Gated Residual Networks with Dilated Convolutions for Supervised Speech Separation

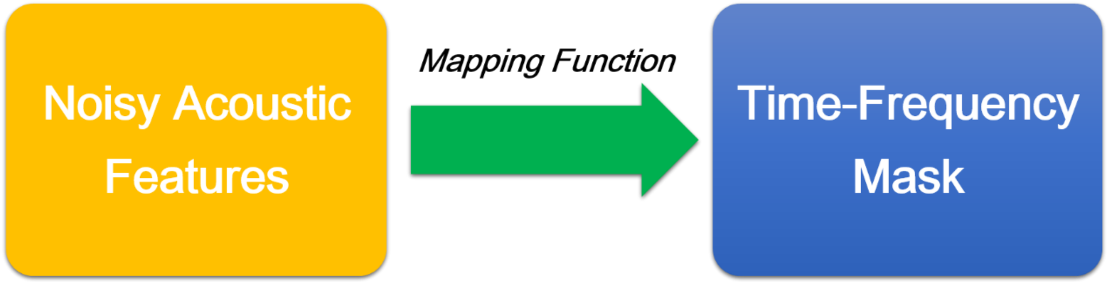
Ke Tan, Jitong Chen and DeLiang Wang

Perception and Neurodynamics Lab (PNL)

The Ohio State University

- Speech separation is the task of separating target speech from its background interference (background noise, interfering speech, or room reverberation).

- Speech separation can be treated as a supervised learning problem, where a mapping from noisy acoustic features to a time-frequency (T-F) mask is learned.



- For supervised speech separation, **contextual information** can effectively facilitate mask estimation. Typically, a window of consecutive time frames is used to provide temporal contexts at each time frame.

- However, contextual information is utilized inadequately given a **fixed-length context window**. A recent approach developed by Chen *et al.* [1] utilizes long-term contexts by treating supervised speech separation as a **sequence-to-sequence mapping**.

- In [1], a 4-layer long short-term memory (LSTM) based model was proposed to deal with **speaker- and noise-independent** speech separation. With a large number of training speakers, the LSTM based model significantly outperforms a deep neural network (DNN) based model.

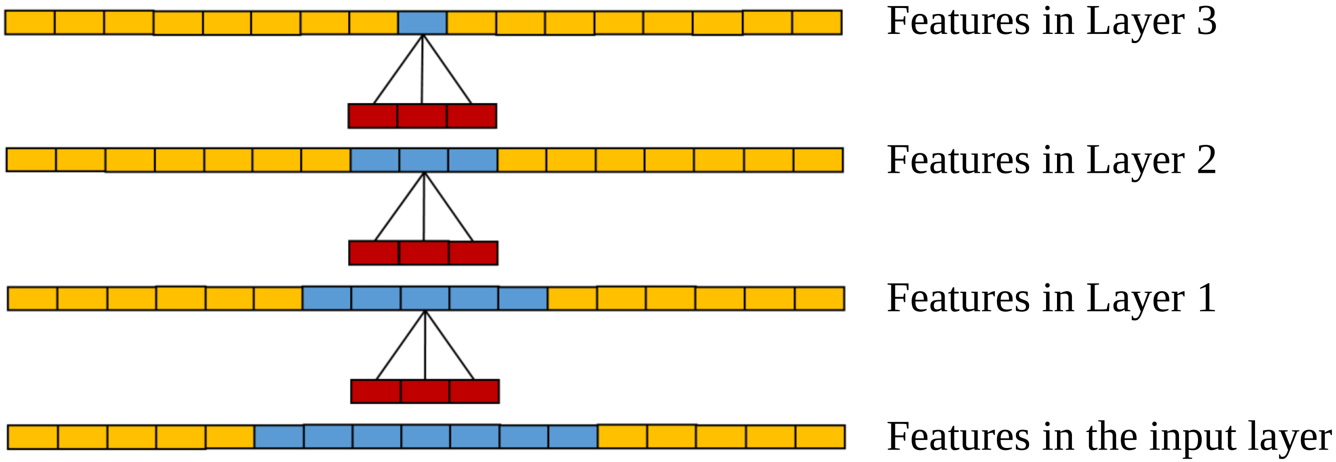
[1] J. Chen and D. L. Wang, "Long short-term memory for speaker generalization in supervised speech separation," The Journal of the Acoustical Society of America, vol. 141, no. 6, pp. 4705–4714, 2017

Motivations

- Motivated by recent study on **dilated convolutions** for **context aggregation** in computer vision, we propose a novel network with dilated convolutions to deal with speaker- and noise-independent speech separation.

