박 사 학 위 논 문
Ph.D. Dissertation

음성 대화 시스템을 위한 신경망의 새로운 음향 환경과 문장에서의 일반화

Generalization of neural network on unseen acoustic environment and sentence for spoken dialog system

2020

김 건 민 (金 建 旻 Kim, Geonmin)

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

박 사 학 위 논 문

음성 대화 시스템을 위한 신경망의 새로운 음향 환경과 문장에서의 일반화

2020

김 건 민

한 국 과 학 기 술 원

전기및전자공학부

음성 대화 시스템을 위한 신경망의 새로운 음향 환경과 문장에서의 일반화

김 건 민

위 논문은 한국과학기술원 박사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2019년 11월 20일

심사위원장 김 대 식 (인)

심 사 위 원 이 수 영 (인)

심 사 위 원 김 회 린 (인)

심 사 위 원 신 진 우 (인)

심 사 위 원 오 상 훈 (인)

Generalization of neural network on unseen acoustic environment and sentence for spoken dialog system

Geonmin Kim

Major Advisor: Daeshik Kim
Co-Advisor: Soo-Young Lee

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering

Daejeon, Korea
Nov 20, 2019

Approved by

Daeshik Kim
Professor of Electrical Engineering

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

DEE
20147012

김건민. 음성 대화 시스템을 위한 신경망의 새로운 음향 환경과 문장에서의 일반화. 전기및전자공학부 . 2020년. 101+v 쪽. 지도교수: 김 대 식, 이 수 영. (영문 논문)
Geonmin Kim. Generalization of neural network on unseen acoustic environment and sentence for spoken dialog system. School of Electrical Engineering . 2020. 101+v pages. Advisor: Daeshik Kim, Soo-Young Lee. (Text in English)

Abstract

The spoken dialog system is required to respond appropriately for diverse user queries. Generalization of spoken dialog system on unseen user query given from unseen sentence and an acoustic environment is discussed in this dissertation.

For the first part, we deal with two general problems in conventional neural sentence representation: (1) estimating embedding of the rare word and (2) no inter-sentence dependency. The above problems are simultaneously addressed with the **hierarchical composition recurrent network (HCRN)**. The HCRN consists of a 3-level hierarchy: character-word-sentence-context. This method is tested on the dialog act classification task with the DAMSL database. Compared to the conventional word-to-sentence hierarchy model, word embedding built by character-to-word hierarchy form morphologically, semantically similar clusters and sentence-to-context hierarchy reduce dialog act classification error especially for the sentence with an omission.

For the second part, we aim speech enhancement without clean speech as the target, since it is generally not obtainable in a real environment and only available for simulated data. We propose the **acoustic and adversarial supervision (AAS)** for clean-free speech enhancement. Acoustic supervision makes enhanced speech maximizes the likelihood on the pre-trained acoustic model. Therefore, enhanced speech focus on maintaining phonetic characteristic but having artifacts as a consequence of over-fitting. Adversarial supervision makes enhanced speech having a general characteristic of clean speech, however, often irrelevant to the noisy speech by consequence of mode-collapse. With proper supervision weight combination, acoustic and adversarial supervision make up for each other's limitations. This method is tested on Librispeech+DEMAND and CHiME-4 database. By visualizing the enhanced speech with different supervision combinations, we understand the aforementioned pros/cons of each supervision. Compared to the enhancement method using clean speech target, AAS achieve lower word error rate although the distance from clean speech is higher.

For the third part, we aim to achieve the source and position robustness of the enhancement model. For source robustness, we remove the source-dependency of enhancement model by using **intermic-ratio, demixing weight** as input and output of the model. Demixing weight is inherently source-independent and intermic-ratio is approximately source-independent when an analysis window is much longer than

impulse response. For position robustness, we propose the **frequency-wise complex multi-layer perceptron** given a prior analysis that position-sensitivity of demixing weight increases from low frequency to high frequency. Moreover, the target for demixing weight varies depending on model size, initialization, and training data in a minibatch since the global optimal of demixing weight is non-uniquely determined. We propose the **reference position regularization** to reduce training target variance by uniquely determine true demixing weight. The proposed method is tested on the simulated reverberant dataset with varying source position while room and mics are fixed. Compared to conventional source-dependent training methods, the proposed source-independent method achieves a higher signal-to-distortion ratio especially the number of training sources is small. While proposed model tend to overfit to training positions, the reference position regularization alleviates signal-to-distortion ratio drop on out-of-training position.

Keywords Sentence representation, Out-of-vocabulary, Dialog context, Clean-free speech enhancement, Source/Position robustness

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Target problems

Spoken Dialog System (SDS) is employed in many applications such as smart speaker (i.e Alexa, Allo), and robotics. A deep neural network is replacing many modules in a spoken dialog system due to superior representation power and generalization capability. Representative applications include speech recognition [1], neural conversation model [2], and text-to-speech [3]. However, it is still difficult for a neural network to behave properly for test data with the unseen condition. In this thesis, we aim to improve generalization for *Language Understanding*, and *Speech Recognition* shown in Figure 1.1.

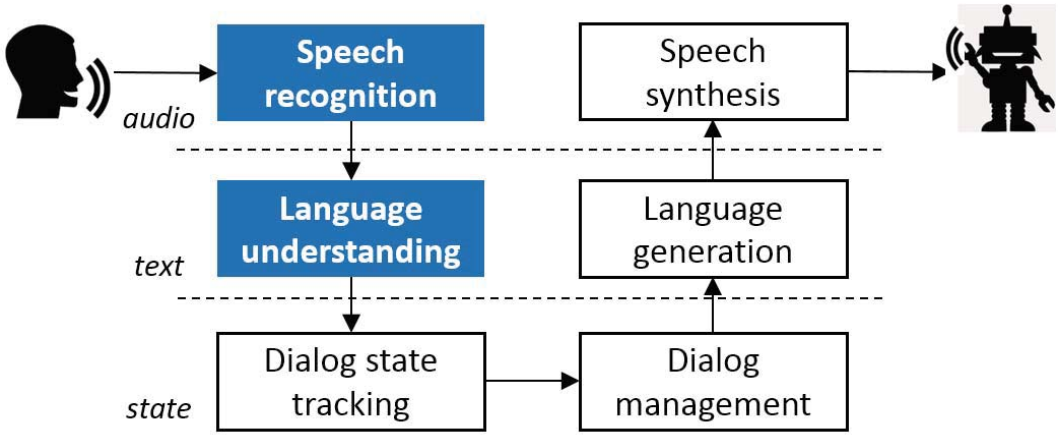


figure 1.1: Diagram for Spoken Dialog System. We address two front parts of the system: speech recognition and language understanding

For *language understanding*, we focus on the problem of the **unseen word and context** in compositional sentence representation. Rare words such as partial word or disfluency can often appear in spoken text, and its word embedding is prone to overfitting since they barely experienced during training. Moreover, the intent of a sentence is often ambiguous without knowing the previous context of the dialog. These two problems are general interests in natural language processing, however, more emphasized in the dialog system.

For robust *speech recognition*, we address speech enhancement on the unseen condition of the acoustic environment, such as additive noise, reverberation, and channel distortion. Specifically, two problems of simulated multi-condition training (SMCT), on which most speech enhancement methods are based, are addressed.

The first problem of the SMCT is using a **simulated clean speech as target** of enhancement training. This approach cannot use noisy speech collected from the real environment. Moreover, the training method is optimized for maximizing signal-to-distortion ratio, but suboptimal for minimizing

the word error rate.

The second problem of the SMCT is a **source and environment sensitivity** of the speech enhancement model. Speech enhancement performance, typically measured by signal-to-distortion ratio, often degraded when test data with unseen sources and the environment is given. Most of the approaches simply make/collect (noisy, clean) paired speech datasets from a diverse environment, and train enhancement model to learn the mapping between them. However, we cannot expect this model generalizes to unseen environment and source.

1.2 Contributions of the thesis

The contributions of the thesis for each problem are summarized as follows.

Problem 1. unseen word and context for compositional sentence representation

- **Hierarchical Composition Recurrent Network (HCRN)**: The HCRN includes a 3-level hierarchy of compositional models: character, word, and sentence. The inclusion of the compositional character model improves the quality of word representation, especially for rare and unseen words. Moreover, the embedding of inter-sentence dependency into sentence representation by the compositional sentence enable context-aware sentence representation. Compared with the conventional stacked RNN, the HCRN deals with segmented shorter sequences at each level, and thereby the vanishing gradient problem is rendered relatively insignificant.
- **Hierarchy-wise learning algorithm**: The HCRN is trained in a hierarchy-wise training fashion, alleviating optimization difficulties in an end-to-end training.

Problem 2. clean speech as target of speech enhancement

- **Reconstruction/Acoustic and adversarial supervision (RAS/AAS)**: The RAS and AAS are designed for improving speech intelligibility, and speech recognition performance respectively. Reconstruction, acoustic, and adversarial supervision each make i) enhancement model learn the latent feature of speech, ii) enhanced speech minimizes the word error rate of the pre-trained acoustic model, and iii) enhanced speech having general characteristics of clean speech. The RAS/AAS are both clean-free enhancement methods, therefore they can use both real noisy data and simulated noisy data for training. AAS shows a lower word error rate compared to speech enhancement methods using clean speech targets.

Problem 3. source/position sensitivity of de-reverberation

for source robustness

- **Intermic ratio (input), Demixing weight (output)**: With multiplicative transfer function (MTF) approximation, source, mic, and transfer function in the frequency domain have an approximately linear relationship. Demixing weight is coefficients for such a linear relationship. The demixing weight is only a function of mixing environment (e.g., room size, reflection coefficients, location of microphone and source) and independent to speech sources, enable source-independent de-reverberation.

  We estimate demixing weight from the inter-mic ratio feature. The inter-mic ratio is the ratio between multi-mic signals which include spatial information. With MTF approximation, the intermic ratio is solely a function of mixing weight, and suitable for estimating source-independent demixing weight.

for position robustness

- **Frequency-wise complex multi-layer perceptron**: There are three basic properties for enhancement model architecture. First, parameters for each frequency are independent in a model because our model is based on frequency-wise linear demixing weight, and the model needs to learn different position sensitivity per frequency. Second, parameters are shared across time because the mixing process is stationary. Thrid, parameters, and operation are defined in the complex field since input and output are complex-valued data.
- **Reference position regularization**: The global optimal of demixing weight is not uniquely determined. Therefore, demixing weight for each position may vary depending on the size/initialization of the neural network and other training positions neural network can experience. To reduce training target variance, we uniquely determine the demixing weight on the determined system by adding more equations with reference position. The reference position regularization can be interpreted as making a sharp spatial filter. For example, demixing weight suppresses sources from the reference position (i.e., background music, motor sound nearby robot microphones) enhances the user's voice from the target position.

Chapter 2. Compositional sentence representation from character within large context text

2.1 Problem

Sentence representations are usually built from representations of constituent word sequences using a compositional word model. Many compositional word models based on neural networks have been proposed, and have been used for sentence classification [4, 5] or generation [6, 7] tasks. However, learning to represent a sentence based on constituent word sequences involves two difficulties. First, estimating the embedding of rare words suffers from the data-sparsity problem and poorly estimated embedding can cause sentence representations of inferior quality. Second, conventional sentence representation does not take into account inter-sentence dependency, which is an important linguistic context for understanding the intention of the sentence.

In this paper, we propose a Hierarchical Composition Recurrent Network (HCRN), which consists of a hierarchy of 3 levels of compositional models: character, word, and sentence. Sequences at each level are composed of a Recurrent Neural Network (RNN). In the HCRN, the output of the lower levels of the compositional model is fed into higher levels. Sentence representation by the HCRN enjoys several advantages compared to sentence representation by a single compositional word model. From the compositional character model, the word representation is built from characters by modeling morphological processes shared by different words. In this way, the data-sparsity problem with rare words is resolved. From the compositional sentence model, inter-sentence dependency can be embedded into sentence representation. Sentence representation with inter-sentence dependency can capture implicit intention as well as explicit semantics of a given sentence. Training the HCRN in an end-to-end fashion has optimization difficulties because a deep hierarchical recurrent network may suffer from the vanishing gradient problem across different levels in the hierarchy. To alleviate this, a hierarchy-wise language learning algorithm is proposed, and it is empirically shown that it improves the network's optimization. The efficacy of the proposed method is verified on a spoken dialogue act classification task. The task is to classify the communicative intentions of sentences in spoken dialogues. Compared to conventional sentence classification, this task presents two challenging problems. First, it requires that the model estimate representations of spoken words which often include rare and partial words. Second, understanding the dialogue context is often required to clarify the meanings of sentences within a given dialogue. The HCRN with the hierarchy-wise learning algorithm achieves *state-of-the-art* performance on the SWBD-DAMSL database. The source code of this work is available at `github.com/gmkim90/HCRN_DA`.

2.2 Related work

The out-of-vocabulary problem has been addressed by building word representation from its constituents [8, 9] or copying mechanism [10, 11]. Our model is based on former approach as in Figure 2.1. The representative constituents are character and morpheme. A character as a basic unit can be applied to any word, and robust to misspelling. However, a character itself has no semantic/syntactic information and often making a long sequence to process. As a basic unit, morpheme has semantic/syntactic information. However, the performance is sensitive to the morpheme analyzer. Our model is based on character as a basic unit and applied for learning sentence representation of spoken dialog containing many partial words and ungrammatical sentences.

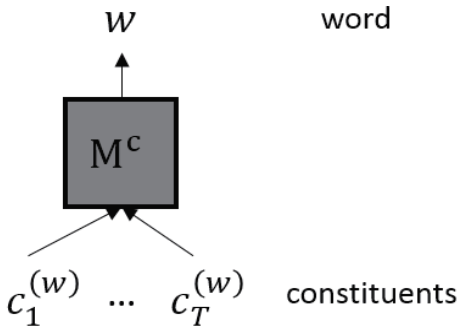


figure 2.1: Building word from constituents. The representative constituents are character and morpheme.

Lack of context problem have been addressed by conveying context information into sentence representation as shown in Figure 2.2.

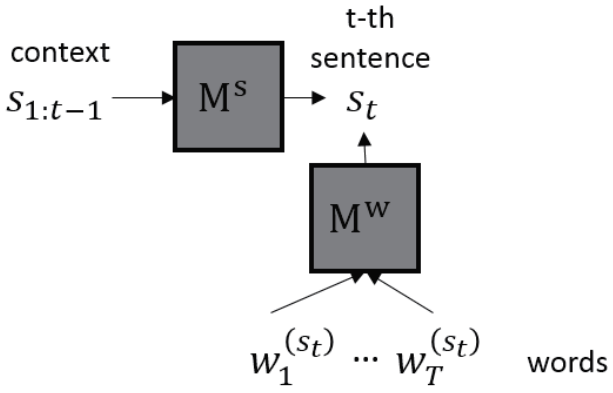


figure 2.2: Conveying dialog context into sentence representation.

Both word composition from constituents, and embedding context to the sentence makes sequence longer and training challenging. The difficulty for RNNs in learning long-range dependencies within character sequences has been addressed in [12]. Hierarchical RNNs have been proposed as one possible solution, which design RNN architecture in which different layers learn at different speeds of dynamics [13, 14, 15]. Compared with these models, we separate the hierarchy of sequences on the recurrent neural network. Each hierarchy deals with segmented shorter sequences, and thereby the vanishing

gradient problem is rendered relatively insignificant. There are several recent studies on representing large context text hierarchically for document classification [16] and on the dialogue response model [17]. These approaches benefit from hierarchical representations that represent long sequences as a hierarchy of shorter sequences. However, the basic unit used in these approaches is the word, and models that begin at this level of representation open themselves to the data sparsity problem. This problem is somewhat resolved by building word representations from character sequences. Successful examples can be found in language modeling [18, 19] and machine translation [20]. em has been addressed by building word representation from its constituents [8, 9] or coping mechanism.

2.3 Method

2.3.1 Hierarchical composition recurrent network (HCRN)

Figure 2.3 shows our proposed Hierarchical Composition Recurrent Network (HCRN). Consider a dialogue D consisting of sentence sequences $s_{1:T_D}$ and its associated label $t_{1:T_D}$. The HCRN consists of a hierarchy of RNNs with compositional character, compositional word and compositional sentence levels. At each level, each sequence encoding($\mathbf{e}$) is obtained by the hidden neuron($\mathbf{h}$) of RNN at the end of the sequence [21, 16]. Compositional Sentence model additionally takes a binary vector which indicates speaker identity change across dialogue. The composition rule is summarized in Table 2.1. The notation of well-known transformations are represented as follows: gating units such as LSTM or GRU are represented as g, and Multi Layer Perceptron with Softmax non-linearity as r.

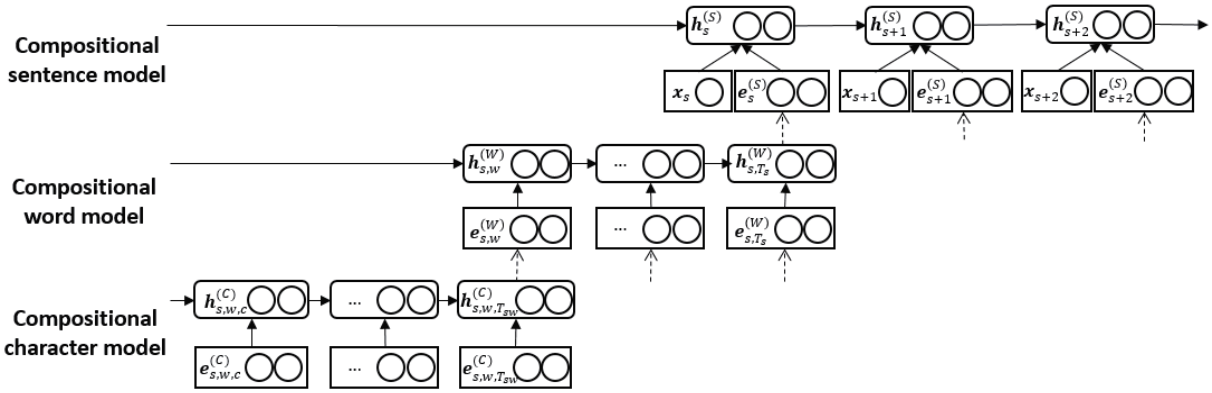


figure 2.3: Illustration of the Hierarchical Composition Recurrent Network. The thick arrows indicate affine and non-linear transformation, the thin arrows indicate identity transformation. For simplicity, each level is shown with one layer.

table 2.1: Composition rules used in each hierarchy of HCRN

Compositional Character	Compositional Word	Compositional Sentence
$\mathbf{h}^{(C)}_{s,w,c} = g(\mathbf{h}^{(C)}_{s,w,c-1}, \mathbf{e}^{(C)}_{s,w,c})$	$\mathbf{h}^{(W)}_{s,w} = g(\mathbf{h}^{(W)}_{s,w-1}, \mathbf{e}^{(W)}_{s,w})$	$\mathbf{h}^{(S)}_{s} = g(\mathbf{h}^{(S)}_{s-1}, \mathbf{e}^{(S)}_{s}, \mathbf{x}_s)$
$\mathbf{e}^{(W)}_{s,w} = \mathbf{h}^{(C)}_{s,w,T_{sw}}$	$\mathbf{e}^{(S)}_{s} = \mathbf{h}^{(W)}_{s,T_s}$	$P(y_s \vert \mathbf{s}_{1:s}) = r(\mathbf{h}^{(S)}_{s})$

Loss is defined as the negative log-likelihood of the label of the sentences within dialogue.

$$L(s_{1:T_D}, l_{1:T_D}) = -\log P(y_{1:T_D} = l_{1:T_D} | s_{1:T_D}) = -\sum_{t=1}^{T_D} \log P(y_t = l_t | s_{1:t})$$

Advantage of hierarchical composition

One advantage of the HCRN is its ability to learn long character sequences. Figure 2.4 compare hierarchical composition and conventional stacked RNN. While stacked RNN should process sequence with length O(WC) to generate sentence vector, hierarchical composition enable O(W+C) sequence

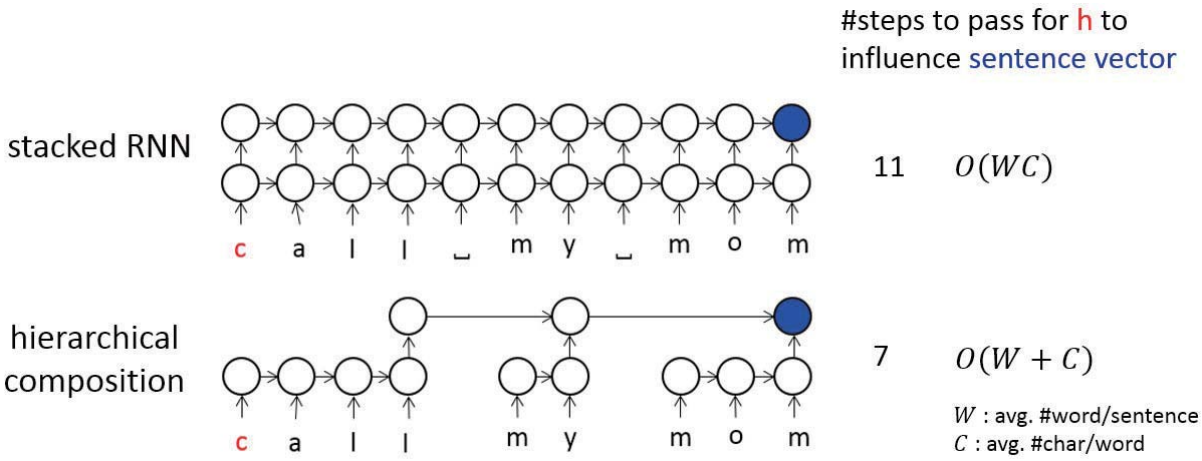


figure 2.4: Advantage of hierarchical composition. Each hierarchy process short segmented sequence, and thereby having less vanishing gradient problem during backpropagation through time.

length.

For example, in our experiment, the sentence becomes a much longer sequence when represented by characters (37.92) compared to words (8.28), in average. While conventional stacked RNNs have difficulty when dealing with very long sequences [22, 23], the hierarchy of the HCRN deals with segmented short sequences at each level so vanishing gradient problems during back-propagation through time are relatively insignificant. Each level of the HCRN uses a different speed of dynamics during sequence processing, so that the model can learn both short-range and long-range dependencies in large text samples.

Moreover, *character to word* enables a smaller number of parameters compared to non-compositional word embedding. Non-compositional word embedding requires word vocabulary size(100K) X embedding dimension. However, compositional words require character vocabulary size(100) + character-to-word model parameter. Therefore, the compositional word is parameter efficient for large vocabulary natural language processing tasks. The following abbreviations are used for the rest of this paper: the compositional character model (CC), the compositional word model (CW), the compositional sentence model (CS), and the multi layer perceptron (MLP).

2.3.2 Hierarchy-wise learning algorithm

To alleviate the optimization difficulties of end-to-end training of HCRN, hierarchy-wise language learning is proposed. In the hierarchy of composition models, the lower level composition network is trained first, higher level composition layers are gradually added after the lower level network is optimized for a given objective function. This approach is inspired by the unsupervised layer-wise pre-training algorithm in [24], which is known to provide better initialization for subsequent supervised learning. Word-hierarchy proceeds in an unsupervised way by reconstructing the character sequence of each word using RNN Autoencoder, proposed in [25]. This stage is viewed as *learning to spell*. Sentence-hierarchy and Discourse-hierarchy learning proceed in a supervised way to classify the given label of the sentence.

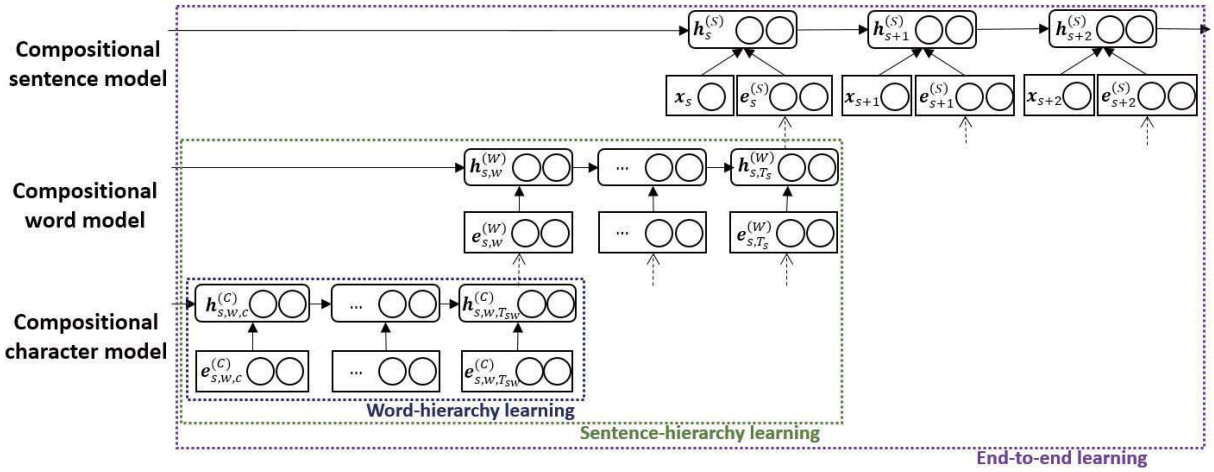


figure 2.5: Illustration of hierarchy-wise learning. Lower level is pre-trained and served initialization for end-to-end training.

2.4 Experiment

2.4.1 Task and dataset

The HCRN was tested on a spoken dialogue act classification task. The dialogue act (DA) is the communicative intention of a speaker in each sentence. We chose the SWBD-DAMSL database[1], which is a subset of the Switchboard-I (LDC97S62) dialogue corpus annotated with DA for each sentence. The SWBD-DAMSL has 1155 dialogues on 70 pre-defined topics, 0.22M sentences, 1.4M word tokens and a 42-class tagset shown in Table 2.2. The number of elements in the character dictionary is 31 including 26 letters, - (indicating a partial word), '(indicating possessive case), . (indicating abbreviation), <noise> (indicating non-verbal sound) and <unk> (indicating unknown symbols) for all other characters. We follow the train/test set division in [26]: 1115/19 dialogues, respectively. Validation data includes 19 dialogues chosen from the training data. After pre-processing of the corpus, the number of sentences in the train/test/validation sets are 197370, 4190 and 3315 respectively. All letters are converted into lower-case. Disfluency tags and special punctuation marks such as (? ! ,) which cannot be produced by a speech recognizer are removed.

Detail of pre-processing

We pre-processed raw text according to several rules below. At first, all letters are converted into lower-case. Disfluency tags and special punctuation marks such as (? ! ,) which cannot be produced by a speech recognizer are removed. We also merged sentences with segment tags into previous unfinished sentences. Segment tags indicate the interruption of one speaker by another. It is difficult even for human beings to predict tags of segmented sentences because sentence segments often do not provide enough information for the reliable ascription of its DA. Sentences that interrupt others are placed after combined sentences.

table 2.2: 42-class tagset of dialogue act provided from SWBD-DAMSL. Classes are sorted from the most frequent to the least frequent, from top-left to bottom-right with column-major order.

Non-opinion	Declarative question	Other answers
Backchannel	Backchannel(question)	Opening
Opinion	Quotation	Or clause
Abandoned	Summarize	Dispreferred answer
Agreement	Non-yes answer	3rd party talk
Appreciation	Action-directive	Offers
Yes-No-Question	Completion	Self talk
Non-verbal	Repeat phrase	Downplayer
Yes answer	Open question	Accept part
Closing	Rhetorical question	Tag question
Wh-question	Hold before answer	Declartive question
No answer	Reject	Apology
Acknowledgment	Non-no answer	Thanking
Hedge	Non-understand	Others

[1]The dataset is available at `https://web.stanford.edu/~jurafsky/swb1_dialogact_annot.tar.gz`

2.4.2 Hyperparameters

We employ the Gated Recurrent Unit (GRU) as a basic unit of the RNN[27, 28]. The configuration of the HCRN is represented by the hierarchy of the compositional level and its size, $CC_{size}-CW_{size}-CS_{size}$ as shown in Table 2.3. In all supervised learning, the classifier consists of 3-layers and 128 hidden units MLP with Rectified Linear Units and Softmax non-linearity. A common hyperparameter setting is used in all experiments. All weights are initialized from a uniform distribution within [-0.1, 0.1] except for the pre-trained weights. We optimized all networks with adadelta [29] with decay rate (ρ) 0.9 and constant (ϵ) 10^{-6} , gradient clipping with threshold 5, and a batch size of 64, 64, 8 for word, sentence, and discourse hierarchy learning. Early stopping based on validation loss was used to prevent overfitting.

table 2.3: Size of compositional model at each level, represented by (# layers) × (# cell in each layer). Note that the complexity of the model increases as the level of hierarchy increases, following the assumption that the complexity of composition increases as the level of language increases.

	CC	CW	CS
Small	1×64	2×128	2×256
Large	2×128	3×256	3×512

2.4.3 Result: Unsupervised word-hierarchy learning

During word-hierarchy learning, CC is jointly trained with the RNN Decoder to reconstruct input character sequences. The number of all unique words in the training set is 19353. The *end of word* token is appended to every end of the character sequence. Learning is terminated if validation loss fails to decrease by 0.1% for three consecutive epochs.

table 2.4: Reconstruction performance of the RNN Encoder-Decoder on words in the vocabulary and out-of-vocabulary (OOV). The length column presents the mean and standard deviation (in parentheses) of the character length of words for which complete reconstruction failed.

Model	In Vocabulary			Out of Vocabulary		
	CPER (%)	WRFR (%)	Length	CPER (%)	WRFR (%)	Length
$CC_{1\times64}$	0.39	2.25	13.1(2.6)	2.06	9.17	12.3(2.2)
$CC_{2\times128}$	0	0	-	1.21	5.28	12.7(2.4)

Pre-training performance itself is evaluated by sequence reconstruction ability. For reconstruction, the RNN Decoder generates character sequences from the encoder vector which is the last time step hidden neuron of CC. Generation is performed based on greedy sampling at each time step. The performance is evaluated on two measures: Character Prediction Error Rate (CPER) and Word Reconstruction Fail Rate (WRFR). CPER measures the ratio of incorrectly predicted characters to the reconstructed sequence. WRFR is the ratio of words where complete reconstruction fails out of the total words in the test set. The reconstruction performance of the RNN Encoder-Decoder on words both in vocabulary and out-of-vocabulary (OOV) is summarized in Table 2.4. Overall, the model almost perfectly reconstructs the character sequences of the training data, and even generalizes well for the unseen words. The large

size model outperforms the small size model. Almost all cases in which reconstruction failed involved sequences longer than 12 characters on average.

2.4.4 Result: Supervised word and sentence-hierarchy learning

Initialization of CC: Random VS. Pre-trained The performance of sentence-hierarchy learning with and without the word-hierarchy learning are compared to evaluate how the pre-trained CC provides useful initialization for sentence-hierarchy learning. With the pre-trained CC, at first, the parameters of the CC are frozen, and the CW and MLP are trained for 1 epoch[2]. After that, the whole architecture consisting of the CC, CW and MLP is jointly trained. Evaluation was performed on architectures with different CC and CW sizes (see Table 2.3).

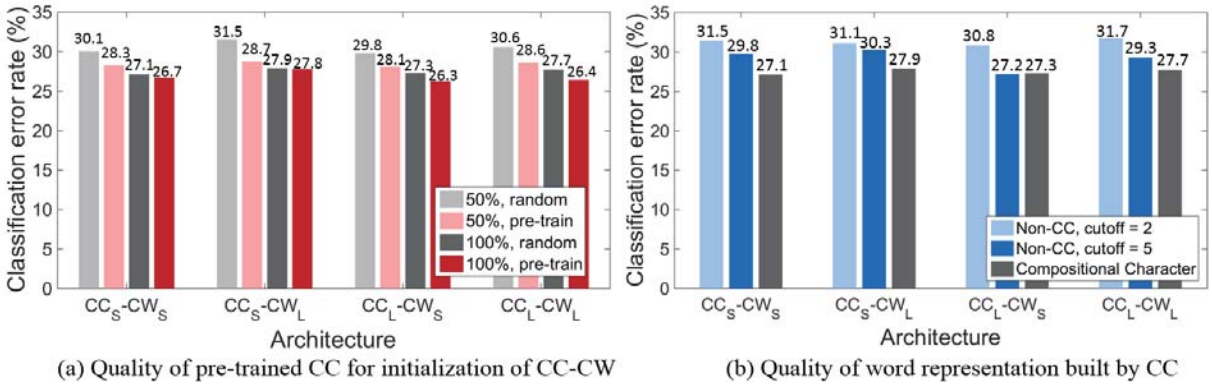


figure 2.6: Result to show the quality of our pre-trained CC for initialization on sentence-hierarchy learning. (a) Test error rate (%) of comparing learning CC-CW with different initialization : pre-trained CC and random. (b) Test error rate (%) of comparing learning CC-CW and CW. At each level, size is represented as either small (S) or large (L) in Table 2.3.

