Hybrid neural networks for on-device directional hearing

Anran Wang[1], **Maruchi Kim**[1], **Hao Zhang**[2], **Shyamnath Gollakota**[1]
[1]University of Washington, [2]ETH Zürich
{anranw,mkimhj,gshyam}@cs.washington.edu, hao.zhang@inf.ethz.ch

Abstract — On-device directional hearing requires audio source separation from a given direction while achieving stringent human-imperceptible latency requirements. While neural nets can achieve significantly better performance than traditional beamformers, all existing models fall short of supporting low-latency causal inference on computationally-constrained wearables. We present Hybrid-Beam, a hybrid model that combines traditional beamformers with a custom lightweight neural net. The former reduces the computational burden of the latter and also improves its generalizability, while the latter is designed to further reduce the memory and computational overhead to enable real-time and low-latency operations. Our evaluation shows comparable performance to state-of-the-art causal inference models on synthetic data while achieving a 5x reduction of model size, 4x reduction of computation per second, 5x reduction in processing time and generalizing better to real hardware data. Further, our real-time hybrid model runs in 8 ms on mobile CPUs designed for low-power wearable devices and achieves an end-to-end latency of 17.5 ms.

Introduction

Directional hearing is the ability to amplify speech from a specific direction while reducing sounds coming from other directions. This has multiple applications ranging from medical devices to augmented reality and wearable computing. Directional hearing aids can help individuals with hearing impairments who have increased difficulty hearing in the presence of noise and interfering sounds (Doclo et al. 2010; Brayda et al. 2015). It can also be combined with augmented reality headsets to customize the sounds and noises from different directions, e.g., sensors like gaze trackers can enable a wearer to be in a noisy room and amplify the speech from a specific direction, simply by looking at the source.

For decades, the predominant approach to achieving this goal was to perform beamforming (Krim and Viberg 1996; Brayda et al. 2015; Chhetri et al. 2018). While these signal processing techniques can be computationally light-weight, they have a limited performance (Souden, Benesty, and Affes 2010; Kumatani et al. 2012). Recent work has shown that neural networks achieve exceptional source separation in comparison (Luo and Mesgarani 2019; Jenrungrot et al. 2020) but are computationally expensive and to date, cannot run on-device on wearable computing platforms.

Directional hearing applications however impose stringent computational, real-time and low-latency requirements that are not met by any existing source separation networks. Specifically, compared to other audio applications like teleconferencing where latencies on the order of 100 ms are adequate, directional hearing requires real-time audio processing with much more stringent latency requirements. For example, medical hearing aid research shows that we need a latency less than 20 ms to be tolerable (Stone and Moore 1999). This is also true for augmented reality applications so that any modified sound sources are correctly synced, both visually and audibly, in the brain (Gupta et al. 2020).

These stringent low-latency constraints are challenging to meet on wearable and medical devices. Directional hearing requires not only processing the continuous audio input stream, but also generating a continuous output stream within these real-time constraints. While powerful GPUs and specialized inference accelerators (e.g., TPU) can speed up the network run-time (Wang, Wei, and Brooks 2019), they are usually not available on a wearable device given their power, size and weight requirements. In fact, even the CPU capabilities and memory bandwidth available on wearables can be significantly constrained even compared to smartphones. For example, the iPhone 12 CPU is more than 10 times faster than that used in Google glasses and Apple watch. Offloading computation to other devices (e.g., smartphone) is not an option given the additional wireless roundtrip latency on the order of tens of milliseconds.

In this paper, we show for the first time that real-time directional hearing using deep learning can be achieved on computationally-constrained platforms. Instead of designing an end-to-end neural net to perform the task, we create a hybrid model that combines lightweight beamforming algorithms with neural networks. Our key insight is that the beamforming algorithms can provide spatial hints to a neural net that can drastically reduce the network complexity and its computational cost while achieving similar source separation performance to state-of-the-art causal neural networks that are computationally expensive. At a high level, while neural networks are a powerful tool to approximate functions, approximating beamforming functions like matrix inversion can increase the network complexity making them computationally expensive. However, traditional beamformers can provide useful theoretically-derived spatial hints to the neural net, in a computationally inexpensive manner.

We present the network architecture shown in Fig. 2 where we first input the signals from the multiple microphones into three different beamformers. The output of the beamformers along with the original signals from the multiple microphones is then fed into a causal neural net model that is optimized for memory overhead and inference time on mobile CPUs. Specifically, we use complex tensors throughout our network to reduce half of our model size while achieving a comparable accuracy. Compared to real-valued networks, complex representation also restricts the degree of freedom of the parameters by enforcing correlation between the real and imaginary parts, which enhances the generalization capacity of the model since phase encodes essential spatial information. We also design a combination of dilated and strided complex convolution stacks to reduce memory footprint and memory copy per time step while keeping a large receptive field. Finally, we simplify the temporal convolutional network to run more efficiently.

Compared to state-of-the-art causal models, we are able to achieve comparable separation performance with a 5x reduction of model size, 4x reduction of computation, and 5x reduction of processing time, enabling it to run in real-time on a CPU suitable for wearable devices with low latency. We also evaluate our model trained entirely on simulated data on real data recorded in conference rooms using a smart glasses prototype with a custom six-microphone array and a gaze tracker, which achieves real-time, gaze-controlled directional hearing with an end-to-end latency of 17.5 ms.

Related Work

Beamformers based on statistical signal processing. Beamforming techniques are designed to combine multichannel sensor signals to achieve directionality. Linear filters can statistically create constructive interference at the direction of interest and destructive interference elsewhere. Non-adaptive beamformers such as Barlett (delay-and-sum) beamformers and superdirective beamformers construct a constant linear filter applied to the signal (Krim and Viberg 1996). Adaptive beamformers such as minimum-variant distortionless-response (MVDR) beamformers and linearly constrained minimum variance (LCMV) beamformers additionally utilize spatial information from the mixture signal to inform the filter construction (Souden, Benesty, and Affes 2010). While they can be computationally inexpensive, their separation result is limited since they use the spatial cues but do not efficiently capture the acoustic cues.

