

# SANDGLASSET: A LIGHT MULTI-GRANULARITY SELF-ATTENTIVE NETWORK FOR TIME-DOMAIN SPEECH SEPARATION

Max W. Y. Lam<sup>\*</sup>   Jun Wang<sup>\*</sup>   Dan Su<sup>\*</sup>   Dong Yu<sup>†</sup>

<sup>\*</sup> Tencent AI Lab, Shenzhen, China

<sup>†</sup> Tencent AI Lab, Bellevue WA, USA

## ABSTRACT

One of the leading single-channel speech separation (SS) models is based on a TasNet with a dual-path segmentation technique, where the size of each segment remains unchanged throughout all layers. In contrast, our key finding is that multi-granularity features are essential for enhancing contextual modeling and computational efficiency. We introduce a self-attentive network with a novel sandglass-shape, namely Sandglasstet, which advances the state-of-the-art (SOTA) SS performance at significantly smaller model size and computational cost. Forward along each block inside Sandglasstet, the temporal granularity of the features gradually becomes coarser until reaching half of the network blocks, and then successively turns finer towards the raw signal level. We also unfold that residual connections between features with the same granularity are critical for preserving information after passing through the bottleneck layer. Experiments show our Sandglasstet with only 2.3M parameters has achieved the best results on two benchmark SS datasets – WSJ0-2mix and WSJ0-3mix, where the SI-SNRi scores have been improved by absolute 0.8 dB and 2.4 dB, respectively, comparing to the prior SOTA results.

**Index Terms**— Speech separation, multi-granularity, self-attentive network, single-channel

## 1. INTRODUCTION

Separating a relatively clean speech signal in the presence of multiple speaking voices is a fundamental and crucial problem (a.k.a. “cocktail party problem” [1, 2]) for many downstream speech processing tasks [3–5]. In this paper, we focus on the single-channel speech separation (SS) task, which is considerably more challenging than in a multi-channel setting but at the same time applies to broader scenarios, e.g., telephone conversations, many VoIP usage cases, and numerous smartphone applications. The performance of single-channel SS has been recently advanced by a variety of deep learning methods [6–8]. The current leading methods are based on the time-domain audio separation network (TasNet) [6], which takes waveform inputs and directly reconstruct sources by computing time-domain loss with utterance-level permutation invariant training (u-PIT) [9, 10]. In particular, there are many variants of TasNets: the long short-term memory (LSTM) based TasNet [6], the Conv-TasNet [7, 11], the dual-path recurrent neural network (DPRNN) [8], the dual-path Transformer network (DPTNet) [12], the gated DPRNN [13] and the Wavesplit [14].

Previous works of TasNets have shown that a smaller window for encoding improves the separation performance [8, 12, 13], leading to much longer sequences, which poses special challenges for modeling long-term global dependencies. To handle the very long sequences, current SOTA methods [8, 12, 13] employ a dual-path segmentation technique, which performs over a whole encoded

sequence and divides it into intra-segment and inter-segment sequences, to which we simply refer as local and global sequences. A common strategy in prior works [8, 12, 13] is to use RNNs to model both the local and global sequences. Instead, we find that a self-attentive network (SAN) [15] would be a better structure to model the global sequence. Given an  $n$ -length sequence, in SAN every element can connect to another element using a direct path (i.e., in  $\mathcal{O}(1)$  time) rather than recursively processing, resetting, and updating memory (i.e., in  $\mathcal{O}(n)$  time) as in RNNs. Although SAN is notorious for its inefficiency in processing very long sequences due to its inherent quadratic cost, the global sequence length in the dual-path setting becomes feasible for SAN to model.

