

Libri2Mix (Cosentino et al., 2020). In this dataset, the target speech in each audio mixture was randomly selected from a subset of LibriSpeech's train-100 (Panayotov et al., 2015) and mixed with uniformly sampled Loudness Units relative to Full Scale (LUFS) (Series, 2011) between -25 and -33 dB. To simulate real-world scenarios, random noise samples were added to the audio mixture with loudness uniformly sampled between -38 and -30 LUFS. The audio mixtures contain two different speakers and have a duration of 3 seconds.

WAHM! (Wichern et al., 2019). This dataset acts as a noisy version of WSJ0-2Mix (Hershey et al., 2016). In the WSJ0-2Mix dataset, the overlap between two speaker audios in the mixture is 100%, which is too idealistic, and thus model trained on this dataset does not achieve generalization to speeches from a broader range of speakers (Cosentino et al., 2020). In the WHAM! dataset, speeches were mixed with noise recorded in scenes such as cafes, restaurants and bars. The signal-to-noise ratio of the noise was uniformly sampled between -6db and 3db, making it more challenging audio mixtures than WSJ0-2Mix.

4.2 MODEL AND TRAINING CONFIGURATIONS

We constructed two models with different parameter scales: TSANet Base and TSANet Large. We set the kernel sizes of the audio encoder and decoder on these two models to 4 ms and 2 ms, stride to 1 ms and 0.5 ms. The number of channels were set to 512 for all experiments. Other hyperparameter settings were kept the same for both models. The number of down-sampling S was set to 4. The number of channels C of all convolutional layers in each layer was set to 512, and the SC method proposed by A-FRCNN (Hu et al., 2021) was used to unfold $B = 16$ times at the macroscopic level. For the MHSA layer, we set the number of attention heads to 8, the dimension to 512, and used sine and cosine functions with different frequencies for the position encoding. In addition, we set the number of channels for the three convolutional layers in the MLP layer to (512, 1024, 512), the

[1] http://www.merl.com/demos/deep-clustering/create-speaker-mixtures.zip

kernel sizes to (1, 5, 1), the stride sizes to (1, 1, 1), and the bias settings to (False, True, False). To avoid overfitting, we set the probability of all dropouts to 0.1.

We trained all models for 500 epochs. The batch size was set to 1 at the utterance level. Our proposed model used the Adam (Kingma & Ba, 2015) optimizer with an initial learning rate of 0.001. Once the best model was not found for 15 successive epochs, we adjusted the learning rate to half of the previous one. Moreover, we stopped training early when the best model is not found for 30 successive epochs. During the training process, we used gradient cropping with a maximum L2 norm of 5 to avoid gradient explosion. For all experiments, we used $8\times$ GeForce RTX 3080 for training and testing. The PyTorch implementation of our method can be found in the supplementary material. This project is licensed under the MIT licence.

4.3 Evaluation Metrics

In all experiments, we have mainly reported the results of scale-invariant signal-to-noise ratio improvement (SI-SNRi) (Le Roux et al., 2019) and signal-to-distortion ratio improvement (SDRi) (Vincent et al., 2006) used to measure clarity of separated audios. For measuring model efficiency, we have reported the processing time consumption per second (real-time factor, RTF) for all models, indicated by "RTF" in the tables throughout this paper. It was calculated by processing ten audio tracks of 1 second in length and 16 kHz in sample rate on CPU and GPU (total processing time / 10), represented as "CPU RTF" and "GPU RTF", respectively. The numbers are then averaged after running 1000 times. Also, we used the parameter size and the number of multiply-accumulate operations (MACs) to measure the model scale, where the MACs were calculated using the open-source tool PyTorch-OpCounter[2]. This toolkit is used under the MIT license.

5 Results

5.1 Ablation study

To better understand the effectiveness of our proposed TSANet model, we investigated the impacts of different designs in the model for separation performance, including the DBC encoder, the SA module, and the LFT decoder. As a default setting, we only used TSANet Base for fast validation during the ablation experiments. All experimental results were obtained by training and testing using the LRS2-2Mix dataset.