In addition, pre-training on two different training dataset sizes (50 % and 100 %) are compared. The results are shown in Figure 2.6(a). Pre-training consistently reduces the test error rate on the various architectures, especially when fewer training data are available.

CC VS. non-compositional word embedding We compare two different methods to build word representation in this section: CC and conventional non-compositional word embedding (=non-CC). Since CC is used to learn the morphological structures of words, it is not comparable with widely used pre-trained word embedding such as Word2Vec [30], which aims to learn semantic/syntactic similarities between different words. Therefore, for a fair comparison we randomly initialized both models rather than employing pre-trained word embedding.

For the non-CC embedding method, we set two different cutoff frequencies: $\tau_c = 5$ (6294 words), $\tau_c = 2$ (11746 words). Word embedding size is 64 and 128 for both CC and non-CC.

Figure 2.6(b) shows the comparison of test error rates with the above settings. Non-compositional

[2]The number of epochs to freeze the pre-trained model is chosen as the best parameter from preliminary experiments on the validation set.

word embedding with the high cutoff ($\tau_c = 5$) outperforms the low cutoff ($\tau_c = 2$). This is because the data sparsity problem during the estimation of rare words is more severe for the model with the lower cutoff setting. Compared to the non-CC, CC outperforms or is on a par, with fewer parameters.

Table 2.5 shows the 3-nearest neighbors of word representation built by two different methods: CC and non-compositional representation. For each method, the model with the best test accuracy is chosen. For word representation built by CC, retrieved nearest words usually have a similar analogy with similar meaning. Moreover, rare words and OOVs such as partial words can be mapped to semantically similar words, which are usually estimated by a single unknown token in non-compositional word representation.

table 2.5: Comparison of word representation built from CC and non-compositional word embedding. The nearest 3 words by Euclidean distance are retrieved for given target word.

Word representation method	Unigram counts $\geqq 5$		$1 \leqq$ Unigram counts $\leqq 4$		Out of Vocabulary (OOV)	
	uh-huh	really	emphasizing	probab-	environmentalism	seventy-eights
Compositional Character Model	uh-oh huh-uh um	reall- real very	emphasize emphasis surpassing	probably probability probable	environmentalist environmentals environmental	ninety-eight seventeenth twentiy-six
Non-compositional ($\tau_c = 5$)	hmm helpful yeah	believe very frankly	-		-	

Effects of Hierarchical Composition

In order to learn character-level representation of large text such as sentence and dialogue, our model is hierarchically composed. As a preliminary experiment, we compared the performance of hierarchical (Fig 6-(a), sentence-hierarchy learning in paper) and non-hierarchical (Fig 6-(b), conventional stacked RNN) architectures on dialogue act classification tasks with the SWBD-DAMSL database.

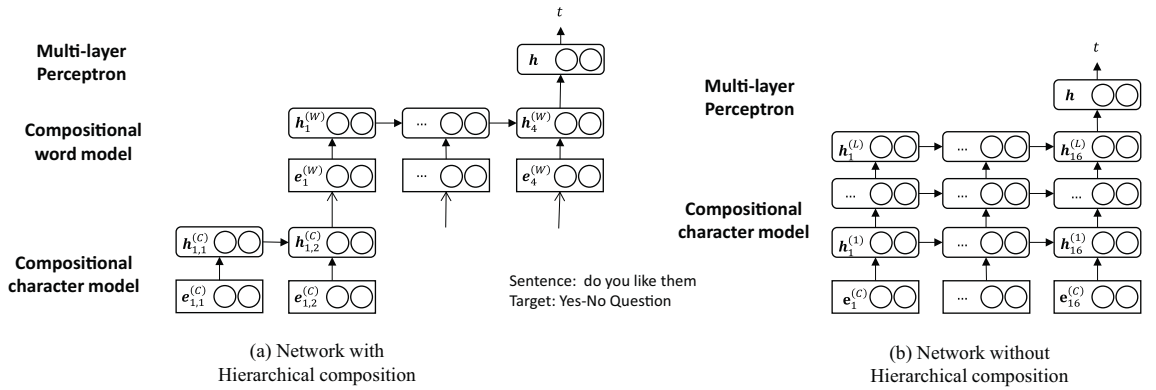


figure 2.7: Comparison of network to learn sentence representation from character. (a) With hierarchical composition (b) Without hierarchical composition(conventional stacked RNN)

For a fair comparison, hierarchical composition and non-hierarchical composition network have the

same number of hidden units in each layer. For the non-hierarchical composition network, we add blank tokens in character sequences to indicate word boundaries. Table 8 shows that the hierarchical composition network outperforms the non-hierarchical composition network. It is because the network with hierarchical composition has a compositional word model (CW) which can be viewed as a specially designed compositional character model (CC), where the composition is only allowed between hidden neurons at the ends of words. Therefore, the network with hierarchical composition has shorter sequences in each layer compared to the non-hierarchical model, where the vanishing gradient problem becomes relatively insignificant in each layer of RNN.

table 2.6: Test error rate comparison of network with/without hierarchical composition. The relative improvement (Rel.) from non-hierarchical to hierarchical composition is also reported. '+' indicates consecutive layers in a conventional stacked RNN.

Non-hierarchical composition		Hierarchical composition		
Model	Error (%)	Model	Error (%)	Rel. (%)
$RNN_{1L,64H+2L,128H}$	35.80	$CC_{1L,64H} - CW_{2L,128H}$	27.13	24.22
$RNN_{1L,64H+3L,256H}$	36.38	$CC_{1L,64H} - CW_{3L,256H}$	27.81	23.56

2.4.5 Result: Discourse-hierarchy learning

During discourse-hierarchy learning, the CS on top of the $CC_{2\times128} - CW_{2\times128}$ is trained. For the first 5 epochs, the network is trained with the CC and CW frozen. Then, the whole network is jointly optimized.

Optimization difficulty of end-to-end learning We compared the learning curves of discourse-hierarchy learning using two different model initializations: with the pre-trained model from sentence-hierarchy learning, and with random initialization (end-to-end learning). The learning curve in Figure 2.8 clearly shows that initializing with the pre-trained model significantly alleviates optimization difficulties.

table 2.7: Test error rate of sentence hierarchy learning and discourse hierarchy learning. Discourse-hierarchy learning outperforms Sentence-hierarchy learning.

Hierarchy	Model	Err (%)
Sentence	$CC_{2\times128} - CW_{2\times128}$	26.27
Discourse	$CC_{2\times128} - CW_{2\times128} - CS_{2\times256}$	**22.73**
	$CC_{2\times128} - CW_{2\times128} - CS_{3\times512}$	22.99

Effects of dialogue context on sentence representation Table 2.7 shows the test classification error rate of sentence-hierarchy learning and discourse-hierarchy learning. Compared to sentence-hierarchy learning, discourse-hierarchy learning improves performance significantly.

Figure 7 shows percent improvement when dialogue context is incorporated for each class label. Two

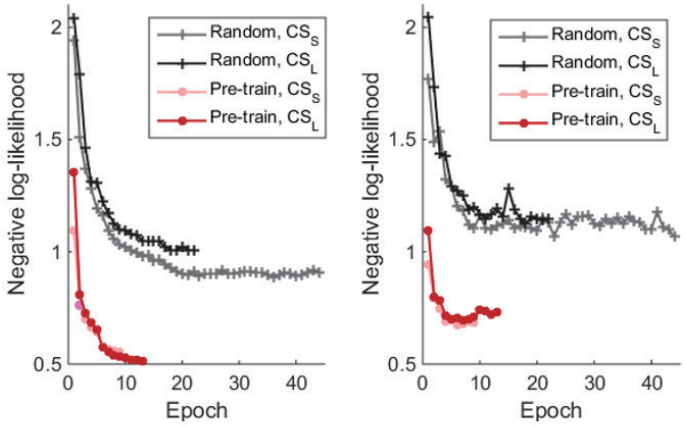


figure 2.8: Learning curve on (a) training data and (b) test data. The objective function is converged to a much lower value when the model employs initialization from the pre-trained model resulting from sentence-hierarchy learning.

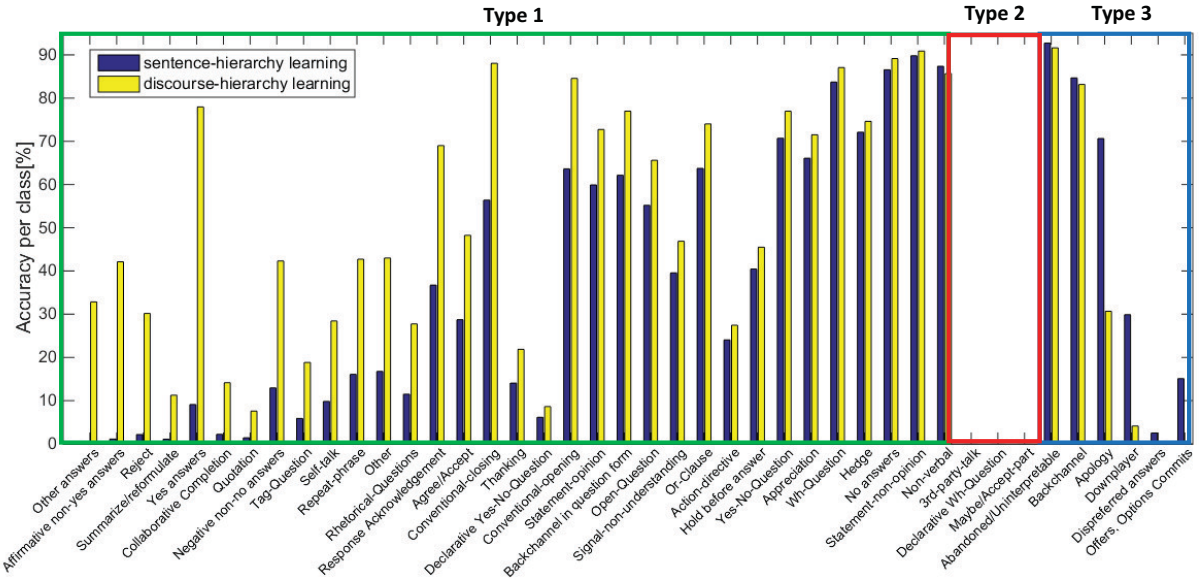


figure 2.9: Class accuracy for sentence-hierarchy learning and discourse-hierarchy learning.

models that achieve the best test set accuracy on sentence-hierarchy learning (without dialogue context) and dialogue-hierarchy learning (with dialogue context) are compared. Classes are sorted by descending order of relative improvement rate (from sentence-hierarchy to discourse-hierarchy). 33 out of 42 class improved with discourse-hierarchy learning (Type 1). Performance is degraded with discourse-hierarchy learning for 6 classes (Type 3). Also, there are 3 classes where both methods fail to predict. (Type 2)

To qualitatively analyze the improvement, we show examples of sentences on the test set for which prediction is improved by dialogue context. Analysis is done with model $CC_{2\times128} - CW_{2\times128} - CS_{2\times256}$, which achieved the best test accuracy during discourse-hierarchy learning. Table 2.8 shows a dialogue example including 8 sentences, comparing sentence-hierarchy and discourse-hierarchy learning. Highlighted sentences indicate cases where discourse-hierarchy learning predicts correctly while sentence-hierarchy

table 2.8: An example of dialogue segment containing 8 sentences. Predictions of label from model of sentence-hierarchy learning (without dialogue-context) and discourse-hierarchy learning (with dialogue-context) are provided along with true labels.

Dialogue segment	True	Without Context	With Context
A: and uh quite honestly i just got so fed up with it i just could not stand it any more	S	S	S
B: is that right	**BQ**	**YQ**	**BQ**
A: yeah	**A**	**B**	**A**
A: i mean this is the kind of thing you look at	**S**	**O**	**S**
B: yeah	B	B	B
A: you sit there	S	S	S
A: and when you are writing up budgets you wonder okay how much money do we need	**S**	**WQ**	**S**

* S=Statement, A=Agreement, B=Backchannel, WQ=Wh-Question
YQ=Yes-No-Question, O=Opinion, BQ=Backchannel-Question

learning fails to predict. For example, "yeah" in the 3rd sentence of the example can be interpreted as both Agreement and Backchannel, and an informed decision between the two is only possible when the dialogue context is available. This example demonstrates that sentence representation with dialogue context helps to distinguish confusing dialogue acts.

Comparison with other methods Several other methods for dialogue act classification are compared with our approach in Table 2.9. Our approach outperforms the other benchmarks, achieving a 22.7% classification error rate on the test set. Similar approaches employ a neural network that hierarchically composes sequences starting from word sequences [31, 32, 17]. We conjecture that the improvement demonstrated by our model is due to two factors. First, our model builds word representations from constituent characters and so suffers less from the data sparsity problem when learning the embedding of rare words. Second, the hierarchy-wise language learning method alleviates the optimization difficulties of the deep hierarchical recurrent network.

table 2.9: Performance comparison with other methods for dialogue act classification on SWBD-DAMSL.

Method	Test err. (%)
Class based LM + HMM [26]	29.0
RCNN [31]	26.1
HCRN with word as basic unit + End-to-End learning* [17]	24.9
Utterance feature + Tri-gram context + Active learning + SVM [33]	23.5
Discourse model + RNNLM [32]	23.0
HCRN with character as basic unit + Hierarchy-wise learning	**22.7**

*We evaluated this performance by ourselves due to task difference.

2.5 Conclusion

In this paper, we introduced the Hierarchical Composition Recurrent Network (HCRN) model consisting of a 3-level hierarchy of compositional models: character, word, and sentence. The inclusion of the compositional character model improves the quality of word representation, especially for rare and OOV words. Moreover, the embedding of inter-sentence dependency into sentence representation by the compositional sentence model significantly improves the performance of dialogue act classification. The HCRN is trained in a hierarchy-wise language learning fashion, alleviating optimization difficulties with end-to-end training. In the end, the proposed HCRN using the hierarchy-wise learning algorithm achieves *state-of-the-art* performance with a test classification error rate of 22.7 % on the dialogue act classification task on the SWBD-DAMSL database.

Chapter 3. Clean-free speech enhancement

3.1 Problem

Speech observation can be corrupted with additive noise, reverberation/channel distortion shown in Figure 3.1. The corruption process is given below:

$$y(t) = s(t) * h(t) + n(t) \tag{3.1}$$

where $y(t)$ is observed speech, $s(t)$: clean speech, $h(t)$: reverberation and channel distortion, and $n(t)$: additive noise. This corruption harm speech intelligibility and speech recognition performance. The speech enhancement aims to improve either speech intelligibility or speech recognition.

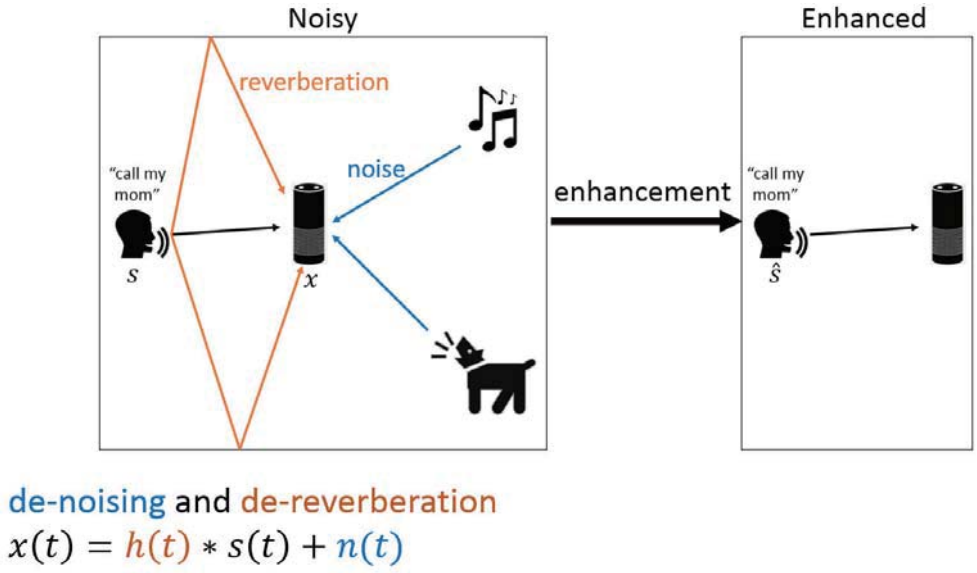


figure 3.1: Two major problems of speech enhancement: de-noising and de-reverberation.

Most of the speech enhancement approaches are based on supervised learning, which requires clean speech paired with noisy speech to learn the relationship between them. Since such pairs are generally unknown, noisy speech needs to be generated artificially from clean speech, assuming that they will match the target noisy environment. However, speech enhancement methods relying on clean speech targets have several limitations.

Firstly, the acoustic room simulator requires extensive environment information (i.e., room size distribution, reverberation time, source to microphone distance, and noise type) [34] to convolve the room impulse response and add noise to the clean speech. This information can be estimated from noisy speech; however, this itself is a challenging problem [35, 36].

Secondly, the acoustic model trained on simulated data is often not generalized well in a real environment [37]. This is because simulation may not fully cover the real environment or represent characteristics other than additive noise and reverberation (e.g., Lombard effect [38]). Moreover, these approaches cannot use real noisy speech collected from the target environment, as shown in Figure 3.2.

Thirdly, when enhancement is used as a pre-processing stage for speech recognition, enhancement towards clean speech may not be the optimal approach. Speech recognition requires the phonetic characteristics in the enhanced speech to be preserved while suppressing other non-verbal details. However, yielding enhanced outputs that resemble clean speech is different from this direction.

To avoid the use of clean speech targets, we use reconstruction and adversarial supervision (RAS) and acoustic and adversarial supervision (AAS) for improving speech intelligibility and speech recognition respectively. These learning algorithms are tested on a single-channel additive noise case.

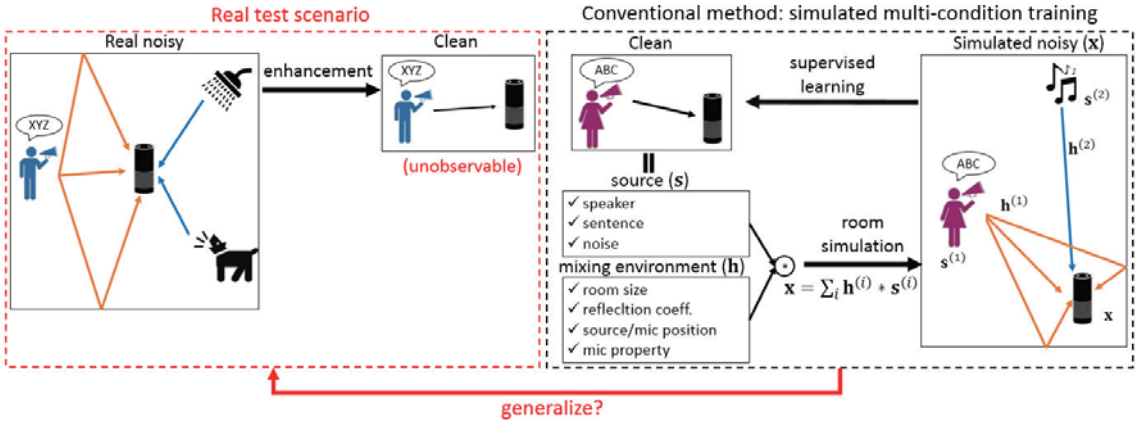


figure 3.2: Generally, clean speech corresponding to noisy speech cannot be obtained in real environment. Therefore, speech enhancement methods using such pairs cannot use data collected from real environment.

3.2 Related work

3.2.1 Environmentally robust ASR: categorization

There are many works for environmentally robust speech recognition. These techniques are divided into front-end (speech enhancement), back-end and joint front and back-end approach as shown in Figure 3.3.

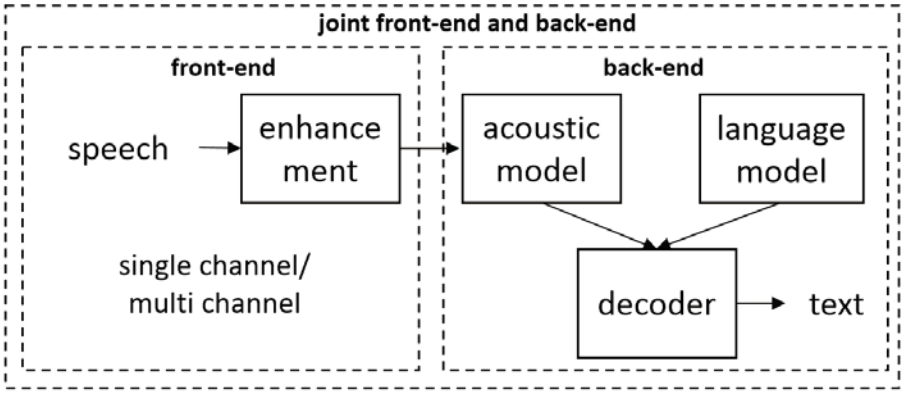


figure 3.3: Categorization of approaches for environmentally robust speech recognition.

The single-channel speech enhancement method can be divided into three categories: signal processing with prior domain knowledge, mapping based method, and masking based method. For signal processing with prior domain knowledge, spectral subtraction combined with voice activity detection [39] is one method. Pure noise spectrogram is estimated from the non-speech part and subtracted from all noisy mixture spectrogram while keeping it as positive. Another method is Wiener filter [40], which assumes clean speech can be estimated by a linear combination of the previous few time steps of mixture signal. Both methods are simple (i.e., no learning) and work well for stationary noise such as fan noise. However, this method does not work for non-stationary noise. The mapping/masking based methods employ supervised learning for making input noisy mixture speech towards clean speech. The examples are denoising autoencoder [41, 42], Convolutional neural network [43], and Recurrent neural network [44, 45]. Masking based methods [46, 47, 48] estimate mask and multiplied with a noisy spectrogram. It assumes the mixing/demixing process as linear (masking) in the frequency domain, and its mask can be estimated from the mixture. Empirically, masking based methods often outperform mapping-based methods. Many papers explain this phenomenon that learning mask is easier than learning speech output since mask normally has less dynamic range than speech. Both mappings based and masking based methods require paired data between noisy and clean speech, which is not available in the real environment.

For the back-end approach, it directly trains speech recognizer while maintaining corrupted speech. The first representative method is multi-condition training (MCT, [49]). MCT uses all the corrupted speech for the training speech recognizer. While the method is simple, it requires a large amount of

table 3.1: The summary of categorization of approaches for environmentally robust ASR.

	Description		Representative works	Limitation
Front-end	single channel	unsupervised	spectral subtraction, wiener filter	stationary noise assumption
		supervised	denoixing autoencoder, CNN, RNN, ideal ratio mask	require (corrupted, clean) paired data
	multi channel	unsupervised	MVDR beamforming, ICA	require direction of arrival iterative optimization for every test data
		supervised	GEV beamforming	require (corrupted, clean) paired data
Back-end	re-train speech recognizer	with clean/corrupted	multi-condition training	require (clean, text) paired data
		with corrupted only	model adaptation	catastrophic forgetting
Joint FE and BE	single channel		multi-task learning of SE, SR	require (corrupted, clean) paired data
	multi channel		ICA + ASR	tested on isolated word recognition

speech to cover all the corrupted environment. Also, it is inefficient when the amount of the domain is large. Another method is to adapt parameters of speech recognizer with data from a new domain [50]. However, this method can easily over-fit to adaptation data and forget knowledge about source domain data. [50] solve this problem by adding an additional single layer before the input of speech recognizer, which is similar to our transformer network.

For the joint front and back-end approach, a straightforward way is to apply the front-end and back-end approach separately. However, the mismatch between the goal of two methods (i.e. speech intelligibility and speech recognition) makes total optimization procedure converge to a suboptimal solution. Another method is joint training of speech enhancement network and speech recognition network, the direction we pursue [51]. These are summarized in Table 3.1.

Our method is categorized into the joint front and back-end approach since we train enhancement model under supervision from acoustic and discriminator models with corresponding objective function: Connectionist Temporal Classification (CTC) and Generative Adversarial Network (GAN). Our method does not require paired data between noisy speech and clean speech, which is naturally unavailable in the real environment. Our method learn speech enhancement module with speech recognition metric (i.e. Word Error Rate (WER)) which is different from conventional speech enhancement metric (i.e. Signal-to-Noise Ratio (SNR)).

3.2.2 Simulated clean as target for enhancement

for robust speech recognition

For robust speech recognition, [52, 53] jointly train both enhancement model and acoustic model. Figure 3.4 show architecture of joint training of enhancement and acoustic model. They employ two loss function: L_{DCE}, L_{AM}

$$\min_{E,A} L_{AM} = -\mathbb{E}_{(\mathbf{x},\mathbf{t})}[\log p(\mathbf{t}|E(\mathbf{x}); A)] \tag{3.2}$$

$$\min_{E} L_{DCE} = -\mathbb{E}_{(\mathbf{x},\mathbf{s})}[d(E(\mathbf{x}), \mathbf{s})] \tag{3.3}$$

where, AM, DCE stands for acoustic model and distance between clean and enhanced speech. This model requires paired data between clean and noisy speech, thereby cannot be applied for real noisy training data.

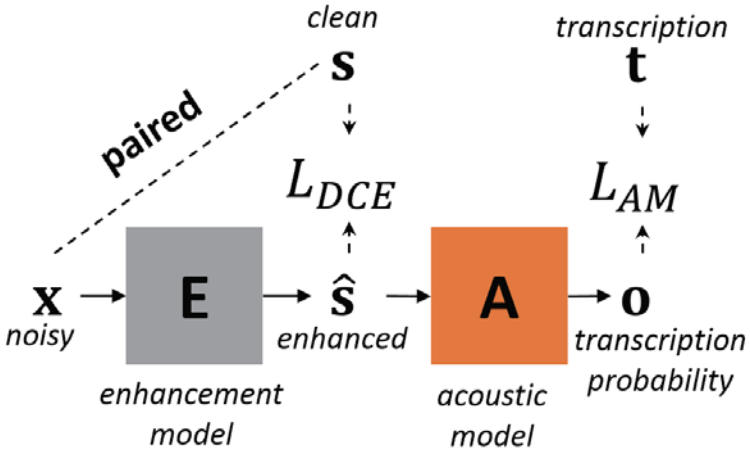


figure 3.4: Illustration of joint training of enhancement model and acoustic model designed for robust speech recognition.

for realistic clean speech

For realistic clean speech (i.e., no artifacts on enhanced speech), [54, 55] employ conditional generative adversarial network. Figure 3.5 show conditionaal generative adversarial network architecture. They employ two loss function: L_{DCE}, V_{cGAN}

$$\min_{E} L_{DCE} = -\mathbb{E}_{(\mathbf{x},\mathbf{s})}[d(E(\mathbf{x}), \mathbf{s})] \tag{3.4}$$

$$\min_{E}\max_{D} V_{cGAN} = \mathbb{E}_{(\mathbf{x},\mathbf{s})}[D(\mathbf{s}, \mathbf{x})] - \mathbb{E}_{\mathbf{x},\mathbf{z}\ N(0,I)}[D(E(\mathbf{x}, \mathbf{z}, \mathbf{x})] \tag{3.5}$$

Speech enhancement is related to domain transfer problems (e.g., image-to-image [56] and voice conversion [57]) where the source and target domains are the noisy and clean recording environments, respectively. The representative work is the frequency speech enhancement generative adversarial network (FSEGAN, [55]) which employs two losses: the distance from the clean to enhanced speech and the

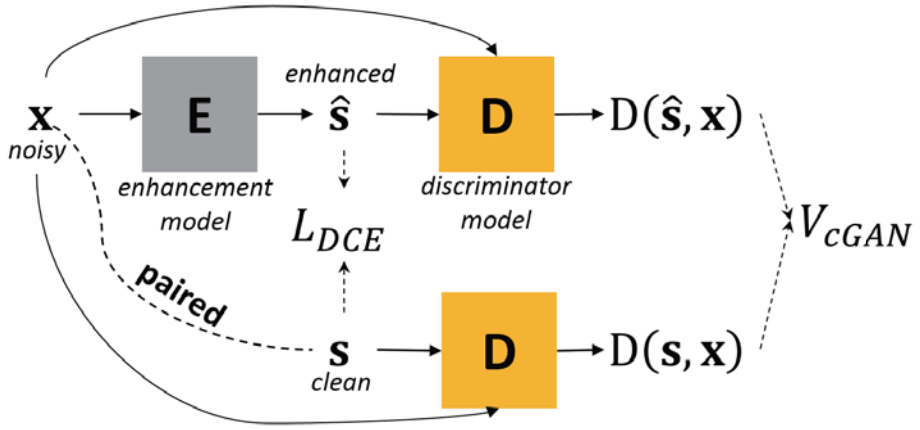


figure 3.5: Illustration of conditional generative adversarial network designed for generating realistic clean speech.

loss function for the conditional generative adversarial network (cGAN, [58]). Given a source domain ($\mathbf{x}_s$) and a target domain ($\mathbf{x}_t$) data, cGAN optimizes the min-max game between a generator (G) and a discriminator (D) with the value function (V) given by

$$\min_G \max_D V_{cGAN}(G,D) = \mathbb{E}_{(\mathbf{x}_s,\mathbf{x}_t)\sim p(\mathbf{x}_s,\mathbf{x}_t)}[\log D(\mathbf{x}_s,\mathbf{x}_t)] + \mathbb{E}_{\mathbf{x}_s\sim p(\mathbf{x}_s),\mathbf{z}\sim N(0,I)}[\log(1-D(\mathbf{x}_s,G(\mathbf{x}_s,\mathbf{z})))]. \tag{3.6}$$

$$\min_G L_{DCE}(G) = \mathbb{E}_{(\mathbf{x}_s,\mathbf{x}_t)\sim p(\mathbf{x}_s,\mathbf{x}_t),\mathbf{z}\sim N(0,I)}[d(G(\mathbf{x}_s,\mathbf{z}),\mathbf{x}_t)] \tag{3.7}$$

Here, G is trained to deceive D, which judges whether a given pair of cross-domain samples come from the real data $(\mathbf{x}_s,\mathbf{x}_t)$ or are generated from the source domain and random noise $\mathbf{z}$ $(\mathbf{x}_s, G(\mathbf{x}_s,\mathbf{z}))$. Two losses of FSEGAN require the paired clean and noisy speeches, not available in the real environment.

Usually, domain transfer problems require unsupervised learning because the paired data between different domains are expensive to be obtained. Therefore, many domain transfer models based on cGAN [56, 59, 60] remove the dependency of the paired source ($\mathbf{x}_s$)in a discriminator and use the unpaired cGAN (upcGAN) whose value function (V) is

$$\min_G \max_D V_{upcGAN}(G,D) = \mathbb{E}_{\mathbf{x}_t\sim p(\mathbf{x}_t)}[\log D(\mathbf{x}_t)] + \mathbb{E}_{\mathbf{x}_s\sim p(\mathbf{x}_s)}[\log(1-D(G(\mathbf{x}_s)))], \tag{3.8}$$

where $\mathbf{z}$ is often omitted to learn deterministic generator.

However, upcGAN can lead the transferred sample $G(\mathbf{x}_s)$ merely having the general characteristics of the target domain, since the discriminator judges the transferred sample without seeing the paired source domain sample. This problem can be alleviated by imposing additional regularization [61] on a generated sample, such as cycle-consistency loss [59, 60] shown in Figure 3.6. However, this loss is not applicable for speech enhancement because the original noisy speech cannot be reconstructed from enhanced speech since there are infinite possible noises to mix. Instead, we encourage the enhanced sample to be recognized correctly by an acoustic model as an alternative regularization.

