Study on audio source separation algorithms under various conditions, ranging from determined to more realistic conditions

March 2021

Li Li

Study on audio source separation algorithms under various conditions, ranging from determined to more realistic conditions

Graduate School of Systems and Information Engineering
University of Tsukuba

March 2021

Li Li

A Doctoral Dissertation
submitted to Graduate School of Systems and Information Engineering,
University of Tsukuba
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Thesis Committee:

Advisor	Dr. Shoji Makino Professor of University of Tsukuba
Subadvisor	Dr. Hirokazu Kameoka Senior Distinguished Researcher at NTT Communication Science Laboratories Adjunct Associate Professor of National Institute of Informatics Adjunct Associate Professor of University of Tsukuba
Subadvisor	Dr. Takeshi Yamada Associate Professor of University of Tsukuba
Subadvisor	Dr. Kazuhiro Fukui Professor of University of Tsukuba
Subadvisor	Dr. Jun Sakuma Professor of University of Tsukuba

Acknowledgements

I would like to express my best gratitude to all people who have supported me during my master's course and doctoral course at University of Tsukuba and the research student period at the University of Tokyo (UT).

First of all, I would like to give special thanks to my thesis advisor, Prof. Shoji Makino, at University of Tsukuba, who gave me the greatest freedom to choose the research topics and his valuable guidance and full support throughout my research. I am also grateful to him for providing me many precious opportunities to come into contact with international research, which has broadened my perspective and helped me find my own direction as a researcher.

I would like to express my sincerest gratitude to Dr. Hirokazu Kameoka from Nippon Telegraph and Telephone Corporation (NTT), who was an adjunct associate professor at UT and one of my supervisors when I was at UT. He has provided me fruitful comments and helpful suggestions at all levels of my research throughout my career. His attempts to understand my inaccurate expressions and affirmation of my immature ideas have been truly encouraging, which boost my self-confidence. The core of this work began with his elegant ideas. I could not have accomplished most of the work without his support. I am also grateful to him for willing to provide me an irreplaceable opportunity to start my research in his lab about six years ago when I knew nothing about signal processing.

I would sincerely like to thank Dr. Kazuhito Koishida from Microsoft Corporation for inviting me to Microsoft. It was a precious opportunity to do research as an intern at the world's most advanced IT company. I will never forget the days I spent discussing with the brilliant researchers of the group. I believe that the work experience at Microsoft will have a positive impact on my future career. My work

on IVA in Chapter 4 was a direct collaboration with him, and those days we spent discussing the problems and algorithms were memorable times.

I would especially like to express my gratefulness to Prof. Tomoki Toda from Nagoya University for giving me a grateful opportunity to start my research on singing voice and electrolaryngeal speech. Although this work is not included in this thesis, the experiment at Toda Lab has greatly expanded my research interests and field. The discussions and comments during my short stay were truly helpful and inspired me a lot.

I would like to offer my gratitude to Dr. Takeshi Yamada, Associate Professor of University of Tsukuba, and Researcher Mitsuo Matsumoto from University of Tsukuba for their comprehensive support during my research life in the laboratory and class-related helps so that I could concentrate on my research.

I would like to express my deep gratitude to Prof. Kazuhiro Fukui and Prof. Jun Sakuma from University of Tsukuba, member of the dissertation committee, for their valuable comments on the dissertation.

I am very grateful to Prof. Hiroshi Saruwatari from UT and Prof. Nobutaka Ono from TMU for their fruitful comments, advice, and encouragement that they shared with me through all the conferences and meetings I have attended. I would like to thank Dr. Daichi Kitamura for his advice and all the materials he shared online, such as papers, codes, and slides, which helped me to better understand the research in this thesis.

I would like to express my deep gratitude to all the members from Multimedia Lab, Kameoka Lab, Saruwatari Lab, and NTT Communication Lab for their assistance, friendship, and cooperation. It is my pleasure to share most of the time in these years with them. These wonderful times motivated me during my research and gave me a sense of belonging in a foreign country. I would also like to sincerely thank my colleagues and coauthors, some of whom have become close friends. I would like to thank in particular Kouei Yamaoka (now with TMU) and Shota Inoue for spending a significant amount of time proofreading my writings in Japanese and sharing research information and ideas with me, and Masakazu Une and Jennifier Santoso for sharing many meals and many laughs with me in the tough period of the last year of the Ph.D course. Those days and nights were precious and unfor-

gettable. I would like to express my gratitude to Takuya Higuchi (now with Apple Inc.), who was my tutor at Kameoka Lab, Tomohiko Nakamura (now with UT), and Norihiro Takamune (now with UT), for their kind help and patient guidance when I was at UT. I would like to thank Ms. Akemi Taguchi and Ms. Kiyoko Kuramochi, who are secretaries in Multimedia Lab, for their warm supports and help throughout my whole campus life. I would also like to express my appreciation to Ms. Naoko Tanji, who was an assistant technical staff in Kameoka Lab (now with Saruwatari Lab) and Ms. Takako Kumai, who was an assistant technical staff in Saruwatari Lab, for their kind and meticulous care when I started my study and research life in Japan.

This work was partly supported by JSPS KAKENHI Grant Numbers 18J20059, 17H01763, and 19H04131, JST CREST Grant Number JPMJCR19A3, and SECOM Science and Technology Foundation.

Finally, I would like to express my heartfelt love and gratitude to my parents and best friends, who unconditionally supported all my decisions and gave me encouragement whenever I felt depressed. I would also like to express my love to my cat, who accompanied me during my two years of living alone in Japan and gave me a lot of fun. Without them, I would not have been able to finish this dissertation and my doctoral course in Japan.

Abstract

We deal through this thesis with the problem of enhancing the target speech in noisy recorded signals by separating the target speech signal from other non-target signals. Although humans are able to focus on and understand the speech of interest in a complex acoustic environment, the presence of noise and interference can significantly reduce the intelligibility and comprehension of speech.

Depending on the relationship between the number of sources and microphones, the source separation problem is classified into determined and underdetermined cases. The determined case is a well-posed problem, where a sufficient number of observations are available. In contrast, the underdetermined case, including single-channel situations, is an ill-defined problem, lacking information for solving the problem. Determined methods are preferred thanks to the satisfactory performance, but more microphones are usually needed to meet the condition in real-life situations. Hence, application scenarios are limited. This raises the importance of underdetermined methods since it is much easier to achieve the underdetermined condition, especially the single-channel condition. Furthermore, different devices and applications have different hardware configurations and prerequisites, e.g., low computational cost and low latency, which should also be considered when developing methods for realistic environments. This thesis aims to develop source separation methods that achieve high performance in the determined case and methods can be applied in more realistic conditions in real life.

The main topic in Chapter 3 is to improve the source separation performance of frequency domain independent component analysis (FDICA)-based determined methods by incorporating a source model with stronger representation power into the framework. We train source models using deep generative models (DGM) that

include variational autoencoder (VAE) and generative adversarial network (GAN). VAE allows us to formulate the training and separation criteria to be consistent, whereas GAN has a high potential to achieve a more precise model. Convergence-guaranteed optimization algorithms are derived for parameter estimation. We further propose a fast optimization algorithm to reduce the computational cost, which estimates parameters that approximately maximizes the posterior. The incorporation of the DGM-based source model was confirmed to be effective through experimental evaluations.

In both Chapter 4 and Chapter 5, we focus on underdetermined methods to deal with cases where the determined condition does not hold. Chapter 4 propose a geometric information-guided multichannel source separation method, which combines beamforming-based geometric constraints and independent vector analysis (IVA). A parameter estimation algorithm is derived based on the auxiliary function approach. Besides, an online extension of the method to real-time applications is performed by applying autoregressive calculation to the signal statistics. We confirmed through the experiments the effectiveness of both offline and online methods introduced in this chapter.

As the most costless condition to achieve, it is unavailable to use spatial information in the single-channel condition, making it challenging to achieve high source separation performance compared to multichannel conditions. However, it has the advantage of being applicable to situations where the spatial characteristics change over time and thus has the broadest range of applications. In Chapter 5, we derive a convergence-guaranteed basis training algorithm based on auxiliary function approach for discriminative nonnegative matrix factorization (DNMF), one powerful monaural source separation method without neural networks. Experimental results revealed the effectiveness of the basis matrix trained with the proposed method in monaural source separation.

Abstract in Japanese

人間は複雑な音響環境においても特定の音声に注意を向け，理解することができるが，雑音や干渉音の存在は発話の明瞭度や理解度を著しく低下させる．本論文では，ノイジーな録音信号に含まれる目的の音声信号と非目的音声信号を分離することで，目的音声を強調する問題を取り扱う．

音源分離問題は音源とマイクロホンの数の関係性によって，優決定条件と劣決定条件に大別される．優決定条件は十分な数の観測信号が分離の手掛かりとして利用可能な良定義問題であるのに対し，シングルチャンネルを含む劣決定条件は問題を解くための情報が不足している悪定義問題である．優決定条件の手法は高い分離性能が得られて好ましい一方で，多くの場合では音源数よりマイクロホン数が多いという条件を満たすために多数のマイクロホンが必要である．そのため，実環境において適用可能なシーンが限られている．従って，より容易に条件を満たせる劣決定条件の手法が重要になる．更に，適用するデバイスやアプリケーションによってハードウェア構成，許容される計算コストや遅延が異なる．実環境において動作する手法を開発するためにそれらの制約を考慮しなければならない．本論文では，優決定条件からより現実的な条件で適用可能な手法までの音源分離手法群を提案する．

第 3 章では，周波数領域の独立成分分析（FDICA）に基づく優決定条件の手法を拡張し，分離性能を向上させることを目的としている．具体的には，FDICA の枠組における音源モデル部分の精緻化を実現するため，変分自己符号化器（VAE）や敵対生成ネットワーク（GAN）と呼ばれる深層生成モデルを導入する．VAE を用いることでネットワーク学習と推論時に同一の最適化規準を用いることができ，GAN を用いた場合はより高精度な音源モデルが得られることが期待できる．これらの提案手法に対して，我々は収束が保証されるパラメータ最適化アルゴリズムを提案する．更に，計算コストを削減するために，最大事後確率が得られるパラメータを近似計算する高速な最適化アルゴリズムを提案する．評価実験により，深層生成モデルを音源モデル

に導入するアプローチが音源分離性能の向上に有効であることが示された．

第 4 章では，幾何的制約に基づく多チャンネル音源分離手法を提案する．提案手法として，ビームフォーミングに基づく幾何的制約と独立ベクトル分析（IVA）の目的関数の組み合わせによって定式化され，補助関数法に基づいて収束性が保証されるパラメータ最適化アルゴリズムを導出する．更に，この提案手法をリアルタイムアプリケーションに適用するために，信号統計量の計算に自己回帰計算を適用することで，提案手法のオンライン化を実現する．評価実験により，本章で提案した 2 つの手法の有効性を確認した．

複数のマイクが必要になる多チャンネル条件に対し，マイク一つがあれば成立するシングルチャンネルは最も容易に満たせる条件である．シングルチャンネル音源分離手法は，マイク間の空間情報を利用できないため，多チャンネル手法に比べて高い分離性能の実現は困難であるが，空間特性が時変的なシーンにも適用可能で，最も幅広い場面に応用可能である．第 5 章では，深層学習を用いない強力なシングルチャンネル音源分離手法である識別的非負値行列因子分解（DNMF）のための基底学習アルゴリズムを提案する．提案手法は補助関数法に基づき導出された最適化アルゴリズムであるため，収束性が保証されている．評価実験により，提案手法で学習した基底行列はシングルチャンネル音源分離の性能向上に有効であることを明らかにした．