Effect of DBC encoder. To examine the influence on separation performance between two different features as input to the SA module: multi-scale fused features $\boldsymbol{G}$ and top-layer features $\boldsymbol{F}_{S+1}$, we experimented with each of them separately. The experimental results are shown in Table 1. We selected multi-scale fused features obtained by the DBC encoder as input to the MHSA layer because it achieved better separation performance (0.4 dB ↑) without a remarkable increase in computational costs (0.05 s ↑). This is due to the dense bottom-up connections in the DBC encoder that employ parameter-free average pooling layers to perform the down-sampling operation.

Table 1: Comparison over separation performance and model complexity between different multi-scale fused top-layer features used for input into the MHSA layer.

Methods	SI-SNRi (dB)	SDRi (dB)	Params (M)	MACs (G/s)	CPU RTF (s)
$\boldsymbol{G}$	**13.2**	**13.5**	**2.3**	**4.7**	0.79
$\boldsymbol{F}_{S+1}$	12.8	13.1	2.3	4.7	**0.74**

In addition, we also investigated two different strategies for multi-scale feature fusion, including concatenation and summation. Table 2 shows the results. We observed that summation provides a performance gain compared to concatenation without an increase in computational cost or parameter count. Therefore, we employed summation operation as the fusion strategy.

[2]https://github.com/Lyken17/pytorch-OpCounter

Table 2: Comparison over separation performance and model complexity between different fusion strategies.

Fusion strategy	SI-SNRi (dB)	SDRi (dB)	Params (M)	MACs (G/s)	CPU RTF (s)
Summation	**13.2**	**13.5**	**2.3**	**4.7**	**0.79**
Concatenation	13.0	13.3	3.6	6.1	0.83

Effect of MHSA and MLP layers. We have investigated the effectiveness of MHSA and MLP layers on the separation performance and inference efficiency in Table 3. When removing one of the MHSA and MLP layers, we observed that the impact of MLP ($\downarrow$ 1.3 dB) on performance was more significant than that of the MHSA ($\downarrow$ 0.7 dB). One possible reason may be that the perceptual field of the top-layer features is already large enough so that the transformer structure, known to be good at modeling long sequences, does not improve separation performance as much. In addition, we observed from "- ALL" in Table 3 that the these layers played a critical role in improving separation performance. After removing these layers, the model training could not converge well, and its separation performance dropped dramatically, with an SI-SNRi of only 10.2 dB. These results suggest that these layers effectively learn the interactions between features of different scales to enrich the global feature representation in order to improve model performance.

Table 3: Comparison on the importance of MHSA and MLP layers. The "-" in the first column means that this layer is removed from the original model. "- ALL" indicates that both MHSA and MLP are removed.

Methods	SI-SNRi (dB)	SDRi (dB)	Params (M)	MACs (G/s)	CPU RTF (s)
TSANet	**13.2**	**13.5**	2.3	4.7	0.79
- MHSA	12.5	12.8	1.3	3.7	0.73
- MLP	11.9	12.2	1.3	3.7	0.73
- ALL	10.2	10.5	**0.2**	**2.6**	**0.68**

Effect of SA module. To investigate the fusion strategy between attention signals $\boldsymbol{G}^m$ and local features $\boldsymbol{F}_i$ at each layer, we manipulated attention fusion with summation and multiplication strategies. The experimental results are shown in Table 4. We found that, by using the multiplication operation, we can improve the separation performance by 0.4 dB while the inference efficiency is almost the same as the addition operation. One possible reason for this performance improvement is that the multiplication operation is often used to learn the relationship between different features (Dumoulin et al., 2018). This is in line with our original design intention to use global features as an aid to produce feature details lost in the bottom-up information transfer.

Simultaneously, we also analyzed whether selective attention was necessary or not. When selective attention was absent, the separation performance decreased by 1.6 dB compared to the baseline model (Mcl). This indicated that the selective attention of global features to different layers was a critical factor in the separation performance improvement. The results led to a similar conclusion to previous studies (Woldorff et al., 1993; Lakatos et al., 2013; Talsma et al., 2007) that selective attention can affect auditory processing.