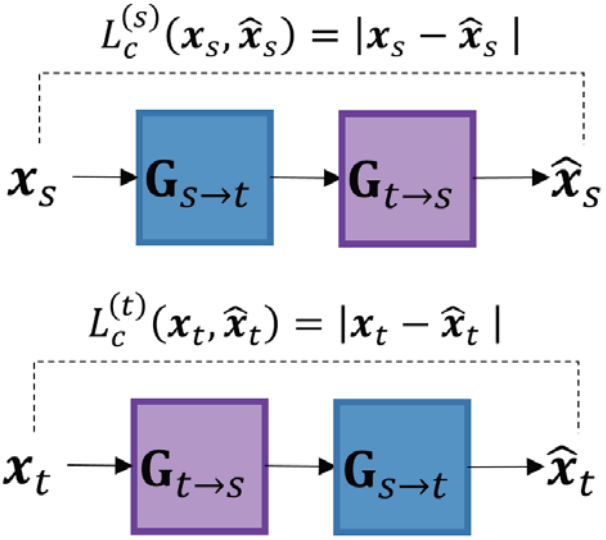


figure 3.6: Illustration of cycle-consistency loss for alleviating mode collapse problem of learning GAN.

3.2.3 Clean-free enhancement

There are a few approaches that do not require simulated clean speech as a target. We pursue this direction since these approaches can use noisy speech collected from the real environment as training data. One method is spectral subtraction shown in Figure 3.7. It estimates time-averaged noise spectrogram from the non-speech interval, and subtract noise spectrogram for whole speech time frames. This approach does not work for non-stationary noise.

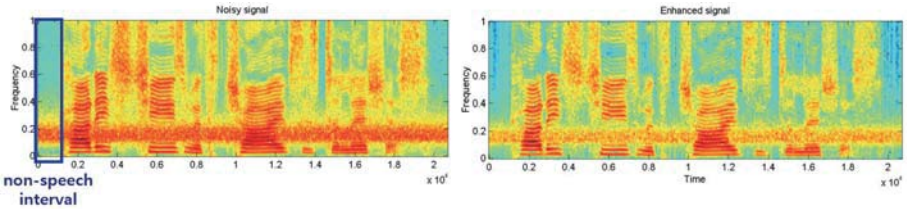


figure 3.7: Illustration of spectral subtraction.

Another method is [62] employing independent component analysis and isolated word classifier. The method is shown in Figure 3.8. It train speech and noise seperator (E) by independent component analysis and fine-tune separated speech with speech classifier. This method is applied for isolated word recognition.

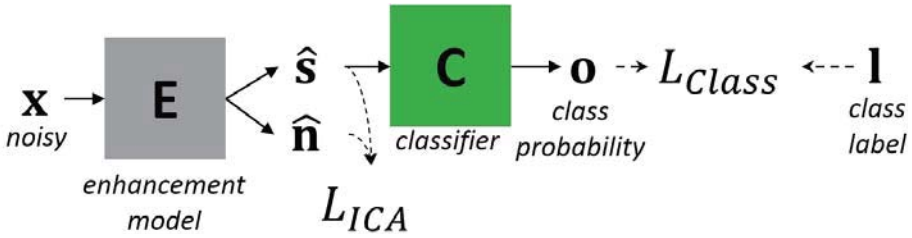


figure 3.8: Combining ICA and speech classifier as one example of clean-free enhancement.

3.2.4 Boundary equilibrium GAN

Boundary Equilibrim GAN (BEGAN [63]) deals with the problem that balancing the convergence of discriminator and generator is challenging during GAN training. Also, it is hard to monitor the progress of learning due to the non-monotonic nature of the learning curve. The solution to this problem is using the boundary equilibrium technique. This enforces maintaining the ratio between expected loss of generated sample, and the expected loss of real sample during the whole training. The modified GAN loss function, based on Proportional Control Theory, is given as follows:

$$L_G = l_D(G(\mathbf{z})) \tag{3.9}$$

$$L_D = l_D(\mathbf{x}) - k_t \cdot l_D(G(\mathbf{z})) \tag{3.10}$$

$$k_{t+1} = k_t + \lambda_k(\gamma l_D(\mathbf{x}) - l_D(G(\mathbf{z})) \tag{3.11}$$

$$l_D(\mathbf{x}) = |\mathbf{x} - D(\mathbf{x})| \tag{3.12}$$

If discriminator overpower generator, which is typical situation in GAN training, $l_D(\mathbf{x})$ is small, $l_D(G(\mathbf{z}))$ is large which makes k_t small, which implies discriminator penalizes generator less. If generator overpower discriminator, then the formula operate in opposite way. As by-product of this technique, convergence measure is available to determine end point of learning:

$$M_{converge} = l_D(\mathbf{x}) + |\gamma l_D(\mathbf{x}) - l_D(G(\mathbf{z}))| \tag{3.13}$$

The first term measure how good enough for discriminator discriminate real data. The second term measure how loss of real data and generated data is balanced.

3.3 Method

This chapter first describes verifying the problem of transforming the data with classification loss only on a toy dataset. After that, the proposed methods are described: reconstruction and adversarial supervision (RAS) and acoustic and adversarial supervision (AAS), designed to improve speech intelligibility and speech recognition performance respectively. The system architecture of RAS and AAS is shown in Figure 3.9.

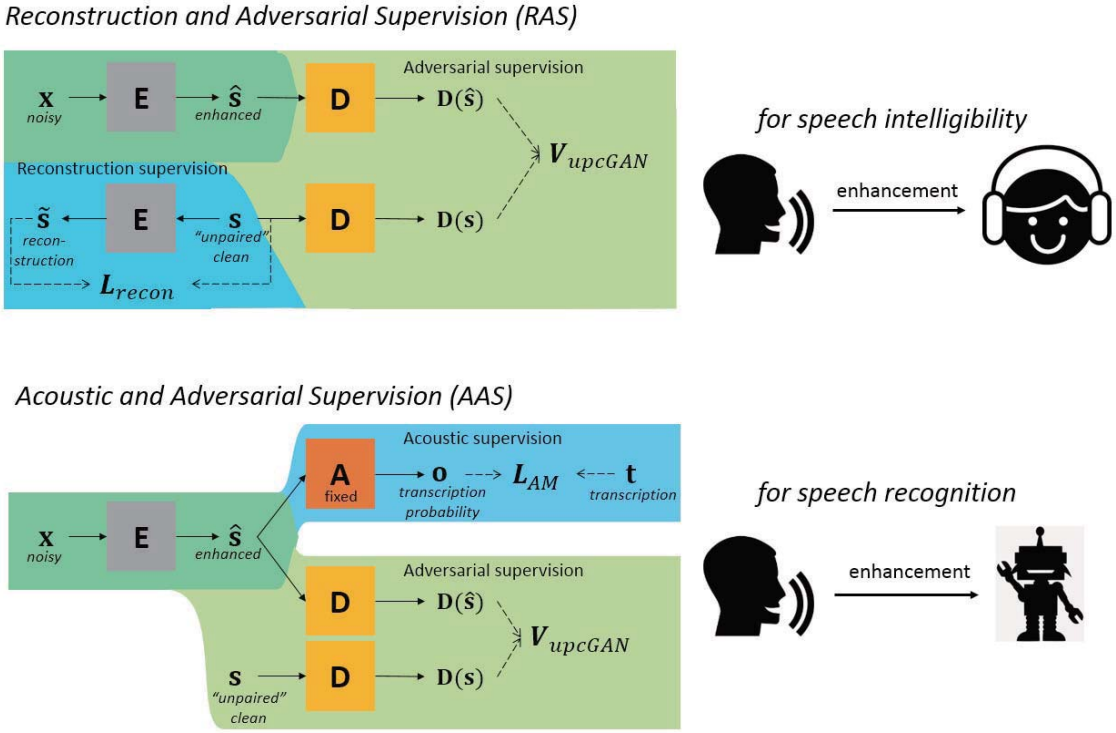


figure 3.9: The system architecture of RAS and AAS. Each learning criteria is designed for improving speech intelligibility and speech recognition.

3.3.1 Reconstruction and adversarial supervision (RAS)

Graphical model

Figure 3.10 and 3.11 shows graphical model and its detail architecture. It is basically encoder-decoder architecture, and designed to work as denoising autoencoder.

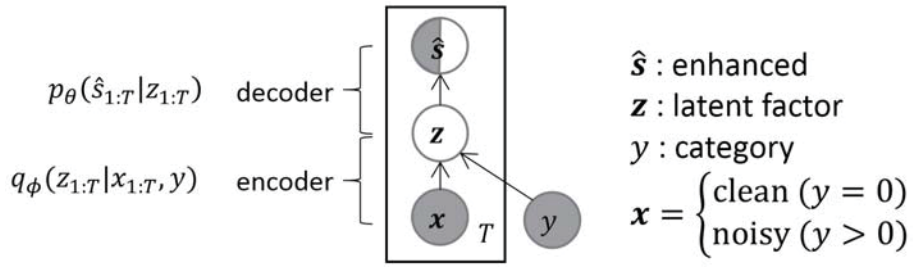


figure 3.10: Graphical model of encoder-decoder architecture.

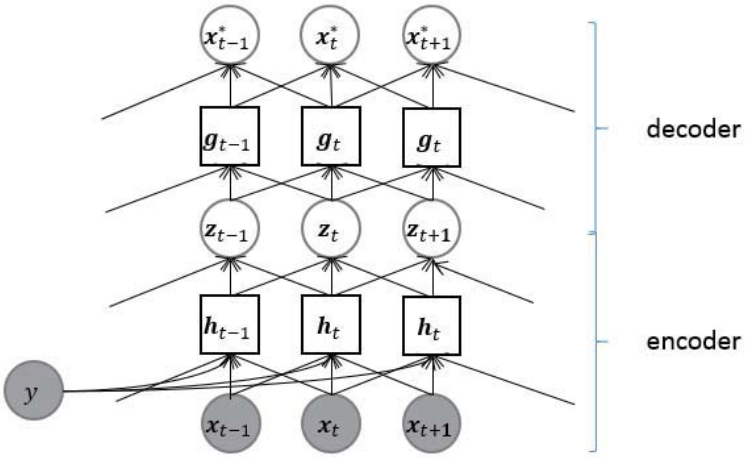


figure 3.11: Detail architecture of CNN encoder-decoder architecture.

Encoder extract characteristic of clean speech. It can be seen as phoneme recognizer. Encoder is based on Convolutional Neural Network to extract local temporal-frequency information of spectral feature. Distribution of encoding latent variable is defined as factored gaussian distribution.

$$q_\phi(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}, y) = \prod_{t=1}^{T} q_\phi(\mathbf{z}_t|\mathbf{x}_{R(t)}, y) \tag{3.14}$$

$$\mathbf{z}_t \sim N(\mu(\mathbf{x}_{R(t)}, y), \sigma(\mathbf{x}_{R(t)}, y)) \tag{3.15}$$

Decoder generate coherent speech from extracted information. Decoder is based on Convolutional Neural Network to generate every frames of spectral feature. Distribution of generated spectral feature is defined by Gaussian Mixture Model.

$$p_\theta(\hat{\mathbf{s}}_{1:T}|\mathbf{z}_{1:T}) = \prod_{t=1}^{T} p_\theta(\hat{\mathbf{s}}_t|\mathbf{z}_{R(t)}) \tag{3.16}$$

Learning algorithm (RAS)

To make the proposed network work as denoising autoencoder, reconstruction and adversarial supervision (RAS) is proposed. The upper part of Figure 3.9 shows RAS learning framework.

The main task is an adversarial enhancement. It takes noisy speech as input. Supervision to output is given by using the concept of Generative Adversarial Network (GAN). In other words, the distribution of output is forced to match the distribution of clean speech. The main task is defined with the following formula:

$$\min_{E} \max_{D} V_{GAN}(E, D) = l_D(\hat{\mathbf{s}}) - l_D(E(\mathbf{x}, y > 0)) \tag{3.17}$$

Discriminator (D) needs to minimize the loss while generator (G) needs to maximize the loss. The main task alone is not enough to learn the detailed structure of clean speech, and often hard to converge to optimal Nash equilibrium due to its adversarial nature.

The auxiliary task is clean sample reconstruction. It has a shared goal with the main task: learn to generate clean speech from the encoded latent variable ($\mathbf{z}$). It takes clean speech as input. Supervision to output is given by two objectives. One is comparing with the clean sample itself. In other words, the network is learned towards reconstructing a clean sample. The objective function is given as follows:

$$L_{recon} = E_{q_\phi(\mathbf{z}_{1:T}|\mathbf{x}_{1:T},y)}[\sum_{t=1}^{T} \log p_\theta(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{z}_{R(t)}, y)] - \sum_{t=1}^{T} KL(q_\phi(\mathbf{z}_t|\mathbf{x}_{R(t)}, y)||p(\mathbf{z}_t)) \tag{3.18}$$

$$\min_{E} \max_{D} V_{adv,recon}(E, D) = l_D(\hat{\mathbf{s}}) - l_D(E(\mathbf{x}_c, y = 0))) \tag{3.19}$$

The RAS is the multi-task learning of reconstruction and adversarial supervision. The objective function of the RAS is given as follows:

$$\min_{E} \max_{D} V_{GAN}(E, D) + L_{recon}(E) \tag{3.20}$$

3.3.2 Acoustic and adversarial supervision (AAS)

While reconstruction and adversarial supervision (RAS) is designed to improve speech intelligibility, it may not be optimal for speech recognition. Wwe propose acoustic and adversarial supervision (AAS) for a speech-enhancement learning algorithm, as shown in lower part of Figure 3.9. The proposed method consists of three models: Enhancement (E), Acoustic (A), and Discriminator (D). For the following description, $\mathbf{x}$, $\mathbf{s}$, $\hat{\mathbf{s}}$, $\mathbf{o}$, and $\mathbf{t}$ are the noisy mixture, (unpaired) clean speech, enhanced speech, grapheme probability, and transcription, respectively. We assume $\mathbf{s}$ and pairs of $(\mathbf{x}, \mathbf{t})$ are available for the training data.

Preliminary : Learning enhancement model with classification loss

For visualizing the weakness of learning transformer with classification loss only, we perform the preliminary experiment on the MNIST dataset. First, pre-train MNIST classifier with architecture given in Figure 3.12. It achieved 99.2% test accuracy. We consider two cases of input for transformer clean MNIST and noisy MNIST (MNIST with white noise ($N(0, I)$). Noisy MNIST achieves 58.2% accuracy when it is feed-in to pre-trained MNIST.

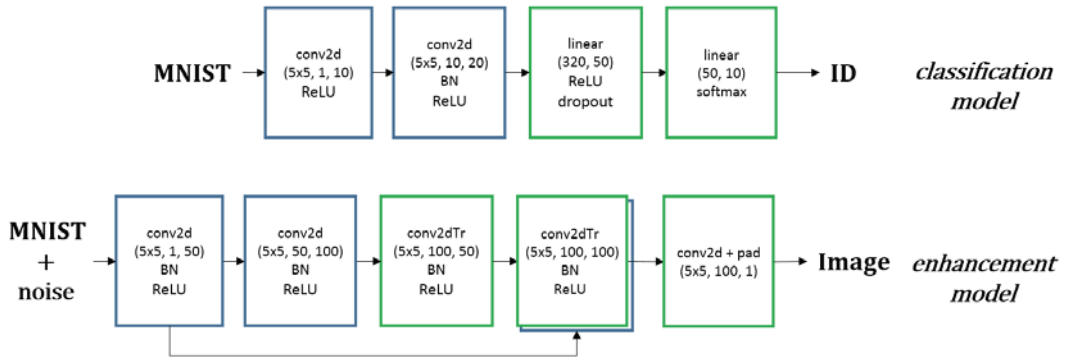


figure 3.12: Architecture of classification model and enhancement model for noisy MNIST enhancement experiment.

One of the possible solution for transformer network is that autoencoder and denoising autoencoder for those cases. We train transformer network for each cases with 10 epochs.

Clean MNIST adaptation achieves 97.0% test accuracy at epoch 10. Noisy MNIST adaptation achieves 99.0% test accuracy at epoch 10. Figure 3.13 shows the input and output of the enhancement network for clean and noisy MNIST case. For the clean MNIST cases, the output of the enhancement network seems almost a faint image but contains the structure of digit. For noisy MNIST cases, the image seems faint and there is almost no recognizable digit structure in the image. From this example, we can conclude that learning to enhance samples towards increasing classification accuracy only does not guarantee an enhanced sample lies in real data manifold. We conjecture that these artifacts are a sign of over-fitting of enhancement model and try to suppress such artifacts to improve generalization.

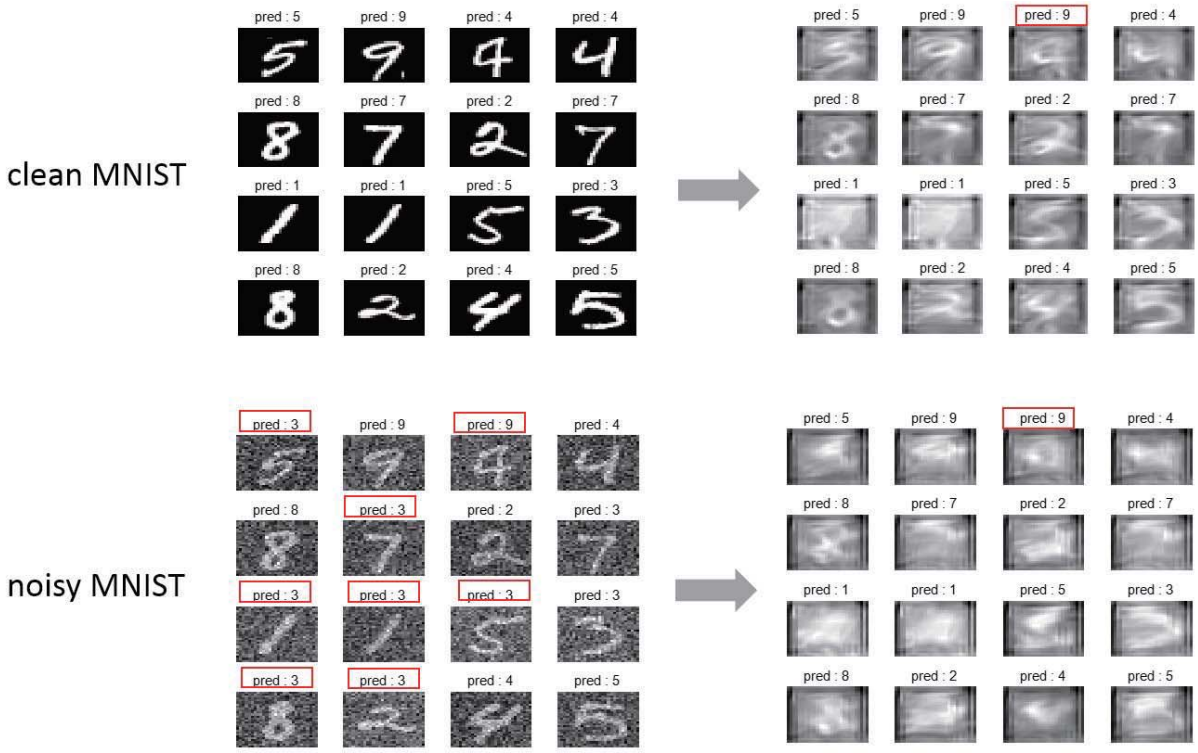


figure 3.13: After enhancing noisy MNIST with only classification loss, input and output is plotted (top: clean MNIST, bottom: noisy MNIST)

Acoustic supervision

Acoustic supervision trains the enhancement model to maximize the likelihood of transcription of the noisy sample. The pre-trained acoustic model (AM) provides the enhancement model with top-down information of the phonetic features essential for correct recognition. This is motivated by the top-down attention mechanism of humans, applied for noise-robust speech recognition [64], and N-best rescoring [65]. Although this supervision does not require a specific type of AM, we employ a neural network with connectionist temporal classification (CTC, [66]). The CTC is used to label a sequence without requiring explicit alignment between the input and label sequences. Moreover, a grapheme is used as the output unit of the neural network, so that AM does not require a lexicon, which allows generating out-of-vocabulary words during inference. The CTC loss function is given by

$$L_{CTC}(E) = -\mathbb{E}_{(\mathbf{x},\mathbf{t})\sim p(\mathbf{x},\mathbf{t})}[\log p(\mathbf{t}|E(\mathbf{x}))], \tag{3.21}$$

$$p(\mathbf{t}|E(\mathbf{m})) = \sum_{\pi \in Align(E(\mathbf{m}),\tilde{\mathbf{t}})} \prod_{f} o_f^{\pi}, \tag{3.22}$$

where $\tilde{\mathbf{t}}$ is a sequence with CTC-blank added between every pair of graphemes in $\mathbf{t}$, the beginning, and the end. The likelihood of $\mathbf{t}$ given $E(\mathbf{x})$ is defined as sum of single path likelihoods across all possible alignments ($Align(E(\mathbf{x}), \tilde{\mathbf{t}})$).

Note that CTC introduce additional blank symbol to deal with spectral frame which does not align

with any character as well as multiple same character in a raw (i.e. two 'l's in hello). This blank is inserted at start/end of character sequence as well as between characters. This augmented character sequence is used as target sequence. Figure 3.14 shows trellis example of matching spectral feature and the character sequence 'CAT'.

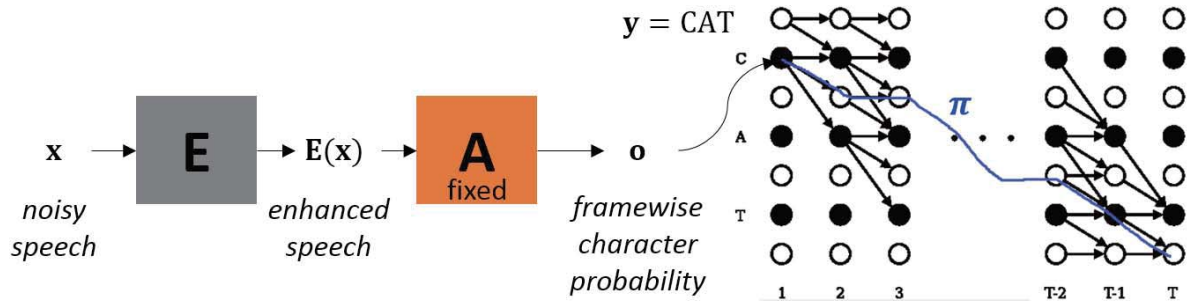


figure 3.14: Illustration of Connectionist Temporal Classification for speech recognition.

Computing CTC loss sums log-probability over all alignment, which requires O(C^T) computation. However, this computation can be efficiently divided by the dynamic programming method, which is called the forward-backward algorithm. The main idea for this approach is that if two alignments meet at the same output at the same time step, we can merge them.

Adversarial supervision

Adversarial supervision encourages the enhanced speech to have the characteristics of clean speech. We employ upcGAN shown in Figure 3.15. The training convergence of upcGAN is improved further by leveraging the techniques of boundary equilibrium GAN (BEGAN, [63]).

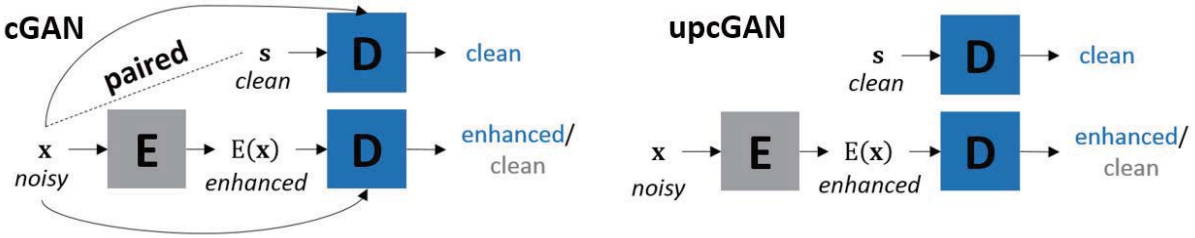


figure 3.15: Comparison of conditional GAN (cGAN) and unpaired conditional GAN (upcGAN).

Firstly, the discriminator (D) auto-encodes the inputs ($l_D(\mathbf{x}) = |\mathbf{x} - D(\mathbf{x})|$) instead of using binary logistic prediction to enhance training efficiency by providing diverse directions of the gradients within the minibatch [67]. Secondly, to balance the power of the discriminator (D) and the enhancement (E) model, the importance of loss on the clean sample ($\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}[l_D(\mathbf{s})]$) is controlled by the proportional control theory [63] given by formula (6). This control helps to maintain the ratio of loss between clean and enhanced data as the pre-defined constant ($\gamma \in [0, 1]$): $\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[l_D(E(\mathbf{x}))] / \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}[l_D(\mathbf{s})] = \gamma$. The

final value function for D and E is given by

$$\min_E \max_D V_{upcBEGAN}(E, D) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[l_D(E(\mathbf{x}))] - 1/(k_t + \epsilon)\, \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}[l_D(\mathbf{s})], \tag{3.23}$$

$$k_{t+1} = k_t + \lambda(\gamma\, \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}[l_D(\mathbf{s})] - \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[l_D(E(\mathbf{x}))]), \tag{3.24}$$

where $k_t \in [0, 1], k_0 = 0, \epsilon = 10^{-8}$.

Multi-task learning

An enhancement model trained using acoustic supervision directly increases the likelihood of transcription on the AM. However, such a model is not unique and depends on the initialization of model parameters and training data. Due to the non-uniqueness, the enhanced output is not guaranteed to converge towards natural speech and often includes artifacts. Moreover, the optimal parameters differ depending on training data, which may not generalize well on an unseen data.

To constrain the solution, we employ the adversarial supervision as an auxiliary task. The adversarial supervision regularizes the enhanced speech having fewer artifacts, leading to the improved generalization on unseen data.

Both losses are combined with weight w_{AC} and w_{AD} as

$$\min_E \max_D w_{AC} L_{CTC}(E) + w_{AD} V_{upcBEGAN}(E, D). \tag{3.25}$$

3.4 Experiment

3.4.1 Dataset

The dataset used in the paper is listed in table 3.2.

table 3.2: The summary of the database used in experiment.

Name	Abbreviation	Generation process	Additive noise		Reverberation	Size [h] (tr/te)
			#type (tr/te)	SNR [dB] (tr/te)		
Librispeech +DEMAND	S960	Simulated	10/5	{15, 10, 5, 0}/ {17.5, 12.5, 7.5, 2.5}	No	960/10
CHiME4 (single-mic)	S15	Simulated	4/4	5	Yes ($RT_{60} = 88ms$)	15/4
	R3	Real	4/4	N/A	Yes	3/1

The RAS is tested on *Voicebank + DEMAND* dataset. Clean speech comes from 13 h of VoiceBank [68], and babble noise come from 1 h of DEMAND [69]. For training and testing, there are 28 and 2 disjoint speakers. Mixture has 4 SNR levels with equal amounts : 15, 10, 5, 0.

The AAS is tested on *Librispeech + DEMAND*, and *CHiME-4* dataset. Librispeech + DEMAND [69] is a large-scale simulated dataset for evaluating enhancement for additive noise. For the training and validation data, 10 types of noise with SNR = {15, 10, 5, 0} are mixed. For the test data, 5 types of unseen noise with SNR = {17.5, 12.5, 7.5, 2.5} are mixed. The noise type, interval, and SNR are randomly selected for each clean utterance. We generate the simulated noisy speech as much as the clean Librispeech (i.e., 960, 10, and 10 h for training, validation, and test, respectively).

CHiME-4 [70] provides read speech recorded from noisy environments with a 6-channel tablet microphone. It includes speech with additive noise (4 types) and reverberation. It provides 15, 3, 6, and 5 h of speech for simulated training, real training, validation, and test set, respectively. The acoustic room simulator [71] is used to generate multi-channel simulated training data that convolve single-channel clean speech with 88 ms impulse response estimated from 65 recordings of tablet microphones, and add 4 types of background noise. During training, the multi-channel data is sampled randomly to make the enhancement model robust to slight changes in mic position [72, 73]. Among the 6 channels, we report the WER of the 5th channel in the test data.

3.4.2 Settings

In all experiments, all the parameters of the neural network are randomly initialized with the distribution $N(0, 0.1^2)$. Adam optimizer [74] with learning rate 10^{-5} and minibatch size 30 is used for training the model. The performance on the test data is reported when the word error rate (WER) on the validation data is the minimum out of 100 epochs. We use $\gamma = 0.5, \lambda = 0.001$ for optimizing the $V_{upcBEGAN}$.

For the language model (LM), 4-gram trained with the Librispeech text corpus is used.[1] 100-best hypotheses, obtained by beam search on the acoustic model (AM), are rescored by combining AM score and length normalized word-level LM score [75] given by

$$S = \log p_{AM}(\mathbf{y}|\mathbf{x}) + \alpha \log(p_{LM}(\mathbf{y})/|\mathbf{y}|^{\beta}). \tag{3.26}$$

Generator generates normalized spectrogram. For qualitative analysis, we convert linear spectrogram to waveform by inverse Fourier transform and overlap-and-sum method. This procedure requires clean speech's phase information, which is estimated by Griffin-Lim reconstruction algorithm [76].

The enhancement model is pre-trained as auto-encoder so that training can start from model generating noisy speech. Without it, acoustic supervision maintains high loss value. We conjecture that noisy speech is a better starting point than random speech since noisy speech and clean speech have many structural similarities and training needs to learn the only difference between them.

The speech feature is selected as a single-channel log-mel filterbank output (LMFB). A single-channel mic signal is generally easily obtainable compared to the multi-mic signal. Also, log-mel filterbank output is widely used in many speech recognition systems. The example of LMFB output is shown in Figure 3.16.

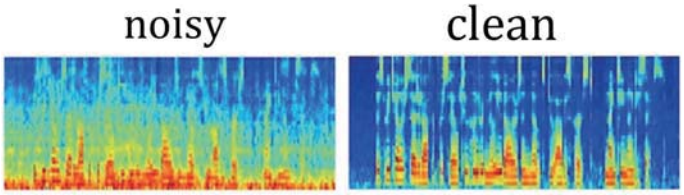


figure 3.16: Examples of log-mel filterbank output feature.

[1] The resources are available in `http://www.openslr.org/11/`.

Detailed architecture

Figure 3.17 shows the detailed architecture setting used for testing RAS and AAS respectively.

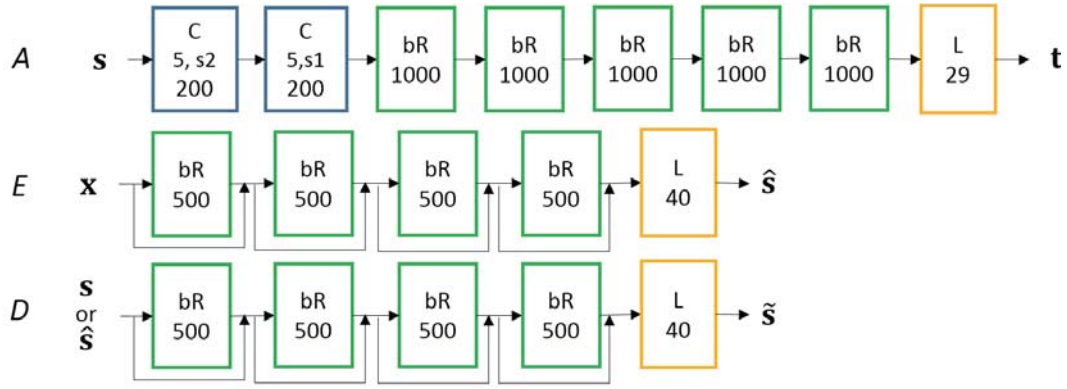


figure 3.17: Detailed enhancement model architecture (acoustic (A), enhancement (E), and discriminator (D) model). Each box describes the layer type (C: 1D convolutional, bR: bidirectional LSTM-RNN, L: linear) and the kernel size (width, stride, #map) for C, #unit for bR and L.

The speech feature is chosen as LMFB features. The architecture of A is based on a stack of convolutional and long short-term memory (LSTM) recurrent layers. Each convolutional layer is followed by batch normalization and rectified linear unit nonlinearity. Each recurrent layer is followed by a sequence-wise batch normalization layer [77].

