MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Unified Architecture for Multichannel End-to-End Speech Recognition with Neural Beamforming

Ochiai, T.; Watanabe, S.; Hori, T.; Hershey, J.R.; Xiao, X.

TR2017-192 October 2017

Abstract

This paper proposes a unified architecture for end-to-end automatic speech recognition (ASR) to encompass microphone-array signal processing such as a state-of-the-art neural beamformer within the end-to-end framework. Recently, the end-to-end ASR paradigm has attracted great research interest as an alternative to conventional hybrid paradigms with deep neural networks and hidden Markov models. Using this novel paradigm, we simplify ASR architecture by integrating such ASR components as acoustic, phonetic, and language models with a single neural network and optimize the overall components for the end-to-end ASR objective: generating a correct label sequence. Although most existing end-to-end frameworks have mainly focused on ASR in clean environments, our aim is to build more realistic end-to-end systems in noisy environments. To handle such challenging noisy ASR tasks, we study multichannel end-to-end ASR architecture, which directly converts multichannel speech signal to text through speech enhancement. This architecture allows speech enhancement and ASR components to be jointly optimized to improve the end-to-end ASR objective and leads to an end-to-end framework that works well in the presence of strong background noise. We elaborate the effectiveness of our proposed method on the multichannel ASR benchmarks in noisy environments (CHiME-4 and AMI). The experimental results show that our proposed multichannel end-to-end system obtained performance gains over the conventional end-to-end baseline with enhanced inputs from a delay-and-sum beamformer (i.e., BeamformIT) in terms of character error rate. In addition, further analysis shows that our neural beamformer, which is optimized only with the end-to-end ASR objective, successfully learned a noise suppression function.

IEEE Journal of Selected Topics in Signal Processing

Unified Architecture for Multichannel End-to-End Speech Recognition with Neural Beamforming

Tsubasa Ochiai, *Student Member, IEEE,* Shinji Watanabe, *Senior Member, IEEE,* Takaaki Hori, *Senior Member, IEEE,* John R. Hershey, *Senior Member, IEEE* and Xiong Xiao, *Member, IEEE*

***Abstract*—This paper proposes a unified architecture for end-to-end automatic speech recognition (ASR) to encompass microphone-array signal processing such as a state-of-the-art neural beamformer within the end-to-end framework. Recently, the end-to-end ASR paradigm has attracted great research interest as an alternative to conventional hybrid paradigms with deep neural networks and hidden Markov models. Using this novel paradigm, we simplify ASR architecture by integrating such ASR components as acoustic, phonetic, and language models with a single neural network and optimize the overall components for the end-to-end ASR objective: generating a correct label sequence. Although most existing end-to-end frameworks have mainly focused on ASR in clean environments, our aim is to build more realistic end-to-end systems in noisy environments. To handle such challenging noisy ASR tasks, we study multichannel end-to-end ASR architecture, which directly converts multichannel speech signal to text through speech enhancement. This architecture allows speech enhancement and ASR components to be jointly optimized to improve the end-to-end ASR objective and leads to an end-to-end framework that works well in the presence of strong background noise. We elaborate the effectiveness of our proposed method on the multichannel ASR benchmarks in noisy environments (CHiME-4 and AMI). The experimental results show that our proposed multichannel end-to-end system obtained performance gains over the conventional end-to-end baseline with enhanced inputs from a delay-and-sum beamformer (i.e., BeamformIT) in terms of character error rate. In addition, further analysis shows that our neural beamformer, which is optimized only with the end-to-end ASR objective, successfully learned a noise suppression function.**

***Index Terms*—multichannel end-to-end ASR, neural beamformer, encoder-decoder network.**

I. Introduction

Over the last decade, with the advent of deep neural networks (DNN) in automatic speech recognition (ASR), ASR performance has significantly improved compared to conventional systems based on Gaussian mixture models (GMMs) and hidden Markov models (HMMs) [1]. Although such deep learning-based approaches have replaced several components of the conventional ASR system, current systems continue to adopt a complicated module-based architecture that consists of several separate components, such as acoustic, phonetic, and language models. To build such a complicated architecture, we require wide and deep knowledge about each component, which makes it difficult to develop and tune ASR systems for every applications.

T. Ochiai is with Doshisha University.

S. Watanabe, T. Hori and John R. Hershey are with Mitsubishi Electric Research Laboratories (MERL).

X. Xiao is with Nanyang Technological University, Singapore.

Manuscript received Month Day, Year; revised Month Day, Year.

Recently, as an alternative to such complicated architecture, an end-to-end ASR paradigm has attracted great research interest because it simplifies the above architecture with a single neural network-based architecture [2]–[11]. One of promising directions is an attention-based encoder-decoder framework, which integrates all relevant components using recurrent neural networks (RNNs) and an attention mechanism [2]–[8]. Using the attention mechanism, the framework deals with dynamic time alignment problems within the network and solves the ASR problem as a sequence-to-sequence mapping problem from acoustic feature to word/character label sequences. In addition to the simplified system architecture, another important motivation of the end-to-end framework is that the entire inference procedure can be consistently optimized to improve such final ASR objectives as word/character error rate (WER/CER).

However, previous research on end-to-end frameworks mainly focused on the ASR problem in a single-channel setup without speech enhancement. Considering real world applications, we must also study such frameworks in a multichannel setup with speech enhancement. Actually, recent benchmark studies show that multichannel processing with microphone-array speech enhancement techniques (especially beamforming methods) produces substantial improvements in the presence of strong background noise for conventional HMM/DNN hybrid systems [12], [13]. In light of the above trends, in this paper, we extend the existing attention-based encoder-decoder framework by integrating multichannel speech enhancement components into the end-to-end framework and propose a multichannel end-to-end ASR, which directly converts multichannel speech signal to text through speech enhancement. As a speech enhancement component of our multichannel ASR system, we adopt a recently proposed beamforming technique using neural networks, which we call a neural beamformer. Because a neural beamformer can be formalized as a fully differentiable network, the beamforming component can be jointly optimized with the end-to-end ASR component, based on the backpropagated gradients from the final ASR objective.

Recent studies on neural beamformers can be categorized into two types: 1) beamformers with a filter estimation network [14]–[16] and 2) those with a mask estimation network [17]–[24]. In both approaches, an enhanced signal is obtained by applying linear filters in the time-frequency domain, which is based on the conventional formalization of the filter-and-sum beamformer. The main difference between them is how to produce such linear filters using a neural network. In the former approach, the network directly estimates the multichan-

nel filter coefficients. In the latter approach, the network first estimates the time-frequency masks and then predicts speech and noise statistics based on the estimated masks. Finally, with these statistics, the multichannel filter coefficients are computed based on the well-studied beamforming designs, such as the minimum variance distortionless response (MVDR) beamformer [25], [26] and the generalized eigenvalue (GEV) beamformer [27].

In this paper, we propose to use both types of neural beamformers with mask and filter estimation networks as the speech enhancement component for the end-to-end framework. However, motivated by the successes of the mask-based beamforming approaches [17]–[19], [28] in recent noisy ASR benchmarks (e.g., CHiME 3 and 4 challenges), this paper mainly focuses on the mask-based neural beamformer.

Our mask-based neural beamformer adopts a MVDR formalization given a reference microphone [26] since computing the derivatives is relatively simple. Also, this beamformer has an additional neural network-based attention mechanism for the reference microphone selection, which obtains robustness against microphone geometries and speaker positions. This allows the entire procedures of the neural beamformer, including the reference selection, to be invariant to microphone geometries including the number of channels, the microphone locations, and the microphone ordering. Therefore, our proposed multichannel end-to-end ASR can deal with input signals from various microphone geometries without reconfiguration and re-training. Of course, because the channel attention mechanism is also formalized as a differentiable network, the entire procedures of the neural beamformer can be jointly optimized with the end-to-end ASR based on the backpropagated gradients from the end-to-end ASR objective.

This paper intends to extend our previous study [29], which outlines the formalization of our multichannel end-to-end ASR and showed its basic experimental results. That is, the paper details its formalization by providing pseudo codes of proposed main algorithms and descriptions of how to implement complex-valued procedures in neural beamformers based on the real-valued functions using existing deep learning libraries. The paper also shows additional experimental comparisons and analyses to show the effectiveness of our multichannel end-to-end ASR.

In general, the term "end-to-end" has a wide meaning. In this paper, we define it based on the following two characteristics: (1) all the procedures from input sequences to output sequences are represented as a single neural network-based architecture (fully differentiable network), and (2) the entire network can be consistently optimized with a single ASR-level objective. Our proposed multichannel end-to-end ASR architecture satisfies these characteristics. More specifically, all the procedures from the front-end multichannel speech enhancement to back-end speech recognition are represented as single neural network-based architecture, and the entire network is optimized based on the backpropagated gradients from the final ASR objective.

The remainder of this paper is summarized as follows. In Section II, we briefly explain the conventional attention-based encoder-decoder framework. Section III describes the formalizations of the adopted neural beamformers and introduces the attention-based reference selection mechanism. Section IV connects the components described in Sections II and III and describes the overall processing chain for our proposed multichannel end-to-end ASR. In Sections VI and VII, we explain the experiments of the multichannel ASR benchmarks in noisy environments (CHiME-4 and AMI) and demonstrate the effectiveness of our proposed method. We conclude this paper in Section VIII. Additionally, in Appendix, we describe implementation details for the neural beamformers, an additional experiment with a conventional HMM/DNN hybrid framework, and a notation list.

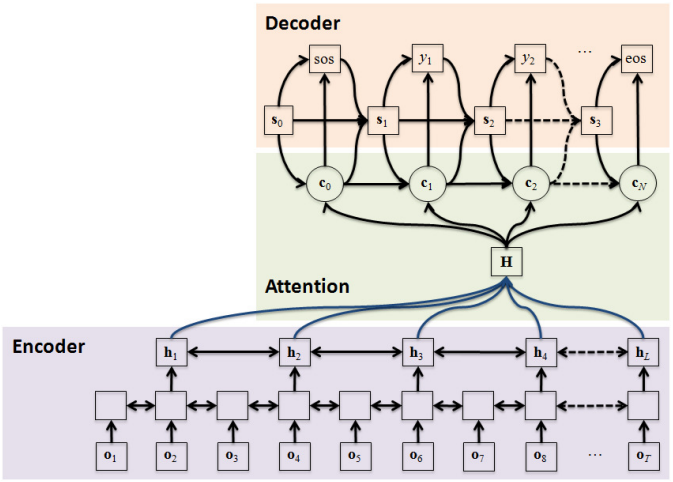


Fig. 1. Structure of attention-based encoder-decoder framework. Encoder transforms input feature sequence O into high-level feature sequence H, and then decoder generates output label sequence Y through attention mechanism.

II. Attention-based encoder-decoder networks

This section explains a conventional attention-based encoder-decoder framework, which directly deals with variable length input and output sequences. The framework consists of two RNNs, an encoder and a decoder, both of which are connected by an attention mechanism. Fig. 1 illustrates its overall architecture.