Contents

List of Figures

List of Tables

Abbreviations

ACVAE	auxiliary classifier variational autoencoder
AuxDNMF	auxiliary function-based discriminative nonnegative matrix factorization
AuxIVA	auxiliary function-based independent vector analysis
BiLSTM	bidirectional long short-term memory
BLSTM	bidirectional long short-term memory
BM	blocking matrix
BSS	blind source separation
CNN	convolutional neural network
ConvDC	deep clustering with convolutional neural network
CVAE	conditional variational autoencoder
DC	deep clustering
DGM	deep generative model
DNMF	discriminative nonnegative matrix factorization
DNN	deep neural network
DOA	direction of arrival
EM	expectation-maximization
EU	Euclidean
FastMVAE	fast multichannel variational autoencoder
FDICA	frequency-domain independent component analysis
GAN	generative adversarial network
GCAV-IVA	geometrically constrained independent vector analysis with auxiliary function and vector coordinate descent
GCIVA	geometrically constrained independent vector analysis

GLU	gated linear unit
GSC	generalized sidelobe canceller
HEAD	hybrid exact approximate joint diagonalization
ICA	independent component analysis
IDLMA	independent deelpy low-rank matrix analysis
ID	identity
ILRMA	independent low-rank matrix analysis
IP	iterative projection
iSTFT	inverse short-time Fourier transform
IS	Itakura-Saito
IVA	independent vector analysis
JS	Jensen-Shannon
KL	Kullback-Leibler
LCMV	linearly constrained minimum variance
LGM	local Gaussian model
LSGAN	least square generative adversarial network
LSTM	long short-term memory
MAP	maximum a posterior
ML	maximum likelihood
MM	majorization-minimization
MPDR	minimum power distortionless response
MSGAN	multichannel star generative adversarial network
MU	multiplicative update
MVAE	multichannel variational autoencoder
NMF	nonnegative matrix factorization
oAuxIVA	online algorithm of auxiliary function-based independent vector analysis
oGCVA-IVA	online algorithm of geometrically constrained independent vector analysis with auxiliary function and vector coordinate descent
PESQ	the perceptual evaluation of speech quality
PIT	permutation invariant training

PoE	product-of-experts
RIR	room impulse response
RNN	recurrent neural network
SAR	sources-to-artifacts ratio
SDR	source-to-distortion ratio
SIR	source-to-interferences ratio
SNR	Source-to-noise ratio
StarGAN	star generative adversarial network
STFT	short-time Fourier transform
STOI	short-time objective intelligibility
TDOA	time difference of arrival
VAE	variational autoencoder
VCD	vectorwise coordinate descent
WGAN-GP	Wasserstein generative adversarial network with gradient penalty
WGAN	Wasserstein generative adversarial network

Chapter 1

Introduction

1.1 Background

As one of the essential communication tools for human beings, speech endows us with a natural and efficient way to interact with the world. Besides speech, we are always surrounded by various sounds in a natural environment, e.g., music, mechanical sound, and ambient noise. Since multiple sounds usually occur at the same time, they acoustically interfere with each other. Although a human can considerably separate and focus on listening to the speech of interest among these sounds, it is unavoidable that the presence of acoustic interferences decreases speech understandability. In addition, it is a much more difficult problem for machines such as smartphones, note PCs, and robots to understand the target speech in such a complicated acoustic environment.

Speech enhancement [1] is an important technique to solve this problem, which aims to increase the speech understandability distorted by noise and interferences. When we treat one specific speech source in a recorded mixture signal as the target and the other sources as interferences or noise, enhancing the specific speech source can be considered as a target and non-target source separation problem. Therefore, *source separation* [2], whose objective is to recover one or more sources contained in a recorded signal, is another promising technique to enhance speech. Speech enhancement and source separation are fundamental technology for audio signal processing, which have a wide range of applications. Examples

include hearing aid devices, automatic speech recognition, speaker identification, teleconferencing systems, and smart home devices. These devices have different hardware configurations (e.g., number of microphones) and are used in various situations where the prerequisites are different. This indicates the importance of developing source separation algorithms that serve different situations.

With a microphone array consisting of several synchronized microphones, one promising approach to source separation is blind source separation (BSS) [3], which separates individual sources without any information about the sources and microphones. Thanks to this property, the BSS technique is accessible for many applications. However, one limitation is that the number of microphones is needed to be equal to or greater than the number of sources, which is called determined or overdetermined conditions. This means that the usage of more microphones (e.g., 4, 6, or more) is necessary to meet the requirement in a realistic environment. For example, to enhance a target speech in a cafeteria, since we need to consider noise from other customers, tableware, kitchen, background music, and outside, we have to use more than 6 microphones.

This raises the importance of developing source separation methods for underdetermined situations where the number of microphones is less than the number of sources. However, due to the difficulty of obtaining sufficient information from observations with less microphones, it is often necessary to use additional assumptions (e.g., sparsity) and a priori information about sources and microphone arrays to effectively solve the underdetermined source separation. One particular case of the underdetermined situation is single-channel, where only a single microphone is available. Although small devices with dual microphones are now widespread, there are still many devices that have only one microphone. Moreover, since single-channel methods do not require any spatial information, it can also be applied to separating signals in situations where spatial characteristics are time-varying, e.g., moving sources, a common occurrence in the real environment. Therefore, as the esaiest case to achieve, single-channel methods have the broadest range of applications.

1.2 Related work

The source separation problem can be categorized depending on its assumptions and conditions. In this thesis, we use the terms “determined/underdetermined/single-channel” and “blind/guided/supervised’ to properly categorize these methods. The terms “determined”, “underdetermined” and “single-channel” indicate the relationship between the numbers of sources and microphones, while the terms “blind”, “guided”, and “supervised” indicate whether prior information is available for solving the problem or whether a training phase is required.

As blind determined method, BSS for the determined situation is one of the most fundamental theories in these problems. In particular, independent component analysis (ICA) [4] has been well studied for ages. Since a mixing system of acoustic signals becomes a convolutive mixture due to the effect of room reverberation, and it is a more difficult problem than the instantaneous mixing system, frequency-domain ICA (FDICA) [5] has been established to deal with such convolutive mixture using Fourier transform, where the instantaneous mixing system is assumed to be approximately held in the time-frequency domain. Furthermore, the frequency-domain approach provides the flexibility of utilizing various models for the time-frequency representation of source signals. The approach involves independent vector analysis (IVA) [6, 7] and independent low-rank matrix analysis (ILRMA) [8, 9], which are extensions of FDICA. With these methods, a high-quality blind speech separation was achieved. Recently, motivated by the impressive power of deep neural networks (DNNs), some methods have been proposed to incorporate DNNs into the FDICA framework, which is categorized as “supervised determined” methods. Independent deeply low-rank matrix analysis (IDLMA) [10] is one of these methods, which trains a DNN for each source so that the trained DNN can work as a source-dependent noise reduction system.

In many applications, the source position can be roughly known in advance since it can be estimated by sound localization methods, determined by image/video processing, or simply known by geometry. Geometrically constrained BSS [11, 12] is a framework that utilizes this prior information to guide the separation system so that the desired target signal is output from a prespecified channel. This con-

cept has already been adopted to ICA, IVA, and ILRMA [12–14]. The geometric constraints allow us to manually control the spatial and frequency responses of the separation system estimated by a BSS method, which provides the flexibility of determining to preserve or suppress a signal originating from a specific direction. With the constraint, BSS methods is able to work well even there exists additional diffuse noise. Furthermore, the constraints can be designed as a blocking matrix (BM) [15] so that the corresponding channel can produce good estimate of interference and noise for contructing a generalized sidelobe canceller (GSC) [16], making it possible to deal with underdetermined situations. From this point of view, the geometrically constrained BSS methods can be considered as “guided underdetermined” methods.

For single-channel audio source separation, nonnegative matrix factorization (NMF) [17, 18] is a powerful approach. NMF factorizes an observed magnitude (or power) spectrogram, interpreted as a nonnegative matrix, into the product of two nonnegative matrices, which amounts to approximating the observed spectra by a linear sum of basis spectra scaled by time-varying amplitudes. In a supervised setting, NMF is first applied to train the basis spectra of each source. At separation time, NMF is applied to the spectrogram of a mixture signal using the pretrained spectra. The source signal can then be separated out using a Wiener filter. A typical way to train the basis spectra of each source is to minimize the objective function of NMF. However, the basis spectra obtained in this way do not ensure that the separated signal will be optimal. To address this, a framework called discriminative NMF (DNMF) [19] has been proposed, and several works have been done to solve this bilevel optimization problem [19–21]. All these methods are categorized as “supervised single-channel” methods.

1.3 Objective and overview of thesis

The aim of this thesis is to propose source separation methods that achieve high performance in the determined case and methods can be applied in more realistic conditions in real life.

This thesis consists of three parts. Fig. 5.1 shows the overwiew of the thesis.

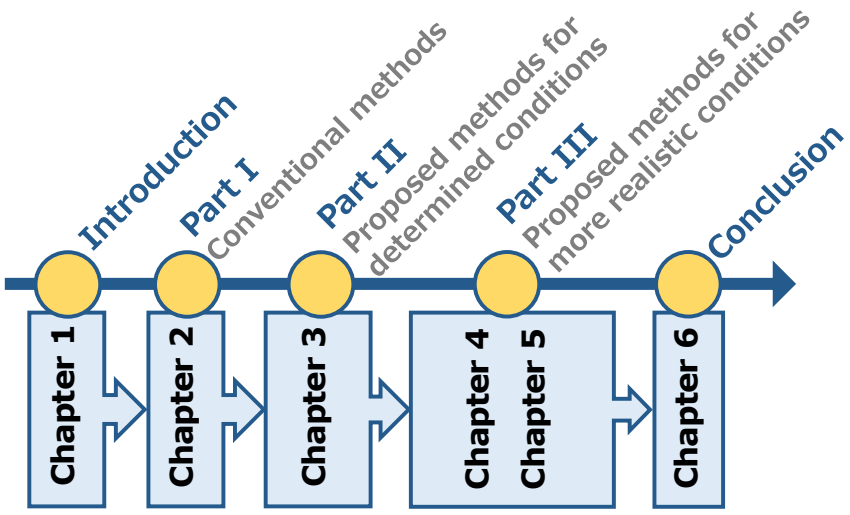

Figure 1.1: Overview of this thesis

In the first part, Chapter 2, we provide some preliminaries, which are necessary for later discussions. In particular, we formulate audio source separation problems for multichannel and single-channel cases. We then explain conventional BSS methods based on signal independence and supervised NMF methods. Finally, we introduce the evaluation criteria. In the second part, Chapter 3 describes the details of the proposed methods for determined source separation, which incorporate supervised-learned source models pretrained using neural networks into conventional FDICA-based methods. Also, a fast parameter estimation algorithm for reducing computational cost is proposed. In the third part, we propose algorithms aiming to address more realistic situations where the determined condition does not hold. Chapter 4 proposes a geometric information-guided multichannel source separation method. After explaining the offline optimization algorithm, an online extension is developed. Chapter 5 deals with a single-channel source separation problem. We first introduce discriminative NMF methods; then derive a new effective optimization algorithm. The effectiveness of these methods is validated via experiments. Finally, Chapter 6 concludes the entire contents and contributions in this dissertation.

