

Robust Front-End for Multi-Channel ASR using Flow-Based Density Estimation

Hyeongju Kim*, Hyeonseung Lee, Woo Hyun Kang, Hyung Yong Kim and Nam Soo Kim

Department of Electrical and Computer Engineering and INMC, Seoul National University, South Korea
 {hjkim, hslee, whkang, hykim}@hi.snu.ac.kr, nkim@snu.ac.kr

Abstract

For multi-channel speech recognition, speech enhancement techniques such as denoising or dereverberation are conventionally applied as a front-end processor. Deep learning-based front-ends using such techniques require aligned clean and noisy speech pairs which are generally obtained via data simulation. Recently, several joint optimization techniques have been proposed to train the front-end without parallel data within an end-to-end automatic speech recognition (ASR) scheme. However, the ASR objective is sub-optimal and insufficient for fully training the front-end, which still leaves room for improvement. In this paper, we propose a novel approach which incorporates flow-based density estimation for the robust front-end using non-parallel clean and noisy speech. Experimental results on the CHiME-4 dataset show that the proposed method outperforms the conventional techniques where the front-end is trained only with ASR objective.

1 Introduction

Robust multi-channel speech recognition is a challenging task since the acoustic interferences such as background noise and reverberation degrade the quality of input speech. It is known that an automatic speech recognition (ASR) system, which is trained on clean speech, works poorly in noisy environments due to the mismatch in acoustic characteristics [Gong, 1995]. For robust multi-channel ASR, recent studies usually employ a front-end component that leverages a denoising algorithm such as the Minimum Variance Distortionless Response (MVDR) or a dereverberation algorithm (e.g., the Weighted Prediction Error, WPE) [Barker *et al.*, 2017; Kinoshita *et al.*, 2016]. Even though these denoising and dereverberation methods have brought substantial improvements for an ASR system [Vincent *et al.*, 2013], they are usually designed for enhancing speech in stationary environments.

For handling more realistic acoustic environments, speech enhancement techniques based on deep neural network

(DNN) have been developed, which basically require time-aligned parallel clean and noisy speech data for training [Erdogan *et al.*, 2016]. These techniques usually train the model to optimize signal level criteria such as signal to noise ratio (SNR), independently of speech recognition accuracy. To make the speech enhancement algorithm a more efficient front-end for ASR, recent studies have proposed to optimize the speech enhancement model using the ASR objective within an end-to-end ASR scheme [Heymann *et al.*, 2019; Ochiai *et al.*, 2017; Kinoshita *et al.*, 2017]. This training method allows to use non-parallel clean and noisy data for training the front-end.

However, since the ASR objective only focuses on preserving the phonetic information of the input speech, it is insufficient for fully training the speech enhancement model and guaranteeing generalized performance improvement. Moreover, conventional approaches do not take into account the distribution of the target clean speech signal on which the original ASR system is trained. To overcome these limitations, we propose a novel method that applies flow-based density estimation to the robust front-end using non-parallel clean and noisy speech. In the proposed method, a flow-based density estimator is trained with clean speech and the front-end receives the additional generative loss from the density estimator. In other words, the front-end performs multi-task learning. The auxiliary objective induces the front-end to learn more regularized representations, which in turn improves the performance of the ASR module on the noisy CHiME-4 evaluation set.

Our main contributions are as follows:

- We propose a novel approach that combines density estimation with multi-channel ASR to exploit the probability distribution of speech signal for robust front-end.
- We present a new flow-based model *MelFlow* for estimating the probability distribution of mel-spectrograms.
- We demonstrate that our multi-task learning strategy results in better performance on WERs over noisy speech compared to the scheme that depends only on the ASR objective.

2 Related Work

Denoising and dereverberation methods are originally designed to estimate clean signal and can be applied without

*Contact Author

any training [Wölfel and McDonough, 2009]. These traditional methods, however, require time-consuming iterative process and work well only in stationary environmental conditions. To handle more realistic acoustic environments, modern speech enhancement techniques usually employ DNNs to remove background noise and directly estimate the desired signal [Han *et al.*, 2015].

For multi-channel ASR, denoising and dereverberation techniques have been employed as front-ends and reported to produce some improvements in noisy speech recognition [Wölfel and McDonough, 2009]. However, the direct application of enhancement-based algorithms to ASR has some problems. One of the main problems is that the ASR accuracy is not taken into consideration when training the front-end, thus the resulting features may lack phonetic information. Another critical problem is that training the conventional enhancement modules require parallel dataset (i.e., pairs of aligned clean and noisy speech signals). To alleviate the problems, recent approaches optimize the front-end and ASR models jointly using the ASR objective [Heymann *et al.*, 2019; Ochiai *et al.*, 2017].