Given feature sequence $O = \{\mathbf{o}_t \in \mathbb{R}^{D_\mathrm{O}} | t = 1, \cdots, T\}$, where $\mathbf{o}_t$ is a D_O-dimensional feature vector (e.g., a log Mel filterbank) at input time step t and T is the input sequence length, the network estimates the *a posteriori* probabilities for output label sequence $Y = \{y_n \in \mathcal{V} | n = 1, \cdots, N\}$, where y_n is a label symbol (e.g., a character) at output time step n, N is the output sequence length, and $\mathcal{V}$ is a set of labels as follows:

$$P(Y|O) = \prod_n P(y_n|O, y_{1:n-1}), \tag{1}$$

$$H = \mathrm{Encoder}(O), \tag{2}$$

$$\mathbf{c}_n = \mathrm{Attention}(\mathbf{a}_{n-1}, \mathbf{s}_{n-1}, H), \tag{3}$$

$$P(y_n|O, y_{1:n-1}) = \mathrm{Decoder}(\mathbf{c}_n, \mathbf{s}_{n-1}, y_{1:n-1}), \tag{4}$$

where $y_{1:n-1}$ is a label sequence that consists of y_1 through y_{n-1}.

For input sequence O, the encoder RNN in Eq. (2) first transforms it to L-length feature sequence $H = \{\mathbf{h}_l \in \mathbb{R}^{D_{\mathrm{H}}} | l = 1, \cdots, L\}$, where $\mathbf{h}_l$ is a D_{H}-dimensional state vector of the encoder's top layer at subsampled time step l. Next the attention mechanism in Eq. (3) integrates all encoder outputs H into a D_{H}-dimensional context vector $\mathbf{c}_n \in \mathbb{R}^{D_{\mathrm{H}}}$ using L-dimensional attention weight vector $\mathbf{a}_n \in [0,1]^L$ that represents a soft alignment of the encoder outputs at output time step n. Then the decoder RNN in Eq. (4) updates hidden state $\mathbf{s}_n$, estimates the *a posteriori* probability for output label y_n at output time step n, and further estimates the *a posteriori* probabilities for output sequence Y, based on RNN recursiveness. For the attention mechanism, we adopted a location-based attention mechanism [3] (See Appendix A).

Here, special tokens for start-of-sentence (sos) and end-of-sentence (eos) are added to label set $\mathcal{V}$. The decoder starts the recurrent computation with the sos label and continues to generate output labels until the eos label is emitted.

In this framework, the whole network , including the encoder, the attention mechanism, and the decoder, can be optimized to generate a correct label sequence. Such consistent optimization of all the relevant procedures is the main motivation of the end-to-end framework. For more details of each component (i.e., the encoder, the attention mechanism, and the decoder), refer to our previous paper [29] or the original papers on attention-based encoder-decoder networks [4], [5].

III. Neural beamformers

A. Overview

This section explains neural beamformer techniques, which are integrated with the encoder-decoder network in the following section. This paper uses frequency-domain beamformers [14] rather than time-domain ones [15], because the frequency-domain beamformers achieve significant computational complexity reduction in multichannel neural processing [30].

Let $x_{t,f,c} \in \mathbb{C}$ be an STFT coefficient of c-th channel noisy signal at time-frequency bin (t, f), and let $g_{t,f,c} \in \mathbb{C}$ be a corresponding beamforming filter coefficient. In the frequency domain representation, a filter-and-sum beamformer obtains enhanced STFT coefficient $\hat{x}_{t,f} \in \mathbb{C}$ as follows:

$$\hat{x}_{t,f} = \begin{cases} \mathbf{g}_{t,f}^{\dagger} \mathbf{x}_{t,f} & \text{(time-variant filter)} \\ \mathbf{g}_{f}^{\dagger} \mathbf{x}_{t,f} & \text{(time-invariant filter),} \end{cases} \tag{5}$$

where $\mathbf{x}_{t,f} = \{x_{t,f,c}\}_{c=1}^{C} \in \mathbb{C}^C$ is the spatial vector of the signals obtained from all the microphones for each time-frequency bin (t, f). $\mathbf{g}_{t,f} = \{g_{t,f,c}\}_{c=1}^{C} \in \mathbb{C}^C$ and $\mathbf{g}_f = \{g_{t,f,c}\}_{c=1}^{C} \in \mathbb{C}^C$ are corresponding *time-variant* and *time-invariant* filter coefficients, respectively. C is the numbers of channels. $\dagger$ represents the conjugate transpose.

In this paper, we adopt two types of neural beamformers, which basically follow Eq. (5): 1) with a filter estimation network and 2) with a mask estimation network. The main difference between them is how to produce the filter coefficients: $\mathbf{g}_{t,f}$ or $\mathbf{g}_f$. The following subsections describe each approach.

B. Filter estimation network approach

A neural beamformer with a filter estimation network directly estimates time-variant filter coefficients $\{\mathbf{g}_{t,f}\}_{t=1,f=1}^{T,F}$ as network outputs, where F is the number of dimensions of the STFT signals.

The following Algorithm 1 summarizes the overall procedures to obtain the enhanced features, and Fig. 2(a) illustrates an overview of the procedures. The main part of this algorithm is to predict complex-valued filter coefficients $\mathbf{g}_{t,f}$ with a real-valued neural network, Filternet$(\cdot)$, which is described below.

Algorithm 1 Overall procedures of neural beamformer with filter estimation network

Require: multichannel STFT input sequences $\{X_c\}_{c=1}^{C}$
1: $\{\mathbf{g}_{t,f}\}_{t=1,f=1}^{T,F} = \text{Filternet}(\{X_c\}_{c=1}^{C})$ ▷ Eqs. (6)-(8)
2: **for** $t = 1$ to T **do**
3: **for** $f = 1$ to F **do**
4: $\hat{x}_{t,f} = \mathbf{g}_{t,f}^{\dagger} \mathbf{x}_{t,f}$ ▷ Eq. (5)
5: **end for**
6: **end for**
7: return $\hat{X} = \{\hat{x}_{t,f}\}_{t=1,f=1}^{T,F}$

1) Filter estimation network: This approach uses a single real-valued BLSTM network to predict the real and imaginary parts of the complex-valued filter coefficients at every time step. We introduce $2FC$-dimensional output layers to separately compute the real and imaginary parts of the filter coefficients.

Let $\bar{\mathbf{x}}_t = \{\Re(\mathbf{x}_{t,f}), \Im(\mathbf{x}_{t,f})\}_{f=1}^{F} \in \mathbb{R}^{2FC}$ be an input feature of a $2FC$-dimensional real-valued vector for the BLSTM network, which is obtained by concatenating the real and imaginary parts of all STFT coefficients in all channels at time step t. Then the network outputs time-variant filter coefficients $\mathbf{g}_{t,f}$ as follows:

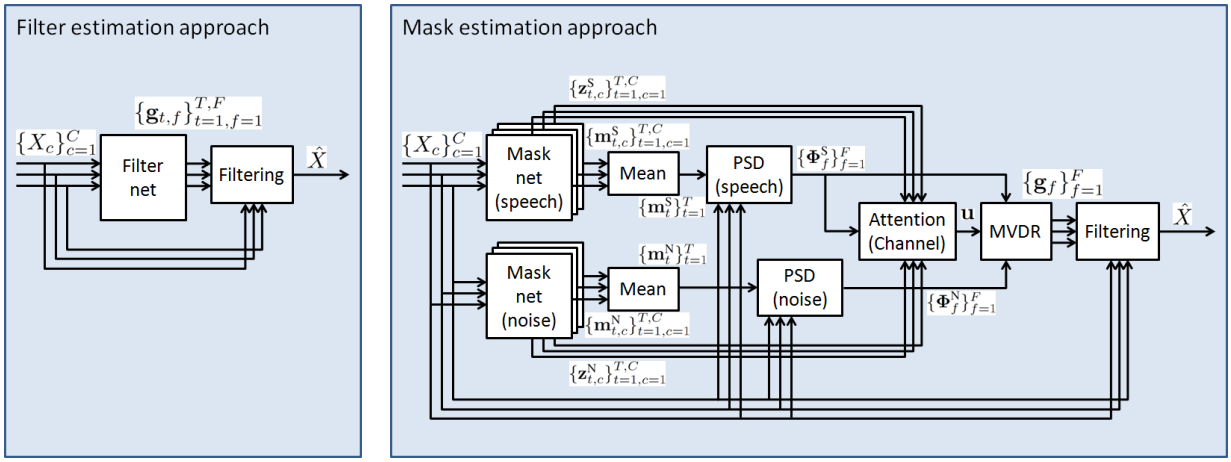
$$Z = \mathrm{BLSTM}(\{\bar{\mathbf{x}}_t\}_{t=1}^{T}), \tag{6}$$

$$\Re(\mathbf{g}_{t,f}) = \tanh(\mathbf{W}_f^{\Re} \mathbf{z}_t + \mathbf{b}_f^{\Re}), \tag{7}$$

$$\Im(\mathbf{g}_{t,f}) = \tanh(\mathbf{W}_f^{\Im} \mathbf{z}_t + \mathbf{b}_f^{\Im}), \tag{8}$$

where $Z = \{\mathbf{z}_t \in \mathbb{R}^{D_{\mathrm{Z}}} | t = 1, \cdots, T\}$ is a sequence of the D_{Z}-dimensional output vectors of the BLSTM network. $\Re(\mathbf{g}_{t,f})$ and $\Im(\mathbf{g}_{t,f})$ are the real and imaginary parts of the filter coefficients. $\mathbf{W}_f^{\Re} \in \mathbb{R}^{C \times D_{\mathrm{Z}}}$ and $\mathbf{W}_f^{\Im} \in \mathbb{R}^{C \times D_{\mathrm{Z}}}$ are the weight matrices that output real and imaginary part of the filter coefficients for frequency f, and $\mathbf{b}_f^{\Re} \in \mathbb{R}^C$ and $\mathbf{b}_f^{\Im} \in \mathbb{R}^C$ are their corresponding bias vectors. Eqs. (6)-(8) correspond to Filternet$(\cdot)$ in Algorithm 1. Using estimated filters $\{\mathbf{g}_{t,f}\}_{t=1,f=1}^{T,F}$, the enhanced STFT coefficients $\{\hat{x}_{t,f}\}_{t=1,f=1}^{T,F}$ are obtained based on Eq. (5).

2) Remarks: This approach has several possible issues due to its formalization. The first issue is the high flexibility of estimated filters $\{\mathbf{g}_{t,f}\}_{t=1,f=1}^{T,F}$, which are composed of a large number of unconstrained variables ($2TFC$) estimated from few observations. This causes problems, such as training difficulties and over-fitting. The second is that the network structure depends on the number and order of the channels. Therefore, a new filter estimation network has to be trained when we change microphone configurations.



(a) Filter estimation approach (b) Mask estimation approach

Fig. 2. Structures of neural beamformers: (a) filter estimation network, which directly estimates the filter coefficients; (b) mask estimation network, which estimates time-frequency masks and gets filter coefficients based on MVDR formalization.

C. Mask estimation network approach

The neural beamformer with a mask estimation network first estimates the time-frequency masks. Then cross-channel power spectral density (PSD) matrices (also known as spatial covariance matrices) are predicted based on the estimated masks. Finally, they are used to compute the time-invariant filter coefficients $\{\mathbf{g}_f\}_{f=1}^{F}$ based on the well-studied MVDR formalization.

The key point of the mask-based neural beamformer is that it constrains the estimated filters based on well-founded array signal processing principles, which can solve/suppress the issues described in Section III-B2. This is the main difference between this approach and the filter estimation network approach described in Section III-B. Also, mask-based beamforming approaches have achieved great performance in recent noisy ASR benchmarks [17]–[19], [28]. Motivated by this background, we focus on a neural beamformer with a mask estimation network more than a filter estimation network.