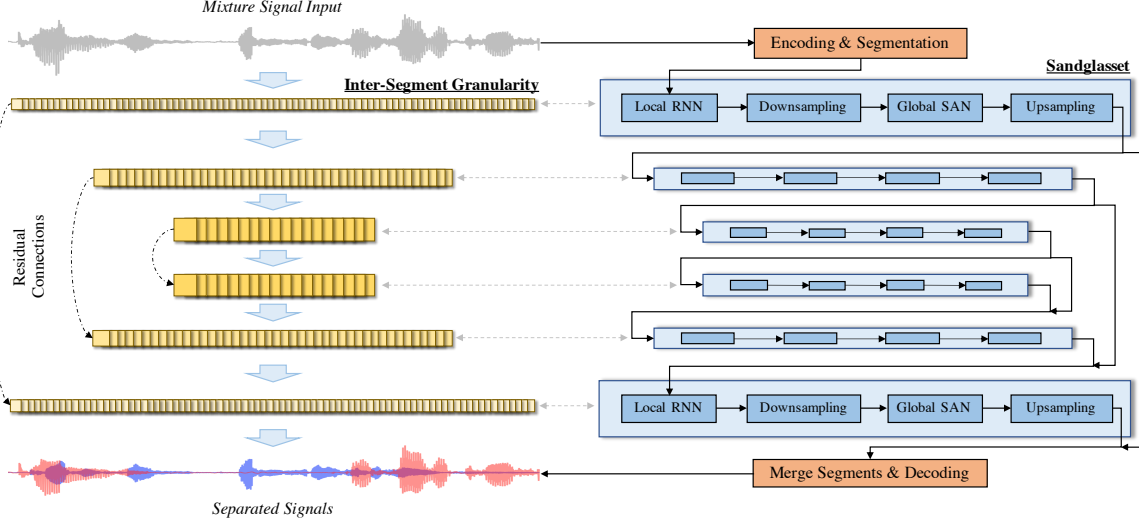
Moreover, existing segmentation-based models generally use a fixed segment size unchanged throughout all layers of computation. Our finding is that the modeling capabilities of these networks could not be fully exploited if constantly modeling the global sequences with only one fixed granularity. Especially, time-domain signals essentially have different abstract contexts, e.g., phonemes, syllables, or words, at various granularity levels. Furthermore, SANs have been proven superior for modeling high-level contexts in a number of tasks [16–19]. These together inspire us to design a new neural network architecture, where features are modeled in multi-granularity by SANs. Consequently, we propose a novel neural network architecture called Sandglasstet, for its sandglass shape and its modest model size and complexity. Forward along each of its blocks, the granularity of the features gradually becomes coarser until reaching half of the network blocks, and then successively turns finer towards the raw signal level. We also unfold that residual connections between features with the same granularity are critical for preserving information after passing through the bottleneck layer.

Finally, the proposed Sandglasstet, which is very light with only 2.3M model parameters, has achieved the SOTA results on two benchmark speech separation datasets – WSJ0-2mix and WSJ0-3mix, where the SI-SNRi scores have been pushed to 20.8 dB and 17.1 dB, surpassing the prior SOTA results by a large margin of absolute 0.8 dB and 2.4 dB. Moreover, compared to the smallest model in literature – DPRNN, our proposed Sandglasstet is remarkably lighter with 58.4% less memory and 66.0% fewer floating-point operations. To the best of our knowledge, Sandglasstet is the first work that models multi-granularity segments using SANs in signal processing.

## 2. SANDGLASSET

### 2.1. Overall Architecture

Our proposed Sandglasstet is composed of  $N$  blocks, as presented on the right diagram of Fig. 1. If information flows from top to bottom, the first  $N/2$  blocks constitute an inverted pyramid in Sand-



**Fig. 1:** An illustration of the information flow inside Sandglassnet. The left diagram shows the multi-granularity features with variable segment sizes that form a sandglass shape; on the right, it shows the Sandglassnet blocks, each of which models a granularity depicted on the left.

glassnet, where the signal frames are successively down-sampled into shorter feature sequences of larger segments in coarser time scales, i.e., large-granularity, high-level abstract features. Then, the last  $N/2$  blocks constitute a pyramid, where these high-level features are then inversely up-sampled back into longer feature sequences of smaller segments in finer time scales, i.e., fine-granularity, low-level features. To preserve information, the up-sampled features in the last  $N/2$  blocks are aggregated with the earlier computed features with the same granularity using residual connections. This processing is useful for better signal reconstruction as well as avoiding gradient vanishing issues. This sandglass-shape processing strategy is capable of modeling multi-scale temporal granularity to process the input signal hierarchically and progressively, e.g., processing sounds, syllables, and words at different block levels successively. In the remainder of this section, we present the inner machinery of each module in a block as shown in the right diagram of Fig. 1.