3 Baseline

3.1 Neural Beamforming Method

A filter-and-sum beamforming method is a typical denoising technique for enhancing multi-channel signal. In the filter-and-sum beamforming, a speech image at the reference microphone is estimated by using a linear filter operating as follows:

$$y_{t,f} = \sum_{c=1}^C h_{t,f,c} s_{t,f,c}, \quad (1)$$

where $s_{t,f,c} \in \mathbb{C}$ is short-time Fourier transform (STFT) coefficient, $h_{t,f,c} \in \mathbb{C}$ is a beamforming filter coefficient and $y_{t,f} \in \mathbb{C}$ is an estimated speech image. Subscripts t, f, c denote the c -th channel of a signal at a time-frequency bin (t, f) . While conventional methods optimize $h_{t,f,c}$ based on a signal-level objective, recent studies train $h_{t,f,c}$ jointly within an ASR architecture [Meng *et al.*, 2017; Ochiai *et al.*, 2017]. This kind of data-driven approach is called the neural beamforming method and can be classified into two categories: (i) filter estimation approach and (ii) mask estimation approach. The filter estimation approach estimates the time-variant filter coefficients $\{h_{t,f,c}\}_{t=1, f=1, c=1}^{T, F, C}$ directly but suffers from unstable training due to high flexibility [Meng *et al.*, 2017]. On the other hand, the mask estimation approach optimizes time-invariant filter coefficients $\{h_{f,c}\}_{f=1, c=1}^{F, C}$ and has been reported to achieve improved performances in multi-channel speech recognition [Ochiai *et al.*, 2017]. Also, the mask estimation approach can be applied to any microphone configurations. Given the advantages of the latter, this paper focuses on the mask estimation approach.

Mask estimation approach. To get time-invariant coefficients $\{h_{f,c}\}_{f=1, c=1}^{F, C}$, we first calculate a speech mask $m_{t,f,c}^S \in [0, 1]$. An input feature $\tilde{s}_{t,c} \in \mathbb{R}^F$ is the aggregation of the amplitudes of the c -th channel's time-frequency

bin along the frequency axis at time t :

$$\tilde{s}_{t,c} = \left\{ \sqrt{\Re(s_{t,f,c})^2 + \Im(s_{t,f,c})^2} \right\}_{f=1}^F. \quad (2)$$

The speech mask $m_{t,f,c}^S$ is acquired from the input feature $\tilde{s}_{t,c}$ as follows:

$$\{\mathbf{o}_{t,c}^S\}_{t=1}^T = \text{BiLSTM}(\{\tilde{s}_{t,c}\}_{t=1}^T; \theta_S), \quad (3)$$

$$\{m_{t,f,c}^S\}_{f=1}^F = \text{sigmoid}(\text{FCLayer}(\mathbf{o}_{t,c}^S; \phi_S)), \quad (4)$$

where BiLSTM is a real-valued bidirectional LSTM network, $\mathbf{o}_{t,c}^S \in \mathbb{R}^{D_{out}}$ is the output of BiLSTM and FCLayer is a fully connected network from $\mathbb{R}^{D_{out}} \mapsto \mathbb{R}^F$. A cross-channel power spectral density (PSD) matrix $\Phi_f^S \in \mathbb{C}^{C \times C}$ for a speech signal can be obtained with a channel-averaged mask $m_{t,f}^S$ as follows:

$$m_{t,f}^S = \frac{1}{C} \sum_{c=1}^C m_{t,f,c}^S, \quad (5)$$

$$\Phi_f^S = \frac{1}{\sum_{t=1}^T m_{t,f}^S} \sum_{t=1}^T m_{t,f}^S \mathbf{s}_{t,f} \mathbf{s}_{t,f}^\dagger, \quad (6)$$

where $\mathbf{s}_{t,f} = \{s_{t,f,c}\}_{c=1}^C \in \mathbb{C}^C$ is the channelwise concatenated vector of the STFT coefficients and \dagger represents Hermitian transpose. Using the same architecture with different parameters θ_N, ϕ_N , another PSD matrix $\Phi_f^N \in \mathbb{C}^{C \times C}$ for a noise signal is derived in the same way. Finally, the time-invariant linear filter coefficient $h_{f,c}$ is computed with the MVDR formulation as follows:

$$\{h_{f,c}\}_{c=1}^C = \frac{\Phi_f^{N-1} \Phi_f^S}{\text{Tr}(\Phi_f^{N-1} \Phi_f^S)} \mathbf{r}, \quad (7)$$

where $\text{Tr}(\cdot)$ is the trace operator and $\mathbf{r} \in \mathbb{R}^C$ is the one-hot vector indicating the index of a reference microphone. We can integrate another network to estimate the reference microphone in case the index of the reference is not specified and not available.