Algorithm 2 summarizes the overall procedures to obtain the enhanced features, and Fig. 2(b) illustrates an overview of the procedures[1]. Each procedure is described below.

1) MVDR formalization given reference microphones: This paper adopts one MVDR formalization given the reference microphones [26], which computes the time-invariant filter coefficients $\mathbf{g}_f$ in Eq. (5) as follows:

$$\mathbf{g}_f = \frac{(\mathbf{\Phi}^{\mathrm{N}}_f)^{-1}\mathbf{\Phi}^{\mathrm{S}}_f}{\mathrm{Tr}((\mathbf{\Phi}^{\mathrm{N}}_f)^{-1}\mathbf{\Phi}^{\mathrm{S}}_f)}\mathbf{u}, \tag{9}$$

where $\mathbf{\Phi}^{\mathrm{S}}_f \in \mathbb{C}^{C\times C}$ and $\mathbf{\Phi}^{\mathrm{N}}_f \in \mathbb{C}^{C\times C}$ are the PSD matrices for speech and noise signals, respectively. $\mathbf{u} \in \mathbb{R}^C$ is a one-hot vector representing a reference microphone, and $\mathrm{Tr}(\cdot)$ represents the matrix trace operation. Eq. (9) corresponds to $\mathrm{MVDR}(\cdot)$ in Algorithm 2.

[1]Due to space limitations, the procedures corresponding to State_Feat$(\cdot)$ and Spatial_Feat$(\cdot)$ in Algorithm 2 are not shown in Fig. 2(b).

Algorithm 2 Overall procedures of neural beamformer with mask estimation network

Require: multichannel STFT input sequences $\{X_c\}_{c=1}^{C}$

for $c = 1$ to C **do**
  $\{\mathbf{m}^{\mathrm{S}}_{t,c}\}_{t=1}^{T}, \{\mathbf{z}^{\mathrm{S}}_{t,c}\}_{t=1}^{T} = \mathrm{Masknet}^{\mathrm{S}}(X_c)$ ▷ Eqs. (12)-(13)
  $\{\mathbf{m}^{\mathrm{N}}_{t,c}\}_{t=1}^{T}, \{\mathbf{z}^{\mathrm{N}}_{t,c}\}_{t=1}^{T} = \mathrm{Masknet}^{\mathrm{N}}(X_c)$ ▷ Eqs. (14)-(15)
end for
for $t = 1$ to T **do**
  $\mathbf{m}^{\mathrm{S}}_t = \mathrm{Mean}(\{\mathbf{m}^{\mathrm{S}}_{t,c}\}_{c=1}^{C})$ ▷ Eq. (16)
  $\mathbf{m}^{\mathrm{N}}_t = \mathrm{Mean}(\{\mathbf{m}^{\mathrm{N}}_{t,c}\}_{c=1}^{C})$ ▷ Eq. (17)
end for
for $f = 1$ to F **do**
  $\mathbf{\Phi}^{\mathrm{S}}_f = \mathrm{PSD}(\{\mathbf{m}^{\mathrm{S}}_t\}_{t=1}^{T}, \{\mathbf{x}_{t,f}\}_{t=1}^{T})$ ▷ Eq. (10)
  $\mathbf{\Phi}^{\mathrm{N}}_f = \mathrm{PSD}(\{\mathbf{m}^{\mathrm{N}}_t\}_{t=1}^{T}, \{\mathbf{x}_{t,f}\}_{t=1}^{T})$ ▷ Eq. (11)
end for
for $c = 1$ to C **do**
  $\mathbf{q}_c = \mathrm{State\_Feat}(\{\mathbf{z}^{\mathrm{S}}_{t,c})\}_{t=1}^{T}, \{\mathbf{z}^{\mathrm{N}}_{t,c})\}_{t=1}^{T})$ ▷ Eq. (20)
  $\mathbf{r}_c = \mathrm{Spatial\_Feat}(\{\phi^{S}_{f,c,c'}\}_{f=1,c'=1}^{F,C})$ ▷ Eq. (21)
end for
$\mathbf{u} = \mathrm{Attend\_Channel}(\{\mathbf{q}_c\}_{c=1}^{C}, \{\mathbf{r}_c\}_{c=1}^{C})$ ▷ Eqs. (18)-(19)
for $f = 1$ to F **do**
  $\mathbf{g}_f = \mathrm{MVDR}(\mathbf{\Phi}^{\mathrm{S}}_f, \mathbf{\Phi}^{\mathrm{N}}_f, \mathbf{u})$ ▷ Eq. (9)
end for
for $t = 1$ to T **do**
  for $f = 1$ to F **do**
    $\hat{x}_{t,f} = \mathbf{g}_f^{\dagger}\mathbf{x}_{t,f}$ ▷ Eq. (5)
  end for
end for
return $\hat{X} = \{\hat{x}_{t,f}\}_{t=1,f=1}^{T,F}$

2) Mask-based estimation for PSD matrices: Let $m^{\mathrm{S}}_{t,f} \in [0,1]$ and $m^{\mathrm{N}}_{t,f} \in [0,1]$ respectively be the time-frequency masks for speech and noise signals. Based on a previous work [17], [28], the PSD matrices are robustly estimated using the

expectation with respect to time-frequency masks as follows:

$$\mathbf{\Phi}_f^{\mathrm{S}} = \frac{1}{\sum_{t=1}^{T} m_{t,f}^{\mathrm{S}}} \sum_{t=1}^{T} m_{t,f}^{\mathrm{S}} \mathbf{x}_{t,f} \mathbf{x}_{t,f}^{\dagger}, \tag{10}$$

$$\mathbf{\Phi}_f^{\mathrm{N}} = \frac{1}{\sum_{t=1}^{T} m_{t,f}^{\mathrm{N}}} \sum_{t=1}^{T} m_{t,f}^{\mathrm{N}} \mathbf{x}_{t,f} \mathbf{x}_{t,f}^{\dagger}. \tag{11}$$

Eqs. (10) and (11) correspond to $\mathrm{PSD}(\cdot)$ in Algorithm 2.

3) Mask estimation network: To estimate the time-frequency masks for every c-th channel ($\mathbf{m}_{t,c}^{\mathrm{S}} = \{m_{t,f,c}^{\mathrm{S}}\}_{f=1}^{F}$ and $\mathbf{m}_{t,c}^{\mathrm{N}} = \{m_{t,f,c}^{\mathrm{N}}\}_{f=1}^{F}$), we use two real-valued BLSTM networks: one for a speech mask and another for a noise mask. Unlike the filter estimation network, because the masks are estimated separately for each channel, $2F$-dimensional output layers are used to separately compute the real and imaginary parts of the time-frequency masks.

Let $\bar{\mathbf{x}}_{t,c} = \{\Re(x_{t,f,c}), \Im(x_{t,f,c})\}_{f=1}^{F} \in \mathbb{R}^{2F}$ be the $2F$-dimensional real-valued input features for the BLSTM networks, which is obtained by concatenating the real and imaginary parts of all the STFT features at c-th channel. Given input sequence $\bar{X}_c = \{\bar{\mathbf{x}}_{t,c} \in \mathbb{R}^{2F} | t = 1, \cdots, T\}$, each network outputs the time-frequency masks separately for each channel as follows:

$$Z_c^{\mathrm{S}} = \mathrm{BLSTM}^{\mathrm{S}}(\bar{X}_c), \tag{12}$$

$$\mathbf{m}_{t,c}^{\mathrm{S}} = \mathrm{sigmoid}(\mathbf{W}^{\mathrm{S}} \mathbf{z}_{t,c}^{\mathrm{S}} + \mathbf{b}^{\mathrm{S}}), \tag{13}$$

$$Z_c^{\mathrm{N}} = \mathrm{BLSTM}^{\mathrm{N}}(\bar{X}_c), \tag{14}$$

$$\mathbf{m}_{t,c}^{\mathrm{N}} = \mathrm{sigmoid}(\mathbf{W}^{\mathrm{N}} \mathbf{z}_{t,c}^{\mathrm{N}} + \mathbf{b}^{\mathrm{N}}), \tag{15}$$

where $Z_c^{\mathrm{S}} = \{\mathbf{z}_{t,c}^{\mathrm{S}} \in \mathbb{R}^{D_{\mathrm{Z}}} | t = 1, \cdots, T\}$ is a sequence of D_{Z}-dimensional output vectors of the BLSTM network for a speech mask over c-th channel's input sequence $\bar{X}_c$. Z_c^{N} is the BLSTM output sequence for a noise mask. $\mathbf{W}^{\mathrm{S}} \in \mathbb{R}^{F \times D_{\mathrm{Z}}}$ and $\mathbf{W}^{\mathrm{N}} \in \mathbb{R}^{F \times D_{\mathrm{Z}}}$ are the weight matrices that output speech and noise masks. $\mathbf{b}^{\mathrm{S}} \in \mathbb{R}^{F}$ and $\mathbf{b}^{\mathrm{N}} \in \mathbb{R}^{F}$ are their corresponding bias vectors. Eqs. (12) and (13) correspond to $\mathrm{Masknet}^{\mathrm{S}}(\cdot)$, while Eqs. (14) and (15) correspond to $\mathrm{Masknet}^{\mathrm{N}}(\cdot)$ in Algorithm 2.

After computing the speech and noise masks for each channel, mean masks $\mathbf{m}_t = \{m_{t,f}\}_{f=1}^{F}$ are obtained as follows:

$$\mathbf{m}_t^{\mathrm{S}} = \frac{1}{C} \sum_{c=1}^{C} \mathbf{m}_{t,c}^{\mathrm{S}}, \tag{16}$$

$$\mathbf{m}_t^{\mathrm{N}} = \frac{1}{C} \sum_{c=1}^{C} \mathbf{m}_{t,c}^{\mathrm{N}}. \tag{17}$$

Eqs. (16) and (17) correspond to $\mathrm{Mean}(\cdot)$ in Algorithm 2. These mean masks are used to predict PSD matrices ($\mathbf{\Phi}_f^{\mathrm{S}}$ and $\mathbf{\Phi}_f^{\mathrm{N}}$) in Eqs. (10) and (11).

The mask-based MVDR neural beamformer, given reference microphones, was previously proposed in [18], [19], but our neural beamformer further extends it with attention-based reference selection, which is described in the next subsection.

4) Attention-based selection for reference microphones: To incorporate the reference microphone selection in the neural beamformer framework, we apply the idea of an attention mechanism to estimate reference microphone vector $\mathbf{u}$ in Eq. (9). Based on an attention mechanism's characteristics, this allows the reference selection to work for arbitrary numbers and orders of channels.