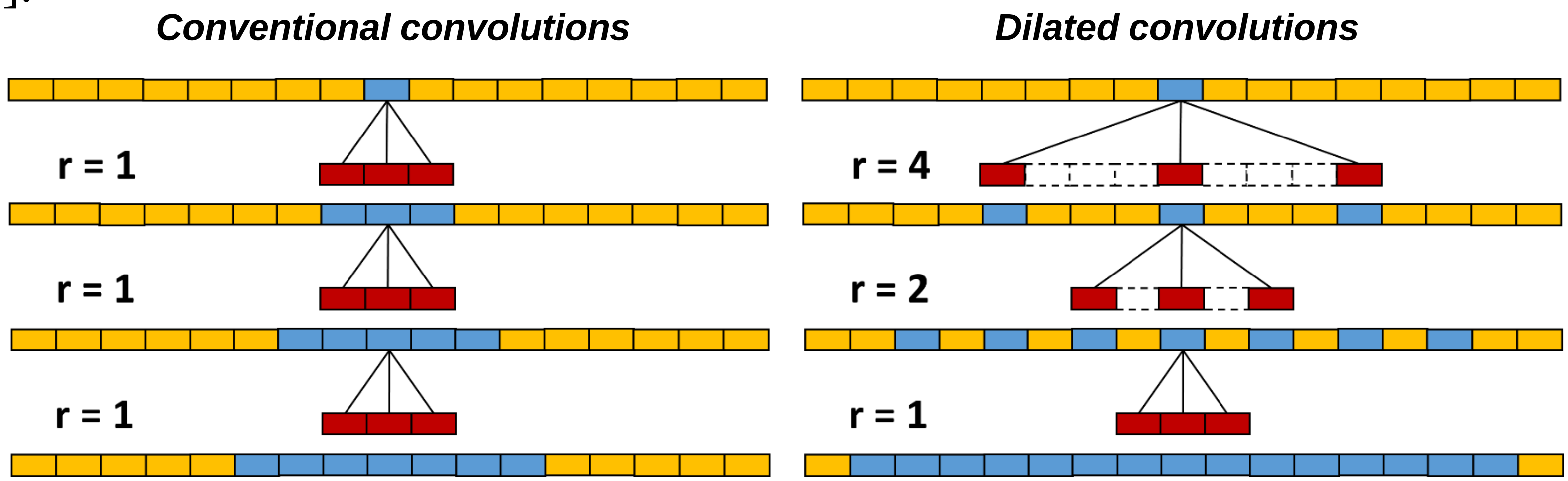
- As in [1], speech separation is treated as a sequence-to-sequence mapping in this study.

- In convolutional neural networks (CNNs), contextual information is augmented essentially through the expansion of the **receptive fields**. A receptive field is a region in the input space that affects a particular high-level feature.



- Traditionally, there are two ways to achieve this goal:
  (1) to increase the network depth ➡ vanishing gradient problem
  (2) to enlarge the kernel size.

- **Dilated convolutions** were first proposed for multi-scale context aggregation in [2].



