SPEECH ENHANCEMENT USING DEEP DILATED CNN

BY

KAIZHI QIAN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Professor Mark Hasegawa-Johnson

ABSTRACT

In recent years, deep learning has achieved great success in speech enhancement. However, there are two major limitations regarding existing works. First, the Bayesian framework is not adopted in many such deep-learning-based algorithms. In particular, the prior distribution for speech in the Bayesian framework has been shown useful by regularizing the output to be in the speech space, and thus improving the performance. Second, the majority of the existing methods operate on the frequency domain of the noisy speech, such as spectrogram and its variations. We propose a Bayesian speech enhancement framework, called BaWN (Bayesian WaveNet), which directly operates on raw audio samples. It adopts the recently announced WaveNet, which is shown to be effective in modeling conditional distributions of speech samples while generating natural speech. Experiments show that BaWN is able to recover clean and natural speech.

Multi-channel speech enhancement with ad-hoc sensors has been a challenging task. Speech model guided beamforming algorithms are able to recover natural sounding speech, but the speech models tend to be oversimplified to prevent the inference from becoming too complicated. On the other hand, deep learning based enhancement approaches are able to learn complicated speech distributions and perform efficient inference, but they are unable to deal with variable number of input channels. Also, deep learning approaches introduce a lot of errors, particularly in the presence of unseen noise types and settings. We have therefore proposed an enhancement framework called DeepBeam, which combines the two complementary classes of algorithms. DeepBeam introduces a beamforming filter to produce natural sounding speech, but the filter coefficients are determined with the help of a monaural speech enhancement neural network. Experiments on synthetic and real-world data show that DeepBeam is able to produce clean, dry and natural sounding speech, and is robust against unseen noise.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I would like to acknowledge my graduate advisor, Professor Mark Hasegawa-Johnson, who has given me lots of research opportunities, guidance and insights. His broad knowledge and research attitude deeply cultivated me to become an independent, innovative and upright researcher.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Deep learning has been widely used in speech enhancement tasks because its strong representation power is capable of characterizing complex noise distributions. For example, some works directly predict output spectrum using deep neural networks (DNN) or denoising auto-encoders [1, 2, 3, 4]. A series of works [5, 6] applied different deep learning architectures to predict ideal ratio masks. In addition, several works performed speech separation using various deep learning architectures [7, 8].

However, these approaches have two major limitations. First, these deep learning algorithms rarely incorporate an explicit prior model for clean speech or a Bayesian framework, which has been shown effective for speech enhancement [9]. While the variability of noise is hardly tractable, the clean speech signal is highly structured, and thus a prior speech model can regularize enhanced speech to become speech-like. Without the speech model, many deep learning algorithms are not generalizable to noise without highly similar characteristics.

On the other hand, existing Bayesian speech enhancement algorithms mostly model speech using simple probability distribution in order to have closed-form solutions. For example, a large body of such works assume HMM-GMM models [10, 11, 12, 13] or Laplacian models [14, 15, 16, 17]. Others make looser assumptions on kurtosis or negentropy of speech distribution [18, 19]. For these algorithms, building a more accurate model for speech becomes a bottleneck, which can potentially be opened by deep learning.

The second limitation regarding the existing deep learning based approach is that most deep learning algorithms operate on amplitude spectrum, such as short-time Fourier transform or cochleargram. The noisy phase spectrum is directly applied to the enhanced speech without restoring the clean phase spectrum, which may suffer from phase distortion. Also, in some spectral restoration methods, the time domain signal is recovered by overlap-add, which is prone to artifacts and discontinuities. However, applying deep learning directly to speech waveform is difficult because the high sampling rate requires large temporal memory and receptive field size.

Fortunately, the recently announced WaveNet [20] has demonstrated a strong capability in modeling raw audio waveforms. Its receptive field size is significantly boosted by stacking dilated convolution layers with exponentially increasing dilation rates. Experiments have shown that it is able to generate random babbles with high naturalness. Moreover, WaveNet is probabilistic, which naturally fits into the Bayesian framework.

Motivated by these observations, we propose a Bayesian speech enhancement algorithm using deep learning structures inspired by WaveNet, called the Bayesian WaveNet (BaWN). BaWN directly predicts the clean speech audio samples by estimating the prior distribution and the likelihood function of clean speech using WaveNet-like architectures, which are the two major components of the Bayesian network. It promotes a happy marriage between the Bayesian framework and the deep learning techniques: the former broadens the generalizability for the latter, and the latter improves the model accuracy for the former.

Multi-channel speech enhancement with ad-hoc sensors has long been a challenging task [21]. As the traditional benchmark in multi-channel enhancement tasks, beamforming algorithms do not work well with with ad-hoc microphones. This is because most beamformers need to calibrate the speaker location as well as the interference characteristics, so that they can turn the beam toward the speaker, while suppressing the interference. However, neither parameter can be accurately measured, due to the missing sensor position information and microphone heterogeneity [22].

Another class of beamforming algorithms avoid measuring the speaker position and interference. Instead, they introduce prior knowledge on speech, and find the optimal beamformer by maximizing the “speechness” criteria, such as sample kurtosis [18], negentropy [19], speech prior distributions [16, 17], fitting glottal residual [23] etc. In particular, the GRAB algorithm [23] is able to outperform the closest microphone strategy even in very adverse real-world scenarios. Despite their success, these algorithms are limited by their oversimplified prior knowledge. For example, GRAB only models glottal energy, resulting in vocal tract ambiguity.

On the other hand, deep learning techniques are well known for their ability to capture complex probability dependencies and efficient inference, and thus have been widely used in single-channel speech enhancement tasks [6, 7, 8, 24, 25, 26]. Unfortunately, directly applying deep enhancement networks to multi-channel enhancement suffers from two difficulties. First, deep enhancement techniques often produce a lot of artifacts and nonlinear distortions [24, 25] which are perceptually undesirable. Second, neural networks often generalize poorly to unseen noise and configurations, whereas in speech enhancement with ad-hoc sensors, such variability is large.

As it turns out, these problems can in turn be resolved by traditional beamforming. Therefore, several algorithms [27, 28, 29, 30, 31] have been proposed that apply deep learning to predict time-frequency masks, and then beamforming to produce the enhanced speech. However, these methods are confined to frequency domain, which incurs two problems for our application. First, they to not work well for ad-hoc microphones because of the spatial correlation estimation errors. Second, our application is for human consumption, but the frequency-domain methods suffer from phase distortions and discontinuities, which impede perceptual quality.

Motivated by this observation, we have proposed an enhancement framework for ad-hoc microphones called DeepBeam, which combines deep learning and beamforming, and which directly works on waveform. DeepBeam introduces a time-domain beamforming filter to produce natural sounding speech, but the filter coefficients are iteratively determined with the help of WaveNet [20]. It can be shown that despite the error-prone enhancement network, DeepBeam is able to converge approximately to the optimal beamformer under some assumptions. Experiments on both the simulated and real-world data show that DeepBeam is able to produce clean, dry and natural sounding speech, and generalize well to various settings.

CHAPTER 2

ALGORITHM

2.1 Bayesian WaveNet

The problem is formulated within the Bayesian framework. Denote $X_{0:T-1}$ as the random process of the clean speech, which is quantized into Q levels, $q_{0:Q-1}$, via the μ-law encoding [32], so each X_t is a discrete variable. The subscript $0 : T-1$ denotes a set with subscripts running from 0 through $T-1$. Denote $Y_{0:T-1}$ as the random process of the observed noisy signal. In this thesis, only additive noise is considered, but the framework is generalizable to other types of interferences. Our task is to infer the clean speech $\hat{x}_t$ given a set of noisy observations $Y_{0:T} = y_{0:T}$. For notational ease, probability mass functions will be abbreviated, e.g. $p(X_t = x_t|Y_t = y_t)$ as $p(x_t|y_t)$.

