Supplementary Materials for "Speech Separation Using an Asynchronous Fully Recurrent Convolutional Neural Network"

Xiaolin Hu[1*], **Kai Li**[1], **Weiyi Zhang**[1], **Yi Luo**[2], **Jean-Marie Lemercier**[3], **Timo Gerkmann**[3]
[1]Department of Computer Science and Technology,
Tsinghua Laboratory of Brain and Intelligence (THBI),
IDG/McGovern Institute of Brain Research Tsinghua University, Beijing, China
[2]Department of Electrical Engineering, Columbia University, NY, USA
[3]Department of Informatics, University of Hamburg, Hamburg, Germany
xlhu@tsinghua.edu.cn, {lk21,wy-zhang19}@mails.tsinghua.edu.cn,
y.luo@columbia.edu, lemercier@informatik.uni-hamburg.de,
timo.gerkmann@uni-hamburg.de

1 Training Method and Evaluation Metrics

When the FRCNN is unfolded through time with any scheme, it becomes a feedforward model and we use the standard BP algorithm to train it. The object is to maximize the scale-invariant signal-to-noise ratio (SI-SNR) [5]. SI-SNR for each speaker is defined as

$$\text{SI-SNR} = 10 \times \log_{10} \left(\frac{\|\mathbf{A}_{\text{target}}\|_2^2}{\|\mathbf{e}_{\text{noise}}\|_2^2} \right), \tag{1}$$

where

$$\mathbf{A}_{\text{target}} = \frac{\langle \widehat{\mathbf{s}}, \mathbf{s} \rangle \, \mathbf{s}}{\|\mathbf{s}\|_2^2}, \quad \mathbf{e}_{\text{noise}} = \widehat{\mathbf{s}} - \mathbf{A}_{\text{target}}. \tag{2}$$

In above equations, $\mathbf{s} \in R^{1\times T}$ denotes the original single-source speech, $\widehat{\mathbf{s}} \in R^{1\times T}$ denotes the estimated speech, $\langle \cdot, \cdot \rangle$ denotes inner product, and $\|\cdot\|_2^2$ denotes l_2-norm.

It is worth noting that the estimated speech signal and the real clean speech signal do not necessarily have the same speaker arrangement order. This is called the *label permutation problem* and we use the utterance-level label permutation invariant training (uPIT) [3] to solve this problem.

We used the scale-invariant signal-to-noise ratio improvement (SI-SNRi) [5] and signal-to-distortion ratio improvement (SDRi) [10] as the evaluation metrics to measure the speech separation accuracies of models. They are calculated as follows:

$$\text{SI-SNRi} = \text{SI-SNR}(\widehat{\mathbf{s}}, \mathbf{s}) - \text{SI-SNR}(\mathbf{s}, \mathbf{x}), \tag{3}$$

$$\text{SDRi} = \text{SDR}(\widehat{\mathbf{s}}, \mathbf{s}) - \text{SDR}(\mathbf{s}, \mathbf{x}), \tag{4}$$

where $\mathbf{x}$ donates the mixture of different speaker audio.

2 Sample Results on Libri2Mix

To give readers an impression of the separation results of different models, we randomly selected some sample audio files from the Libri2Mix [1] datasets and put in the *sample results* folder. The

*Corresponding author.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

models presented include DPCL [2], uPIT [3], Conv-TasNet [7], SuDoRMRF 1.0x [9], DualPathRNN [6] and A-FRCNN-16. All separated voices were saved in a folder named by the model name. Since all mixtures of speeches were from two speakers, each folder has two subfolders, s1 and s2. We have created an offline web page for these samples *index.html*. Please open it with your explorer. In most examples we found that separated speeches by A-FRCNN-16 sounded clearer than those by other methods.

3 Transfer to DualPathRNN and Sandglasset

We were interested in whether the MSF method in A-FRCNN is transferable to other architectures. We chose DualPathRNN [6] and Sandglasset [4] as the baseline models. Both of them use intra-chunk and inter-chunk operations repeatedly to fuse information.