To formalize the attention mechanism, we adopt two types of time-invariant and channel-dependent features: 1) time-averaged state feature $\mathbf{q}_c \in \mathbf{R}^{2D_{\mathrm{Z}}}$ and 2) PSD-based spatial feature $\mathbf{r}_c \in \mathbb{R}^{2F}$. With these feature vectors, reference microphone vector $\mathbf{u}$ is estimated as follows:

$$\tilde{k}_c = \tilde{\mathbf{w}}^{\mathrm{T}} \tanh(\mathbf{V}^{\mathrm{Q}} \mathbf{q}_c + \mathbf{V}^{\mathrm{R}} \mathbf{r}_c + \tilde{\mathbf{b}}), \tag{18}$$

$$u_c = \frac{\exp(\beta \tilde{k}_c)}{\sum_{c=1}^{C} \exp(\beta \tilde{k}_c)}, \tag{19}$$

where $\tilde{\mathbf{w}} \in \mathbb{R}^{1 \times D_{\mathrm{V}}}$, $\mathbf{V}^{\mathrm{Q}} \in \mathbb{R}^{D_{\mathrm{V}} \times 2D_{\mathrm{Z}}}$, and $\mathbf{V}^{\mathrm{R}} \in \mathbb{R}^{D_{\mathrm{V}} \times 2F}$ are trainable weight parameters, and $\tilde{\mathbf{b}} \in \mathbb{R}^{D_{\mathrm{V}}}$ is a trainable bias vector. β is the sharpening factor. Eqs. (18) and (19) correspond to $\mathrm{Attend\_Channel}(\cdot)$ in Algorithm 2.

Time-averaged state feature $\mathbf{q}_c$ is extracted from the BLSTM networks for the speech and noise masks in Eqs. (12) and (14) as follows:

$$\mathbf{q}_c = \frac{1}{T} \sum_{t=1}^{T} \{\mathbf{z}_{t,c}^{\mathrm{S}}, \mathbf{z}_{t,c}^{\mathrm{N}}\}, \tag{20}$$

Eq. (20) corresponds to $\mathrm{State\_Feat}(\cdot)$ in Algorithm 2.

PSD-based spatial feature $\mathbf{r}_c$, which incorporates the spatial information into the attention mechanism, is extracted from speech PSD matrix $\mathbf{\Phi}_f^{\mathrm{S}}$ in Eq. (10) as follows:

$$\mathbf{r}_c = \frac{1}{C-1} \sum_{c'=1, c' \neq c}^{C} \{\Re(\phi_{f,c,c'}^{\mathrm{S}}), \Im(\phi_{f,c,c'}^{\mathrm{S}})\}_{f=1}^{F}, \tag{21}$$

where $\phi_{f,c,c'}^{S} \in \mathbb{C}$ is the entry in the c-th row and the c'-th column of speech PSD matrix $\mathbf{\Phi}_f^{\mathrm{S}}$. To select a reference microphone, since the spatial correlation related to speech signals is more informative, we only use speech PSD matrix $\mathbf{\Phi}_f^{\mathrm{S}}$ as a feature. Eq. (21) corresponds to $\mathrm{Spatial\_Feat}(\cdot)$ in Algorithm 2.

Note that, in this mask-based MVDR neural beamformer, the masks for each channel are computed separately using the same BLSTM network, and the mask estimation network is independent of the channels. Similarly, the reference selection network is also independent of the channels. Therefore, the neural beamformer deals with input signals with arbitrary numbers and orders of channels without network re-training or re-configuration.

IV. Multichannel end-to-end ASR

A. Unified architecture

In this work, we propose a unified architecture of a multi-channel end-to-end ASR, which integrates all components with a single network architecture. We adopt neural beamformers (Section III) as a speech enhancement part and an attention-based encoder-decoder (Section II) as a ASR part. Fig. 3 overviews our proposed system.

The entire procedures to generate sequence of output labels Y from sequence of multichannel input signals $\{X_c\}_{c=1}^{C}$ are formalized as Algorithm 3:

Algorithm 3 Overall procedures of multichannel end-to-end ASR system

Require: multichannel STFT input sequences $\{X_c\}_{c=1}^{C}$
1: $\hat{X} = \mathrm{Enhance}(\{X_c\}_{c=1}^{C})$ ▷ Algorithm 1 or 2
2: $\hat{O} = \mathrm{Feature}(\hat{X})$ ▷ Eqs. (22)-(23)
3: $H = \mathrm{Encoder}(\hat{O})$ ▷ Eq. (2)
4: $\mathbf{s}_0 = \mathbf{0},\ \mathbf{c}_0 = \mathbf{0},\ \mathbf{a}_0 = \mathbf{0},\ y_0 = 0$ (sos)
5: $n = 1$
6: **repeat**
7: $\quad \mathbf{c}_n = \mathrm{Attention}(\mathbf{a}_{n-1}, \mathbf{s}_{n-1}, H)$ ▷ Eq. (3)
8: $\quad y_n = \mathrm{Decoder}(\mathbf{c}_n, \mathbf{s}_{n-1}, y_{1:n-1})$ ▷ Eq. (4)
9: $\quad n = n + 1$
10: **until** eos is emitted
11: return Y

$\mathrm{Enhance}(\cdot)$ is a speech enhancement function realized by the neural beamformer with the filter estimation network (Algorithm 1) or the mask estimation network (Algorithm 2).

$\mathrm{Feature}(\cdot)$ is a feature extraction function that connects the neural beamformer and the encoder-decoder network . In this work, followed by the use of (log) Mel filterbank-based features in previous studies (e.g., a single-channel end-to-end ASR setup [3], [5], a single-channel joint training setup of speech enhancement and HMM/DNN hybrid [31], and a multichannel HMM/DNN hybrid setup [14], [21]), we adopt a normalized log Mel filterbank as an input acoustic feature of the encoder-decoder network. In other words, the feature extraction function transforms the enhanced STFT coefficients that are output from the front-end neural beamformer to the enhanced acoustic feature (i.e., normalized log Mel filterbank) for inputting to the back-end attention-based encoder-decoder network. Enhanced acoustic feature $\hat{\mathbf{o}}_t \in \mathbb{R}^{D_\mathrm{O}}$ was obtained from enhanced STFT coefficients $\hat{\mathbf{x}}_t \in \mathbb{C}^F$ as follows:

$$\mathbf{p}_t = \{\Re(\hat{x}_{t,f})^2 + \Im(\hat{x}_{t,f})^2\}_{f=1}^{F}, \tag{22}$$

$$\hat{\mathbf{o}}_t = \mathrm{Norm}(\log(\mathrm{Mel}(\mathbf{p}_t))), \tag{23}$$

where $\mathbf{p}_t \in \mathbb{R}^F$ is a real-valued vector of the power spectrum of the enhanced signal at time step t. $\mathrm{Mel}(\cdot)$ represents the operation of $D_\mathrm{O} \times F$ Mel matrix multiplication, and $\mathrm{Norm}(\cdot)$ represents the operation of the global mean and variance normalization so that the mean and variance of each dimension become 0 and 1.

Note that since computation of the normalized log Mel filterbank is fully differentiable, the gradients from the speech recognition part (i.e., the attention-based encoder-decoder network) can be backpropagated to the speech enhancement part (i.e., the neural beamformer)[2].

[2]Because the computation of the log Mel transformation is differentiable, it can also be optimized similarly to the other network functions (i.e., $\mathrm{Enhance}(\cdot)$, $\mathrm{Encoder}(\cdot)$, $\mathrm{Attention}(\cdot)$, and $\mathrm{Decoder}(\cdot)$). However, in our preliminary experiments, the optimization of the log Mel transformation did not improve the performance (it caused underfitting). Therefore, in this paper, we fixedly adopted the standard log Mel transformation.

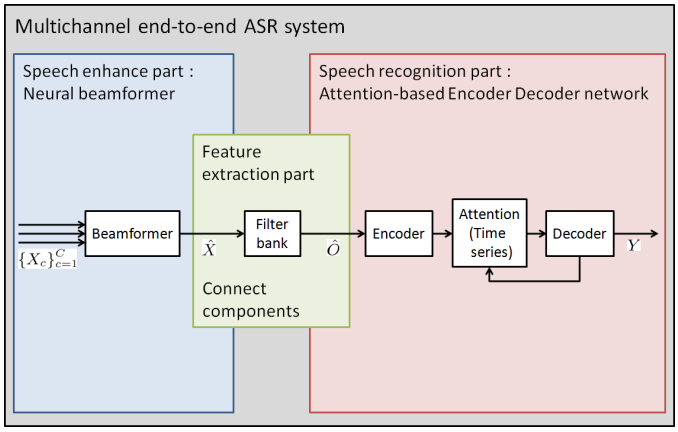


Fig. 3. Overview of our proposed multichannel end-to-end ASR system, which converts multichannel speech signal to text through speech enhancement. The mask-based neural beamformer works as a speech enhancement part and the attention-based encoder-decoder works as a ASR part, where feature extraction function (i.e., extracting filterbank coefficients from enhanced STFT coefficients) connects those components.

$\mathrm{Encoder}(\cdot)$, $\mathrm{Attention}(\cdot)$, and $\mathrm{Decoder}(\cdot)$ are respectively defined in Eqs. (2), (3), and (4). They receive the sequence of the enhanced log Mel filterbank-like features $\hat{O}$ as input and generate a sequence of predicted label symbols Y.

Thus, we can build a multichannel end-to-end ASR system that converts multichannel speech signal to text through speech enhancement with a single network architecture. Note that because all of the procedures, such as enhancement, feature extraction, encoder, attention, and decoder, are connected with differentiable graphs, we can optimize the overall inference for the entire end-to-end ASR objective, which is generating a correct label sequence.

B. Training objective

Learning the attention mechanism of encoder-decoder networks in a noisy environment is difficult because the time-alignments are easily corrupted by noise [7]. To suppress such training difficulty in noisy environments, we adopt a joint CTC-attention loss [7] for our end-to-end ASR objective, where connectionist temporal classification (CTC) [9] is another type of end-to-end framework.

Joint CTC-attention loss resembles a multi-task learning approach. In addition to the loss used for encoder-decoder networks, it also utilizes the loss used for CTC. Because CTC loss imposes a left-to-right constraint on the time-alignment, it helps the encoder network and the attention mechanism learn appropriate time-alignments in the presence of strong background noise.

To define the joint CTC-attention loss, a CTC decoder is added to the encoder's top layer, where the encoder is shared by the attention-based and CTC decoders. Then joint CTC-attention loss $\mathcal{L}$ is formalized as follows:

$$\mathcal{L} = \gamma * (-\log P^{*}_{\mathrm{ATT}}(Y|X)) + (1-\gamma) * (-\log P_{\mathrm{CTC}}(Y|X)), \tag{24}$$

where $P^{*}_{\mathrm{ATT}}(Y|X)$ is the posteriors estimated by the attention-based encoder-decoder, $P_{\mathrm{CTC}}(Y|X)$ is the posteriors estimated

TABLE I
CORPUS INFORMATION.

CHiME-4	Hours	Speakers
Training	3 (real) + 15 (simu)	4 (real) + 83 (simu)
Development	2.9 (real) + 2.9 (simu)	4 (real) + 4 (simu)
Evaluation	2.2 (real) + 2.2 (simu)	4 (real) + 4 (simu)
AMI	Hours	Speakers
Training	78 (real)	135 (real)
Development	9 (real)	18 (real)
Evaluation	9 (real)	16 (real)

TABLE II
CONDITIONS RELATED TO FEATURE EXTRACTION.

Input for encoder-decoder	Log Mel filterbank (40-dim)
Input for neural beamformer	STFT + DC offset (257-dim)
Sampling frequency	16 kHz
Frame length	25 ms
Frame shift	10 ms
Window function	Hamming

by the CTC, and $\gamma \in [0, 1]$ is an interpolation weight. For the encoder-decoder network, approximated posteriors $P^*_{\text{ATT}}(Y|X)$ are used for the training objective instead of true posteriors $P(Y|X)$ in Eq. (1) as follows:

$$P^*_{\text{ATT}}(Y|X) = \prod_n P(y_n|O, y^*_{1:n-1}), \tag{25}$$

where $y^*_{1:n-1}$ is the ground truth of the label subsequence until output time step $n-1$.