We apply a sub-optimal greedy inference scheme for $X_{0:T-1}$. Given inferred values of the past samples $\hat{x}_{0:t-1}$, the inferred value of the current sample, $\hat{x}_t$, is defined as the posterior expectation

$$\hat{x}_t \triangleq \mathbb{E}\left[X_t|X_{t-\tau_1:t-1} = \hat{x}_{t-\tau_1:t-1}, Y_{t-\tau_2:t+\tau_2} = y_{t-\tau_2:t+\tau_2}\right] \tag{2.1}$$

Here we have made a Markov assumption that the probabilistic dependence of X_t upon variables in the distant past and far future is negligible, when the closer ones, $X_{t-\tau_1:t-1}$ and $Y_{t-\tau_2:t+\tau_2}$, are given. The terms τ_1 and τ_2 denote the range of dependence on $X_{0:T-1}$ and $Y_{0:T-1}$, respectively. Therefore, the following posterior distribution should be evaluated:

$$\begin{aligned} &p(X_t = x_t|X_{t-\tau_1:t-1} = \hat{x}_{t-\tau_1:t-1}, Y_{t-\tau_2:t+\tau_2} = y_{t-\tau_2:t+\tau_2}) \\ \triangleq &p(x_t|\hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \\ \propto &p(x_t|\hat{x}_{t-\tau_1:t-1}) \cdot p(y_{t-\tau_2:t+\tau_2}|\hat{x}_{t-\tau_1:t-1}, x_t) \end{aligned} \tag{2.2}$$

where the $\triangleq$ sign denotes the abbreviation.

Define the likelihood function as

$$L(x_t; \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \triangleq p(y_{t-\tau_2:t+\tau_2}|\hat{x}_{t-\tau_1:t-1}, x_t) \tag{2.3}$$

Then Eq. (2.2) can be rewritten into

$$\begin{aligned} &p(x_t|\hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \\ =&\underbrace{p(x_t|\hat{x}_{t-\tau_1:t-1})}_{\text{prior model}} \cdot \underbrace{L(x_t; \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2})}_{\text{likelihood model}} \end{aligned} \tag{2.4}$$

The BaWN architecture is based on Eq. (2.4). As shown in Figure 2.1(a), it consists of two models. The first model is called the prior model, or the speech model, modeling the prior distribution of clean speech signals. For each time t, it takes $\hat{x}_{t-\tau_1:t-1}$ as input, and outputs a Q-dimensional vector of the log estimated pmf $\log \hat{p}(x_t|\hat{x}_{t-\tau_1:t-1})$ up to an unknown constant.

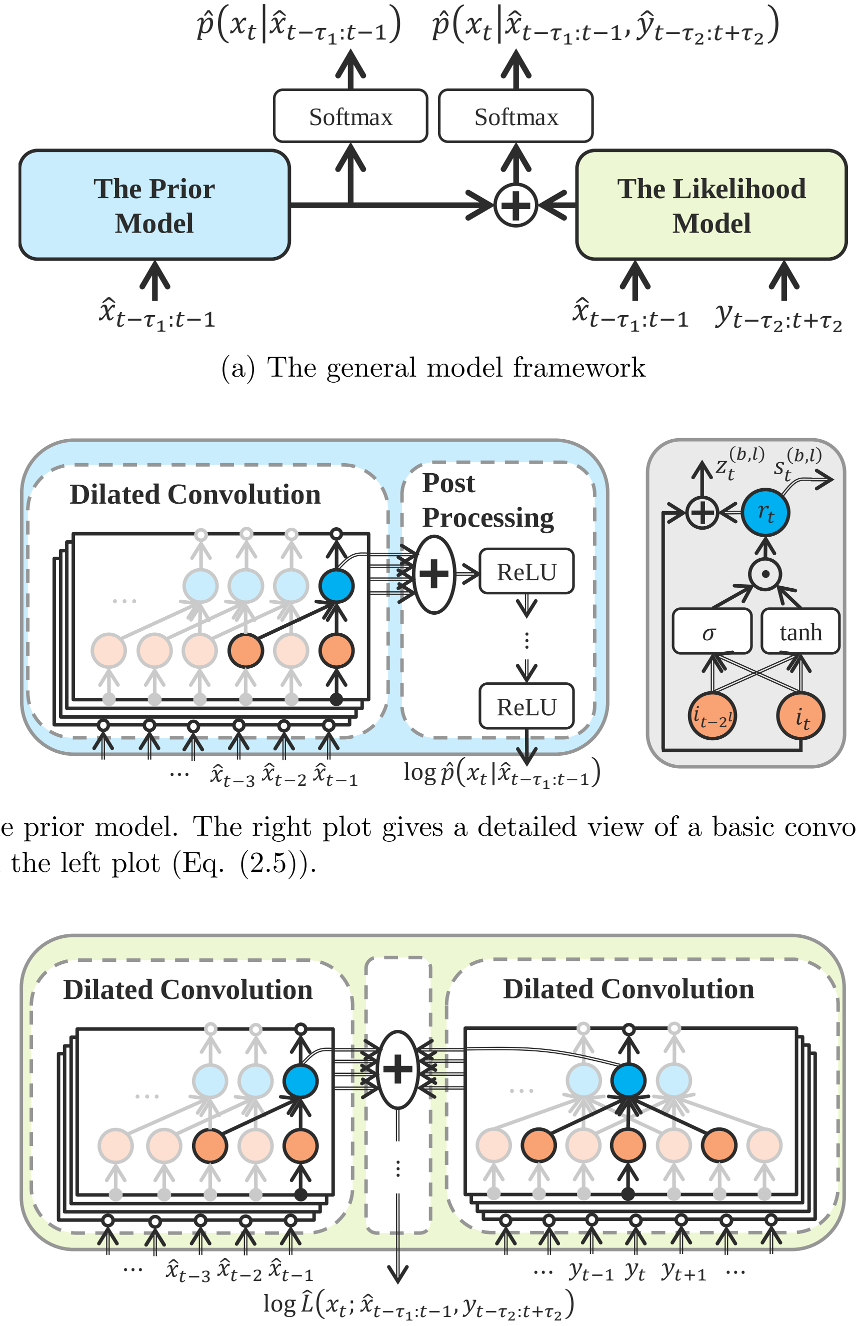
The second model is called the likelihood model, or the noise model, modeling the likelihood function. It takes as inputs $\hat{x}_{t-\tau_1:t-1}$ and $y_{t-\tau_2:t+\tau_2}$, and outputs a Q-dimensional vector of the estimated log likelihood function $\log \hat{L}(x_t; \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2})$ up to an unknown constant.

The two outputs are added and then passed through a softmax nonlinearity. Notice that the exponential function in softmax turns addition into multiplication; the normalization step in softmax removes any unknown constant. Therefore it can be easily shown, from Eq. (2.4), that the output of the softmax nonlinearity is the $p(x_t|\hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2})$ of interest. Also, the output of the prior model, passing through a softmax nonlinearity alone, becomes the prior distribution $p(x_t|\hat{x}_{t-\tau_1:t-1})$.

The following two subsections introduce the two models respectively.

2.1.1 The Prior Model

The prior model replicates the architecture of WaveNet because it performs a similar task. As shown in Figure 2.1(b), the prior model consists of two modules. The first is the dilated convolution module, which contains a stack of B_1 blocks with L_1 layers for each. The l-th layer in the b-th block is a 1D causal convolution layer through time, with kernel size 2 and dilation rate 2^l. For each time t, it produces two vector outputs—a hidden output $z_t^{(b,l)}$,


$\hat{p}(x_t|\hat{x}_{t-\tau_1:t-1})$
$\hat{p}(x_t|\hat{x}_{t-\tau_1:t-1},\hat{y}_{t-\tau_2:t+\tau_2})$
Softmax
Softmax
The Prior Model
The Likelihood Model
$\hat{x}_{t-\tau_1:t-1}$
$\hat{x}_{t-\tau_1:t-1}$
$y_{t-\tau_2:t+\tau_2}$
Dilated Convolution
Post Processing
ReLU
ReLU
$\cdots$ $\hat{x}_{t-3}$ $\hat{x}_{t-2}$ $\hat{x}_{t-1}$
$\log \hat{p}(x_t|\hat{x}_{t-\tau_1:t-1})$
$z_t^{(b,l)}$
$s_t^{(b,l)}$
r_t
σ
tanh
i_{t-2^l}
i_t
Dilated Convolution
Dilated Convolution
$\cdots$ $\hat{x}_{t-3}$ $\hat{x}_{t-2}$ $\hat{x}_{t-1}$
$\cdots$ y_{t-1} y_t y_{t+1} $\cdots$
$\log \hat{L}(x_t;\hat{x}_{t-\tau_1:t-1},y_{t-\tau_2:t+\tau_2})$

(a) The general model framework

(b) The prior model. The right plot gives a detailed view of a basic convolution unit in the left plot (Eq. (2.5)).

