Online Self-Attentive Gated RNNs for Real-Time Speaker Separation

Ori Kabeli[1*], *Yossi Adi*[1*], *Zhenyu Tang*[3], *Buye Xu*[2], *Anurag Kumar*[2]

[1]Facebook AI Research, TLV, Israel
[2]Facebook Reality Labs, Redmond, WA, USA
[3]University of Maryland, College Park, MD, USA
{orik, xub, anuragkr90, adiyoss}@fb.com, zhy@umd.edu

Abstract

Deep neural networks have recently shown great success in the task of blind source separation, both under monaural and binaural settings. Although these methods were shown to produce high-quality separations, they were mainly applied under offline settings, in which the model has access to the full input signal while separating the signal. In this study, we convert a non-causal state-of-the-art separation model into a causal and real-time model and evaluate its performance under both online and offline settings. We compare the performance of the proposed model to several baseline methods under anechoic, noisy, and noisy-reverberant recording conditions while exploring both monaural and binaural inputs and outputs. Our findings shed light on the relative difference between causal and non-causal models when performing separation. Our stateful implementation for online separation leads to a minor drop in performance compared to the offline model; 0.8dB for monaural inputs and 0.3dB for binaural inputs while reaching a real-time factor of 0.65. Samples can be found under the following link: https://kwanum.github.io/sagrnnc-stream-results/.

1. Introduction

In real acoustic environments, a speech source of interest is usually corrupted by interfering sounds. The human auditory system excels at attending to a target speech source, where the cocktail party problem [1] aims to develop such capabilities in intelligent devices and systems. An important aspect of the cocktail party problem is speaker separation which aims to separate multiple concurrent speech signals of interest from a sound mixture. In *blind source separation*, the condition and the scene of the mixed sources are unknown to the separation system. Recently, research on speaker source separation has seen a great leap in performance due to the success of deep learning models considering both frequency domain [2, 3, 4, 5, 6], and time-domain [7, 8, 9, 10, 11, 12, 13] modeling.

To apply source separation models to real-time systems, (e.g., VR headsets) these should have the ability to process and separate sources in an online fashion (i.e., the model separates current mixture samples without having access to future samples). However, despite the success of prior works, it mostly considers processing the input speech in an offline manner via non-causal models (i.e., the model has access to the full input speech signal before performing separation). Specifically, State of The Art (SOTA) models such as the one proposed in [10, 14, 15, 16, 13] were developed using the inter- and intra-chunking operations together with bidirectional LSTMs. This makes the conversion to real-time and online-streaming evaluation challenging due to the dependencies in the latent representation structure. Figure 1 provides a visual description of the latent representation structure. Notice that the chunking operation involves overlapping between segments, which requires a buffering and a synchronization mechanism to execute the RNNs correctly. In addition, a proper historical context of chunks needs to be managed for the self-attention module.

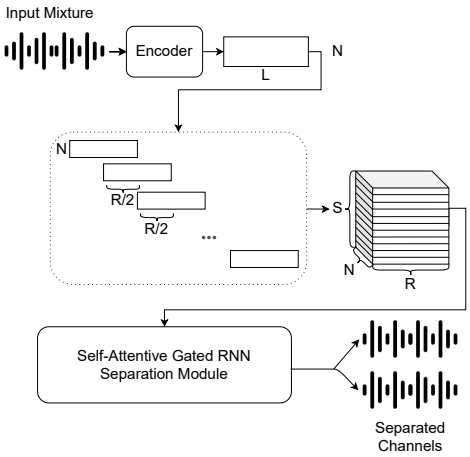


Figure 1: *An illustration of the separation model. Input audio is encoded into N dimensional frames, which next are grouped into segments of length R and chunked with an overlap of 50%. Chunks are processed by the Self-Attentive Gated RNN module, and their output is decoded into two separate speaker streams.*

In this work, we study and analyze the conversion of the self-attentive gated-RNN model proposed by [16] to a real-time and streaming mode, where we consider both monaural and binaural inputs and outputs. We evaluate the discrepancy between offline, causal, and online-streaming models and under anechoic, noisy, and noisy-reverberant settings. Our results shed light on the relative difference between online and offline models when performing separation. We observe a drop of $\sim$3dB on average in terms of Signal-to-Noise Ratio improvement over the mixture. Moreover, following our stateful implementation for the online separation model, we observe a drop of less than 0.8dB for monaural inputs and less than 0.3dB for binaural inputs while reaching a real-time factor of $\sim$0.65.

