Inference Efficient Source Separation Using Input-dependent Convolutions

Shogo Seki and Li Li
CyberAgent, Inc., Japan
{seki_shogo,li_li}@cyberagent.co.jp

***Abstract*—This paper proposes a large-capacity but inference-efficient source separation model that considers input mixtures. Convolutional neural networks (CNNs) are fundamental elements for building robust separation models, and CNN-based separation models are yet attractive in their lower computational complexity than recent competitive separation models. One premise of these models is that a single kernel is applied to the input at each convolution. This does not take into account different input mixtures, such as those with speakers of the same or different genders, resulting in suboptimal separation performance. To overcome the issue, the proposed method employs an input-dependent convolution called conditionally parameterized convolution (CondConv) for CNN-based separation models. CondConv contains multiple kernels and generates an aggregated single kernel depending on the input. This can increase the network capacity while maintaining the computation complexity during inference as a standard convolution, enabling separation models to account for the input mixtures. Through the experimental evaluations under a speech separation task, the proposed input-dependent convolution approach consistently improves several CNN-based separation models while maintaining a negligible increase in inference.**

I. Introduction

Source separation is a technique for separating the underlying source signals present in an observed mixture signal, and it has various practical applications. Especially in single-channel source separation, where observed signals are obtained from a single microphone, significant progress has been made owing to the continuous growth of deep neural networks (DNNs).

As one of the most successful separation models, Conv-TasNet [1] is a well-known approach that designs an encoder-separator-decoder framework and constitutes them with convolutional neural networks (CNNs), achieving source separation in the time domain. For other successful separation models, dual-path modules introduce an efficient way to model long-form signals [2], and band-split architectures bring full-/sub-band modeling of signals [3]. These separation models were initially based on recurrent neural networks (RNNs), but they have been further developed by introducing more sophisticated architectures such as Transformers [4] and state-space models (SSMs) [5], [6], achieving state-of-the-art performances [7]–[9]. Although it is no longer easy to outperform novel competitive separation models, CNN-based separation models [10], [11], including Conv-TasNet, are yet attractive in their lower computational complexities in developing lightweight and low-latency systems.

Convolutional layers play a pivotal role in CNN-based separation models, where a single kernel is applied to individual inputs. On the other hand, a wide variety of source mixtures can appear as mixtures to input the separation models. For example, in speech mixtures between two speakers, mixed speech can be obtained from speaker pairs of the same gender (intra-gender) or different genders (inter-gender). In intra-gender speaker mixtures, the speakers of a mixture signal have closely overlapped pitch ranges and convolution kernels are desired to distinguish the differences. These kernels, however, are inappropriate for inter-gender speaker mixtures because it is possible to overseparate the components from a single speaker. The converse case can happen vice versa. In such a situation, convolutional layers cannot handle different types of input mixtures separately, which may fail to transform hidden representations and lead to suboptimal separation model performance.

To overcome this limitation, we propose to employ conditionally parameterized convolution (CondConv) [12], one of input-dependent convolution approaches [12]–[14], for CNN-based separation models. CondConv consists of multiple kernels and a routing module to generate linear combination coefficients according to the input. An aggregated single kernel is then generated as a linear combination of the multiple kernels and applied to the input. This enables convolutional layers to use different kernels for each input, and separation models can be expected to separate individual input mixtures effectively. Furthermore, since CondConv uses a simple routing module and aggregates not input but kernels, the computational cost is almost equivalent to a standard convolution while retaining a more extensive network capacity.

II. Preliminaries: CNN-based separation models

We begin with formulating a single-channel source separation problem addressed in this paper and then overview the convolutional encoder-separator-decoder architecture. Since the proposed method is based on CNN-based separation models, we use Conv-TasNet to provide a simple description.