3.2 ESPnet

ESPnet [Watanabe *et al.*, 2018] is an end-to-end ASR which is based on connectionist temporal classification (CTC) and attention mechanism. ESPnet has an attention-based encoder-decoder structure and shares encoder representations to optimize both CTC and attention-based cross entropy objectives jointly. This joint multi-task learning framework has been known to improve performance and achieve fast convergence [Kim *et al.*, 2017]. ESPnet also incorporates the neural beamforming method as a pre-processor and optimizes the front-end within the end-to-end ASR scheme. For decoding, the weighted average of attention-based and CTC scores is used to eliminate irregular alignments. We use ESPnet as a base ASR architecture and integrate a density estimator into the multi-channel ASR in the next section.

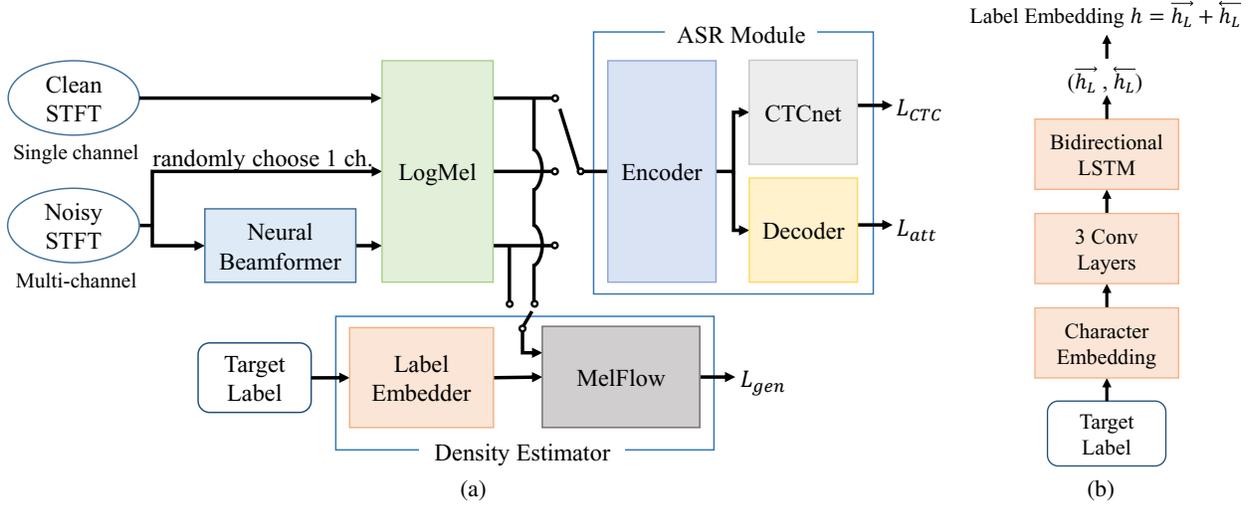


Figure 1: (a) The overall structure of the proposed model. (b) The structure of Label Embedder.

4 Proposed Model

Here, we propose to incorporate a flow-based density estimation task within the multi-channel end-to-end ASR. Our insight is that the ASR objective is insufficient for fully training the front-end since it only focuses on preserving the phonetic information. For the robust front-end, we now exploit the distribution of the target clean speech on which the original ASR system is trained.

We also present a novel flow-based generative model *MelFlow* for estimating the likelihood of mel-spectrograms. Many flow-based models for estimating the distribution of raw audio have been studied [Prenger *et al.*, 2019; Ping *et al.*, 2018], but have not been applied to mel-spectrograms. We introduce a new flow-based model for mel-spectrograms and explain the architecture of *MelFlow*.

4.1 Flow-based Generative Model

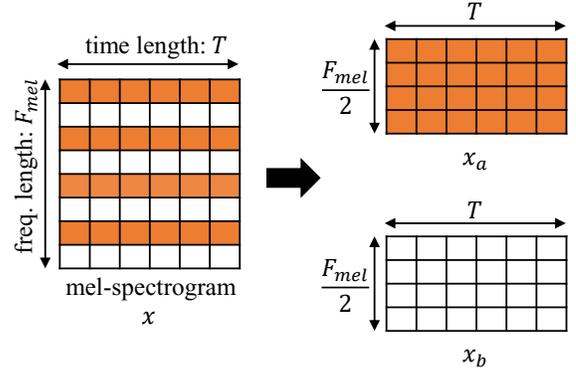
A flow-based model is a generative model which consists of a stack of invertible mappings from a simple distribution $p_Z(z)$ to a complex distribution $p_X(x)$ [Dinh *et al.*, 2016]. Let f_i be a mapping from z^{i-1} to z^i , $z^0 = x$ and $z^n = z$ ($z^i \in \mathbb{R}^D$ for $i = 0, \dots, n$). Then $x \in \mathbb{R}^D$ is transformed into z through a chain of invertible mappings:

$$z = f_n \circ f_{n-1} \circ \dots \circ f_1(x). \quad (8)$$

By change of variables theorem, the log-likelihood of data x is expressed as follows:

$$\log p_X(x) = \log p_Z(z) + \sum_{i=1}^n \log \left| \det \left(\frac{\partial f_i}{\partial z^{i-1}} \right) \right|. \quad (9)$$

By maximizing $\log p_X(x)$, we obtain a density estimator of data x . A typical choice for the prior distribution $p_Z(z)$ is a zero-mean isotropic multivariate Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$. If z is obtained by Eq. (8), the first term $\log p_Z(z)$ in Eq. (9) can be calculated easily. However, it is too expensive to compute the second term $\sum_{i=1}^n \log \left| \det \left(\frac{\partial f_i}{\partial z^{i-1}} \right) \right|$ directly. To reduce the


 Figure 2: Splitting a mel-spectrogram into x_a and x_b

computational complexity, f_i is required to have a tractable Jacobian. The affine coupling layer [Dinh *et al.*, 2016] satisfies such a requirement and is defined as follows:

$$z_a^i = z_a^{i-1}, \quad (10)$$

$$z_b^i = z_b^{i-1} \odot \exp(\sigma(z_a^{i-1})) + \mu(z_a^{i-1}), \quad (11)$$

where $z_a^i \in \mathbb{R}^{\frac{D}{2}}$ is the first half, $z_b^i \in \mathbb{R}^{\frac{D}{2}}$ is the last half of z^i , $\mu(\cdot)$ and $\sigma(\cdot)$ are the functions from $\mathbb{R}^{\frac{D}{2}} \mapsto \mathbb{R}^{\frac{D}{2}}$, and \odot stands for the element-wise product. The Jacobian matrix of the affine coupling layer is a lower triangular matrix and $\log \left| \det \left(\frac{\partial f_i}{\partial z^{i-1}} \right) \right|$ can be computed efficiently:

$$\log \left| \det \left(\frac{\partial f_i}{\partial z^{i-1}} \right) \right| = \sum_{j=1}^{\frac{D}{2}} \sigma(z_a^{i-1})_j, \quad (12)$$

where $\sigma(z_a^{i-1})_j$ is the j -th element of $\sigma(z_a^{i-1})$.

4.2 MelFlow

We now turn to building a density estimator for mel-spectrograms. An input data $x \in \mathbb{R}^{F_{mel} \times T}$ is a mel-

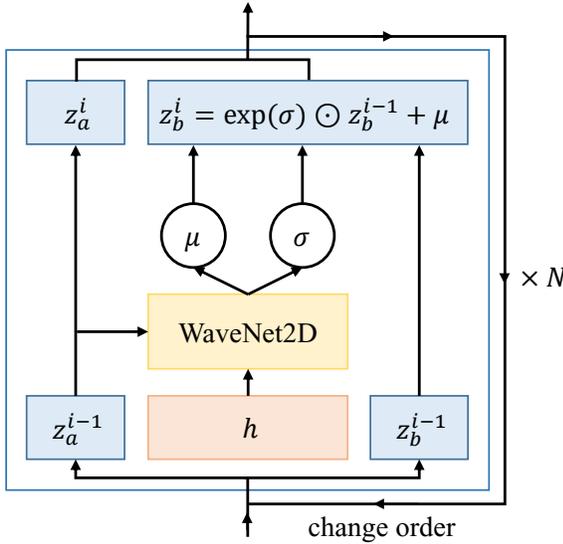


Figure 3: MelFlow network

spectrogram where F_{mel} is a fixed frequency bin length of the mel-spectrograms and T is a variable time-length depending on an utterance. It is split into $\mathbf{x}_a \in \mathbb{R}^{\frac{F_{mel}}{2} \times T}$ and $\mathbf{x}_b \in \mathbb{R}^{\frac{F_{mel}}{2} \times T}$ along the frequency axis, which is depicted in Figure 2. Let $\mathbf{z}_a^0 = \mathbf{x}_a$ and $\mathbf{z}_b^0 = \mathbf{x}_b$. Throughout one of flow stacks, \mathbf{z}_a^{i-1} and \mathbf{z}_b^{i-1} are transformed in a different way: \mathbf{z}_a^{i-1} remains and \mathbf{z}_b^{i-1} is transformed into \mathbf{z}_b^i as follows