Related Work The task of online source separation was long studied under various settings [17, 18, 19, 20, 21, 7, 22, 23, 24, 25]. The authors of [18, 19] separating the input in a sliding

*equal contribution

window while leveraging the Expectation-Maximization algorithm to estimate the model parameters. This leaves an open question of *who will these separation models perform under the online setting?* To answer this question one must first design an online version of the models proposed by [10, 14, 15, 16].

The authors in [21, 7, 22] proposed causal and real-time separation models, however, their performance is not on par with the models proposed by [10, 15, 13]. In [24] the authors suggested a speaker-aware online separation method, in which the authors include an additional speaker identification loss. Although this method shows impressive results, it is orthogonal to our approach since it is based on the causal Conv-TasNet model. The authors in [20] presented neural network-based models for speaker separation, counting, and diarization for meeting analysis. However, the authors evaluated their method on a single channel and anechoic setting only. Recently, the authors of [23] proposed an online separation model under the multi-channel setting, while the authors in [25] suggested a modification of the TasNet model [21] for binaural inputs and outputs. In section 5 we empirically demonstrate that the proposed method is superior to this method under all recording conditions.

The rest of the paper is organized as follows: in section 2 we detail all the notations used throughout the paper, while in section **??** we provide a background description of the evaluated models. In section 4 and section 5 we present the proposed method and evaluate its performance against several baseline methods. Lastly, in section 6 we conclude while pointing out possible future work.

2. Problem Setting

Anechoic room. Consider a recording mixture of C different sources $\boldsymbol{s}^j \in \mathbb{R}^T$, where $j \in [1, \ldots, C]$ in an anechoic enclosure where the source length, T can vary. The mixed signal is therefore:

$$\boldsymbol{x} = \sum_{j=1}^{C} \alpha^j \cdot \boldsymbol{s}^j, \tag{1}$$

where α^j is the scaling factor of the j-th source. Although this model is commonly used to demonstrate separation abilities, anechoic noiseless environments are hard to find in the real world.

Noisy room. To better model real world condition, we additionally consider an additive background noise. As a results Equation Eq. (1) is modified to:

$$\boldsymbol{x} = \sum_{j=1}^{C} \alpha^j \cdot \boldsymbol{s}^j + \boldsymbol{n}, \tag{2}$$

where $\boldsymbol{n}$ is a non stationary additive noise in an unknown Signal-to-Noise Ratio (SNR). Such modeling better capture real world settings, however it assumes no reverberation.

Noisy reverberant room. Lastly, to simulate a real-world including reverberation an Acoustic Transfer Function (ATF) which relate the sources and the microphones is considered together with additive noise as follows:

$$\boldsymbol{x} = \sum_{j=1}^{C} \alpha^j \cdot \boldsymbol{s}^j * \boldsymbol{h}^j + \boldsymbol{n}, \tag{3}$$

where $\boldsymbol{h}^j$ is the ATF of the j-th source to the microphone.

Under all three cases, we focus on the fully supervised setting, in which we are provided with a training set $\mathcal{S} = \{\boldsymbol{x}_i, (\boldsymbol{s}_i^1, \cdots, \boldsymbol{s}_i^C)\}_{i=1}^m$, and our goal is learn a model that given an unseen mixture $\boldsymbol{x}$, estimates C separate channels, $\hat{\boldsymbol{s}}$. In this study, we evaluate both monaural and binaural speaker separation, under the monaural setting we maximize the widely used Scale-Invariant Signal-to-Noise Ratio (SI-SNR) [7] to the ground truth signals when considering reordering of the output channels $(\hat{\boldsymbol{s}}^{\pi(1)}, \cdots, \hat{\boldsymbol{s}}^{\pi(C)})$ for the optimal permutation π. The SI-SNR is defined as:

