HYBRID SUBBAND-FULLBAND GATED CONVOLUTIONAL RECURRENT NEURAL NETWORK FOR MULTICHANNEL SPEECH ENHANCEMENT

Benjamin Stahl, Alois Sontacchi

Institute of Electronic Music and Acoustics
University of Music and Performing Arts
Graz, Austria

ABSTRACT

We present a novel method combining fullband and subband gated convolutional recurrent neural networks for multichannel deep neural speech enhancement to the **Sp**eech **E**nhancement for **A**ugmented **R**eality (SPEAR) challenge. To target the challenge's task of direction-dependent enhancement of a single speaker, a direction-dependent feature extraction combined of matched filters and a maximum-directivity beamformer was employed. When evaluated on the SPEAR challenge development datasets, our method achieved clear improvements in the PESQ and siSDR metrics over a maximum-directivity beamformer baseline.

1. BACKGROUND

1.1. SPEAR Challenge Task

In the **Sp**eech **E**nhancement for **A**ugmented **R**eality (SPEAR) challenge, the task is to remove background noise and speech from all but one speaker from signals recorded by the microphones of a head-mounted device [1]. The relative direction of this "desired" speaker is given. The challenge provided training and development datasets based on the Easycom dataset [2]. The head mounted device in these datasets has four microphones on a pair of glasses and two microphones positioned in the ears.

1.2. Convolutional Recurrent Neural Network

The convolutional recurrent neural network (CRN)[1] is a common architecture in short-time-Fourier-transform (STFT)-domain audio processing. It has been applied to speech enhancement, acoustic echo cancellation, and active noise control [3, 4, 5, 6]. As shown in Fig. 1, it is comprised of an encoder with multiple convolutional layers, a corresponding convolutional decoder, inner recurrent neural network (RNN), and skip connections between corresponding convolutional encoder and decoder layers.

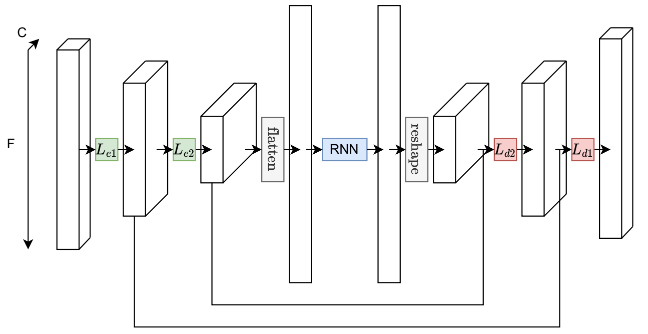

Fig. 1. Exemplary two-layer convolutional recurrent neural network. Time dimension is not visualized.

The CRN's input is three-dimensional (omitting the batch dimension), with shape $C_{\text{inp}} \times T \times F$, where C_{inp} denotes the number of input channels, T denotes the number of time frames, and F denotes the number of frequency bins. The encoder and decoder convolutional kernels are two dimensional, with kernel shape $(K_t \times K_f)$ and convolutions carried out along the time and frequency dimensions. Convolutions are carried out in a causal way along the time dimension, if the neural network needs to be causal. Along the frequency dimension, a stride > 1 is typically employed, such that the frequency dimension is progressively shrinked in the encoder and re-inflated in the decoder, which applies "transposed convolutions" along the frequency dimension. Before being processed by the RNN, the output of the innermost encoder layer is flattened along the channel and frequency dimensions. Thus, the RNN processes a representation of the full input. With the skip connections between encoder and decoder, the outputs of the encoder layers are concatenated to the input of the corresponding decoder layers along the channel dimension.

1.3. Gated Convolutional Layers

Gated convolutional layers have been proposed as a modification to standard convolutional layers, allowing for additional control of information flow [7, 8, 9]. A gated convolutional layer is shown in Fig. 2. Next to the main convolutional layer,

[1] Note that in this work, although there are other convolutional recurrent architectures, we particularly refer to the convolutional recurrent U-net architecture by the term "convolutional recurrent neural network".

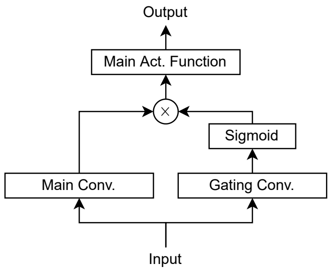

Fig. 2. Gated convolutional layer

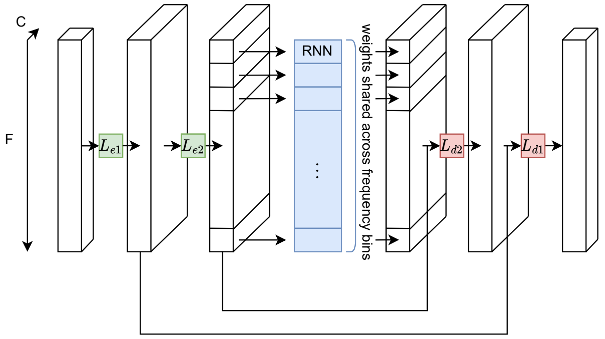

Fig. 3. Exemplary two-layer inplace convolutional-recurrent neural network. Time dimension is not visualized.

another convolutional layer with a sigmoid activation function generates a gating output, which is multiplied to the output of the main convolutional layer before it passes the activation function.