(c) The likelihood model. The middle module is the post processing module, whose structure is similar to that in (b).

Figure 2.1: The model architecture. Compound arrows denote that the node is multiplied by a weight matrix before sent to the next unit. Circled add and circled dot denote element-wise addition and multiplication respectively. The data path that generates the current output at time t is highlighted.

which is fed into the convolution layer above, and a skip output $s_t^{(b,l)}$, which is directly fed into the second module. The nonlinearity applied is a gated activation unit [33] with residual structure [34]. Formally,

$$f_t^{(b,l)} = \tanh\left(W_{f0}^{(b,l)} i_t^{(b,l)} + W_{f1}^{(b,l)} i_{t-2^l}^{(b,l)} + d_f^{(b,l)}\right) \tag{2.5a}$$

$$g_t^{(b,l)} = \sigma\left(W_{g0}^{(b,l)} i_t^{(b,l)} + W_{g1}^{(b,l)} i_{t-2^l}^{(b,l)} + d_g^{(b,l)}\right) \tag{2.5b}$$

$$r_t^{(b,l)} = f_t^{(b,l)} \odot g_t^{(b,l)} \tag{2.5c}$$

$$z_t^{(b,l)} = i_t^{(b,l)} + W_z^{(b,l)} r_t^{(b,l)} + d_z^{(b,l)} \tag{2.5d}$$

$$s_t^{(b,l)} = i_t^{(b,l)} + W_s^{(b,l)} r_t^{(b,l)} + d_s^{(b,l)} \tag{2.5e}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\odot$ denotes element-wise multiplication, and $i_t^{(b,l)}$ denotes the input to this layer,

$$i_t^{(b,l)} = \begin{cases} z_t^{(b,l-1)} & \text{if } l > 0 \\ z_t^{(b-1,L_1-1)} & \text{if } l = 0, b > 0 \\ W_i \hat{x}_t & \text{otherwise} \end{cases} \tag{2.6}$$

The second module is the post-processing module, which sums all the skip outputs of time t, $s_t^{(0:B_1-1,0:L_1-1)}$, and passes it to a stack of 1×1 convolution (fully connected within time t) layers with ReLU activation. The receptive field size is shown as

$$\tau_1 = B_1\left(2^{L_1} - 1\right)$$

2.1.2 The Likelihood Model

The likelihood model is more complex than the prior model. This is because 1) in addition to $\hat{x}_{t-\tau_1:t}$, which is the input to both models, the likelihood model also takes $y_{t-\tau_2:t+\tau_2}$ as input; 2) the prior model is causal, but the likelihood model is non-causal.

To address these complexities, we adapt the original WaveNet structure to that shown in Figure 2.1(c). The likelihood model also has a dilation convolution module and a post-processing module, but the dilation module now contains two parts. The first part deals with the input $\hat{x}_{t-\tau_1:t}$, and has the same structure as in Eqs. (2.5) and (2.6). The second part deals with the input $y_{t-\tau_2:t+\tau_2}$, and has almost the same structure, except for two differences. First, the number of blocks and layers within each block is changed to B_2 and L_2 respectively, to accommodate τ_2, which can be different from τ_1.

Second, instead of a causal convolution with kernel size 2, this part imposes a non-causal convolution with kernel size 3 to account for future dependency. Formally, Eqs. (2.5a) and (2.5b) are adapted to

$$
\begin{aligned}
f_t^{(b,l)} &= \tanh\left(W_{f0}^{(b,l)} i_t^{(b,l)} + W_{f1}^{(b,l)} i_{t-2^l}^{(b,l)} + W_{f-1}^{(b,l)} i_{t+2^l}^{(b,l)} + d_f^{(b,k)}\right) && (2.7a) \\
g_t^{(b,l)} &= \sigma\left(W_{g0}^{(b,l)} i_t^{(b,l)} + W_{g1}^{(b,l)} i_{t-2^l}^{(b,l)} + W_{g-1}^{(b,l)} i_{t+2^l}^{(b,l)} + d_g^{(b,l)}\right) && (2.7b)
\end{aligned}
$$

The post-processing module in the likelihood model is the same as that in the prior model, except that it sums all the skip outputs from both parts of the dilated convolution module.

2.2 Deep Beamformer

To formally define the problem, denote $s[t]$ as the clean speech signal. Suppose there are K channels of observed signals, $y_k[t], k = 1, \cdots, K$, which are represented as

$$
y_k[t] = s[t] * i_k[t] + n[t] * j_k[t] \tag{2.8}
$$

where $*$ denotes discrete convolution, $n(t)$ denotes additive noise, and $i_k[t]$ and $j_k[t]$ are the impulse responses of the signal reverberation and noise reverberation in the k-th channel, respectively. Our goal is to design a τ-tap beamformer $h_k[t], k = 1, \cdots, K$, whose output is defined as

$$
x[t] = \sum_{k=1}^{K} y_k[t] * h_k[t] \tag{2.9}
$$

For notational brevity, define

$$
\begin{aligned}
&\boldsymbol{s} = [s[1], \cdots, s[T]]^T \quad \boldsymbol{x} = [x[1], \cdots, x[T]]^T \\
&\boldsymbol{y}_k = [y_k[1], \cdots, y_k[T]]^T \quad \boldsymbol{y} = [\boldsymbol{y}_1^T, \cdots, \boldsymbol{y}_K^T]^T \\
&\boldsymbol{h} = [h_1[1], \cdots, h_1[\tau], h_2[1], \cdots, h_K[\tau]]^T
\end{aligned} \tag{2.10}
$$

which are all random vectors. Also define convolutional matrices

$$\boldsymbol{Y}_k = \begin{bmatrix} y_k[1] & & & \\ y_k[2] & y_k[1] & & \\ \vdots & \vdots & \ddots & \\ y_k[\tau] & y_k[\tau-1] & \cdots & y_k[1] \\ \vdots & \vdots & & \vdots \\ y_k[T] & y_k[T-1] & \cdots & y_k[T-\tau+1] \end{bmatrix} \tag{2.11}$$

and

$$\boldsymbol{Y} = [\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_K] \tag{2.12}$$

With these notations, Eq. (2.9) can be simplified as

$$\boldsymbol{x} = \boldsymbol{Y}\boldsymbol{h} \tag{2.13}$$

The target of designing the beamformer is to minimize the weighted mean squared error (MSE):

$$\min_{\boldsymbol{x}=\boldsymbol{Y}\boldsymbol{h}} \mathbb{E}\left[\|\boldsymbol{x}-\boldsymbol{s}\|_{\boldsymbol{W}}^2 | \boldsymbol{y}\right] \tag{2.14}$$

where $\|\boldsymbol{x}\|_{\boldsymbol{W}}^2 = \boldsymbol{x}^T\boldsymbol{W}\boldsymbol{x}$; $\boldsymbol{W}$ is a positive definite weight matrix, which, in our case, is a diagonal matrix of $\mathrm{Var}^{-1}(s[t]|\boldsymbol{y})$.

Equation (2.14) is a Wiener filtering problem [35], whose solution is

$$\boldsymbol{x}^* = \boldsymbol{P}\mathbb{E}[\boldsymbol{s}|\boldsymbol{y}] \tag{2.15}$$

where

$$\boldsymbol{P} = \boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{W}\boldsymbol{Y})^{-1}\boldsymbol{Y}^T\boldsymbol{W} \tag{2.16}$$

is in fact the *projection matrix* onto the beamforming output space. So by Eq. (2.15), $\boldsymbol{x}^*$ is essentially projecting $\mathbb{E}[\boldsymbol{s}|\boldsymbol{y}]$ onto the space that is representable by the beamforming filter.

As shown by Eq. (2.15), solving the Wiener filtering problem requires computing $\mathbb{E}[\boldsymbol{s}|\boldsymbol{y}]$, which, due to the complex probabilistic dependencies, we would like to introduce a deep neural network to learn. However, as discussed, training a neural network to directly predict $\mathbb{E}[\boldsymbol{s}|\boldsymbol{y}]$ from the multi-channel input $\boldsymbol{y}$ suffers from inflexible input dimensions, artifacts and poor generalization. DeepBeam tries to resolve these problems and find an approximate solution.