$$\text{SI-SNR}(\boldsymbol{s}^j, \hat{\boldsymbol{s}}^j) = 10 \log_{10} \frac{||\tilde{\boldsymbol{s}}^j||^2}{||\tilde{\boldsymbol{e}}^j||^2}, \tag{4}$$

where $\tilde{\boldsymbol{s}}^j = \frac{\langle \boldsymbol{s}^j, \hat{\boldsymbol{s}}^j \rangle \boldsymbol{s}^j}{||\boldsymbol{s}^j||^2}$ and $\tilde{\boldsymbol{e}}^j = \hat{\boldsymbol{s}}^j - \tilde{\boldsymbol{s}}^j$.

Under the binaural setting, we use the plain SNR (also with the optimal permutation) rather than the SI-SNR as the training objective. The rationale is that SI-SNR training cannot preserve the interaural cues in the binaural estimates, as the power scale of the estimated signals is insusceptible to training due to the scale invariance.

3. Background

Recall, our goal is to evaluate recent SOTA methods for blind source separation under the online setting. Here we briefly describe the models used in this study.

For the single-channel setting, we use a similar model to the one proposed by [14, 15] equipped with a self-attention mechanism, while in the binaural setting we follow the model proposed in [16]. Both models are generally comprised of three main components: *encoding and chunking*, *block processing*, and *decoding and overlap-add*. In the first stage, a time-domain input mixture is transformed into a set of overlapped chunks via encoding and chunking, which leads to a 3-D embedding tensor. Subsequently, the 3-D embedding tensor is passed into stacked self-attentive gated-RNN blocks to perform intra-chunk (local) and inter-chunk (global) modeling alternately and iteratively. The 3-D representation learned by the last RNN block is decoded and then transformed back to the time domain by an overlap-add operator.

3.1. Monaural Separation

Given a T-sample input waveform $\boldsymbol{x} \in \mathbb{R}^T$, an encoder is used to segment and encode $\boldsymbol{x}$ into L overlapped time frames with a frame size of P and a hop size of $P/2$, yielding a 2-D embedding $\mathbf{U} \in \mathbb{R}^{N \times L}$. Specifically, the encoder consists of a 1-D strided convolutional layer with N output channels, followed by a rectified linear activation function. We divide the time frames into S overlapped chunks with a chunk size of R and a hop size of $R/2$. These chunks are then concatenated into a 3-D embedding $\widetilde{\mathbf{W}} = [\mathbf{W}_1, \ldots, \mathbf{W}_S] \in \mathbb{R}^{N \times S \times R}$, where $\mathbf{W}_1, \ldots, \mathbf{W}_S \in \mathbb{R}^{N \times R}$ are the 2-D chunks. Subsequently, the 3-D embedding $\widetilde{\mathbf{W}}$ is fed into a series of B gated-RNN blocks, as proposed in [14].

Similar to [14, 15], we use a multi-scale loss for training, which necessitates producing a waveform estimate for each speaker after each block. We decode the output embedding of each block with a decoder, which comprises a parametric rectified linear function [26] followed by a 2-D 1×1 convolutional layer with $C \cdot N$ output channels. The decoded feature of size $CN \times S \times R$ is divided into C 3-D representations of size $N \times S \times R$, corresponding to the C speech sources. These 3-D

representations are transformed back to waveforms by two successive overlap-add operations at the chunk level and the frame level, respectively.