Blind source separation. Another classical problem formulation is blind source separation where each channel receives a different unknown linear combination of a few independent sound sources (Comon and Jutten 2010). Without spatial hints such as directions, the problem formulation is often under-deterministic but a few spatial clustering methods such as independent component analysis (Haykin and Chen 2005) and Gaussian mixture model (Higuchi et al. 2017) can obtain a solution assuming a small number of sound sources. Recently, neural network architectures have been proposed to achieve blind source separation. Frequency-domain approaches learn the frequency-time mask for each sound source that is applied to the mixture spectrogram (Chen et al. 2018a; Hu et al. 2020). The spectrogram as input and output makes it inefficient for use in our application since it requires lookahead of tens of milliseconds. Time-domain approaches such as Demucs (Défossez, Synnaeve, and Adi 2020), TasNet (Luo and Mesgarani 2018) FasNet (Luo et al. 2019), TAC (Luo et al. 2020), Conv-TasNet (Luo and Mesgarani 2019) and its variants(Gu et al. 2019b; Défossez et al. 2019; Luo et al. 2020; Han, Luo, and Mesgarani 2020) optimize for the learnt filters that convolve with the mixture signals to separate each sound source. While time-domain approaches allow causal construction and more effective separation, they are not designed to match directions with each separated signal from the mixture, and the computation grows exponentially with the number of sources. (Jenrungrot et al. 2020) simultaneously separates each sound source as well as identify their directions; their model however is not causally constructed. They are also an overkill for use with directional hearing since unlike blind source separation it only requires separating the speech from a specific direction.

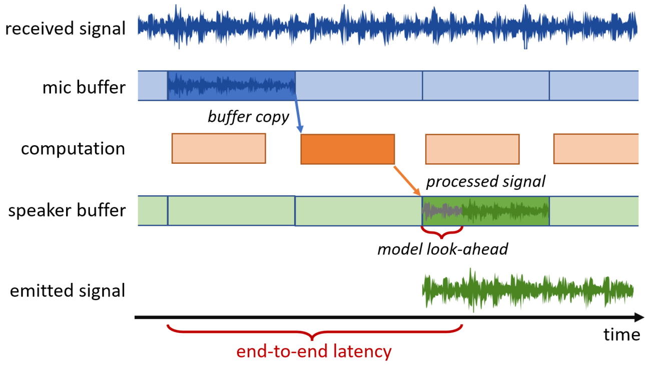
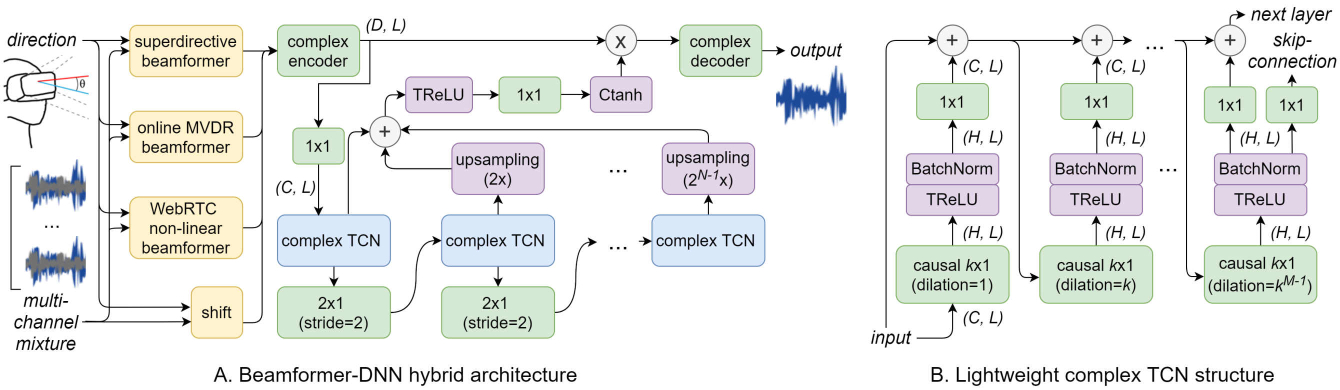


Figure 1: End-to-end latency for real-time hearing enhancement

Neural beamformers. To address the specific problem where the direction is provided as input, beamformer designs have been proposed using neural networks. (Chen et al. 2018b) presents a multi-pass bi-directional LSTM network using spectral, spatial, and angle features. Similarly, (Gu et al. 2019a) design a LSTM network on the spectrogram and attention mechanisms. Neither of these networks are causal in structure. (Gu and Zou 2020) extends Conv-TasNet (Luo and Mesgarani 2019) by feeding spatial features along with the first channel and achieves better results than LSTM-based methods, (Qian et al. 2018) uses a combination of beamformers and neural networks to enhance speech. They are computationally expensive and does not meet the delay requirements of mobile CPUs. (Fedorov et al. 2020) achieves low-power speech enhancement using LSTM, but it is not for multichannel source separation.

Improving MVDR with neural nets. (Zhang et al. 2021; Xu et al. 2020; Xiao et al. 2017; Tammen, Fischer, and Do-



A. Beamformer-DNN hybrid architecture

B. Lightweight complex TCN structure

Figure 2: The architecture of the HybridBeam system. (A): the end-to-end network diagram. (B): the structure of the simplified temporal convolutional network (TCN).

clo 2019) replace matrix inversion and PCA within a MVDR beamformer with a neural net. We take the inverse approach where we utilize the MVDR output for more efficient neural net feature extraction and design. It is also noteworthy that the structures of these prior designs are not causal in nature. Further, they are computationally expensive and can not run in real-time on a mobile CPU. Our joint beamformer-neural net approach instead reduces the complexity of the neural net using the features from the beamformers.