2.2.1 The Algorithm Overview

As mentioned, DeepBeam introduces a deep enhancement network to learn the posterior expectation, while addressing its limitations. First, DeepBeam is regularized by the beamformer to generalize well to unseen noise and microphone configurations. Second, it tolerates the distortions and artifacts generated by the neural network. Formally, the neural network outputs an inaccurate prediction of the posterior expectation $\mathbb{E}[\boldsymbol{s}|\boldsymbol{\xi}]$,

$$f(\boldsymbol{\xi}) = \mathbb{E}[\boldsymbol{s}|\boldsymbol{\xi}] + \boldsymbol{\varepsilon}(\boldsymbol{\xi}) \tag{2.17}$$

where $\boldsymbol{\xi}$ is a *single-channel* noisy observation, and $\boldsymbol{\varepsilon}(\boldsymbol{\xi})$ is the prediction error. The goal of DeepBeam is to approximate the optimal beamformer given the inaccurate enhancement network. Algorithm 1 shows the description of the DeepBeam algorithm. A graph of the DeepBeam framework is shown in Figure 2.2.

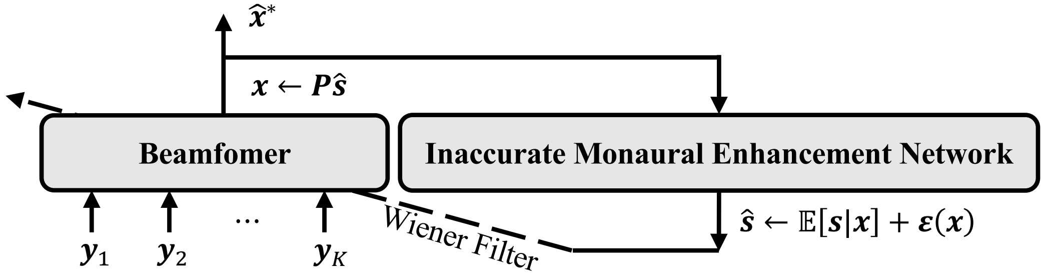

$\hat{\boldsymbol{x}}^*$
$\boldsymbol{x} \leftarrow \boldsymbol{P}\hat{\boldsymbol{s}}$
Beamfomer
Inaccurate Monaural Enhancement Network
$\hat{\boldsymbol{s}} \leftarrow \mathbb{E}[\boldsymbol{s}|\boldsymbol{x}] + \boldsymbol{\varepsilon}(\boldsymbol{x})$
Wiener Filter
$\boldsymbol{y}_1$
$\boldsymbol{y}_2$
$\cdots$
$\boldsymbol{y}_K$

Figure 2.2: DeepBeam framework.

Algorithm 1 essentially alternates between the posterior expectation and projection iteratively. It will be shown in section 2.2.3 that as long as the error term $\boldsymbol{\varepsilon}$ is not too large, this iteration will approximately converge to the optimal beamformer output.

One elegance of DeepBeam is that $\boldsymbol{x}^{(n)}$ can be regarded as a noisy observation, and shares some statistical structures with the true noisy observations, $\boldsymbol{y}_k$. To see this, notice that by Eq. (2.19), $\boldsymbol{x}^{(n)}$ is the output of a beamformer on $\boldsymbol{y}$. Therefore, it can be shown that $\boldsymbol{x}^{(n)}$ also takes the form of Eq. (2.8), with the same speech and noise source, but with a different im-

Algorithm 1 The DeepBeam algorithm.

```
Input: Multi-channel noisy speech observations $\boldsymbol{y}$;
    A neural network that predicts $f(\boldsymbol{\xi})$ (Eq. (2.17)) from any single-channel
    noisy observation $\boldsymbol{\xi}$.
Output: Beamformer output $\hat{\boldsymbol{x}}^*$.

Initialization:
Find the 'cleanest' channel $k^*$ by finding the channel that has the smallest
    0.4 quantile of its squared sample points.
Set $\boldsymbol{x}^{(0)} = \boldsymbol{y}_{k^*}$.
Iteration:
for $n = 1$ to maximum number of iterations do
    Feed $\boldsymbol{x}^{(n-1)}$ to the monaural enhancement network, and obtain its output
        $\hat{\boldsymbol{s}}^{(n)} = f(\boldsymbol{x}^{(n-1)}) = \mathbb{E}[\boldsymbol{s}|\boldsymbol{x}^{(n-1)}] + \boldsymbol{\varepsilon}(\boldsymbol{x}^{(n-1)})$    (2.18)
    Update the beamformer coefficients and output
        $\boldsymbol{x}^{(n)} = \boldsymbol{P}\hat{\boldsymbol{s}}^{(n)}$    (2.19)
end for
return $\hat{\boldsymbol{x}}^* = \boldsymbol{x}^{(N)}$
```

pulse response. This justifies the use of one monaural enhancement network to take care of all the $\boldsymbol{x}^{(n)}$.

2.2.2 Enhancement Network Structure

DeepBeam is a general framework, in which the choice of the neural network structure is not fixed. The following network structure is just one of the structures that produce competitive results.

The enhancement network applied here is similar to [25], which is inspired by WaveNet [20]. Formally, denote the *quantized* speech samples as $\tilde{s}[t]$, and the samples of $\boldsymbol{x}^{(n)}$ as $x^{(n)}[t]$. Then the enhancement network predicts the posterior probability mass function (PMF) of $\tilde{s}[t]$:

$$p(\tilde{s}[t]|\boldsymbol{x}^{(n)}) \approx p(\tilde{s}[t]|x^{(n)}[t-\tau_r], \cdots, x^{(n)}[t+\tau_r]) \tag{2.20}$$

Here we have restricted the probabilistic dependency to span τ_r time steps. Cross-entropy is applied as the loss function.

Similar to WaveNet, the enhancement network consists of two modules. The first module, called the dilated convolution module, contains a stack of dilated convolutional layers with residual connections and skip outputs. The second module, called the post processing module, sums all the skip outputs and feeds them into a stack of fully connected layers before producing the final output.

There are two major differences from the standard WaveNet structure. First, the input to the enhancement network is the noisy observation waveform $\boldsymbol{x}^{(n)}$ instead of the clean speech. Second, to account for the future dependencies, the convolutional layers are noncausal 1×3 instead of the causal 1×2.

After the posterior distribution is predicted, the posterior moments, $\boldsymbol{E}[\boldsymbol{s}|\boldsymbol{x}^{(n)}]$ and $\mathrm{Var}[s[t]|\boldsymbol{y}]$ (for computing $\boldsymbol{W}$), are computed as the moments of the predicted PMF.

2.2.3 Convergence Analysis

In order to analyze the convergence property of DeepBeam, we assume the following bound on the error term:

$$\mathbb{E}[\|\boldsymbol{P}\boldsymbol{\varepsilon}(\boldsymbol{x}^{(n)})\|_{\boldsymbol{W}}^2|\boldsymbol{y}] \leq \rho\mathbb{E}[\|\boldsymbol{x}^{(n)} - \boldsymbol{s}\|_{\boldsymbol{W}}^2|\boldsymbol{y}] \tag{2.21}$$

where $\rho < 0.5$ is some constant. This assumption is actually not quite stringent, because it bounds not the weighted norm of $\boldsymbol{\varepsilon}(\boldsymbol{x}^{(n)})$ itself, but its projected value $\boldsymbol{P}\boldsymbol{\varepsilon}(\boldsymbol{x}^{(n)})$. In fact, the projection can drastically reduce the weighted norm of the error term. For example, most of the artifacts and nonlinear distortions that the enhancement network introduces cannot possibly be generated by beamforming on $\boldsymbol{y}$, and therefore will be removed by the projection. The only errors that are likely to remain are residual noise and reverberations. This is one advantage of combining beamforming filter and neural network. This assumption is also very intuitive. It means that the projected output error is always smaller than input error.

Then, we have the following theorem.

