Interspeech 2024
1-5 September 2024, Kos, Greece

PARIS: Pseudo-AutoRegressIve Siamese Training for Online Speech Separation

Zexu Pan[1], *Gordon Wichern*[1], *Francois G. Germain*[1], *Kohei Saijo*[1,2], *Jonathan Le Roux*[1]

[1]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA
[2]Waseda University, Tokyo, Japan

pan_zexu@u.nus.edu, {wichern,germain,leroux}@merl.com, saijo@pcl.cs.waseda.ac.jp

Abstract

While offline speech separation models have made significant advances, the streaming regime remains less explored and is typically limited to causal modifications of existing offline networks. This study focuses on empowering a streaming speech separation model with autoregressive capability, in which the current step separation is conditioned on separated samples from past steps. To do so, we introduce pseudo-autoregressive Siamese (PARIS) training: with only two forward passes through a Siamese-style network for each batch, PARIS avoids the training-inference mismatch in teacher forcing and the need for numerous autoregressive steps during training. The proposed PARIS training improves the recent online SkiM model by 1.5 dB in SI-SNR on the WSJ0-2mix dataset, with minimal change to the network architecture and inference time.

Index Terms: autoregressive, speech separation, source separation, online

1. Introduction

Speech signals often overlap with each other in natural scenes, but the human brain has the inherent ability to separate such signals. It is crucial to equip machines with such ability, which is termed as the "cocktail party problem" [1], as it serves as a crucial frontend for other tasks such as automatic speech recognition or speaker localization.

The task of speech separation [2] tackles the "cocktail party problem" by separating a mixture signal into individual clean signals. Many great network architectures have been designed in the deep learning era, such as Conv-TasNet [3], DPRNN [4], TF-GridNet [5], and others [6–10]. Target speaker extraction is another active line of research that aims at only separating the speech of a person of interest, conditioned on an auxiliary signal such as reference speech [11–13], visual recordings [14–18], brain signals [19, 20], metadata [21], or distance [22].

The majority of speech separation and extraction networks are primarily designed and evaluated for offline processing. Online streaming models typically emerge as causal modifications of offline networks. However, such offline networks are usually developed for utterance-level processing, which is available to them, while online processing naturally entails having access to the separated speech signal from past steps when processing the newly acquired mixture signal at the current step, making the case for actively exploiting this information. Speech generation networks such as WaveNet [23] and SampleRNN [24] have unveiled the power of generating high-fidelity natural speech in an autoregressive (AR) manner by feeding in past-step model outputs. Similarly, we aim to use the separated signals in the past step as a prior to condition the model in the current-step separation in an autoregressive manner, with the hope of enhancing the separated speech quality.

Although autoregressive models are powerful, their training is tedious as the model needs to forward-pass every feature frame sequentially in steps. A common technique to train AR networks is teacher forcing [25], which uses ground truth as the past-step estimate during training and utilizes model output during inference, but the mismatch may become problematic in speech models as the error propagates quickly with high frame rates of speech signals.

In the literature, the Listen and Grouping network attempted online autoregressive speech separation by proposing a multi-time-step prediction training (MCT) [26, 27]. For each batch in training, the model is initialized with teacher forcing and then performs the forward pass for a number of time steps before back-propagation. The model performs better when the number of steps is close to the model's receptive field. Alternatively, a speech enhancement work proposed iterative autoregression (IA) [28], which forward-passes the whole utterance instead. IA trains the model using teacher forcing in a first pass, then replaces the conditioning ground truth with the model's own outputs for multiple passes iteratively, and the loss is back-propagated only for the last pass. Both works involve forward-passing the model many times to reduce the mismatch between teacher-forcing training and autoregressive inference.

Unlike generative language models in which the autoregressive feedback conditioning signal is a necessity, speech separation models are usually non-generative, and thus can separate the mixture signals adequately based only on mixture speech signals. Taking advantage of this, it is possible to use such non-autoregressive model output as the conditioning signal to train an autoregressive speech separation model. The Neuro-Heed [29] speaker extraction model used a non-autoregressive model output to train an autoregressive speaker encoder for enhancing the online speaker representation. Inspired by it, we aim to use such non-autoregressive model output, which we refer to as pseudo-autoregressive separated speech signals in this paper, to enhance the model's separation capability in an autoregressive way.