1.4. Inplace Convolutional Recurrent Neural Network

The processing of a joint representation of all frequency bins in the RNN in (G)CRNs has been shown to be be suboptimal in multichannel speech enhancement [9]. Inplace (gated) convolutional recurrent neural networks (I(G)CRNs) therefore use a different approach: With the encoder and decoder, the frequency dimension is no longer shrinked and re-inflated but is held constant over all layers, which is achieved by using stride $= 1$ along the frequency dimension and padding the frequency dimension with zeros on each side. The input of the recurrent layers therefore has the size $C_{\mathrm{N}} \times T \times F$, where C_{N} is the number of output channels of the innermost encoder layer. A RNN with input and output dimension C_{N} separately processes each element along the frequency dimension of the encoder output. As in the convolutional layers, weights are shared across frequency bins (this can be implemented by flattening the encoder output along the batch and frequency dimensions, creating a RNN batch size of $B \cdot F$, where B is the input batch size). Figure 3 shows the ICRN architecture. It can be considered a convolutional variant of [10], as it computes an output for a center frequency bin from a region of the input around this bin. The width of this frequency region, which we refer to as subband receptive field width, can be computed as $2D \cdot (K_f - 1) \cdot R$, where D is the number of layers in the encoder and decoder, K_f is the kernel size along the frequency dimension, and R is the frequency resolution, i.e., width of each STFT bin.

2. MODEL DESCRIPTION

Figure 4 shows the signal flow with our proposed method.

2.1. Direction-Dependent Feature Extraction

We denote the STFT of the M-channel microphone input by $\mathbf{x}_{t,f}$, where t and f denote frame and frequency index, respectively. Furthermore, we denote the relative source direction (azimuth and polar angles) by $\boldsymbol{\theta}_t$. Using $\boldsymbol{\theta}_t$, the nearest neighbour device transfer function $\mathbf{d}_{t,f}$ is selected from the provided set of acoustic transfer functions (ATFs)[2]. Matched filters $\mathbf{W}_{\mathrm{MF},t,f}$ are computed from $\mathbf{d}_{t,f}$ as

$$\mathbf{W}_{\mathrm{MF}_{t,f}} = \operatorname{diag}\left\{ \frac{\mathbf{d}_{t,f}}{||\mathbf{d}_{t,f}||^2} \right\}. \tag{1}$$

Additionally, a maximum-directivity beamformer is computed as

$$\mathbf{w}_{\mathrm{MaxDir}_{t,f}} = \frac{\boldsymbol{\Omega}_f^{-1}\mathbf{d}_{t,f}}{\mathbf{d}_{t,f}^H \boldsymbol{\Omega}_f^{-1}\mathbf{d}_{t,f}}, \tag{2}$$

where $\boldsymbol{\Omega}_f^{-1}$ is the inverse of the (diagonal-loaded) spherical diffuse-field covariance matrix of the microphone array. Note that we omit the indices t and f from here.

We denote the outputs of the filters described above as

$$\mathbf{y}_{\mathrm{MF}} = \mathbf{W}_{\mathrm{MF}}^H \mathbf{x} \tag{3}$$

and

$$y_{\mathrm{MaxDir}} = \mathbf{w}_{\mathrm{MaxDir}}^H \mathbf{x}. \tag{4}$$

The real and imaginary parts of $\mathbf{y}_{\mathrm{MF}}$ and y_{MaxDir} are concatenated, resulting in a feature vector with $2 \cdot (M+1)$ channels. To facilitate convergence, the feature vectors are normalized by the square root of the per-frequency-bin signal variances σ_f estimated from $\mathbf{y}_{\mathrm{MF}}$ by averaging over all recordings in the training dataset.

2.2. Hybrid Subband-Fullband Processing

In [11], a hybrid fullband-subband architecture outperformed non-hybrid fullband and subband architectures that were upscaled for a fair comparison. In the fullband-subband architecture, the input features first pass a fullband network, whose output is concatenated with the input features and then processed by the subband network. In our proposed method, we reversed

[2] We centered the phase of the device acoustic transfer functions around a linear phase offset of 0.7 ms.