Method

The problem of real-time direction hearing can be formulated as follows. Say, we have N sound sources $s_{1..N}$ emitted from angles $\theta_{1..N}$ with respect to an array with c microphones. The signal received by the i^{th} microphone is,

$$y_i(t) = \sum_{j=1}^{N} \sum_{\tau=-\infty}^{0} H_{i,j}(\tau) s_j(t-\tau) + N(t)$$

Here, $N(t)$ is random noise and $H_{i,j}$ is the impulse response associated with sound source j and microphone i that captures multi-path and reverberations. At a given time t and a known θ_k, our goal is to estimate the acoustic signal, $s_k(t)$, emitted from the direction θ_k, given $y(t-W)\cdots y(t+L)$, where W is the reception field, and L is a small look-ahead.

Latency Requirements

Fig. 1 shows the composition of various sub-components that contribute to the latency. First, the sound signals get sampled by the multiple microphones and fed into a memory buffer. When the buffer is full, the data in the buffer is then processed by a computation program that consists of our neural network and signal processing techniques. The result, i.e., the speech from a specific direction, is then transferred to be played back through the speakers.

Consequently, to reduce the end-to-end latency to less than 20 ms, we should 1) reduce the buffer size; 2) minimize the processing time; and 3) reduce the look-ahead duration for the model. Reducing all these parameters while achieving good performance is challenging for multiple reasons. First, a very short buffer size (say 3 ms) may cause jitters due to the timing fluctuation in the operating system scheduling. A smaller buffer size also inversely increases with the frequency of the computation calls, each of which involves a constant overhead. Second, to ensure real-time operation, the computation block has to process the acoustic data from each buffer block within the duration of the block. That is, the computational time to process a 8 ms buffer should be less than 8 ms. Neural networks however are known for their heavy computational requirements, and none of existing models are designed for such a small end-to-end latency and real-time signal processing on computationally constrained CPUs. Third, a large look-ahead can improve source separation but also increases latency.

HybridBeam Architecture

Fig. 2 shows the overall architecture of our hybrid model. The input multi-channel signals from the microphone array and the target angle θ_k are first passed to three lightweight beamformers that result in three different versions of the beamformed signals. These beamformed signals are concatenated with the original multi-channel signals and fed into our neural network that is designed to output the separated acoustic signal from the target direction θ_k.

Prebeamforming. We use the following beamformers to extract features: a) superdirective beamformer (Krim and Viberg 1996) that is optimized under diffused noise; b) online adaptive MVDR beamformer (Habets et al. 2010) that extracts the spatial information from the past to suppress noise and interference; and 3) WebRTC non-linear beamformer (Kleijn 2016) that enhances a simple delay-and-sum beamformer by suppressing the time-frequency components that are more likely noise or interference. These three statistical beamformers span the different classes of beamforming techniques from non-adaptive, adaptive and non-linear approaches. As a result, they provide a diversity of spatial information as input to the neural network. Moreover, they are all computationally efficient — it takes 0.8ms to run all three beamformers for a 8 ms signal block on a mobile CPU.

Additionally, the input channels are shifted to aim at the input direction, so that each channel samples the direct path of the signal at the same time in the far-field:

$$\hat{y}(f) = y(f)\exp(j2\pi f(t_i(\theta) - t_0(\theta)))$$

$t_i(\theta)$ is the time-of-arrival from direction θ on mic i. These shifted channels along with the output of the beamformers are concatenated together and fed into a neural network.

Neural Network Model. Our neural net is inspired by time-domain models like Conv-TasNet (Luo and Mesgarani 2019) and has a linear encoder, a linear decoder and a separator module, all with 1D convolutional layers. However, we make significant modifications to the network to reduce its memory footprint, memory copy overhead and to be computational lightweight that we describe in detail below.

Complex Tensor Representation. We use complex tensors throughout our network to halve our model size (each parameter can be represented as $[R, -I; I, R]$, instead of full 2×2 matrices.) while achieving a comparable accuracy. Complex representation is a powerful tool for acoustic signal processing. For example, complex multiplication capture rotation in the complex domain and can easily manipulate the signal phase. Thus, complex neural networks are found to be more effective for applications such as wireless communication (Marseet and Sahin 2017) and noise suppression (Hu et al. 2020; Bassey, Qian, and Li 2021). Compared to real-valued networks, complex representation also restricts the degree of freedom of the parameters by enforcing correlation between the real and imaginary parts, which enhances the generalization capacity of the model in other applications. Besides the benefit of reduced model size, complex representation can be especially important for beamforming since phase encodes essential spatial information.

Fully-complex neural networks however lack the capability to efficiently approximate *conjugate* operation and *phase scaling* where the phase of a complex number gets multiplied by a constant. To mitigate this, we insert an additional component-wise operation before each $\mathbb{C}\mathbf{ReLU}$ activation. We call them together a new $\mathbf{TReLU}$ activation. Specifically, we define $\mathbf{TReLU}(\mathbf{x}_{c,t})$ as follows:

$$ReLU(h_c^{(rr)}\Re(\mathbf{x}_{c,t}) + h_c^{(ri)}\Im(\mathbf{x}_{c,t}) + b_c^{(1)})$$
$$+jReLU(h_c^{(ri)}\Re(\mathbf{x}_{c,t}) + h_c^{(ii)}\Im(\mathbf{x}_{c,t}) + b_c^{(2)})$$

Here $\mathbf{x}$ is the complex input of the activation function, c, t are the channel and time indices, respectively, and h, b are parameters to train. Intuitively, the operation linearly transforms the 2D complex space that can simulate both conjugate and phase scaling, and then $ReLU$ activation is performed on real and imaginary parts independently. This is equivalent to the scaling operation in the complex batch normalization (Trabelsi et al. 2017), but we decouple it from the batch normalization which is moved after the $ReLU$ operation. Note that the additional computation of $\mathbf{TReLU}$ is negligible compared to the convolutional layers.