We propose PARIS, a Pseudo-AutoRegressIve Siamese training paradigm to train an online autoregressive speech separation model. In training, the model performs two passes at the utterance level in a Siamese-style network. The first pass is through the first network, which only takes in the mixture signal as input and the output is treated as the pseudo-autoregressive separated speech signals. The second pass is through the second network, which takes in the mixture signal and the shifted pseudo-autoregressive separated speech signals from the first pass to perform speech separation. The model weights of the networks are shared. During inference, the model is conditioned on its own past step outputs in a real autoregressive

10.21437/Interspeech.2024-1066

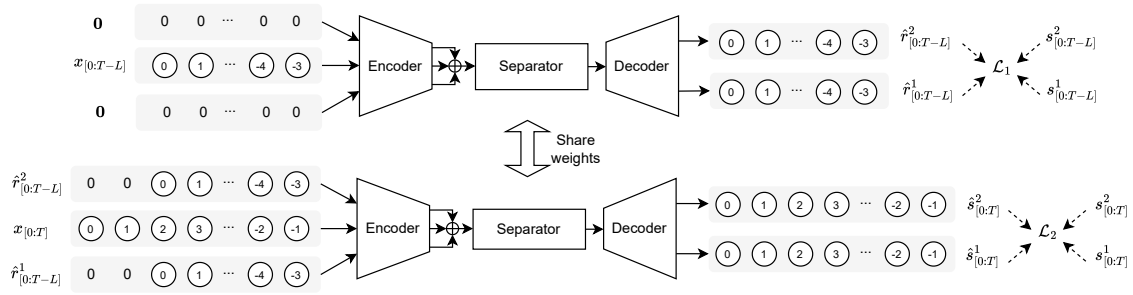


Fig. 1: *The proposed Pseudo-AutoRegressIve Siamese training (PARIS) for an online autoregressive speech separation model, which is illustrated with chunk size $L = 2$. The upper network and lower network are identical with shared weights. Only the lower network is used for streaming inference. All streams are encoded with the same encoder and concatenated ($\oplus$) along the channel dimension.*

way. We compare PARIS with traditional training on the skipping memory long short-term memory network (SkiM) which is known for its low latency and excellent performance for online speech separation [30]. With only an additional projection layer added to the network architecture to incorporate the past separated speech signals, PARIS achieves a 1.5 dB improvement in terms of scale-invariant signal-to-noise ratio (SI-SNR) [31] on the WSJ0-2mix dataset [2].

2. Online Autoregressive Separation

Given a speech mixture waveform $x = (x_t)_{t=1,\dots,T}$ of length T, a two-speaker separation model estimates $\hat{s} = (\hat{s}^1, \hat{s}^2)$ that approximate the clean signal of individual speakers $s = (s^1, s^2)$. In online inference, a causal speech separation model handles x in multiple steps, processing at each step k a chunk of L newly acquired waveform samples $x_{[t_k:t_k+L]}$[1], with start time t_k determined by the hop size. For an autoregressive speech separation model $f(\cdot)$, the estimation of $\hat{s}_{[t_k:t_k+L]}$ is further conditioned on the outputs $\hat{s}_{[t_k-L:t_k]}$ from the past step:

$$\hat{s}_{[t_k:t_k+L]} = f(x_{[t_k:t_k+L]}, \hat{s}_{[t_k-L:t_k]}) \tag{1}$$

A speech separation model typically consists of a speech encoder, a separator or a mask estimator, and a speech decoder, irrespective of whether the model operates in the time domain or the frequency domain. In the frequency domain, the chunk size L is usually the short-time Fourier transform (STFT) window length, while in the time domain, it is usually the kernel size of the first one-dimensional convolutional layer (e.g., [3]). It is common for the speech encoder to have a hop size typically a fraction of the kernel size, and the speech decoder to have an overlap-add operation with the same hop size. We here refer to as "past step" the fully reconstructed chunk that ends just before the beginning of the current chunk. Ideally, we would like to use the same speech encoder to process the past signals and the mixture signal, thus the chunk size of the past signal used at each step is also set to L here.