V. RELATION TO PREVIOUS WORKS

Several related studies exist on neural beamformers based on filter estimation [14]–[16] and mask estimation [17]–[24]. The main difference is that these previous studies used a component-level training objective within conventional hybrid frameworks, while our work focuses on the entire end-to-end ASR objective. For example, some previous work [17], [19], [20], [23] used a signal-level objective (binary mask classification or regression) to train a network given parallel clean and noisy speech data. On the other hand, other works [14]–[16], [18], [21], [22] used ASR objectives (HMM state classification or sequence-discriminative training), but they remain based on the hybrid approach. Speech recognition with raw multichannel waveforms [32], [33] is also classified into neural beamformers, where the filter coefficients are represented as the network parameters of convolutional neural networks (CNNs), but again these methods are still based on the hybrid approach. Note that the above learning based beamforming approaches can be viewed as an extension of likelihood-maximizing (LIMA) beamformer [34], where beamforming filter coefficients are optimized with HMM/GMM acoustic models based on a maximum likelihood criterion.

If we focus on the network architecture design of the beamforming part aside from the end-to-end framework, our beamformer is based on an MVDR formalization given a reference microphone, which was also previously used in [18], [19]. The difference of our beamformer from those approaches is that it can automatically select a reference microphone within a neural network framework. In [18], [19], one channel is fixedly used as a reference microphone for all utterances by considering microphone geometries. However, our method introduces attention-based reference microphone selection, which allows the beamformer to choose appropriate reference microphones automatically in terms of the entire end-to-end ASR objective without any prior information of microphone geometries.

Similarly, if we only focus on the above automatic reference selection function aside from our entire framework, there exist prior studies [35], [36], which have a function to select dominant channels for a multichannel ASR. [35] uses an attention mechanism to perform channel selection from the pool of multichannel feature candidates in the filterbank domain, while [36] hardly selects dominant features with a max-pooling layer in the hidden state domain. These approaches mainly differ from ours in a sense that they do not hold the beamforming function. This is because they perform their enhancement process in the filterbank or hidden state domain rather than in the STFT domain, and cannot perform beamforming in principle due to the lack of spatial information.

Regarding end-to-end speech recognition, all existing studies are based on a single channel setup. For example, most focus on a standard clean ASR setup without speech enhancement [2]–[4], [6]–[10]. Several research discussed end-to-end ASR in a noisy environment [5], [11], but these method deals with noise robustness by preparing various types of simulated noisy speech for training data without incorporating multichannel speech enhancement in their end-to-end frameworks.

VI. EXPERIMENTAL CONDITIONS

A. Data corpora and feature representation

We compared the effectiveness of our multichannel end-to-end system to a baseline end-to-end system with noisy or beamformed speech signals. Even though we had two multichannel ASR benchmarks, CHiME-4 [37] and AMI [38], we mainly used the CHiME-4 corpus to demonstrate our experiments.

CHiME-4, an ASR task for public noisy environments, consists of speech recorded using a tablet device with 6-channel microphones in four environments: cafe (CAF), street junction (STR), public transportation (BUS), and pedestrian area (PED). It contains real and simulated data. From the 6-channel microphones, we excluded the second channel signals, which were captured by a microphone under the tablet, and used the rest five channels for the following multichannel experiments ($C = 5$).

AMI, an ASR task for meetings, consists of speech recorded using 8-channel circular microphones ($C = 8$). It contains only real data. The amount of training data for AMI is larger than one for CHiME-4.

ChiME-4 consisted of read speech spoken by native English speakers. while AMI consisted of highly spontaneous speech

spoken by mostly non-native English speakers. Such basic information of the above corpora as the number of hours and speakers is summarized in Table I.

We used 40-dimensional log Mel filterbank coefficients as an input feature vector for the encoder-decoder network ($D_{\mathrm{O}} = 40$) and 257-dimensional STFT-based features (256 STFT coefficients and 1 DC offset) as an input feature vector for the neural beamformer ($F = 257$). Conditions related to feature extraction are briefly summarized in Table II.

B. Evaluated systems

We compared the following seven ASR systems: 1) NOISY, 2) BEAMFORMIT, 3) FILTER_NET, 4) MASK_NET (FIX), 5) MASK_NET (ATT), 6) ERDOGAN's_MVDR, and 7) HEYMANN's_GEV.

NOISY and BEAMFORMIT are the baseline single-channel end-to-end systems that did not include the speech enhancement part in the training phase of their frameworks. Their end-to-end networks were trained only with noisy speech data by following a conventional multi-condition training strategy [37]. During decoding, NOISY used single-channel noisy speech data from 'isolated 1ch track' in CHiME-4 as input, while BEAMFORMIT used the enhanced speech data obtained from the 5-th channel signals with BeamformIt [39], which is a well-known weighted delay-and-sum beamformer, as input.

FILTER_NET, MASK_NET (FIX), and MASK_NET (ATT) are the multichannel end-to-end systems described in Section IV. To evaluate the validity of the reference channel selection, we prepared MASK_NET (ATT) based on a mask-based beamformer with an attention-based reference selection described in Section III-C4, and MASK_NET (FIX) with the 5-th channel as a fixed reference microphone, located on the front in the middle of the tablet device. During training, we adopted a multi-condition training strategy; in addition to optimization with the enhanced features through the neural beamformers, we also used the noisy multichannel speech data as input of the encoder-decoder networks without passing through the neural beamformers to improve the robustness of the encoder-decoder networks.

In addition to the comparison with a conventional delay-and-sum beamformer (BEAMFORMIT), we compared our approach with other state-of-the-art neural beamformer implementations [19], [40], which achieved great ASR performances for conventional hybrid frameworks in the recent CHiME-4 challenge. ERDOGAN's_MVDR and HEYMANN's_GEV also used the same baseline system as well as NOISY and BEAMFORMIT. During decoding, the enhanced speech data produced by the state-of-the-art neural beamformers are used as input to the baseline system.

ERDOGAN's_MVDR adopted the MVDR formalization, similar to our approach, but it always used the 5-th channel as the reference microphone. Therefore, it closely resembles our MASK_NET (FIX). The main difference between them is the training objective. ERDOGAN's_MVDR are separately optimized based on the signal-level objective independent of the ASR component using parallel clean and noisy speech data. On the other hand, MASK_NET (FIX) is jointly optimized based on the end-to-end ASR objective with the ASR component only using noisy speech data. In addition, the structure of mask estimation network is also different from our setting [19].

Different from our approach, HEYMANN's_GEV adopted GEV formalization, which requires the estimation of a steering vector based on eigenvalue decomposition instead of estimating the reference microphone vector. In recent studies on neural beamformers, such a GEV-based neural beamformer is a popular alternative to the MVDR-based neural beamformer. To obtain the enhanced signals, we utilized the software tools provided in the GitHub repository (https://github.com/fgnt/nn-gev) [17].

Table III briefly summarizes the main differences among each evaluated system. "Training objective" indicates that the beamformer was trained based on the ASR-level or the signal-level objective, and "Joint optimization" indicates whether the beamformer was jointly optimized with the end-to-end ASR back-end. "Use neural network" indicates whether the beamformer used the neural network-based architecture, and "Use parallel speech" indicates whether clean speech was used to train the beamformer.

Note that all the evaluated systems used the same network structure, which is described in Section VI-C. In addition, the hyperparameters for the training and decoding conditions, which are described in Section VI-D, were set based on the development accuracy of the NOISY system and shared among all the evaluated systems.

TABLE III
SUMMARY OF EVALUATED SYSTEMS: FILTER_NET AND MASK_NET CORRESPOND TO PROPOSED METHOD.

System	Training objective	Joint optimization	Use neural network	Use parallel speech
BEAMFORMIT [39]	signal-level	No	No	No
FILTER_NET [3]	ASR-level	Yes	Yes	No
MASK_NET [3]	ASR-level	Yes	Yes	No
ERDOGAN's_MVDR [19]	signal-level	No	Yes	Yes
HEYMANN's_GEV [40]	signal-level	No	Yes	Yes

C. Network configurations

1) Encoder-decoder networks: In this experiment, we used a 4-layer BLSTM with 320 cells in the encoder ($D_{\mathrm{H}} = 320$) and a 1-layer LSTM with 320 cells in the decoder. In the encoder, we subsampled the hidden states of the first and second layers and used every second hidden state for the subsequent layer's inputs. Therefore, the number of hidden states at the encoder's output layer was reduced to $L = T/4$. After every BLSTM layer, we used a linear projection layer with 320 units to combine the forward and backward LSTM outputs. For the attention mechanism of the time-alignment,

[3]FILTER_NET and MASK_NET basically follow the formalization in [16] and [19]. However, based on our multichannel end-to-end ASR concept, they are jointly optimized with the end-to-end ASR back-end based on the ASR-level objective.

TABLE IV
NETWORK CONFIGURATIONS.

Model	Layer	Units	Type	Activation
Encoder	L1 - L4	320	BLSTM + Projection	tanh
Decoder	L1	320	LSTM	tanh
	L2	48	Linear	softmax
Filter_net	L1 - L3	320	BLSTM + Projection	tanh
	L4	2570	Linear	tanh
Mask_net	L1 - L3	320	BLSTM + Projection	tanh
	L4	514	Linear	sigmoid

TABLE V
CONDITIONS RELATED TO TRAINING AND DECODING.

Parameter initialization	Uniform distribution ([-0.1, 0.1])
Optimization technique	AdaDelta + gradient clipping
Training objective	Joint CTC-attention loss ($\gamma = 0.9$)
Training epoch	15
Beam size	20
Length penalty	0.3
Allowed hypothesis length	$0.3 \times L \sim 0.75 \times L$ (CHiME-4)

we adopted a location-based attention mechanism, where 10 centered convolution filters of width 100 were used to extract the location-based features. We set the attention inner product dimension to 320 and the sharpening factor to 2.

2) Neural beamformers: We used a similar 3-layer BLSTM with 320 cells ($D_{\mathrm{Z}} = 320$) without the subsampling technique. After every BLSTM layer, we also used a linear projection layer with 320 units. For the attention mechanism of the reference selection, we used the same attention inner product dimension ($D_{\mathrm{V}} = 320$) and sharpening factor ($\beta = 2$) as those of the encoder-decoder network.

Network configurations, except the attention mechanisms, are briefly summarized in Table IV. All of the above networks were implemented using Chainer [41].

D. Training and decoding

In the training stage, all the parameters were initialized with range [-0.1, 0.1] of a uniform distribution. We used the AdaDelta algorithm [42] with gradient clipping [43] for optimization and initialized AdaDelta hyperparameters $\rho = 0.95$ and $\epsilon = 1^{-8}$. Once the loss over the validation set was degraded, we decreased AdaDelta hyperparameter ϵ by multiplying it by 0.01 at each subsequent epoch. To boost the optimization in a noisy environment, we adopted a joint CTC-attention multi-task loss function [7], as described in Section IV-B. We set the interpolation weight to 0.9 ($\gamma = 0.9$). The training procedure was stopped after 15 epochs.

For decoding, we used a beam search algorithm [44] with a beam size of 20 at each output step to reduce the computation cost. CTC scores were also used to re-score the hypotheses. We adopted a length penalty term [3] to the decoding objective and set the penalty weight to 0.3. In the CHiME-4 experiments, we only allowed hypotheses whose lengths were within $0.3 \times L$ and $0.75 \times L$ during the decoding, while the hypothesis lengths in the AMI experiments were automatically determined based on the above scores. Note that we pursued a pure end-to-end setup without external lexicon or language models and used CER as an evaluation metric.