A. Problem Formulation

Suppose that there is a single microphone that receives a mixture signal consisting of J source signals. Let $s_j[n]$ and $x[n]$ be the waveform samples of the j-th source signal and the observed signal, where $n \in \{1, \dots, N\}$ is the sample index in the time domain. The observed signal is written as:

$$x[n] = \sum\nolimits_{j=1}^{J} s_j[n]. \tag{1}$$

Single-channel source separation problem is the task of separating out each source signal $s_j[n], j \in 1, \ldots, J$ given a single-channel mixture signal $x[n]$.

B. Encoder-Separator-Decoder Architecture

Before feeding into a convolutional encoder, an observed mixture signal is first divided into overlapped segments of length L, denoted as $\mathbf{x}(t) \in \mathbb{R}^L$, where $t \in \{1, \ldots, T\}$ represents the frame index and T is the total number of segments.

Each mixture segment $\mathbf{x}(t)$ is encoded by a one-dimensional convolutional layer with a learnable weight $\mathbf{B} \in \mathbb{R}^{N \times L}$:

$$\mathbf{h}(t) = \mathcal{H}(\mathbf{B}\mathbf{x}(t)), \tag{2}$$

where $\mathcal{H}(\cdot)$ represents an arbitrary activation function such as rectified linear unit (ReLU) [15].

Encoded representation $\mathbf{h}(t) \in \mathbb{R}^N$ is then fed into a separator network, e.g., temporal convolutional network (TCN) [16] in Conv-TasNet, to estimate masks $\mathbf{m}_j(t) \in \mathbb{R}^N$. The TCN consists of repeated stacks of dilated convolutional blocks with exponentially increasing dilation factors. Each block is composed of three-layer convolutional networks, where the first and last layers are pointwise convolutions and the middle layer is a depthwise convolution with dilation[1]. Given an input $\mathbf{e}_{\text{input}}$, each block outputs the transformed representation $\mathbf{e}_{\text{output}}$:

$$\mathbf{e}_{\text{output}} = (\texttt{PConv} \circ g \circ \texttt{DConv} \circ g \circ \texttt{PConv})(\mathbf{e}_{\text{input}}), \tag{3}$$

where $\circ$ represents the composition of functions, and $g(\cdot)$ represents a composite function of parametric ReLU (PReLU) [17] and layer normalization [18]: $g(\mathbf{e}) = (\texttt{LayerNorm} \circ \texttt{PReLU})(\mathbf{x})$.

The latent representations of each source signal are predicted by: $\mathbf{v}_j(t) = \mathbf{m}_j(t) \odot \mathbf{h}(t)$, where $\odot$ represents element-wise multiplication. The estimated representation $\mathbf{v}_j(t) \in \mathbb{R}^N$ is decoded into a source segment with a one-dimensional transposed convolutional layer with a parameter $\mathbf{U} \in \mathbb{R}^{L \times N}$:

$$\hat{\mathbf{s}}_j(t) = \mathbf{U}\mathbf{v}_j(t), \tag{4}$$

Finally, separated signals are obtained by applying overlap-add to waveform segments.

III. Proposed Method

In the proposed method, we adapt CondConv to the conventional CNN-based separation models. Furthermore, we also present an accelerating approach for GPU computing.

[1]Although the original Conv-TasNet contains a skip connection branch in each block [1], we omit it for simplicity. Indeed, we validated that the skip connection did not help significantly improve performance in our preliminary experiments.

A. Conditionally Parameterized Convolution (CondConv)

CondConv differs from a standard convolution in that it contains multiple kernels and an additional routing module. Let x and $W_k, k \in \{1, \ldots, K\}$ be the input and k-th kernel of a CondConv, respectively.

CondConv aggregates each kernel W_k to generate a single kernel by a linear combination:

$$f(x; W_1, \ldots, W_K) = \left(\sum\nolimits_k \alpha_k(x) W_k\right) * x, \tag{5}$$

where $\alpha_k(\cdot)$ is an input-dependent scalar weight computed from the routing module. K is the number of experts since CondConv is equivalent to a mixture of experts (MoE) formulation as follows:

$$\left(\sum\nolimits_k \alpha_k(x) W_k\right) * x = \sum\nolimits_k \alpha_k(x) \left(W_k * x\right). \tag{6}$$

Thus, this implies that CondConv has the same capacity as the MoE model.