## 2.2. Encoding and Segmentation

### 2.2.1. TasNet Encoder

First of all, the input signal is a time-domain waveform mixture  $\mathbf{x} \in \mathbb{R}^T$ . Similar to other TasNet systems [6–8], the input mixture signal is encoded into a sequence of 50%-overlapping frames, denoted by  $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_L] \in \mathbb{R}^{M \times L}$ , where  $M$  is a hyperparameter that is generally referred to as the *window length*, and  $L = \lceil 2T/M \rceil$ . In TasNet, we use a ReLU-gated 1D convolutional layer to replace the traditional short-time Fourier transform (STFT) for signal encoding:

$$\hat{\mathbf{X}} = \text{ReLU} \left( \text{Conv1D} \left( \tilde{\mathbf{X}}; \mathbf{U} \right) \right), \quad (1)$$

where  $\text{Conv1D}(\tilde{\mathbf{X}}; \mathbf{U})$  denotes the 1D convolution operation applied on  $\tilde{\mathbf{X}}$  parameterized by a learnable weight  $\mathbf{U} \in \mathbb{R}^{E \times M}$  with  $1 \times 1$  kernels,  $\text{ReLU}(\cdot)$  is the element-wise rectified linear unit used in [8, 12] to ensure non-negative outputs, and  $E$  is the dimensionality of each encoded frame. Instead of directly using  $\hat{\mathbf{X}}$  for the subsequent computation, we linearly map the matrix into bottleneck features  $\mathbf{X} = \mathbf{B}\hat{\mathbf{X}} \in \mathbb{R}^{D \times L}$ , where  $\mathbf{B} \in \mathbb{R}^{D \times E}$  and  $D < E$ .

### 2.2.2. Segmentation Module

Given a sequence of frames in matrix form  $\mathbf{X} \in \mathbb{R}^{D \times L}$ , we use a segmentation module to split  $\mathbf{X}$  into  $S$  50%-overlapping segments, each of length  $K$ . The first and last segments are padded with zeros to create  $S = \lceil 2L/K \rceil$  equal-size segments. These segments can be packed together to create a 3D tensor, denoted by  $\mathcal{X} \in \mathbb{R}^{D \times K \times S}$ . Note that the segment size  $K$  is a hyperparameter that can be used to control the scale of the locality. The segments  $\mathcal{X}$  are then passed to a stack of Sandglassnet blocks.

## 2.3. Sandglassnet Blocks

For the  $b$ -th block, we are given a 3D tensor input  $\mathcal{X}_b \in \mathbb{R}^{D \times K \times S}$ , enclosing  $S$  segments each containing  $K$  frames of  $D$  dimensions. To make the following recurrence relations mathematically sound, we define  $\mathcal{X}_1 = \mathcal{X}$ . As shown in Fig. 1, each Sandglassnet block consists of mainly two operations – firstly processing the intra-segment sequence using a recurrent neural network for modeling locality, as in [8], and secondly modeling the inter-segment sequence using a SAN to capture the global dependencies. Interleaving with these two modules, a downsampling and an upsampling operation alter the granularity of the global sequence to be processed by the SAN.

### 2.3.1. Recurrent Neural Network for Local Sequence Processing

In our task, intra-segment sequences are the local sequences, each of length  $K$ , which contain subtle local details, e.g. temporal or spectral continuity, spectral structure, timbre, etc., which are rather irrelevant to the long-term context. In Sandglassnet, we assign the local sequence processing task to a one-layer RNN. Specifically, in each Sandglassnet block, the 3D tensor  $\mathcal{X}_b^{LR} = \mathcal{X}_b$  obtained from the segmentation process is passed to a bi-directional LSTM of  $H$  hidden nodes. Here, for ease of reference, we use  $\mathcal{X}^{LR}$  and  $\mathcal{Y}^{LR}$  to respectively denote the inputs for the local RNN and the outputs from the local RNN. The superscript  $LR$  is used to differentiate from the corresponding input-output pairs in the global SAN model.

$$\mathcal{Y}_b^{LR} = \left[ \mathbf{M}_b \cdot \text{BiLSTM}_b \left( \mathcal{X}_b^{LR}[:, s, :] \right) + \mathbf{c}_b, s = 1, \dots, S \right], \quad (2)$$

where  $\cdot$  is used to denote matrix multiplication,  $\mathcal{X}_b^{LR}[:, s, :] \in \mathbb{R}^{D \times K}$  refers to the local sequence within the  $s$ -th chunk,  $\mathbf{M}_b \in \mathbb{R}^{D \times 2H}$  and  $\mathbf{c}_b \in \mathbb{R}^D$  are the parameters of a linear transformation.