$$(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \text{WaveNet2D}(\mathbf{z}_a^{i-1}), \quad (13)$$

$$\mathbf{z}_a^i = \mathbf{z}_a^{i-1}, \quad (14)$$

$$\mathbf{z}_b^i = \exp(\boldsymbol{\sigma}) \odot \mathbf{z}_b^{i-1} + \boldsymbol{\mu}, \quad (15)$$

where WaveNet2D can be any function $\mathbb{R}^{\frac{F_{mel}}{2} \times T} \mapsto \mathbb{R}^{F_{mel} \times T}$ and a multiplicative term $\boldsymbol{\sigma} \in \mathbb{R}^{\frac{F_{mel}}{2} \times T}$ and an additive term $\boldsymbol{\mu} \in \mathbb{R}^{\frac{F_{mel}}{2} \times T}$ depend only on \mathbf{x}_a . In this work, we use multiple layers of dilated 2D-convolutions with gated-tanh nonlinearities, residual connections and skip connections for WaveNet2D. WaveNet2D is similar to WaveNet [Oord *et al.*, 2016], but different in that WaveNet2D is composed of non-causal 2D-convolutions. The Jacobian determinant in Eq. (12) is computed as follows:

$$\log \left| \det \left(\frac{\partial \mathbf{f}_i}{\partial \mathbf{z}_{i-1}} \right) \right| = \sum_{f=1}^{\frac{F_{mel}}{2}} \sum_{t=1}^T \sigma_{f,t}. \quad (16)$$

MelFlow achieves a more flexible and high-expressive model by stacking multiple flow operations as illustrated in Figure 3. Also, we change the order of \mathbf{z}_a^i and \mathbf{z}_b^i before each flow operation (i.e. \mathbf{z}_a^i and \mathbf{z}_b^i are transformed by Eq. (15) alternately).

We can further improve the model by exploiting target labels. Note that we use the density estimator to train the front-end and target labels are available during training. As shown in Figure 1(b), a target label is first embedded into a sequence

Algorithm 1 A joint training step with density estimation

Input: A mini-batch S , a target label L , the neural beamformer M_{NB} , the density estimator M_{DE} and the ASR module M_{ASR}

if S consists of *clean speech* (single channel) **then**

$L_{ASR} \leftarrow$ compute the ASR loss with (S, L, M_{ASR})

$L_{gen} \leftarrow$ compute the NLL with (S, L, M_{DE})

optimize M_{ASR} and M_{DE} with (L_{ASR}, L_{gen})

else if S consists of *noisy speech* (multi-channel) **then**

$S_{enh} \leftarrow$ enhance by passing S through M_{NB}

sample $u \sim \text{unif}(0, 1)$

if $u < 0.5$ **then**

$S_{rand} \leftarrow$ choose 1 channel randomly from S

$(S_{ASR}, S_{gen}) \leftarrow (S_{rand}, S_{enh})$

else

$(S_{ASR}, S_{gen}) \leftarrow (S_{enh}, S_{enh})$

end if

$L_{ASR} \leftarrow$ compute the ASR loss with (S_{ASR}, L, M_{ASR})

$L_{gen} \leftarrow$ compute the NLL with (S_{gen}, L, M_{DE})

optimize M_{ASR} and M_{NB} with (L_{ASR}, L_{gen})

end if

of vectors through Character Embedding. A stack of 3 convolutional layers is applied to the sequence of vectors and the output is passed into a bidirectional LSTM. A label embedding \mathbf{h} is finally obtained by summing the last hidden state of the forward path $\overrightarrow{\mathbf{h}}_L$ and the backward path $\overleftarrow{\mathbf{h}}_L$ in the bidirectional LSTM:

$$\mathbf{h} = \overrightarrow{\mathbf{h}}_L + \overleftarrow{\mathbf{h}}_L. \quad (17)$$

Label Embedder can be considered as a compact version of the encoder in [Shen *et al.*, 2018]. The attention mechanism isn't included due to the restriction of GPU memory which is mostly occupied by the ASR module. We now reformulate Eq. (13) by adding a global condition \mathbf{h} to WaveNet2D as follows:

$$(\boldsymbol{\sigma}, \boldsymbol{\mu}) = \text{WaveNet2D}(\mathbf{z}_a^{i-1}, \mathbf{h}). \quad (18)$$

Considering the fact that T is a variable time-length, the generative loss L_{gen} (the negative log-likelihood, NLL) is defined as:

$$L_{gen} = -\frac{1}{TF_{mel}} \log p_X(\mathbf{x}), \quad (19)$$

where F_{mel} is a fixed frequency-bin length and T varies by the input data \mathbf{x} .