For the routing module, we design a network composed of global average pooling, dropout, and fully-connected layer, and routing weights $\alpha_1(x), \ldots, \alpha_K(x)$ are obtained through a sigmoid function:

$$\alpha_1(x), \ldots, \alpha_K(x) = (\texttt{Sigmoid} \circ r)(x), \tag{7}$$

$$r(x) = (\texttt{FC} \circ \texttt{DropOut} \circ \texttt{GlobalAveragePool})(x). \tag{8}$$

Since it is possible to replace the standard convolutional layers in CNN-based models with CondConvs, a convolutional encoder eq. (2) can be rewritten as follows:

$$\mathbf{h}(t) = \mathcal{H}\left(\sum\nolimits_k \left[\beta_k(\mathbf{x}(t))\mathbf{B}_k\right] \mathbf{x}(t)\right), \tag{9}$$

where, $\beta_k(\cdot)$ and $\mathbf{B}_k$ represent k-th routing weight and learnable weight, respectively. Similarly, a convolutional decoder eq. (4) can be reformulated as follows:

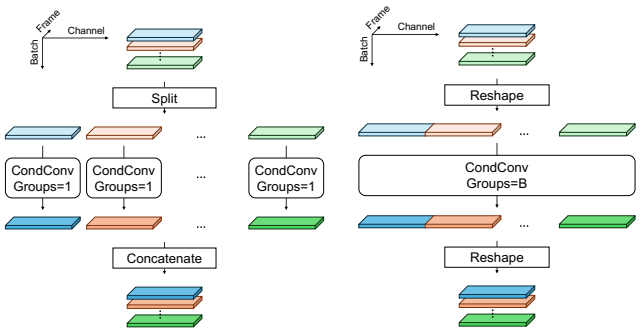
$$\hat{\mathbf{s}}_j(t) = \sum\nolimits_k \left[\gamma_k(\mathbf{x}(t))\mathbf{U}_k\right] \mathbf{v}_j(t), \tag{10}$$

where, $\gamma_k(\cdot)$ represents the k-th routing weight, and $\mathbf{U}_k$ denotes k-th learnable weight of the convolutional decoder. For the separator, it is also possible to replace the convolutional layers in the same manner.

B. Implementation

It is worth noting that since the routing weights in CondConv depend on input samples, it is required repeated convolution operations for minibatch data during training. One way to perform CondConv is to apply convolutions sample by sample, however, we found this causes slow training. To fully receive fast GPU computing, we develop a modified algorithm using a group convolution.

Fig. 1 shows a comparison of the algorithms using a naive convolution and a group convolution. In the naive convolution approach, it is required to repeat the CondConv operation with the same number of batch sizes, which is inefficient in GPU computing and results in a bottleneck during training.



(a) Naive convolution. (b) Group convolution.

Fig. 1: Comparison of algorithms using (a) naive convolution and (b) group convolution.

Focusing on the fact that every CondConv operation in Fig. 1 (a) are independent of each other, CondConv operations can be regarded as a large group convolution (Fig. 1 (b)) [19], [20]. This implementation is beneficial for GPU acceleration, enabling fast training.

IV. Experimental Evaluation

A. Settings

The proposed approach was experimentally evaluated under a speaker separation scenario where the task is to separate out two sources from single-channel mixture signals. We used the WSJ0-2mix dataset [21], a popular dataset to benchmark monaural speaker separation systems. WSJ0-2mix uses samples from the WSJ0 corpus and is composed of 20000, 5000, and 3000 two-speaker mixtures in the training, validation, and test sets, respectively. The speakers in the training and validation sets are disjoint from those in the test set. For each mixture, two utterances are overlapped, and mixed with at random signal-to-noise ratios (SNRs) in the range $[-5, 5]$ dB. In the evaluation, the sampling rate of each mixture was downsampled at 8 kHz.