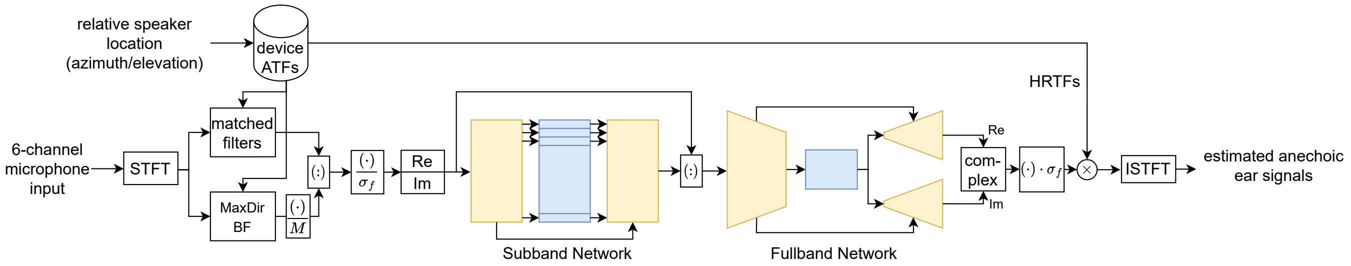

Fig. 4. Signal flow diagram of proposed method

the order of fullband and subband networks, as we observed slightly better performance with the subband → fullband order. We used the IGCRN and GCRN architectures described in Section 1 for the subband and fullband networks, respectively. With the fullband network, two decoders respectively output the real / imaginary part of an enhanced single-channel signal. Finally, the normalization factors σ_f are re-multiplied on this single-channel signal.

2.3. Spatialization

The device ATFs of channel five and six corresponding to $\boldsymbol{\theta}_t$, i.e., head-related transfer functions are multiplied on the enhanced single-channel signal to create left and right ear signals, which are passed through an inverse STFT in order to create the final processed time-domain ear signals.

2.4. Cost Function

The STFT of the processed time-domain ear signals is denoted by $\widehat{y}$, and the STFT of the anechoic reference ear signals is denoted by y. As in [12], both $\widehat{y}$ and y are normalized by the reference active speech root-mean-squared level. We employ the complex compressed mean-squared error (CCMSE) [13] loss function, which is computed as follows:

$$\mathcal{L}_{\text{CCMSE}}(y,\widehat{y}) = (1-\alpha)\left\langle \left(|y|^c - |\widehat{y}|^c\right)^2 \right\rangle + \alpha \left\langle \left| |y|^c \cdot e^{j\angle y} - |\widehat{y}|^c \cdot e^{j\angle \widehat{y}} \right|^2 \right\rangle, \quad (5)$$

where $c < 1$ is a compression exponent, α is a phase-sensitivity factor, and $\langle\cdot\rangle$ denotes averaging over time index, frequency index, and left/right ear signals.

2.5. Model Hyperparameters and Training Procedure

Our proposed method operates at a sampling rate of 16000 Hz; therefore, all microphone and reference signals are down-sampled from the original sampling rate of 48000 Hz. In processing, we use a STFT with DFT size and window length 508 (= 31.75 ms), hop size 254, and a $\sqrt{\text{Hann}}$ window in both the STFT and the inverse STFT. We use all six microphones of the device ($M = 6$).

The subband network consists of four gated convolutional layers in the encoder and four in the decoder, which all have 75 output channels, a kernel size of 1×5, and an ELU activation function, except for the last decoder layer, which has 2 output channels and no activation function. The subband receptive field width is 1008 Hz. The RNN within the subband network is a two-layer gated recurrent unit (GRU) with an input and output size of 75. The fullband network also comprises 4 gated convolutional layers in the encoder and decoder. To keep the model compact, we chose to keep the number of channels in each layer constant at 64 in the fullband network. The kernel size is 2×3 and the stride along the frequency dimension is 2. The RNN within the fullband network is a two-layer Group GRU with an input and output size of 960, four groups, and representation rearrangement between layers. The number of output channels in the last layer in each decoder is 1, with no output activation function used.

The cost function STFT uses a DFT size and window length of 400 (= 25 ms), hop size 200, and a Hann window. The cost function parameters are set at $c = 0.3$ and $\alpha = 0.3$. To train the model, we used the synthetic training datasets D2, D3, and D4. We define one epoch as training on a 4-second snippet of each of the 1899 60-second reference files in these datasets. The snippets are not selected entirely randomly but are constrained to contain at least 60% ”active speech”. In this process, ”active speech” is defined as ”any speaker talking” in 30% of cases and ”target speaker talking” in 70% of cases.

We implemented our proposed subband-fullband model using the Pytorch Python package. The model was trained for 2500 epochs with a batch size of 2. The Adam optimizer with a constant learning rate of 10^{-4} was used. Every 50 epochs, the model was validated on the development datasets D2, D3, and D4, from which we processed all full recordings[3]. PESQ and siSDR metrics were computed on the segments defined in the provided segments file. The best epoch was selected based on the average PESQ.