Similarly to [16] we apply a self-attention [27] mechanism before feeding the input into each gated-RNN block. We first divide a 3-D representation into a set of 2-D slices $\mathbf{Z} \in \mathbb{R}^{M \times N}$, where $M = R$ for intra-chunk modeling and $M = S$ for inter-chunk modeling. Each slice is linearly projected to a query matrix $\mathbf{Q}$, a key matrix $\mathbf{K}$ and a value matrix $\mathbf{V}$ by three different projection layers, where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{M \times D}$ and D is set to 64. We apply a scaled dot-product attention function:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = SoftMax(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D}})\mathbf{V}, \qquad (5)$$

where $SoftMax(\cdot)$ denotes the softmax function across columns. The output of the attention function is computed as a weighted sum of the values, where the weight assigned to each value is derived by measuring the similarities between the queries and the keys. Subsequently, all the attention output slices are merged and then linearly projected back to the size of the input 3-D representation. With a skip connection, this representation is concatenated with the input to the self-attention block and then projected back to the original size. The use of self-attention is motivated by its great success in improving separation and sequence modeling for binaural mixtures [16].

3.2. Binaural Separation

For binaural separation, we follow the same model as proposed in [16]. In which a *reference encoder* and a *non-reference encoder* are employed to process the binaural mixture waveforms. The resulting 2-D embeddings are concatenated and then linearly projected to the size of $N \times L$. Subsequently, we successively perform block processing, decoding, and overlap-add, akin to the monaural setting. Notice, the separation outputs always correspond to the reference ear. In order to get a multiple-inputs multiple-outputs system we alternately treating each ear as the reference. Specifically, the separation outputs for the left ear are obtained by treating the left ear as the reference ear and the right ear as the non-reference. The separation outputs for the right ear are obtained by swapping the inputs of the two ears. Note that the same system is used for separation in both channels. Such a cross-ear referencing strategy selects the target channel by exploiting discriminative information within the ordered pair of channels.

4. Method

Recall, our model is comprised of a 1-D convolutional encoder, a chunking operation, which converts the 2D input into a 3D tensor of overlapping segments, and a series of self-attentive gated RNNs. Originally, these models were designed to work in a non-causal manner, i.e., see the full input sequence in advance.

Converting the model into its causal version can be straightforward: i) converting the encoder into a causal convolution, ii) changing all RNNs into uni-directional and iii) replacing the self-attention with a causal self-attention, similarly to the one proposed in [28]. However, it remains unclear how to modify the chunking operation under real-time and online settings. Since the RNN blocks are being applied over both chucking dimensions (i.e., R and S) which consists of a 50% overlap, the RNN which processes the R dimension sees future context. Hence a synchronization scheme is needed to fully convert it into an online and real-time separation model. In the next subsections, we suggest both a *Stateless* and a *Stateful* approaches for operating the model in a real-time and streaming manner.

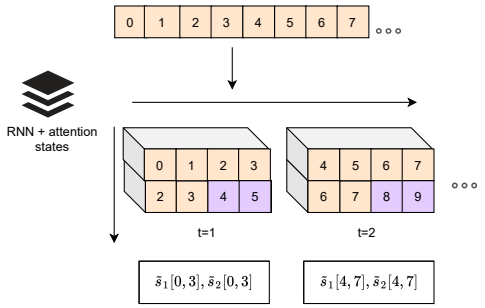


Figure 2: *An illustration of the stateful streaming mode, shown for simplicity with a segment size $R = 4$. Input audio is buffered until it reaches the length of $1\frac{1}{2} \times R$. Then a segment and an additional half-segment future context (squares [0-5] in illustration) are pulled from the buffer. These $1\frac{1}{2}\times$ segments are chunked and processed by the SAGRNN module. Processed chunks are merged together into an output segment (squares [0-3] in illustration).*

4.1. Stateless Mode

Our first evaluation uses a naive implementation of audio streaming in which the model does not store any state between processing audio segments, denoted as *stateless*. Under the stateless approach, the model is being fed with audio chunks, each of which contains a segment of audio, future context, and historical context. The historical context is crucial for better modeling long-range dependencies to improve the separation results.

In this mode, the model forward function is idempotent. Thus, no state is retained within neither the RNNs nor the processing of the segment (chunking, merging, etc.). The audio chunks are processed using a sliding window with a hop size of R. The future context equals $R/2$ to accommodate for chunks overlap, and we set the historical context to be 640ms. We experimented with several lengths of historical context and found 640ms to perform the best while maintaining a sensible RTF of close to real-time.