We chose CNN-based separation models, Conv-TasNet [1], SuDoRM-RF [10], and its lightweight variants (SuDoRM-RF++ and SuDoRM-RF++GC) [11] as the baseline models and developed the proposed method by replacing standard convolutional layers in these models with CondConv layers. SuDoRM-RF and its variants are CNN-based efficient separation models, where each separator uses the stacks of U-net-like network architecture [22]. For the Conv-TasNet model, we used `espnet` implementation[2] and training recipe. For the other models, we used official implementation[3]. For CondConv layers, we set the number of experts at four, and the dropout rate was set to 0.2. All the models were trained with four-second segments, and the Adam optmizer [23] with an initial learning rate of 0.001 was used. We used the negative scale-invariant signal-to-distortion ratio (SI-SDR) [24] as the loss function, and utterance-wise permutation invariant training (PIT) was used to solve the permutation problems [25].

[2]https://github.com/espnet/espnet

[3]https://github.com/etzinis/sudo_rm_rf

TABLE I: Comparative studies for (a) modules, (b) network sizes, and (c) layers, where bold values represent the best separation performance.

(a) Comparison of encoder (Enc.), separator (Sep.), and decoder (Dec.) modules with (✓) or without (✗) CondConv.

Enc.	Sep.	Dec.	# Params [M]	# MACs [G/s]	SI-SDRi [dB]
✗	✗	✗	8.71	7.02	15.63
✗	✗	✓	8.73	7.02	16.21
✗	✓	✗	34.78	7.03	15.90
✗	✓	✓	34.80	7.03	16.07
✓	✗	✗	8.73	7.02	15.69
✓	✗	✓	8.74	7.02	16.29
✓	✓	✗	34.80	7.03	16.40
✓	✓	✓	34.81	7.03	**16.82**

(b) Comparison of separation models using standard convolutions (Conv) with large network size and using CondConv.

Method	# Params [M]	# MACs [G/s]	SI-SDRi [dB]
Conv	8.71	7.02	15.63
Conv 4x	34.84	28.07	15.85
CondConv	34.81	7.03	**16.82**

(c) Comparison of separation models using various CondConvs, where the values in brackets represent the numbers of experts.

Method	# Params [M]	# MACs [G/s]	SI-SDRi [dB]
Conv	8.71	7.02	15.63
CondConv (1)	8.75	7.02	15.78
CondConv (2)	17.44	7.02	16.00
CondConv (4)	34.81	7.03	**16.82**
CondConv (8)	69.56	7.04	16.28
CondConv (16)	139.05	7.05	16.12

As the evaluation metrics, we used the SI-SDR improvements (SI-SDRi) to evaluate the separation performance. We also investigated the multiply-accumulate operations (MACs) for one-second input to estimate computational efficiency and used `torchprofile`[4].

B. Results

We first investigated the proposed CondConv approach using Conv-TasNet. TABLE I shows a comparison of Conv-TasNet models using standard convolutions and CondConvs.

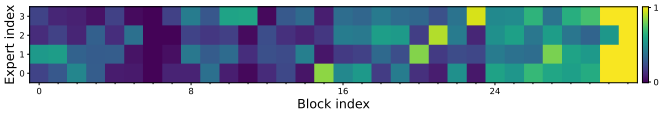
1) Comparison of module-level replacements: In our comprehensive evaluation, we examined the encoder, separator, and decoder by replacing the convolutional layers with CondConvs. We found that, although replacing the encoder with CondConv has little effect, replacing the separator and decoder is effective. Moreover, replacing all the modules was the most effective, and we applied CondConv to all the convolutional layers in the following evaluation.