The naive way of training an autoregressive model is to run the model step-by-step for an utterance during training, but this makes training extremely slow. In machine translation and natural language processing, teacher forcing [25] is commonly used to speed up training, and scheduled sampling [32] is used to minimize the resulting training-inference mismatch. Similarly, MCT [26] and IA [28] have been proposed for speech separation and enhancement respectively, but both involve teacher forcing with several forward passes for each batch, which is still slow compared to non-autoregressive model training.

[1] We use a Python-style notation here where the last index is omitted.

3. PARIS

We propose a Siamese-style training paradigm named PARIS to train an autoregressive speech separation model, without the need for teacher forcing or forwarding the model multiple steps for each batch.

3.1. Network architecture

PARIS relies on two identical networks in parallel with shared weights, as shown in Fig. 1. The network could be virtually any speech separation or target speaker extraction model, as networks generally follow the encoder-separator-decoder architecture. We build on top of the two-speaker-separation SkiM model [30] in this work.

The network should receive the mixture signal and both past-step output signals as inputs. Similarly to [33], we use the same speech encoder to process the signals independently. We opt for concatenation fusion, which concatenates the three features along the channel dimension after the convolution layer in the speech encoder. Only an additional linear layer is added to the original SkiM model to project the concatenated features back to the original feature dimension.

3.2. Data flow

We illustrate PARIS with chunk size $L = 2$ in Fig. 1. For each batch in training, there are two forward passes before backpropagation. The first pass is through the upper network to obtain the "pseudo" past separated speech $\hat{r} \in \{\hat{r}^1, \hat{r}^2\}$. The upper network is similar to conventional speech separation model training, except that there are two redundant zero vectors as input besides the mixture signal $x_{[0:T-L]}$ to accommodate the Linear projection layer after the speech encoder. The second pass is through the lower network that simulates the autoregressive speech separation, using pseudo past separated speech $\hat{r}$ as the conditioning prior. The $\hat{r}$ is right shifted by the chunk size L when inputting to the lower network such that the current step estimation is paired with the last step pseudo-separated speech. The gradient of $\hat{r}$ is also detached from the computational graph when feeding $\hat{r}$ into the lower network.

Since the two networks have identical architecture and shared weights, with the only difference being the conditioning signal, and they are both trained to separate the signals, we expect their outputs to become similar as training progresses, which is why we refer to $\hat{r}$ as "pseudo" past separated speech. During streaming inference, the data flows only through the lower network step-by-step, with its outputs fed back as input in the next step of processing, in a true autoregressive fashion.

Table 1: *We use the validation set autoregressive streaming decoding results here to select the best settings for our proposed PARIS training. The validation set (CV) and test set (TT) SI-SNR values are reported in dB. α is the scalar weight in Eq. (4), stop gradient refers to detaching the gradient of $\hat{r}$ from the computational graph when passing it into the lower network, training prior refers to the whether pseudo speech $\hat{r}$ or ground truth speech s is passed into the lower network, shared weights refers to sharing network weights between the upper and lower network, and loss refers to whether SNR or SI-SNR loss is used in Eq. (2 and 3).*

Sys. #	α	Stop gradient	Train prior	Share weights	Loss	CV SI-SNR	TT SI-SNR
1	0.00				SNR	14.7	14.1
2	0.25	✓	Pseudo	✓	SNR	15.2	14.7
3	0.50				SNR	15.0	14.3
4	0.25	✗	Pseudo	✓	SNR	14.3	13.7
5		✓	Pseudo	✓	SI-SNR	14.6	13.5
6		✓	GoundTruth	✓	SNR	0.6	–4.5
7		✓	Pseudo	✗	SNR	15.0	14.3