$$R_i = 2i - 1$$

$$R_i = 2^i - 1$$

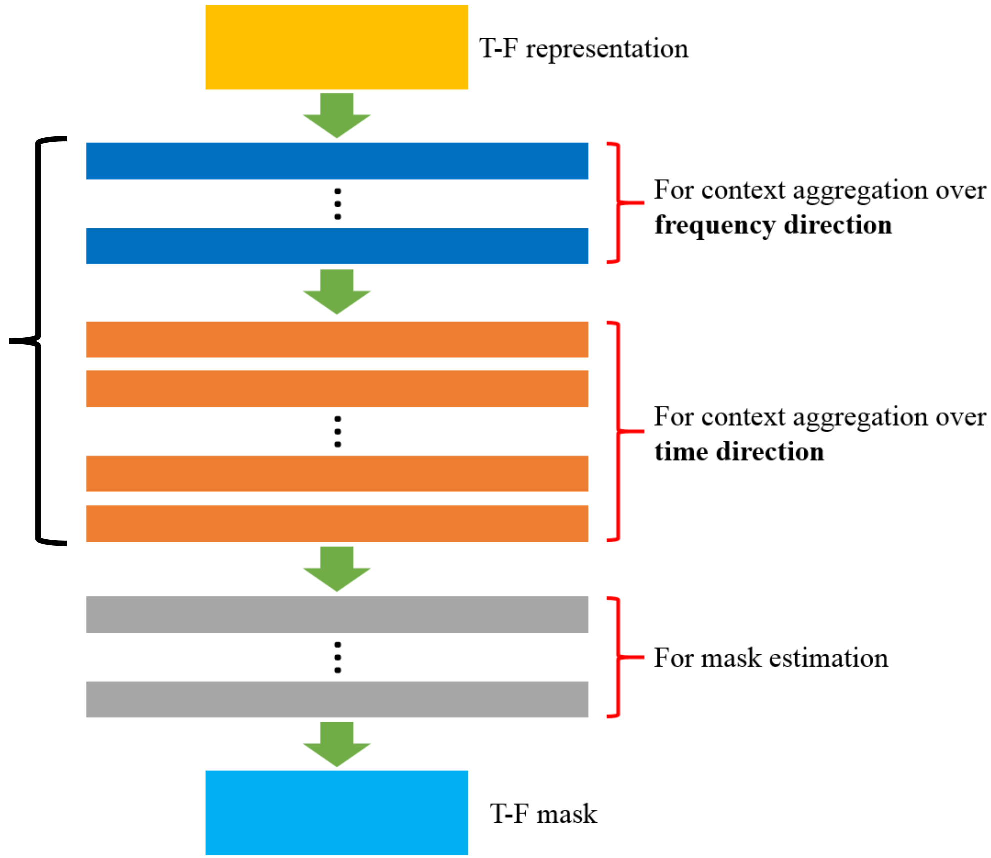
- Note that r is a factor called **dilation rate**.

[2] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in International Conference on Learning Representations (ICLR), 2016.

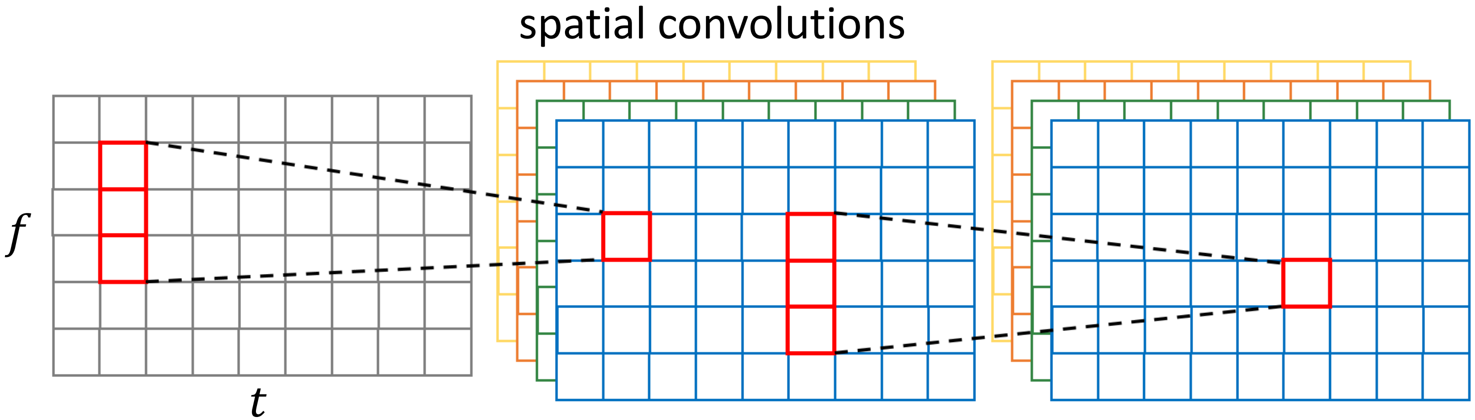
Dilated Convolutions

- For the T-F representation of an utterance, the number of time frames, T, and the number of frequency channels, F, are typically imbalanced ($T > F$). So we may need many convolutional layers to aggregate contexts over time, but we do not need that many layers to aggregate contexts over frequency.