Despite its simplicity and ease of implementation, this approach wastes computation and produces channel estimation with significantly lower quality. Moreover, our results show that even with a large historical context of (>640ms), in which the Real Time Factor (RTF) is greater than 1 ($\sim$ 1.3), the results are still not on par with the causal, non-streaming model.

4.2. Stateful Mode

Next, we design the stateful-model-based approach. With it, audio segments are fed into the model in a streaming manner. The model maintains an internal state between each forward calls to keep track of historical context. We maintain a separate state for each of the RNNs and self-attention blocks. Notice, our GatedRNN block is composed of two separable RNN blocks operating on different dimensions. At inference time, the R dimension stays fixed while the S dimension varies as a function of the input length ($S \approx \lfloor 2L/R \rfloor - 1$). Hence, we need to maintain a state only for the RNNs which processing inputs over the S axis.

Recall, we reconstruct back the signal using an overlap-

and-add operation. Due to the 50% overlap between chunks, our method needs to store the future context of half a segment (i.e., $R/2$). As a result, to process segment i the first half of segment $i+1$ is buffered as the future context. Formally, to estimate the separated sources for segment i our model gets as input the i_{th} segment of size R, together with a future context of half a segment of size $R/2$, denoted by seg_i. The RNNs process seg_i and output $\tilde{s}_i$, which is followed by an overlap-and-add operation to estimate the waveform.

Next, to further process the $i+1_{th}$ segment accurately, our method keeps states for the RNN hidden states, and historical context for the self-attention. Notice, the future context from segment i is the first half of the segment $i+1$. A visual example can be found in Figure 2. For simplicity, we demonstrate the model's operation with segment size $R=4$. We buffer the input audio until it reaches the length of $1\frac{1}{2} \times R$. Then, we process the first segment and keep the future context in the buffer for further processing.

Using the stateful method, we demonstrate that results meet the real-time requirement of an RTF lower than 1 ($\sim$0.65) while preserving the audio quality almost intact, as tested under several settings and objective measurements.

4.3. Model latency

The latency of the streaming models depends on three terms: i) RTF; ii) processed segment size; iii) and future context. Formally, we measure the latency as follows,

$$Latency = R + RTF * R + FC, \quad (6)$$

where FC is the future context. Our chosen segment size corresponds to 64ms of audio at a sample rate of 8Khz, where on each model forward pass the model processes one segment of audio. The internal buffers of the model keep a future context that is defined to be 32ms (to account for future half-segments needed to process the current segment). Overall, the latency of the proposed method is on average 138ms, when considering RTF of 0.65 (64 + 64*0.65 + 32).

5. Results

We evaluated the proposed streaming implementation considering both monaural and binaural settings. For the monaural experiments, we used the wsj2-mix [2], WHAM! [29], and WHAMR! [30] as the anechoic, noisy, and noisy reverberant settings respectively.

Under the binaural setup, we used the noise-free and noisy datasets as suggested in [16]. To create the noisy-reverberant dataset, we generally followed the procedure of the noisy datasets, except that the simulated binaural room impulse responses (BRIRs), instead of the head-related impulse responses (HRIRs), were used to be convolved with the speech and noise signals to simulate room acoustics effects. The BRIRs were created in two steps: 1) room acoustics simulation and 2) HRIR convolution. Given a room and a source location, we simulated the sound field at the listener's position based on a combination of image-source and ray-tracing approaches [31, 32]. The image-source method (up to six orders of reflections) was used to simulate the sound pressure of the direct sound and the early reflections at the center location of the listener's head. Each incident sound wave was then convolved with the listener's HRIRs corresponding to the incident angle to compose the early portion of the binaural impulse response. The ray-tracing method was used to simulate the energy flow of the late reverberation (beyond six orders of reflections). The direction, timing, and energy information of the acoustic energy flow were used to create the late portion of the BRIR based on the head orientation and the averaged distance between the left and right ears. Ten thousand "shoebox" rooms were created for simulation, with each length, width, and height being randomly sampled between 2m and 10m. Frequency-independent absorption coefficients were randomly assigned to the walls. And we select a set of impulse responses that have the reverberation time uniformly distributed between 0.1s and 1s. In each room, the source and listener locations were also randomly sampled. Different HRIRs from the CIPIC HRTF Database [33] were randomly selected to generate BRIRs for each room.