The conditions related to training and decoding are briefly summarized in Table V.

TABLE VI
CHARACTER ERROR RATE [%] FOR CHIME-4 CORPUS.

Model	dev simu	dev real	eval simu	eval real
NOISY	25.0	24.5	34.7	35.8
BEAMFORMIT	21.5	19.3	31.2	28.2
FILTER_NET	19.1	20.3	28.2	32.7
MASK_NET (FIX)	15.5	18.6	23.7	28.8
MASK_NET (ATT)	**15.3**	**18.2**	**23.7**	**26.8**
ERDOGAN's_MVDR [19]	16.2	**18.2**	24.3	**26.7**

VII. EXPERIMENTAL RESULTS

A. Comparison of character error rate

1) CHiME-4: Table VI shows the recognition performance of CHiME-4 with six systems. The result shows that BEAMFORMIT, FILTER_NET, MASK_NET (FIX), and MASK_NET (ATT) outperformed NOISY, confirming the effectiveness of combining speech enhancement with the attention-based encoder-decoder framework. The comparison of MASK_NET (FIX) and MASK_NET (ATT) validates the using of the attention mechanism for reference channel selection. FILTER_NET also improved the performance more than NOISY, but not as much as MASK_NET (ATT). This is because optimizing the filter estimation network is difficult due to a lack of restrictions to estimate filter coefficients, and it needs optimization, as suggested by a previous work [14]. Finally, MASK_NET (ATT) achieved better recognition performance than BEAMFORMIT, proving the effectiveness of our unified architecture rather than a pipe-line combination of speech enhancement and (end-to-end) speech recognition.

Table VI also shows that the performance of MASK_NET (ATT) is comparable to ERDOGAN's_MVDR, which is a state-of-the-art neural beamformer implementation. Note that MASK_NET (ATT) successfully achieved a good performance without requiring parallel clean and noisy speech data. This result suggests that we can eliminate the requirement of parallel speech data for training by the end-to-end optimization of the ASR system.

We also evaluated HEYMANN's_GEV, but the performance is quite poor[4]. We assume that this result was caused by the speech distortions produced by the GEV-based beamformer. Although the MVDR-based beamformer suppressed the speech distortions, the GEV-based beamformer ignored the speech distortions and only focused on the noise reduction. Such speech distortions sometimes degrade ASR performance, when we input the beamformed signals to existing ASR systems.

[4]For example, CER for eval_real is 71.6.

TABLE VII
CHARACTER ERROR RATE [%] FOR AMI CORPUS.

Model	dev	eval
NOISY	41.8	45.3
BEAMFORMIT	44.9	51.3
MASK_NET (ATT)	**35.7**	**39.0**

TABLE VIII
CHIME-4 VALIDATION ACCURACIES [%] FOR MASK_NET (ATT) WITH DIFFERENT NUMBERS AND ORDERS OF CHANNELS.

Model	channel	dev
NOISY	isolated_1ch_track	87.9
MASK_NET (ATT)	5_6_4_3_1	91.2
MASK_NET (ATT)	3_4_1_5_6	91.2
MASK_NET (ATT)	5_6_4_1	91.1
MASK_NET (ATT)	6_4_3_1	90.4
MASK_NET (ATT)	5_6_4	90.9
MASK_NET (ATT)	6_4_1	90.1

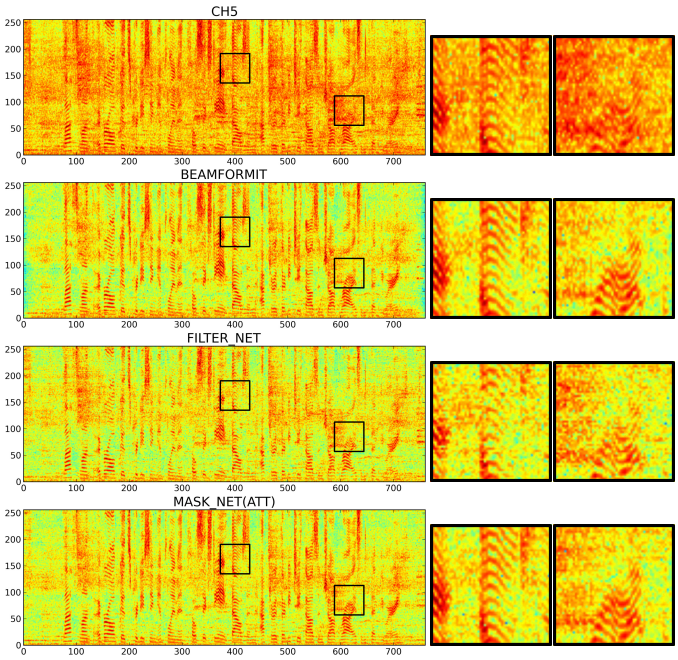


Fig. 4. Comparison of log-magnitude spectrograms of a CHiME-4 utterance with the 5-th channel noisy signal, enhanced signal with BeamformIt, and enhanced signal with our proposed FILTER_NET and MASK_NET (ATT)

2) AMI: To further investigate the effectiveness of our proposed multichannel end-to-end framework, we also experimented with the AMI corpus. Table VII compares the recognition performance of three systems: NOISY, BEAMFORMIT, and MASK_NET (ATT). In NOISY, we used noisy speech data from the 1st channel in AMI as input to the system. Table VII shows that, even in the AMI, our proposed MASK_NET (ATT) achieved better recognition performance than the baseline systems (NOISY and BEAMFORMIT), confirming the effectiveness of our proposed multichannel end-to-end framework. BEAMFORMIT was worse than NOISY even with the enhanced signals. This phenomenon is sometimes observed in noisy speech recognition where the distortion caused by the sole speech enhancement degrades the performance without re-training. Since our end-to-end system jointly optimized the speech enhancement part with the ASR objective, it can avoid such degradations.

B. Influence on number and order of channels

As we discussed in Section III-C, one unique characteristic of our proposed MASK_NET (ATT) is its robustness/invariance against the number and order of channels without re-training. Table VIII shows the influence of the CHiME-4 validation accuracies on the number and order of the channels. The validation accuracy was computed conditioned on ground truth labels $y^*_{1:n-1}$ in Eq. (4) during the decoder's recursive label prediction, which has a strong correlation with CER. The second column of the table represents the channel indices, which were used as input of the same MASK_NET (ATT) network.

Comparison of 5_6_4_3_1 and 3_4_1_5_6 shows that the order of the channels did not affect the recognition performance of MASK_NET (ATT) at all, as we expected. In addition, even when we used fewer than three or four channels as input, MASK_NET (ATT) still outperformed NOISY (single channel). These results confirm that our proposed multichannel end-to-end system can deal with input signals with an arbitrary number and order of channels without any re-configuration and re-training.

In addition to the above analyses, comparing the setups using the same number of channels, 5_6_4_1 and 5_6_4 outperformed 6_4_3_1 and 6_4_1, respectively. The observation is due to the fact that the 5-th channel is the single best channel in the real dev and eval sets of CHiME-4 task.

C. Visualization of beamformed features

To analyze the behavior of our developed speech enhancement component with a neural beamformer, Fig. 4 visualizes the spectrograms of the same CHiME-4 utterance for four signals: 1) the 5-th channel noisy signal, 2) an enhanced signal with BEAMFORMIT, 3) an enhanced signal with FILTER_NET, and 4) an enhanced signal with MASK_NET (ATT). We confirmed that BEAMFORMIT, FILTER_NET, and MASK_NET (ATT) successfully suppressed noise compared to the 5-th channel signal by eliminating the blurred red areas overall. In addition, by focusing on the second black boxes, the harmonic structure, which was corrupted in the 5-th channel signal, was recovered in BEAMFORMIT, FILTER_NET, and MASK_NET (ATT).

This result suggests that our proposed MASK_NET (ATT) successfully learned a noise suppression function that resembles the conventional beamformer, although it is optimized based on the end-to-end ASR objective, without explicitly using clean data as a target.

Similar to BEAMFORMIT and MASK_NET (ATT), FILTER_NET seems to learn some kind of noise suppression

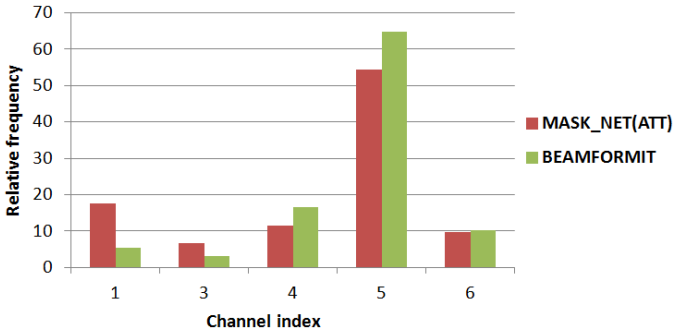


Fig. 5. Histogram of reference microphone selected by BEAMFORMIT and MASK_NET (ATT)

function. However, it seems to over-suppress the overall signals compared to BEAMFORMIT and MASK_NET (ATT) (e.g., inside of the left black boxes), which may cause speech distortion.

D. Histogram of selected reference microphone

To analyze the behavior of our proposed attention mechanism for reference microphone selection, Fig. 5 illustrates a histogram of the selected reference microphone for the development set with two systems: BEAMFORMIT and MASK_NET (ATT). As described in Eqs. (18) and (19), our proposed reference selection mechanism is formalized in a probabilistic way, but in this figure, the frequency is counted assuming that the channel index with the highest probability is selected. BEAMFORMIT selected a reference microphone using a metric based on the signal-level criterion, i.e., pairwise cross-correlation in time domain [39].

Fig. 5 shows that both BEAMFORMIT and MASK_NET (ATT) selected the 5-th channel most frequently. That result seems plausible from the viewpoint of microphone geometries, because the 5-th channel is located on the front and the center of the tablet device, and therefore, it is expected to capture relatively clean speech signals. Our preliminary result also shows that the 5-th channel is the single best performing channel in the array. One interesting finding is that the trends in the selected reference seem similar, although MASK_NET (ATT) only learned the reference selection mechanism to improve the end-to-end ASR objective.

VIII. Conclusion

To handle the challenging noisy ASR tasks, we extended an existing attention-based encoder-decoder framework by integrating a neural beamformer and proposed a unified architecture of a multichannel end-to-end ASR. This architecture allows the overall inference in multichannel speech recognition (i.e., from speech enhancement to speech recognition) to be optimized based on the end-to-end ASR objective, and leads to an end-to-end framework that works well in the presence of strong background noise. In addition, because it is formalized independent of microphone geometries, it can deal with input signals with an arbitrary number and order of channels without any re-configuration and re-training. Our experimental results on challenging noisy ASR benchmarks (CHiME-4 and AMI) show that the proposed framework outperformed the end-to-end baseline with noisy and delay-and-sum beamformed inputs. In addition, visualization of beamformed features shows that our neural beamformer successfully learned a noise suppression function, although it is optimized based on the end-to-end ASR objective, without using parallel clean and noisy speech data.