Chapter 2

Audio source separation

2.1 Introduction

In this chapter, we provide some preliminaries about audio source separation, which are necessary for later discussions. We first give formulations of source separation for multichannel and single-channel situations. Next, we explain supervised NMF methods for single-channel source separation and BSS methods based on signal independence for determined multichannel source separation. We also introduce the auxiliary function approach, a critical optimization method used through this thesis. Finally, we review criteria usually applied to evaluate the source separation performance.

2.2 Formulation of source separation problems

2.2.1 Multichannel case

Let us consider I microphones capture J source signals, where $x_i(t)$ and $s_j(t)$ denote the signal of time t observed at the ith microphone and the jth source signal, respectively. We use $x_i(f,n)$ and $s_j(f,n)$ to denote the correspecding complex-valued short-time Fourier transform (STFT) coefficients, where f and n are the frequency and time indices, respectively. We denote the vectors containing

$x_1(f,n),\ldots,x_I(f,n)$ and $s_1(f,n),\ldots,s_J(f,n)$ by

$$\mathbf{x}(f,n) = [x_1(f,n),\ldots,x_I(f,n)]^{\mathsf{T}} \in \mathbb{C}^I, \tag{2.1}$$

$$\mathbf{s}(f,n) = [s_1(f,n),\ldots,s_J(f,n)]^{\mathsf{T}} \in \mathbb{C}^J, \tag{2.2}$$

where $(\cdot)^{\mathsf{T}}$ denotes transpose. When the length of the analysis window of STFT is sufficiently longer than that of impulse response and the mixing system is time-invariant, the relationship between the source signals and observed signals can be approximated as an instantaneous mixture model at each frequency bin as

$$\mathbf{x}(f,n) = \mathbf{A}(f)\mathbf{s}(f,n), \tag{2.3}$$

where $\mathbf{A}(f) = [\mathbf{a}_1(f),\ldots,\mathbf{a}_J(f)] \in \mathbb{C}^{I\times J}$ is called the mixing matrix. Here, $\mathbf{a}_j(f) = [a_{1,j}(f),\ldots,a_{I,j}(f)]^{\mathsf{T}} \in \mathbb{C}^I$ is the array manifold vector, also called steering vector, which models the acoustic paths for jth source in frequency domain.

In a determined situation, where $I = J$, the mixing matrix is a full-rank square matrix. Therefore, we can define an inverse matrix of $\mathbf{A}(f)$ that separates the mixture signals as

$$\mathbf{y}(f,n) = \mathbf{W}^{\mathsf{H}}(f)\mathbf{x}(f,n), \tag{2.4}$$

where $(\cdot)^{\mathsf{H}}$ denotes the Hermitian transpose, $\mathbf{W}(f) = [\mathbf{w}_1(f),\ldots,\mathbf{w}_J(f)] \in \mathbb{C}^{I\times J}$ is called demixing matrix, and $\mathbf{y}(f,n) = [y_1(f,n),\ldots,y_J(f,n)]^{\mathsf{T}} \in \mathbb{C}^J$ is the vector containing separated source signals. Here, $\mathbf{w}_j(f) = [w_{1j}(f),\ldots,w_{Ij}(f)]^{\mathsf{T}} \in \mathbb{C}^I$ is a demixing filter for the jth source in a mixture signal. The aim of determined source separation is to estimate $\mathcal{W} = \{\mathbf{W}(f)\}_f$ from the observation $\mathcal{X} = \{\mathbf{x}(f,n)\}_{f,n}$ with assumptions and available prior information. The waveform of separated signal $y_j(t)$ is obtained by applying inverse STFT (iSTFT) to $y_j(f,n)$.

2.2.2 Single-channel case

For the single-channel separation problem, the case of $I = 1$, the time-invariant mixing system is simplified as

$$x(f,n) = \sum_j a_j(f) s_j(f,n), \tag{2.5}$$

where the microphone index i is omitted. Single-channel source separation is more difficult than a multichannel problem since the difference of phase and amplitude between microphones cannot be utilized. Therefore, more prior knowledge and assumptions are usually needed to achieve single-channel source separation.

2.3 Nonnegative matrix factorization for single-channel source separation

NMF refers to a technique for modeling spectra of audio sources. Since audio sources usually have distinct structures in the time-frequency domain, e.g., STFT domain, the basic idea of NMF is to learn these structures by factorizing the observed spectrograms into two low-rank matrices, which are corresponding to the spectral templates and scaling coefficients. With the learned spectral templates, NMF is able to represent the corresponding sources even in mixture signals, which makes separation possible.

2.3.1 Basic principle of NMF

Given a power spectrogram $\mathbf{P} = \{\mathsf{p}(f,n)\}_{f,n} \in \mathbb{R}^{\geq 0, F\times N}$ or a magnitude spectrogram of an audio signal, which can be interpreted as a nonnegative matrix, NMF factorizes it into the product of a basis matrix $\mathbf{B} = \{\mathsf{b}_k(f)\}_{f,k} \in \mathbb{R}^{\geq 0, F\times K}$ and an activation (coefficient) matrix $\mathbf{H} = \{\mathsf{h}_k(n)\}_{n,k} \in \mathbb{R}^{\geq 0, K\times N}$:

$$\mathbf{P} \approx \mathbf{Q} = \mathbf{BH} \tag{2.6}$$

$$\mathsf{p}(f,n) \approx \mathsf{q}(f,n) = \sum_k \mathsf{b}_k(f)\mathsf{h}_k(n), \tag{2.7}$$

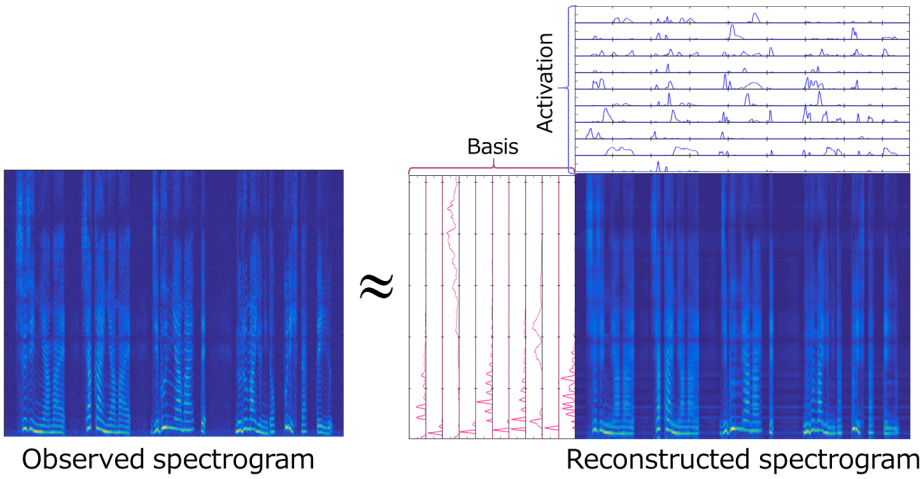

Figure 2.1: An example of applying NMF to spectrogram of speech signal.

where k denotes the index of spectral template in the basis matrix, and $\mathbf{Q} = \{\mathsf{q}(f,n)\}_{f,n} \in \mathbb{R}^{\geq 0, F\times N}$. NMF learns the underlying spectral structures of the spectrogram and approximately represents the spectrogram as a linear combination of the learned spectral templates with time-varying coefficients. Since the objective of NMF is to reduce the data demension and find out underlying data structures, typically the number of spectral templates K is set to be a small value as $K \ll \min(F,N)$, which is equivalent to approximating the spectrogram by a lower rank matrix. It is important to note that NMF assumes that the observed data are additive in nature, where is approximately true when applying NMF to magnitude or power spectrograms. Fig. 2.1 shows an example of applying NMF to a spectrogram of speech signal with $K = 10$. In the basis matrix, we can observe that hormonic structures of the speech are successfully extracted.

NMF leads to different optimization problems according to the definition of the measurement of the dissimilarity between **P** and **Q**. Most widely used goodness-of-fit criteria are Euclidean distance (EU), generalizaed Kullback-Leibler divergence (KL), which is also known as I-divergence [22], and Itakura-Saito divergence (IS) [23]. These criteria of $\mathsf{q}(f,n)$ from $\mathsf{p}(f,n)$ are defined as follows:

$$\mathcal{D}_{\mathsf{EU}}(\mathbf{P}|\mathbf{Q}) = ||\mathbf{P}-\mathbf{Q}||_F^2 = \sum_{f,n} |\mathsf{p}(f,n) - \mathsf{q}(f,n)|_F^2, \tag{2.8}$$

$$\mathcal{D}_{\mathsf{KL}}(\mathbf{P}|\mathbf{Q}) = \sum_{f,n} (\mathsf{p}(f,n)\log\frac{\mathsf{p}(f,n)}{\mathsf{q}(f,n)} - \mathsf{p}(f,n) + \mathsf{q}(f,n)), \tag{2.9}$$

$$\mathcal{D}_{\mathsf{IS}}(\mathbf{P}|\mathbf{Q}) = \sum_{f,n}(\frac{\mathsf{p}(f,n)}{\mathsf{q}(f,n)} - \log\frac{\mathsf{p}(f,n)}{\mathsf{q}(f,n)} - 1). \tag{2.10}$$

Here, $||\cdot||_F^2$ is the squared Frobenious norm. Note that all these metrices are special cases of β-divergence [24], where $\beta = 2$, $\beta = 1$, and $\beta = 0$ are corresponding to EU distance, KL divergence and IS divergence, respectively. Using these metrices, NMF is formulated as an optimization problem with respect to **B** and **H**.

$$\mathcal{F}(\mathbf{B},\mathbf{H}) = \underset{\mathbf{B},\mathbf{H}}{\operatorname{argmin}}\, \mathcal{D}_\beta(\mathbf{P}|\mathbf{BH}), \tag{2.11}$$

where $\mathcal{D}_\beta(\cdot|\cdot)$ denotes the abovementioned metrices.

When assuming that each observed time-frequency bin $\mathsf{p}(f,n)$ is generated independently from the normal distribution, Poission distribution, or exponential distribution with mean of $\mathsf{q}(f,n) = \sum_k \mathsf{b}_k(f)\mathsf{h}_k(n)$, the optimization problem of NMF with EU distance, KL divergence, or IS divergence is equivalent to the problem of the maximum likelihood (ML) estimation of **B** and **H** with the likelihood function $p(\mathbf{P};\mathbf{B},\mathbf{H})$. Therefore, NMF can be explained as a generative model.

2.3.2 Auxiliary function approach and multiplicative update algorithms

The objective of NMF is to find the optimal **B** and **H** that minimize the dissimilarity between **BH** and **P** under the nonnegative constraint. Although it is usually difficult to obtain the analytical expression of the global optimum, we can computationally find a local optimum using the auxiliary function approach, also known as majorization-minimization (MM) principle [25]. Note that the auxiliary function approach itself is not an algorithm, but a description of how to construct an optimization algorithm.

When constructing an auxiliry function-based algorithm to minimize a certain objective function, the main issue is how to design an appropriate auxiliary function called "*majorizer*" that is guaranteed to never be below the objective function.