Complex Masking. The separator outputs a complex mask range from 0 to 1 that is multiplied with the encoder output to feed into the decoder. While the mask cannot go beyond 1, the trainable encoder and decoder could mitigate this limitation. We apply a $tanh$ operation to the amplitude of the complex tensor while preserving the angle component:

$$\mathbf{Ctanh}(\mathbf{x}) = tanh(||\mathbf{x}||) * \frac{\mathbf{x}}{||\mathbf{x}||}$$

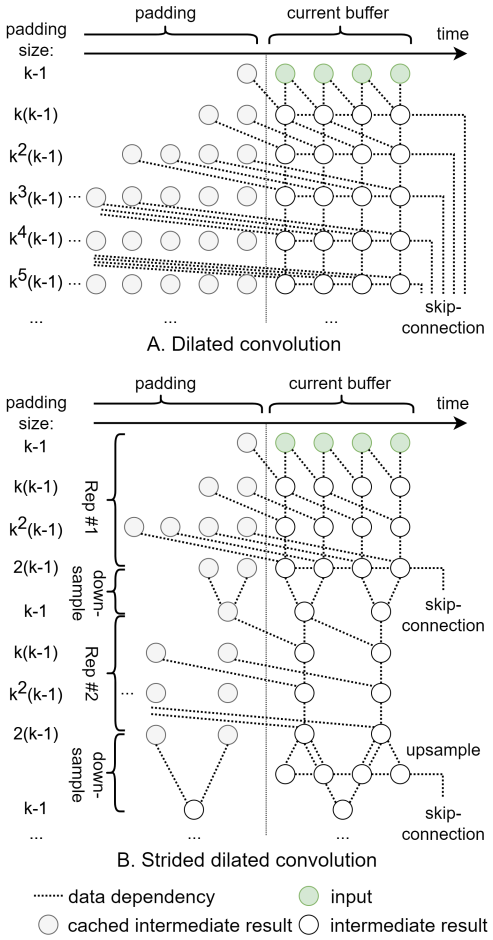


Figure 3: The strided dilated convolution structure when $M = 3$ and $k = 2$. The total padding size is much reduced because of the downsampling layer, and skip-connections get upsampled accordingly before summed up.

Strided Dilated Convolution. Temporal convolutional networks (TCN) utilize causal dilated convolution to efficiently enlarge the audio receptive field and achieve good separation performance (Luo and Mesgarani 2019). For real-time applications, while intermediate convolution results from the past can be cached for fast computation, the memory copy overhead is significant when our latency requirement is in the order of milliseconds. For example, Fig. 3A shows how the Conv-TasNet (Luo and Mesgarani 2019) architecture processes a stream of audio buffers. The input padding for each convolution layer contains temporal information that is computed while processing the previous buffers, and the input is shifted left and set as the new padding for the new buffer input. The shifted padding increases exponentially for latter layers due to a large receptive field. The shift operation is usually implemented using array copy and the state-of-the-art Conv-TasNet model requires 25 MB memory copy per input, that takes approximately 10 ms on a Raspberry Pi.

To reduce the memory copy overhead, we design a combination of dilated and strided convolution stacks. As shown

Configurations	Type	Hyperparameters	recep. field	# params	# MAC/s	lookahead
HybridBeam	hybrid	k=4, N=3, M=3, H=64, C=64, D=256	0.22s	0.72M	2.1G	1.5ms
HybridBeam+	hybrid	k=3, N=4, M=4, H=96, C=64, D=256	0.61s	1.1M	2.8G	1.5ms
TSNF	DNN	N=512, L=16, H=512, Sc=128, X=8, R=3	0.77s	5.2M	10.4G	390ms
TAC-F	DNN	H=32, L=64, W=64,K=64	∞	2.8M	14.5G	∞
Online MVDR	traditional	-	-	-	-	0
Mod. TSNF	DNN	N=512, L=32, H=512, Sc=128, X=8, R=3	0.77s	5.2M	10.4G	1.5ms
Mod. TAC-F	DNN	H=32, L=64, W=64,K=64	∞	2.3M	11.6G	4ms

Table 1: The specification of our model and baselines

in Fig. 3B, the network consists of a stack of N TCNs. Between each TCN which contains M dilated convolution layers, we add a 2x1 convolution layer with $stride = 2$ to downsample the signal and effectively reduce the size of the padding for the following layers. The skip-connections get upsampled using the nearest neighborhood method accordingly to the original sampling rate before summed up. Similar concept has been explored in (Tzinis, Wang, and Smaragdis 2020), and we extend it to the TCN structure. Compared to the original TCN stacks in (Luo and Mesgarani 2019), our strided dilated convolution technique requires the same $O(k^M N)$ padding to achieve a much larger $O(k^M 2^N)$ receptive field instead of $O(k^M N)$. For the specific parameters we implement, this reduces memory copy by >90%.

Simplified Temporal Convolutional Network. We further simplify the original TCN design. First, we use conventional convolution instead of depthwise separable convolution (D-conv) which has lower MACs for the same channel capacity but is usually memory-bounded and 4-8x less efficient than normal convolution operations on mobile CPUs (Zhang, Lo, and Lu 2020). In our experiments with a few state-of-the-art mobile DNN inference engines, normal convolution was much faster than D-conv. Second, we apply skip-connections on only the last convolution layer for each TCN stack, as shown in Fig. 2B. We found that this reduction of skip-connections reduces computation by approximately 20%. Third, we relax the dilation growth factor k to more than 2. Since our receptive field is $O(k^M 2^N)$, by increasing k, we would need less M and N, and so lower number of layers, to achieve the same receptive field.

Evaluation

We prototype and train the neural network in PyTorch (Paszke et al. 2019), and rewrite the model in TensorFlow (Abadi et al. 2016) as TensorFlow supports NHWC tensor layout which is faster on mobile CPUs. The model get converted to the input formats of two DNN inference engines, the MNN from Alibaba and Arm NN. The latter supports NEON and 16 bit float (FP16) primitives for ARMv8.2 CPUs. We use PulseAudio to access the microphones in real time and use a sampling rate of 16 kHz and 16 bit bitwidth.