Under all settings, we consider two speakers only in the input mixtures. Each dataset contains 20,000, 5,000, and 3,000 mixtures in the training, validation, and testing sets, respectively.

Table 1: *Results for single channel speaker separation using causal and non-causal models under various recording conditions. Results are reported in terms of SI-SNR improvement over the mixture in dB. We additionally compare results from non-streaming, stateful and stateless streaming modes.*

	# param.	Anechoic	Noisy	Noisy reverberant	Causal
Conv-TasNet [7]	5.1 M	15.3	12.7	8.3	✗
DPRNN [10]	3.6 M	18.8	13.9	10.3	✗
SAGRNN	7.6 M	19.7	15.2	12.2	✗
TasNet [21]	32.0 M	10.8	8.7	7.2	✓
Conv-TasNet [7]	5.1 M	10.6	8.6	7.2	✓
SAGRNNc	4.7 M	16.1	12.6	10.2	✓
SAGRNNc (*Stateless*)	4.7 M	9.5	7.3	7.0	✓
SAGRNNc (*Stateful*)	4.7 M	15.3	12.4	10.1	✓

5.1. Monaural Speaker Separation

We start by evaluating the proposed method for single channel separation. We compared our method against LSTM-TasNet [21] (causal), Conv-TasNet [7] (both causal and non-causal), Dual-Path RNN (DPRNN) [10] (non-causal), and GatedRNN [14] (both causal and non-causal). Table 1 summarizes the results.

Converting the SAGRNN model to be causal (denoted as SAGRNNc) yields a drop of 3.6dB, 2.6dB, and 2dB on the anechoic, noisy, and noisy-reverberant settings respectively in terms of SI-SNR improvement over the mixture. These results are superior by a significant margin to the causal baseline methods. Then, we evaluate the SAGRNN model under the online setting when considering both stateless and stateful modes. Results are presented on Table 1 (bottom).

Notice, we observe a large drop in performance under the stateless mode for all settings (e.g., 6.6dB decrease in performance for anechoic samples). These results suggest that the separation model does benefit from a large historical context. Moreover, the RTF of the stateless approach is on average $\sim$ 1.3, hence does not meet the real-time requirement. However, when considering the stateful approach, the gap between the streaming and non-streaming modes is significantly smaller (e.g., 0.8 dB decrease in performance for the anechoic samples) while its RTF is on average $\sim$0.65.

The discrepancy in performance between the offline and stateful evaluations is caused due to: (1) the self-attention layers look at a finite buffer of historical segments to process their dependencies with the current segment of audio (as opposed to looking at all past segments). (2) The overlap-and-add operation outputs a reminder of audio that is trimmed to fit the orig-

Table 2: *Binaural speaker separation results using causal and non-causal models under various recording conditions. Results for SDRi and SNRi are reported in dB while ESTOI results are reported in percentage.*

	SDRi	SNRi	ESTOI	PESQ	SDRi	SNRi	ESTOI	PESQ	SDRi	SNRi	ESTOI	PESQ	Causal
	Anechoic				Noisy				Noisy reverberant				
MIMO TasNet [25]	21.14	20.69	95.53	3.73	14.40	15.23	63.79	2.41	3.95	7.62	27.81	1.73	✓
Oracle MB-MVDR	17.13	10.44	95.77	3.66	4.98	4.90	42.71	1.79	1.74	4.01	29.44	1.79	✓
MIMO SAGRNN [16]	27.19	26.88	98.08	4.06	17.53	17.95	75.14	2.78	8.31	9.56	36.42	2.00	✗
MIMO SAGRNNc	25.73	24.12	97.2	3.86	15.04	15.91	67.11	2.41	6.37	8.70	30.51	1.85	✓
MIMO SAGRNNc (Stateless)	23.57	22.06	96.0	3.67	12.81	13.22	62.50	2.11	3.78	7.47	27.12	1.68	✓
MIMO SAGRNNc (Stateful)	25.63	24.11	97.0	3.85	14.73	15.77	66.52	2.35	5.41	8.20	29.55	1.79	✓

inal input audio length. The slight dislocation caused by the output of the overlap-and-add is negligible in an offline setting but becomes more apparent when iteratively processing small chunks of data.