Lemma 1. *If we use $\mathcal{F}(\Theta)$ to denote an objective function that we want to minimize with respect to Θ, and $\mathcal{F}^+(\Theta,\Lambda)$ to denote its auxiliary function, satisfying*

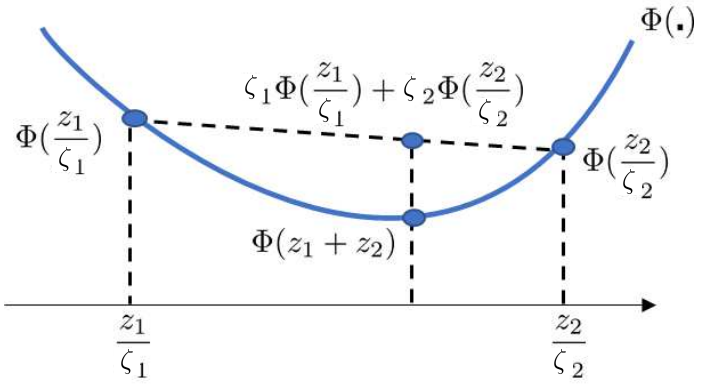

Figure 2.2: Jensen's inequality of $\mathrm{I} = 2$ case.

$\mathcal{F}(\Theta) = \min_\Lambda \mathcal{F}^+(\Theta, \Lambda)$*, then* $\mathcal{F}(\Theta)$ *is non-increasing under the following updates of auxiliary variable* Λ *and parameter* Θ*:*

$$\hat{\Lambda} = \underset{\Lambda}{\operatorname{argmin}}\, \mathcal{F}^+(\Theta, \Lambda), \tag{2.12}$$

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\, \mathcal{F}^+(\Theta, \Lambda). \tag{2.13}$$

Thus, if $\mathcal{F}(\Theta)$ *is bounded below, a stationary point of* $\mathcal{F}(\Theta)$ *can be found by iteratively performing these updates.*

Proof of Lemma 1*.* Suppose we set Θ to an arbitrary value $\tilde{\Theta}$. We will prove that $\mathcal{F}(\Theta)$ is non-increasing after the update (2.12) and (2.13). From (2.12), one obtains $\mathcal{F}(\tilde{\Theta}) = \mathcal{F}^+(\tilde{\Theta}, \hat{\Lambda})$, and it is obvious from (2.13) that $\mathcal{F}^+(\tilde{\Theta}, \hat{\Lambda}) \geq \mathcal{F}^+(\hat{\Theta}, \hat{\Lambda})$. By definition, one sees from (2.12) that $\mathcal{F}^+(\hat{\Theta}, \hat{\Lambda}) \geq \mathcal{F}(\hat{\Lambda})$. Therefore, we can immediately prove that $\mathcal{F}(\tilde{\Theta}) = \mathcal{F}^+(\tilde{\Theta}, \hat{\Lambda}) \geq \mathcal{F}^+(\hat{\Theta}, \hat{\Lambda}) \geq F(\hat{\Theta})$. □

It should be noted that this concept is adopted in many existing algorithms including algorithms for NMF [24]. For example, the expectation-maximization (EM) algorithm [26] builds a surrogate for a likelihood function of latent variable models by using Jensen's inequality. In general, if we can build a tight majorizer that is easy to optimize, we can expect to obtain a fast-converging algorithm. In addition, auxiliary function-based algorithms are notable in that there are no tuning parameters.

An useful inequality for designing majorizer is Jensen's inequality, which is invoked in algorithms of NMF and will be used through the thesis. For arbitrary convex function Φ with I nonnegative arguments $z_1, \ldots, z_\mathrm{I}$, Jensen's inequality shows

$$\mathbf{\Phi}(\sum_{\mathrm{i}} z_{\mathrm{i}}) \leq \sum_{\mathrm{i}} \zeta_{\mathrm{i}} \mathbf{\Phi}(\frac{z_{\mathrm{i}}}{\zeta_{\mathrm{i}}}), \tag{2.14}$$

where $\zeta_1, \ldots, \zeta_\mathrm{I}$ are nonnegative weights satisfying $\sum_{\mathrm{i}} \zeta_{\mathrm{i}} = 1$. The equality of (2.14) holds if and only if

$$\zeta_{\mathrm{i}} = \frac{z_{\mathrm{i}}}{\sum_{\mathrm{i}'} z_{\mathrm{i}'}}. \tag{2.15}$$

Fig. 2.2 shows an illustration of Jensen's inequality for a convex function with $\mathrm{I} = 2$.

With the auxiliary function approach and Jensen's inequality introduced above, we can derive the well-known multiplicative update (MU) algorithms for NMF. We first derive an algorithm for NMF using EU distance. The objective function we want to minimize can be expressed as

$$\mathcal{D}_{\mathsf{EU}}(\mathbf{P}|\mathbf{BH}) \stackrel{c}{=} \sum_{f,n} \Big(-2\mathsf{p}(f,n)\mathsf{q}(f,n) + \mathsf{q}^2(f,n) \Big), \tag{2.16}$$

where $=^c$ denotes equality up to a constant term. We want to design a majorizer such that the elements of matrices are separated into individual terms. Since a quadratic function x^2 is convex and arguments $\mathsf{b}_k(f)$ and $\mathsf{h}_k(n)$ are nonnegative, we can invoke Jensen's inequality to obtain a function upper bounding the second term in (2.16) as

$$\Big(\sum_k \mathsf{b}_k(f)\mathsf{h}_k(n) \Big)^2 \leq \sum_k \zeta_{k,f,n} \Big(\frac{\mathsf{b}_k(f)\mathsf{h}_k(n)}{\zeta_{k,f,n}} \Big)^2, \tag{2.17}$$

where $\zeta_{k,f,n}$ is a positive weight that sums to unity. The equality of (2.17) holds if and only if when

$$\zeta_{k,f,n} = \frac{\mathsf{b}_k(f)\mathsf{h}_k(n)}{\sum_{k'} \mathsf{b}_{k'}(f)\mathsf{h}_{k'}(n)}. \tag{2.18}$$

Therefore, a majorizer for the objective function $\mathcal{D}_{\mathsf{EU}}$ can be obtained as

$$\mathcal{D}^{+}_{\mathsf{EU}}(\mathbf{B},\mathbf{H},\boldsymbol{\zeta}) = \sum_{f,n}\Big(\mathsf{p}(f,n)^2 - 2\mathsf{p}(f,n)\sum_k \mathsf{b}_k(f)\mathsf{h}_k(n) + \sum_k \frac{\mathsf{b}_k^2(f)\mathsf{h}_k^2(n)}{\zeta_{k,f,n}}\Big). \quad (2.19)$$

Here, $\boldsymbol{\zeta} = \{\zeta_{k,f,n}\}_{k,f,n}$ denotes a set of paramter $\zeta_{k,f,n}$. By setting the partial derivative with respect to $\mathsf{b}_k(f)$ and $\mathsf{h}_k(n)$ at 0, we can derive an iteratively algorithm that consists of performing (2.18) and

$$\mathsf{b}_k(f) \leftarrow \frac{\sum_n \mathsf{p}(f,n)\mathsf{h}_k(n)}{\sum_n \mathsf{h}_k^2(n)/\zeta_{k,f,n}}, \quad (2.20)$$

$$\mathsf{h}_k(n) \leftarrow \frac{\sum_f \mathsf{p}(f,n)\mathsf{b}_k(f)}{\sum_f \mathsf{b}_k^2(f)/\zeta_{k,f,n}}. \quad (2.21)$$

By subsituting (2.18) into (2.20) and (2.21), we obtain the following MU algorithm:

$$\mathsf{b}_k(f) \leftarrow \mathsf{b}_k(f)\frac{\sum_n \mathsf{p}(f,n)\mathsf{h}_k(n)}{\sum_n \mathsf{q}(f,n)\mathsf{h}_k(n)}, \quad (2.22)$$

$$\mathsf{h}_k(n) \leftarrow \mathsf{h}_k(n)\frac{\sum_f \mathsf{p}(f,n)\mathsf{b}_k(f)}{\sum_f \mathsf{q}(f,n)\mathsf{b}_k(f)}. \quad (2.23)$$

Similar to EU distance, we then derive the MU algorithm for KL divergence

$$\mathcal{D}_{\mathsf{KL}}(\mathbf{P}|\mathbf{BH}) \stackrel{c}{=} \sum_{f,n}\Big(-\mathsf{p}(f,n)\log \mathsf{q}(f,n) + \mathsf{q}(f,n)\Big). \quad (2.24)$$

Because the first term in (2.24) has the "log-of-sum" form that is nonlinear, it is difficult to derive the closed-form solutions. Since the negative logarithm function is a convex function, we can invoke Jensen's inequality to construct an upper bound with "sum-of-log" form

$$-\log\sum_k \mathsf{b}_k(f)\mathsf{h}_k(n) \le -\sum_k \zeta_{k,f,n}\log\frac{\mathsf{b}_k(f)\mathsf{h}_k(n)}{\zeta_{k,f,n}}. \quad (2.25)$$

The equality holds if and only if

$$\zeta_{k,f,n} = \frac{\mathsf{b}_k(f)\mathsf{h}_k(n)}{\sum_{k'}\mathsf{b}_{k'}(f)\mathsf{h}_{k'}(n)}. \quad (2.26)$$

Then, a majorizer of the objective function (2.24) can be written as

$$\mathcal{D}^{+}_{\mathsf{KL}}(\mathbf{B},\mathbf{H},\boldsymbol{\zeta}) = \tag{2.27}$$
$$\Big(\mathsf{p}(f,n)\log\mathsf{p}(f,n) - \mathsf{p}(f,n)\sum_k \zeta_{f,n,j}\log\frac{\mathsf{b}_k(f)\mathsf{h}_k(n)}{\zeta_{k,f,n}} - \mathsf{p}(f,n) + \sum_k \mathsf{b}_k(f)\mathsf{h}_k(n)\Big),$$

where $\zeta_{k,f,n} > 0$ satisfies $\sum_k \zeta_{k,f,n} = 1$. The MU algorithm for the KL case can be derived in the same way as EU distance:

$$\mathsf{b}_k(f) \leftarrow \mathsf{b}_k(f)\frac{\sum_n \mathsf{p}(f,n)\mathsf{h}_k(n)/\mathsf{q}(f,n)}{\sum_n \mathsf{h}_k(n)}, \tag{2.28}$$
$$\mathsf{h}_k(n) \leftarrow \mathsf{h}_k(n)\frac{\sum_f \mathsf{p}(f,n)\mathsf{b}_k(f)/\mathsf{q}(f,n)}{\sum_f \mathsf{b}_k(f)}. \tag{2.29}$$

For IS divergence, we construct a majorizer for the objective function

$$\mathcal{D}_{\mathsf{IS}}(\mathbf{P}|\mathbf{BH}) \stackrel{c}{=} \sum_{f,n}\Big(\frac{\mathsf{p}(f,n)}{\mathsf{q}(f,n)} + \log\mathsf{q}(f,n)\Big). \tag{2.30}$$

Here, we need to design upper bound for both terms of $1/x$ and $\log x$. For term $1/x$, we can obtain the following inequality with Jensen's inequality:

$$\frac{1}{\sum_k \mathsf{b}_k(f)\mathsf{h}_k(n)} \leq \sum_k \zeta_{k,f,n}\Big(1/\frac{\mathsf{b}_k(f)\mathsf{h}_k(n)}{\zeta_{k,fn}}\Big), \tag{2.31}$$

where $\zeta_{k,f,n}$ is a positive parameter that satisfies $\sum_k \zeta_{k,f,n} = 1$. Since tangent lines for a concave function is never below the original function, we can utilize this property to design the upper bound for the term $\log x$:

$$\log\sum_k \mathsf{b}_k(f)\mathsf{h}_k(n) \leq \log\alpha_{f,n} + \frac{1}{\alpha_{f,n}}(\sum_k \mathsf{b}_k(f)\mathsf{h}_k(n) - \alpha_{f,n}). \tag{2.32}$$