The current system suffers from data sparseness issues due to the lack of lexicon and language models in the back-end decoder-network processing, unlike the conventional hybrid approach. The results reported in this paper convincingly show the effectiveness of the proposed framework for the front-end processing, they still have a room for the improvement to reach the state-of-the-art performance by solving the above back-end processing issues. Our most important future work is to overcome these data sparseness issues in the back-end processing by developing adaptation techniques of an end-to-end framework by incorporating linguistic resources.

Appendix A
Location-based attention mechanism for time alignment

This section describes the formalization of our adopted location-based attention mechanism [3] represented below:

$$\mathbf{c}_n = \mathrm{Attention}(\mathbf{a}_{n-1}, \mathbf{s}_{n-1}, H).$$

This corresponds to $\mathrm{Attention}(\cdot)$ in Eq. (3). Based on the attention mechanism, the attention weight vector $\mathbf{a}_n$ and the context vector $\mathbf{c}_n$ are estimated as follows:

$$\{\mathbf{f}_{n,l}\}_{l=1}^{L} = \mathbf{F} * \mathbf{a}_{n-1}, \tag{26}$$

$$k_{n,l} = \mathbf{w}^{\mathrm{T}} \tanh(\mathbf{V}^{\mathrm{S}}\mathbf{s}_n + \mathbf{V}^{\mathrm{H}}\mathbf{h}_l + \mathbf{V}^{\mathrm{F}}\mathbf{f}_{n,l} + \mathbf{b}), \tag{27}$$

$$a_{n,l} = \frac{\exp(\alpha k_{n,l})}{\sum_{l=1}^{L} \exp(\alpha k_{n,l})}, \tag{28}$$

$$\mathbf{c}_n = \sum_{l=1}^{L} a_{n,l}\mathbf{h}_l, \tag{29}$$

where $\mathbf{w} \in \mathbb{R}^{1\times D_{\mathrm{W}}}$, $\mathbf{V}^{\mathrm{H}} \in \mathbb{R}^{D_{\mathrm{W}}\times D_{\mathrm{H}}}$, $\mathbf{V}^{\mathrm{S}} \in \mathbb{R}^{D_{\mathrm{W}}\times D_{\mathrm{S}}}$, and $\mathbf{V}^{\mathrm{F}} \in \mathbb{R}^{D_{\mathrm{W}}\times D_{\mathrm{F}}}$ are trainable weight matrices. $\mathbf{b} \in \mathbb{R}^{D_{\mathrm{W}}}$ is a trainable bias vector. $\mathbf{F} \in \mathbb{R}^{D_{\mathrm{F}}\times D_{\mathrm{f}}}$ is a trainable convolution filter. α is a sharpening factor, and $*$ represents the convolution operation. Eqs. (26)-(29) correspond to $\mathrm{Attention}(\cdot)$ in Eq. (3).

The convolution operation is performed with a stride of 1 along the time axis, and the filter $\mathbf{F}$ produce D_{F}-dimensional feature vector $\mathbf{f}_{n,l}$ at each time step l, where we adopt the zero-padding technique for the edge region.

Appendix B
Real-valued computation for complex-valued inverse operation

To implement the mask-based MVDR neural beamformer, we have to deal with complex-valued operations. However, to the best of our knowledge, most deep learning libraries do not support such complex-valued operations. To bypass this issue,

we implement such complex-valued operations using real-valued operations by separately computing real and imaginary parts.

In this section, we describe the mathematical formula for a complex-valued inverse, which is the most complicated operation in our implementation. Let $\mathbf{C} = \mathbf{A} + i\mathbf{B}$ be a complex-valued matrix, where $\mathbf{A}$ and $\mathbf{B}$ are real-valued matrices corresponding to real and imaginary parts. The inverse of $\mathbf{C}$ can be computed separately for the real and imaginary parts as follows [45]:

$$\Re(\mathbf{C}^{-1}) = (\mathbf{A} + \mathbf{B}\mathbf{A}^{-1}\mathbf{B})^{-1} \tag{30}$$

$$\Im(\mathbf{C}^{-1}) = (\mathbf{A} + \mathbf{B}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}\mathbf{A}^{-1}. \tag{31}$$

This formula allows us to implement the complex-valued inverse only with a combination of the real-valued operations.

Because other fundamental operations such as addition, multiplication, and division can also be computed separately, we can easily implement the neural beamformer (including complex-valued operations) with the existing sophisticated deep learning libraries.

APPENDIX C
ADDITIONAL EXPERIMENT: COMPARISON OF PERFORMANCES BETWEEN THE END-TO-END ASR AND HMM/DNN HYBRID SYSTEMS

In this section, we report an additional experimental result, which compares the performance between the end-to-end ASR and HMM/DNN hybrid systems in a noisy ASR task. Through the experiment, we investigate the promising future research direction to further improve the performance of the multichannel end-to-end ASR framework.

A. Condition

Our experimental conditions basically follow Section VI. Here we focus on the conditions related to the HMM/DNN hybrid system.

For the evaluations, we utilized the official baseline HMM/DNN hybrid system that was included in the ChiME-4 corpus. The system was optimized using sequence-discriminative training with 5-th channel noisy speech data and applied the language model re-scoring technique. Detailed descriptions of it are shown in a Kaldi recipe [5].

B. Result

Table IX shows CERs for the four systems, where the first and second rows correspond to NOISY and MASK_NET (ATT) in Table VI, and the third row (hybrid + BEAMFORMIT) corresponds to CHiME-4's official baseline setup. "Recognizer" denotes the type of back-end ASR recognizer: one for the end-to-end ASR framework (i.e., the attention-based encoder-decoder network) and another for the HMM/DNN hybrid framework. "Input signal" denotes the type of signal input to the framework: one for the enhanced signal with BEAMFORMIT and another for the enhanced signal with MASK_NET (ATT). Note that although the fourth row used the HMM/DNN hybrid system as the back-end ASR recognizer, the input signal (i.e., MASK_NET (ATT)) was produced by the mask-based neural beamformer developed within our multichannel end-to-end ASR framework.

TABLE IX
COMPARISON OF CHARACTER ERROR RATES [%] FOR CHIME-4 CORPUS BETWEEN END-TO-END ASR AND HMM/DNN HYBRID SYSTEMS.

Recognizer	Input signal	Dev-simu	Dev-real	Eval-simu	Eval-real
End-to-end	BEAMFORMIT	21.5	19.3	31.2	28.2
End-to-end	MASK_NET (ATT)	**15.3**	**18.2**	**23.7**	**26.8**
Hybrid	BEAMFORMIT	3.8	3.1	6.7	7.1
Hybrid	MASK_NET (ATT)	**2.7**	**2.9**	**3.9**	**6.0**

The results show that a performance gap exists between end-to-end ASR and HMM/DNN hybrid systems in noisy ASR task. On the other hand, a comparison with the third (hybrid + BEAMFORMIT) and fourth (hybrid + MASK_NET (ATT)) rows shows that the input signal enhanced by the multichannel end-to-end ASR framework (MASK_NET (ATT)) achieved lower CER values than the signal enhanced by BeamformIt (BEAMFORMIT). These results suggest that the neural beamformer, jointly optimized within our multichannel end-to-end framework, produced more suitable enhanced speech inputs at least for the HMM/DNN hybrid system than CHiME-4's official baseline beamformer, BeamformIt. In other words, our developed multichannel end-to-end ASR framework has probably already achieved reasonable beamformers, even though it was optimized under the end-to-end ASR-oriented criterion.

Based on the above finding, we obtained insight into future research directions to further improve the performance of an end-to-end ASR system in noisy ASR task. To boost the discriminative power of a total multichannel end-to-end ASR system, we need to investigate how to improve such end-to-end ASR back-end as attention-based encoder-decoder networks, especially in noisy ASR tasks.

Several possible reasons might explain the performance gap between end-to-end ASR and HMM/DNN hybrid systems. The main reason is probably the existence of external lexicon or language models. In this paper, to pursue a pure end-to-end setup[6], we did not utilize such external lexicon or language models. On the other hand, the HMM/DNN hybrid system utilized such lexicon and language models, which provide effective language regularity. Amount of the training data in CHiME-4 (i.e., 18 hours) is probably not sufficient for the end-to-end framework to obtain such language regularity.

In the previous studies (e.g., [4], [47]), the external language model is applied to the decoding procedure of the end-to-end ASR systems and shown to be effective to improve the ASR performance in the case when the amount of training data is not sufficient to learn the language regularity. Because the amount of training data in CHiME-4 corpus is relatively small, even in our experimental setup, it would be helpful to

[5]https://github.com/kaldi-asr/kaldi/tree/master/egs/chime4/s5_6ch Kaldi is a popular open-source toolkit for conducting ASR experiments [46].

[6]Strictly speaking, the use of such external lexicon or language models do not follow our end-to-end definition because it was separately optimized with the end-to-end ASR framework using another training corpus including a large amount of text data. That is why we focused on a pure end-to-end setup without external lexicon or language models in our experimental setup.

compensate the insufficiency of the language regularity of the end-to-end systems.

Another possible reason is the immaturity of the adopted end-to-end ASR architecture. In this paper, we adopted the standard architecture of the attention-based encoder-decoder networks [4]. Adopting more sophisticated architectures, which are being studied (e.g., [6]), will probably improve the performance of the end-to-end ASR back-end.

APPENDIX D
NOTATION LIST

This section provides a brief reference, which describes the notations used in this paper. Basically, each notation is described once in the tables.

Basic indices	
t	input time step
n	output time step
l	subsampled time step
f	frequency index
c	channel index
T	length of input sequence
N	length of output sequence
L	length of subsampled input sequence
F	dimension of STFT signals
C	number of channels

Encoder-Decoder (Section II and Appendix A)	
$\mathbf{o}_t$	acoustic feature vector at t
O	sequence of input acoustic feature
y_n	label symbol at n
Y	sequence of output label symbol
$\mathcal{V}$	set of label symbols
$P(Y\|X)$	posteriors predicted by encoder-decoder
$\mathbf{h}_l$	state vector of encoder's top layer at l
H	sequence of encoder's output
$\mathbf{c}_n$	context vector at n
$\mathbf{a}_n$	attention weight at n
$\mathbf{s}_n$	state vector of decoder's top layer at n
D_{O}	dimension of acoustic feature vector
D_{H}	dimension of encoder's state vector
$\mathbf{f}_{n,l}$	location-based feature at (n,l)
$\mathbf{F}$	convolution filter for attention mechanism
$\mathbf{w}$	weight vector for attention inner product
$\mathbf{b}$	bias vector for attention mechanism
$\mathbf{V}^{\mathrm{S}}$	weight matrix for decoder's state $\mathbf{s}_n$
$\mathbf{V}^{\mathrm{H}}$	weight matrices for encoder's state $\mathbf{h}_l$
$\mathbf{V}^{\mathrm{F}}$	weight matrix for location-based feature $\mathbf{f}_{n,l}$
α	sharping factor
D_{W}	dimension of attention inner product
D_{S}	dimension of decoder's state vector
D_{F}	number of filters for attention mechanism
D_{f}	filter width for attention mechanism

Neural beamformer (Section III)	
$\hat{x}_{t,f}$	enhanced STFT coefficient at (t,f)
$x_{t,f,c}$	STFT coefficient at (t,f,c)
$\mathbf{x}_{t,f}$	vector of STFT coefficient at (t,f)
$g_{t,f,c}$	beamforming filter coefficient at (t,f,c)
$\mathbf{g}_{t,f}$	vector of time-variant filter coefficient at (t,f)
$\mathbf{g}_f$	vector of time-invariant filter coefficient at f
X_c	sequence of input STFT feature for c