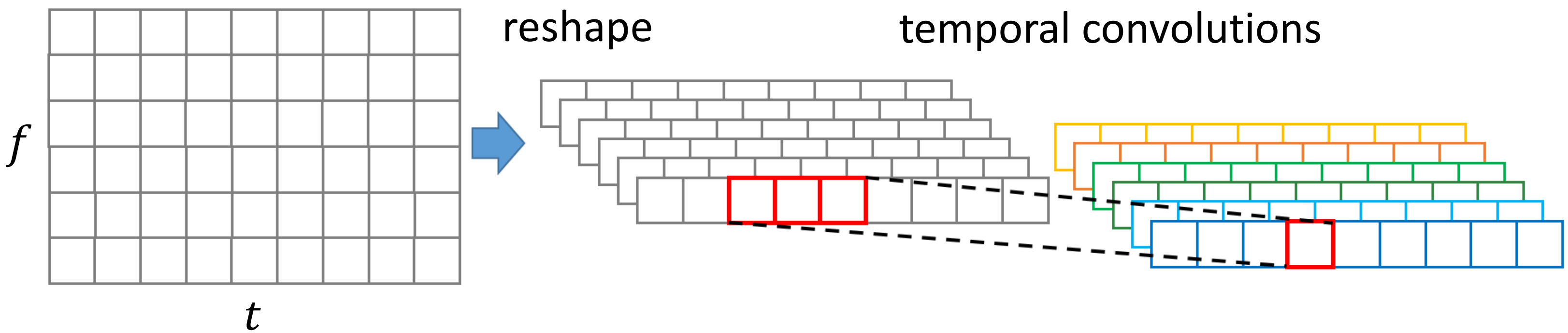
Aggregate contexts over time and frequency, **separately**



- To capture contexts in the **frequency direction**, we use **2-D convolutions** (or **spatial convolutions**) on the T-F representation of speech:



- To capture contexts in the **time direction**, we use **1-D convolutions** (or **temporal convolutions**) on the T-F representation of speech:



- **Time-dilated convolutions:**

Time-dilated convolutions were first developed in [3] for speech recognition by using an asymmetric version of spatial dilated convolution with dilation in the time direction but not in the frequency direction. In this study, we use the ***1-D version of time-dilated convolutions***, where dilation is applied to ***temporal convolutions***.

- **Frequency-dilated convolutions:**

To aggregate contextual information over the frequency dimension, we apply dilation to the aforementioned ***spatial convolutions***, where the kernels of size 3×1 are placed along the frequency direction. For convenience, we appropriately call them frequency-dilated convolutions.

[3] T. Sercu and V. Goel, "Dense prediction on sequences with time-dilated convolutions for speech recognition," NIPS End-to-end Learning for Speech and Audio Processing Workshop, 2016.

Gated Linear Units

- Gating mechanisms allow for modeling more complex interactions by controlling the information flow. LSTM-style gating mechanisms are applied to convolutions in [4]:

$$\begin{aligned}\mathbf{y} &= \tanh(\mathbf{x} * \mathbf{W}_1 + \mathbf{b}_1) \odot \sigma(\mathbf{x} * \mathbf{W}_2 + \mathbf{b}_2) \\ &= \tanh(\mathbf{v}_1) \odot \sigma(\mathbf{v}_2)\end{aligned}$$

where $\mathbf{v}_1 = \mathbf{x} * \mathbf{W}_1 + \mathbf{b}_1$ and $\mathbf{v}_2 = \mathbf{x} * \mathbf{W}_2 + \mathbf{b}_2$. Convolution operation and element-wise multiplication are denoted by $*$ and $\odot$, respectively. $\mathbf{W}$'s and $\mathbf{b}$'s represent kernels and biases, respectively. σ denotes the sigmoid function.

[4] A. Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, and K. Kavukcuoglu, "Conditional image generation with pixelcnn decoders," in Advances in Neural Information Processing Systems, 2016, pp. 4790–4798.

- The gradient of the LSTM-style gating is:

$$\nabla[\tanh(\mathbf{v}_1)\odot\sigma(\mathbf{v}_2)] = \tanh'(\mathbf{v}_1)\nabla\mathbf{v}_1\odot\sigma(\mathbf{v}_2) + \sigma'(\mathbf{v}_2)\nabla\mathbf{v}_2\odot\tanh(\mathbf{v}_1)$$

where $\tanh'(\mathbf{v}_1), \sigma'(\mathbf{v}_2) \in (0,1)$.