The equality holds if and only if

$$\alpha_{f,n} = \sum_k \mathsf{b}_k(f)\mathsf{h}_k(n). \tag{2.33}$$

By replacing terms $1/x$ and $\log x$ in the objective function with the right hand of

(2.31) and (2.32), we obtain the majorizer

$$\begin{aligned}&\mathcal{D}_{\mathsf{IS}}^{+}(\mathbf{B},\mathbf{H},\boldsymbol{\zeta},\boldsymbol{\alpha}) \\ &=\sum_{f,n}\Big(\sum_{k}\frac{\mathsf{p}(f,n)\zeta_{k,f,n}^{2}}{\mathsf{b}_{k}(f)\mathsf{h}_{k}(n)}+\sum_{k}\frac{\mathsf{b}_{k}(f)\mathsf{h}_{k}(n)}{\alpha_{f,n}}-\log\mathsf{p}(f,n)+\log\alpha_{f,n}-2\Big),\end{aligned} \tag{2.34}$$

where $\boldsymbol{\alpha}=\{\alpha_{f,n}\}_{f,n}$ denotes a set of parameters $\alpha_{f,n}$. Similarly, by setting the partial derivative with respect to $\mathsf{b}_k(f)$ and $\mathsf{h}_k(n)$ at 0 and subsituting the update rules of auxiliary variables into the closed-form solutions, we obtain the MU algorithm for NMF with IS divergence as follows:

$$\mathsf{b}_{k}(f)\leftarrow\mathsf{b}_{k}(f)\Big(\frac{\sum_{n}\mathsf{p}(f,n)\mathsf{h}_{k}(n)/\mathsf{q}(f,n)^{2}}{\sum_{n}\mathsf{h}_{k}(n)/\mathsf{q}(f,n)}\Big)^{1/2}, \tag{2.35}$$

$$\mathsf{h}_{k}(n)\leftarrow\mathsf{h}_{k}(n)\Big(\frac{\sum_{f}\mathsf{p}(f,n)\mathsf{b}_{k}(f)/\mathsf{q}(f,n)^{2}}{\sum_{f}\mathsf{b}_{k}(f)/\mathsf{q}(f,n)}\Big)^{1/2}. \tag{2.36}$$

It is noteworthy that nonnegativity constraint of NMF can be satisfied with the MU algorithms by easily initializing all the parameters $\mathsf{b}_k(f)$ and $\mathsf{h}_k(n)$ with positive values. These auxiliary function-based update rules are equivalent to those derived in a heuristic way, where the partial derivative of the cost function $\mathcal{F}(\mathbf{B},\mathbf{H})$ with respective to the parameter is decomposed to two nonnegative terms, i.e. $\partial_{\mathbf{B}}\mathcal{F}=\partial_{\mathbf{B}}^{+}\mathcal{F}-\partial_{\mathbf{B}}^{-}\mathcal{F}$, where $\partial_{\mathbf{B}}^{+}\mathcal{F}\geq 0$ and $\partial_{\mathbf{B}}^{-}\mathcal{F}\geq 0$. Then the parameter $\mathbf{B}$ can be updated as $\mathbf{B}\leftarrow\mathbf{B}\circ(\partial_{\mathbf{B}}^{-}\mathcal{F}/\partial_{\mathbf{B}}^{+}\mathcal{F})^{\eta}$, where $\circ$ and $/$ denote element-wise multiplication and division, and $\eta>0$ is a stepsize similar to that involved in a gradient descent [27].

2.3.3 Source separation with supervised NMF

When applying NMF to single-channel source separation in a supervised manner, there are two phases, namely, training phase and separation phase. At training phase, NMF is applied individually to power (or magnitude) spectrograms of training samples $\mathbf{S}_j=\{|s_j(f,n)|^2\}_{f,n}$ to obtain spectral templates of each source,

$$\tilde{\mathbf{B}}_j,\tilde{\mathbf{H}}_j=\underset{\mathbf{B}_j,\mathbf{H}_j}{\operatorname{argmin}}\,\mathcal{D}_{\beta}(\mathbf{S}_j|\mathbf{B}_j\mathbf{H}_j)+\lambda_{\mathsf{sparse}}||\mathbf{H}_j||_p. \tag{2.37}$$

Here, $\lambda_{\text{sparse}}||\mathbf{H}_j||_p$ is a regularization term for promoting the sparsity of $\mathbf{H}_j$, which is typically applied to improve the performance [28]. λ_{sparse} is a parameter that weighs the importance of the regularization term and $||\cdot||_p$ denotes L_p norm, where L_1 and L_2 are common choices.

At separation time, the concatenated basis matrix $\tilde{\mathbf{B}} = [\tilde{\mathbf{B}}_1, \ldots, \tilde{\mathbf{B}}_J]$ is fixed at the pretrained basis spectra, and the activation matrix $\hat{\mathbf{H}}$ is estimated by fitting the NMF model to the power spectrogram of observed mixture signal $\mathbf{X} = \{|x(f,n)|^2\}_{f,n}$,

$$\hat{\mathbf{H}} = \underset{\mathbf{H}}{\operatorname{argmin}}\, \mathcal{D}_\beta(\mathbf{X}|\tilde{\mathbf{B}}\mathbf{H}) + \lambda_{\text{sparse}}||\mathbf{H}||_p. \tag{2.38}$$

Once $\tilde{\mathbf{B}}$ and $\hat{\mathbf{H}}$ are obtained, each source can be separated by a Wiener filter constructed using the estimated power spectrograms as follows:

$$\mathbb{Y}_j = \frac{\tilde{\mathbf{B}}_j\hat{\mathbf{H}}_j}{\tilde{\mathbf{B}}\hat{\mathbf{H}}} \circ \mathbb{X}, \tag{2.39}$$

where $\mathbb{Y}_1, \ldots, \mathbb{Y}_J$ are thus ensured to sum to the magnitude spectrogram $\mathbb{X} = \sqrt{\mathbf{X}}$ of the mixture signal. Here, $\circ$, $\frac{\cdot}{\cdot}$ and $\sqrt{\cdot}$ denote element-wise multiplication, division and square-root, respectively. The time-domain signal $y_j(t)$ is then obtained by applying iSTFT to the magnitude spectrogram $\mathbb{Y}_j$ and the phase spectrogram of the mixture signal. Note that speech enhancement is a special case with $J = 2$ and $j = \{\mathrm{s}, \mathrm{n}\}$, where s and n denote speech and noise, respectively.

It is obvious that the source separation and speech enhancement performance of NMF is greatly affected by the pretrained basis matrix $\tilde{\mathbf{B}}$ and the activation matrix $\hat{\mathbf{H}}$. However, since the spectral templates are trained individually for each source, it becomes challenging to achieve a high estimation accuracy of the activation matrix when spectral structures of different sources have high similarity. Fig. 2.3 shows an example of spectrograms of the reference clean speech signal and the speech signal enhanced using supervised NMF. Since there exist similar basis spectra in speech and noise, oversuppression occurs, and some unsuppressed noise components remain. To overcome this problem, various methods using prior information or characteristics of sources have been proposed, such as temporal

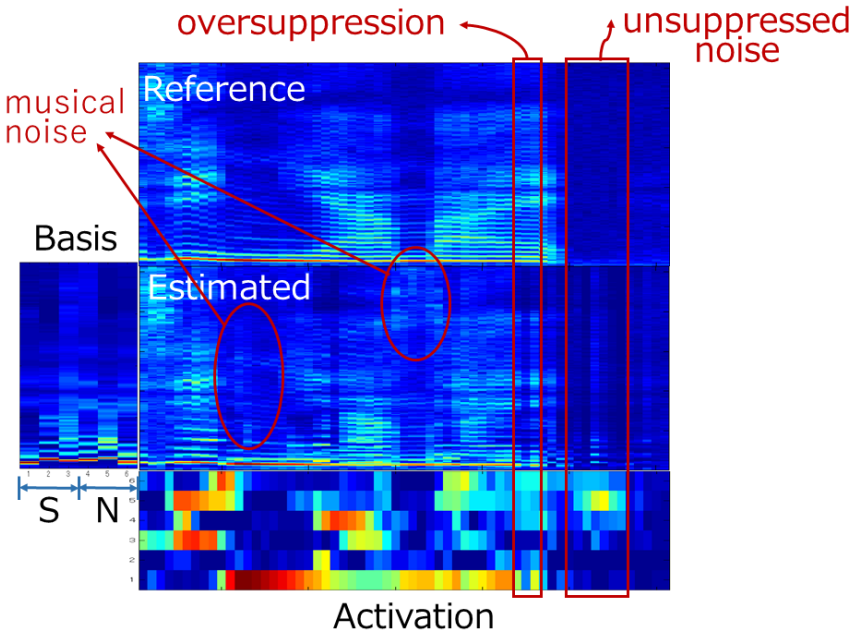

Figure 2.3: An example of speech enhancement using standard NMF. Oversuppression occurs and unsuppressed noise components remain in the spectrogram of the enhanced speech.

dynamics [29–31] and co-occurrence statistics [32]. Meanwhile, it is still unclear how to train spectral templates to yield optimal performance for source separation, especially for those separating sources by filtering as (2.39). Many efforts have also been made to train a more effective basis matrix [19, 28, 33–35].

2.4 Determined blind source separation with signal independence

The aim of determined BSS is to estimate $\mathcal{W}$ from $\mathcal{X}$ without any prior knowledge. The most popular approach to BSS is ICA and the term is sometimes regarded as synonymous with BSS. ICA assumes that the source signals follow non-Gaussian distributions and are statistically independent with each other.

2.4.1 ICA and FDICA

ICA was originally developed for instantaneous mixtures in the time domain [36–39], where no delays and reverberation are considered; then applied to convolutive mixtures [40, 41]. However, the estimation of demixing filters in the time domain is challenging since the number of parameters drastically increases when the filter length becomes large.