Simulated Dataset

To gather a large amount of training data, we use software to simulate random reverberate noisy rooms using the image source model (Scheibler, Bezzam, and Dokmanić 2018). The rooms are simulated using absorption rates of real materials and a maximum RT60 of 500 ms. By default, we use a virtual 6-mic circular array with a radius of 5 cm. The distance between the virtual speakers and the microphone array is at least 0.8 m, and the direction of arrival differences of the speakers are at least 10°. The input direction is modeled as the groundtruth plus a random error less than 5° simulating the gaze tracking measurement error (Sipatchin, Wahl, and Rifai 2020). We place virtual speakers at random locations within the room playing random speech utterances from the VCTK dataset (Veaux et al. 2017), meanwhile simulating diffused noise from the MS-SNSD dataset (Reddy et al. 2019) and WHAM! dataset (Wichern et al. 2019). The combined speech power to noise ratio is randomized between $[5, 25]$ dB. 10%, 40%, 40%, 10% of the generated clips consists of 1-4 speakers, respectively, and we apply random gain within $[-5, 0]$ dB to each speaker. We guarantee that there exists speech utterance overlap for 2-4 speaker scenarios. We render the synthetic audio and generate 4s clips. We generate a total of 8000 clips as training set, 400 clips as validation set, and 200 clips as test set. No speech clips or noise has appeared in more than one of these three sets. To evaluate the performance on different microphone number and array layouts on various wearable form factors, we additionally generate datasets using three custom microphone array layouts on a virtual reality (VR) headset as shown in Fig. 4.

Configuration	SI-SDRi	SDRi	PESQ
HybridBeam	11.3 dB	10.1 dB	1.61
HybridBeam+	**13.3 dB**	11.4 dB	1.92
HybridBeam+ w/ FP16	13.1 dB	11.2 dB	1.92
HybridBeam+ w/o BF	10.9 dB	9.7 dB	1.70
Online MVDR	5.2 dB	5.9 dB	1.52
Mod. TSNF	13.1 dB	**11.6 dB**	**1.94**
Mod. TAC-F	12.1 dB	11.1 dB	1.77

Table 2: The quantitative performances of our method compared with baselines using a circular 6-mic array

#mic	# sources				
	1	2	3	4	Overall
4	11.9 dB	10.8 dB	10.5 dB	8.4 dB	10.6 dB
5	11.8 dB	12.0 dB	11.9 dB	9.1 dB	11.7 dB
6	11.6 dB	13.4 dB	13.5 dB	10.2 dB	12.9 dB

Table 3: SI-SDRi performance using three custom microphone array layout under different number of sound sources

Model Specification

We use two specifications for our model. The encoder and decoder both have a kernel size of 32 and a stride of 8. The rest of the hyperparameters are listed in Table 1. The lookahead comes from the transposed convolution in the decoder.

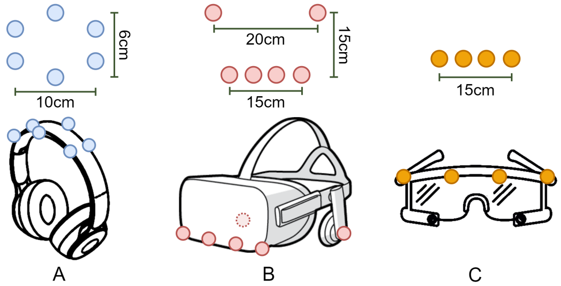


Figure 4: Potential mic-array layouts. (A): six-mic hexagon array on top of a headphone; (B): five-mic sub-array of a six-mic array (the microphone on the left/right ear is disabled when the input direction is on the opposite side) on an AR headset; (C) four-mic linear array on a pair of smart glasses.

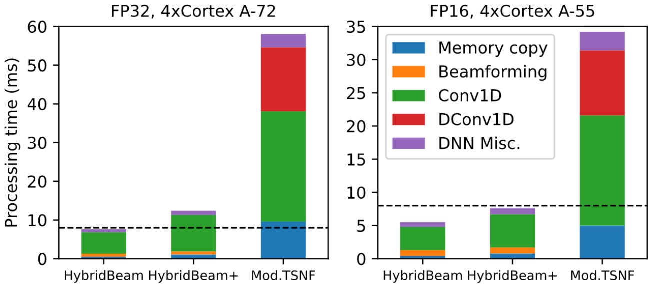


Figure 5: The processing time composition. The dashed line is maximum processing time to achieve real-time operation.

We use three baselines for reference: 1) traditional, online MVDR beamformer; 2) modified Temperal Spatial Neural Filter (TSNF) (Gu and Zou 2020), where we replace the Tas-Net structure with a causal Conv-TasNet structure and use the identical encoder as ours to achieve the same lookahead duration; and 3) modified TAC-FasNet (TAC-F) (Luo et al. 2019), where we replace bidirectional RNN with directional RNN for causal construction, conduct the same alignment operation to the multi-channel input before feeding into the network, and output only one channel.

Training Procedure

When synthesizing each training audio clips, we additionally synthesize another version where only one of the sound source and the first microphone are present, and no reverberation is rendered. This version is used as the groundtruth when the direction input is the direction of the present source. Hence, our model is trained to *simultaneously* do de-reverberation, source separation and noise suppression. We use a 1:10 linear combination of scale-invariant signal-to-distortion ratio (SI-SDR) (Le Roux et al. 2019) and mean L1 loss as training subjective, where the former is used to measure the speech quality, and the latter regulates the output power to be similar to the groundtruth.