4.3 Joint Training with Density Estimation

The proposed model incorporates the density estimator to the base ASR architecture as shown in Figure 1(a). We take advantage of non-parallel clean and noisy speech data by employing the density estimation task. To be specific, when clean speech data comes in a mini-batch, the ASR module and the density estimator are trained ordinarily. If a noisy speech data comes in the mini-batch, cases are divided into two. In the first case with a probability 0.5, the ASR module receives a randomly chosen channel of the noisy speech. In the other case, the ASR module receives enhanced speech

from the neural beamformer. The ASR module and the neural beamformer are trained for both cases while the density estimator is only used for computing L_{gen} . Algorithm 1 describes a joint training step with density estimation.

5 Experiments

In order to evaluate the proposed method in a noisy speech scenario, we conducted a set of experiments using the CHiME-4 dataset.

5.1 Dataset

CHiME-4 is a speech recognition dataset which is recorded by a multi-microphone tablet device in every day, noisy environments. The tablet device is equipped with 6-channel microphones where 5 of them face forward and the other one faces backward. In this work, we excluded the speech data recorded by the microphone facing backward; hence the number of channels C was 5. CHiME-4 employs two types of data: (i) speech data recorded in real noisy environments (i.e., on a bus, cafe, pedestrian area, and street junction), and (ii) simulated speech data that is generated by manually mixing clean speech data with background noise. Also, the dataset is divided into training, development and evaluation sets. The training set consists of 3 hours of real noisy utterances from 4 speakers and 15 hours of simulated noisy utterances from 83 speakers. The development set consists of 2.9 hours of real and simulated noisy utterances from 4 speakers, respectively. The evaluation set consists of 2.2 hours of utterances for each real and simulated noisy data.

We also employed Wall Street Journal (WSJ) read speech for single channel clean speech dataset. WSJ’s si-284 set contains 82 hours of clean utterances and was used only for training the model.

5.2 Model Configurations

Neural Beamformer. To compute 200 STFT coefficients (i.e., $F=201$), the 25ms-width Hanning window with a 10ms shift was used. We used a 3-layer bidirectional LSTM with 300 cells for BiLSTM in Eq. (3). Also, a linear projection layer with 300 units was inserted after every layer of bidirectional LSTM. For FCLayer in Eq. (4), a 1-layer linear transformation was used. To estimate the reference microphone, a 2-layer linear transformation was used with tanh activation. The reference microphone vector r was finally estimated using the softmax function.

LogMel. STFT coefficients were converted to mel-spectrograms by LogMel. The mel-scale is primarily used to mimic the non-linear human ear perception of sound. In our experiments, F_{mel} was 80.

Label Embedder. We used a 16-dimensional character embedding. The kernel sizes of 1D convolutional layers were set to be 3 and the sizes of input and output were the same as 16. The ReLU activation and the batch normalization were used at the end of each convolutional layer. We stacked 3 convolutional layers. The sizes of the hidden state in the bidirectional LSTM were 256 and the 2 last hidden states of the forward and backward paths were summed to obtain the label embedding $h \in \mathbb{R}^{h_{dim}}$ where h_{dim} was set to be 256.

MelFlow. We used MelFlow consisting of 8 affine coupling layers. For each WaveNet2D, the kernel sizes for the first and last convolutional layer were set to be 1. The rest of the layers (i.e., middle 4 layers) was composed of 20 channels and kernel with size 3, and used for residual connections, skip connections and gated-tanh unit. For conditioning the label embedding h globally, a fully connected layer was included in WaveNet2D. All the weights of the last convolutional layers in WaveNet2D were initialized to be zero. This initialization has been known to result in the stable training procedure.

ASR module. For Encoder, a 4-layer 2D convolutional network and a 3-layer bidirectional LSTM with 1024 cells were used. The kernel sizes were set to be (3,3) for all layers in the convolutional network and channels were set to be (1, 64), (64, 64), (64, 128) and (128, 128), respectively. A linear projection layer with 1024 units was inserted after every layer of bidirectional LSTM in Encoder. To boost the ASR optimization, we adopted a joint CTC-attention loss function. For CTCnet, we used a 1-layer linear transformation with output dimension 52 indicating characters. For Decoder, a unidirectional LSTM with 1024 cells and a 1-layer linear transformation were used. To connect Encoder and Decoder, we leveraged the attention mechanism.