### 2.3.2. Self-Attentive Network for Multi-Granularity Modeling

After processing the intra-segment sequences each of length  $K$ , we aim at modeling the inter-segment sequences, each of length  $S$ . Noted that inter-segment sequences are likely to encode the contextual information of the speech signal. In Sandglasset, we employ a variable-context-aware self-attentive network (SAN) to capture the global dependencies in different time scales.

Instead of directly taking  $\mathcal{Y}_b^{LR}$  as the input to a SAN, we first apply a layer normalization operation  $\text{LN}(\cdot)$  to the LR layer's output and add a residual connection to the block input:

$$\mathcal{X}_b^{GA} = \text{LN}(\mathcal{Y}_b^{LR}) + \mathcal{X}_b, \quad (3)$$

which is then re-sampled to modify the time scale for global processing across segments:

$$\mathcal{Y}_b^{GA} = \text{US}_b \left( \text{SAN}_b \left( \text{DS}_b \left( \mathcal{X}_b^{GA} \right) \right) \right) \quad (4)$$

where  $\text{US}_b(\cdot)$  and  $\text{DS}_b(\cdot)$  are the upsampling and downsampling operations, respectively, which are defined as the follows:

$$\text{US}_b(\mathcal{X}) = \begin{cases} \text{ConvTrans1D}_K(\mathcal{X}; 4^b) & \text{if } b \leq N/2; \\ \text{ConvTrans1D}_K(\mathcal{X}; 4^{N-b-1}) & \text{if } b > N/2, \end{cases} \quad (5)$$

$$\text{DS}_b(\mathcal{X}) = \begin{cases} \text{Conv1D}_K(\mathcal{X}; 4^b) & \text{if } b \leq N/2; \\ \text{Conv1D}_K(\mathcal{X}; 4^{N-b-1}) & \text{if } b > N/2, \end{cases} \quad (6)$$

where  $\text{Conv1D}_A(\cdot; B)$  and  $\text{ConvTran1D}_A(\cdot; B)$  respectively denote the 1D and 1D transposed convolution operations along the axis of length  $A$  with a kernel size of  $B$  and a stride length of  $B$  such that the resultant length becomes  $\lfloor A/B \rfloor$  (in DS) or  $\lfloor AB \rfloor$  (in US) long. We also employ the variable-context-aware self-attentive network  $\text{SAN}_b(\cdot)$ , which is modified from the pioneering work [15]. For simplicity, we generally define our SAN for any input  $\mathcal{X} \in \mathbb{R}^{D \times S \times K}$ :

$$\text{SAN}(\mathcal{X}) = [\text{SelfAttn}(\text{LN}(\mathcal{X}[:, :, k]) + \mathbf{P}), k = 1, \dots, K], \quad (7)$$

where  $\mathbf{P}$  denotes the positional encoding matrix as introduced in [15], and  $\mathcal{X}[:, :, k] \in \mathbb{R}^{D \times S}$  refers to the inter-segment sequence. Here,  $\text{SelfAttn}(\cdot)$  is a typical multi-head self-attention function that linearly projects an input matrix  $\mathbf{X} \in \mathbb{R}^{D \times S}$  into three forms of matrices, commonly denoted as query  $\mathbf{Q}_j$ , key  $\mathbf{K}_j$ , and value  $\mathbf{V}_j$  matrices to compute the scaled dot-product attention for different heads  $j = 1, \dots, J$ , which are finally combined by a concatenation plus a matrix multiplication:

$$[\mathbf{Q}_j \ \mathbf{K}_j \ \mathbf{A}_j]^\top = [\mathbf{W}_j^Q \ \mathbf{W}_j^K \ \mathbf{W}_j^V]^\top \mathbf{X} + [\mathbf{b}_j^Q \ \mathbf{b}_j^K \ \mathbf{b}_j^V]^\top \quad (8)$$

$$\mathbf{A}_j = \text{Softmax} \left( \frac{\mathbf{Q}_j^\top \mathbf{K}_j}{\sqrt{D/J}} \right) \mathbf{V}_j \quad (9)$$

$$\mathbf{A} = \mathbf{W} \cdot \text{Concat}(\mathbf{A}_1, \dots, \mathbf{A}_J) \quad (10)$$

$$\text{SelfAttn}(\mathbf{X}) = \text{LN}(\mathbf{X} + \text{DROP}(\mathbf{A})) \quad (11)$$

where  $\text{DROP}(\cdot)$  denotes the dropout technique [20], and  $\mathbf{W} \in \mathbb{R}^{D \times D}$ ,  $\mathbf{W}_j^Q, \mathbf{W}_j^K, \mathbf{W}_j^V \in \mathbb{R}^{D/J \times D}$  and  $\mathbf{b}_j^Q, \mathbf{b}_j^K, \mathbf{b}_j^V \in \mathbb{R}^{D/J}$  are the parameters for SAN.

### 2.3.3. Residual Connections to Prevent Information Loss

One of the highlights in Sandglasset is to add residual connections between pairs of Sandglasset blocks that are of the same granularity. This technique is used to prevent information loss after passing through the middle blocks, where the granularity is on the coarsest scale. Mathematically, we define

$$\mathcal{X}_{b+1}^{LR} = \begin{cases} \mathcal{Y}_b^{GA} & \text{if } b \leq N/2; \\ \mathcal{Y}_b^{GA} + \mathcal{Y}_{b-N/2}^{GA} & \text{if } b > N/2, \end{cases} \quad (12)$$

which also defines the recurrence relation between the  $b$ -th and the  $(b+1)$ -th Sandglasset block. Our experimental result indicates that in practice adding residual connections is critical to remedy raw signal level details for improving signal reconstruction and to avoid gradient vanishing issues for better parameter learning.

A seminal work in signal processing – U-Net [21–23] seems a similar idea to ours for re-sampling and combining features at different time scales. Nonetheless, Sandglasset is very different in many aspects: (1) we have downsampling and upsampling operations together performed in one block; (2) our multi-granularity features are only processed by the SANs within each block; and (3) the residual connections across Sandglasset blocks are purely based on addition.

## 2.4. Merge Segments and Decoding

### 2.4.1. Mask Estimation

After passing through  $N$  Sandglasset blocks, we obtain a 3D tensor output  $\mathcal{X}_{N+1}^{LR} \in \mathbb{R}^{D \times K \times S}$ , which can be used to estimate masks for  $C$  sources. To do so, we first transform the last block's output using a PReLU-gated 2D convolutional layer to obtain a 4D tensor of shape  $C \times E \times K \times S$ :

$$\mathcal{Y} = \text{Conv2D} \left( \text{PReLU} \left( \mathcal{X}_{N+1}^{LR} \right); \mathbf{C} \right), \quad (13)$$

where  $\text{Conv2D}(\mathcal{Y}; \mathbf{C})$  denotes the 2D convolution operation applied on  $\mathcal{Y}$  parameterized by a learnable weight  $\mathbf{C} \in \mathbb{R}^{C \times E \times D}$  with a  $1 \times 1$  kernel,  $\text{PReLU}(\cdot)$  is the element-wise parametric ReLU. We then merge the output segments  $\mathcal{Y}$  using an  $\text{OverlapAdd}^1$  approach [8] to match the shape of the mixture frames  $\hat{\mathbf{X}} \in \mathbb{R}^{E \times L}$  for masking:

$$\mathbf{M} = \text{ReLU}(\text{OverlapAdd}(\mathcal{Y})), \quad (14)$$

where  $\odot$  is the element-wise product operation.