Instead of solving the time-domain deconvolution, FDICA [5, 40, 42] was proposed, where demixing matrix defined in the time-frequency domain $\mathbf{W}(f)$ is estimated for the separation. In this approach, the instantaneous mixture model in the frequency domain is applied as (2.3), where the length of the impulse response is assumed to be shorter than the STFT window length. The problem definition is then simplified from a convolutional formula to a multiplicative formula and the complex-valued ICA techniques for instantaneous mixtures can then be applied independently in each frequency bin. From the relationship (2.4) defined in the frequency domain, we can show that

$$p(\mathbf{x}(f,n)|\mathbf{W}(f)) = |\mathbf{W}^{\mathsf{H}}(f)|^2 p(\mathbf{y}(f,n)), \tag{2.40}$$

$$= |\mathbf{W}^{\mathsf{H}}(f)|^2 \prod_j p(y_j(f,n)) \tag{2.41}$$

where $|\mathbf{W}^{\mathsf{H}}(f)|^2$ is the Jacobian of the complex-valued mapping $\mathbf{x}(f,n) \mapsto \mathbf{y}(f,n)$. Therefore, the negative log-likelihood of $\mathcal{X}$ given $\mathcal{W}$ is expressed as

$$\mathcal{L}_{\mathrm{FDICA}}(\mathcal{X}|\mathcal{W}) = -\sum_{j,f,n} \log p(\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)) - N \sum_f \log|\det \mathbf{W}(f)|^2. \tag{2.42}$$

By dividing $\mathcal{L}_{\mathrm{FDICA}}(\mathcal{X}|\mathcal{W})$ by the number of frames N and replacing the sample mean with the expectation operator, we obtain the normalized negative log-likelihood

$$\mathcal{L}_{\mathrm{FDICA}}(\mathcal{X}|\mathcal{W}) = -\mathbb{E}\Big[\sum_{j,f} \log p(\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f))\Big] - \sum_f \log|\det \mathbf{W}(f)|^2, \tag{2.43}$$

which is the objective function that FDICA aims to minimize. In [5], parameter

update rules modefied from the ICA optimization algorithm based on the steepest gradient descent as

$$\mathbf{W}(f) \leftarrow \mathbf{W}(f) + \eta\big\{\mathbb{E}[\psi(\mathbf{y}(f,n))\mathbf{x}^{\mathsf{H}}(f,n)] + (\mathbf{W}^{\mathsf{H}}(f))^{-1}\big\} \tag{2.44}$$

and that based on the natural gradient descent as

$$\mathbf{W}(f) \leftarrow \mathbf{W}(f) + \eta\big\{\mathbb{E}[\psi(\mathbf{y}(f,n))\mathbf{y}^{\mathsf{H}}(f,n)] + \boldsymbol{I}\big\}\mathbf{W}(f) \tag{2.45}$$

were derived. The original ICA algorithm is often called Bell-Sejnowski algorithm [38] derived from another ICA principle called Infomax approach. Here,

$$\psi(\mathbf{y}(f,n)) = \frac{\partial \log p(\mathbf{y}(f,n))}{\partial \mathbf{y}(f,n)} \tag{2.46}$$

is called activation function or score function, η is a stepsize parameter, and $\boldsymbol{I}$ is a $J \times J$ identity matrix.

Since FDICA performs paramter estimation in a frequency bin-wise manner, there is a permutation ambiguity in the separated components for each frequency. Therefore, we need to group together the separated components of different frequency bins that originate from the same source after separation, namely,

$$\mathbf{y}(f,n) = \mathrm{P}(f)\mathbf{y}(f,n). \tag{2.47}$$

Here, $\mathrm{P}(f)$ is a permutation matrix. This process is called permutation alignment. Numerous approaches have been proposed to solve the permutation alignment, including exploiting the dependence of separated signals across frequencies [43–47] and utilizing the spatial information, e.g., direction of arrival (DOA) and time difference of arrival (TDOA) [48–51]. The former is effective for sources having clearly different time structures and is robust to reverberations, whereas the latter is effective in a low reverberation and is related to sound localization. Moreover, since the ICA-based methods separate signals solely based on the signal independence,

there exists scaling ambiguities, namely,

$$\mathbf{y}(f,n) = \Lambda(f)\mathbf{y}(f,n), \tag{2.48}$$

where $\Lambda(f)$ is a diagonal matrix. The simplest way for recovering signal scales is projecting them to the observed signals, as

$$\hat{\mathbf{y}}_j(f,n) \leftarrow \mathbf{W}^{-1}(f)(\boldsymbol{e}_j \circ \mathbf{y}(f,n)), \tag{2.49}$$

where $\hat{\mathbf{y}}_j(f,n) = [\hat{y}_{1,j}(f,n),\ldots,\hat{y}_{I,j}(f,n)]^{\mathsf{T}}$ denotes the esitmated source image of source j at all the microphones whose scale is fitted to the observed signal at each mirophone. $\boldsymbol{e}_j$ is the jth column of the $J \times J$ identity matrix. This calculation is called the back projection technique [43].

2.4.2 IVA and time-varying IVA

Another solution for permutation problem is to make appropriate assumption on the probability density of signals $p(\mathbf{y}(f,n))$ to avoid the frequency bin-wise optimization problem. Typically, this solution is more preferable since the relationship between frequency bins can be used not only to solve the permutation problem but also as a clue for separation, which can lead to higher source separation performance.

IVA [6, 7] is one of such methods that simultaneously solves the BSS and permutation problem, which is a multivariate extension of FDICA. IVA models all frequency bins as a variable $\boldsymbol{y}_j(n) = [y_j(1,n),\ldots,y_j(F,n)]^{\mathsf{T}}$ that follows a spherically symmetric multivariate distribution and thus higher-order correlations between the frequency components can be considered, where the spherically symmetric property means that the distribution is a function of only the norm of multivariate vector variable, i.e., $p(\boldsymbol{y}_j(n)) = f(||\boldsymbol{y}_j(n)||)$. The normalized negative log-likelihood function of IVA is expressed as

$$\mathcal{L}_{\mathsf{IVA}}(\mathcal{X}|\mathcal{W}) = -\mathbb{E}\Big[\sum_j \log p(\boldsymbol{y}_j)\Big] - 2\sum_f \log|\det \mathbf{W}(f)| \tag{2.50}$$

$$= \sum_j \mathbb{E}\Big[\mathcal{G}(\boldsymbol{y}_j)\Big] - 2\sum_f \log|\det \mathbf{W}(f)| \tag{2.51}$$

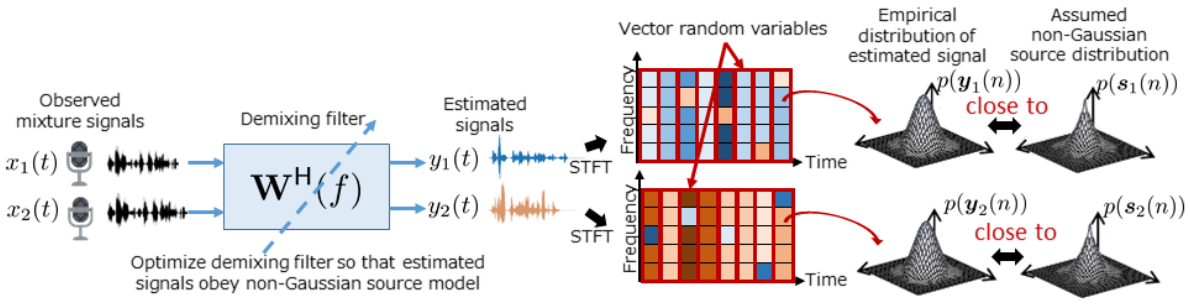

Figure 2.4: Illustration of source model in IVA, where non-Gaussian spherically symmetric source distribution $p(\boldsymbol{s}_j(n))$ is assumed for all the frames of sources.

where $\mathcal{G}(\boldsymbol{y}_j(n)) = -\log p(\boldsymbol{y}_j(n))$ is called the constrast function. Note that the ML estimation based on (2.51) is equivalent to the well-known estimation that maximizes the independence between all the sources with the KL divergence [52].

One typical choice of the probability density function is using spherically symmetric multivariate Laplace distribution [6, 7, 53] as a super-Gaussian distribution for modeling sources $\boldsymbol{s}_j(n) = [s_j(1,n), \ldots, s_j(F,n)]^{\mathrm{T}}$. The distribution is defined as

$$p(\boldsymbol{s}_j(n)) \approx p(\boldsymbol{y}_j(n)) \propto \exp\Big(-\sqrt{\sum_f |y_j(f,n)|^2}\Big) \tag{2.52}$$

with unit variance for j, f, and n. IVA based on this source distribution is called *Laplace IVA*. Therefore, the constrast function for Laplace IVA is obtained as follows:

$$\mathcal{G}(\boldsymbol{y}_j(n)) = -\log p(\boldsymbol{y}_j(n)) \stackrel{c}{=} ||\boldsymbol{y}_j(n)||_2 = \sqrt{\sum_f |y_j(f,n)|^2}. \tag{2.53}$$

Here, $||\cdot||_2$ denotes L_2 norm of a vector. Fig. 2.4 shows the source estimation of IVA, where non-Gaussian spherically symmetric source distribution $p(\boldsymbol{s}_j(n))$ is assumed for all the frames of sources.

Several optimization methods have been applied to this optimization problem, including the natural gradient descent [6, 7, 53]. Although these methods are straightforward, there is a tradeoff between the convergence speed and the stability. To address this weakness, a fast and stable parameter estimation algorithm, called AuxIVA [54], has been derived based on the auxiliary function approach.

In [54], a majorizer is designed for objective function (2.51) as follows:

$$\begin{aligned}
\mathcal{L}_{\mathrm{IVA}}(\mathcal{W}) &\propto \sum_j \mathbb{E}[\mathcal{G}(\boldsymbol{y}_j(n))] - 2\sum_f \log|\det \mathbf{W}(f)| \\
&\leq \sum_j \Big\{ \mathbb{E}\Big[\frac{G_R'(r_j)}{2r_j} \cdot \sum_f |y_j(f,n)|^2\Big] + R_j \Big\} - 2\sum_f \log|\det \mathbf{W}(f)| \\
&= \sum_j \Big\{ \sum_f \mathbf{w}_j^{\mathsf{H}}(f)\mathbb{E}\Big[\frac{G_R'(r_j)}{2r_j}\mathbf{x}(f,n)\mathbf{x}^{\mathsf{H}}(f,n)\Big]\mathbf{w}_j^{\mathsf{H}}(f) + R_j \Big\} - 2\sum_f \log|\det \mathbf{W}(f)| \\
&= \frac{1}{2}\sum_j \Big\{ \sum_f \mathbf{w}_j^{\mathsf{H}}(f)\mathbf{Q}_j(f)\mathbf{w}_j(f) + R_j \Big\} - 2\sum_f \log|\det \mathbf{W}(f)| \\
&=: \mathcal{L}_{\mathsf{IVA}}^{+}(\mathcal{W},\mathcal{Q}), \qquad (2.54)
\end{aligned}$$

where $G_R(r)$ is a continuous and differentiable function of a real variable r satisfying that $G_R'(r)/r$ is continuous everywhere and it is monotonically decreasing in $r \geq 0$. R_j is a constant term independent of $\mathbf{w}_j(f)$ for any f, and $\mathcal{Q} = \{\mathbf{Q}_j(f)\}_{j,f}$ is a set of auxiliary variable $\mathbf{Q}_j(f)$ defined as

$$\mathbf{Q}_j(f) = \mathbb{E}\Big[\frac{G_r'(r_j)}{r_j}\mathbf{x}(f)\mathbf{x}^{\mathsf{H}}(f)\Big]. \qquad (2.55)$$

The equality holds if and only if

$$r_j = ||\boldsymbol{y}_j(n)||_2 = \sqrt{\sum_f |\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)|^2}. \qquad (2.56)$$

Note that most of the IVA contrast functions used in the literature [6,7,53], including Laplace IVA, meet the conditions of $G_R(r)$, such as

$$G_R(r) = Qr, \qquad (2.57)$$

where Q is a positive constant. By setting partial derivative $\partial\mathcal{L}_{\mathsf{IVA}}^{+}/\partial\mathbf{w}_j^*(f)$ at 0, we obtain

$$\frac{1}{2}\mathbf{Q}_j(f)\mathbf{w}_j(f) - \frac{\partial}{\partial\mathbf{w}_j^*(f)} 2\log|\det \mathbf{W}(f)| = 0, \qquad (2.58)$$

where $(\cdot)^*$ denotes conjugate of complex number. Rearranging (2.58) using a matrix formula $(\partial/\partial \mathbf{W}(f)) \det \mathbf{W}(f) = \mathbf{W}^{-\mathsf{T}}(f) \det \mathbf{W}(f)$, the problem can be expressed as hybrid exact approximate joint diagonalization (HEAD) problem as

$$\mathbf{w}_{j'}^{\mathsf{H}}(f)\mathbf{Q}_j(f)\mathbf{w}_j(f) = \delta_{j'j}, \tag{2.59}$$

where the closed-form solution for updating all of $\mathbf{w}_j(f)$ simultaneously is an open problem. $\delta_{j'j}$ denotes the Kronecker delta, whose value is 1 when $j' = j$ and 0 otherwise. AuxIVA therefore proposed a sequential update rules for $\mathbf{W}(f)$, which updates $\mathbf{w}_j(f)$ while keeping other $\mathbf{w}_{j':j'\neq j}(f)$ fixed. The update rules are given as

$$\mathbf{w}_j(f) \leftarrow (\mathbf{W}(f)\mathbf{Q}_j(f))^{-1}\boldsymbol{e}_j, \tag{2.60}$$

$$\mathbf{w}_j(f) \leftarrow \mathbf{w}_j(f)/\sqrt{\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{Q}_j(f)\mathbf{w}_j(f)}. \tag{2.61}$$

Here, $\boldsymbol{e}_j$ is the jth column of the $J \times J$ identity matrix. To summarize, the algorithm of AuxIVA includes updating auxiliary variable (2.56), (2.55), and updating demixing filter (2.60), (2.61) in order for all j. This efficient update algorithm is also called the iterative projection (IP) method, which was first applied to ICA [55].