2) Comparison with models of similar size: To clarify the performance when compared with models of the same size, we compared the proposed CondConv approach with a large Conv-TasNet model. The conventional large Conv-TasNet performs better than small Conv-TasNet, but the computational cost is

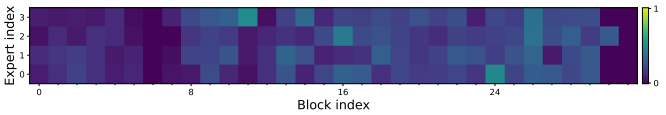
[4]https://github.com/zhijian-liu/torchprofile

TABLE II: SI-SDRi [dB] for mixtures from intra-gender and inter-gender speaker pairs, where the values in brackets represent the differences between methods.

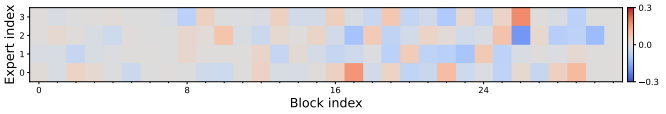
Method	Intra-gender	Inter-gender	Total
Conv	14.54 (±0.00)	16.69 (±0.00)	15.63 (±0.00)
CondConv	**16.05** (+1.51)	**17.56** (+0.87)	**16.82** (+1.19)



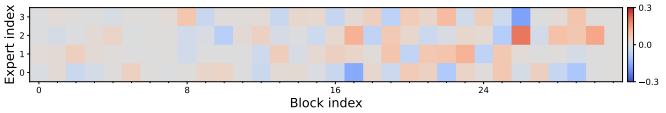
(a) Mean



(b) Standard deviation



(c) Mean difference of intra-gender speaker pairs from total pairs.



(d) Mean difference of inter-gender speaker pairs from total pairs.

Fig. 2: Visualization of routing weight statistics of the last CondConv layers at each TCN block: (a) mean, (b) standard deviation and mean differences of (c) intra-gender and (d) inter-gender speaker pairs from total pairs.

linearly increased. On the other hand, the proposed CondConv approach achieves not only better separation performance than large Conv-TasNet but also maintains similar computational costs as small Conv-TasNet.

3) Comparison with the number of experts: We compared CondConvs with different numbers of experts, where the numbers of experts were set as $\{1, 2, 4, 8, 16\}$. It is found that the separation performance tends to be improved when increasing the number of experts. On the other hand, it was also found that too many experts did not necessarily provide significant improvement. This might be because the network capacity became too large, making the training difficult. The CondConv, with four experts, achieved the best results, and we used the same number of experts in the following experiments.

We next analyzed how the proposed approach improves separation performance. TABLE II compares separation performances for different types of two-speaker pairs. The proposed approach consistently improves performance. When comparing different speaker pairs, the proposed method contributes more to improving performance for intra-gender cases.

Visualization of the routing weights for each TCN block in Conv-TasNet using CondConv is shown in Fig. 2, where the horizontal axis represents the block index and the vertical axis represents the expert index. In the shallow block, routing weights have a small standard deviation, and similar weights are used for all experts, which is consistent with the previous finding that replacing the encoder with CondConv has little improvement. In the deeper block, routing weights tend to vary.

TABLE III: Comparison with other separation models

Method	# Params [M]	# MACs [G/s]	SI-SDRi [dB]
DPRNN [2]	2.6	42.2	18.8
SepFromer [26]	26.0	59.5	20.4
TF-GridNet [7]	14.5	231.1	23.5
SPMamba [8]	6.1	N/A	†22.5
DPMamba [9]	59.8	N/A	24.4
Conv-TasNet			
Conv	8.71	7.02	15.63
CondConv	34.81	7.03	**16.82**
SuDoRM-RF			
Conv	2.66	10.60	16.70
CondConv	10.27	10.61	**16.80**
SuDoRM-RF++			
Conv	2.72	9.11	16.29
CondConv	10.59	9.12	**16.33**
SuDoRM-RF++GC			
Conv	0.30	3.33	13.19
CondConv	1.14	3.33	**14.27**