### 2.4.2. Decoder for Waveform Reconstruction

Finally, the  $c$ -th source signal is reconstructed by applying the  $c$ -th estimated mask to the initially computed mixture frames  $\hat{\mathbf{X}}$  and then using  $\text{OverlapAdd}$  to merge frames into waveform:

$$\hat{s}_c = \text{OverlapAdd}(\hat{\mathbf{X}} \odot \mathbf{M}_c). \quad (15)$$

Last but not least, given  $C$  estimated sources, the scale-invariant source-to-noise ratio (SI-SNR) loss [6] is used with u-PIT [9] to learn the network parameters and to solve the permutation problem.

<sup>1</sup>[https://github.com/tensorflow/tensorflow/blob/r1.12/tensorflow/contrib/signal/python/ops/reconstruction\\_ops.py](https://github.com/tensorflow/tensorflow/blob/r1.12/tensorflow/contrib/signal/python/ops/reconstruction_ops.py)

### 3. EXPERIMENTS

#### 3.1. Experimental Setup

##### 3.1.1. Data

To compare with the SOTA speech separation networks, we used two benchmark datasets for evaluation – WSJ0-2mix and WSJ0-3mix [24], which are generated from the Wall Street Journal (WSJ0) [25] dataset by randomly mixing clean utterances from different speakers at a sampling rate of 8 kHz with SNRs between 0 dB and 5 dB. The separation datasets consist of 30 hours of training, 10 hours of validation, and 5 hours of test data from 16 unseen speakers. Both WSJ0-2mix and WSJ0-3mix have been widely used as the benchmark in single-channel speech separation [6–8, 26–29].

##### 3.1.2. Implementation Details

In our implementation, we used the setting of encoder-decoder modules in [6, 7] and the segmentation module described in [8]. In particular, we set  $M = 4$ ,  $E = 256$ , and  $D = 128$ . For Sandglassset, we used 6 Sandglassset blocks, i.e.,  $N = 6$ . In the first Sandglassset block, we used an initial segment size  $K = 256$ , which would be shortened/prolonged by a factor of 4 in the first/last three blocks, as described in Eq. (5-6). Within each Sandglassset block, we used local Bi-LSTM with 128 hidden units, i.e.,  $H = 128$ . The global SAN was set to be 8-head, i.e.,  $J = 8$  with a 0.1 dropout rate. For training, we used Adam [30] optimizer with an initial learning rate of 0.001 and a decaying rate of 0.98. The optimization was stopped if no lower validation loss was obtained for 10 consecutive epochs.

##### 3.1.3. Mixing Same-Speaker Utterances as Post Training

By inspecting the poor separation cases of Sandglassset in WSJ0-2mix, we found that those mixture inputs shared a common characteristic that both speakers have a similar voice timbre, so that the model may keep both voices in the two output signals. This reveals that our model’s separation is highly dependent on the voice timbre of different speakers. We conceived that one of the main reasons is that mixture inputs with similar voice timbres are rare in the training set, which makes it hard for our model to learn to differentiate those similar voices. To alleviate this problem, we designed a simple, easy-to-implement post training method for Sandglassset. In particular, after the convergence of the normal training, we expanded the training set by adding dynamically mixed utterances from the same speaker in a 1:1 ratio in sample size relative to the original training data.

#### 3.2. Performance Comparisons

The SISNRi and SDRi performances of Sandglassset in WSJ0-2mix are reported in Table 1. First of all, for an ablation study on our proposed multi-granularity (MG) strategy, we trained an ablated baseline system – “Sandglassset (SG)”, in which each Sandglassset block uses a single-granularity strategy with a fixed segment size ( $K = 256$ ). Comparing “Sandglassset (SG)” to “Sandglassset (MG)”, we can see a significant drop in SI-SNRi and SDRi scores if Sandglassset was deprived of the multi-granularity mechanism. This asserts our initial expectation that multi-granularity can better exploit SANs for modeling multi-level contexts. For another ablation study, we trained a Sandglassset without residual connections, denoted by “Sandglassset (w/o RES)”, which produced a much-degraded performance. We also found that by using the simple post training strategy

**Table 1:** Comparison of performances on the WSJ0-2mix test set. The models that exploit speaker IDs as additional information for training and testing are marked with “+ Spk ID”. † denotes our estimated model size based on the authors’ description.