Effect of LFT layer. To examine the effect of the LFT layer on separation performance, we designed two ablation experiments. The results in Table 5 show that the separation performance could be improved by adding higher-level features to the LFT layer and modulating the current layer features using affine transform since removing it would result in a 0.8 dB and 0.7 dB decrease in SI-SNRi and SDRi, respectively, and no acceleration in computational efficiency.

5.2 Comparisons with state-of-the-art methods

As shown in Table 6, we conducted extensive experiments to quantitatively compare speech separation performance of our proposed TSANet with state-of-the-art speech separation models on three

Table 4: Comparison over separation performance and efficiency among summation operations, multiplication operations, and no global features. "Mcl" indicates a multiplication operation; "Sum" indicates a summation operation; "None" means that no global features are used, i.e., no selective attention is used to modulate the different layers of features.

Strategy	SI-SNRi (dB)	SDRi (dB)	Params (M)	MACs (G/s)	CPU RTF (s)
Mcl	13.2	13.5	2.3	4.7	0.79
Sum	12.8	13.1	2.3	4.7	0.78
None	11.9	12.2	2.3	4.7	0.73

Table 5: Comparison with the impact of the LFT layer on the separation performance and efficiency. "$\surd$" indicates that the LFT layer is used to modulate the current-layer features. "$\times$" indicates that the LFT layer is not used.

LFT layer	SI-SNRi (dB)	SDRi (dB)	Params (M)	MACs (G/s)	CPU RTF (s)
$\surd$	13.2	13.5	2.3	4.7	0.79
$\times$	12.4	12.8	2.3	4.6	0.79

datasets (LRS2-2Mix, Libri2Mix and WHAM!). Currently, models with good separation performance are time-domain models, including BLSTM-TasNet (Luo & Mesgarani, 2018), Conv-TasNet (Luo & Mesgarani, 2019), SuDORM-RF (Tzinis et al., 2020), DualPathRNN (Luo et al., 2020), Sepformer (Subakan et al., 2021), and A-FRCNN (Hu et al., 2021). SuDORM-RF contains two variants with good performance, suffixed with 1.0x and 2.5x, indicating that these variants consist of 16 and 40 blocks, respectively. A-FRCNN has a suffix "16" denoting that the model is expanded 16 times at the macro level. LRS2-2Mix is the newly proposed speech separation dataset in this paper, so none of these models reported results on this dataset. We trained and tested these models using the Asteroid Toolkit (Pariente et al., 2020) to obtain the missing results.

As shown in Table 6, TSANet Large obtained better separation performance and little complexity than the state-of-the-art models on the three datasets, clearly demonstrating that the separation network we designed has better trade-off between separation performance and efficiency. In the LRS2-2Mix dataset, TSANet Large obtained about 5% separation performance improvement, saving about 91% of parameters and 90% of computation costs compared to the previous SOTA model Sepformer. Likewise, TSANet Large also obtained the best separation performance and the least computational cost in the Libri2Mix and WHAM! datasets. In addition, the experimental results of TSANet Base showed that increasing window length and hop size of the audio encoder and decoder will reduce the separation performance while effectively reducing model complexity to accelerate inference, which is consistent with the findings in (de Oliveira et al., 2022). Although TSANet Base did not achieve the best separation performance, the model did obtain competitive results to a certain extent. For example, on the LRS2-2Mix dataset, about 50% of the model computation is saved compared to TSANet Large, while only 7% of the separation performance is lost.