ASR loss. When the ASR module is trained with only the attention loss, it usually suffers from misalignment because the attention mechanism is too flexible to predict the right alignments. It has been reported that the CTC loss enforces monotonic alignments between speech and label sequences due to the left-to-right constraint [Kim *et al.*, 2017]. Thus the auxiliary CTC loss helps the attention model to have proper alignments and boosts the whole training procedure. The CTC loss L_{CTC} can be calculated efficiently with the forward-backward algorithm and the attention loss L_{att} is also easily obtained with a teacher forcing method at Decoder. The joint CTC-attention objective L_{ASR} is expressed as follows with a tuning parameter λ :

$$L_{ASR} = \lambda L_{CTC} + (1 - \lambda) L_{att}, \quad (20)$$

where we set λ to 0.5 for the experiments.

Total loss. The total loss L_{tot} is defined as:

$$L_{tot} = L_{ASR} + \beta L_{gen}, \quad (21)$$

where β is a hyperparameter. We experimented with different values of β .

Baseline. We used ESPnet as the baseline. The baseline doesn’t have Label Embedder and MelFlow in Figure 1(a). All the other configurations were the same as the proposed model.

6 Results

We compared the noisy speech recognition performances of the baseline and the proposed model on the CHiME-4 dataset. The baseline was trained with only the ASR objective L_{ASR} . We used 2 types of the proposed model in the experiment: one with both Label Embedder and MelFlow and the other with MelFlow. Also, various values of the hyperparameter β in Eq. (21) were used in the experiments: 1, 0.25, 0.1 and

		development set		evaluation set	
Model		simulated data	real data	simulated data	real data
w/o label condition	Baseline	9.1	9.2	13.6	17.2
	Proposed Model ($\beta = 1$)	8.9	9.5	13.2	17.3
	Proposed Model ($\beta = 0.25$)	8.8	9.1	12.7	17.0
	Proposed Model ($\beta = 0.1$)	8.7	9.1	13.2	17.4
	Proposed Model ($\beta = 0.01$)	9.1	8.9	13.2	17.2
with label condition	Proposed Model ($\beta = 1$)	8.6	9.1	12.9	16.7
	Proposed Model ($\beta = 0.25$)	8.1	9.0	13.1	16.8
	Proposed Model ($\beta = 0.1$)	8.5	9.1	13.2	16.7
	Proposed Model ($\beta = 0.01$)	8.4	8.9	13.3	16.3

Table 1: Word error rate [%] on CHiME-4 dataset

	SDR	ESTOI	PESQ
Baseline	15.75	0.83	1.87
Proposed Model ($\beta = 1$)	14.44	0.82	1.83
Proposed Model ($\beta = 0.25$)	15.78	0.83	1.87
Proposed Model ($\beta = 0.1$)	15.85	0.83	1.88
Proposed Model ($\beta = 0.01$)	15.87	0.83	1.88

Table 2: Speech enhancement scores on CHiME-4 simulated evaluation set

0.01. Attention scores and CTC scores were averaged at a ratio of 7:3 and a beam search algorithm with the beam size 20 was used for decoding. An RNN-based language model was also used to enhance the quality of speech recognition. Word error rates (WERs) of the outputs of the different models are shown in Table 1. Overall, the proposed model without the label condition showed better performances than the baseline. For $\beta = 0.25$, the proposed model outperformed the baseline with an absolute decrease of 0.9% in terms of WER on the simulated noisy data in the evaluation set. However, the improvement was not obvious over the real noisy data in both development and evaluation sets. When the label condition was incorporated into the model, the overall performance showed significant improvement and, surprisingly, the WERs of the proposed model were improved in all cases. For $\beta = 0.01$, the average WER on the real noisy data in the evaluation set achieved 16.3%. The experiment demonstrates that the auxiliary objective from the density estimation task leads the front-end to learn more general representations and this leads to the improved performance of noisy speech recognition. The difference of performances between the models with/without the label condition suggests that the accurate density estimator should be used in order that the front-end gets more benefits from the generative loss.

One may ask whether the proposed model achieves improvements in respect of speech enhancement. Unfortunately, the answer is no. Speech enhancement scores are illustrated in Table 2. We evaluated speech-to-distortion ratio (SDR [Vincent *et al.*, 2006]), extended short-time objective intelligibility (ESTOI [Jensen and Taal, 2016]), and percep-

tual evaluation of speech quality (PESQ [Rix *et al.*, 2001]) between the enhanced speech and the reference speech in the evaluation set. The CHiME-4 dataset provides the clean data recorded by the close-talk microphone and we used this data as the reference speech. We used the proposed model with the label condition for the speech enhancement evaluation. The overall scores of the proposed model were almost same as the ones of the baseline. This implies that in the proposed model the representation after the front-end is generalized and useful for the ASR module but this doesn't necessarily mean the improvement of the metrics of speech enhancement. Multi-task learning of speech enhancement and density estimation could be beneficial for the speech enhancement scores and we leave it for future work.