Model	Params.	SI-SNRi	SDRi
BLSTM-TasNet [6]	23.6M	13.2	13.6
Conv-TasNet [7]	8.8M	15.3	15.6
Conv-TasNet + MBT [29]	8.8M	15.5	15.9
FurcaNeXt [28]	51.4M	18.4	-
DPRNN [8]	2.6M	18.8	19.1
DPTNet [12]	2.7M	20.2	20.6
<b>Sandglassset (w/o RES)</b>	<b>2.3M</b>	20.1	20.3
<b>Sandglassset (SG)</b>	<b>2.3M</b>	20.3	20.5
<b>Sandglassset (MG)</b>	<b>2.3M</b>	20.8	21.0
<b>Sandglassset (MG) + PT</b>	<b>2.3M</b>	<b>21.0</b>	<b>21.2</b>
Gated DPRNN + Spk ID [13]	7.5M	20.1	-
Wavesplit + Spk ID [14]	†42.5M	<b>21.0</b>	<b>21.2</b>

**Table 2:** Comparison of performances on the WSJ0-3mix test set.

Model	Params.	SI-SNRi	SDRi
Conv-TasNet [7]	8.8M	12.7	13.1
DPRNN [8]	2.6M	14.7	-
<b>Sandglassset (MG)</b>	<b>2.3M</b>	<b>17.1</b>	<b>17.4</b>
Gated DPRNN + Spk ID [13]	7.5M	16.7	-
Wavesplit + Spk ID [14]	†42.5M	<b>17.3</b>	<b>17.6</b>

described in Section 3.1.3, the performance of Sandglassset can be further improved.

Overall, the proposed Sandglassset has achieved the best separation performance with parameters as few as 2.3M, which is the lightest model size that is ever reported for the SS tasks. We would like to emphasize that, to focus on studying the advantage of the network architecture only, we purposely avoid using any speaker information to help further increase the scores of Sandglassset, unlike what has been done in the two most recent systems “Gated DPRNN + Spk ID” [13] and “Wavesplit + Spk ID” [14]. Comparing to the strongest reference model regardless of speaker information, Sandglassset has attained an absolute improvement of 0.8 dB SI-SNRi. The WSJ0-3mix result of Sandglassset, as shown in Table 2, also consistently shows an absolute improvement of 2.4 dB SI-SNRi over the best reference model with no speaker information.

#### 3.3. Computational Cost Analysis

Moreover, thanks to some coarser-scale global processing, another merit of Sandglassset is a significant reduction in computational cost, relative to a model that is comparable in size – DPRNN. In Table 3, we reported the runtime memory and the floating-point operations (FLOPs)<sup>2</sup> which indicates the model efficiency for processing each second of mixture input. Finally, compared to the best performing DPRNN (i.e., 2-sample window), Sandglassset consumed 58.4% less memory and 66.0% fewer FLOPs.

### 4. CONCLUSIONS

This paper proposes a novel sandglass-shape network for time-domain single-channel speech separation, namely Sandglassset. This

<sup>2</sup><https://github.com/sovrasov/flops-counter.pytorch>

**Table 3:** Comparison of computational costs.

Model	Params.	Memory (GB)	GFLOPs ( $10^9$ )
DPRNN [8]	2.6M	1.97	84.7
<b>Sandglassnet</b>	<b>2.3M</b>	<b>0.82</b> ( $\downarrow$ 58.4%)	<b>28.8</b> ( $\downarrow$ 66.0%)

advanced network architecture combines the advantages of the self-attention networks and the proposed multi-granularity mechanism to hierarchically and progressively model high-level, large-granularity contexts and low-level, fine-granularity details. In our experiment, Sandglassnet achieved state-of-the-art results on two benchmark datasets, especially, with the lightest model size that has ever been reported for SS tasks. Comparing to the previous smallest and strongest model in literature, our proposed model is also very light in terms of memory (58.4% less) and computations (66% fewer), which suggests Sandglassnet a more economical and practical model for industrial deployment.