[3] For memory reasons, we had to process the array signals in chunks of 10 seconds (except for the last chunk, which was shorter) overlapping by 1.5 seconds. This also applies to the submitted processed evaluation data.

Table 1. Metrics for unprocessed signals, maximum-directivity beamformer baseline, and proposed method on development datasets; PESQ is given in MOS points, siSDR is given in dB.

	dataset D2		dataset D3		dataset D4	
	PESQ	siSDR	PESQ	siSDR	PESQ	siSDR
unproc.	1.09	-12.3	1.12	-14.3	1.12	-11.0
baseline	1.17	-5.5	1.20	-9.0	1.14	-6.0
proposed	**1.85**	**5.2**	**1.82**	**2.3**	**1.66**	**4.9**

2.6. Compared Methods

In addition to the proposed method, we evaluated metrics on the maximum-directivity beamformer baseline, which was part of the SPEAR challenge's supplementary materials[4]. This baseline uses a sampling rate of 48000 Hz, a window length of 768 samples (= 16 ms), and a hopsize of 384 samples.

2.7. Processing Delay and Computational Cost

Assuming that audio in-/output buffering is done in blocks of length <hopsize>, our method has a processing delay of window length $+$ hopsize $= 47.6\,\mathrm{ms}$. The hybrid subband-fullband neural network has 4.12 million parameters and a computational cost of $12.95 \cdot 10^9$ multiply-accumulate operations (MACs) per second.

3. RESULTS ON DEVELOPMENT DATASETS

Table 1 shows the average Perceptual Evaluation of Speech Enhancement (PESQ) [14] and scale-invariant signal-to-distortion ratio (siSDR) [15] metrics for the synthetic datasets D2, D3, and D4. As can be seen, the proposed method clearly improves on the maximum-directivity beamformer baseline.

4. REFERENCES

[1] Pierre Guiraud, Sina Hafezi, Patrick A. Naylor, Alastair H. Moore, Jacob Donley, Vladimir Tourbabin, and Thomas Lunner, "An introduction to the speech enhancement for augmented reality (spear) challenge," in *2022 International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2022, pp. 1–5.

[2] Jacob Donley, Vladimir Tourbabin, Jung-Suk Lee, Mark Broyles, Hao Jiang, Jie Shen, Maja Pantic, Vamsi Krishna Ithapu, and Ravish Mehra, "Easycom: An augmented reality dataset to support algorithms for easy communication in noisy environments," *arXiv preprint arXiv:2107.04174*, pp. 1–9, 2021.

[3] Ke Tan and DeLiang Wang, "Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6865–6869.

[4] Sebastian Braun, Hannes Gamper, Chandan K.A. Reddy, and Ivan Tashev, "Towards efficient models for real-time deep noise suppression," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 656–660.

[5] Hao Zhang, Ke Tan, and DeLiang Wang, "Deep learning for joint acoustic echo and noise cancellation with nonlinear distortions.," in *Interspeech*, 2019, pp. 4255–4259.

[6] Hao Zhang and DeLiang Wang, "A deep learning approach to active noise control.," in *Interspeech*, 2020, pp. 1141–1145.

[7] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al., "Conditional image generation with PixelCNN decoders," *Advances in neural information processing systems*, vol. 29, pp. 1–9, 2016.

[8] Ke Tan and DeLiang Wang, "Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 380–390, 2020.

[9] Jinjiang Liu and Xueliang Zhang, "Inplace Gated Convolutional Recurrent Neural Network for Dual-Channel Speech Enhancement," in *Proc. Interspeech 2021*, 2021, pp. 1852–1856.

[10] Xiaofei Li and Radu Horaud, "Narrow-band deep filtering for multichannel speech enhancement," *arXiv preprint arXiv:1911.10791*, pp. 1–13, 2019.

[11] Xiang Hao, Xiangdong Su, Radu Horaud, and Xiaofei Li, "Fullsubnet: A full-band and sub-band fusion model for real-time single-channel speech enhancement," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6633–6637.

[12] Sebastian Braun and Ivan Tashev, "Data augmentation and loss normalization for deep noise suppression," in *Speech and Computer: 22nd International Conference, SPECOM 2020, St. Petersburg, Russia, October 7–9, 2020, Proceedings 22*. Springer, 2020, pp. 79–86.

[13] Sebastian Braun and Ivan Tashev, "A consolidated view of loss functions for supervised deep learning-based speech enhancement," in *2021 44th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2021, pp. 72–76.

[4] github.com/ImperialCollegeLondon/spear-tools

[14] 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 *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, 2001, vol. 2, pp. 749–752 vol.2.

[15] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey, "SDR — half-baked or well done?," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.