Network Evaluation on Synthetic Datasets

Table 2 shows the SI-SDR (SI-SDRi) and SDR improvements (SDRi). We find that all DNN-based approaches outperform traditional MVDR beamformer. Our HybridBeam+ model that uses a slightly larger model achieves comparable results with the causal and low lookahead version of (Gu and Zou 2020), but using significant less number of parameters and computation. We additionally evaluate two variants of our large model. First, we use 16 bit float format (FP16) instead of 32 bits and see only a 0.2 dB drop in both SI-SDRi and SDRi. Using FP16 drastically reduce the inference time on platforms that support native FP16 instructions. Second, we remove the three beamformers and retrain the network. The SI-SDRi drops by more than 2 dB, which shows the usefulness of prebeamforming. We used bootstrapping sampling techniques to evaluate our testset 100 times. The 25th, 50th and 75th percentile are 12.99 dB, 13.32 dB and 13.61 dB respectively on our HybridBeam+ model.

RT60(s)	< 0.2	$0.2 - 0.4$	$0.4 - 0.6$	> 0.6
SI-SDRi	13.5 dB	13.3 dB	12.8 dB	11.1 dB

Table 4: SI-SDRi under different reverberation time

Removed BF	None	SD	MVDR	Nonlinear
SI-SDRi	13.3 dB	13.1 dB	12.0 dB	12.4 dB

Table 5: SI-SDRi when we remove a beamformer.

Table 3 shows the SI-SDRi results on the custom microphone array layouts under 1-4 sources. Note that the training set contains all 1-4 source scenarios and each row of the table shows the performance on the testset under a same trained model. We see that adding microphones consistently improves the result of more than one source scenarios.

We also evaluate the performance with different reverberation time (RT60) in Table. 4. We see a performance degradation with a RT60 greater than 0.6 s, likely due to a limited receptive field. Table 5 shows an ablation study, where we remove one out of the three beamformers and retrain the network. We also evaluated HybridBeam+ with only one reference channel (the first microphone channel) without shifting along with the output of the beamformers as input. The resulting SI-SDRi is only 0.2 dB lower, which indicates the usefulness of pre-beamforming. Table 6 shows how our model performs with different directional differences between two sources. The separation performance increases as the angular difference between the sources increases. When there is no direction error in the input, the SI-SDRi improves for smaller angular differences.

We also compared our results with two real-valued networks with the same HybridBeam+ structure: (1) a real-valued version trained with dimensions adjusted to match the number of trainable parameters in our complex-valued network and 2) a real-valued network constructed with a same number of CNN channels (thus, twice the number of trainable parameters). The first network had a 0.5 dB SI-SDRi drop compared to our complex network. The second topline network achieved a 0.6 dB SI-SDRi gain. This shows that our complex-valued network has a good tradeoff.

On-device Latency Analysis

We deploy the models on two mobile development boards to measure the processing latency: a Raspberry Pi 4B with a four-core Cortex A-72 CPU and a four-core low-power Cortex A-55 developement board which support FP16 operations, both running at 2 GHz. The former is a popular $35

diff (°)	10-20	20-30	30-40	40-50	>50
w/ error (dB)	7.4	10.6	13.3	13.5	13.7
w/o error (dB)	7.9	11.7	13.5	13.8	13.9

Table 6: Different direction differences between 2 sources.

single-board computer, and the latter CPU is designed for low-power wearable devices and efficient cores on smartphones for lightweight tasks like checking emails.

We run the model in real-time and the buffer size is set to 128 samples (8 ms). Recall from Fig. 1 that the processing time should be less than 8 ms to guarantee real-time operation. Fig. 5 shows the processing time of of our models as well as the causal version of (Gu and Zou 2020). We find that with a comparable source separation performance, inference using our model takes much less time. Specifically, memory copy overhead is significantly reduced because of the strided dilated convolution, so does computation because of an overall smaller model with vanilla convolution. Finally, with a lookahead of 1.5 ms, our models can run on our two platforms in real-time with a 17.5 ms end-to-end latency.

Network Evaluation on Hardware Data

Hardware dataset. To evaluate model generalization, we implement a headset prototype and test with actual hardware. We modify a Seeed ReSpeaker 6-Mic Circular Array kit and place the microphones in the configuration in Fig. 4B around a HTC Vive Pro Eye VR headset in Fig. 6. The headset's gaze tracker provides the direction of arrival for our model. In addition to generating synthesized data using the above procedure but with the actual microphone layout, we also collect hardware data in two different rooms: one large, empty and reverberate conference room (approximately $5 \times 7m^2$), denoted as *Room La*, and one smaller, regular room with desks (approximately $3 \times 5m^2$), denoted as *Room Sm*. The playback speech is from the same VCTK dataset but is played from a portable Sony SBS-XB20 speaker. We place the speaker at 1 m and different angles within $-75°$ to $75°$. We calibrate the speaker-microphone delay and phase distortions in an anechoic chamber using a chirp signal and apply the calibration to the original signal. After data collection, we randomly add two recordings whose direction of arrival difference is more than $10°$ as the mixture signal. We pick the calibrated original speech and the direction of arrival of one of them as groundtruth and input direction to our model.

We use our model to test on hardware datasets collected in the conference rooms. We first see if training on only synthesized data can generalize to hardware data. We choose the best baseline on the synthetic datasets for comparison. Table 7 shows that the previous work generalizes poorly and does not work in real data. Manual inspection indicates that the the model sometimes predicts wrong sound sources. This is mostly because the features used by TSNF is highly affected by noise and interference and is not robust in real-world scenarios. In contrast, our model generalizes and outperforms MVDR baseline. Our hypothesis is that our model focuses on improving the already beamformed signals instead of deciding which source to separate, which is a harder problem. We also mix the 50% actual recordings with 50% synthesized data as the training set and test on the recordings in another room. The model performs better and achieves another 3 dB gain, regardless of the room acoustic properties.

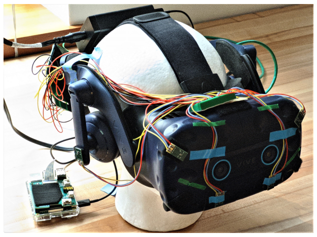


Figure 6: Gaze-controlled directional hearing AR prototype.