3.3. Loss function

Both outputs from the upper and lower networks are constrained with a loss function, for which we maximize the signal-to-noise ratio (SNR) between the network outputs and the clean speech signal s^1 and s^2:

$$\mathcal{L}_1 = -10\log_{10}(\frac{||s||^2}{||\hat{r}-s||^2}) \quad (2)$$

$$\mathcal{L}_2 = -10\log_{10}(\frac{||s||^2}{||\hat{s}-s||^2}) \quad (3)$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are applied to upper and lower network outputs respectively. Utterance-level permutation invariant training is applied during training to address the permutation problem [2, 34]. The overall loss is the weighted sum of the two losses with a scalar α:

$$\mathcal{L}_{overall} = \alpha * \mathcal{L}_1 + (1-\alpha) * \mathcal{L}_2 \quad (4)$$

We choose the scale-sensitive loss function SNR instead of the widely used scale-invariant signal-to-noise ratio (SI-SNR) [31] for PARIS, as it would be difficult to properly normalize the network output to be fed back as input in step-by-step streaming settings if the SI-SNR loss were used.

4. Experimental setup

4.1. Dataset

We evaluated our method on the widely-used WSJ0-2mix dataset [2]. We used the 8 kHz two-speaker "min" version of the dataset, which contains 30 hours of training data (TR), 10 hours of validation data (CV), and 5 hours of test data (TT). The two speaker's speech signals are mixed at a random relative signal-to-noise ratio (SNR) between -5 and 5 dB.

4.2. Model and training settings

We use the online SkiM [30] model as our baseline in this paper, with hyperparameters following the ESPNet configuration [35]. The encoder has kernel size L set to 8, stride set to 4, and embedding size set to 128. The SkiM separator LSTM has a unit size of 384 and a non-overlapping segment size of 50.

We implement both the SkiM baseline and our proposed model using PyTorch. We use the Adam optimizer [36] with an initial learning rate of 0.001. After the 50th epoch, the learning rate decreases by 3% every other epoch, and the training stops at the 150th epoch. The batch size of the baseline is set to 16 while the batch size of our model is set to 8 such that the GPU memory occupation during training are similar. During training, the utterances are fixed to 4 seconds, while the full utterance is used for evaluation.

5. Results

In this section, we first use the validation set result to select our best training settings for PARIS, then compare our selected system with the baseline SkiM. We evaluate the signal quality of the separated speech signals using SI-SNR [31] and SDR [37], and the perceptual quality using perceptual evaluation of speech quality (PESQ) [38]. The higher the better for all three metrics. We use SI-SNR as our main measure when describing the results, as other measures show similar trends. We give every differently trained system a number (Sys. #) for clarity.

5.1. Model tuning

In Table 1, we first select the best α value by comparing Systems 1-3. System 2 with $\alpha = 0.25$ achieves the best validation SI-SNR of 15.2 dB, showing the importance of putting more emphasis on the lower network as its data flow is closer to real autoregressive settings compared to the upper network.

In System 4, we do not stop the gradient of $\hat{r}$ when feeding it to the lower network, thus $\mathcal{L}_2$ flows through the upper network, affecting the upper network's ability to estimate clean speech signals as an independent network, and the validation SI-SNR drops by 0.9 dB compared to System 2, justifying our proposal to stop the gradient.

In System 5, we use SI-SNR loss for $\mathcal{L}_1$ and $\mathcal{L}_2$, so the network outputs are not bounded and properly normalized, affecting the lower network's performance, and the validation SI-SNR drops by 0.6 dB compared to System 2.

In System 6, we use teacher forcing to train the lower network, which conditions on the ground truth s instead of $\hat{r}$. System 6 can separate the training set very well but the validation SI-SNR is only 0.6 dB, showing that there is significant mismatch between training and inference.

In System 7, we do not share the weights between the upper and lower networks, thus expecting a larger mismatch between the upper and lower networks' outputs. The validation SI-SNR drops by 0.2 dB compared to System 2.