†`https://github.com/JusperLee/SPMamba`

We compare the routing weights for mixtures of intra-gender and inter-gender speaker pairs (Fig. 2 (c) and (d)). As described above, routing weights in shallow blocks show similar activation patterns in both intra-gender and inter-gender cases. Meanwhile, routing weights in deeper blocks get active and vary depending on each expert. Furthermore, we can see that some of the experts behave complementary depending on the type of speaker pairs. This demonstrates that the proposed CondConv approach takes input mixtures into account.

We finally evaluated the proposed approach for different CNN-based methods and compared it with novel competitive separation models, and the results are shown in TABLE III. We can see that the recent state-of-the-art separation models achieve sufficient separation performances. In contrast, CNN-based separation models achieve at least four or more times lower computational costs than these powerful models. We also find that the proposed CondConv approach consistently achieved better performance in all the CNN-based models, though performance improvements depend on separation models. Hence, the proposed approach has a large potential to boost the existing separation models.

V. Conclusions

This paper presented an input-dependent convolution approach for CNN-based source separation models. The proposed method employs CondConv and replaces the standard convolutional layers in separation models, enabling the models to increase the network capacity while maintaining the computation complexity during inference. Through the experimental evaluation under a two-speaker separation task on WSJ0-2mix, the proposed method consistently improved performance for various separation models, demonstrating that it considers the mixture's input types during separation.

References

[1] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[2] Y. Luo, Z. Chen, and T. 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*, 2020, pp. 46–50.

[3] Y. Luo and J. Yu, "Music Source Separation With Band-Split RNN," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1893–1901, 2023.

[4] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is All You Need," in *Advances in Neural Information Processing Systems*, 2017.

[5] A. Gu, K. Goel, and C. Re, "Efficiently Modeling Long Sequences with Structured State Spaces," in *International Conference on Learning Representations*, 2022.

[6] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[7] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, "TF-GridNet: Integrating Full- and Sub-Band Modeling for Speech Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 3221–3236, 2023.

[8] K. Li and G. Chen, "SPMamba: State-space model is all you need in speech separation," *arXiv preprint arXiv:2404.02063*, 2024.

[9] X. Jiang, C. Han, and N. Mesgarani, "Dual-path Mamba: Short and Long-term Bidirectional Selective Structured State Space Models for Speech Separation," *arXiv preprint arXiv:2403.18257*, 2024.

[10] E. Tzinis, Z. Wang, and P. Smaragdis, "Sudo RM -RF: Efficient Networks for Universal Audio Source Separation," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2020, pp. 1–6.

[11] E. Tzinis, Z. Wang, X. Jiang, and P. Smaragdis, "Compute and Memory Efficient Universal Sound Source Separation," *Journal of Signal Processing Systems*, vol. 94, no. 2, pp. 245–259, 2022.

[12] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "CondConv: Conditionally Parameterized Convolutions for Efficient Inference," in *Advances in Neural Information Processing Systems*, 2019.

[13] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic Convolution: Attention Over Convolution Kernels," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 030–11 039.

[14] C. Kosma, G. Nikolentzos, and M. Vazirgiannis, "Time-Parameterized Convolutional Neural Networks for Irregularly Sampled Time Series," *arXiv preprint arXiv:2308.03210*, 2023.

[15] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *International Conference on Machine Learning*, 2010, pp. 807–814.

[16] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[18] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[19] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5987–5995.

[20] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6848–6856.

[21] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 31–35.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[23] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[24] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – Half-baked or Well Done?" In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 626–630.

[25] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[26] C. Subakan, M. Ravanelli, S. Cornell, F. Grondin, and M. Bronzi, "Exploring self-attention mechanisms for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2169–2180, 2023.