In addition to model parameters and MACs, inference speed on CPU as well as training and inference speed on GPU are also important indicators of model complexity, as models with small parameters may also be slow to train and infer. By using GPU and CPU RTF as metrics for evaluating model complexity, we evaluated in more detail the performance of our approach and that of previous SOTA separation models. In Table 7, we observed that TSANet Base could outperform the currently proposed speech separation methods in model complexity while achieving a competitive separation performance. Specifically, compared to the state-of-the-art separation model Sepformer, TSANet Base lost only 2% in separation performance while being 54% faster in backward time, significantly reducing the time consumption during training. Most tasks benefit from larger model sizes, such as pre-trained models (Devlin et al., 2019; Liu et al., 2019) in the natural language processing field. By experimenting with TSANet Base for scale-up, our proposed TSANet Large achieves a 1.0 dB gain in separation performance, but we did not observe a significant decrease in efficiency on GPU. Compared with lightweight models A-FRCNN and SuDORM-RF 2.5x, TSANet Large achieved 9%

Table 6: Quantitative comparison between TSANet and other state-of-the-art systems on three datasets (test set). "-" indicates that their results have not been reported in other papers. "*" denotes that they were not reported in the original paper but were trained and tested using the Asteroid Toolkit (Pariente et al., 2020). The toolkit was used under the MIT licence. Larger SI-SNRIi and SDRi values indicate better separation performance.

Model [Years]	LRS2-2Mix		Libri2Mix		WHAM!		Params	MACs
	SI-SNRi	SDRi	SI-SNRi	SDRi	SI-SNRi	SDRi	(M)	(G/s)
BLSTM-TasNet [2018]	6.1*	6.8*	7.9	8.7	9.8	-	23.6	43.0
Conv-TasNet [2019]	10.6*	11.0*	12.2	12.7	12.7	-	5.6	10.2
SuDoRM-RF1.0x [2020]	11.0*	11,4*	13.5	14.0	12.9	13.3	2.7	**4.6**
SuDoRM-RF2.5x [2020]	11.3*	11.7*	14.0	14,4	13,7	14.1	6.4	10.1
DualPathRNN [2020]	12.7*	13.0*	14.1	14.6	13.7	14.1	2.7	85.3
Sepformer [2021]	13.5	13.8*	16.5	17.0	14.4	15.0	26.0	86.9
A-FRCNN-16 [2021]	13.0*	13.3*	16.7	17.2	14.5	14.8	6.1	125.3
TSANet Base (*ours*)	13.2	13.5	16.9	17.4	14.8	15.0	2.3	4.7
TSANet Large (*ours*)	**14.2**	**14.5**	**17.4**	**17.9**	**15.2**	**15.4**	**2.3**	9.1

and 26% separation performance gain, 63% and 54% GPU inference time reduction, 51% and 53% GPU training time reduction, and 67% and 54% CPU inference time reduction, respectively. This demonstrates the potential of TSANet to further improve separation performance without significantly increasing computational cost. It can be easily deployed to low-resource devices and can be guaranteed to run successfully.

Table 7: Comparison over model parameters, inference time and separation performance on the LRS2-2Mix test set. The test environment for feedforward RTF (F GPU RTF) and backward RTF (B GPU RTF) is Nvidia GeForce RTX 2080 Ti, while the test environment for CPU RTF is Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz, single-threaded.

Model	SI-SNRi	F GPU RTF (ms)	B GPU RTF (ms)	CPU RTF (s)
BLSTM-TasNet	6.1	233.85	654.14	5.9
Conv-TasNet	10.6	15.28	56.91	0.82
SuDoRM-RF 1.0x	11.0	27.86	95.37	2.00
SuDoRM-RF 2.5x	11.3	64.70	228.57	3.91
DualPathRNN	12.7	88.79	241.54	8.13
A-FRCNN-16	13.0	81.16	218.65	5.32
Sepformer	13.5	65.61	184.91	7.55
TSANet Base (*ours*)	13.2	11.76	86.56	0.79
TSANet Large (*ours*)	14.2	23.77	97.92	1.78

6 CONCLUSIONS

To fill the gap between the performance of lightweight models and state-of-the-art models, we designed an encoder-decoder speech separation network with selective attention that is well-suited for edge devices. Our method was inspired by the brain's mechanism for processing auditory scene analysis. The extensive experimental results showed that TSANet could significantly reduce the model's scale and complexity while ensuring separation performance and inference speed. Specifically, the good trade-off between performance and efficiency of TSANet benefits from the three components, especially the SA module. We performed ablation studies for all three components. The results demonstrated that adding these three components into the encoder-decoder network can obtain better separation performance without significant increase in model's complexity and runtime cost.