## 5. REFERENCES

- [1] E Colin Cherry, “Some experiments on the recognition of speech, with one and with two ears,” *The Journal of the acoustical society of America*, vol. 25, no. 5, pp. 975–979, 1953.
- [2] Simon Haykin and Zhe Chen, “The cocktail party problem,” *Neural computation*, vol. 17, no. 9, pp. 1875–1902, 2005.
- [3] Arun Narayanan and DeLiang Wang, “Investigation of speech separation as a front-end for noise robust speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 826–835, 2014.
- [4] Max WY Lam, Jun Wang, Xunying Liu, Helen Meng, Dan Su, and Dong Yu, “Extract, adapt and recognize: an end-to-end neural network for corrupted monaural speech recognition,” *Proc. INTERSPEECH*, pp. 2778–2782, 2019.
- [5] Thilo von Neumann, Keisuke Kinoshita, Lukas Drude, Christoph Boeddeker, Marc Delcroix, Tomohiro Nakatani, and Reinhold Haeb-Umbach, “End-to-end training of time domain audio separation and recognition,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7004–7008.
- [6] Yi Luo and Nima Mesgarani, “Tasnet: time-domain audio separation network for real-time, single-channel speech separation,” in *Proc. ICASSP*. IEEE, 2018, pp. 696–700.
- [7] 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*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [8] Yi Luo, Zhuo Chen, and Takuya Yoshioka, “Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation,” *arXiv preprint arXiv:1910.06379*, 2019.
- [9] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen, “Permutation invariant training of deep models for speaker-independent multi-talker speech separation,” in *Proc. ICASSP*. IEEE, 2017, pp. 241–245.
- [10] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, Jesper Jensen, Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen, “Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks,” *TASLP*, vol. 25, no. 10, pp. 1901–1913, 2017.
- [11] Shaojie Bai, J Zico Kolter, and Vladlen Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271*, 2018.
- [12] Jingjing Chen, Qirong Mao, and Dong Liu, “Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation,” *arXiv preprint arXiv:2007.13975*, 2020.
- [13] Eliya Nachmani, Yossi Adi, and Lior Wolf, “Voice separation with an unknown number of multiple speakers,” *arXiv preprint arXiv:2003.01531*, 2020.
- [14] Neil Zeghidour and David Grangier, “Wavesplit: End-to-end speech separation by speaker clustering,” *arXiv preprint arXiv:2002.08933v1*, 2020.
- [15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [16] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang, “Disan: Directional self-attention network for rnn/cnn-free language understanding,” *arXiv preprint arXiv:1709.04696*, 2017.
- [17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [18] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov, “Transformer-xl: Attention language models beyond a fixed-length context,” *arXiv preprint arXiv:1901.02860*, 2019.
- [19] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena, “Self-attention generative adversarial networks,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 7354–7363.
- [20] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [22] Daniel Stoller, Sebastian Ewert, and Simon Dixon, “Wave-u-net: A multi-scale neural network for end-to-end audio source separation,” in *19th International Society for Music Information Retrieval Conference, ISMIR*, 2018.
- [23] Hyeon-Seok Choi, Hoon Heo, Jie Hwan Lee, and Kyogu Lee, “Phase-aware single-stage speech denoising and dereverberation with u-net,” *arXiv preprint arXiv:2006.00687*, 2020.
- [24] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” in *Proc. ICASSP*. IEEE, 2016, pp. 31–35.
- [25] J Garofalo, D David Graff, D Paul, and D Pallett, “Continuous speech recognition (csr-i) wall street journal (wsj0) news, complete. linguistic data consortium, philadelphia (1993),” .

- [26] Yuzhou Liu and DeLiang Wang, “Divide and conquer: A deep casa approach to talker-independent monaural speaker separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2092–2102, 2019.
- [27] Zhong-Qiu Wang, Jonathan Le Roux, DeLiang Wang, and John R Hershey, “End-to-end speech separation with unfolded iterative phase reconstruction,” *arXiv preprint arXiv:1804.10204*, 2018.
- [28] Liwen Zhang, Ziqiang Shi, Jiqing Han, Anyan Shi, and Ding Ma, “Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks,” in *International Conference on Multimedia Modeling*. Springer, 2020, pp. 653–665.
- [29] Max WY Lam, Jun Wang, Dan Su, and Dong Yu, “Mixup-breakdown: a consistency training method for improving generalization of speech separation models,” *Proc. ICASSP*, 2020.
- [30] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.