- Typically, the vanishing gradient problem arises as the network depth increases. The downscaling factors $\tanh'(\mathbf{v}_1)$ and $\sigma'(\mathbf{v}_2)$ could make it worse. To alleviate this problem, gated linear units (GLUs) were developed in [5]:

$$\begin{aligned}\mathbf{y} &= (\mathbf{x} * \mathbf{W}_1 + \mathbf{b}_1)\odot\sigma(\mathbf{x} * \mathbf{W}_2 + \mathbf{b}_2) \\ &= \mathbf{v}_1\odot\sigma(\mathbf{v}_2)\end{aligned}$$

The gradient of GLUs

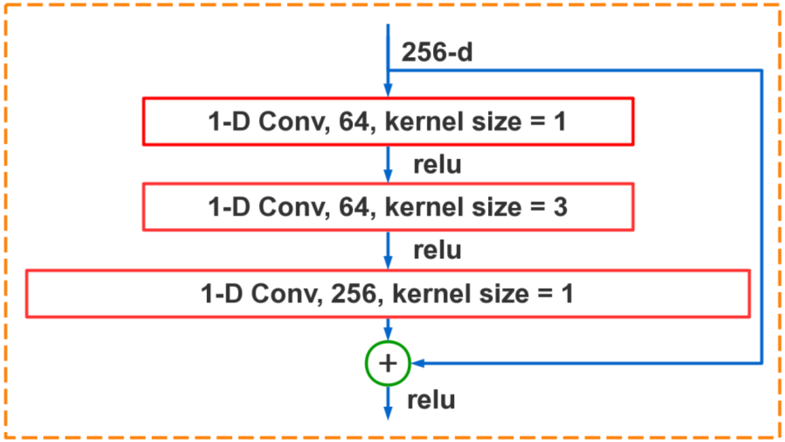
$$\nabla[\mathbf{v}_1\odot\sigma(\mathbf{v}_2)] = \nabla\mathbf{v}_1\odot\sigma(\mathbf{v}_2) + \sigma'(\mathbf{v}_2)\nabla\mathbf{v}_2\odot\mathbf{v}_1$$

includes a path $\nabla\mathbf{v}_1\odot\sigma(\mathbf{v}_2)$ without downscaling, allowing for the gradient flow through layers.

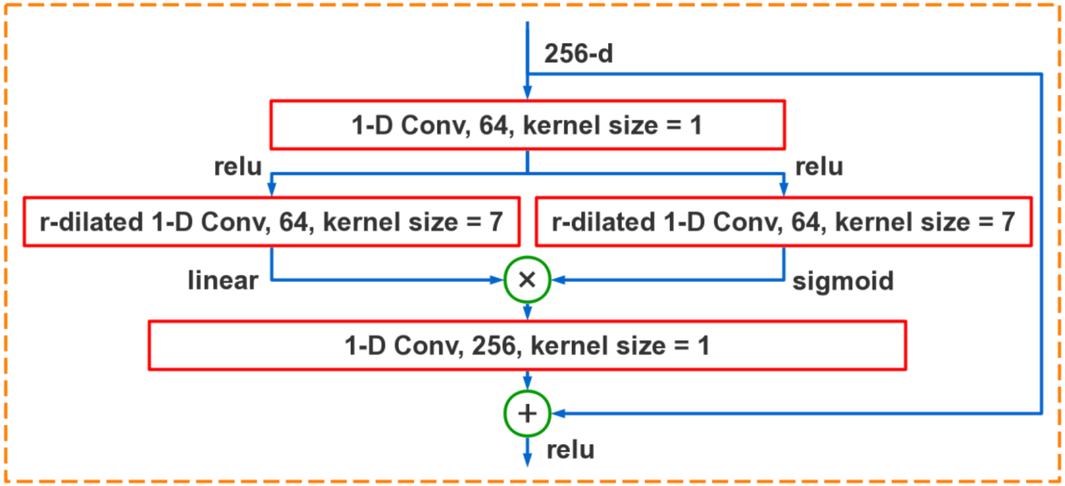
[5] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in Proceedings of the 34th International Conference on Machine Learning, 2017, vol. 70, pp. 933–941.