Model	Training	Test	SI-SDRi
MVDR	None	Room La	4.5 dB
		Room Sm	4.2 dB
TSNF	Synthetic	Room La	3.1 dB
Mod. TSNF			-1.6 dB
HybridBeam+	Synthetic	Room La	5.6 dB
	Synthetic+Room Sm	Room La	8.7 dB
	Synthetic+Room La	Room Sm	8.2 dB

Table 7: Performance using real world data.

Conclusion and Limitations

We believe that while this paper makes important contributions in demonstrating low-complexity neural networks for directional hearing, there is scope for improvements.

Larger real-world datasets. Our model can generalize to real recordings using synthetic training data, but is not as good as using synthetic test set, because of audio distortion like hardware nonlinearity and audio refraction. Generalization could be further improved with a larger scale real-world data collection under various rooms and devices.

Binaural settings. When we retrain the network with 2 microphones and up to 2 sources, we had an overall SI-SDRi performance of only 6.1 dB. This is likely because we depend on the beamformers to reduce the computational complexity of the neural networks. With 2 microphones, superdirective, MVDR and WebRTC beamformers only provided 1.8, 1.9, -0.6 dB respectively. Binaural beamformers (Liao et al. 2015; Srinivasan 2008; Hadad et al. 2017) can potentially be used to improve performance.

Specialized hardware. We use low-power CPU to deploy our model because of its extensive adoption in wearables and high flexibility. Recent low-power DNN accelerators can run DNN more efficiently (Wang, Wei, and Brooks 2019) and can be used to further improve our DNN efficiency, while running the beamformers on a CPU or microcontroller.

Lower latency. While our target end-to-end latency can be useful for wearables, close-canal hearing aids or hearing devices with active noise cancellation, an open-canal setting requires more stringent latency requirements (Stone et al. 2008). This may need hardware-software co-design.

Runtime optimizations. Techniques like network pruning and quantization may potentially help existing models run on-device. Those techniques however may also help our model run on even more resource-constraint hardware.

Acknowledgments

We thank Les Atlas, Steve Seitz, Laura Trutoiu and Ludwig Schmidt for their important feedback on this work.

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 265–283.

Bassey, J.; Qian, L.; and Li, X. 2021. A survey of complex-valued neural networks. *arXiv preprint arXiv:2101.12249*.

Brayda, L.; Traverso, F.; Giuliani, L.; Diotalevi, F.; Repetto, S.; Sansalone, S.; Trucco, A.; and Sandini, G. 2015. Spatially selective binaural hearing aids. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, 957–962.

Chen, Z.; Li, J.; Xiao, X.; Yoshioka, T.; Wang, H.; Wang, Z.; and Gong, Y. 2018a. Cracking the cocktail party problem by multi-beam deep attractor network. arXiv:1803.10924.

Chen, Z.; Xiao, X.; Yoshioka, T.; Erdogan, H.; Li, J.; and Gong, Y. 2018b. Multi-channel overlapped speech recognition with location guided speech extraction network. In *2018 IEEE SLT Workshop*, 558–565. IEEE.

Chhetri, A.; Hilmes, P.; Kristjansson, T.; Chu, W.; Mansour, M.; Li, X.; and Zhang, X. 2018. Multichannel audio front-end for far-field automatic speech recognition. In *2018 EUSIPCO*, 1527–1531. IEEE.

Comon, P.; and Jutten, C. 2010. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press.

Défossez, A.; Synnaeve, G.; and Adi, Y. 2020. Real Time Speech Enhancement in the Waveform Domain. *ArXiv*, abs/2006.12847.

Défossez, A.; Usunier, N.; Bottou, L.; and Bach, F. 2019. Music source separation in the waveform domain. *arXiv preprint arXiv:1911.13254*.

Doclo, S.; Gannot, S.; Moonen, M.; Spriet, A.; Haykin, S.; and Liu, K. R. 2010. Acoustic beamforming for hearing aid applications. *Handbook on array processing and sensor networks*, 269–302.

Fedorov, I.; Stamenovic, M.; Jensen, C.; Yang, L.-C.; Mandell, A.; Gan, Y.; Mattina, M.; and Whatmough, P. N. 2020. TinyLSTMs: Efficient neural speech enhancement for hearing aids. *arXiv preprint arXiv:2005.11138*.

Gu, R.; Chen, L.; Zhang, S.-X.; Zheng, J.; Xu, Y.; Yu, M.; Su, D.; Zou, Y.; and Yu, D. 2019a. Neural Spatial Filter: Target Speaker Speech Separation Assisted with Directional Information. In *Interspeech*, 4290–4294.

Gu, R.; Wu, J.; Zhang, S.-X.; Chen, L.; Xu, Y.; Yu, M.; Su, D.; Zou, Y.; and Yu, D. 2019b. End-to-end multi-channel speech separation. *arXiv preprint arXiv:1905.06286*.

Gu, R.; and Zou, Y. 2020. Temporal-spatial neural filter: Direction informed end-to-end multi-channel target speech separation. *arXiv preprint arXiv:2001.00391*.

Gupta, R.; Ranjan, R.; He, J.; Gan, W.-S.; and Peksi, S. 2020. Acoustic transparency in hearables for augmented reality audio: Hear-through techniques review and challenges. In *Audio Engineering Society Conference: 2020 AES International Conference on Audio for Virtual and Augmented Reality*. Audio Engineering Society.

Habets, E. A.; Benesty, J.; Gannot, S.; and Cohen, I. 2010. The MVDR beamformer for speech enhancement. In *Speech Processing in Modern Communication*, 225–254. Springer.

Hadad, E.; Marquardt, D.; Pu, W.; Gannot, S.; Doclo, S.; Luo, Z.-Q.; Merks, I.; and Zhang, T. 2017. Comparison of two binaural beamforming approaches for hearing aids. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 236–240. IEEE.

Han, C.; Luo, Y.; and Mesgarani, N. 2020. Real-time binaural speech separation with preserved spatial cues. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6404–6408. IEEE.

Haykin, S.; and Chen, Z. 2005. The cocktail party problem. *Neural computation*, 17(9): 1875–1902.