7 Conclusion

In this work, we presented the novel method which employs flow-based density estimation for robust multi-channel ASR. We also proposed *MelFlow* to estimate the distribution of mel-spectrograms of clean speech. In the experiments, we demonstrated that the proposed model shows better performance than the conventional ASR model in terms of word error rate (WER) on noisy multi-channel speech data. We verified that the auxiliary generative objective helps the front-end to learn more regularized representations which lead to improvements on noisy speech recognition.

For future work, we will apply an autoregressive model or a Gaussian mixture model (GMM) to estimate the probability density on behalf of *MelFlow*. Also, we will apply our joint training scheme with density estimation to speech enhancement.

Acknowledgments

This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1701-04.

References

[Barker *et al.*, 2017] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe. The third ‘chime’speech

- separation and recognition challenge: Analysis and outcomes. *Computer Speech & Language*, 46:605–626, 2017.
- [Dinh *et al.*, 2016] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.
- [Erdogan *et al.*, 2016] Hakan Erdogan, John R Hershey, Shinji Watanabe, Michael I Mandel, and Jonathan Le Roux. Improved mvdr beamforming using single-channel mask prediction networks. In *Interspeech*, pages 1981–1985, 2016.
- [Gong, 1995] Yifan Gong. Speech recognition in noisy environments: A survey. *Speech communication*, 16(3):261–291, 1995.
- [Han *et al.*, 2015] Kun Han, Yuxuan Wang, DeLiang Wang, William S Woods, Ivo Merks, and Tao Zhang. Learning spectral mapping for speech dereverberation and denoising. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(6):982–992, 2015.
- [Heymann *et al.*, 2019] Jahn Heymann, Lukas Drude, Reinhold Haeb-Umbach, Keisuke Kinoshita, and Tomohiro Nakatani. Joint optimization of neural network-based wpe dereverberation and acoustic model for robust online asr. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6655–6659. IEEE, 2019.
- [Jensen and Taal, 2016] Jesper Jensen and Cees H Taal. An algorithm for predicting the intelligibility of speech masked by modulated noise maskers. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):2009–2022, 2016.
- [Kim *et al.*, 2017] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4835–4839. IEEE, 2017.
- [Kinoshita *et al.*, 2016] Keisuke Kinoshita, Marc Delcroix, Sharon Gannot, Emanuël AP Habets, Reinhold Haeb-Umbach, Walter Kellermann, Volker Leutnant, Roland Maas, Tomohiro Nakatani, Bhiksha 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*, 2016(1):7, 2016.
- [Kinoshita *et al.*, 2017] Keisuke Kinoshita, Marc Delcroix, Haeyong Kwon, Takuma Mori, and Tomohiro Nakatani. Neural network-based spectrum estimation for online wpe dereverberation. In *Interspeech*, pages 384–388, 2017.
- [Meng *et al.*, 2017] Zhong Meng, Shinji Watanabe, John R Hershey, and Hakan Erdogan. Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 271–275. IEEE, 2017.
- [Ochiai *et al.*, 2017] Tsubasa Ochiai, Shinji Watanabe, Takaaki Hori, and John R Hershey. Multichannel end-to-end speech recognition. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2632–2641. JMLR. org, 2017.
- [Oord *et al.*, 2016] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- [Ping *et al.*, 2018] Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel wave generation in end-to-end text-to-speech. *arXiv preprint arXiv:1807.07281*, 2018.
- [Prenger *et al.*, 2019] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.
- [Rix *et al.*, 2001] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE, 2001.
- [Shen *et al.*, 2018] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.
- [Vincent *et al.*, 2006] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469, 2006.
- [Vincent *et al.*, 2013] Emmanuel Vincent, Jon Barker, Shinji Watanabe, Jonathan Le Roux, Francesco Nesta, and Marco Matassoni. The second ‘chime’ speech separation and recognition challenge: An overview of challenge systems and outcomes. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 162–167. IEEE, 2013.
- [Watanabe *et al.*, 2018] Shinji Watanabe, Takaaki Hori, Shigeaki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplín, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. Espnet: End-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015*, 2018.
- [Wölfel and McDonough, 2009] Matthias Wölfel and John W McDonough. *Distant speech recognition*. Wiley Online Library, 2009.