Both E and D are multi-layer bidirectional LSTM-RNNs, whose input and output are LMFB features. Moreover, they have a residual connection between the input and output of each layer for better convergence [78].

3.4.3 Comparable loss functions

As the single channel speech enhancement baseline, we evaluate the Wiener filter method [79], with smoothing factor $\beta = 0.98$. For methods relying on clean speech target, we evaluate the method minimizing the L1 distance between clean and enhanced LMFB feature (DCE), and FSEGAN [55] described in Section 3.2.

The optimal hyperparameters (i.e., the number of hidden layers and neurons of the models, (α, β)) were selected based on yielding the minimum WER on validation data, under the DCE loss function. Selected hyperparameters and architecture of E are the same across all of the comparable loss functions.

3.4.4 Pre-trained speech recognizer

For the pre-training of the speech recognizer, we utilize the Librispeech dataset [80]. It consists of 960 h of reading speech as a training set. This corpus is recorded from various conditions by volunteers. It is pre-processed by noise removal and volume normalization. It provides two kinds of test conditions: test-clean and test-other. They are divided by difficulty of the dataset which is measured by WER of the pre-trained speech recognizer.

Table 3.3 shows the performance of speech recognizer by two inference methods: acoustic model only (greedy search) and acoustic model combined with a word-level language model. For the language model, we utilize 4-gram trained from the Librispeech corpus [81], and implemented by kenLM[82]. Beam search with width 300 is employed during inference.

table 3.3: ASR performance of pretrained speech recognizer, tested on librispeech benchmark corpus. Performance is compared between two different inference mode : acoustic model only and acoustic model combined with language model.

Inference	Dataset	WER [%]	CER [%]
AM only (Greedy)	train	4.64	1.08
	valid	24.53	8.62
	test-clean	14.46	4.27
	test-other	37.32	14.09
AM + LM (4-gram + Beam search)	train	1.62	0.44
	valid	15.73	6.90
	test-clean	5.71	3.20
	test-other	16.9	7.86

Word Error Rate (WER) has a significant difference between 'AM only' and 'AM with LM', however, Character Error Rate (CER) has less difference between them. It is because, the acoustic model often makes mistake for a minor portion of character to produce misspelled word, however, word error rate counts this misspelled word as incorrect thus makes WER inferior.

Table 3.4 shows state-of-the-art performance for Librispeech benchmark [83]. For test-clean, our

table 3.4: State-of-the-art performance for speech recognition on Librispeech corpus

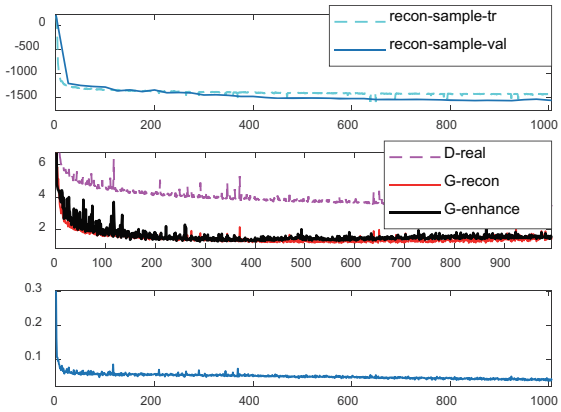
WER (clean)	WER (other)	Computational model	Data	Output unit
4.8	14.5	Gated CNN-CTCvariant + Weight normalization	960hr	grapheme
5.33	13.25	CNN-BiRNN-CTC	11940hr	grapheme
5.51	13.97	HMM-DNN + pNorm	960hr	triphone
8.01	22.49	HMM-(SAT)GMM	960hr	triphone

pre-trained speech recognizer achieves WER 5.71%. The current 1st place achieves WER 4.8% by gated CNN-CTC variant-weight normalization method [84]. The architecture the same as ours utilizes CNN-BiRNN-CTC, and originally proposed from [75]. However, [75] augment noisy speech by mixing noise samples with the Librispeech corpus and utilize a total of 11940 hours of speech as a training set. It achieves WER 5.33%.

3.4.5 Results: RAS

Loss function

Figure 3.18 shows loss function. The first row shows sample-level reconstruction. The second row shows an adversarial loss. It shows three kinds of terms. The first term implies reconstruction loss of real data. The second term implies reconstruction loss of generated speech from clean speech. The third term implies reconstruction loss of generated speech from noisy speech. The third term is directly related to the main task, while the second term corresponds to the auxiliary task.



$$L_{rec,sample} = E_{q_\phi(\mathbf{z}_{1:T}|\hat{\mathbf{x}}_{1:T},y)}[\sum_{t=1}^{T} \log p_\theta(\hat{\mathbf{x}}_t|\mathbf{z}_{R(t)})]$$

$$L_{real} = l_D(\hat{\mathbf{X}})$$
$$L_{recon} = l_D(G(\mathbf{X}_c, y = 0))$$
$$L_{enhance} = l_D(G(\mathbf{X}_m, y > 0))$$

$$M_{converge} = l_D(\hat{\mathbf{X}}) + |\gamma l_D(\hat{\mathbf{X}}) - l_D(G(\mathbf{X}, y))|$$

figure 3.18: Loss function of RAS algorithm.

Case analysis

Table 3.5 compare SNR of enhanced speech trained by the RAS and supervised learning (i.e., using clean speech as target). Improvement rate is higher whenever SNR off original mixture is low.

table 3.5: Enhanced SNR result by the RAS algorithm and supervised learning.

mixture SNR	RAS		supervised	
	train	test	train	test
15	5.35	5.22	23.83	12.21
10	4.78	3.72	23.73	11.02
5	4.33	4.02	24.08	8.83
0	1.85	2.02	23.74	6.45

Without clean target speech, the RAS achieve lower SNR than supervised learning.

Figure 3.20 shows generated results where its SNR has the highest and lowest for each different mixture SNR cases (0 dB). In this figure, firstly we can see that many noises in the non-voice part is filtered out. The web-based audio sample is provided in `https://github.com/gmkim90/speech_transf_disco`.

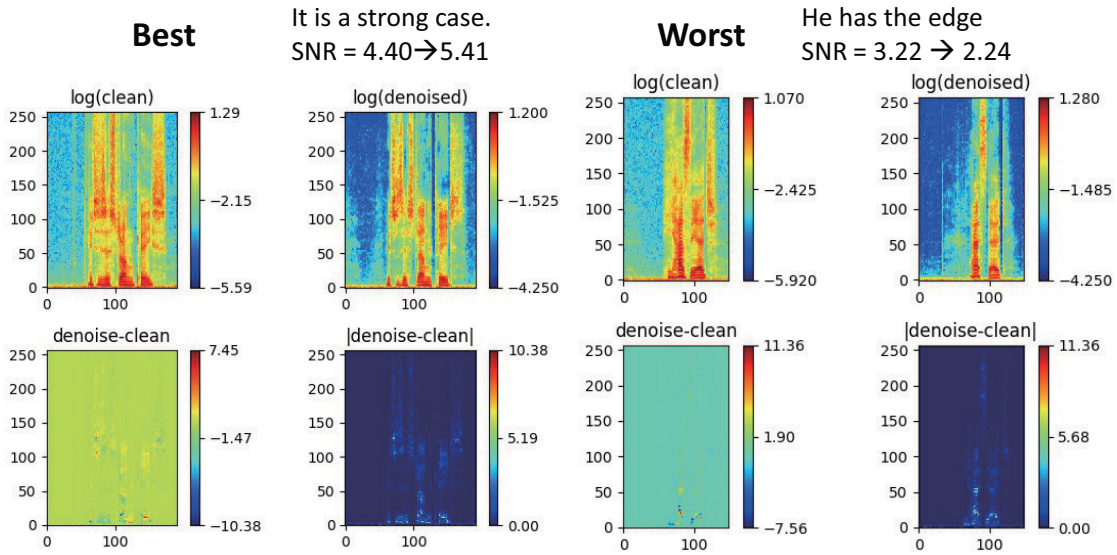


figure 3.19: Generated sample (SNR of mixture = 5).

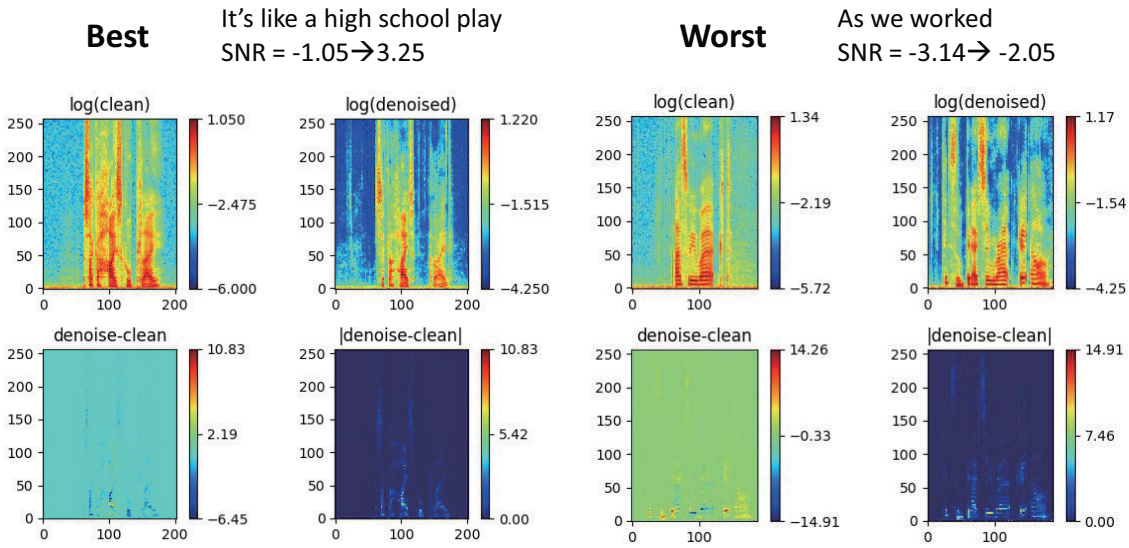


figure 3.20: Generated sample (SNR of mixture = 0).

Difference across frequency

In this section, we compare SNR across differenct frequency. Figure 3.22 compares denoised signal and clean signal across 20 different frequencies in low range frequency when average mixture SNR is 5. Most power difference comes from low frequency.

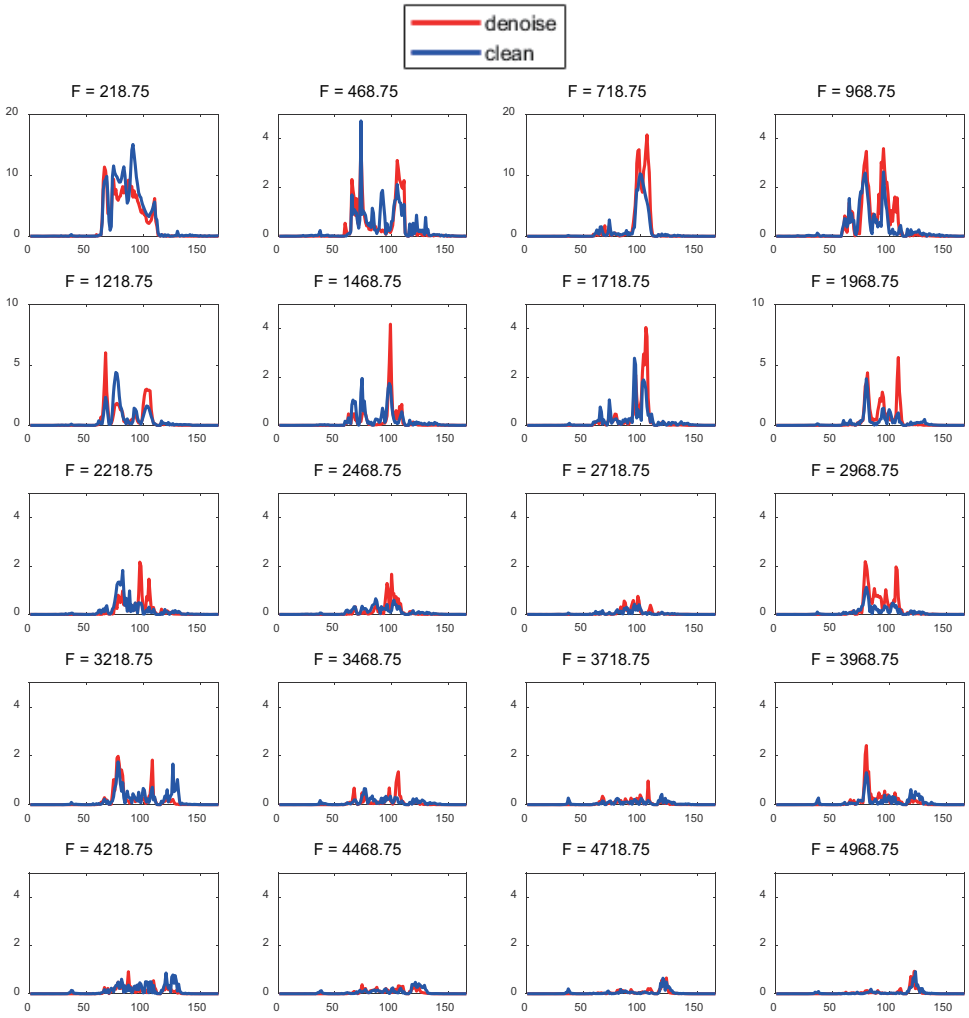


figure 3.21: Difference of sample across different frequencies (SNR of mixture = 15).

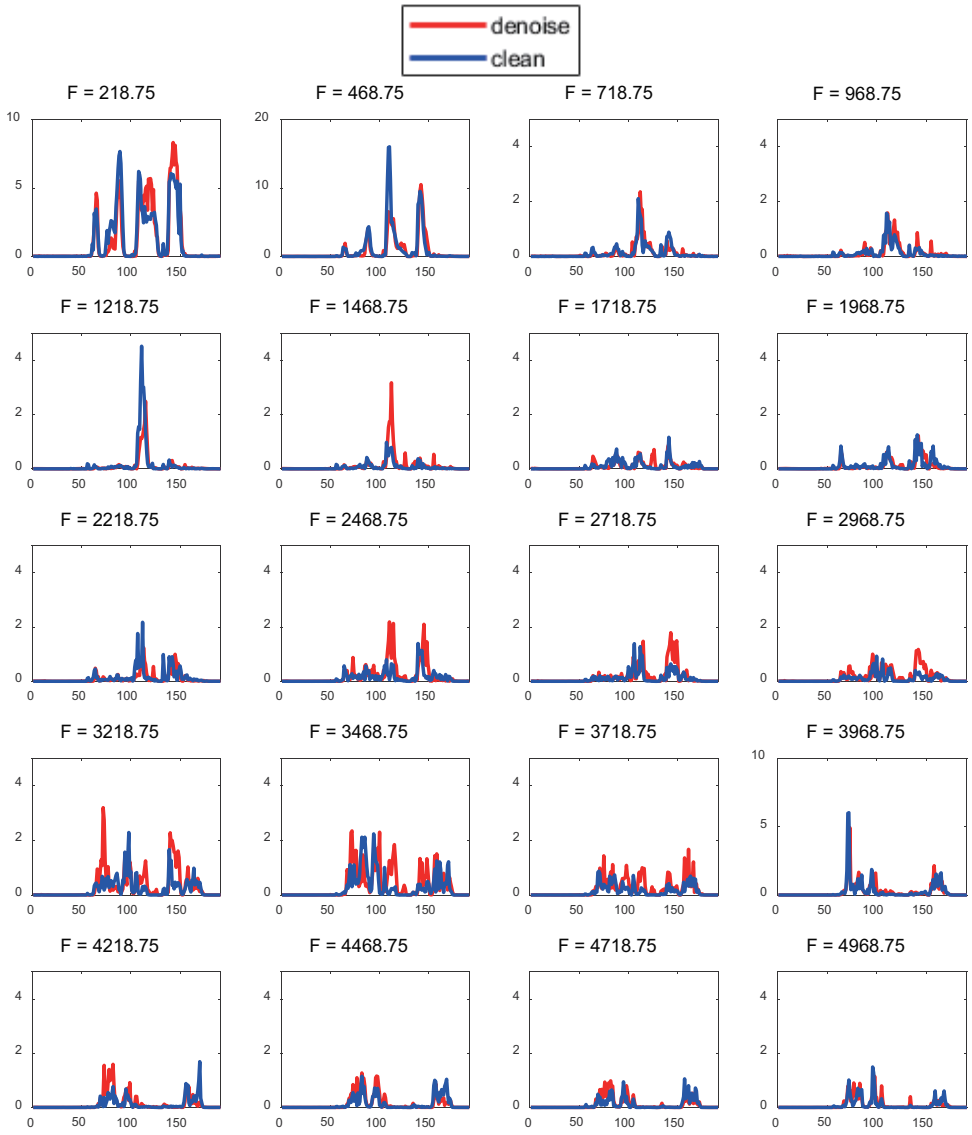


figure 3.22: Difference of sample across different frequencies (SNR of mixture = 5).

3.4.6 Results: AAS

Enhanced feature obtained with different loss functions

Fig. 3.23 shows the LMFB features of noisy, paired clean, and enhanced speech obtained using different loss combinations on the simulated test sets. The enhanced feature obtained using the acoustic supervision ($w_{AC} = 1, w_{AD} = 0$) contains the characteristic of voice (e.g., harmonics) in the noisy mixture, and artifacts (e.g., the horizontal line for a few frequencies). Compared to acoustic supervision, the enhanced feature obtained using adversarial supervision ($w_{AC} = 0, w_{AD} = 1$) shows fewer artifacts but has less voice characteristic at low frequency. The multi-task learning of AAS ($w_{AC} = 1, w_{AD} = 10^5$)

maintains voice characteristics in the generated samples while suppressing the artifacts. This tendency is consistently observed in both noisy datasets.

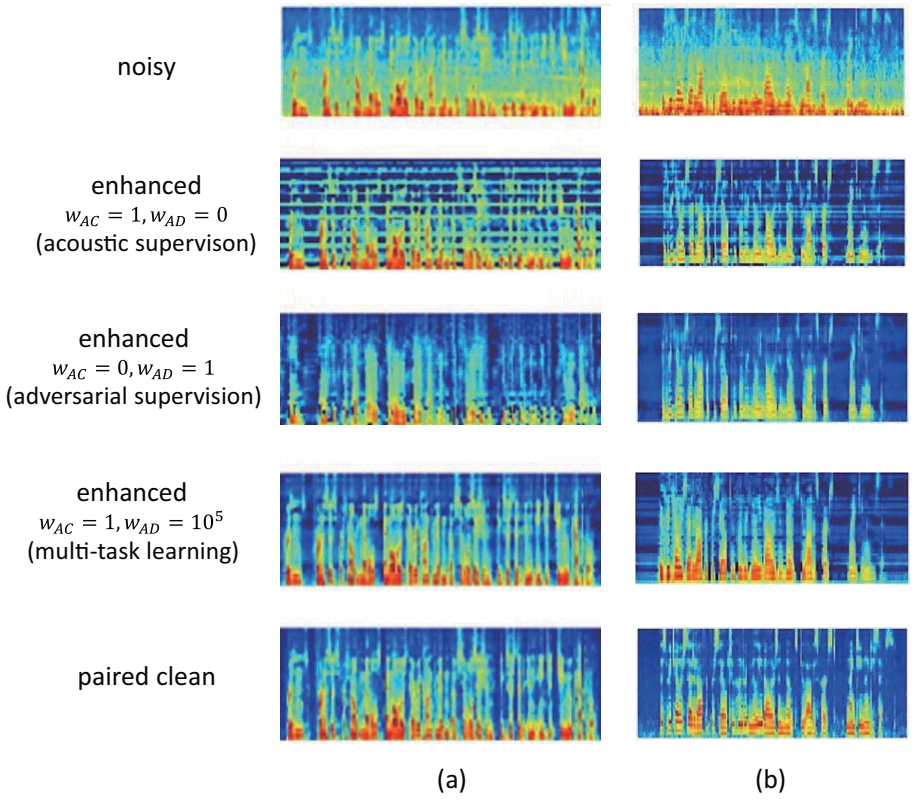


figure 3.23: Enhanced test LMFB features obtained using different task combination. (a) Metro noise with SNR=5 in Librispeech+DEMAND. (b) Bus noise with reverberation in CHiME-4

WERs and distance between clean and enhanced feature

Fig. 3.24 compares the WERs obtained using values of $w_{AD} \in \{0, 10^4, 10^5, 10^{5.25}, 10^{5.5}, 10^{5.75}, 10^6, 10^7\}$ given $w_{AC} = 1$. On both datasets, the lowest WER on the validation data is observed when w_{AD} is between 10^5 to 10^6 and starts to increase at some point.

Figure 3.25 show learning curve to analyze different behavior of loss and gradient as w_{AD} varies over different range. The first row comparing the gradient norm of each supervision at different task weights. And the second row is the corresponding losses of each supervision. At $w_{AD} = 10^5$, the gradient norm of adversarial supervision is the most similar to that of acoustic supervision, indicating that the effects of both supervisions are balanced. At $w_{AD} = 10^6$, the gradient norm of adversarial supervision exceeds that of acoustic supervision, degrading the acoustic model loss (i.e., negative log-likelihood of transcription) which is closely related to the WER. This analysis implies that we can estimate a proper range of task weight by balancing the gradient norm of each task.

Table 3.6 and 3.7 show the WER and DCE (normalized by the number of frames) on the test set of Librispeech + DEMAND, and CHiME-4. The Wiener filtering method shows lower DCE, but higher WER than no enhancement. We conjecture that a lower DCE comes from removal of noise,

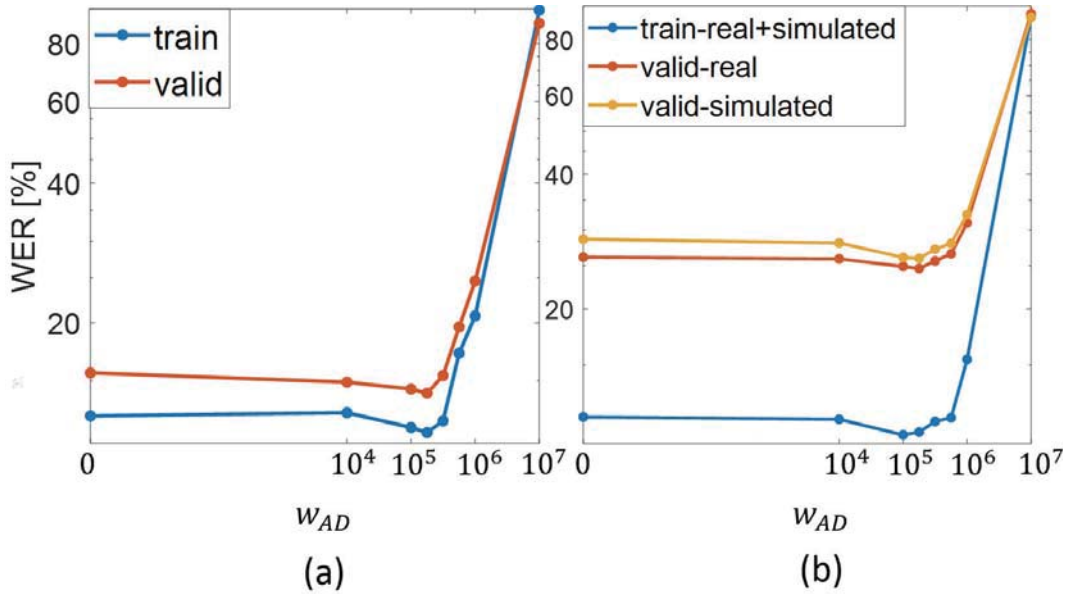


figure 3.24: WER with varying loss weight for adversarial supervision (a) on Librispeech + DEMAND, and (b) on CHiME-4

and higher WER comes from speech distortion. The detailed visualization is given in the next section. The adversarial supervision (i.e., $w_{AC} = 0, w_{AD} > 0$) consistently shows very high WER (i.e., $> 90\%$), because the enhanced sample tends to have less correlation with noisy speech, as shown in Fig. 3.23.

In Librispeech + DEMAND, acoustic supervision (15.6%) and multi-task learning (14.4%) achieves a lower WER than minimizing DCE (15.8%) and FSEGAN (14.9%). The same tendency is observed in CHiME-4 (i.e. acoustic supervision (27.7%) and multi-task learning (26.1%) show lower WER than minimizing DCE (31.1%) and FSEGAN (29.1%)).

Because the AM is trained on Librispeech, reducing DCE is directly related to lowering the WER in Librispeech+DEMAND, but does not ensure lowering of the WER in CHiME-4. This explains the slight WER difference between AAS and FSEGAN in Librispeech+DEMAND and the large difference in CHiME-4.

Table 3.8 shows the WERs on the simulated and real test sets when AAS is trained with different training data. With the simulated dataset as the training data, FSEGAN (29.6%) does not generalize well compared to AAS (25.2%) in terms of WER. With the real dataset as the training data, AAS shows severe overfitting since the size of training data is small. When AAS is trained with simulated and real datasets, it achieves the best result (24.7%) on the real test set.

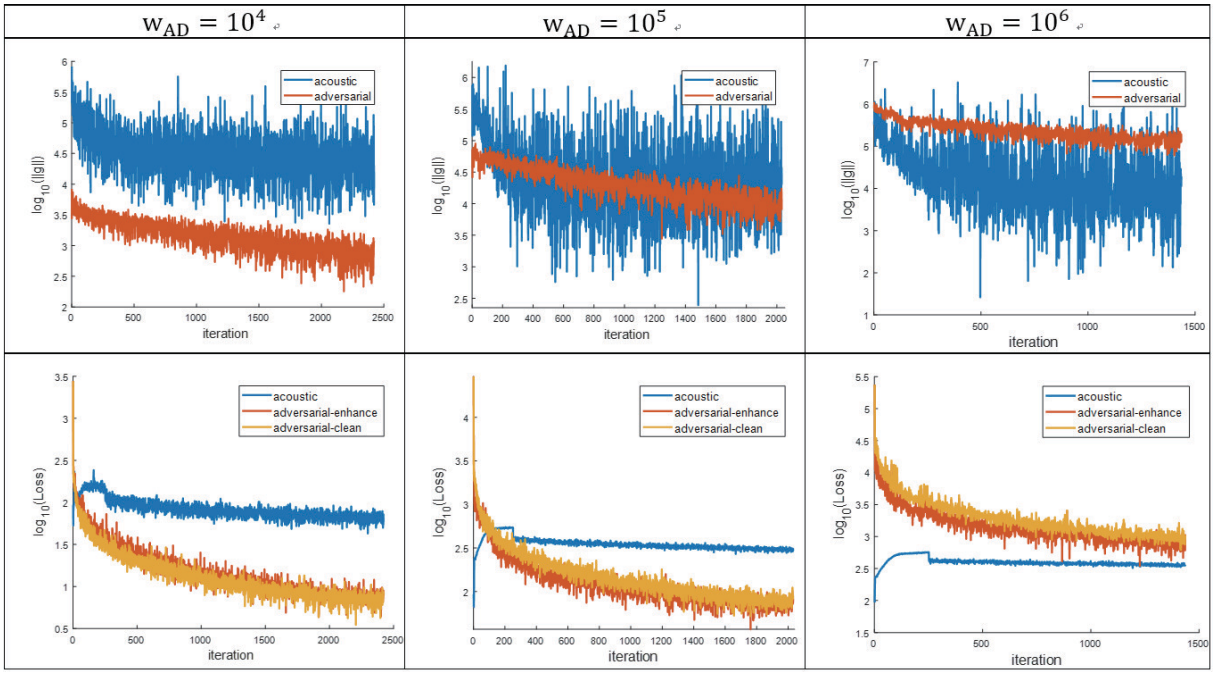


figure 3.25: The gradient and learning curve with different task weight of adversarial supervision.

table 3.6: WERs (%) and DCE of different speech enhancement methods on Librispeech + DEMAND test set

Method	WER (%)	DCE
No enhancement	17.3	0.828
Wiener filter	19.5	0.722
Minimizing DCE	15.8	**0.269**
FSEGAN	14.9	0.291
AAS ($w_{AC} = 1, w_{AD} = 0$)	15.6	0.330
AAS ($w_{AC} = 1, w_{AD} = 10^5$)	**14.4**	0.303
Clean speech	5.7	0.0

table 3.7: WERs (%) and DCE of different speech enhancement methods on CHiME4-simulated test set

Method	WER (%)	DCE
No enhancement	38.4	0.958
Wiener filter	41.0	0.775
Minimizing DCE	31.1	**0.392**
FSEGAN	29.1	0.421
AAS ($w_{AC} = 1, w_{AD} = 0$)	27.7	0.476
AAS ($w_{AC} = 1, w_{AD} = 10^5$)	**26.1**	0.462
Clean speech	9.3	0.0

table 3.8: WERs (%) of obtained using different training data of CHiME-4

Loss	Training data	Test WER	
		simulated	real
AM + 10^5upcGAN (AAS)	S15	26.1	25.2
	R3	37.3	35.2
	SR18	25.9	24.7
DCE +10^{-2}cGAN (FSEGAN)	S15	29.1	29.6

Comparison with the Wiener filter

Figure 3.26 and 3.27 show randomly selected examples of enhanced speech by the Wiener filter.

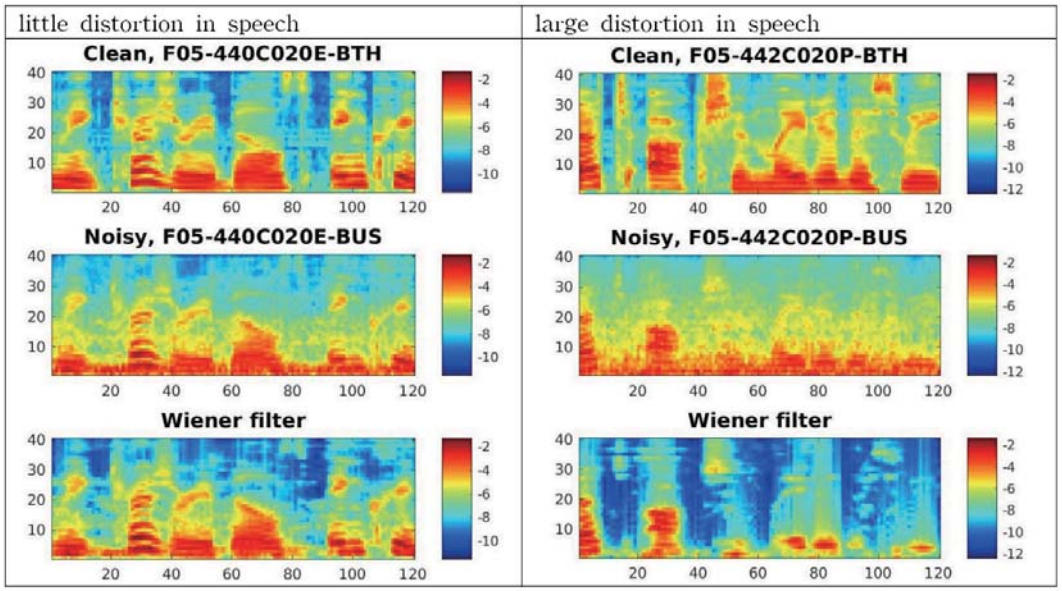


figure 3.26: Examples of enhanced speech by the Wiener filter (case = bus noise).