5.2. Binaural Speaker Separation

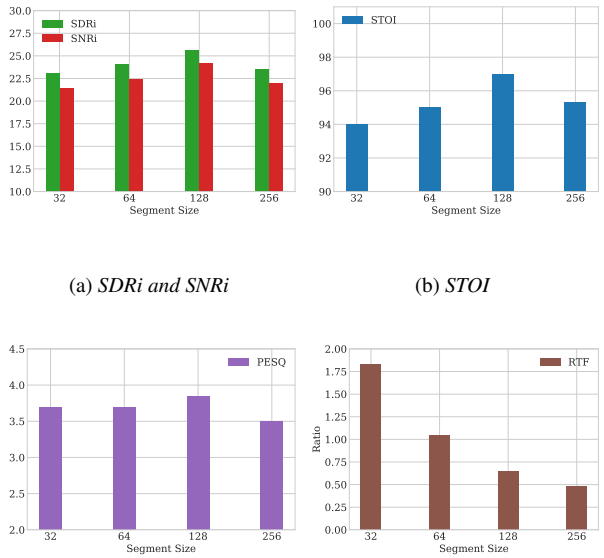
Next, we evaluated the causal SAGRNN model under the binaural setting. In the following, the input is a binaural signal and the output is a binaural estimate of the separated sources. We refer to this system as Multiple-Inputs Multiple-Outputs (MIMO). Results are summarizes in Table 2. We report SNR and Signal-to-Distortion Ratio (SDR) improvement over the mixture, together with Extended STOI (ESTOI) and PESQ. Results suggest that the causal SAGRNN is superior to the baseline methods under all settings and evaluation metrics. However, the gap between causal and non-causal models is smaller in the binaural setting than the gap in the single-channel setting. This may happen due to the multiple inputs, which provide additional tracking information to the model.

When considering an online evaluation, we observe a similar trend in which the stateless approach performs worse than the stateful approach, however, the gap is smaller than the one under the single-channel setting. Notice, under the stateful setting, the gap between streaming and non-streaming modes is negligible.

Interestingly, the binaural and monaural implementations show similar RTF rations - this is due to the model implementation utilizes GPU parallelism for the processing of the different audio channels, while the processed segment size stays the same in both settings.

The effect of segment size. Lastly, we analyze the effect of the segment size, R. Recall, as stated in equation Eq. (6), the latency is a function of the segment size. Hence, in order to better understand the effect of the segment size of the model performance and latency we trained several models where $R \in \{32, 64, 128, 256\}$ corresponding to latency of $[65.6, 85.9, 142.1, 323.1]$ milliseconds respectively. For this set of experiments, we trained a binaural separation model using the anechoic datasets and report SNR and SDR improvement over the mixture, ESTOI, PESQ, and RTF.

Results suggests that while setting $R = 128$ we reach the best performance in terms of SDRi, SNRi STOI, and PESQ. When considering RTF, $R = 256$ gets better real-time ratios, however this comes at the expense of separation performance. Notice, $R = \{32, 64\}$ does not meet the real time requirements and reaching RTF greater than 1. Figure 3 summarizes the results.



(a) *SDRi and SNRi* (b) *STOI*

(c) *PESQ* (d) *RTF*

Figure 3: *Results for different segment size values ($R \in \{32, 64, 128, 256\}$). Results are reported for the clean anechoic binaural speaker separation using the stateful streaming mode. We report SNR and SDR improment over the mixture, PESQ, STOI and RTF.*

6. Conclusion & Future Work

In this work, we studied the SAGRNN model under online and real-time settings. We report results for both monaural and binaural inputs under anechoic, noisy, and noisy-reverberant recording conditions, in which we explored both stateless and stateful modes. Our findings suggest that converting the SAGRNN to a causal model costs a drop of $\sim$3dB on average for a single channel and 0.8dB on average for the binaural setup. Moreover, when evaluating the models under the online setting, our empirical study suggests the stateful mode is superior to the stateless approach while reaching RTF of 0.65 on average.