The abovementioned model ensures that all the frequency components in the same source have higher-order correlation, which solves the permutation problem. However, assuming source signal at each time frame following the same distribution is inappropriate since audio signals are time-varying. Instead of spherically symmetric Laplace distribution, in [56], the circularly symmetric complex Gaussian distribution with time-varying variance $\mathsf{v}_j(n)$ is introduced to the conventional IVA, the probability density of which is expressed as

$$p(\boldsymbol{s}_j(n)) \approx p(\boldsymbol{y}_j(n)) = \frac{1}{\pi \mathsf{v}_j(n)} \exp\Big(-\frac{|\boldsymbol{y}_j(n)|^2}{\mathsf{v}_j(n)}\Big). \tag{2.62}$$

Here, the time-varying variance $\mathsf{v}_j(n)$ is shared over the frequency bins in each time frame. Similar to (2.52), the distribution (2.62) has the spherically symmetric property and all the frequency components thus has higher-order correlation. Note that although the temporal source model $p(\boldsymbol{y}_j(n))$ is assumed to follow the Gaussian distribution, the global source model $p(\boldsymbol{Y}_j)$ becomes the super-Gaussian distribu-

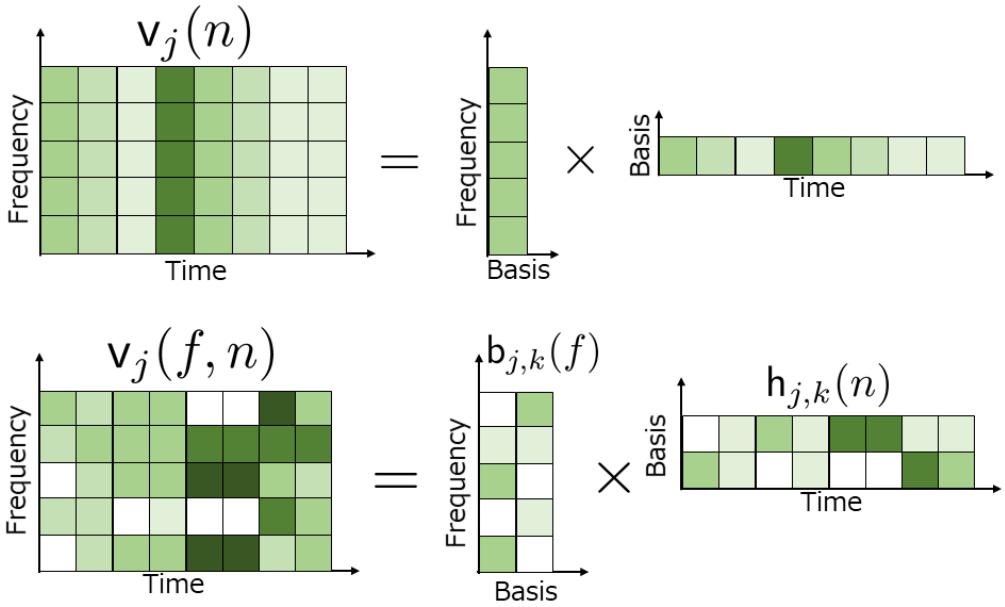

Figure 2.5: Illustration of source models (variance structures) in time-varying IVA (upper) and ILRMA (bottom), where color shade in each time-frequency bin indicates scale of variance. Time-varying IVA has frequency-uniform variance whereas ILRMA employs ISNMF as source model so that variance matrix is low-rank and can be expressed by limited number of spectral templates.

tion with $\boldsymbol{Y}_j = \{y_j(f,n)\}_{f,n}$, because of the time-varying variance [57]. IVA based on the source model (2.62) is referred to as *time-varying IVA*. The upper figure in Fig. 2.5 shows the source model of time-varying IVA. This source model amounts to assuming the magnitudes of the frequency components originating from the same source, which is expressed as a flat spectral basis, to vary coherently over time.

2.4.3 ILRMA

Although IVA and time-varying IVA can solve source separation and permutation problem simultaneously and achieve better performance than FDICA, they cannot capture the specific harmonic structures of each source since frequency-uniform variance is used for defining the source model $p(\boldsymbol{s}_j(n))$. To further increase the flexibility of modeling spectral structures, ILRMA has been proposed, which incorporates the NMF concept into the source model.

ILRMA assumes that each time-frequency bin of source $s_j(f,n)$ independently follows a zero-mean complex proper Gaussian distribution with variance $\mathsf{v}_j(f,n)$, which is called local Gaussian model (LGM) [58, 59]:

$$p(y_j(f,n)) \approx p(s_j(f,n)) = \mathcal{N}_{\mathbb{C}}(s_j(f,n)|0, \mathsf{v}_j(f,n)) \tag{2.63}$$

$$= \frac{1}{\pi \mathsf{v}_j(f,n)} \exp\Big(- \frac{|s_j(f,n)|^2}{\mathsf{v}_j(f,n)} \Big) \tag{2.64}$$

where $\mathsf{v}_j(f,n) = \mathbb{E}[|s_j(f,n)|^2]$ is the power density of signal. This is equivalent to extending the stationary distribution with uniform variance for all j, f, n assumed in the Laplace IVA or frequency-uniform variance $\mathsf{v}_j(n) = 1$ assumed in the time-varying IVA to a more flexible model with time-frequency-wise variance. Similar to the time-varying Gaussian distribution, (2.63) is a super-Gaussian distribution because of the time-varying variance, which thus can be used for ICA-based method. The negative log-likelihood function of the parameter set $\mathcal{W}$ and $\mathcal{V} = \{\mathsf{v}_j(f,n)\}_{j,f,n}$ is given as

$$\mathcal{L}_{\mathrm{ILRMA}}(\mathcal{X}|\mathcal{W},\mathcal{V}) \stackrel{c}{=} \sum_{j,f,n} \Big(\log \mathsf{v}_j(f,n) + \frac{|y_j(f,n)|^2}{\mathsf{v}_j(f,n)} \Big) - 2N \sum_f \log|\det \mathbf{W}(f)|, \tag{2.65}$$

which is the objective function of ILRMA.

Moreover, ILRMA assumes the variance matrix $\mathbf{V}_j = \{\mathsf{v}_j(f,n)\}_{f,n}$ is low-rank and can be decomposed into two matrices, which amounts to express $\mathsf{v}_j(f,n)$ as a linear sum of spectral templates $\mathsf{b}_{j,1}(f), \ldots, \mathsf{b}_{j,k}(f), \ldots, \mathsf{b}_{j,K_j}(f) \geq 0$ scaled by time-varying magnitudes $\mathsf{h}_{j,1}(n), \ldots, \mathsf{h}_{j,k}(n), \ldots, \mathsf{h}_{j,K_j}(n) \geq 0$:

$$\mathsf{v}_j(f,n) = \sum_k^{K_j} \mathsf{b}_{j,k}(f) \mathsf{h}_{j,k}(n). \tag{2.66}$$

Here, $\mathbf{B}_j = \{\mathsf{b}_{j,k}(f)\}_{k,f}$ and $\mathbf{H}_j = \{\mathsf{h}_{j,k}(n)\}_{k,n}$ are the sourcewise basis and activation matrices including K_j spectral templates and activations, respectively. Since each time-frequency bin of source $s_j(f,n)$ is assumed to follow the complex proper Gaussian distribution, which has a circularly symmetric property in the complex plane, namely, the probability only depends on the amplitude $|s_j(f,n)|$ or power $|s_j(f,n)|^2$, the time-frequency bin of complex-valued observation $x(f,n) = \sum_j s_j(f,n)$

follows complex Gaussian distribution

$$p(x(f,n)) = \frac{1}{\pi \mathsf{v}(f,n)} \exp\Big(- \frac{|x(f,n)|^2}{\mathsf{v}(f,n)} \Big) \tag{2.67}$$

because of the reproductive property in complex Gaussian distribution, where $\mathsf{v}(f,n) = \sum_j \mathsf{v}_j(f,n)$. Here, if we represent the variance $\mathsf{v}(f,n)$ as the linear sum of two low-rank matrices $\mathsf{v}(f,n) = \sum_k \mathsf{b}_k(f)\mathsf{h}_k(n)$, the negative log-likelihood function of **B** and **H** can be obtained as

$$\mathcal{L}(\mathbf{B},\mathbf{H}) = \sum_{f,n} \Big(\log \pi + \log \sum_k \mathsf{b}_k(f)\mathsf{h}_k(n) + \frac{|x(f,n)|^2}{\sum_k \mathsf{b}_k(f)\mathsf{h}_k(n)} \Big). \tag{2.68}$$

Minimizing this function is equivalent to minimizing the IS divergence between the power spectrogram of the observed signal $\mathbf{X} = \{|x(f,n)|^2\}_{f,n}$ and $\mathsf{v}(f,n)$ since

$$\mathcal{D}_{\mathrm{IS}}(\mathbf{X}|\mathbf{BH}) = \sum_{f,n} \Big(\frac{|x(f,n)|^2}{\sum_k \mathsf{b}_k(f)\mathsf{h}_k(n)} - \log \frac{|x(f,n)|^2}{\sum_k \mathsf{b}_k(f)\mathsf{h}_k(n)} - 1 \Big) \tag{2.69}$$

$$\stackrel{c}{=} \sum_{f,n} \Big(\frac{|x(f,n)|^2}{\sum_k \mathsf{b}_k(f)\mathsf{h}_k(n)} + \log \sum_k \mathsf{b}_k(f)\mathsf{h}_k(n) \Big), \tag{2.70}$$

which is also equivalent to minimizing the objective function of ILRMA (2.65) with respect to the source model $\mathsf{v}_j(f,n)$. Therefore, ILRMA can also be interpreted as a model that incorporates the NMF model based on the IS divergence into the time-varying IVA model. The bottom figure in Fig. 2.5 shows the source model of ILRMA, where the variance matrix $\mathbf{V}_j$ is represented as the linear sum of two spectral templates.