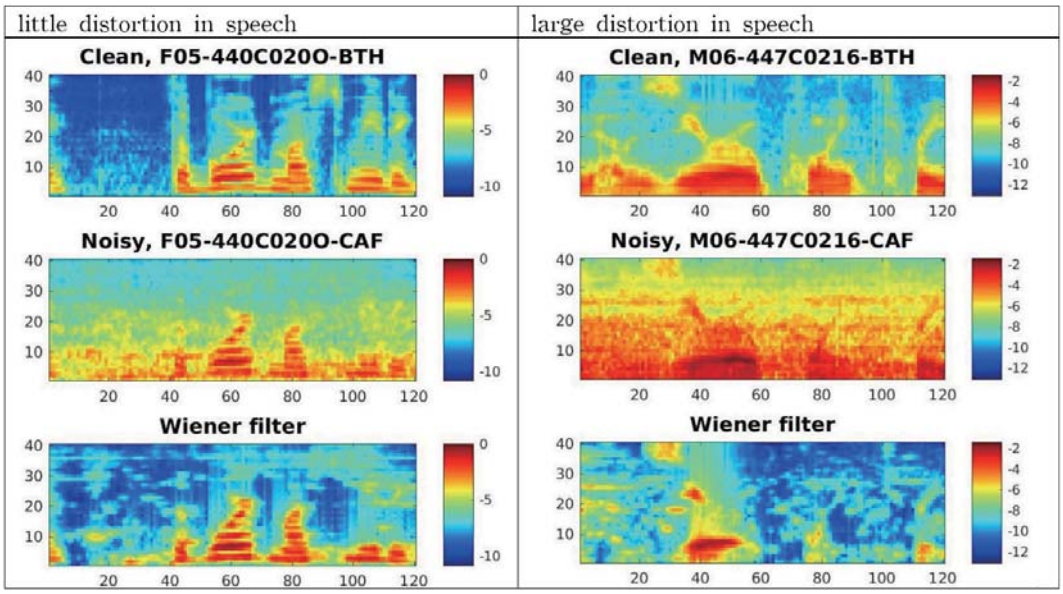


figure 3.27: Examples of enhanced speech by the Wiener filter (case = cafe noise).

Compared to no enhancement baseline, Wiener filter based enhancement [40] show lower DCE, but higher WER. Samples generated with the Wiener filter method seems to generate the speech with reduced background noise, but remaining speech is distorted as well. We conjecture that distortion in speech may degrade WER of the pre-trained speech recognizer.

3.5 Conclusion

Speech enhancement models with clean speech as the target have several limitations. First, clean speech is generally not obtainable in the real environment. Second, it is optimal for maximizing signal-to-distortion ratio but suboptimal for minimizing the word error rate. To avoid relying on clean speech targets, we propose the training speech enhancement model with multi-task learning. Reconstruction and adversarial supervision (RAS) and acoustic and adversarial supervision (AAS), designed to improve speech intelligibility and speech recognition performance respectively. Each supervision maximizes the likelihood of transcription on the pre-trained acoustic model and ensures general characteristics of clean speech in the enhanced output, which improves generalization on unseen noisy speech. The proposed method was tested on two datasets: Librispeech + DEMAND and CHiME-4. By visualizing the enhanced feature, we demonstrated the role of each supervision. AAS showed a lower word error rate compared to speech enhancement methods using a clean target. The proposed AAS can be combined with any acoustic model of a given clean speech and noisy speech with transcription.

Chapter 4. Source/Position robust speech enhancement

4.1 Problem

Figure 4.1 illustrates the concept of simulated multi-condition training. Because of the difficulty of collecting paired (noisy, clean) speech on the various environment, the majority of the speech enhancement training is based on this method.

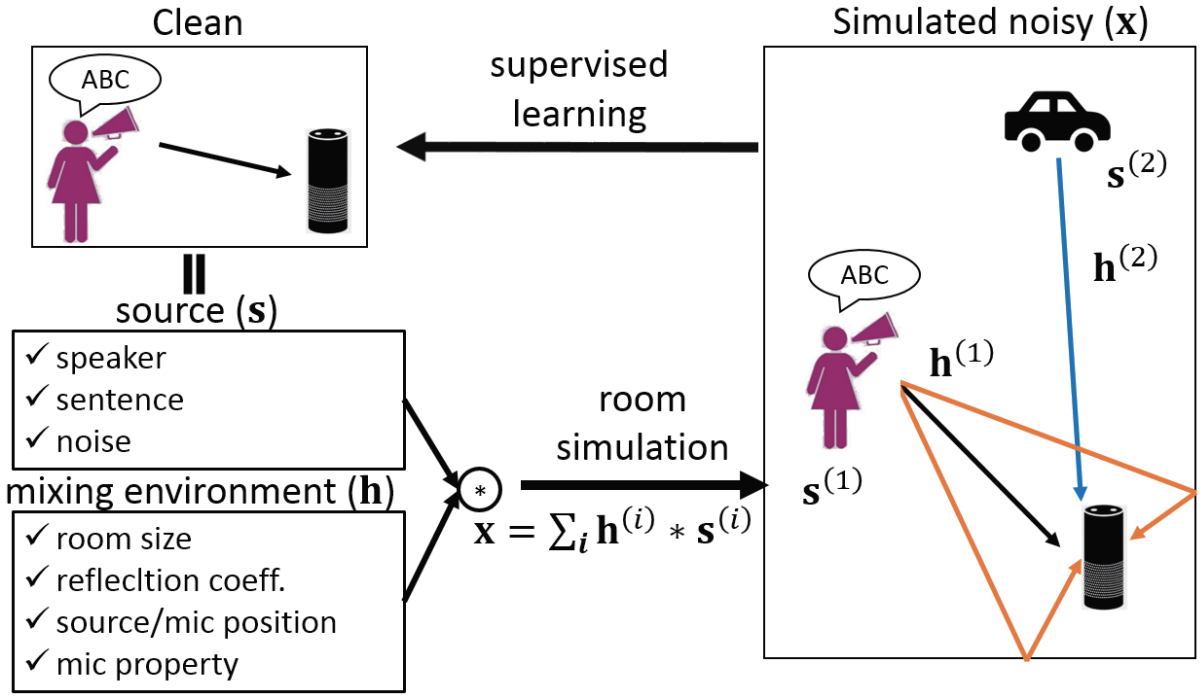


figure 4.1: Illustration of concept of simulated multi-condition training.

The noisy audio mixtures are generated by summation of convolution of several sources ($\mathbf{s}$) and impulse response ($\mathbf{h}$) between each source and microphone ($\mathbf{x}$). Sound sources vary by speaker, content or non-speech noise. The speech enhancement model are trained to find mapping from noisy to clean speech based on supervised learning.

Making simulated multi-condition training robust to environmental factors is one of the general interests. The source and position are the factors of interest in this chapter. We restrict the problem on a single-source case, therefore, de-reverberation is addressed and de-noising is not considered. The de-reverberation is important for far-field speech recognition applications such as AI conversational agents in a hands-free phone or robot. Moreover, we restrict change of environment only by source position while fixing room and multi-mics. The example scenario could be meeting speech recognition in a conference

room.

In this chapter, the following research questions are addressed:

- What makes simulated multi-condition training source/position sensitive?
- How to design source/position robust training algorithm with considering:
  - input/output
  - objective function
  - training data

4.2 Related work

4.2.1 Frequency-wise linear mixing/demixing

Single source $s(t)$ is recorded at M microphones $(x_m(t), m = 1, ..., M)$:

$$x_m(t) = \sum_{tau=0}^{T-1} h_m(\tau) s(t - \tau) \tag{4.1}$$

where $h_m(\tau), T$ is impulse response of m^{th} mic and its length in time domain.

Then, short-time fourier transform of $x_m(t)$ is:

$$X_{mf}[n] = \sum_{t=0}^{F-1} w(t) x_m(nJ + t) e^{-jw_f t} \tag{4.2}$$

where $w_f = \frac{2\pi(f-1)}{F}, F, J$ is fourier basis, window size, and shift size respectively.

If we choose window size much larger than length of impulse response (i.e., $F \gg T$), then $X_{mf}[n]$ is simplified as linear, known as multiplicative transfer function approximation [85]:

$$X_{mf}[n] \approx H_{mf} S_f[n] \tag{4.3}$$

The demixing weight W_{mf} exists as pseudo-inverse of mixing weights (H_{mf}):

$$\hat{S}_f[n] = \sum_{m=1}^{M} W_{mf} X_{mf}[n] \approx S_f[n] \tag{4.4}$$

Figure 4.2 illustrates frequency-wise linear mixing demixing process.

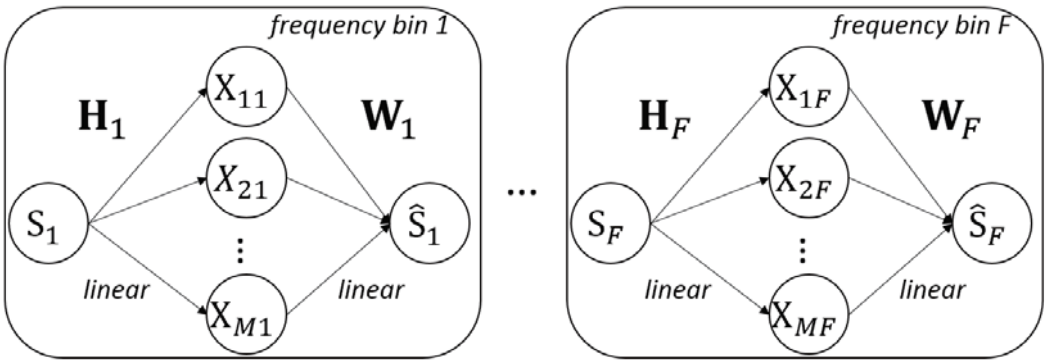


figure 4.2: Frequency-wise linear mixing/demixing process.

4.2.2 De-reverberation

Multi-channel beamforming

Representative works are minimum variance distortionless response (MVDR) beamforming [86], generalized eigenvalue (GEV) beamforming [87]. The performance of MVDR beamforming is sensitive to an estimation of direction of arrival (DoA). GEV beamformer maximizes SNR of output while not requiring DoA. However, it requires paired data between noisy sources and clean output. Independent component analysis (ICA) is used to separate mixed signals by enforcing statistical independence between multiple outputs. To choose the desired signal from multiple outputs, one needs constraints for a signal such as the direction of arrival or closeness.

Neural multi-condition training

For de-reverberation using neural network, methods are categorized as choice of *input/output* and *computation method across frequency*.

- choice of input and output

Many neural de-reverberation methods choose input and output as in Figure 4.3. Input is multi-mic signal and output is clean speech [88, 89, 90, 91, 92, 93] or demixing weight [94, 95, 96, 97]. Among them, demixing weight output is empirically outperforming speech output. Many researchers believe that it is because the demixing weight has less dynamic range than speech output and making training easier. On both models, the training process involves the complexity of source which require a variety of source for training data and complex enough model to learn such variation. However, our model chooses source-independent input and output making training affect less by a variety of sources.

- computation across frequency

With de-reverberation methods on frequency domain, there are 3 types of methods depending on computation design on frequency.

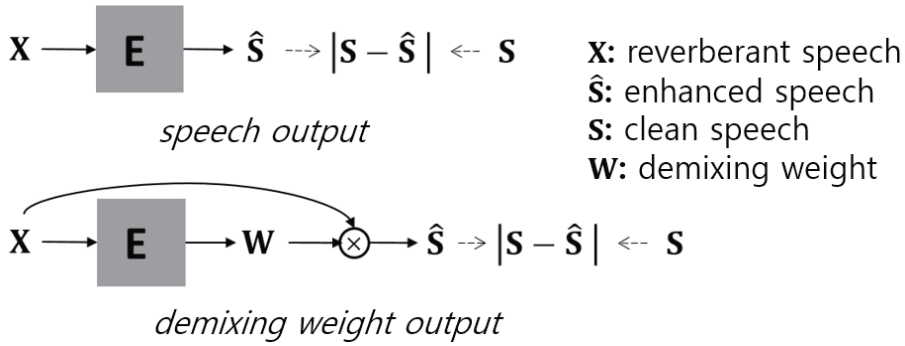


figure 4.3: Neural de-reverberation normally use multi-mics input and speech or demixing weight as output.

The first category share weights on frequency [94, 91, 92, 93]. We insist that sharing weight on frequency is inefficient to learn different position sensitivity on each frequency. Moreover, acoustic

features in the spectrogram do not share across frequency (i.e., local patterns in high frequency and low-frequency area is different). The deep complex U-net [94], shown in Figure 4.4 is the representative model we compared on experiment section.

The second category uses fully-connected weight on frequency such as multi-layer perceptron or recurrent neural network. However, it makes the model having an unnecessarily large number of parameters.

The third category uses frequency-wise estimation, estimating the demixing weight of each frequency independently. This scheme has been used in convolutive blind source separation [98, 99]. However, they iteratively optimize the demixing weight for every test data. However, we estimate the demixing weight from the inter-mic ratio without iterative optimization.

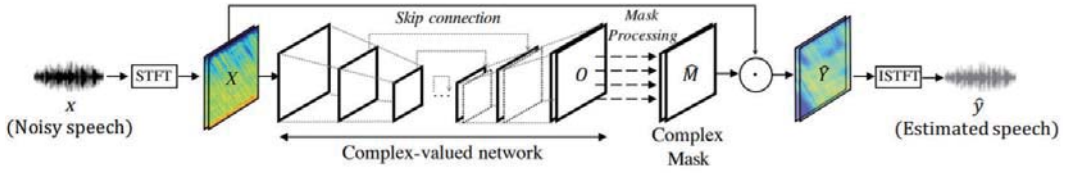


figure 4.4: Architecture of deep complex U-net.

4.2.3 Complex neural network

Complex neural network is expected to learn more generalizable solution than real neural network as complex number include real number and enable more flexible representation. Complex neural networks in many works [100, 101, 102, 94] show fast convergence and better generalization compared to real neural network, especially when learning relationship between complex numbers such as spectrogram.

Complex number z is represented as real (x)-imaginary (y) or magnitude (r) - phase (ϕ).

$$z = x + iy = re^{i\phi} \tag{4.5}$$

$$r = \sqrt{x^2 + y^2} \tag{4.6}$$

$$\phi = atan2(y,x) = \begin{cases} tan^{-1}(y/x) & x > 0 \\ tan^{-1}(y/x) + \pi & x < 0, y >= 0 \\ tan^{-1}(y/x) - \pi & x < 0, y < 0 \\ \pi/2 & x = 0, y > 0 \\ -\pi/2 & x = 0, y < 0 \\ undefined & x = 0, y = 0 \end{cases} \tag{4.7}$$

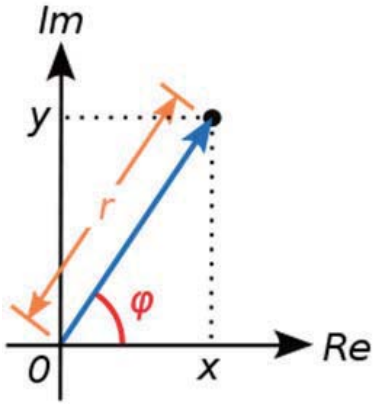


figure 4.5: Illustration of complex plane.

Let us review complex differentiable condition of complex-valued function. Given $f(z) = u(x,y) + iv(x,y)$, f is complex differentiable iff for every $z_0 \in \mathbb{C}, f^{'}(z_0) := \lim_{\Delta z \to 0}[\frac{f(z_0+\Delta z)-f(z_0)}{\Delta z}]$ exists. Here are the sufficient condition for complex differentiable:

- u, v is differentiable with respect to x, y
- satisfy Cauchy-Riemann equation :
  $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}$

The derivative in complex domain is defined as Wirtinger derivative [103]:
$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial z} + i\frac{\partial f}{\partial y}\frac{\partial x}{\partial y} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - i\frac{\partial f}{\partial y}\right)$

Among the layers of a real neural network, most of the learnable layers (i.e., linear, convolutional) can be extended to complex layers since they satisfy both conditions. For example, complex version of

linear and convolutional layers are expressed as follows:

$$\mathbb{C}linear(z; W, w) = Wz + w = (Ax - By + a) + i(Ay + Bx + b) \tag{4.8}$$

$$\mathbb{C}conv(z; W, w) = W * z + w = (A * x - B * y + a) + i(A * y + B * x + b) \tag{4.9}$$

However, the operation involving ordering, such as ReLU and max-pooling, cannot be used in complex domain since ordering is not defined between two complex numbers. However, there is non-linearity which is differentiable in some range of complex plane. For example,

$$\mathbb{C}ReLU(z) = ReLU(x) + iReLU(y) \tag{4.10}$$

is differentiable when $\phi \in [0, \frac{\pi}{2}]$ or $\phi \in [\pi, \frac{3\pi}{2}]$. In practice, a single complex layer represents input and output by a set of real and imaginary neurons. Although complex numbers can be expressed by magnitude and phase, it is often avoided since periodicity in phase is hard to represent (e.g., $\phi = 0$ and $\phi = 2\pi$ is same).

For neural network, we are interested in getting gradient of real-valued loss function with respect to the parameters of complex neural network. Let $L, f^n, \mathbf{z}_n, \mathbf{z}_{n+1}$ as real-valued loss, n^{th} layer, its input and output (i.e., $\mathbf{z}_{n+1} = f^n(\mathbf{z}_n)$).

For deriving error back-propagation, we define $\delta_n := \frac{\partial L}{\partial \mathbf{z}_n}$. The relationship between δ_{n+1} and δ_n is derived as follows:

$$\frac{\partial L}{\partial \mathbf{z}_n} = \frac{1}{2}\left(\frac{\partial L}{\partial \mathbf{x}_n} - i\frac{\partial L}{\partial \mathbf{y}_n}\right) \tag{4.11}$$

$$= \frac{1}{2}\left(\frac{\partial L}{\partial \mathbf{x}_{n+1}}\frac{\partial \mathbf{x}_{n+1}}{\partial \mathbf{x}_n} + \frac{\partial L}{\partial \mathbf{y}_{n+1}}\frac{\partial \mathbf{y}_{n+1}}{\partial \mathbf{x}_n}\right) - \frac{i}{2}\left(\frac{\partial L}{\partial \mathbf{x}_{n+1}}\frac{\partial \mathbf{x}_{n+1}}{\partial \mathbf{y}_n} + \frac{\partial L}{\partial \mathbf{y}_{n+1}}\frac{\partial \mathbf{y}_{n+1}}{\partial \mathbf{y}_n}\right) \tag{4.12}$$

$$= \frac{1}{2}\frac{\partial L}{\partial \mathbf{x}_{n+1}}\left(\frac{\partial \mathbf{x}_{n+1}}{\partial \mathbf{x}_n} - i\frac{\partial \mathbf{x}_{n+1}}{\partial \mathbf{y}_n}\right) - \frac{i}{2}\frac{\partial L}{\partial \mathbf{y}_{n+1}}\left(\frac{\partial \mathbf{y}_{n+1}}{\partial \mathbf{y}_n} - i\frac{\partial \mathbf{y}_{n+1}}{\partial \mathbf{x}_n}\right) \tag{4.13}$$

$$= \begin{cases} \delta_{n+1}\left(\frac{\partial \mathbf{x}_{n+1}}{\partial \mathbf{x}_n} + i\frac{\partial \mathbf{y}_{n+1}}{\partial \mathbf{x}_n}\right) \\ \qquad or \\ \delta_{n+1}\left(\frac{\partial \mathbf{y}_{n+1}}{\partial \mathbf{y}_n} - i\frac{\partial \mathbf{x}_{n+1}}{\partial \mathbf{y}_n}\right) \end{cases} \tag{4.14}$$

For linear layer as an example, weight (W_n) and bias (w_n) gradient is derived as follows:

$$\frac{\partial L}{\partial W_n} = \delta_{n+1}(\mathbf{x}_n - i\mathbf{y}_n) = \delta_{n+1}\mathbf{z}_n^H \tag{4.15}$$

$$\frac{\partial L}{\partial w_n} = \delta_{n+1} \cdot \mathbf{1} \tag{4.16}$$

Finally, step size (Δ) is determined as a conjugate of the weight gradient for gradient descent:

$$\Delta_{\mathbf{W}} = \overline{\frac{\partial L}{\partial \mathbf{W}}} \tag{4.17}$$

From 1^{st} order Taylor series expansion given below, this step size guarantee decreasing loss for every update.

$$L(\mathbf{W} - \eta\overline{\frac{\partial L}{\partial \mathbf{W}}}) \approx L(\mathbf{W}) + \frac{\partial L}{\partial \mathbf{W}}(-\eta\overline{\frac{\partial L}{\partial \mathbf{W}}}) = L(\mathbf{W}) - \eta\left\|\frac{\partial L}{\partial \mathbf{W}}\right\|^2 < L(\mathbf{W}) \tag{4.18}$$

4.3 Method

The overall system is shown in figure 4.6.

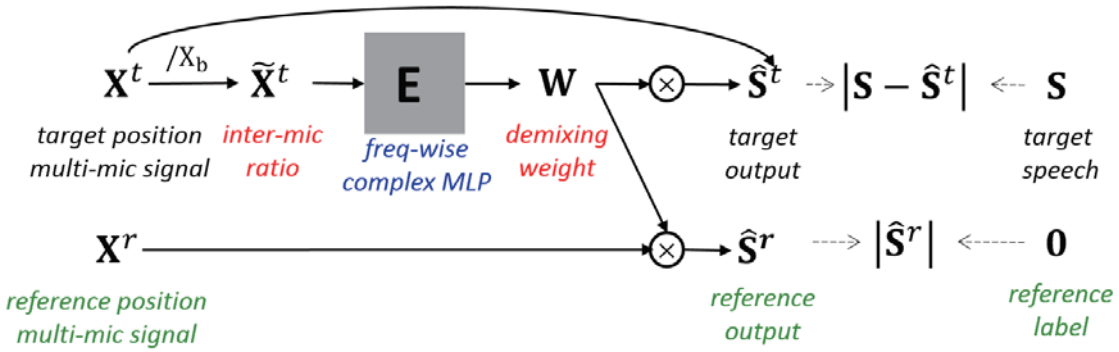


figure 4.6: Illustration of overall system.

In the following chapters, proposed methods for source-robustness, position-robustness and oracle model (i.e., every approximations are correct) will be addressed.

4.3.1 Source-robustness: Source independent input/output

Output: Demixing weight

Consider the mixing/demixing process of source ($\mathbf{s}$) to J mics ($\mathbf{x}$). With using analysis window much longer than length of impulse response, their spectrogram is approximately having linear relationship with coefficient as mixing weight. This approximation is known as multiplicative transfer function approximation [85]. Specifically, for each frequency point:

$$\mathbf{X} = \mathbf{H}S \tag{4.19}$$

$$S = \mathbf{W}\mathbf{X} \tag{4.20}$$

where $\mathbf{X} \in \mathbb{C}^J$, $S \in \mathbb{C}^1$, and $\mathbf{W} \in \mathbb{C}^J$.

Given mixing weight, the demixing weight form vector space:

$$V(\mathbf{W}^*) = \{W_1, ..., W_M | \sum_{m=1}^{M} W_m H_m = 1\} \tag{4.21}$$

If the mixing weights (H_m) of each mic m are given, the problem is trivially solved by finding such vector space. However, mixing weight or impulse response is generally unobservable. Therefore we need to estimate W from the observation mic signal X.

$$\mathbf{W} = f_\theta \mathbf{X} \tag{4.22}$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, d(\mathbf{S}, \mathbf{W}\mathbf{X}) \tag{4.23}$$

where f_θ and d is estimation model and distance metric respectively.

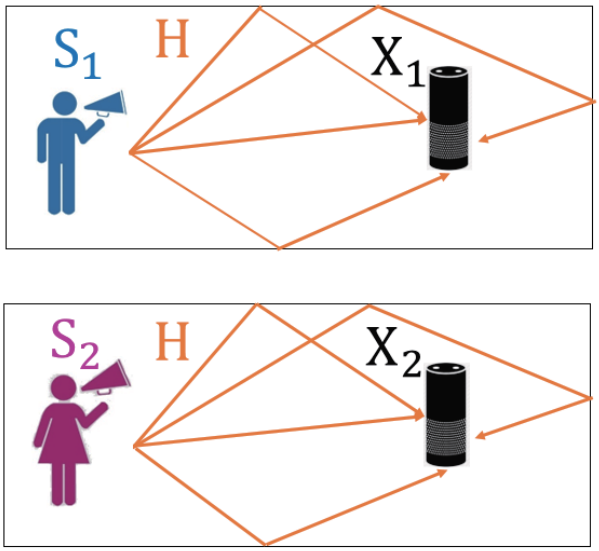


figure 4.7: The mixing weight are only function of the environment, and independent to source.

Here we focus on the fact that W is only a function of mixing environment (e.g., room size, reflection coefficients, location of microphone and source) and independent to speech sources. If two observed speeches are recorded in the same mixing environment, each mixing weights should be the same as shown in figure 4.7. In this case, demixing weight for two observed speeches forms the same vector space and independent of the content of the source. We call this property as **source-independence** of demixing weight.

Input: Inter-mic ratio (IMR)

For pursuing **source-independence** of demixing weight, we propose using intermic ratio (IMR) as input feature. Literally, it is the ratio between every mics and base mic signal on complex domain. With frequency-wise linear mixing/demixing model, intermic ratio can be the ratio between mixing weight (relative transfer function):

$$\tilde{\mathbf{X}} = \frac{X_{\neq b}}{X_b} \approx [\frac{H_1}{H_b}, .., \frac{H_M}{H_b}] \tag{4.24}$$

Note that, relative transfer functions of M microphones are source-independent. The IMR feature is often utilized for localization [35, 36, 37]. Consider two microphone signals $(x_1(t), x_2(t))$ with no reverberation. Each microphone signal is defined by different attenuation (a_i) and time delay (τ_i):

$$x_1(t) = a_1 s(t - \tau_1) \tag{4.25}$$

$$x_2(t) = a_2 s(t - \tau_2) \tag{4.26}$$

$$\frac{X_1(w)}{X_2(w)} = \frac{a_1}{a_2} e^{jw(\tau_2 - \tau_1)} \tag{4.27}$$

Then, intermic ratio in frequency domain is represented by attenuation ratio $(\frac{a_1}{a_2})$ and time delay difference $(\tau_2 - \tau_1)$ of two microphones, which are useful feature for localization.

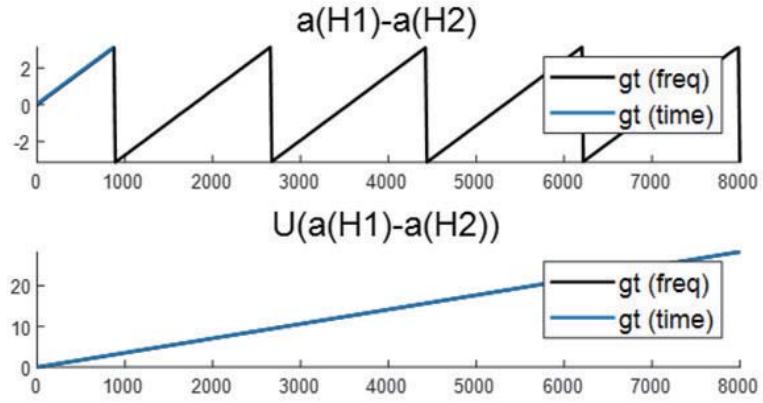


figure 4.8: Illustration of phase unwrapping.

The spatial information is one of the factors to define impulse response. The other factors are room size, reflection coefficients, mic property (i.e., directivity). In this chapter, we constrain the variation of impulse response only by source position (i.e., fix room and mic). Therefore, the spatial information can uniquely determine the mixing environment and we can expect that the inter-mic ratio feature can predict the demixing weight.

For effective localization, we need to consider phase wrapping issue. In observed complex signal phase lies in $(-\pi, \pi]$, since $e^{j\theta} = e^{j\theta+2n\pi}$ for arbitrary integer n. Back to IPD given no reverberation (Equation (4.27)), its observed value becomes:

$$\angle X_1(w) - \angle X_2(w) = mod(w(\tau_2 - \tau_1) + \pi, 2\pi) - \pi = \Phi(w(\tau_2 - \tau_1)) \tag{4.28}$$

Since neural network is hard to learn Φ^{-1}, we pre-process IPD by unwrapping shown in algorithm 1. Figure 4.8 shows before and after phase unwrapping.

Algorithm 1 Phase unwrapping

1: $\hat{\theta(1)} = \theta(1)$
2: **for** $f \leftarrow 2$ to F **do**
3: $\quad \hat{\theta(f)} = \theta(f) + 2\pi\hat{k}$
4: $\quad \hat{k} = \text{argmin}_k \, |\theta(f) + 2\pi k - \hat{\theta}(f-1)| \in \mathbb{Z}$

Note that the Equation (4.27) does not hold in practice. The first factor comes from reverberation. If we consider indirect paths that come from the reflected signal from the wall, the phase of the inter-mic ratio does not follow linear across frequency.

$$x_1(t) = a_1 s(t - \tau_1) + \sum_n a_1^{(n)} s(t - \tau_1^{(n)}) \tag{4.29}$$

$$x_2(t) = a_2 s(t - \tau_2) + \sum_n a_2^{(n)} s(t - \tau_2^{(n)}) \tag{4.30}$$

$$\frac{X_1(w)}{X_2(w)} = \frac{a_1 e^{-jw\tau_1 + \sum_n a_1^{(n)} e^{-jw\tau_1^{(n)}}}}{a_2 e^{-jw\tau_2 + \sum_n a_2^{(n)} e^{-jw\tau_2^{(n)}}}} \tag{4.31}$$

Figure 4.9 compares impulse response and transfer function phase difference of both no reverberation

(RT60 = 0ms) and reverberant (RT60 = 500ms) case. We can see that IPD across frequency deviate from linear due to the reverberation.

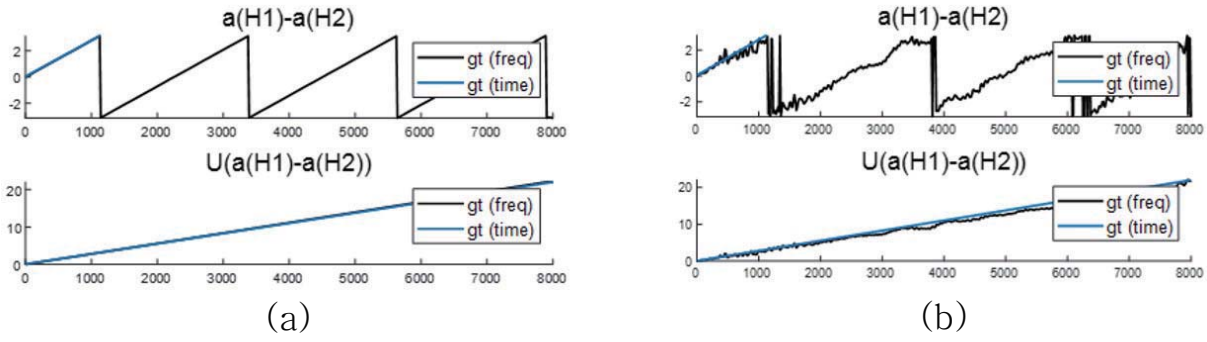


figure 4.9: Illustration of phase difference of two transfer functions (H1, H2).

The second factor comes from the approximation error of multiplicative transfer function approximation. In practice, we use a short-time Fourier transform (STFT) instead of a fast Fourier transform on full signal since it is meaningful to set stationary interval (i.e., frequency components does not change). In each time frame, the amounts of source information differ due to different time delay in each mic. Consider STFT (without effects of window function) of two microphone signals with time delay $\tau_1, \tau_2 (\neq \tau_1)$.

$$\frac{X_1(w)}{X_2(w)} = \frac{a_1 e^{-jw(\tau_1)} S^{\tau_1}(n,w)}{a_2 e^{-jw(\tau_2)} S^{\tau_2}(n,w)} \tag{4.32}$$

Figure 4.10 show intermic phase difference (IPD) of two microphones when no reverberation exists. We can see that IPD across from deviates from linear.

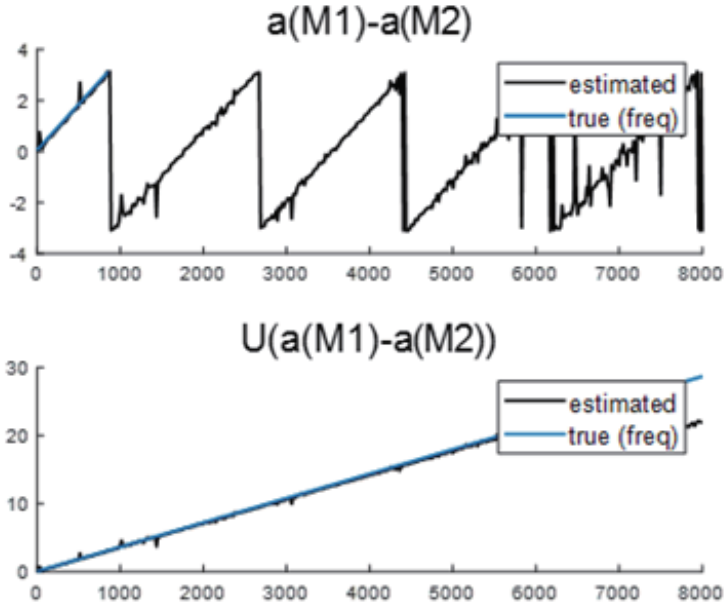


figure 4.10: Illustration of phase difference of two microphones.