Limitations and future work: In this work, the baseline speech dataset contains speech signals with only two speakers. We did not explore datasets with more speakers, leaving these for future work. In

addition, model quantization techniques will be applied with TSANet to create models with smaller sizes and complexity. Finally, it can be deployed into mobile devices and analyzed for performance metrics such as power consumption, aiming for better support for edge computing devices.

REFERENCES

Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

David Attwell and Simon B Laughlin. An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow & Metabolism*, 21(10):1133–1145, 2001.

Adelbert W Bronkhorst. The cocktail-party problem revisited: early processing and selection of multi-talker speech. *Attention, Perception, & Psychophysics*, 77(5):1465–1487, 2015.

Jingjing Chen, Qirong Mao, and Dong 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 (Interspeech)*, pp. 2642–2646, 2020.

Zhuo Chen, Yi Luo, and Nima Mesgarani. Deep attractor network for single-microphone speaker separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 246–250. IEEE, 2017.

E Colin Cherry. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the acoustical society of America*, 25(5):975–979, 1953.

Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent. Librimix: an open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*, 2020.

Danilo de Oliveira, Tal Peer, and Timo Gerkmann. Efficient transformer-based speech enhancement using long frames and stft magnitudes. *arXiv preprint arXiv:2206.11703*, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.

Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio. Feature-wise transformations. *Distill*, 3(7):e11, 2018.

Mounya Elhilali, Juanjuan Xiang, Shihab A Shamma, and Jonathan Z Simon. Interaction between attention and bottom-up saliency mediates the representation of foreground and background in an auditory scene. *PLoS biology*, 7(6):e1000129, 2009.

John Garofolo, David Graff, Doug Paul, and David Pallett. Csr-i (wsj0) complete ldc93s6a. *Web Download. Philadelphia: Linguistic Data Consortium*, 83, 1993.

Elana M Zion Golumbic, Nai Ding, Stephan Bickel, Peter Lakatos, Catherine A Schevon, Guy M McKhann, Robert R Goodman, Ronald Emerson, Ashesh D Mehta, Jonathan Z Simon, et al. Mechanisms underlying selective neuronal tracking of attended speech at a "cocktail party". *Neuron*, 77(5):980–991, 2013.

Simon Haykin and Zhe Chen. The cocktail party problem. *Neural computation*, 17(9):1875–1902, 2005.

John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: discriminative embeddings for segmentation and separation. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 31–35. IEEE, 2016.

Natsumi Y Homma and Victoria M Bajo. Lemniscal corticothalamic feedback in auditory scene analysis. *Frontiers in neuroscience*, pp. 1067, 2021.

Clare Howarth, Padraig Gleeson, and David Attwell. Updated energy budgets for neural computation in the neocortex and cerebellum. *Journal of Cerebral Blood Flow & Metabolism*, 32(7): 1222–1232, 2012.

Xiaolin Hu, Kai Li, Weiyi Zhang, Yi Luo, Jean-Marie Lemercier, and Timo Gerkmann. Speech separation using an asynchronous fully recurrent convolutional neural network. volume 34, pp. 22509–22522, 2021.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 4700–4708, 2017.

EG Jones and TPS Powell. Connexions of the somatic sensory cortex of the rhesus monkey: I.—ipsilateral cortical connexions. *Brain*, 92(3):477–502, 1969.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Peter Lakatos, Gabriella Musacchia, Monica N O'Connel, Arnaud Y Falchier, Daniel C Javitt, and Charles E Schroeder. The spectrotemporal filter mechanism of auditory selective attention. *Neuron*, 77(4):750–761, 2013.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. Sdr–half-baked or well done? In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 626–630. IEEE, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Yi Luo and Nima Mesgarani. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 696–700. IEEE, 2018.

Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8): 1256–1266, 2019.

Yi Luo, Zhuo Chen, and Takuya Yoshioka. Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 46–50. IEEE, 2020.

Nima Mesgarani and Edward F Chang. Selective cortical representation of attended speaker in multi-talker speech perception. *Nature*, 485(7397):233–236, 2012.