Theorem 1. *Suppose Eq.* (2.21) *holds. Then*

$$\limsup_{n\to\infty} \mathbb{E}[\|\boldsymbol{x}^{(n)} - \boldsymbol{x}^*\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] \leq u \tag{2.22}$$

where

$$\begin{aligned} u = & \frac{2\rho}{1-2\rho} \mathbb{E}[\|\boldsymbol{s} - \boldsymbol{x}^*\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] \\ & + \frac{2}{1-2\rho} \sup_n \mathbb{E}[\|\boldsymbol{P}\mathbb{E}[\boldsymbol{s}|\boldsymbol{x}^{(n)}] - \boldsymbol{x}^*\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] \end{aligned} \tag{2.23}$$

Proof. On one hand, from Eqs. (2.18) and (2.19)

$$\begin{aligned} & \mathbb{E}[\|\boldsymbol{P}\boldsymbol{\varepsilon}(\boldsymbol{x}^{(n)})\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] = \mathbb{E}[\|\boldsymbol{x}^{(n+1)} - \boldsymbol{P}\mathbb{E}[\boldsymbol{s}|\boldsymbol{x}^{(n)}]\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] \\ \geq & \frac{1}{2} \mathbb{E}[\|\boldsymbol{x}^{(n+1)} - \boldsymbol{x}^*\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] - \mathbb{E}[\|\boldsymbol{P}\mathbb{E}[\boldsymbol{s}|\boldsymbol{x}^{(n)}] - \boldsymbol{x}^*\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] \end{aligned} \tag{2.24}$$

On the other hand, by orthogonality principle

$$\mathbb{E}[\|\boldsymbol{x}^{(n)} - \boldsymbol{s}\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] = \mathbb{E}[\|\boldsymbol{x}^{(n)} - \boldsymbol{x}^*\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] + \mathbb{E}[\|\boldsymbol{s} - \boldsymbol{x}^*\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] \tag{2.25}$$

Combining Eqs. (2.21), (2.24) and (2.25), we have

$$\mathbb{E}[\|\boldsymbol{x}^{(n+1)} - \boldsymbol{x}^*\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] \leq 2\rho \mathbb{E}[\|\boldsymbol{x}^{(n)} - \boldsymbol{x}^*\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] + (1-2\rho)u \tag{2.26}$$

Create an auxiliary sequence

$$a^{(n)} = \mathbb{E}[\|\boldsymbol{x}^{(n)} - \boldsymbol{x}^*\|_{\boldsymbol{W}}^2 | \boldsymbol{y}] - u \tag{2.27}$$

Then by Eq. (2.26),

$$a^{(n+1)} \leq (2\rho)^n a^{(1)} \tag{2.28}$$

Taking $\limsup_{n\to\infty}$ on both sides of Eq. (2.28) concludes the proof. □

If $u = 0$, then Eq. (2.22) implies mean square convergence to the optimal beamformer output. In actuality, u is nonzero, but it tends to be very small. The first term of u measures the distance between the optimal beamformer output and the true speech. According to our empirical study, when the number of channels is sufficient, the optimal beamformer is able to recover the true speech very well, so the first term is small. The second term of u measures the distance between two posterior expectations $\boldsymbol{P}\mathbb{E}[\boldsymbol{s}|\boldsymbol{x}^{(n)}]$ and

$\boldsymbol{P}\mathbb{E}[\boldsymbol{s}|\boldsymbol{y}]$. The former is conditional on single-channel noisy speech, and the latter on multiple-channel noisy speech. Considering that the speech sample space is highly structured, and that the noisy speech $\boldsymbol{x}^{(n)}$ is relatively clean already, both posterior expectations should be close to the true speech, and thereby close to each other. In a nutshell, with a small u, the DeepBeam prediction is highly accurate. Section 5.2 will verify the convergence behavior of DeepBeam empirically.

CHAPTER 3

EXPERIMENTS

3.1 Bayesian WaveNet

3.1.1 Training the Prior Model

If we replace the input $\hat{x}_{t-\tau_1:t-1}$ with the true clean samples, denoted as $x^*_{t-\tau_1:t-1}$, then the prior model can be trained on clean speech, following a similar paradigm as in WaveNet. Specifically, for each t, given the previous true clean speech, $x^*_{t-\tau_1:t-1}$ as input, the training scheme minimizes the cross entropy between the estimated prior distribution and the empirical distribution. Formally, the training scheme solves the following optimization problem:

$$\max \sum_{t=0}^{T-1} \sum_{i=0}^{Q-1} \mathbb{1}\left\{x^*_t = q_i\right\} \log \hat{p}(X_t = q_i | x_{t-\tau_1:t-1}) \tag{3.1}$$

where $\mathbb{1}\{\cdot\}$ denotes the indicator function, which equals 1 if the statement in its argument is true and 0 otherwise.

So far, we have implemented only the speaker-dependent enhancement task. The generalization to speaker-independent models will be one of our future directions.

3.1.2 Training the Likelihood Model

Once the prior model is trained, the likelihood model can be trained by combining both models to estimate the posterior distribution, as indicated by Eq. (2.2). Ideally, we would like to solve

$$\max \sum_{t=0}^{T-1} \sum_{i=0}^{Q-1} \mathbb{1}\left\{x^*_t = q_i\right\} \log \hat{p}(X_t = q_i | \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \tag{3.2}$$

However, notice that the input of time t contains $\hat{x}_{t-\tau_1:t-1}$, which is a function of the previous time outputs, as shown in Eq. (2.1). Therefore, Eq. (3.2) introduces time recurrence, which causes gradient explosion in practice. An alternative is to replace $\hat{x}_{t-\tau_1:t-1}$ with the true value $x^*_{t-\tau_1:t-1}$ as in prior model training, but this approximation leads to insufficient training, because the model is given too much oracle information about the clean speech.

Our solution is to replace $\hat{x}_{t-\tau_1:t-1}$ with the inferred clean speech produced by the network trained in the *previous iteration*. Denote the previous inferred value as $\hat{x}^{(\text{old})}_{t-\tau_1:t-1}$; then the problem in Eq. (3.2) is reformulated as

$$\max \sum_{t=0}^{T-1} \sum_{i=0}^{Q-1} \mathbb{1}\left\{x^*_t = q_i\right\} \log \hat{p}(X_t = q_i | \hat{x}^{(\text{old})}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \tag{3.3}$$

The previous inferred value $\hat{x}^{(\text{old})}_{t-\tau_1:t-1}$ can be implemented efficiently using the method in [36].

It should be emphasized that while optimizing for Eq. (3.3), the weights of the prior model should be held fixed to prevent deviation from modeling the prior distribution.

3.1.3 Configurations

The three dilated convolutional networks of the WaveNet enhancement model all have 4 blocks of 10 layers, which makes a receptive field size of approximately two to three phones. For each layer, the hidden output has 32 channels and the skip output has 1024 channels. The post-processing modules in both the prior and the likelihood models contain two fully connected layers, each with 1024 hidden nodes. The clean speech is quantized into 256 levels, so the output dimension is 256.

The training dataset consists of a clean training set (for the prior model) and a noisy training set. The clean training set contains a total of 9700 utterances (19 hours) from audio books played by a female speaker [37]. The noisy training set was created by mixing the 9700 clean utterances randomly with 100 environment noises from [4, 38, 39], including train, airport, restaurant and ring tones. The SNR of the noisy training set is set to two levels: 0 dB and -5 dB.

There are two test sets, respectively containing 20 and 100 clean utterances of the same speaker randomly selected from another audio book. For the first test set, called the unseen noise test set, 100 noises were selected from a completely different noise dataset [40] in order to test the generalizablity of BaWN, where the types of noise and recording configurations completely differ from that of the training noise dataset. For investigation purpose, the second test set, called the seen noise test set, contains 20 noises drawn from the training noise dataset.

The input training utterances were first segmented into fixed-length tokens. Then, each clean token was quantized using 256-level μ-law companding and padded with 4092 historical samples based on the receptive field size of the our model. The noisy utterances were not quantized because the model does not make predictions of noisy speech. Each noisy token was padded not only with historical samples but also with the same number of future samples. The target output was a 256 dimensional one-hot vector indicating the quantization level of the desired output sample.

The prior model was trained on all 9700 (19 hours) clean utterances. Due to significantly increased model complexity and the EM-like training procedures, the likelihood model was trained on only 500 (1 hour) utterances from the noisy training set. Though the small amount of training data may lead to an insufficiently trained likelihood model, it actually provides a good opportunity to verify the power of the prior model and test the generalizablity of BaWN. For fair comparison, the DNN-IRM baseline was trained on the complete noisy training set. During testing, each predicted clean sample was fed back as the clean input sample to predict the next clean sample.