This quantity does not source independent anymore, and we call ($\frac{S^{\tau_1}(n,w)}{S^{\tau_2}(n,w)}$) source-residual. To reduce the effects of source-residual on computing the inter-mic ratio, we can think of using a large-size window. However, a large-size window requires high resolution of FFT which often becomes unreliable at some frequencies not included in the speech. This issue can be overcomed by simple frequency response undersampling of IMR, shown in section 4.4.1.

Given room characteristic (i.e., room size, reflection coefficients), localization enables inference of impulse response, since the input of room simulator is room characteristic and source/mic position. Therefore, we expect the inter-mic ratio include information to estimate demixing weight.

Undersampling Frequency respone of IMR can be further improved by under-sampling (or band-pass sampling). Since IMR is used with large window size, it is required to having high frequency resolution. This makes IMR becomes noise-sensitive for few frequencies not in speech. Also, it requires a large number of parameters proportional to number of frequency to learn.

Therefore, IMR can be improved with frequency undersampling (or bandpass sampling). As in Figure 4.11, bandpass filters, around center frequencies, are used to reduce resolution of frequency. It make noisy frequency components in IMR smooth, and reduce the number of parameters. The effects of under-sampling can be seen in Figure 4.12.

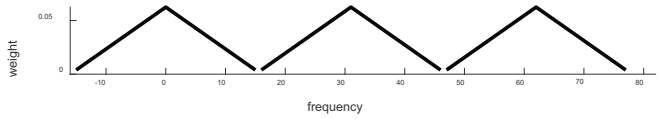


figure 4.11: Illustration of undersample filter.

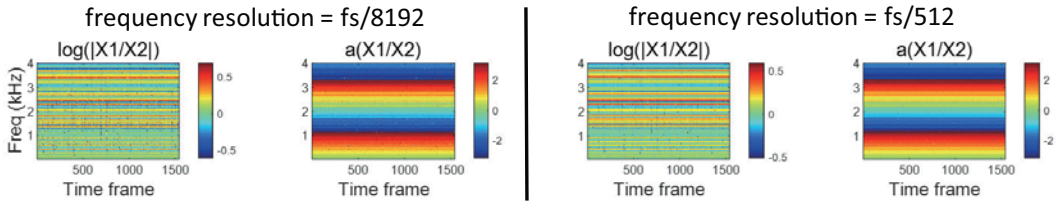


figure 4.12: Illustration of undersampling result.

When computing the inter-mic ratio (IMR), the effects of the source residual may decrease with a larger window size. However, large window size typically requires high-resolution frequency response, often unreliable if the speech does not include a specific frequency component. However, the IMR usually varies slowly across frequency. For example, if we consider IMR with no reverberation case, the phase has a linear relationship to frequency (Equation (4.27)). Therefore, we enhance the IMR feature by averaging neighboring points and make the undersampled version of the IMR feature. This pre-processing enables extracting the IMR feature on a long window with lower frequency resolution. For average filter, the

triangular average filter is used:

$$X_k^d = \sum_{i=dk-d+1}^{dk+d+1} (\frac{1-|i-dk|}{d}) X_i \tag{4.33}$$

where d is the downsampling rate.

Figure 4.13 shows the effects of downsampling on the IMR feature. Microphone signals are recorded with the reverberation time (RT60)100ms, and the FFT point of the IMR is downsampled from 8192 to 512 points. Using a large downsampling rate makes IMR smooth across frequency change.

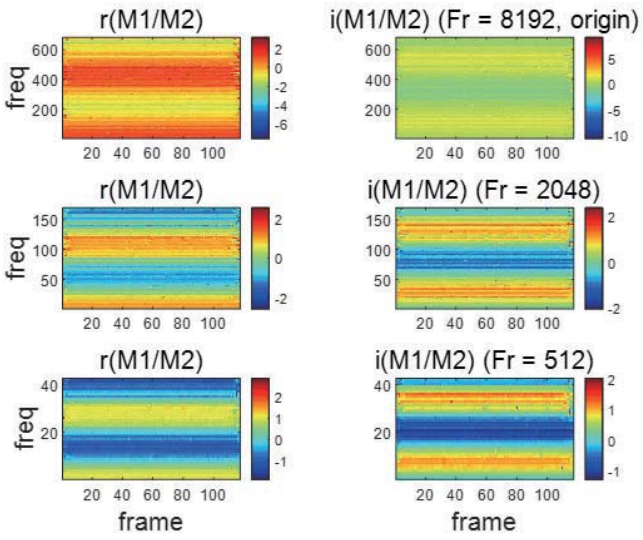


figure 4.13: Result of downsampling IMR.

4.3.2 Position-robustness: Frequency-wise complex MLP/Reference position regularization

Problem analysis: position sensitivity of mixing weight

We analyze characteristics of the problem by investigating how mixing weight vary by position of source. Consider scenario with fixed room, mics, and only source is moving within small area. The figure 4.14 and table 4.1 describes scenario and detail of the settings.

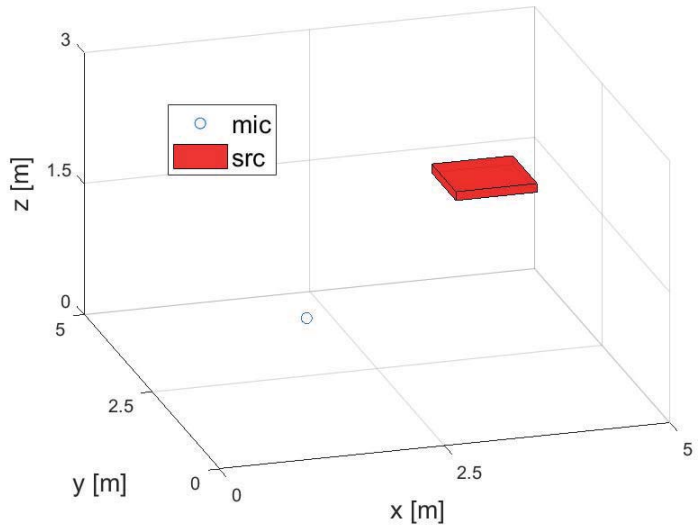


figure 4.14: Illustration of room, mic, and source distribution for the analysis.

table 4.1: The configuration of dataset used for analyzing position sensitivity of mixing weight.

Room size	5x5x3 (m^3)
Mic center	(1.3, 1.1, 1.2) (m)
Mic directivity	Hypercardioid
Mic position	r = 0.05 (m) theta = (50, 230) (deg) phi = (0, 0) (deg)
Source range	[4.0, 4.9]x[4.0, 4.9]x[1.3, 1.4] (m^3)
Position sampling	1 (cm)
RT60	200 (ms)

Figure 4.15 show how mixing weight vary when source position changes. The first/second rows show magnitude/phase of mixing weight respectively. We can observe that mixing weight (H) is sensitive to position on high frequency.

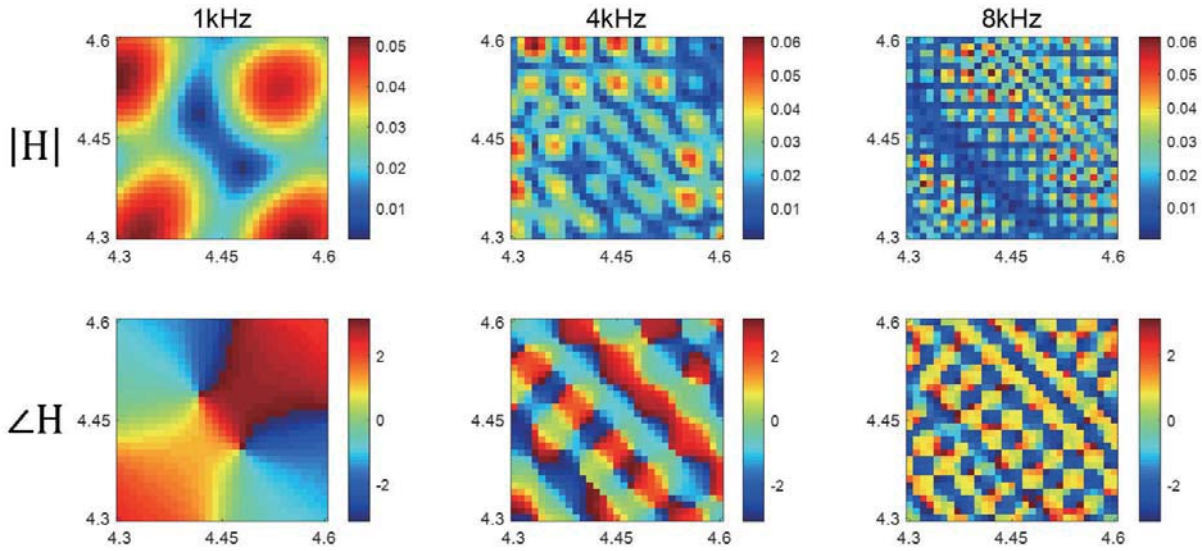


figure 4.15: Illustration of varying mixing weight with respect to position of source.

To understand this result, let us represent mixing weight with R impulses generated from source position p to fixed mic:

$$H(w;p) = a_1^p e^{-jw\tau_1^p} + a_2 e^{-jw\tau_2^p} + ... + a_R e^{-jw\tau_R^p} (\tau_r^p \leq \tau_{r+1}^p) \tag{4.34}$$

$$H(w;p) = (a_1 \cos(w\tau_1^p) + ... + a_R \cos(w\tau_R^p)) - j(a_1 \sin(w\tau_1^p) + ... + a_R \sin(w\tau_R^p)) \tag{4.35}$$

Note that attenuation and time delay of each impulse change with source position p. The first term represents direct impulse, and the remaining terms are reverberation. For each impulse, the phase is proportional to frequency and time delay. The time delay is also proportional to the travel distance of each impulse. The phase of each impulse varies fast in high frequency. Therefore, the sum of impulses varies fast in both magnitude and phase.

This result implies that mixing and demixing weight has different sensitivity on position change of mic and source.

Frequency-wise complex multi-layer perceptron

Based on prior knowledge on physical model of reverberation and signal processing, we set requirements for the parametrization of the enhancement model.

- **Requirement 1** Enhancement model is **complex neural network** (i.e., complex parameters, complex arithmetic).

  The complex neural network supports generalized arithmetic than a real neural network. Moreover, it expects to generalize better for the learning relationship between complex input(inter-mic ratio)/output(demixing weight).

- **Requirement 2** Enhancement model has its **own sets of parameter per frequency**.

  Our model is based on frequency-wise linear mixing/demixing process model shown in section 4.2. The source STFT ($S(k)$) of a specific frequency is independent of mic signal STFT ($X(k')$) of different frequencies, shown in Theorem 1.

  Theorem 1 (Frequency independence)**.** *Source of frequency k is independent to mic of different frequency k'.*

  $$I(S(k); X(k')) = 0 (if k \neq k')$$

  Proof. see Theorem 3 of [104] □

  From this fact, we can expect that demixing weight of single frequency is independent to mic signal at different frequency (i.e., $I(W(k); X(k')) = 0$, if $k \neq k'$). Therefore, we estimate $W(k)$ only from $X(k)$, and having independent parameter per each discrete frequency k.

  The separate parameters per frequency are expected to learn different sensitivity to position change shown in the figure 4.15.

- **Requirement 3** Enhancement model is **stationary**.

  Since we restrict the problem as stationary mixing environment (i.e., source, mic position and room environment), we can expect estimation model sharable to all time frames.

Figure 4.16 shows the simplest baseline architecture satisfying above requirements.

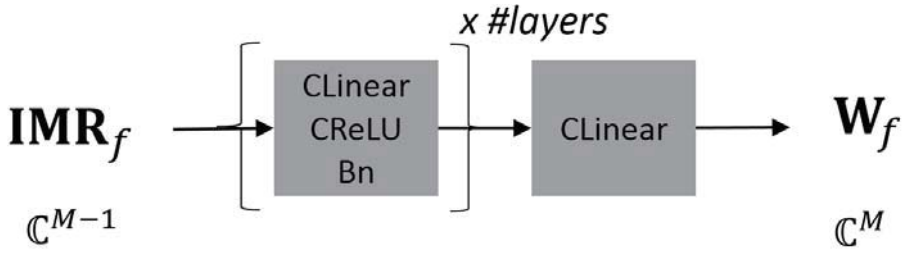


figure 4.16: Illustration of basic architecture.

CLinear, CReLU, Bn stands for complex linear, complex-relu, and batch normalization, defined in section 4.2.

The single-frequency input is often not reliable when the speech does not include a specific frequency component. Moreover, there is a high correlation between neighboring frequencies of the IMR feature. (i.e., when no reverberation, IPD difference of neighboring frequency is $\frac{2\pi(\tau_2-\tau_1)}{N}$, *Equation* (4.28)). Therefore, input to the network can include IMR features of neighboring frequencies as shown in Figure 4.17.

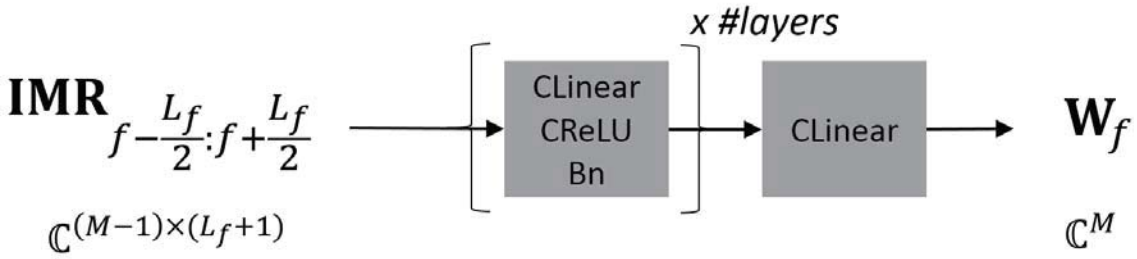


figure 4.17: Illustration of basic architecture improved with multi-frequency input.

The model is named as *frequency-wise complex multi-layer perceptron* since independent parameters exists per frequency. Moreover, parameters are shared across the time frames since the model is assumed to be stationary. To parallelize the above model on all TF units of microphone pairs, we employ depthwise separable convolution [105] with a convolution channel that can be viewed as a frequency bin.

Upsampling layer If IMR features are downsampled on frequency, estimated demixing weight also presented in downsampled frequency resolution. Therefore, it is required to upsample demixing weight on the frequency axis.

If the downsampling rate is $d = 2^n$, n number of upsampling layers are added, where each layer doubles frequency resolution. This layer consists of complex transposed convolution with stride = 2 on the frequency axis, cReLU, and batch normalization.

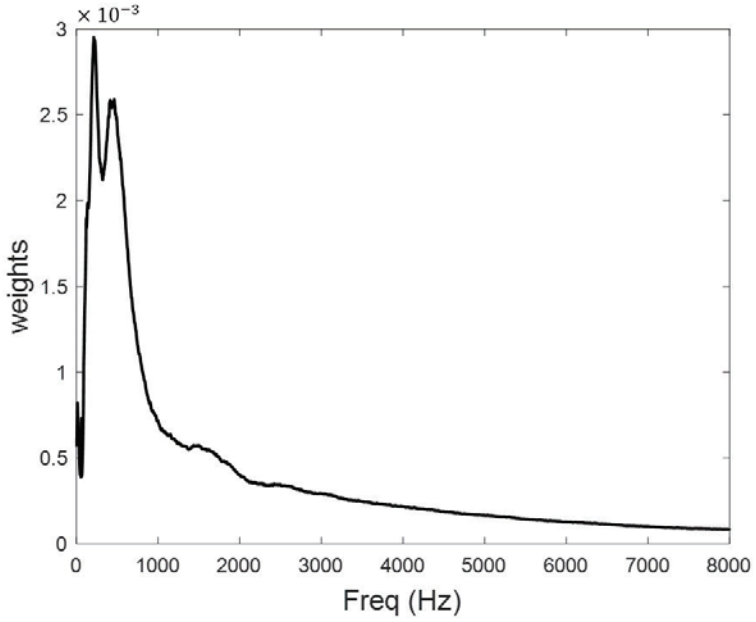


figure 4.18: Imbalanced speech energy distribution. The most of the energy lies below 1kHz.

Loss function

De-reverberation is evaluated with the signal-to-distortion ratio (SDR)[db]:

$$SDR(\hat{S}; S) = 10 \log_{10} \frac{\sum_t \sum_f |S_{tf}|^2}{\sum_t \sum_f |S_{tf} - \hat{S}_{tf}|^2} \tag{4.36}$$

If we assume frequency-wise linear mixing/demixing process shown in section 4.2, SDR is expressed by:

$$SDR(\hat{C}; S) = 10 \log_{10} \frac{\sum_t \sum_f |S_{tf}|^2}{\sum_t \sum_f |S_{tf}|^2 |1 - \hat{C}_{tf}|^2} \tag{4.37}$$

where, $\hat{C}_{tf} = \sum_m \hat{W}_m H_m$. Since enhancement model only affects distortion power, we can choose the distortion power as loss function:

$$L(\hat{C}; S) = \sum_t \sum_f |S_{tf}|^2 |1 - \hat{C}_{tf}|^2 \tag{4.38}$$

However, there is *frequency absence problem* in this loss function. If $|S_{tf}| \approx 0$, $\hat{C}_{tf}$ is weighted with very small portion so that it barely learned. Moreover, $\hat{C}_{tf} = \frac{\hat{S}_{tf}}{S_{tf}}$ can be caluculated with very large value causing numerically unstable behavior during training.

This problem often happens since speech has unbalanced energy distribution (i.e., the majority of the energy lies below 1kHz) shown in Figure 4.18. Moreover, we are using a large size window and it requires high-frequency resolution. High-frequency resolution often involves frequency with almost zero energy.

So far, the evaluation metric (SDR) is source-dependent and loss can suffer from frequency absence problem. When we assume impulse response is known (i.e., simulated data), we can think of source-

independent SDR and loss function without frequency absence problem:

$$SDR(\hat{C}) = 10\log_{10}\frac{\sum_f 1}{\sum_f |1-\hat{C}_f|^2} \tag{4.39}$$

$$L(\hat{C}) = \sum_f |1-\hat{C}_f|^2 \tag{4.40}$$

However, this metric is not available for real data since mixing weight (or impulse response) is generally unobservable. Although this metric is available only for simulated data, it serves source-independent SDR and loss function without a frequency absence problem. We call this metrics *oracle* metrics.

One can expect that real environment performance approach *oracle*, when the source has large energy for every frequency range such as white noise. With speech sources, we can use phonetically balanced speech to learn demixing weight with large energy of diverse frequency range.

The SDR can be measured with scale-invariant sense, by matching the direction of output and target vectors and ignoring scale difference. The scale-invariant SDR between clean (s) and enhanced ($E(x)$, E: enhancement model) is given as follows ([106]):

$$SDR(E;x,s) = 10\log_{10}\frac{\langle s, E(x)\rangle^2}{\|s\|^2\|E(x)\|^2 - \langle s, E(x)\rangle^2} \tag{4.41}$$

In this case, the loss function can be further simplified as cosine simiarity between clean (s) and enhanced ($E(x)$) speech:

$$\underset{E}{\operatorname{argmax}}\, SDR(E;s,x) = \underset{E}{\operatorname{argmax}}\frac{\langle s, E(x)\rangle^2}{\|s\|^2\|E(x)\|^2 - \langle s, E(x)\rangle^2} \tag{4.42}$$

$$= \underset{E}{\operatorname{argmin}}\frac{\|s\|^2\|E(x)\|^2}{\langle s, E(x)\rangle^2} \tag{4.43}$$

$$= \underset{E}{\operatorname{argmin}} -\frac{\langle s, E(x)\rangle^2}{\|s\|^2\|E(x)\|^2} = \underset{E}{\operatorname{argmin}}\, L(E;s,x) \tag{4.44}$$

Reference position regularization

We revisit the ground-truth demixing weight (W) forming vector space as follows:

$$V(\mathbf{W}^*) = \{W_1, ..., W_M | \sum_{m=1}^{M} W_m H_m = 1\} \tag{4.45}$$

This is underdetermined system where W is not uniquely determined. To reduce W variance with respect to neural network size, initialization and choice of training samples. we further constrain vector space to determined system to uniquely set ground-truth W.

Consider $M-1$ reference positions in a room as shown in Figure 4.19. The weighted sum of demixing weights (W_m) and mixing weight from reference positions ($\tilde{H}_m^{(r)}$) need to be zero. With additional equations, the demixing weight is uniquely obtained from determined system.

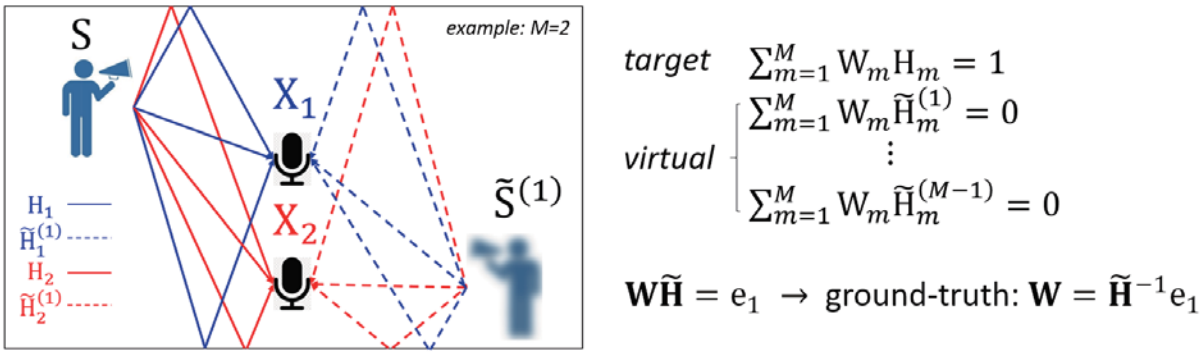


figure 4.19: Adding reference source in a room (the number of mic = 2).

Therefore, the loss function can be simply providing the ground-truth W to estimated W:

$$L(W) = |W - \tilde{H}^{-1} e_1|^2 \tag{4.46}$$

However, the mixing weight(or impulse response) is unobservable in a real environment. Therefore, we can approximately calculate the ground-truth W from mic signal (X) and source (S) with frequency-wise linear mixing/demixing approximation. The ground-truth W is compared in Figure 4.20.

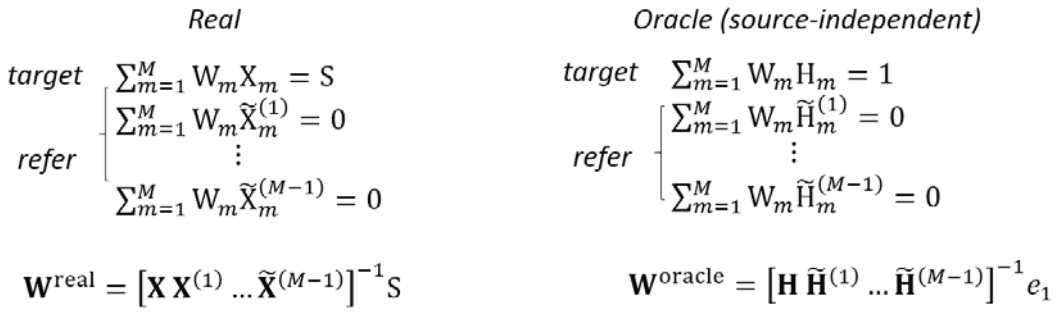


figure 4.20: The ground-truth demixing weight on real and oracle (mixing weight available) condition.

$W^{real} = W^{oracle}$ when inverse matrix in both term exists and frequency-wise linear mixing/demixing holds. However, inverse matrix in real condition often not exists. One example is zero energy of target or reference source energy at some tf bin (i.e., $S_{tf} = 0$ or $\tilde{S}^{(r)}_{tf} = 0$), which is frequently occur in natural speech.

To avoid matrix singularity problem, we can employ additional regularization in a loss function:

$$L(W) = |S - \sum_{m=1}^{M} W_m X_m|^2 + \lambda \sum_{r=1}^{M-1} |\sum_{m=1}^{M} W_m \tilde{X}^{(r)}_m|^2 \tag{4.47}$$

For ground-truth W defined for unseen target positions, the reference positions should be fixed for all target positions.

This reference position regularization can be interpreted as a spatial filter. If we choose a reference position as the opposite side of the source, it filters source coming reference positions. For example, background music from the opposite side and motor sound nearby robot microphones is a representative case of the reference source.

Regarding using reference sources, we empirically answer for the following question: Given room and target position distribution, which choice of reference position make ground-truth W vary smoothly across position?

4.3.3 Oracle model

When the mixing weight (or impulse response) available, we can define the oracle model with the following properties:

- exact relative transfer function

In a real scenario, the inter-mic ratio is approximately the same as the relative transfer function. This approximation may not holds for two conditions. Firstly, when the analysis window is not much larger than the length of the impulse response, multiplicative transfer function approximation does not hold and the inter-mic ratio deviates from the relative transfer function. Secondly, when the source has zero magnitudes at certain time-frequency points (i.e., $S_{tf} = 0$), the inter-mic ratio is numerically unstable and deviates from the relative transfer function.

- no frequency absence problem

In real scenario, the source often has zero magnitude at certain time-frequency point (i.e., $S_{tf} = 0$). Demixing weight cannot be learned at this time-frequency point. In oracle scenario, we use following simplified loss function:

$$L(\hat{C}) = \sum_{f} |1 - \hat{C}_f|^2 \tag{4.48}$$

This loss function can be viewed as assuming source always has non-zero same value at every time-frequency bin.

With above properties, oracle model is shown in Figure 4.21.

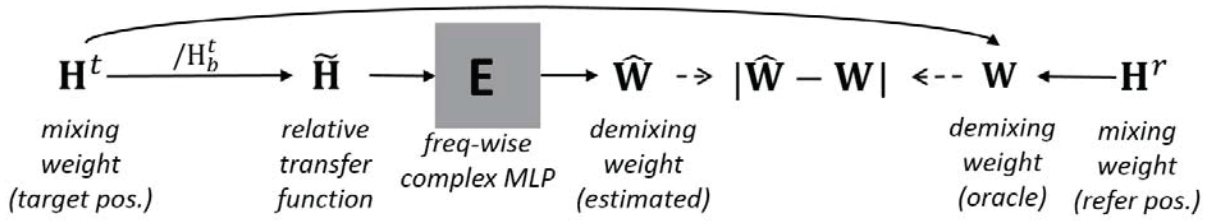


figure 4.21: Illustration of oracle model. In oracle model, 1) relative trasnfer function is exact and, 2) there is no frequency absence problem.

4.4 Experiment

4.4.1 Room impulse response generator

Since directly measuring impulse response is challenging task, the room impulse response (RIR) generator is often used in speech enhancement field. The image method [107] is the basic principle to construct RIR. The basic concept is illustrated in Figure 4.22

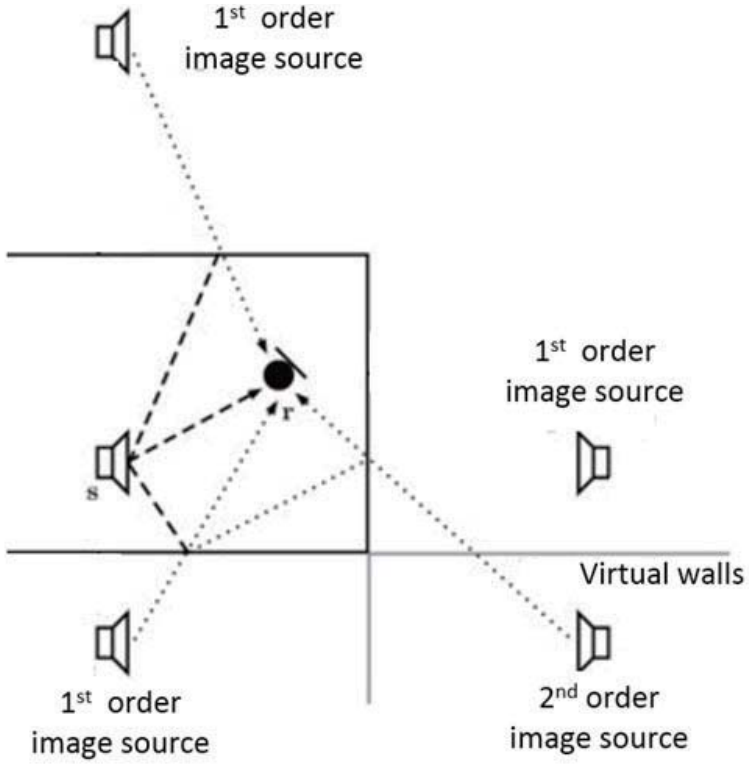


figure 4.22: Illustration ofthe image method used in room impulse response generator.

In the image method, reflected speech (or indirect path) is considered as an independent image source coming from virtual space beyond walls so that image source travels the same distance as the reflected path. The image source is located at the symmetric point behind the wall whenever the indirect path has reflections. The indirect path having n reflections is called n^{th} order image source. The reflection makes signal amplitude decreased and modeled as reflection coefficients (β). Also, signal amplitude attenuates as travel distance increases (i.e., amplitude $\propto$ 1/distance). The reverberation time (RT60), which is time until sound in a reverberant environment to decay 60dB in level, is used to control the length of RIR. Based on pre-defined RT60, the reflection coefficient are adjusted by Sabin-Franklin's formula [108]:

$$RT_{60} = \frac{24\log(10)V}{c\sum_{i=1}^{6} S_i(1-\beta_i^2)} \tag{4.49}$$

where $V, S_{i=1}^{6}, \beta_{i=1}^{6}$ is volume of the room, size of the wall, reflection coefficients of 6 sides of rectangular room.

Given source ($\mathbf{p}_s$), mic position ($\mathbf{p}_m$) impulse response is represented as the sum of reflection coefficient multiplied by delayed, and attenuated impulses:

$$h(t;\mathbf{p}_s,\mathbf{p}_m)=\sum_n(\prod_{i=1}^{6}\beta_i^{r_i(n)})\frac{\delta(t-\tau_n)}{4\pi d_n} \tag{4.50}$$

where for given n^{th} impulse, $\tau_n, d_n, r_i(n)$ is time delay, travel distance and the number of reflection of i^{th} walls respectively.

For a discrete-time version, sets of time delay are mapped to the nearest sampled point and pass a low-pass filter to avoid temporal aliasing problem.

$$h(n;\mathbf{p}_s,\mathbf{p}_m)=\sum_n(\prod_{i=1}^{6}\beta_i^{r_i(n)})\frac{LPF(\delta(n-[\tau_n f_s]))}{4\pi d_n} \tag{4.51}$$

$$LPF(t)=\begin{cases}\frac{1}{2}(1+\cos(\frac{2\pi t}{T_w}))sinc(2\pi f_c t) & -\frac{T_w}{2}<t<\frac{T_w}{2}\\ 0 & otherwise\end{cases} \tag{4.52}$$

where f_s, T_w is sampling frequency and width of impulse (default value = 4ms).

Note that the simulator usually ignores such a physical phenomenon in a real environment:

- phase reversal of sound wave reflecting from a dense wall
- scatter (or diffraction) effects in non-uniform surface

4.4.2 Dataset and settings

Dataset

We generate the simulated dataset to empirically assess source and position robustness of conventional and proposed de-reverberation methods.

The summary of database is given in table 4.2

table 4.2: The summary of the database used in experiment.

Type	Room	Mic	Impulse response				#Source /Position
			Range	Sampling	#Position	RT_{60}	
train	5x5x3 (m^3)	p = (1.3, 1.1, 1.2) (m) r = 0.05m theta = (50, 230) (deg) hypercardioid	target (m^2) [4.3, 4.6]x[4.3, 4.6]x 1.3 reference (m) (0.718, 4.169, 2.046)	uniform, 1cm (max)	961 (max)	200 (ms)	100
valid				random	100		10
test				uniform, 0.1cm (max)	89640 (max)		

Given fixed room and two microphones, training position are selected as 1cm interval (maximal) on 30cm X 30cm xy plane. 100 validation positions are randomly chosen inside training xy-plane. Test position are selected as 0.1cm interval (maximal) on same plane. The number of maximal position are 961, 100, 89640 for train, valid, and test respectively. We also vary training sampling interval to investigate how dense spatial sampling is required to train generalizable estimation models. The reference position is selected randomly outside range of source positions.