5.2. Comparison with baseline

We compare PARIS with baselines in Table 2 on the WSJ0-2mix test set. Our main baseline is System 8, the original SkiM model. Our autoregressive System 2 outperforms SkiM

Table 2: *SI-SNR [dB], SDR [dB], and PESQ results for PARIS and baselines on WSJ0-2mix test set. At inference, the same PARIS system can be run in autoregressive (AR) mode (default), non-AR mode (indicated by †), or pseudo-AR mode (indicated by ‡). All systems have 7.9 million parameters.*

Sys. #	Method	AR	SI-SNR	SDR	PESQ
8	SkiM [30]	✗	13.2	13.6	2.79
9	SkiM (SNR loss)		13.0	13.4	2.78
2	PARIS	✓	14.7	15.1	2.99
$2^{\dagger}$		✗	13.9	14.3	2.87
$2^{\ddagger}$		Pseudo	14.8	15.1	2.99
1	PARIS ($\alpha = 0$)	✓	14.1	14.5	2.89
$1^{\dagger}$		✗	9.0	9.5	2.34
10	Two-stage PARIS	✗	14.6	15.0	3.00

by 1.5 dB in terms of SI-SNR and SDR, and 0.20 for PESQ.

As our System 2 is trained with SNR loss while SkiM in System 8 is trained with SI-SNR loss, we also report System 9 in which SkiM is trained with SNR loss. System 9 has degraded performance compared to System 8, showing the advantage of SI-SNR loss compared to SNR loss for conventional speech separation models. In future work, it would be worth exploring how to adapt SI-SNR loss into PARIS.

We also present the results of System 10, which is trained the same way as System 2, except that we do not shift $\hat{r}$ before passing into the lower network. Therefore, System 10 is a two-stage network, which is non-autoregressive and requires forwarding the network twice during inference, doubling memory cost and inference time. System 10 performs similarly to System 2, with 0.1 dB lower SI-SNR and 0.01 higher PESQ.

We plot a histogram of the test samples' SI-SNR in Fig. 2. Comparing Systems 2 and 8, they have a similar number of samples with negative SI-SNR, but System 2 has fewer samples with SI-SNR between 0 to 15 dB while more samples with SI-SNR greater than 15 dB.

5.3. Ablation studies

We also present several ablation studies of different inference modes in Table 2. Our proposed PARIS decoding in System 2 is autoregressive. However, because of our Siamese-style training, the same network can also be used in non-autoregressive (non-AR) and pseudo-autoregressive (pseudo-AR) decoding modes.

The non-AR decoding mode, denoted as System $2^{\dagger}$, corresponds to using the output of the upper network in Fig. 1, with only the mixture signal as input and zero vectors as conditioning. It obtains an SI-SNR of 13.9 dB, which is 0.8 dB lower than System 2. Interestingly, it is better than SkiM (System 8) by 0.7 dB in SI-SNR, which shows that the coupling with training the lower network appears to advantageously regularize the weights when applied to the upper network compared to training the weights through the latter alone.

In the pseudo-AR decoding mode, denoted as System $2^{\ddagger}$, we use the upper network output $\hat{r}$ instead of the real output $\hat{s}$ as the conditioning prior. System $2^{\ddagger}$ is thus also a two-stage network that involves forwarding both the upper and lower networks. System 2 has similar performance compared with System $2^{\ddagger}$, with the latter being higher in SI-SNR by 0.1 dB.

To further investigate the difference between decoding modes, we present scatter plots of the SI-SNR for different decoding modes on the test samples. In Fig. 3 (a), we plot System $2^{\ddagger}$ (pseudo-AR) against System $2^{\dagger}$ (non-AR). With more samples located in the upper left region, we can see that, for most samples, pseudo-AR decoding improves upon non-AR decoding, showing that conditioning on already separated samples helps in the separation of the incoming samples within an utterance. In Fig. 3 (b), we plot System 2 (AR) against System $2^{\ddagger}$ (pseudo-AR). We can see that the samples have a larger variance in the distribution along the two sides of the identity line. This may be due to the fact that the separation performance of the former steps accumulates and directly affects the latter steps with a compounding effect, causing some samples to improve significantly while some samples degrade significantly. On average, the two systems exhibit similar overall performance as shown in Table 2.