The DualPathRNN has multiple blocks with untied weights. In each block, there is an RNN (called *intra-chunk RNN*) for processing multiple audio segments independently, and another RNN (called *inter-chunk RNN*) for aggregating the outputs of the intra-chunk RNN with all audio segments as inputs. The architecture is shown in Figure S1a.

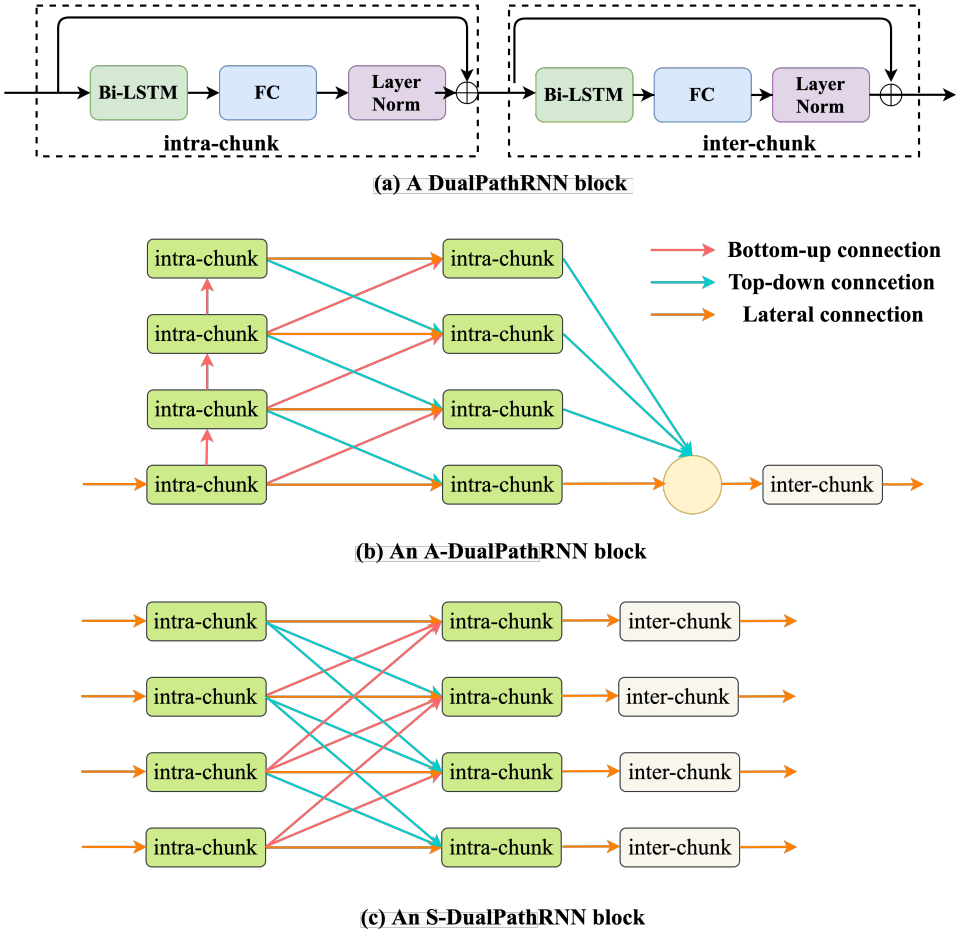

Figure S1: Structures of a DualPathRNN block (a), an A-DualPathRNN block (b) and an S-DualPathRNN block (c).

We make S copies of the intra-chunk RNN which share weights and let them process the input in different scales sequentially, and then use the MSF method in A-FRCNN to fuse the outputs of these intra-chunk RNNs. See Figure S1b. The inter-chunk RNN is not altered. These constitute a new block. To save the number of parameters, we use tied weights in different blocks. We call this model *Asynchronous DualPathRNN* or *A-DualPathRNN*. For comparison, we design a new model

called *Synchronous DualPathRNN* or *S-DualPathRNN* in the same way but adopt the MSF method in S-FRCNN to fuse the outputs of the intra-chunk RNNs. See Figure S1c.