Note that representative directivity of the mic is given in Figure 4.23. We select mic directivity as the hypercardioid to expect mic enhances direct path while suppressing reverberant path.

The number of source per each position is 100, 10, 10. Each sources are disjoint so that we can evaluate source-independent performance. To alleviate frequency absence problem, we try to select phonetically balanced speech training set. We randomly select 100 sentence from 50 male, 50 female speakers.

The reference position is randomly chosen at least 1m away from the target position range.

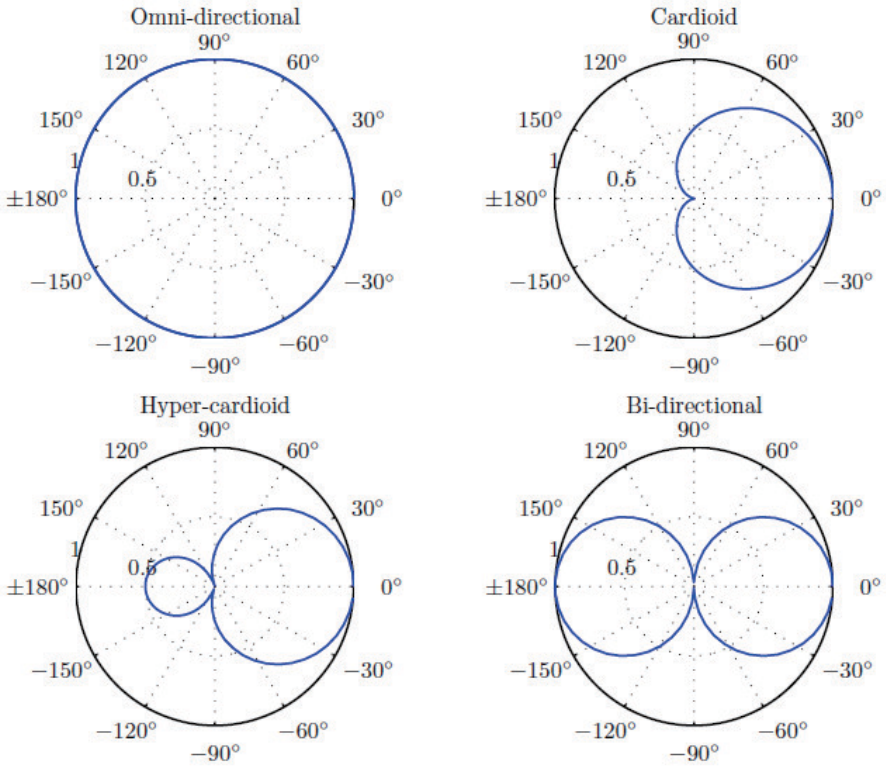


figure 4.23: Representative types of mic directivity. We use hypercardiodi mic.

Hyperparameter

The hyperparameters for input (IMR feature) are window size (=#FFT) and downsample rate. The large size window has lower source residual, however, it may estimate unreliable value for some frequencies due to high resolution of frequency. Moreover, high resolution of frequency increases the number of parameters, since the enhancement model has separate parameter per frequency. The large downsampling rate may generate a smooth response, however, it may lose information.

For the enhancement model, the number of layers, hidden neuron, and size of frequency context window (L_f) is related. The optimal L_f is expected to be related to a frequency resolution of the IMR feature. Table 4.3 list all the hyperparameters to consider.

table 4.3: Hyperparameter searched in experiment.

Input (intermic ratio)	Model (freq-wise complex MLP)
window size $= \{2^9, 2^{10}, 2^{11}, 2^{12}, 2^{13}\}$ downsample rate $= \{1, 2^1, 2^2, 2^3, 2^4\}$	network size (per freq) = {3L x 64H, 4L x 128H, 5L x 256H} frequency context window = {0, 100Hz, 180Hz, 260Hz}

All network parameters are initialized uniformly on [-0.01, 0.01]. The network is optimized with Adam optimizer, and batch size 8, learning rate 0.0001. The test performance is evaluated on model with the lowest validation loss among 100 epochs. We search the task weight of reference position regularization from 1, 0.1, and 0.01.

4.4.3 Result: Source-independence and stationarity

To understand learned representation on neural network, we visualize learned representation of. Figure 4.24 show visualization scenario. We have 4 different sources with 2 same mixing conditions each.

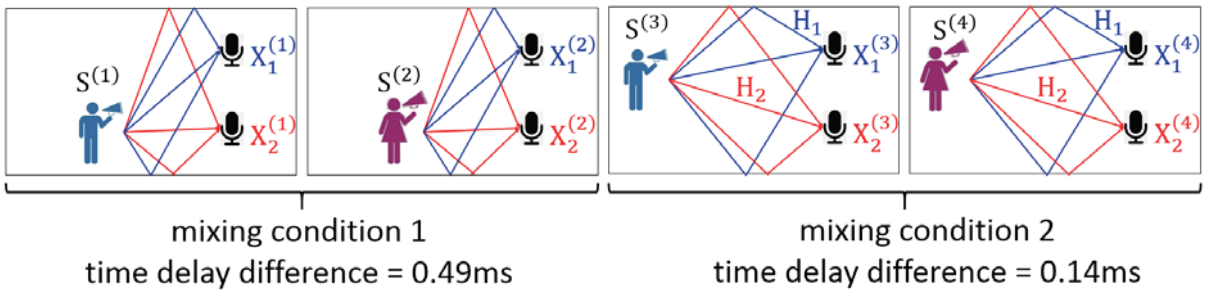


figure 4.24: Visualization of 4 scenarios: 2 mixing environments and 2 sources.

Figure 4.25 show visualization of learned representation on given scenario. Each 4 rows have input of network/output of network/enhanced speech/clean speech respectively.

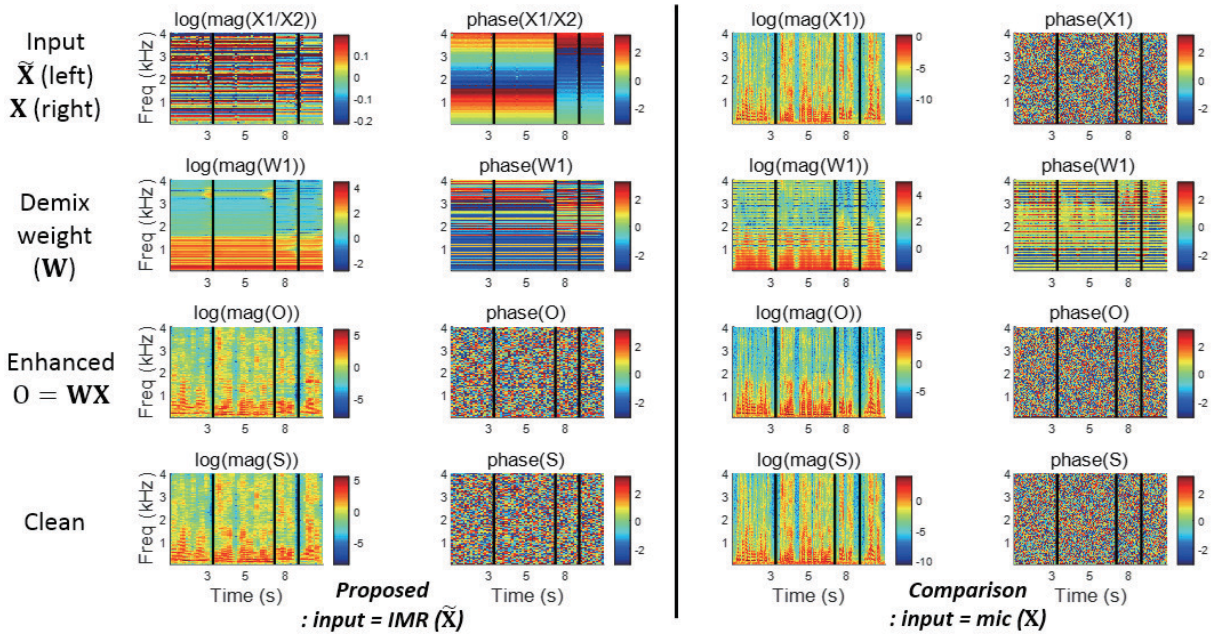


figure 4.25: Visualization of learned representation.

The first half of columns are results from the proposed method (i.e., using IMR input to estimate W). And the other half of columns are results of the comparison model (i.e., using Mic input to estimate W). Both achieve 19.80dB and 16.22dB SDR respectively.

We can see that the proposed IMR and demixing weight estimated by IMR seems similar within utterance 1/2 and 3/4. Note that utterance 1/2 and 3/4 are generated from the same mixing environment. This result implies that IMR and W(IMR) is source-independent and stationary.

However, when using mic input, the estimated demixing weight seems highly correlated with the source, which is definitely not a source-independent.

4.4.4 Result: Non-uniqueness of demixing weight

Figure 4.26 show how much demixing weight (W) vary depending on model. Each sub-image represents demixing weight (W) calculated at every 1cm interval of 30cm x 30cm position. We calculate this result with the model trained with 5cm interval positions. The positions seen during model training are marked as black X in each sub-image.

When reference regularization is not used, the demixing weight can vary depending on neural network parametrization. We use 6 different models by:

- size of neural network: 3Lx64H, 4Lx128H, 5Lx256H (L: #layer, H: #hidden neuron)
- weight initialization variance: 0.1, 0.01

The blue, left part of Figure 4.26 show W average with different models, and the blue-right part is the variance across different models. Compared to mean, variance is non-negligible across 6 models.

However, when we use reference position regularization (red part), W is uniquely determined and converges towards ground-truth W (green part). On training position, estimated W resemble ground-truth W, and the difference is observed when position becomes far from training position.

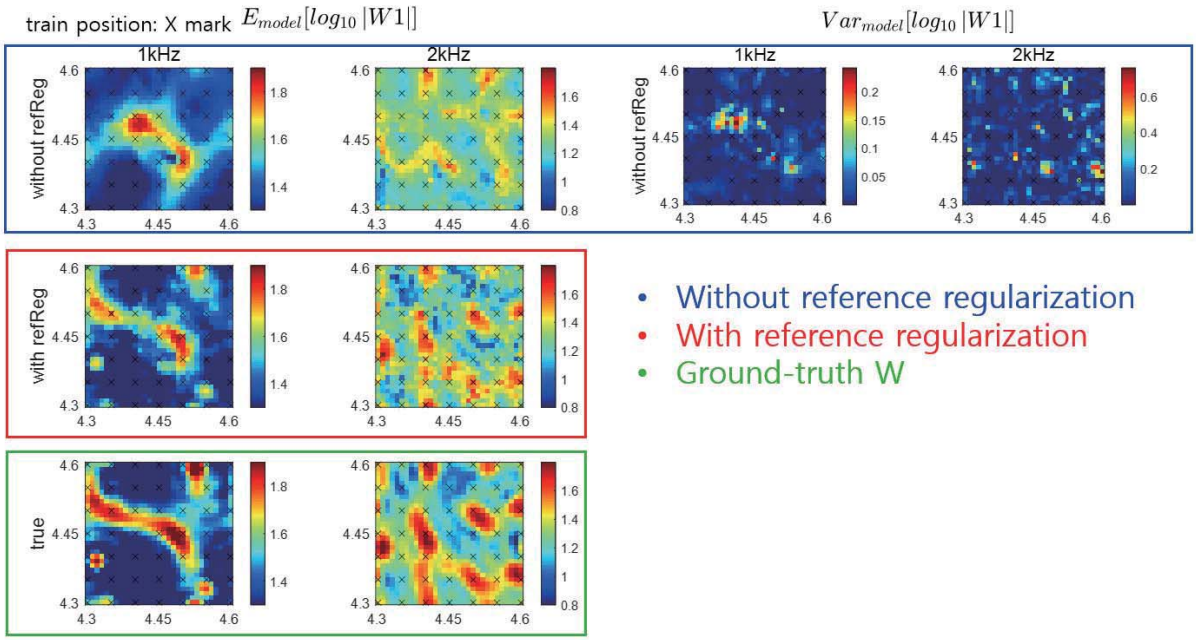


figure 4.26: Visualization of demixing weight (W) with/without reference regularization.

4.4.5 Result: Position/Source robustness of SDR

Figure 4.27 show SDR evaluation on several positions and sources. There are positions with 1cm interval on 30cm X 30cm xy plane. Among every positions, training is done with 5cm interval and marked with black X in sub-images. We evaluate such metrics with 4 different models:

- IMR ->W: input = intermic-ratio, output = demixing weight
- IMR ->W+refIReg: input = intermic-ratio, output = demixing weight, with reference position regularization
- X ->S: input = reverberant mic, output = clean speech
- X ->W: input = reverberant mic, output = demixing weight, with reference position regularization

The first two models are the proposed method, and others are the baseline method.

In the left part of the Figure 4.27, we compute the SDR average across different sources. We can compare the position robustness of each model from this metric. In overall, baseline models underfit, and the proposed method overfit (i.e., high SDR on training position and SDR degraded position become far from training point). However, using reference regularization alleviate degradation of SDR near the training point. The overall performance is summarized in the Table 4.4.

table 4.4: Average SDR of 4 different models on training/test source/position.

	train	test
reverberant	4.71	4.24
IMR->W	25.24	19.80
IMR->W + refReg	23.16	21.38
X->W	17.80	16.22
X->W	14.56	13.54

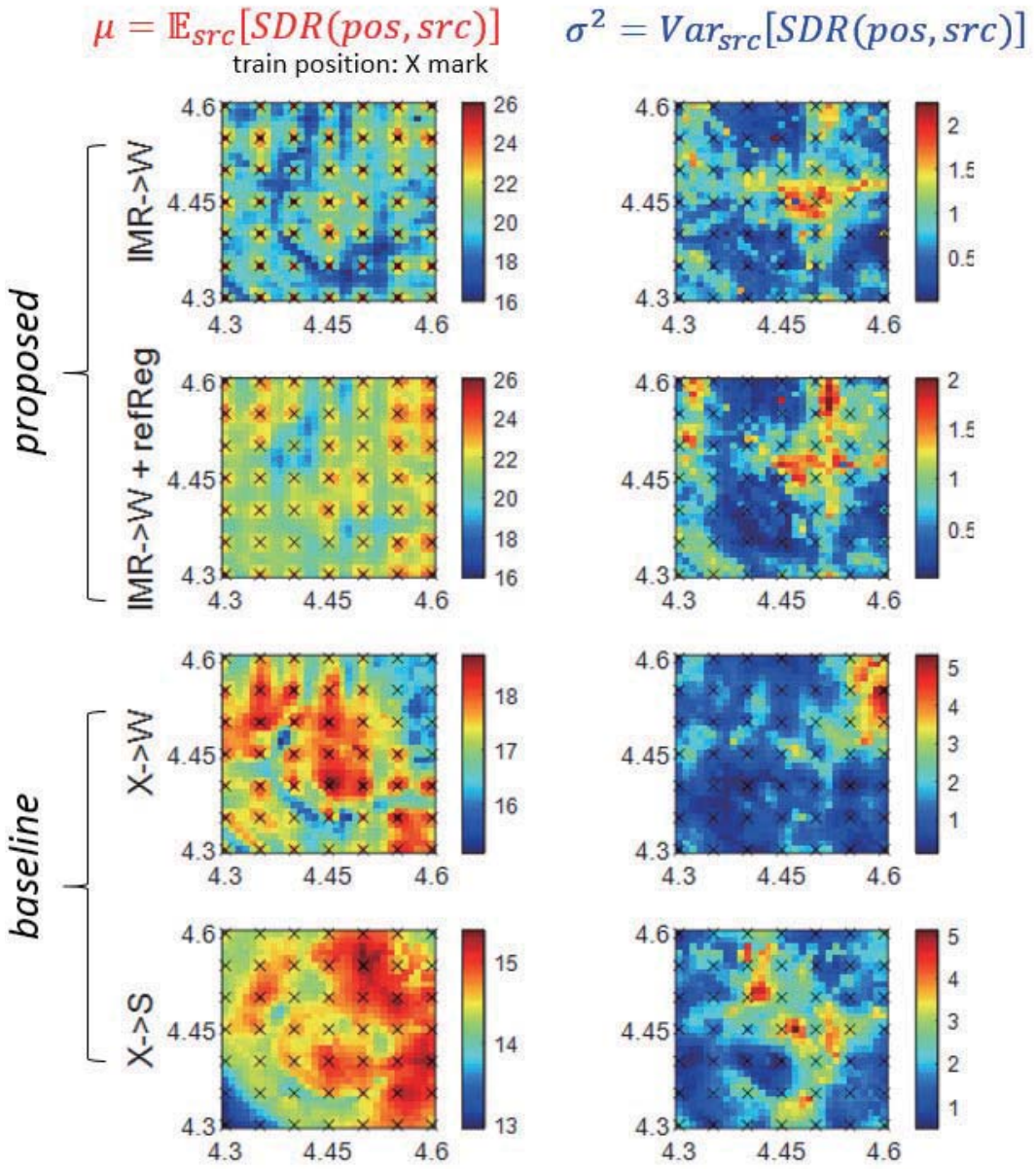


figure 4.27: Mean/variance over source for SDR evaluated on several position and sources. The evaluation is done with 4 different models.

In the right part of the Figure 4.27, we can see the variance of SDR over different sources. We can observe that the proposed methods show a smaller variance than the baseline models. This is because the proposed method is designed to have source independent input and output while the baseline model experience different input and output for every different source.

We quantitatively investigate several statistics of interests:

- average SDR^S :$\mathbb{E}_{p,s}[SDR^S_{p,s}]$
- relative source variance :$Var_s[\mathbb{E}_p[SDR^S_{p,s}]] / \mathbb{E}_{p,s}[SDR^S_{p,s}]$
- average SDR^C :$\mathbb{E}_p[SDR^C_p]$

- relative position sensitivity :$\mathbb{E}_p[|\frac{\frac{\partial SDR_p^C}{\partial p}}{SDR_p^C}|]$

where s, p, SDR^S, SDR^C are source, position, source-dependent SDR, and source-independent SDR respectively.

The models are represented with the following symbols in Table 4.5.

table 4.5: Description of compared models.

Category	Symbol	In->Out	NN	Loss
baseline	b_1	X->S	DCUnet[94]	$\|S - \hat{S}\|^2$
	b_2	X->W	DCUnet[94]	$\|S - \hat{S}\|^2$
proposed	p_1	IMR->W	DCUnet[94]	$\|S - \hat{S}\|^2$
	p_{12}	IMR->W	fwcMLP	$\|S - \hat{S}\|^2$
	p_{123}	IMR->W	fwcMLP	$\|S - \hat{S}\|^2 + \lambda\|\hat{S}^r\|^2$
oracle	o	RTF->W	fwcMLP	$\|W - \hat{W}\|^2$

Figure 4.28 show above statistics on 5 different models. The average SDR is higher in order of oracle, proposed and baseline models. The source variance is lower in the order of proposed models and baseline models. This is because the proposed models are designed having source-independent input and output so the type of source barely affects learning. On the other hand, the baseline models are affected by the type of source. The relative position sensitivity is lower in order of the oracle model, proposed models, and baseline models.

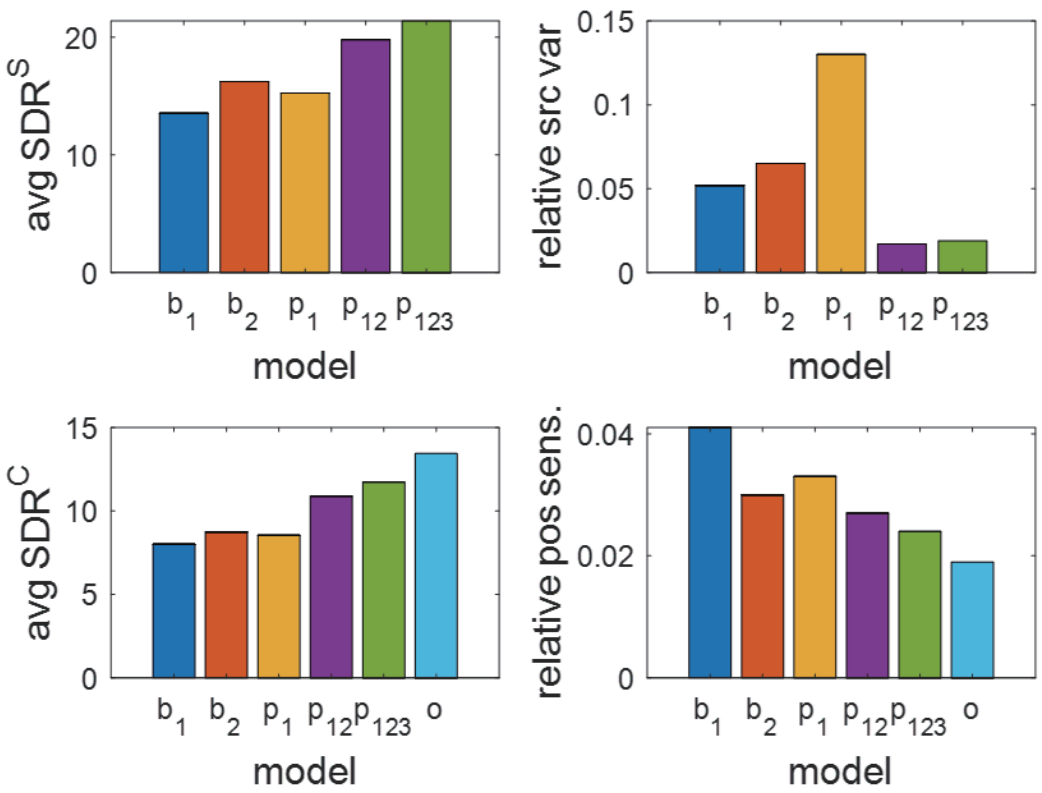


figure 4.28: 4 statistics of interests in SDR, evaluated on 6 different models

4.4.6 Result: Varying training position and source

Figure 4.29 show the effects of varying training number of position and number of source per position. The dotted/solid line are train/test performance respectively.

The left part of figure 4.29 show effects of changing sampling interval on training data: 1cm, 2cm, 5cm, and 10cm. The corresponding number of positions are 961, 256, 49, and 16. By increasing the sampling interval (or decreasing the number of the position), the models overfit to a training position. Interestingly, we can observe that using reference position regularization reduces overfitting when we sample position sparsely (i.e., the yellow curve is superior on small sampling interval while the purple curve is superior on large sampling interval).

The right part of figure 4.29 show effects of changing number of source per position: 1, 5, 100. When decreasing the number of sources per position from 100 to 1, the baseline models (X ->S, X ->W) largely overfit. We conjecture such large overfit comes from source-dependent training of baseline models. On the other hand, proposed models relatively do not affect by the number of the source. The performance gap between oracle and proposed models becomes smaller when the number of the source increases. We conjecture that an increasing number of sources alleviate *frequency absence problem* and improve learning on every frequency range.

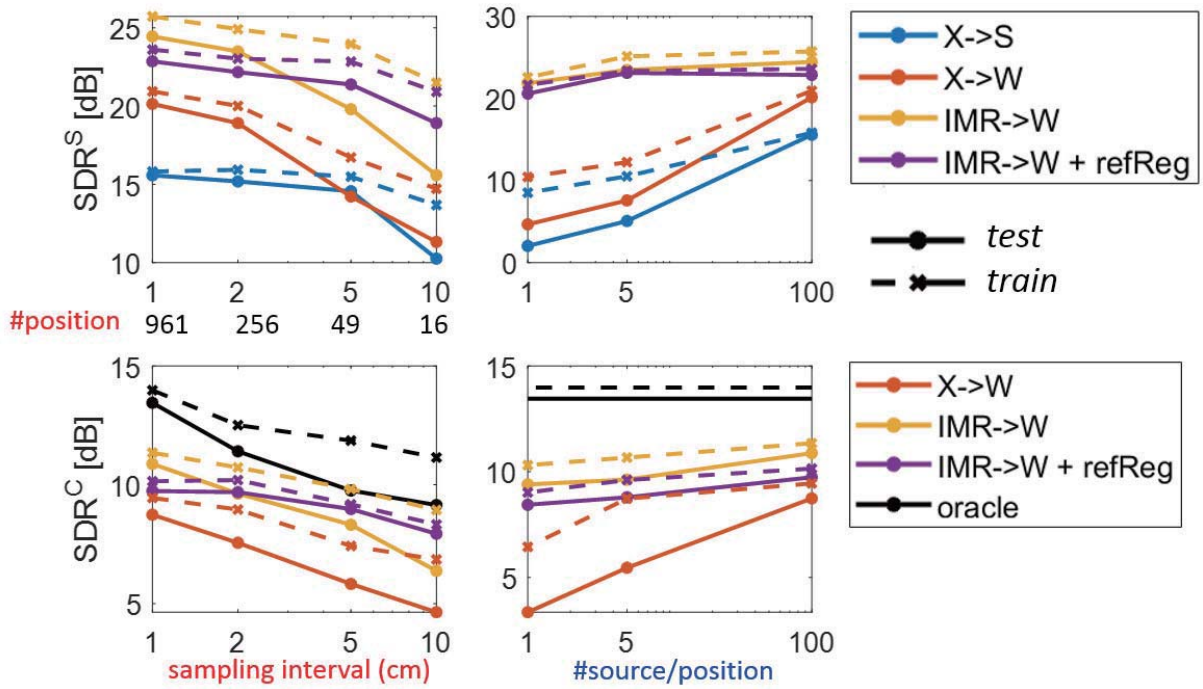


figure 4.29: The effects of varying number of position and the number of position per position for training data.

4.4.7 Result: Effects of hyperparameter

We have baseline model with following settings.

- input: intermic ratio
- neural network type: complex
- frequency parameter: separated
- architecture size: 4Lx128H
- δf: 100Hz
- number of mic: 2

This architecture has 2.15M number of parameters, and achieves 19.80dB SDR on test set. Note that reverberant SDR is 4.24dB. Figure 4.30 show ablation study to empirically assess effects of each settings in the baseline.

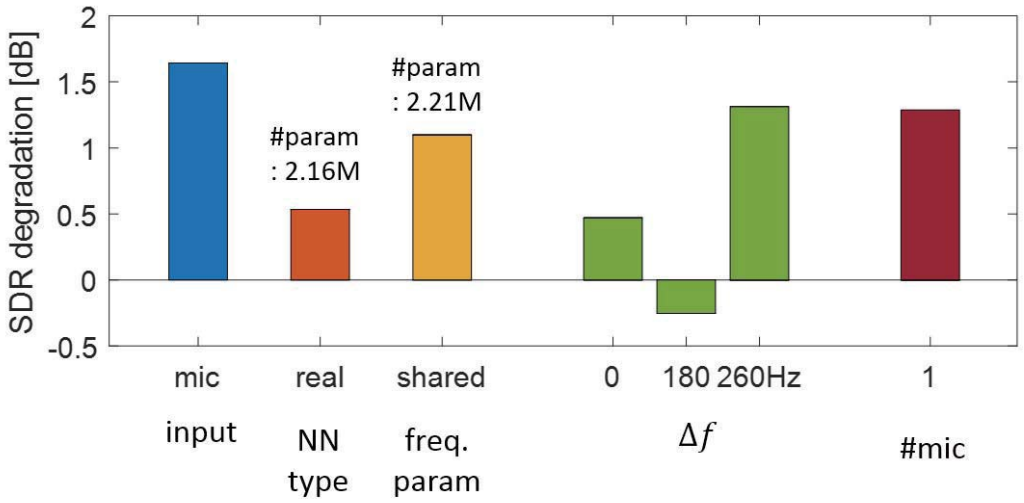


figure 4.30: The effects of each settings in the proposed system.

Especially, window size and frequency resolution are important and correlated hyperparameters. Figure 4.31 show effects of different window size and frequency resolution on performance. Generally, SDR increases with large window size, since multiplicative transfer function approximation becomes exact. And SDR increases with smaller frequency resolution since it can make IMR less noisy and reduce the number of parameters. However, too small frequency resolution degrades the performance.

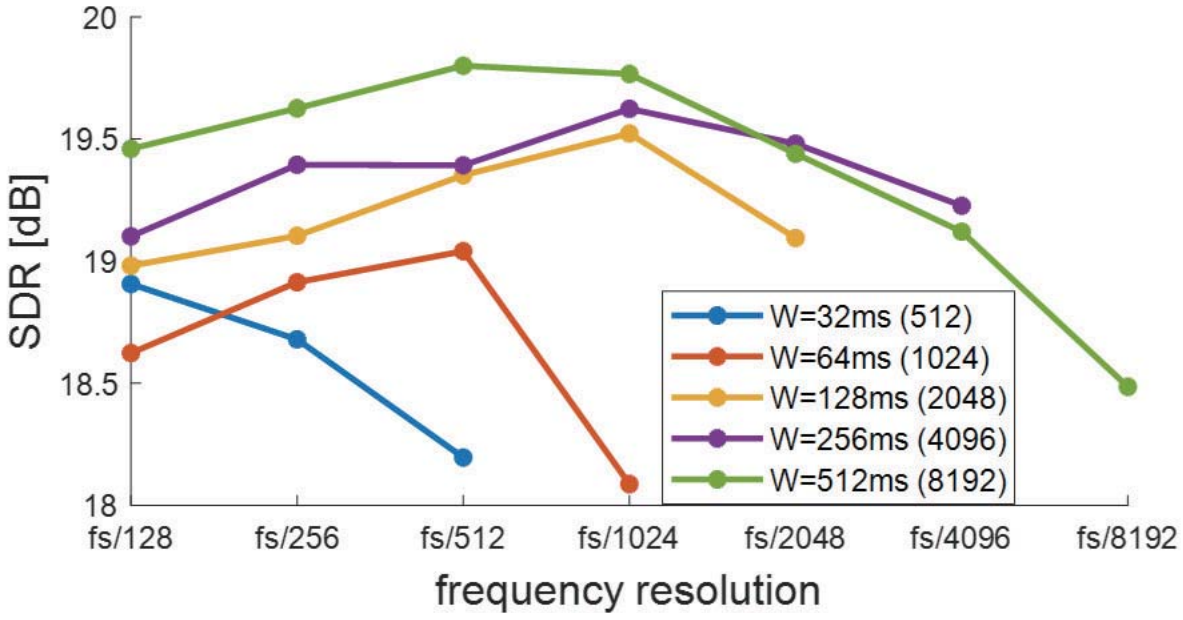


figure 4.31: The effects of window size and frequency resolution on the performance.

4.5 Conclusion

In this work, we propose a de-reverberation model which are robust to source and position variation. For robustness to source variation, we propose using source-independent input/output: inter-mic ratio and demixing weight. The source-independence holds for analysis window size much larger than the length of the impulse response. For position-robustness, we propose a frequency-wise complex multi-layer perceptron (fwcMLP) for the parametrization of the enhancement model and reference position regularization. The fwcMLP learn different position sensitivity per frequency by having independent complex parameters per frequency bin. Reference position regularization provides a unique ground-truth target for demixing weight to reduce training variance with respect to neural network size, initialization, and choice of training samples. We tested our approach on the simulated reverberant database, and the performance is evaluated on various position and sources in a room. We understand source-independence of the inter-mic ratio and demixing weight by visualizing several speeches with the same mixing environment. Compared to the baseline model, the proposed model improves SDR in terms of position-robustness, source-robustness. When reducing the training number of position, we empirically verify reference regularization reduce overfitting when sparse training position is provided. When reducing the number of sources per position, we empirically verify baseline models (source-dependent) severely overfit while the proposed model degraded small. The gap between the proposed model and the oracle model becomes smaller when the number of speech per position increases.

Chapter 5. Conclusion

5.1 Summarization

In this thesis, we improve generalization on the unseen condition of two front-end modules of spoken dialog system: language understanding and speech recognition.

For language understanding, we address rare word and lack of context problems when estimating sentence representation. These two different problems can be addressed simultaneously by learning hierarchical composition rule exists in the dialog. We propose **hierarchical composition recurrent network (HCRN)** to learn hierarchical composition rule in the dialog and **hierarchy-wise training algorithm** to improve the convergence of the HCRN training. Our method achieves a 22.7% test error rate on dialog act classification performance on the SWBD-DAMSL database, achieves a lower error rate than the partially compositional model (24.9%).