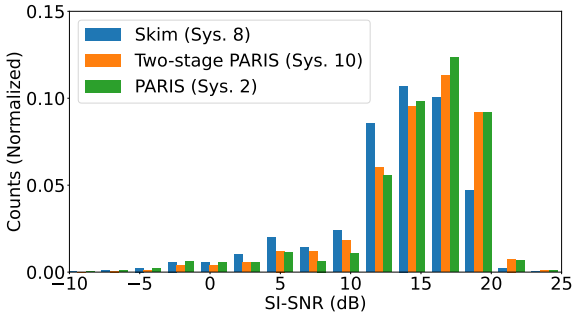
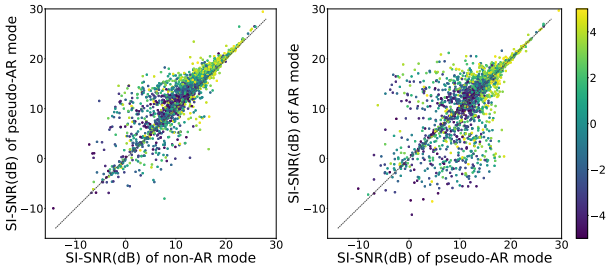


Fig. 2: *Distribution of test sample SI-SNR of SkiM baseline (Sys. 8), two-stage PARIS (Sys. 10), and proposed PARIS (Sys. 2).*



(a) Systems $2^{\ddagger}$ vs $2^{\dagger}$ (b) Systems 2 vs $2^{\ddagger}$

Fig. 3: *Scatter plot of test samples SI-SNR for different decoding modes of our proposed PARIS. Color represents the SNR of the mixture signal.*

We finally present in Table 2 the results of System 1 in AR decoding mode together with System $1^{\dagger}$ in non-AR decoding mode. Even though there is no loss applied to the upper network in System 1, meaning that the network is not directly trained to only receive mixture signals as input, the non-AR System $1^{\dagger}$ still has an SI-SNR of 9.0 dB, showing that the Siamese system creates non-trivial cooperation between the 2 networks

6. Conclusion

We presented PARIS, a training paradigm for autoregressive speech separation models that uses a pseudo-autoregressive scheme to reduce training complexity compared with fully autoregressive training, and training-inference mismatch compared with teacher forcing. PARIS only performs two forward passes for each batch during training, with a Siamese-style network setup. With only minimal change to the network architecture, it results in 1.5 dB improvement on the WSJ0-2mix 2-speaker dataset over regular training for the SkiM architecture.

7. References

[1] A. W. Bronkhorst, "The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions," *Acta Acustica united with Acustica*, vol. 86, no. 1, pp. 117–128, 2000.

[2] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, 2016, pp. 31–35.

[3] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, 2019.

[4] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. ICASSP*, 2020, pp. 46–50.

[5] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, "TF-GridNet: Making time-frequency domain models great again for monaural speaker separation," in *Proc. ICASSP*, 2023.

[6] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *Proc. ICASSP*. IEEE, 2021, pp. 21–25.

[7] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2840–2849, 2021.

[8] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Proc. ICASSP*, 2017, pp. 246–250.

[9] L. Yang, W. Liu, and W. Wang, "TFPSNet: Time-frequency domain path scanning network for speech separation," in *Proc. ICASSP*, 2022.

[10] J. Chen, Q. Mao, and D. Liu, "Dual-Path Transformer Network: Direct context-aware modeling for end-to-end monaural speech separation," in *Proc. Interspeech*, 2020.

[11] C. Xu, W. Rao, E. S. Chng, and H. Li, "SpEx: Multi-scale time domain speaker extraction network," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1370–1384, 2020.

[12] M. Delcroix, T. Ochiai, K. Zmolikova, K. Kinoshita, N. Tawara, T. Nakatani, and S. Araki, "Improving speaker discrimination of target speech extraction with time-domain SpeakerBeam," in *Proc. ICASSP*, 2020, pp. 691–695.

[13] Q. Wang, I. L. Moreno, M. Saglam, K. Wilson, A. Chiao, R. Liu, Y. He, W. Li, J. Pelecanos, M. Nika *et al.*, "VoiceFilter-Lite: Streaming targeted voice separation for on-device speech recognition," *Proc. Interspeech*, pp. 2677–2681, 2020.

[14] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–11, 2018.