Hidehiko Okamoto, Henning Stracke, Patrick Bermudez, and Christo Pantev. Sound processing hierarchy within human auditory cortex. *Journal of Cognitive Neuroscience*, 23(8):1855–1863, 2011.

James O'Sullivan, Jose Herrero, Elliot Smith, Catherine Schevon, Guy M McKhann, Sameer A Sheth, Ashesh D Mehta, and Nima Mesgarani. Hierarchical encoding of attended auditory objects in multi-talker speech perception. *Neuron*, 104(6):1195–1209, 2019.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Manuel Pariente, Samuele Cornell, Joris Cosentino, Sunit Sivasankaran, Efthymios Tzinis, Jens Heitkaemper, Michel Olvera, Fabian-Robert Stöter, Mathieu Hu, Juan M. Martín-Doñas, David Ditter, Ariel Frank, Antoine Deleforge, and Emmanuel Vincent. Asteroid: the PyTorch-based audio source separation toolkit for researchers. In *Annual Conference of the International Speech Communication Association (Interspeech)*, 2020.

Hae-Jeong Park and Karl Friston. Structural and functional brain networks: from connections to cognition. *Science*, 342(6158):1238411, 2013.

Alan J Power, John J Foxe, Emma-Jane Forde, Richard B Reilly, and Edmund C Lalor. At what time is the cocktail party? a late locus of selective attention to natural speech. *European Journal of Neuroscience*, 35(9):1497–1503, 2012.

Krishna C Puvvada and Jonathan Z Simon. Cortical representations of speech in a multitalker auditory scene. *Journal of Neuroscience*, 37(38):9189–9196, 2017.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Josef P Rauschecker and Sophie K Scott. Maps and streams in the auditory cortex: nonhuman primates illuminate human speech processing. *Nature neuroscience*, 12(6):718–724, 2009.

Johanna M Rimmele, Elana Zion Golumbic, Erich Schröger, and David Poeppel. The effects of selective attention and speech acoustics on neural speech-tracking in a multi-talker scene. *Cortex*, 68:144–154, 2015.

Martin Sarter, Ben Givens, and John P Bruno. The cognitive neuroscience of sustained attention: where top-down meets bottom-up. *Brain research reviews*, 35(2):146–160, 2001.

Christoph E Schreiner and Jeffery A Winer. Auditory cortex mapmaking: principles, projections, and plasticity. *Neuron*, 56(2):356–365, 2007.

BS Series. Algorithms to measure audio programme loudness and true-peak audio level. 2011.

Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. Attention is all you need in speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 21–25. IEEE, 2021.

Durk Talsma, Tracy J Doty, and Marty G Woldorff. Selective attention and audiovisual integration: is attending to both modalities a prerequisite for early integration? *Cerebral cortex*, 17(3):679–690, 2007.

Efthymios Tzinis, Zhepei Wang, and Paris Smaragdis. Sudo rm-rf: efficient networks for universal audio source separation. In *IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. volume 30, 2017.

Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469, 2006.

Kai Wang, Bengbeng He, and Wei-Ping Zhu. Tstnn: Two-stage transformer based neural network for speech enhancement in the time domain. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7098–7102. IEEE, 2021.

Yongqiang Wang, Abdelrahman Mohamed, Due Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, et al. Transformer-based acoustic modeling for hybrid speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6874–6878. IEEE, 2020.

Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. WHAM!: Extending Speech Separation to Noisy Environments. In *Annual Conference of the International Speech Communication Association (Interspeech)*, pp. 1368–1372, 2019.

Marty G Woldorff, Christopher C Gallen, Scott A Hampson, Steven A Hillyard, Christo Pantev, David Sobel, and Floyd E Bloom. Modulation of early sensory processing in human auditory cortex during auditory selective attention. *Proceedings of the National Academy of Sciences*, 90 (18):8722–8726, 1993.

Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 241–245. IEEE, 2017.

Robert J Zatorre and Marc Schönwiesner. Cortical speech and music processes revealed by functional neuroimaging. In *The Auditory Cortex*, pp. 657–677. Springer, 2011.