For environmentally robust speech recognition, we address two subproblems in speech enhancement.

The first problem is the usage of clean as the target of the speech enhancement. However, the clean target is only obtainable for simulated data, and not available for corrupted speech recorded from the real environment. Therefore, we propose an alternative learning algorithm: **reconstruction/acoustic and adversarial supervision (RAS/AAS)** designed for improving speech intelligibility and speech recognition performance respectively. The proposed method was tested on two datasets: Librispeech + DEMAND and CHiME-4. By visualizing the enhanced speech with different supervision combinations, we understand the pros/cons of each supervision. Compared to the enhancement method using clean speech target, AAS achieve lower word error rate although the distance from the clean speech is higher.

The second problem is robustness to source and position variation for speech enhancement. For robustness to source variation, we employ source-independent input/output: **intermic-ratio and demixing weight** in large-size window. For position-robustness, we propose **frequency-wise complex multilayer perceptron** and **reference position regularization**. The fwcMLP learn different position sensitivity per frequency by having independent complex parameters per frequency bin. Reference position regularization provides a unique ground-truth target for demixing weight to reduce training target variance across model size, initialization, and choice of training data. We tested our approach on the simulated reverberant database, and the performance is evaluated on various positions and sources in a room. Compared to the baseline model, the proposed model improves SDR in terms of position-robustness, source-robustness.

5.2 Future work

Learning hierarchical composition rule in dialog

Currently, the concept of hierarchical composition from character to dialog is only tested on a dialog act classification task. This technique needs to be studied on more general tasks such as language model, and natural language generation. Moreover, saving dialog context as characters makes a very long sequence. Therefore, efficient information reading/writing algorithms, such as memory/attention mechanisms, need to be further investigated.

Clean-free speech enhancement

It is unclear that what kinds of difference exist between enhanced speech optimized for signal-to-distortion ratio and word error rate. Identifying this difference may help to devise enhancement optimized for speech recognition. Moreover, the proposed acoustic supervision is sensitive to acoustic model performance. Therefore, learning algorithm to improve both enhancement and acoustic model performance needs to be investigated.

Source/Position speech enhancement

The proposed method needs to be modified for a more general case. Firstly, the inter-mic ratio is not source-independent anymore when the number of the source is more than 1. Also, it is unclear that the model can work for varying room, mic position or even moving source. There is another issue for this method applied to the front-end of speech recognition. The model has a source-independent property when the analysis window is much larger than the length of the impulse response. However, using a large analysis window is not suitable for speech recognition which uses a small window (e.g., 25ms) in general. Therefore, pursuing source-independence on small window settings needs to be further investigated.

Bibliography

[1] George Dahl Abdel-rahman Mohamed and Geoffrey Hinton. Deep Belief Networks for phone recognition. In *Neural Information Processing System*, 2009.

[2] Quoc Le Oriol Vinyals. A Neural Conversational Model. In *International Conference on Machine Learning Workshop*, 2015.

[3] Heiga Zen Karen Simonyan Oriol Vinyals Alex Graves Nal Kalchbrenner Andrew Senior Koray Kavukcuoglu Aaron van den Oord, Sander Dieleman. WaveNet: A Generative Model for Raw Audio. In *arXiv preprint arXiv : 1609.03499*, 2016.

[4] Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks. In *NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, 2010.

[5] Y Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Nautral Language Processing (EMNLP)*, pages 1746–1751, 2014.

[6] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014.

[7] Gehring Jonas, Auli Michael, Grangier David, Yarats Denis, and N.Dauphin Yann. Convolutional Sequence to Sequence Learning. In *arXiv preprint arXiv : 1705.03122*, 2017.

[8] Jan a. Botha and Phil Blunsom. Compositional Morphology for Word Representations and Language Modelling. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[9] Alexandra Birch Rico Sennrich, Barry Haddow. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[10] Ramesh Nallapati Bowen Zhou Yoshua Bengio Caglar Gulcehre, Sungjin Ahn. Pointing the Unknown Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[11] Hang Li Victor O.K. Li Jiatao Gu, Zhengdong Lu. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[12] Piotr Bojanowski, Armand Joulin, and Tomas Mikolov. Alternative structures for character-level RNNs. In *arXiv preprint arXiv : 1511.06303*, 2016.

[13] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A Clockwork RNN. In *Proceedings of The 31st International Conference on Machine Learning (ICML)*, volume 32, pages 1863–1871, 2014.

[14] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37, pages 2067–2075, 2015.

[15] Chung Junyoung, Ahn Sungjin, and Bengio Yoshua. Hierarchical Multiscale Recurrent Neural Network. In *International Conference of Learning Representation (ICLR)*, 2017.

[16] Duyu Tang, Bing Qin, and Ting Liu. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, 2015.

[17] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *Special Track on Cognitive Systems at AAAI*, 2016.

[18] Wang Ling, Tiago Luis, Luis Marujo, Ramon Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. In *Proceedings of Empirical Methods on Natural Language Processing (EMNLP)*, pages 1520–1530, 2015.

[19] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-Aware Neural Language Models. In *Proceedings of Association for the Advancement of Artificial Intelligence*, 2016.

[20] Lee Jason, Cho Kyunghyun, and Hofmann Thomas. Fully Character-Level Neural Machine Translation without Explicit Segmentation. In *arXiv preprint arXiv : 1610.03017*, 2017.

[21] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[22] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning Long-Term Dependencies with Gradient Descent is Dicfficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[23] Sepp Hochreiter. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6:107–116, 1998.

[24] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[25] Nitish Srivastava. Unsupervised Learning of Video Representations using LSTMs. In *Proceedings of International Conference of Machine Learning 2015*, volume 37, 2015.

[26] A Stolcke, K Ries, N Coccaro, E Shriberg, R Bates, D Jurafsky, P Taylor, R Martin, C V Ess-Dykema, and M Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373, 2000.

[27] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *arXiv preprint arXiv : 1412.3555v1*, 2014.

[28] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An Empirical Exploration of Recurrent Network Architectures. In *Proceedings of the 32nd International Converenfe on Machine Learning (ICML)*, pages 171–180, 2015.

[29] Matthew D Zeiler. ADADELTA: An Adaptive Learning Rate Method. In *arXiv preprint arXiv : 1212.5701*, 2012.

[30] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[31] Nal Kalchbrenner and Phil Blunsom. Recurrent Convolutional Neural Networks for Discourse Compositionality. In *ACL WS on Continuous Vector Space Models and their Compositionality*, pages 119–126, 2013.

[32] Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. A Latent Variable Recurrent Neural Network for Discourse Relation Language Models. In *arXiv preprint arXiv : 1603.01913*, 2016.

[33] Björn Gambäck, Fredrik Olsson, and Oscar Täckström. Active Learning for Dialogue Act Classification. In *Proceedings of Interspeech 2011*, 2011.

[34] Chanwoo Kim, Ananya Misra, Kean Chin, Thad Hughes, Arun Narayanan, Tara Sainath, and Michiel Bacchiani. Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2017.

[35] Gregoire Lafay, Emmanouil Benetos, and Mathieu Lagrange. Sound event detection in synthetic audio: Analysis of the dcase 2016 task results. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2017.

[36] Youssef El Baba, Andreas Walther, and Emanuel A.P. Habets. 3D room geometry inference based on room impulse response stacks. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 2018.

[37] Emmanuel Vincent, Shinji Watanabe, Aditya Arie Nugraha, Jon Barker, and Ricard Marxer. An analysis of environment, microphone and data simulation mismatches in robust speech recognition. *Computer speech and Language*, 46:535–557, 2017.

[38] Jean-Claude Junqua, Steven Fincke, and Ken Field. The Lombard effect: a reflex to better communicate with others in noise. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, 1999.

[39] Steven F. Boll. Suppression of acoustic noise in speech using spectral substraction. In *Acoustic, Speech and Signal Processing*, 1979.

[40] Pascal Scalart and Jozue Vieira Filho. Speech enhancement based on a priori signal to noise estimation. In *Proceedings of the Acoustics, Speech, and Signal Processing*, 1996.

[41] Shigeki Matsuda Chiori Hori Xugang Lu, Yu Tsao. Speech Enhancement Based on Deep Denoising Autoencoder. In *Interspeech*, 2013.

[42] Takahiro Shinozaki Yasuo Horiuchi Shingo Kuroiwa Takaaki Ishii, Hiroki Komiyama. Reverberant Speech Recognition Based on Denoising Autoencoder. In *Interspeech*, 2013.

[43] Jinwon Lee Se Rim Park. A Fully Convolutional Neural Network for Speech Enhancement. In *arXiv preprint arXiv : 1609.07132*, 2016.

[44] Florian Eyben Gerhard Rigoll Martin Wollmer, Bjorn Schuller. Combining Long Short-Term Memory and Dynamic Bayesian Networks for Incremental Emotion-Sensitive Artificial Listening. In *IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING*, 2010.

[45] Felix Weninger Bjorn Schuller Gerhard Rigoll Martin Wollmer, Zixing Zhang. FEATURE ENHANCEMENT BY BIDIRECTIONAL LSTM NETWORKS FOR CONVERSATIONAL SPEECH RECOGNITION IN HIGHLY NON-STATIONARY NOISE. In *ICASSP*, 2013.

[46] DeLiang Wang Soundararajan Srinivasan, Nicoleta Romanm. Binary and ratio time-frequency masks for robust speech recognition. In *Speech Communication*, 2016.

[47] Donald S. Williamson and DeLiang Wang. Time-Frequency Masking in the Complex Domain for Speech Dereverberation and Denoising. In *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 2017.

[48] Yuxuan Wang and DeLiang Wang. Towards scaling up classification-based speech separation. In *IEEE Transactions on Audio, Speech, and Language Processing*, 2013.

[49] Yongqiang Wang Michael L. Seltzer, Dong Yu. An investigation of deep neural networks for noise robust speech recognition. In *ICASSP*, 2013.

[50] John H.L. Hansen Seyedmahdad Mirsamadi. A Study on Deep Neural Network Acoustic Model Adaptation for Robust Far-field Speech Recognition . In *Interspeech*, 2015.

[51] Tian Tan Kai Yu Yanmin Qian, Mengxiao Bi. Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition. In *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 2016.

[52] John H.L. Hansen Seyedmahdad Mirsamadi. A Study on Deep Neural Network Acoustic Model Adaptation for Robust Far-field Speech Recognition. In *Interspeech*, 2015.

[53] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio. A network of deep neural networks for Distant Speech Recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[54] Santiago Pascual, Antonio Bonafonte, and Joan Serra. SEGAN: Speech Enhancement Generative Adversarial Networks. In *Interspeech*, 2017.

[55] Chris Donahue, Bo Li, and Rohit Prabhavalkar. Exploring Speech Enhancement with Generative Adversarial Networks for Robust Speech Recognition. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[56] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised Image-to-Image Translation Networks. In *Neural Information Processing Systems*, 2017.

[57] Chin Cheng Hsu, Hsin Te Hwang, Yi Chiao Wu, Yu Tsao, and Hsin Min Wang. Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2017.

[58] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. In *arXiv preprint arXiv : 1411.1784*, 2014.

[59] Jun Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[60] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

[61] Hanock Kwak and Byoung-Tak Zhang. Ways of Conditioning Generative Adversarial Networks. In *Workshop on Neural Information Processing Systems*, 2016.

[62] Bae Un-Min and Lee Soo-Young. Combining ICA and Top-Down Attention for Robust Speech Recognition. *Advances in Neural Information Processing Systems*, 2001.

[63] Luke Metz David Berthelot, Thomas Schumm. BEGAN: Boundary Equilibrium Generative Adversarial Networks. In *Neural Information Processing System*, 2017.

[64] Lee Chang-Hoon and Lee Soo-Young. Noise-Robust Speech Recognition Using Top-Down Selective Attention With an HMM Classifier. *IEEE Signal Processing Letters*, 2007.

[65] Kim Ho-Gyeong, Lee Hwaran, Kim Geonmin, Oh Sang-Hoon, and Lee Soo-Young. Rescoring of N-Best Hypotheses Using Top-down Selective Attention for Automatic Speech Recognition. *IEEE Signal Processing Letters*, 2018.

[66] Faustino Gomez Jurgen Schmidhuber Alex Graves, Santiago Fernandez. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *International Conference on Machine Learning*, 2006.

[67] Zhao Junbo, Mathieu Michael, and Yann LeCun. Energy-based Generative Adversarial Networks. In *International Conference on Learning Representation*, 2017.

[68] Simon King Christophe Veaux, Junichi Yamagishi. The voice bank corpus: Design, collection and data analysis of a large regional accent speech database. In *Int. Conf. Oriental COCOSDA, held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation*, 2013.

[69] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent. The diverse environments multichannel acoustic noise database: A database of multichannel environmental noise recordings. In *The Journal of the Acoustical Society of America, vol. 133, no. 5*, 2013.

[70] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe. The third CHiME speech separation and recognition challenge: Analysis and outcomes. *Computer Speech and Language*, 2017.

[71] CHiME-4 Acoustic simulation baseline.

[72] Du Jun, Tu Yan-Hui, Sun Lei, Ma Feng, Wang Hai-Kun, Pan Jia, Liu Cong, Chen Jing-Dong, and Lee Chin-Hui. The USTC-iFlytek System for CHiME-4 Challenge. In *Proceeding on CHiME*, 2016.

[73] Dat Tran, Huy, Zheng Terence Ng, Wen, Sivadas Sunil, Tuan Luong, Trung, and Dung Tran, Anh. The I2R System for CHiME-4 Challenge. In *Proceeding on CHiME*, 2016.

[74] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representation*, 2015.

[75] Rishita Anubhai Jingliang Bai Eric Battenberg Carl Case Jared Casper Bryan Catanzaro Qiang Cheng Guoliang Chen Jie Chen Jingdong Chen Zhijie Chen Mike Chrzanowski Adam Coates Greg Diamos Ke Ding Niandong Du Erich Elsen Jesse Engel Weiwei Fang Linxi Fan Christopher Fougner Liang Gao Caixia Gong Awni Hannun Tony Han Lappi Vaino Johannes Bing Jiang Cai Ju Billy Jun Patrick LeGresley Libby Lin Junjie Liu Yang Liu Weigao Li Xiangang Li Dongpeng Ma Sharan Narang Andrew Ng Sherjil Ozair Yiping Peng Ryan Prenger Sheng Qian Zongfeng Quan Jonathan Raiman Vinay Rao Sanjeev Satheesh David Seetapun Shubho Sengupta Kavya Srinet Anuroop Sriram Haiyuan Tang Liliang Tang Chong Wang Jidong Wang Kaifu Wang Yi Wang Zhijian Wang

Zhiqian Wang Shuang Wu Likai Wei Bo Xiao Wen Xie Yan Xie Dani Yogatama Bin Yuan Jun Zhan Zhenyao Zhu Dario Amodei, Sundaram Ananthanarayanan. Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin . In *International Conference on Machine Learning*, 2016.

[76] Daniel Griffin and James Lim. Signal estimation from modified short-time fourier transform. In *IEEE Transactions on acoustics, speech, and signal processing*, 1984.

[77] Gabriel Pereyra, Ying Zhang, and Yoshua Bengio. Batch Normalized Recurrent Neural Networks. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[78] Shiyu Zhou, Yuanyuan Zhao, Shuang Xu, and Bo Xu. Multilingual recurrent neural networks with residual learning for low-resource speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2017.

[79] Scalart Pascal and Filho Jozue, Vieira. Speech enhancement based on a priori signal to noise estimation. In *International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 1996.

[80] Daniel Povey Sanjeev Khudanpur Vassil Panayotov, Guoguo Chen. Librispeech: an asr corpus based on public domain audio books. In *ICASSP*, 2015.

[81] Daniel Povey. Librispeech language model. 2015.

[82] Jonathan H. Clark Kenneth Heafield, Ivan Pouzyrevsky and Philipp Koehnr. Scalable Modified Kneser-Ney Language Model Estimation. In *ACL*, 2013.

[83] Gabriel Synnaeve. WER ARE WE. 2018.

[84] Ronan Collobert Vitaliy Liptchinsky, Gabriel Synnaeve. Letter-Based Speech Recognition with Gated ConvNets. In *arXiv preprint arXiv : 1609.03193*, 2017.

[85] Yekutiel Avargel and Israel Cohen. System Identification in the Short-Time Fourier Transform Domain With Crossband Filterings. In *IEEE Transactions on Audio, Speech, and Language Processing*, 2007.

[86] J. Capon. High-resolution frequency-wavenumber spectrum analysis. In *IEEE*, 1969.

[87] Reinhold Haeb-Umbach Ernst Warsitz. Blind acoustic beamforming based on generalized eigenvalue decomposition. In *IEEE transactions on audio, speech, and language processing*, 2007.

[88] Joao Felipe Santos and Tiago H. Falk. Speech Dereverberion With Context-Aware Recurrent Neural Networks. In *Interspeech*, 2018.

[89] Kehuang Li Zhen Huang Sabato Marco Siniscalchi Tong Wang Chin-Hui Lee Bo Wu, Minglei Yang. A reverberation-time-aware DNN approach leveraging spatial information for microphone array dereverberation. In *EURASIP journal on advances in signal processing*, 2017.

[90] Yan Zhao, DeLiang Wang, Buye Xu, and Tao Zhang. Late Reverberation Suppression Using Recurrent Neural Networks with Long Short-Term Memory. In *ICASSP*, 2018.

[91] Ju Lin, Sufeng Niu, Zice Wei, Xiang Lan, Adriaan J. van Wijnggaarden, Melissa C. Smith, and Kuang-Ching Wang. Speech enhancement using forked GAN with spectral subtraction. In *Interspeech*, 2019.

[92] Hamidreza Baradaran Kashani, Ata Jodeiri, Mohammad Mohsen Goodarzi, and Iman Sarraf Rezaei. Speech Enhancement via Deep Spectrum Image Translation Network. In *International Conference on Biomedical Engineering*, 2019.

[93] Ori Ernst, Shlomo E. Chazan, Sharon Gannot, and Jacob Goldberger. Speech dereverberation using fully convolutional networks. In *EUSIPCO*, 2018.

[94] Hyeong-Seok Choi, Jang-Hyun Kim, Jaesung Huh, Adrian Kim, Jung-Woo Ha, and Kyogu Lee. Phase-Aware Speech Enhancement with Deep Complex U-Net . In *International Conference on Learning Representation*, 2019.

[95] Donald S. Williamson and DeLiang Wang. Speech dereverberation and denoising using complex ratio masks. In *ICASSP*, 2017.

[96] Zhong-Qui Wang and DeLiang Wang. All-Neural Multi-Channel Speech Enhancement. In *Interspeech*, 2018.

[97] Wolfgang Mack, Soumitro Chakrabarty, Fabian-Robert Stoter, Sebastian Braun, Bernd Edler, and Emanuel A.P. Habet. Single-Channel Dereverberation Using Direct MMSE Optimization and Bidirectional LSTM Networks. In *Interspeech*, 2018.

[98] Taesu Kim, Torbjorn Eltoft, and Te-Won Lee. Independent vector analysis: an extension of ICA to multivariate components. In *International conference on independent component analysis and signal separation*, 2006.

[99] Choong hwan Choi, Wonil Chang, and Soo-Young Lee. Blind source separation of speech and music signals using harmonic frequency dependent independente vector analysis. In *Electronics Letter*, 2012.

[100] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary Evolution Recurrent Neural Network. In *International Conference on Machine Learning (ICML)*, 2016.

[101] Ivo Danihelka, Greg Wayne, Benigno Uria, Nal Kalchbrenner, and Alex Graves. Associative Long Short-Term Memory. In *International Conference on Machine Learning (ICML)*, 2016.

[102] Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, Joao Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep Complex Network. In *International Conference on Learning Representation (ICLR)*, 2018.

[103] Gaetano Fichera. Unification of global and local existence theorems for holomorphic functions of several complex variables. In *Memorie della Accademia Nazionale dei Lincei, Classe di Scienze Fisiche, Matematiche e Naturali*, 1986.

[104] Rakesh Malladi, Don H Johnson, Tandon Nitin Kalamangalam, Giridhar P, and Behnaam Aazhang. Mutual Information in Frequency and Its Application to Measure Cross-Frequency Coupling in Epilepsy. In *IEEE Transactions on Signal Processing Letter*, 2018.

[105] Francois Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[106] Cedric Fevotte Emmanuel Vincent, Remi Gribonval. Performance measurement in blind audio source separation. In *IEEE transactions on audio, speech, and language processing*, 2006.

[107] Jont B Allen and David A Berkley. Image method for efficiently simulating small room acoustics. In *The Journal of the Acoustical Society of America*, 1979.

[108] Allen Pierce. An Introduction to Its Physical Principles and Applications. In *Acoustical Society of America*, 1991.

Summary in Korean

음성대화시스템은 다양한 사용자 질문에 적절한 대답을 내놓기를 요구된다. 학위연구에서는 음성 대화 시스템의 학습 과정에서 경험하지 못한 새로운 문장과 음향환경에서의 일반화 향상에 대한 소문제들을 다루었다.

논문의 첫 번째 부분에서는 신경망을 이용한 문장 표현법의 두 가지 문제를 다루고자 하였다. 문장내 저빈도 단어의 임베딩을 추론하는것과, 주변 문장들의 문맥 정보를 문장 표현에 반영하는 것이다. 상기 문제들을 개선하기 위하여 **계층적 재귀 합성 신경망 (hierarchical composition recurrent network, HCRN)**이 제안되었다. HCRN은 3가지 계층으로 구성되어 각각 글자, 단어, 문장을 입력으로 받아 상위 단위인 단어, 문장, 문맥에 대한 표현을 학습한다. 제안된 모델은 대화 화행 인식 테스크에서 시험되었다. 기존의 단어-문장 계층 모델과 비교하여 글자-단어 계층으로부터 생성된 단어 임베딩은 형태적, 의미적으로 유사한 클러스터를 형성하였다. 문장-문맥 계층의 사용으로 문장의 대화 화행인식 오류율이 생략이 많은 문장에서 크게 감소하였다.

논문의 두 번째 부분에서는 잡음 음성에 대응되는 무잡음 음성을 사용하지 않는 음성 향상 학습법을 목표로 하였다. 실환경에서 수집한 잡음 음성에 대해서는 무잡음 음성이 존재하지 않고 시뮬레이션 방법에서만 사용 가능하기 때문이다. 이를 위하여 **음향 및 대립 지도 (acoustic and adversarial supervision, AAS)**이 제안되었다. 음향 지도는 학습된 음향 모델에 대하여 향상된 음성의 우도를 높일 수 있도록 향상기를 학습시키도록 한다. 이에 따라, 향상된 음성이 음소의 특징을 유지하는데 집중하지만 과적합 현상이 음성에서 왜곡된 특징으로써 나타난다. 대립 지도는 향상된 음성이 무잡음 음성의 일반적인 특징을 가져서 왜곡된 특징이 나타나지 않으나, 모드붕괴에 의해 잡음 음성과 관련이 없는 임의의 무잡음 음성이 생성될 수 있다. 우리는 두 가지 지도의 목적함수를 가중합하여 상호보완적으로 사용하였다. 제안한 방법은 Librispeech+DEMAND와 CHiME-4 데이터에서 평가되었다. 두 가지 지도법으로 학습한 결과를 시각화하여 비교하므로써, 각 지도법의 장단점을 이해할 수 있었다. AAS는 무잡음 음성을 학습의 출력으로 사용하는 방법에 비하여 향상된 음성이 무잡음 음성과의 거리가 멀었으나, 단어 오류율은 낮았다.

논문의 세 번째 부분에서는 음원과 위치에 강인한 음성 향상 문제를 다루었다. 음원에 강인하기 위하여 입출력을 **마이크간 비율 (intermic-ratio)과 디믹싱 가중치 (demixing weight)**으로 사용하여 음성 향상 문제가 음원 독립성을 가지도록 하였다. 분석 윈도우가 임펄스 응답의 길이에 비해 충분히 긴 상황에서 입출력은 음원에 독립적이라는 성질을 가지고 있기 때문이다. 위치에 대한 강인함을 위해 **주파수별 복소 멀티 레이어 퍼셉트론 (frequency-wise multi-layer perceptron)**을 회귀 모델로 사용하였는데, 이는 디믹싱 가중치의 위치에 대한 변화가 저주파에서 고주파로 갈수록 위치에 민감한 특성에 근거하여 설계하였다. 또한, 디믹싱 가중치의 전역 최적해는 유일하게 결정되지 않아서, 회귀 모델의 파라미터 개수 및 초기화 방법, 미니배치 구성에 따라서 매번 변하게 된다. 학습에서 출력이 매번 변하는 현상을 줄이기 위하여 **비교 위치 정규화 (reference position regularization)**가 제안되었다. 제안한 방법은 고정된 방과 마이크에 대하여 음원이 한정된 영역을 움직이는 시뮬레이션 기반의 데이터셋에서 테스트 되었다. 음원종속적인 기존 학습 방법에 비하여, 제안한 음원독립적인 학습 방법은 높은 신호대왜곡비를 기록하였고, 성능 차이는 특히

학습에서 사용한 음원의 개수가 적을 때 컸다. 학습된 모델들은 공통적으로 학습 위치에서 벗어날수록 신호 대왜곡비가 떨어지는 과적합 현상도 있었는데, 과적합 현상은 비교 위치 정규화를 통해서 상당 부분 개선될 수 있었다.

Acknowledgments in Korean

이 논문이 완성되기까지 도움을 주신 모든 분들께 감사를 드립니다. 박사과정은 훌륭한 사람들을 만나고, 연구 능력을 성장시킬 수 있었던 감사한 시간이었습니다.

이수영 교수님, 학부 개별연구 학생때부터 박사학위를 받기까지 많은 것들을 배울 수 있었습니다. 학생이 스스로 문제에 대한 답을 찾아나갈 수 있도록 결과보다는 방법을 알려주시고, 연구/과제/논문으로 성장할 수 있는 기회를 주시며, 바쁜 시간을 쪼개어 매주 연구 미팅을 진행하시어, 교육자로서 연구자로서 모범을 보여 주심에 감사 드립니다. 교수님 속도 많이 썩였던 지도하기 어려웠던 학생이었지만, 끝까지 도와주셔서 박사학위를 받을 수 있었습니다.

김대식 교수님, 김회린 교수님, 신진우 교수님, 오상훈 교수님, 바쁘신 와중에도 논문 심사에 시간 할애해주시고, 보다 나은 논문이 될 수 있도록 건설적인 조언을 해주셔서 감사드립니다.

늘 학생들의 연구실 생활을 지원해주시고 보살펴주셨던 신필호 선생님, 자주 연구 미팅과 친목을 다졌던 KI4AI 연구원 여러분 감사드립니다.

함께 연구실에서 연구하고 우정을 쌓았던 소중한 사람들, 창현형, 원일형, 서연누나, 호경누나, 청안형, 일환형, 병열형, 현아누나, 경호형, 동건, 은수, 보경, 화란, 지현, 지수, 영근, 병근, 성진, 정우 모두 감사합니다.

연락도 자주하지 않고 집안일에 신경쓰지 못한 아들, 동생이었지만 언제나 저를 응원해주신 어머니, 아버지, 형 모두 사랑하고 감사합니다.

학위연구가 완성될 수 있도록 제도적 금전적 지원을 제공해주신 모교 KAIST에도 감사합니다.

Geonmin Kim

CONTACT

Computational NeuroSystem Lab., KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon, South Korea

+82.10.9703.2947
ken.geonmin.kim@gmail.com
github.com/gmkim90

RESEARCH INTERESTS

Dialogue systems, Speech recognition, Speech enhancement
Neural network, Generative model

EDUCATION

Korea Advanced Institute of Science and Technology — Mar. 2013 – Oct. 2019 (expected)
Ph.D, Electrical Engineering,
Advisor: Prof. Soo-Young Lee
GPA: 3.93/4.3

Korea Advanced Institute of Science and Technology — Feb. 2008 – Aug. 2012
B.S., Electrical Engineering
Minor: Mathematical Science
GPA: 3.97/4.3

WORKING EXPERIENCE

Sony Computer Entertainment America, San Mateo, CA — Oct. 2012 – Feb. 2013
Research Intern (mentor: Ruxin Chen)
Worked on multiple keyword spotting in speech

RESEARCH PROJECTS

Speech recognition

Semi-supervised continuous speech recognition (2016)
End-to-end continuous speech recognition (2015)
Acoustic model for Korean syllable (2013-2014) — Aug. 2013 – Feb. 2017, leader
for *spontaneous spoken dialog system for language learning*
Electronics and Telecommunications Research Institute (ETRI)

Speech enhancement

Location-robust blind source extraction — Sep - Nov. 2018, member
for *free-running embedded speech recognition technology for natural language dialogue with robots*
Korea Evaluation Institute of Industrial Technology (KEIT),

Unpaired speech enhancement — Apr - Oct. 2017, leader
for *spontaneous spoken dialog system for language learning*
Electronics and Telecommunications Research Institute (ETRI)

Natural language generation

Article based question-answering and chitchat bot — May – Nov. 2017, co-leader
for *emotional intelligence technology to infer human emotion and carry on dialogue accordingly*
Institute for Information & Communication Technology Promotion (IITP)

PUBLICATION

International Journal

1. Bo-Kyeong Kim, **Geonmin Kim**, Soo-Young Lee, "Style-Controlled Synthesis of Clothing Segments for Fashion Image Manipulation", *IEEE transactions on multimedia*, (2019)

2. **Geonmin Kim**, Hwaran Lee, Bo-Kyeong Kim, Sang-Hoon Oh, Soo-Young Lee, "Unpaired Speech Enhancement by Acoustic and Adversarial Supervision for Speech Recognition", *IEEE Signal Processing Letters*, (2019)

3. Ho-Gyeong Kim, Hwaran Lee, **Geonmin Kim**, Sang-Hoon Oh, Soo-Young Lee, "Rescoring of N-best Hypotheses using Top-down Selective Attention for Automatic Speech Recognition", *IEEE Signal Processing Letters*, (2018)

4. Hwaran Lee, **Geonmin Kim**, Ho-Gyeong Kim, Sang-Hoon Oh, Soo-Young Lee, "Deep CNNs Along the Time Axis With Intermap Poling for Robustness to Spectral Variations", *IEEE Signal Processing Letters*, (2016)

International Conference

1. **Geonmin Kim**, Hwaran Lee, Bo-Kyeong Kim, Soo-Young Lee, "Compositional Sentence Representation from Character within Large Context Text", *International Conference on Neural Information Processing*, (2017)

2. Bo-Kyeong Kim, Suh-Yeon Dong, Jihyeon Roh, **Geonmin Kim**, Soo-Young Lee, "Fusing Aligned and Non-Aligned Face Information for Automatic Affect Recognition in the Wild: A Deep Learning Approach", IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRWS), (2016)

3. Ho-Gyeong Kim, Jihyeon Roh, Hwaran Lee, **Geonmin Kim**, Soo-Young Lee, "Active Learning for Large-scale Object Classification: from Exploration to Exploitation", *International Conference on Human-Agent Interaction*, (2015)

4. **Geonmin Kim**, Chang-Hyun Kim, Soo-Young Lee, "Implement real-time polyphonic pitch detection and feedback system for the melodic instrument player",

International Conference on Neural Information Processing, 2012

AWARDS

NIPS Conversational Intelligence Challenge 2017 — Dec. 2017
Article-based chatbot which can carry on both question-answering and chitchat, *Awarded with 3rd place*.

Qualcomm Innovation Award — Mar. 2015
Active learning for large-scale object classification: from exploration to exploitation

TEACHING EXPERIENCE

Qualcomm innovation award chatbot hackerton committee — Summer 2018
EE476 Audio-Visual Perception Models — Spring 2016-2017
EE538 Neural Networks — Fall 2015-2017

SKILLS

Languages: Python, Lua, MATLAB, C/C++, CUDA
Libraries: PyTorch, Torch7, KALDI

REFERENCES

Soo-Young Lee
Professor Emeritus, School of Electrical Engineering
Director, Institute for Artificial Intelligence
Korea Advanced Institute of Science and Technology
+82.42.350.3431
sylee@kaist.ac.kr