The DIRM baseline was constructed according to [6] and trained on the same 9700 noisy utterances. The 64-channel cochleargrams were extracted from the noisy utterances as the input features. The targets were the ideal-ratio-masks (IRMs) at the corresponding frame and channel. The IRM of the current frame is predicted using 23 neighboring frames centered at the current frame. During testing, the IRMs were predicted and applied to the corresponding noisy utterances to recover clean utterances.

3.2 Deep Beamformer

3.2.1 Enhancement Network Configurations

The enhancement network hyperparameter configurations follow [20]. The network has 4 blocks of 10 dilated convolution layers. There are two post-processing layers. The hidden node dimension is 32, and the skip node dimension is 256. The clean speech is quantized into 256 levels via μ-law companding, and thus the output dimension is 256. The activation function in the dilated convolutional layers is the gated activation unit; that in the post-processing layers is the ReLU function. The output activation is softmax.

The enhancement network is trained on simulated data *only*, which is generated in the same way as in [23]. The speech source, noise source and eight microphones are randomly placed into a randomly sized cubic room. The impulse response from each source to each microphone is generated using the image-source method [41, 42]. The noisy observations are generated according to Eq. (2.8). The reverberation time is uniformly randomly drawn from [100, 300] ms. The energy ratio between the speech source and noise source, E_r, is uniformly randomly drawn from $[-5, 20]$ dB. The speech content is drawn from VCTK [43], which contains 109 speakers. The noise content contains 90 minutes of audio drawn from [4, 38, 39]. The total duration of the training audio is 8 hours. The enhancement network is trained using ADAM optimizer for 400,000 iterations.

3.2.2 Simulated Data Evaluation

The simulated data for evaluation is generated the same way as the training data, except for two differences. First, the source energy ratio, E_r, is set to four levels, -10 dB, 0 dB, 10 dB, and 20 dB. Second, both the speaker and noise can be either seen or unseen in the training set, leading to four different scenarios to test generalizability. It is worth highlighting that the unseen speaker utterances and unseen noise are both drawn from different corpora from training, TIMIT [44] and FreeSFX [40] respectively. Each utterance is 3 seconds in length. The total length of the dataset is 12 minutes.

DeepBeam is compared with GRAB [23], MVDR[1] [45], IVA [16] and the closest channel (CLOSEST), in terms of two criteria:

- **Signal-to-Noise Ratio (SNR)**: The energy ratio of processed clean speech over processed noise in dB.
- **Direct-to-Reverberant Ratio (DRR)**: the ratio of the energy of direct path speech in the processed output over that of its reverberation in dB. Direct path and reverberation are defined as clean dry speech convolved with the peak portion and tail portion of processed room impulse response. The peak portion is defined as ±6 ms within the highest peak; the tail portion is defined as ±6 ms beyond.

3.2.3 Real-world Data Evaluation

DeepBeam and the baselines are also evaluated on the real-world dataset introduced in [23], which consists of two utterances by two speakers mixed with five types of noise, all recorded in a real conference room using eight randomly positioned microphones. The source energy ratio is set such that the SNR for the closest microphone is 10 dB. The utterance in each scenario is around 1 minute long, so the total length of the dataset is 10 minutes.

Besides SNR, a subjective test similar to [23] is performed on Amazon Mechanical Turk. Each utterance is broken into six sentences. In each test unit, called HIT, a subject is presented with one sentence processed by the five algorithms, and asked to assign an MOS [46] to each of them. Each HIT is assigned to 10 subjects.

[1] Clean speech is given for voice activity detection.

CHAPTER 4

RESULTS

4.1 Bayesian WaveNet

4.1.1 Objective Evaluation Results

The performance was measured by the average of SNR, signal-to-artifacts ratio (SAR), signal-to-distortion ratio (SDR), and short-time objective intelligibility (STOI) of the predicted clean utterances. The first three metrics were computed using the BSS-EVAL toolbox [47].

As seen in Table 4.1, the BaWN model outperforms the DNN-IRM model in terms of much higher SNRs. The performance advantage is more significant in the -5 dB case, where BaWN takes the lead in SAR and STOI as well. Also, our model generalizes better to the completely different unseen noise, as the performance drop is smaller. This is remarkable considering that the likelihood model was trained on only one hour of noisy speech and the parameters of the model were not tuned. The prior model has enough knowledge about the distribution of clean speech samples and tends to make non-speech distributions less likely under unseen noise and low SNR, which helps to make better predictions even if the likelihood model is weak. BaWN achieves slightly lower SDR and, in the 0dB case, SAR, because the sequential inference would occasionally generate impulse noise. Yet this does not weaken our argument for BaWN, considering the inherent negative correlation between the SNR and SAR/SDR, and the huge performance gain in SNR.

Table 4.1: Average SNR, SAR, SDR, STOI of the enhanced utterance using DNN-IRM and BaWN. The first three metrics are measured in decibels (dB), and the STOI is measured in percentage (%). Case indicates the input SNR of the training and testing dataset. Noise indicates whether the noise type is covered by the training set. BaWN stands for Bayesian WaveNet. DIRM stands for DNN-IRM.

Case	**Noise**	**Model**	**SNR**	**SAR**	**SDR**	**STOI**
0dB	seen	BaWN	22.2	8.53	8.83	85.7
		DIRM	15.6	10.3	12.3	86.4
	unseen	BaWN	22.1	8.37	8.75	84.3
		DIRM	11.9	8.58	12.7	84.8
-5dB	seen	BaWN	21.6	7.15	7.37	81.7
		DIRM	12.2	6.45	8.53	79.0
	unseen	BaWN	20.3	6.65	6.92	80.7
		DIRM	9.20	5.25	8.24	76.6

4.2 Deep Beamformer

4.2.1 Simulated Data Results

Table 4.2 shows the results. As expected, DeepBeam's performance drops from S1, where both noise and speaker are seen during training, to S4, where neither is seen. However, in terms of SNR, even DeepBeam S4 significantly outperforms MVDR, which is the benchmark in noise suppression. In terms of DRR, DeepBeam matches or surpasses CLOSEST except for -10 dB. GRAB performs worse than in [23], because each utterance is reduced from 10 seconds to 3 seconds, which is more realistic but challenging. In short, of “cleanness” and “dryness”, most algorithms can only achieve one, but DeepBeam can achieve *both* with superior performance.

4.2.2 Real-world Data Results

DeepBeam and the baselines are also evaluated on the real-world dataset introduced in [23], which consists of two utterances by two speakers mixed with five types of noise, all recorded in a real conference room using eight randomly positioned microphones. The source energy ratio is set such that the SNR for the closest microphone is 10 dB. The utterance in each scenario

Table 4.2: Simulated Data Evaluation Results.

	$E_r =$	-10	0	10	20
SNR (dB)	DeepBeam S1	18.5	22.0	26.5	28.4
	DeepBeam S2	17.1	20.3	25.9	27.4
	DeepBeam S3	15.3	19.5	24.1	27.6
	DeepBeam S4	14.1	19.0	23.1	28.5
	GRAB	2.48	12.5	21.6	25.4
	CLOSEST	-5.13	3.38	14.9	24.8
	MVDR	8.41	12.9	22.6	26.7
	IVA	10.3	13.3	16.8	19.2
DRR (dB)	DeepBeam S1	3.45	8.97	11.2	11.5
	DeepBeam S2	7.38	11.9	12.6	11.5
	DeepBeam S3	5.60	4.85	8.43	9.78
	DeepBeam S4	2.11	6.68	7.10	9.31
	GRAB	-0.83	1.70	3.63	3.68
	CLOSEST	8.56	7.32	7.67	8.44
	MVDR	-2.17	-3.47	-3.42	-4.13
	IVA	-8.92	-8.77	-8.81	-8.99

S1: seen speaker, seen noise S2: seen speaker, unseen noise

S3: unseen speaker, seen noise S4: unseen speaker, unseen noise

is around 1 minute long, so the total length of the dataset is 10 minutes.