Recall, our model has a latency of 138ms, this can be further reduced using a shorter future context. For future work, we would like to explore shortening the future context and processing time of an audio segment by exploring various segments and overlapping sizes. Additionally, improving the synchronization mechanisms between the processing done by the RNNs may decrease the overall model latency. Lastly, to support commodity hardware, future work will also include an online real-time CPU implementation of the model, where trade-offs between speed and quality will be analyzed.

7. References

[1] E. C. 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] 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*. IEEE, 2016, pp. 31–35.

[3] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 241–245.

[4] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 246–250.

[5] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2018, pp. 686–690.

[6] Z.-Q. Wang, K. Tan, and D. L. Wang, "Deep learning based phase reconstruction for speaker separation: A trigonometric perspective," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 71–75.

[7] 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.

[8] D. Stoller, S. Ewert, and S. 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, pp. 334–340.

[9] S. Venkataramani, J. Casebeer, and P. Smaragdis, "End-to-end source separation with adaptive front-ends," in *52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 684–688.

[10] 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*. IEEE, 2020, pp. 46–50.

[11] L. Zhang, Z. Shi, J. Han, A. Shi, and D. 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.

[12] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *arXiv preprint arXiv:2002.08933*, 2020.

[13] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 21–25.

[14] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," in *International Conference on Machine Learning*, 2020.

[15] S. E. Chazan, L. Wolf, E. Nachmani, and Y. Adi, "Single channel voice separation for unknown number of speakers under reverberant and noisy settings," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3730–3734.

[16] K. Tan, B. Xu, A. Kumar, E. Nachmani, and Y. Adi, "Sagrnn: Self-attentive gated rnn for binaural speaker separation with interaural cue preservation," *IEEE Signal Processing Letters*, 2020.

[17] R. Mukai, H. Sawada, S. Araki, and S. Makino, "Real-time blind source separation for moving speakers using blockwise ica and residual crosstalk subtraction," in *Proc. ICA*. Citeseer, 2003, pp. 975–980.

[18] M. Togami, "Online speech source separation based on maximum likelihood of local gaussian modeling," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 213–216.

[19] L. S. Simon and E. Vincent, "A general framework for online audio source separation," in *International conference on Latent Variable Analysis and Signal Separation*. Springer, 2012, pp. 397–404.

[20] T. von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, and R. Haeb-Umbach, "All-neural online source separation, counting, and diarization for meeting analysis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 91–95.

[21] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 696–700.

[22] S. Wang, "Online speaker separation using deep clustering," Master's thesis, 2019.

[23] J. Wu, Z. Chen, J. Li, T. Yoshioka, Z. Tan, E. Lin, Y. Luo, and L. Xie, "An end-to-end architecture of online multi-channel speech separation," *arXiv preprint arXiv:2009.03141*, 2020.

[24] 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 *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6379–6383.

[25] C. Han, Y. Luo, and N. Mesgarani, "Real-time binaural speech separation with preserved spatial cues," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 6404–6408.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[28] S. Merity, "Single headed attention rnn: Stop thinking with your head," *arXiv preprint arXiv:1911.11423*, 2019.

[29] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "Wham!: Extending speech separation to noisy environments," *arXiv preprint arXiv:1907.01160*, 2019.

[30] M. Maciejewski, G. Wichern, E. McQuinn, and J. Le Roux, "Whamr!: Noisy and reverberant single-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 696–700.

[31] J. H. Rindel, "The use of computer modeling in room acoustics," *J. Vibroeng. 3, 41–72*, 2000.

[32] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulations and array processing algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[33] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The cipic hrtf database," in *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575)*. IEEE, 2001, pp. 99–102.