- We develop a novel residual block by incorporating time-dilated convolutions and GLUs into the commonly-used bottleneck block.

a commonly-used residual block



our proposed residual block



Network Architecture

- A detailed description of our proposed network architecture is as follows:

layer name	input size	layer hyperparameters	output size
expand_dims	$T \times 161$	-	$1 \times T \times 161$
conv2d_1	$1 \times T \times 161$	$1 \times 3, (1, 1), 16$	$16 \times T \times 159$
conv2d_2	$16 \times T \times 159$	$1 \times 3, (1, 1), 16$	$16 \times T \times 157$
conv2d_3	$16 \times T \times 157$	$1 \times 3, (1, 2), 32$	$32 \times T \times 153$
conv2d_4	$32 \times T \times 153$	$1 \times 3, (1, 4), 32$	$32 \times T \times 145$
reshape	$32 \times T \times 145$	-	$T \times 4640$
conv1d_1	$T \times 4640$	$1, 1, 128$	$T \times 128$
conv1d_2	$T \times 64$	$\left.\begin{matrix} \begin{pmatrix} 1, 1, 64 \\ 7, \underline{\mathbf{1}}, 64 \\ 1, 1, 256 \end{pmatrix} \\ \begin{pmatrix} 1, 1, 64 \\ 7, \underline{\mathbf{2}}, 64 \\ 1, 1, 256 \end{pmatrix} \\ \begin{pmatrix} 1, 1, 64 \\ 7, \underline{\mathbf{4}}, 64 \\ 1, 1, 256 \end{pmatrix} \\ \begin{pmatrix} 1, 1, 64 \\ 7, \underline{\mathbf{8}}, 64 \\ 1, 1, 256 \end{pmatrix} \\ \begin{pmatrix} 1, 1, 64 \\ 7, \underline{\mathbf{16}}, 64 \\ 1, 1, 256 \end{pmatrix} \\ \begin{pmatrix} 1, 1, 64 \\ 7, \underline{\mathbf{32}}, 64 \\ 1, 1, 256 \end{pmatrix} \end{matrix}\right\} \times 3$	$T \times 256$
conv1d_3	$T \times 256$	$1, 1, 256$	$T \times 256$
conv1d_4	$T \times 256$	$1, 1, 128$	$T \times 128$
conv1d_5	$T \times 128$	$1, 1, 161$	$T \times 161$

Formats:

Input and output sizes for 2-D convolutions:
featureMaps* × *timeSteps* × *frequencyChannels

Input and output sizes for 1-D convolutions:
timeSteps* × *featureMaps

Layer hyperparameters:
(kernelSize, dilationRate, outputChannels)

- Dataset: WSJ0 SI-84, including 7138 utterances from 83 speakers. Of the 83 speakers, 6 speakers (3 males and 3 females) are treated as untrained speakers. The models are trained with the remaining 77 speakers.

- (1) Training noises: 10,000 noises from a sound effect library (available at https://www.sound-ideas.com). (2) Test noises: two highly nonstationary noises (babble and cafeteria) from the Auditec CD (available at http://www.auditec.com)

- To create a training mixture, we mix a randomly drawn training utterance with a random cut from the 10,000 training noises at an SNR randomly chosen from {-5, -4, -3, -2, -1, 0} dB. We create 320,000 mixtures for training.

- We use -5 dB and -2 dB for test set. For each SNR, two test sets are created:
  - Test Set 1: 150 mixtures are created from 25×6 utterances of 6 trained speakers (3 males and 3 females).
  - Test Set 2: 150 mixtures are created from 25×6 utterances of 6 untrained speakers (3 males and 3 females).

- In this study, we use the phase-sensitive mask (PSM) [6] as the training target:

$$PSM(t,f) = \frac{|S(t,f)|}{|Y(t,f)|}\cos\theta$$

where $|S(t,f)|$ and $|Y(t,f)|$ represent spectral magnitudes of clean speech and noisy speech, respectively. θ denotes the difference between the clean speech phase and the noisy speech phase within the T-F unit.

- Input features: 161-D short-time Fourier transform (STFT) magnitude spectra.

[6] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 708–712.