Besides SNR, a subjective test similar to [23] is performed on Amazon Mechanical Turk. Each utterance is broken into six sentences. In each test unit, called HIT, a subject is presented with one sentence processed by the five algorithms, and asked to assign an MOS [46] to each of them. Each HIT is assigned to 10 subjects.

Table 4.3 shows the results. As can be seen, DeepBeam outperforms the other algorithms by a large margin. In particular, DeepBeam achieves $>$ 4 MOS in some noise types. These results are very impressive because DeepBeam is only trained on simulated data. The real-world data differ significantly from the simulated data in terms of speakers, noise types and recording environment. Furthermore, some microphones are contaminated by strong electric noise, which is not accounted for in Eq. (2.8). Still, DeepBeam manages to perform well. The neural network used to be vulnerable to unseen scenarios, but DeepBeam has now made it robust.

Table 4.3: Realworld Data Evaluation Results.

Noise Type		N1	N2	N3	N4	N5
SNR (dB)	DeepBeam	**20.1**	**20.0**	**16.9**	**19.6**	**18.7**
	GRAB	18.9	17.4	12.4	18.5	17.4
	CLOSEST	10.0	10.0	10.0	10.0	10.0
	MVDR	10.8	16.5	7.72	14.0	13.4
	IVA	11.7	9.74	6.83	12.4	15.9
MOS	DeepBeam	**3.83**	**3.72**	**3.63**	**4.09**	**4.20**
	GRAB	3.10	3.06	2.93	3.71	3.45
	CLOSEST	2.74	2.68	3.02	3.55	3.50
	MVDR	2.05	2.40	2.28	2.71	2.62
	IVA	1.73	2.03	1.75	1.78	2.08

N1: cell phone N2: CombBind machine N3:paper shuffle

N4: door slide N5: footsteps

CHAPTER 5

DISCUSSION

5.1 Entropy Analysis for Bayesian WaveNet

The effectiveness of the prior model under the Bayesian framework can be further visualized and analyzed by computing the entropies of the estimated prior and posterior distribution of each sample. Specifically

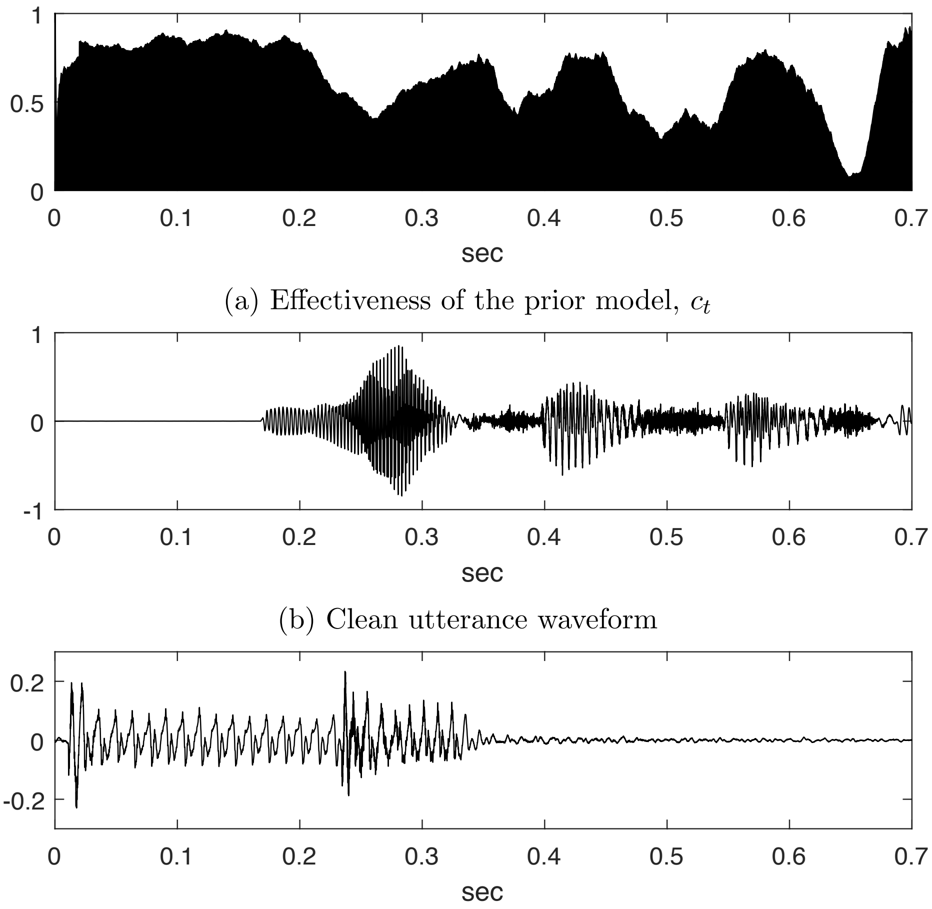
$$\begin{aligned} H_t^{(\mathrm{pr})} &= -\sum_{i=0}^{Q} \hat{p}(X_t = q_i | \hat{x}_{t-\tau_1:t-1}) \\ &\quad \cdot \log_2 \hat{p}(X_t = q_i | \hat{x}_{t-\tau_1:t-1}) \\ H_t^{(\mathrm{post})} &= -\sum_{i=0}^{Q} \hat{p}(X_t = q_i | \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \\ &\quad \cdot \log_2 \hat{p}(X_t = q_i | \hat{x}_{t-\tau_1:t-1}, y_{t-\tau_2:t+\tau_2}) \end{aligned} \tag{5.1}$$

Since the prediction of a sample is more uncertain if the entropy of the corresponding distribution is high, we can conclude that the prior model plays a more important role than the likelihood model at time t if $H_t^{(\mathrm{pr})} < H_t^{(\mathrm{post})}$. Hence we define a prior effectiveness function

$$e_t = \mathbb{1}\left(H_t^{(\mathrm{pr})} < H_t^{(\mathrm{post})}\right) \tag{5.2}$$

to depict the real-time effectiveness of the prior model. e_t is further smoothed by a 20 ms moving average filter.

In Figure 5.1a, using the entropies of the predicted distributions for each sample from the prior model and the likelihood model respectively, a 0-1 vector indicating whether the prior model is more certain than the likelihood model about each predicted sample was computed and then smoothed by a rectangular window of 20 ms. For example, a level of 0.8 at some sample



(a) Effectiveness of the prior model, c_t

(b) Clean utterance waveform

(c) Noise waveform

Figure 5.1: The prior effectiveness function (Eq. (5.2)) of a speech segment, smoothed by a 20 ms moving average filter, with its corresponding utterance and noise.

point indicates that the prior model is more certain than the likelihood model 80% of the time within 20 ms around this sample point.

Figure 5.1 shows the smoothed e_t of a test speech segment (a), as well as its corresponding clean speech (b) and noise (c) waveforms. There are two important observations. First, the prior model is more effective when the SNR is low, as can be seen from the segment before 0.25 s. This is because when the SNR is high enough, the likelihood model can simply pass noisy observation through, which does not rely much on the prior model.

Second, the prior model is more effective after the onset of vowels or voiced consonants. Accordingly, the likelihood model is more effective during unvoiced consonants or at the onset of speech activities, as can be seen from dips in the effectiveness function at around 0.4 s, 0.5 s and 0.65 s. This is because the voiced speech is well structured, so the prior model knows what comes next once it recognizes the phone. On the other hand, the prior model is less certain about the unvoiced phones because they are stochastic and can be easily confused with noise.

5.2 Empirical Convergence Analysis for Deep Beamformer

In order to empirically test whether DeepBeam has a good convergence property, 10 sets of eight-channel simulated data are generated with the S1 setting and $E_r = 10$. To study different numbers of channels, in each sub-test, K channels are randomly drawn from each set of data for DeepBeam prediction, and the resulting SNR convergence curves of the 10 sets are averaged. K runs from 3 to 8.