[15] Z. Pan, G. Wichern, Y. Masuyama, F. G. Germain, S. Khurana, C. Hori, and J. Le Roux, "Scenario-aware audio-visual TF-Gridnet for target speech extraction," in *Proc. ASRU*, 2023.

[16] T. Afouras, J. S. Chung, and A. Zisserman, "My lips are concealed: Audio-visual speech enhancement through obstructions," in *Proc. INTERSPEECH*, 2019, pp. 4295–4299.

[17] J. Wu, Y. Xu, S. Zhang, L. Chen, M. Yu, L. Xie, and D. Yu, "Time domain audio visual speech separation," in *Proc. ASRU*, 2019, pp. 667–673.

[18] Z. Pan, X. Qian, and H. Li, "Speaker extraction with co-speech gestures cue," *IEEE Signal Process. Lett.*, vol. 29, pp. 1467–1471, 2022.

[19] E. Ceolini, J. Hjortkjær, D. D. Wong, J. O'Sullivan, V. S. Raghavan, J. Herrero, A. D. Mehta, S.-C. Liu, and N. Mesgarani, "Brain-informed speech separation (BISS) for enhancement of target speaker in multitalker speech perception," *NeuroImage*, vol. 223, p. 117282, 2020.

[20] Z. Pan, G. Wichern, F. G. Germain, S. Khurana, and J. Le Roux, "NeuroHeed+: Improving neuro-steered speaker extraction with joint auditory attention detection," in *Proc. ICASSP*, 2023.

[21] E. Tzinis, G. Wichern, A. Subramanian, P. Smaragdis, and J. Le Roux, "Heterogeneous target speech separation," in *Proc. Interspeech*, Sep. 2022.

[22] K. Patterson, K. Wilson, S. Wisdom, and J. R. Hershey, "Distance-Based Sound Separation," in *Proc. Interspeech*, 2022, pp. 901–905.

[23] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *Proc. ICML*, 2017.

[24] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional End-to-End neural audio generation model," *Proc. ICLR*, 2017.

[25] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[26] Z.-X. Li, Y. Song, L.-R. Dai, and I. McLoughlin, "Listening and grouping: an online autoregressive approach for monaural speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 4, pp. 692–703, 2019.

[27] H. Wang, Y. Song, Z.-X. Li, I. McLoughlin, and L.-R. Dai, "An online speaker-aware speech separation approach based on time-domain representation," in *Proc. ICASSP*. IEEE, 2020, pp. 6379–6383.

[28] P. Andreev, N. Babaev, A. Saginbaev, I. Shchekotov, and A. Alanov, "Iterative autoregression: a novel trick to improve your low-latency speech enhancement model," in *Proc. Interspeech*, 2023, pp. 2448–2452.

[29] Z. Pan, M. Borsdorf, S. Cai, T. Schultz, and H. Li, "NeuroHeed: Neuro-steered speaker extraction using EEG signals," *arXiv preprint arXiv:2307.14303*, 2023.

[30] C. Li, L. Yang, W. Wang, and Y. Qian, "SkiM: Skipping memory LSTM for low-latency real-time continuous speech separation," in *Proc. ICASSP*. IEEE, 2022, pp. 681–685.

[31] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR–half-baked or well done?" in *Proc. ICASSP*, 2019, pp. 626–630.

[32] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *Proc. NeurIPS*, vol. 28, 2015.

[33] E. Tzinis, S. Venkataramani, Z. Wang, C. Subakan, and P. Smaragdis, "Two-step sound source separation: Training on learned latent targets," in *Proc. ICASSP*. IEEE, 2020, pp. 31–35.

[34] 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 Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 10, pp. 1901–1913, 2017.

[35] C. Li, "Espnet configuration for skim model training on wsj0-2mix (wsj0_2mix_skim_small_causal)," [Online] https://huggingface.co/lichenda/wsj0_2mix_skim_small_causal#enhtrain_enh_skim_causal_smallraw, 2023.

[36] D. P. Kingma and J. Ba, "Adam, a method for stochastic optimization," in *Proc. ICLR*, vol. 1412, 2015.

[37] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, 2006.

[38] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. ICASSP*, 2001, pp. 749–752.