In this experiment, we set the number of stages $S = 5$ again. As mentioned in the main text, training the DualPathRNN is time-consuming. The training time depends on the kernel size and stride in the encoder and decoder. The best result was achieved by setting the kernel size and stride to 2 and 1, respectively, a configuration used to report the results in Table 3 in the main paper. It took about nine days on our 8-GPU server to train the model on the WSJ0-2Mix dataset. To save the computational cost, we used kernel size 18 and stride 8 in the encoder and decoder to explore the combination of the DualPathRNN and A-FRCNN. The results of the DualPathRNN on WSJ0-2Mix with this setting were also reported in the original paper of the DualPathRNN [6].

Table S1 shows the results. The S-DualPathRNN obtained a little bit higher SI-SNRi values than the DualPathRNN on the WSJ0-2Mix and Libri2Mix datasets, while the A-DualPathRNN obtained much higher SI-SNRi values than the DualPathRNN. It indicates that the MSF method in A-FRCNN can improve the performance of the DualPathRNN.

Sandglasset has a similar structure to DualPathRNN, and it also adopts the intra- and inter-chunk block design. The major difference is that in Sandglasset, the bidirectional LSTM (or BiLSTM) in the inter-chunk is changed to a transformer. This modification improved SI-SNRi on WSJ0-2Mix dataset by 1.5 dB. We can transfer our method to Sandglasset in the same way as A-DualPathRNN. The resulting model is called A-Sandglasset and its structure is the same as A-DualPathRNN as illustrated in Figure S1b, and the only difference is that in the inter-chunk the BiLSTM is changed to a transformer. The original Sandglasset is too large to fit in our computing facility. We thus used a smaller version and applied our connection scheme. The Sandglasset hyperparameters are basically the same as those of DualPathRNN. Exceptionally, for the encoder/decoder module we set their kernel size to 16 and the number of channels to 256. The initial segment size was 64. The Sandglasset block input dimension was set to 128. The hidden layer dimension of BiLSTM and transformer were set to 128. The global Self-Attentive Network was set to be 8-head with a 0.1 dropout rate. The A-Sandglasset obtained 0.6 dB higher SI-SNRi value than the plain Sandglasset on the WSJ0-2Mix dataset. See Table S1.

Table S1: The performance of DualPathRNN/Sandglasset and their variants on the WSJ0-2Mix and Libri2mix test sets. The 2nd and 3rd columns are SI-SNRi values. '*' indicates that the SI-SNRi value was obtained by using the asteroid toolkit [8].

Model	WSJ0-2Mix	Libri2Mix	Parameters
DualPathRNN	15.9	13.9*	3.7M
S-DualPathRNN	16.1	14.0	3.5M
A-DualPathRNN	**17.5**	**14.5**	2.9M
Sandglasset	17.3	14.4	2.3M
A-Sandglasset	**17.9**	**14.9**	1.3M

4 Training Time of Different Models

We presented the training time of several typical speech separation models on Libri2Mix in Table S2. It is seen that DualPathRNN required significantly more training time than other models.

References

[1] 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.

[2] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *ICASSP*, pages 31–35, 2016.

Table S2: The training time of typical models on the Libri2mix dataset.

Model	Training Time (h)
Conv-TasNet	14.16
SuDoRM-RF 0.25x	12.50
SuDoRM-RF 0.5x	14.20
SuDoRM-RF 1.0x	19.65
DualPathRNN	142.51
A-FRCNN-4	14.77
A-FRCNN-8	22.79
A-FRCNN-16	39.44
A-FRCNN-4 (sum)	13.31
A-FRCNN-8 (sum)	20.54
A-FRCNN-16 (sum)	38.33

[3] 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*, 25(10):1901–1913, 2017.

[4] Max WY Lam, Jun Wang, Dan Su, and Dong Yu. Sandglasset: A light multi-granularity self-attentive network for time-domain speech separation. In *ICASSP*, pages 5759–5763. IEEE, 2021.

[5] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. SDR–half-baked or well done? In *ICASSP*, pages 626–630. IEEE, 2019.

[6] Yi Luo, Zhuo Chen, and Takuya Yoshioka. Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP*, pages 46–50, 2020.

[7] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *TASLP*, 27(8):1256–1266, 2019.

[8] 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 *Interspeech*, 2020.

[9] 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)*, pages 1–6, 2020.

[10] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *TASLP*, 14(4):1462–1469, 2006.