Filter estimation network (Section III-B)	
$\mathbf{z}_t$	output vector of BLSTM network at t
Z	sequence of output vector of BLSTM network
$\bar{\mathbf{x}}_t$	input vector of BLSTM network at t
$\mathbf{W}_f^{\Re}$	weight matrix to output real part of filters at f
$\mathbf{b}_f^{\Re}$	bias vector to output real part of filters at f
$\mathbf{W}_f^{\Im}$	weight matrix to output imaginary part of filters at f
$\mathbf{b}_f^{\Im}$	bias vector to output imaginary part of filters at f
D_{Z}	dimension of BLTSM network's output

Mask estimation network (Section III-C)	
$\mathbf{\Phi}_f^{\mathrm{S}}$	PSD matrix for speech at f
$\mathbf{\Phi}_f^{\mathrm{N}}$	PSD matrix for noise at f
$\mathbf{u}$	reference microphone vector
$m_{t,f}^{\mathrm{S}}$	mean masks for speech at (t,f)
$m_{t,f}^{\mathrm{N}}$	mean masks for noise at (t,f)
$\mathbf{m}_t^{\mathrm{S}}$	vector of mean mask for speech at t
$\mathbf{m}_t^{\mathrm{S}}$	vector of mean mask for noise at t
$m_{t,f,c}^{\mathrm{S}}$	time-frequency masks for speech at (t,f,c)
$m_{t,f,c}^{\mathrm{N}}$	time-frequency masks for noise at (t,f,c)
$\mathbf{m}_{t,c}^{\mathrm{S}}$	vector of time-frequency mask for speech at (t,c)
$\mathbf{m}_{t,c}^{\mathrm{N}}$	vector of time-frequency mask for noise at (t,c)
$\bar{\mathbf{x}}_{t,c}$	input vector of BLSTM network at (t,c)
$\bar{X}_c$	sequence of input STFT feature $\bar{\mathbf{x}}_{t,c}$ for c
$\mathbf{z}_{t,c}^{\mathrm{S}}$	output vector of BLSTM network for speech mask at (t,c)
Z_c^{S}	sequence of BLSTM network's output for speech mask
$\mathbf{z}_{t,c}^{\mathrm{N}}$	output vector of BLSTM network for noise mask at (t,c)
Z_c^{N}	sequence of BLSTM network's output for noise mask
$\mathbf{W}^{\mathrm{S}}$	weight matrix to output speech mask
$\mathbf{b}^{\mathrm{S}}$	bias vector to output speech mask
$\mathbf{W}^{\mathrm{N}}$	weight matrix to output noise mask
$\mathbf{b}^{\mathrm{N}}$	bias vector to output noise mask
$\mathbf{q}_c$	time-averaged state feature
$\mathbf{r}_c$	PSD-based spatial feature
$\phi_{f,c,c'}^{S}$	entry in c-th row and c'-th column of PSD matrix for speech $\mathbf{\Phi}_f^{\mathrm{S}}$
$\tilde{\mathbf{w}}$	weight vector for attention inner product
$\tilde{\mathbf{b}}$	bias vector for attention mechanism
$\mathbf{V}^{\mathrm{Q}}$	weight matrix for time-averaged state feature $\mathbf{q}_c$
$\mathbf{V}^{\mathrm{R}}$	weight matrix for PSD-based spatial feature $\mathbf{r}_c$
β	sharping factor
D_{V}	dimension of attention inner product

Multichannel end-to-end ASR (Section IV)	
$\hat{X}$	sequence of enhanced STFT feature
$\hat{O}$	sequence of enhanced acoustic feature
$\mathbf{p}_t$	enhanced power spectrum vector at t
$\hat{\mathbf{o}}_t$	enhanced acoustic feature vector at t
$\mathcal{L}$	joint CTC-attention loss
$P_{\mathrm{ATT}}^{*}(Y\|X)$	approximated posteriors predicted by encoder-decoder
$P_{\mathrm{CTC}}(Y\|X)$	posteriors predicted by CTC
γ	interpolation weight

REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," *arXiv preprint arXiv:1412.1602*, 2014.

[3] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 577–585.

[4] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4945–4949.

[5] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, 2016.

[6] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," *arXiv preprint arXiv:1610.03022*, 2016.

[7] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. ICASSP*, 2017, pp. 4835–4839.

[8] L. Lu, X. Zhang, and S. Renals, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5060–5064.

[9] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning (ICML)*, 2014, pp. 1764–1772.

[10] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 167–174.

[11] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," *International Conference on Machine Learning (ICML)*, pp. 173–182, 2016.

[12] K. Kinoshita, M. Delcroix, S. Gannot, E. A. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj *et al.*, "A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, pp. 1–19, 2016.

[13] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Analysis and outcomes," *Computer Speech & Language*, 2016.

[14] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. R. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5745–5749.

[15] B. Li, T. N. Sainath, R. J. Weiss, K. W. Wilson, and M. Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition," in *Interspeech*, 2016, pp. 1976–1980.

[16] Z. Meng, S. Watanabe, J. R. Hershey, and H. Erdogan, "Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 271–275.

[17] 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)*, 2016, pp. 196–200.

[18] X. Xiao, C. Xu, Z. Zhang, S. Zhao, S. Sun, and S. Watanabe, "A study of learning based beamforming methods for speech recognition," in *CHiME 2016 workshop*, 2016, pp. 26–31.

[19] H. Erdogan, T. Hayashi, J. R. Hershey, T. Hori, C. Hori, W.-N. Hsu, S. Kim, J. Le Roux, Z. Meng, and S. Watanabe, "Multi-channel speech recognition: LSTMs all the way through," in *CHiME 2016 workshop*, 2016.

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

[21] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, "BEAMNET: end-to-end training of a beamformer-supported multi-channel ASR system," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5325–5329.

[22] X. Xiao, S. Zhao, D. 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)*, 2017, pp. 3246–3250.

[23] T. Nakatani, N. Ito, T. Higuchi, S. Araki, and K. Kinoshita, "Integrating DNN-based and spatial clustering-based mask estimation for robust MVDR beamforming," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 286–290, 2017.

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

[25] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[26] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Transactions on audio, speech, and language processing*, vol. 18, no. 2, pp. 260–276, 2010.

[27] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 5, pp. 1529–1539, 2007.

[28] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi *et al.*, "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 436–443.

[29] T. Ochiai, S. Watanabe, T. Hori, and J. R. Hershey, "Multichannel end-to-end speech recognition," in *International Conference on Machine Learing (ICML)*, 2017.

[30] T. N. Sainath, A. Narayanan, R. J. Weiss, E. Variani, K. W. Wilson, M. Bacchiani, and I. Shafran, "Reducing the computational complexity of multimicrophone acoustic models with integrated feature extraction," in *Interspeech*, 2016, pp. 1971–1975.

[31] Z.-Q. Wang and D. Wang, "A joint training framework for robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 796–806, 2016.

[32] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4624–4628.

[33] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Interspeech*, 2015, pp. 1–5.

[34] M. L. Seltzer, B. Raj, and R. M. Stern, "Likelihood-maximizing beamforming for robust hands-free speech recognition," *IEEE Transactions on speech and audio processing*, vol. 12, no. 5, pp. 489–498, 2004.

[35] S. Kim and I. Lane, "Recurrent models for auditory attention in multi-microphone distance speech recognition," in *Interspeech*, 2016, pp. 1–9.

[36] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1120–1124, 2014.

[37] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, 2016.

[38] T. Hain, L. Burget, J. Dines, G. Garau, V. Wan, M. Karafi, J. Vepa, and M. Lincoln, "The AMI system for the transcription of speech in meetings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007, pp. 357–360.

[39] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[40] J. Heymann, L. Drude, and R. Haeb-Umbach, "Wide residual blstm network with discriminative speaker adaptation for robust speech recognition," in *CHiME 2016 workshop*, 2016.

[41] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in NIPS*, 2015.

[42] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[43] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *International Conference on Machine Learning (ICML)*, pp. 1310–1318, 2013.

[44] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems (NIPS)*, 2014, pp. 3104–3112.

[45] K. B. Petersen, M. S. Pedersen *et al.*, "The matrix cookbook," *Technical University of Denmark*, vol. 7, p. 15, 2008.

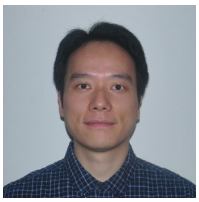
[46] D. Povey, A. Ghoshal, G. Boulianne *et al.*, "The kaldi speech recognition toolkit," in *IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.
[47] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Interspeech*, 2017, pp. 949–953.

Tsubasa Ochiai received B.E. and M.E. degrees in Information Engineering from Doshisha University, Kyotanabe, Japan in 2013 and 2015. He is currently a Ph.D. student at Doshisha University and a Research Fellow of JSPS. His research interests include pattern recognition, deep learning, and speech recognition. He is a member of the Acoustic Society of Japan, IEICE, and IEEE.

Shinji Watanabe is a senior principal research scientist at the Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA. He received a Ph.D. from Waseda University, Tokyo in 2006. From 2001 to 2011, he was a research scientist at NTT Communication Science Laboratories, Kyoto. From January to March in 2009, he was a visiting scholar at the Georgia Institute of Technology, Atlanta, GA. His research interests include Bayesian machine learning and speech and spoken language processing. He has published more than 100 papers and received several awards including a best paper award from the IEICE in 2003. He served as associate editor of the IEEE Transactions on Audio Speech and Language Processing and is a member of several committees including the IEEE Signal Processing Society Speech and Language Technical Committee.

Takaaki Hori received B.E. and M.E. degrees in electrical and information engineering from Yamagata University, Yonezawa, Japan in 1994 and 1996 and a Ph.D. degree in system and information engineering from Yamagata University in 1999. From 1999 to 2015, he was engaged in research on speech recognition and spoken language understanding at Cyber Space Laboratories and Communication Science Laboratories in Nippon Telegraph and Telephone (NTT) Corporation, Japan. He was a visiting scientist at the Massachusetts Institute of Technology (MIT) from 2006 to 2007. Since 2015, he has been a senior principal research scientist at the Mitsubishi Electric Research Laboratories (MERL), Cambridge, Massachusetts, USA. He has written over 90 peer-reviewed papers in speech and language research fields. He received the 24th TELECOM System Technology Award from the Telecommunications Advancement Foundation in 2009, the IPSJ Kiyasu Special Industrial Achievement Award from the Information Processing Society of Japan in 2012, and the 58th Maejima Hisoka Award from Tsushinbunka Association in 2013.

John R. Hershey has been a senior principle research scientist and leader of the speech and audio team at MERL since 2010. Prior to joining MERL, John spent 5 years at IBM's T.J. Watson Research Center in New York, and led the Noise Robust Speech Recognition team. He also spent a year as a visiting researcher in the speech group at Microsoft Research, after receiving his doctorate from UCSD. Over the years he has contributed to more than 100 publications and over 30 patents in the areas of machine perception, speech processing, speech recognition, and natural language understanding.



Xiong Xiao (S'06-M'10) received B. Eng and Ph.D. degrees in computer engineering from Nanyang Technological University (NTU) in 2004 and 2010. He joined Temasek laboratories @ NTU in 2009 and is currently a senior research scientist. His research interests include robust speech processing, machine learning based signal processing, and spoken document retrieval.