Higuchi, T.; Ito, N.; Araki, S.; Yoshioka, T.; Delcroix, M.; and Nakatani, T. 2017. Online MVDR beamformer based on complex Gaussian mixture model with spatial prior for noise robust ASR. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(4): 780–793.

Hu, Y.; Liu, Y.; Lv, S.; Xing, M.; Zhang, S.; Fu, Y.; Wu, J.; Zhang, B.; and Xie, L. 2020. DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement. *arXiv preprint arXiv:2008.00264*.

Jenrungrot, T.; Jayaram, V.; Seitz, S.; and Kemelmacher-Shlizerman, I. 2020. The cone of silence: speech separation by localization. *arXiv preprint arXiv:2010.06007*.

Kleijn, W. B. 2016. Methods and systems for robust beamforming. US Patent 9,502,021.

Krim, H.; and Viberg, M. 1996. Two decades of array signal processing research: the parametric approach. *IEEE signal processing magazine*, 13(4): 67–94.

Kumatani, K.; Arakawa, T.; Yamamoto, K.; McDonough, J.; Raj, B.; Singh, R.; and Tashev, I. 2012. Microphone array processing for distant speech recognition: Towards real-world deployment. In *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, 1–10. IEEE.

Le Roux, J.; Wisdom, S.; Erdogan, H.; and Hershey, J. R. 2019. SDR–half-baked or well done? In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 626–630. IEEE.

Liao, W.-C.; Luo, Z.-Q.; Merks, I.; and Zhang, T. 2015. An effective low complexity binaural beamforming algorithm for hearing aids. In *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 1–5. IEEE.

Luo, Y.; Chen, Z.; Mesgarani, N.; and Yoshioka, T. 2020. End-to-end microphone permutation and number invariant multi-channel speech separation. In *ICASSP*, 6394–6398. IEEE.

Luo, Y.; Han, C.; Mesgarani, N.; Ceolini, E.; and Liu, S.-C. 2019. Fasnet: Low-latency adaptive beamforming for multi-microphone audio processing. In *2019 IEEE ASRU Workshop*, 260–267. IEEE.

Luo, Y.; and Mesgarani, N. 2018. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *ICASSP*, 696–700. IEEE.

Luo, Y.; and Mesgarani, N. 2019. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8): 1256–1266.

Marseet, A.; and Sahin, F. 2017. Application of complex-valued convolutional neural network for next generation wireless networks. In *2017 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, 1–5. IEEE.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037.

Qian, K.; Zhang, Y.; Chang, S.; Yang, X.; Florencio, D.; and Hasegawa-Johnson, M. 2018. Deep learning based speech beamforming. In *2018 IEEE ICASSP*, 5389–5393. IEEE.

Reddy, C. K.; Beyrami, E.; Pool, J.; Cutler, R.; Srinivasan, S.; and Gehrke, J. 2019. A scalable noisy speech dataset and online subjective test framework. *arXiv preprint arXiv:1909.08050*.

Scheibler, R.; Bezzam, E.; and Dokmanić, I. 2018. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE ICASSP*, 351–355. IEEE.

Sipatchin, A.; Wahl, S.; and Rifai, K. 2020. Accuracy and precision of the HTC VIVE PRO eye tracking in head-restrained and head-free conditions. *Investigative Ophthalmology & Visual Science*, 61(7): 5071–5071.

Souden, M.; Benesty, J.; and Affes, S. 2010. A study of the LCMV and MVDR noise reduction filters. *IEEE Transactions on Signal Processing*, 58(9): 4925–4935.

Srinivasan, S. 2008. Low-bandwidth binaural beamforming. *Electronics Letters*, 44(22): 1292–1294.

Stone, M. A.; and Moore, B. C. 1999. Tolerable hearing aid delays. I. Estimation of limits imposed by the auditory path alone using simulated hearing losses. *Ear and Hearing*, 20(3): 182–192.

Stone, M. A.; Moore, B. C.; Meisenbacher, K.; and Derleth, R. P. 2008. Tolerable hearing aid delays. V. Estimation of limits for open canal fittings. *Ear and Hearing*, 29(4): 601–617.

Tammen, M.; Fischer, D.; and Doclo, S. 2019. DNN-based multi-frame MVDR filtering for single-microphone speech enhancement. *arXiv preprint arXiv:1905.08492*.

Trabelsi, C.; Bilaniuk, O.; Zhang, Y.; Serdyuk, D.; Subramanian, S.; Santos, J. F.; Mehri, S.; Rostamzadeh, N.; Bengio, Y.; and Pal, C. J. 2017. Deep complex networks. *arXiv preprint arXiv:1705.09792*.

Tzinis, E.; Wang, Z.; and Smaragdis, P. 2020. Sudo rm-rf: Efficient networks for universal audio source separation. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6. IEEE.

Veaux, C.; Yamagishi, J.; MacDonald, K.; et al. 2017. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*.

Wang, Y. E.; Wei, G.-Y.; and Brooks, D. 2019. Benchmarking tpu, gpu, and cpu platforms for deep learning. *arXiv preprint arXiv:1907.10701*.

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

Xiao, X.; Zhao, S.; Jones, D. L.; Chng, E. S.; and Li, H. 2017. On time-frequency mask estimation for MVDR beamforming with application in robust speech recognition. In *2017 IEEE ICASSP*, 3246–3250. IEEE.

Xu, Y.; Yu, M.; Zhang, S.-X.; Chen, L.; Weng, C.; Liu, J.; and Yu, D. 2020. Neural spatio-temporal beamformer for target speech separation. *arXiv preprint arXiv:2005.03889*.

Zhang, P.; Lo, E.; and Lu, B. 2020. High performance depthwise and pointwise convolutions on mobile devices. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 6795–6802.

Zhang, Z.; Xu, Y.; Yu, M.; Zhang, S.-X.; Chen, L.; and Yu, D. 2021. ADL-MVDR: All deep learning MVDR beamformer for target speech separation. In *ICASSP*, 6089–6093. IEEE.