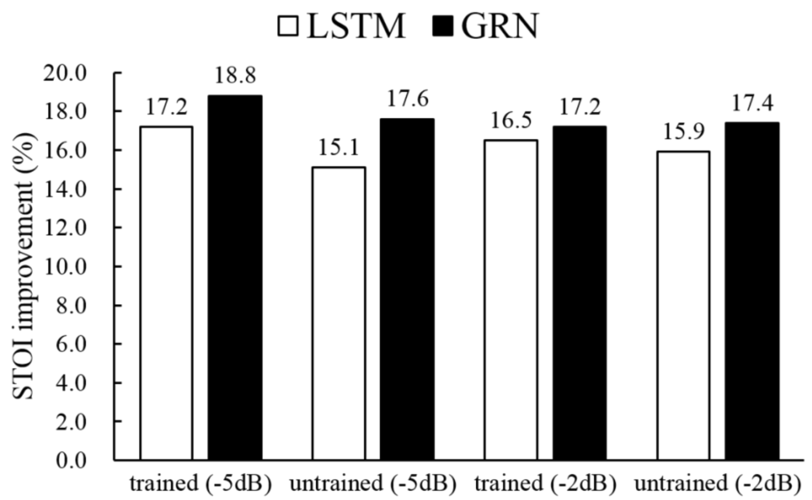
- On trained speakers: (GRN - gated residual network)

metrics	STOI (in %)						PESQ					
SNR	-5 dB			-2 dB			-5 dB			-2 dB		
noises	Avg.	babble	cafeteria	Avg.	babble	cafeteria	Avg.	babble	cafeteria	Avg.	babble	cafeteria
unprocessed	58.0	58.8	57.3	65.9	66.4	65.5	1.57	1.63	1.52	1.74	1.78	1.71
LSTM	75.2	76.4	74.1	82.4	83.2	81.6	2.07	2.05	2.09	2.39	2.37	2.41
GRN	**76.8**	**77.6**	**75.9**	**83.1**	**83.4**	**82.7**	**2.14**	**2.10**	**2.17**	**2.43**	**2.38**	**2.48**

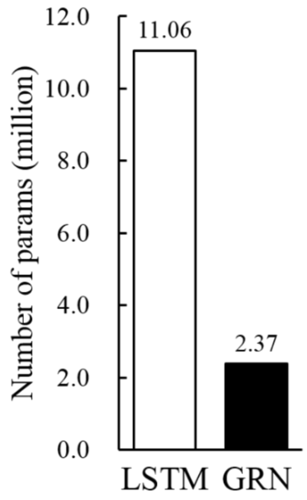
- On untrained speakers:

metrics	STOI (in %)						PESQ					
SNR	-5 dB			-2 dB			-5 dB			-2 dB		
noises	Avg.	babble	cafeteria	Avg.	babble	cafeteria	Avg.	babble	cafeteria	Avg.	babble	cafeteria
unprocessed	58.0	58.5	57.5	65.1	65.5	64.7	1.50	1.56	1.44	1.67	1.71	1.63
LSTM	73.1	73.0	73.2	81.0	81.1	80.9	1.96	1.89	2.04	2.30	2.26	2.34
GRN	**75.6**	**75.8**	**75.3**	**82.5**	**82.5**	**82.4**	**2.05**	**1.99**	**2.11**	**2.35**	**2.30**	**2.40**

- STOI improvements over the unprocessed mixtures (averaged over the two noises):



- Parameter efficiency:



- babble, -5 dB
  untrained speaker:
  - unprocessed:
  - LSTM:
  - GRN:
  - clean:

- cafeteria, -2 dB
  untrained speaker:
  - unprocessed:
  - LSTM:
  - GRN:
  - clean:

Conclusion

- For the sequence-to-sequence mapping, the GRN benefits from its large receptive fields upon the inputs. It consistently outperforms a strong LSTM model in terms of STOI and PESQ.

- The LSTM learns temporal dynamics of speech, but it cannot sufficiently utilize frequency information. The proposed GRN, however, systematically aggregates contexts along both the frequency and the time directions.

- Another advantage of the GRN is its higher parameter efficiency due to the use of shared weights in convolutions.

- We believe that the proposed model lays a sound foundation for investigations towards CNNs for supervised speech separation.

Thank you for your attention!