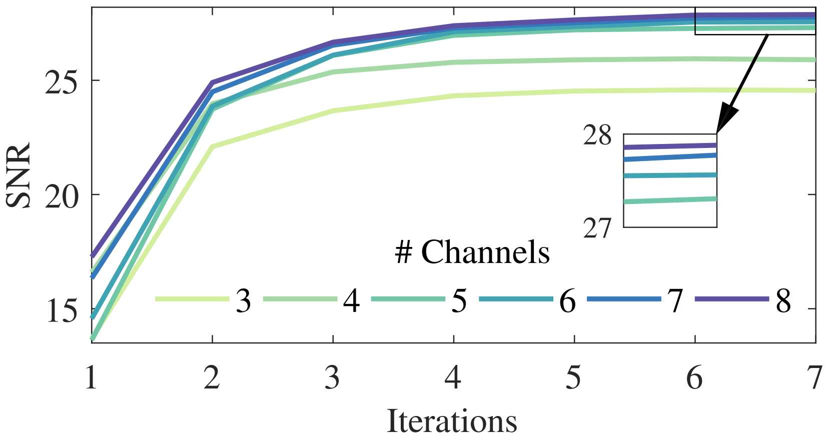


Figure 5.2: SNR convergence curves with different numbers of channels.

Figure 5.2 shows all the averaged convergence curves. As can be seen, DeepBeam converges well in all the sub-tests, which supports our convergence discussions in section 2.2.3. Also, the more channels DeepBeam has, the higher convergence level it can reach, which shows that DeepBeam is able to accommodate different numbers of channels using only one monaural network. We also see that the marginal benefit of having one more channel diminishes.

CHAPTER 6

CONCLUSION

We proposed a WaveNet enhancement model that directly operates on speech waveforms and exploited its generalizability to completely unseen noise. The results showed that our proposed model is able to produce clean speech and outperforms the DNN-IRM model under small-sized training data in terms of generalizability owing to the effectiveness of the prior model.

We also proposed DeepBeam as a solution to multi-channel speech enhancement with ad-hoc sensors. DeepBeam combines the complementary beamforming and deep learning techniques, and has exhibited superior performance and generalizability in terms of noise suppression, reverberation cancellation and perceptual quality. DeepBeam is a step closer toward resolving the longstanding tradeoff of perceptual quality and generalizability in deep enhancement networks, and demonstrates the power of bridging the signal processing and deep learning areas.

REFERENCES

[1] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, “An experimental study on speech enhancement based on deep neural networks,” *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.

[2] D. Liu, P. Smaragdis, and M. Kim, “Experiments on deep learning for speech denoising,” in *INTERSPEECH*, 2014, pp. 2685–2689.

[3] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, “Speech enhancement based on deep denoising autoencoder,” in *INTERSPEECH*, 2013, pp. 436–440.

[4] A. Kumar and D. Florêncio, “Speech enhancement in multiple-noise conditions using deep neural networks,” in *INTERSPEECH*, 2016, pp. 3738–3742.

[5] J. Chen, Y. Wang, S. E. Yoho, D. Wang, and E. W. Healy, “Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises,” *The Journal of the Acoustical Society of America*, vol. 139, no. 5, pp. 2604–2612, 2016.

[6] J. Chen and D. Wang, “Long short-term memory for speaker generalization in supervised speech separation,” in *INTERSPEECH*, 2016, pp. 3314–3318.

[7] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, “Deep learning for monaural speech separation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1562–1566.

[8] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, “Discriminatively trained recurrent neural networks for single-channel speech separation,” in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2014, pp. 577–581.

[9] Z. Ou and Y. Zhang, “Probabilistic acoustic tube: a probabilistic generative model of speech for speech analysis/synthesis.” in *AISTATS*, 2012, pp. 841–849.

[10] Y. Ephraim, “A Bayesian estimation approach for speech enhancement using hidden Markov models,” *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 725–735, 1992.

[11] H. Sameti, H. Sheikhzadeh, L. Deng, and R. L. Brennan, “HMM-based strategies for enhancement of speech signals embedded in nonstationary noise,” *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 5, pp. 445–455, 1998.

[12] D. Y. Zhao and W. B. Kleijn, “HMM-based gain modeling for enhancement of speech in noise,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 882–892, 2007.

[13] A. Kundu, S. Chatterjee, A. S. Murthy, and T. Sreenivas, “GMM based Bayesian approach to speech enhancement in signal/transform domain,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 4893–4896.

[14] R. Martin and C. Breithaupt, “Speech enhancement in the DFT domain using Laplacian speech priors,” in *International Workshop on Acoustic Echo and Noise Control (IWAENC)*, vol. 3, 2003, pp. 87–90.

[15] T. Lotter and P. Vary, “Speech enhancement by MAP spectral amplitude estimation using a super-Gaussian speech model,” *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 1110–1126, 2005.

[16] T. Kim, H. T. Attias, S.-Y. Lee, and T.-W. Lee, “Blind source separation exploiting higher-order frequency dependencies,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 70–79, 2007.

[17] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, “Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1626–1641, 2016.

[18] B. W. Gillespie, H. S. Malvar, and D. A. Florêncio, “Speech dereverberation via maximum-kurtosis subband adaptive filtering,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 6, 2001, pp. 3701–3704.

[19] K. Kumatani, J. McDonough, B. Rauch, D. Klakow, P. N. Garner, and W. Li, “Beamforming with a maximum negentropy criterion,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 5, pp. 994–1008, 2009.

[20] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.

[21] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*. Springer Science & Business Media, 2013.

[22] S. Markovich-Golan, A. Bertrand, M. Moonen, and S. Gannot, "Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks," *Signal Processing*, vol. 107, pp. 4–20, 2015.

[23] Y. Zhang, D. Florêncio, and M. Hasegawa-Johnson, "Glottal model based speech beamforming for ad-hoc microphone arrays," *INTERSPEECH*, pp. 2675–2679, 2017.

[24] K. Qian, Y. Zhang, S. Chang, X. Yang, D. Florêncio, and M. Hasegawa-Johnson, "Speech enhancement using Bayesian Wavenet," *INTERSPEECH*, pp. 2013–2017, 2017.

[25] D. Rethage, J. Pons, and X. Serra, "A Wavenet for speech denoising," *arXiv preprint arXiv:1706.07162*, 2017.

[26] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[27] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 196–200.

[28] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks." in *INTERSPEECH*, 2016, pp. 1981–1985.

[29] X. Xiao, S. Zhao, D. L. Jones, E. S. Chng, and H. Li, "On time-frequency mask estimation for MVDR beamforming with application in robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3246–3250.

[30] X. Zhang, Z.-Q. Wang, and D. Wang, "A speech enhancement algorithm by iterating single-and multi-microphone processing and its application to robust ASR," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 276–280.

[31] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, "DNN-based speech mask estimation for eigenvector beamforming," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 66–70.

[32] "Pulse code modulation (PCM) of voice frequencies," *International Telecommunication Union (ITU)*, 1988.

[33] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, and K. Kavukcuoglu, “Conditional image generation with pixelCNN decoders,” in *Advances in Neural Information Processing Systems*, 2016, pp. 4790–4798.

[34] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[35] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series.* MIT Press, Cambridge, MA, 1949.

[36] T. L. Paine, P. Khorrami, S. Chang, Y. Zhang, P. Ramachandran, M. A. Hasegawa-Johnson, and T. S. Huang, “Fast wavenet generation algorithm,” *arXiv preprint arXiv:1611.09482*, 2016.

[37] S. King and V. Karaiskos, “The Blizzard Challenge 2013,” *Proc. Blizzard Workshop*, 2013.

[38] “Freesound,” https://freesound.org/, 2015.

[39] G. Hu, “100 nonspeech sounds,” http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html, 2015.

[40] “FreeSFX,” http://www.freesfx.co.uk/, 2017.

[41] J. B. Allen and D. A. Berkley, “Image method for efficiently simulating small-room acoustics,” *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[42] E. A. Lehmann and A. M. Johansson, “Diffuse reverberation model for efficient image-source simulation of room impulse responses,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1429–1439, 2010.

[43] J. Yamagishi, “English multi-speaker corpus for CSTR voice cloning toolkit,” http://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html.

[44] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, “DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1,” *NASA STI/Recon Technical Report n*, vol. 93, 1993.

[45] L. Griffiths and C. Jim, “An alternative approach to linearly constrained adaptive beamforming,” *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 1, pp. 27–34, 1982.

[46] F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer, "CrowdMOS: An approach for crowdsourcing mean opinion score studies," in *ICASSP*, 2011, pp. 2416–2419.

[47] C. Févotte, R. Gribonval, and E. Vincent, "BSS_EVAL toolbox user guide–revision 2.0," 2005.