The optimization algorithm of ILRMA consists of iteratively updating the demixing matrix $\mathbf{W}(f)$, the basis templates $\mathcal{B} = \{b_{j,k}(f)\}_{f,j,k}$, and the activation matrix $\mathcal{H} = \{h_{j,k}(n)\}_{n,j,k}$. By fixing $\mathcal{B}$ and $\mathcal{H}$, the differential of (2.65) with respect to $\mathbf{W}(f)$ becomes equivalent to that of the auxiliary function in Laplace IVA. Therefore, the update rules of $\mathbf{W}(f)$ are derived based on the IP method, which are expressed as

$$\mathbf{Q}_j(f) = \mathbb{E}\Big[\frac{1}{\mathsf{v}_j(f)} \mathbf{x}(f)\mathbf{x}^{\mathsf{H}}(f) \Big], \tag{2.71}$$

$$\mathbf{w}_j(f) \leftarrow (\mathbf{W}(f)\mathbf{Q}_j(f))^{-1}\boldsymbol{e}_j, \tag{2.72}$$

$$\mathbf{w}_j(f) \leftarrow \mathbf{w}_j(f)/\sqrt{\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{Q}_j(f)\mathbf{w}_j(f)}. \tag{2.73}$$

By fixing $\mathbf{W}(f)$, the differential of (2.65) becomes equivalent to the differential of the cost function in NMF with IS divergence. Therefore, the update rules of $\mathsf{b}_{j,k}(f)$ and $\mathsf{h}_{j,k}(n)$ are give as

$$\mathsf{b}_{j,k}(f) \leftarrow \mathsf{b}_{j,k}\Big(\frac{\sum_n |y_j(f,n)|^2\mathsf{h}_{j,k}(n)/\mathsf{v}_j(f,n)^2}{\sum_n \mathsf{h}_{j,k}(n)/\mathsf{v}_j(f,n)}\Big)^{1/2}, \tag{2.74}$$

$$\mathsf{h}_{j,k}(n) \leftarrow \mathsf{h}_{j,k}\Big(\frac{\sum_f |y_j(f,n)|^2\mathsf{b}_{j,k}(f)/\mathsf{v}_j(f,n)^2}{\sum_f \mathsf{b}_{j,k}(f)/\mathsf{v}_j(f,n)}\Big)^{1/2}. \tag{2.75}$$

Since both $\mathbf{W}(f)$ and $\mathsf{v}_j(f,n)$ have scale ambiguity, the following normalization is applied at each iteration to eliminate the scale ambiguity in $\mathbf{W}(f)$:

$$\mathbf{w}_j(f) \leftarrow \mathbf{w}_j(f)z_j^{-1}, \tag{2.76}$$

$$y_j(f,n) \leftarrow y_j(f,n)z_j^{1}, \tag{2.77}$$

$$\mathsf{v}_j(f,n) \leftarrow \mathsf{v}_j(f,n)z_j^{-2}, \tag{2.78}$$

$$\mathsf{b}_{j,k}(f) \leftarrow \mathsf{b}_{j,k}(f)z_j^{-2}, \tag{2.79}$$

where

$$z_j = \sqrt{\frac{1}{FN}\sum_{f,n}|y_j(f,n)|^2} \tag{2.80}$$

is the normalization coefficient given as the sourcewise average power. Note that the normalization do not change the value of (2.65). The scale of the separated signal $y_j(f,n)$ is restored by the back-projection technique after the optimization. It is noteworthy that the log-likelihood of ILRMA is non-decreasing at each iteration of the algorithm and shown experimentally to converge quickly.

2.5 Evaluation criteria

A necessary aspect of the development of speech enhancement algorithms is how to evaluate the goodness of the enhanced speech. In general, the evaluation can be done by comparing the enhanced signals to the reference signals or listensing to the enhanced speech, namely, objective and subjective evaluations. Although subjective evaluation is generally more accurate and reliable, it needs higher humanity cost and more time. Therefore, in this thesis, we mainly use the objective evaluation.

To evaluate the distortions introduced by speech enhancement or source separation algorithms to the enhanced speech signals, we use the source-distrotion ratios (SDRs) that defined as the ratio of the energies of the reference signal and the error between the enhanced and reference signal

$$\text{SDR [dB]} = 10\log_{10}\frac{\sum_t s^2(t)}{\sum_t[\hat{s}(t)-s(t)]^2}, \tag{2.81}$$

where $s(t)$ and $\hat{s}(t)$ are reference and estimated speech signals at time t, respectively. The error between the reference and estimated signals is divided more specifically into interference, noise, and artifacts error terms as

$$\hat{s}(t) = s(t) + e_{\text{interf}}(t) + e_{\text{noise}}(t) + e_{\text{artif}}(t). \tag{2.82}$$

By using these error terms, measurements for which called sources-to-noise ratios (SNRs), source-to-interferences ratios (SIRs), and sources-to-artifacts ratios (SARs) are respectively defined as following:

$$\text{SNR [dB]} = 10\log_{10}\frac{\sum_t s^2(t)}{\sum_t e^2_{\text{noise}}(t)}, \tag{2.83}$$

$$\text{SIR [dB]} = 10\log_{10}\frac{\sum_t s^2(t)}{\sum_t e^2_{\text{interf}}(t)}, \tag{2.84}$$

$$\text{SAR [dB]} = 10\log_{10}\frac{\sum_t[s(t)+e_{\text{noise}}(t)+e_{\text{interf}}(t)]^2}{\sum_t e^2_{\text{artif}}(t)}. \tag{2.85}$$

SNR, SIR, and SAR evaluate how much noise remains in the estimated signal, how much interferences remains in the estimated signal, and how much artifacts

are generated during the signal processing, respectively. In speech enhancement tasks, since all the sources excluding the target speech are treated as noise, also the interference sources, SIR and SNR defined as above, practically measure the same error. In practical, all these criteria are calucalted using *BSS_EVAL* toolbox [60] in this thesis, which was originally implemented using MATLAB.

To evaluate the enhanced speech from various aspects, we also use short-time objective intelligibility measure (STOI) [61] to measure speech intelligibility. STOI is an objective measurement that shows a high correlation with the intelligibility of noisy speech of listening experiments. The score is given as the average of the sample envelope linear correlation between the clean and enhanced envelop vectors calculated based on the short-time segments [61,62], which is defined in $[0, 1]$. An implementation called PySTOI is given at https://github.com/mpariente/pystoi. We also use the perceptual evaluation of speech quality (PESQ) [63] to evaluate the speech quality, which is developed to model subjective evaluation used in telecommunications and is standardized ITU-T recommendation P.862. The PESQ score is given in the range of [0, 5]. Strictly speaking, PESQ is an inappropriate metric for evaluating the performance of speech processing algorithms since it uses clean speech as the test signal. However, PESQ is positively correlated with the STOI score and is thus used as a reference metric for evaluation. We use the Python implementation given at https://github.com/vBaiCai/python-pesq for calculating the PESQ score. For all the metrics mentioned here, the higher scores indicate better performance.

Chapter 3

Determined methods incorporating supervised-learned source model

3.1 Introduction

The frequency-domain BSS approach provides the flexibility of allowing us to utilize various models for the time-frequency representations of source signals, such as in IVA and ILRMA, which leads to a high source separation performance in determined situations. Owing to the fact that ILRMA reduces to time-varying IVA when it has only one flat basis spectrum, ILRMA can be interpreted as a generalized IVA method that incorporates a source model with stronger representation power, which has been shown to significantly improve source separation performance [9]. However, one drawback is that ILRMA can fail to work for sources with spectrograms that do not comply with the low-rank assumption, such as speech [9]. This indicates the importance of developing a more precise source model with stronger representation power.

Given the recent advances achieved by DNN-based speaker separation methods, including deep clustering (DC) [64,65] and permutation invariant training (PIT) [66,67], a discriminative approach has recently proved powerful in monaural source separation tasks, including both speaker-dependent and speaker-independent scenarios [68–71]. The success of these single-channel DNN-based methods attests to the excellent ability of DNNs to capture and learn the structure of spectrograms.

As an alternative to the NMF model, some attempts have also been made to incorporate DNNs for modeling the spectrograms of sources for multichannel source separation [10, 72, 73]. The idea is to replace the process for estimating the power spectra of source signals in a source separation algorithm with the forward computations of pretrained DNNs. This can be viewed as a process of refining the estimates of the power spectra of the source signals at each iteration of the algorithm. While this approach is particularly appealing in that it can take advantage of the strong representation power of DNNs for estimating the power spectra of source signals, one weakness is that unlike ILRMA, the log-likelihood is not guaranteed to be non-decreasing at each iteration of the algorithm.

On the basis of these facts, in this chapter, we introduce multichannel source separation methods using deep generative models (DGM) for source spectrogram modeling, including variational autoencoders (VAEs) [74, 75] and generative adversarial networks (GANs) [76]. We call the method using VAE source model *the multichannel variational autoencoder method* (MVAE), and that using GAN *the multichannel star GAN method* (MSGAN). Different from the algorithms of IVA and ILRMA, where source models are estimated in a blind manner, the proposed methods use a supervised pretrained source model to estimate source signals in the mixture signals. It is worth noting that there have been some attempts to apply DGM to monaural speech enhancement and source separation tasks [77–81], which was later extended into multichannel tasks [82–84]. As far as we know, the methods introduced in this chapter were the first to propose the application of VAEs and GANs to multichannel source separation. We propose two optimization algorithms for the proposed method. One is guaranteed to be non-decreasing at each iteration of the log-likelihood in order to demonstrate the full potential of the proposed method. The other is a fast algorithm to reduces computational time and cost so that it can be applied to more practical applications.

3.2 Multichannel variational autoencoder method

3.2.1 Problem formulation

Let us consider a determined situation where the number of sources equals to that of microphones, namely, $I = J$. The relationship between observed signals $\mathbf{x}(f,n)$ and source signals $\mathbf{s}(f,n)$ is described as (2.4). We assume that each source signal $s_j(f,n)$ independently follows a zero-mean complex Gaussian distribution with power spectral density $\mathsf{v}_j(f,n) = \mathbb{E}[|s_j(f,n)|^2]$:

$$s_j(f,n) \sim \mathcal{N}_{\mathbb{C}}(s_j(f,n)|0, \mathsf{v}_j(f,n)), \tag{3.1}$$

which is the same as the source model assumed in ILRMA. When $s_j(f,n)$ and $s_{j'}(f,n)(j' \neq j)$ are independent, $\mathbf{s}(f,n)$ follows

$$\mathbf{s}(f,n) \sim \mathcal{N}_{\mathbb{C}}(\mathbf{s}(f,n)|\mathbf{0}, \mathbf{V}(f,n)). \tag{3.2}$$

Namely, the separated signals $\mathbf{y}(f,n)$ approximately follows

$$\mathbf{y}(f,n) \sim \mathcal{N}_{\mathbb{C}}(\mathbf{s}(f,n)|\mathbf{0}, \mathbf{V}(f,n)), \tag{3.3}$$

where $\mathbf{V}(f,n)$ is a diagonal matrix with diagonal entries $\mathsf{v}_1(f,n), \ldots, \mathsf{v}_J(f,n)$. From the relationship between the separated signals and mixture signals given as (2.4) and (3.3), we can show that $\mathbf{x}(f,n)$ follows

$$\mathbf{x}(f,n) \sim \mathcal{N}_{\mathbb{C}}(\mathbf{x}(f,n)|\mathbf{0}, (\mathbf{W}^{\mathsf{H}}(f))^{-1}\mathbf{V}(f,n)\mathbf{W}(f)^{-1}). \tag{3.4}$$

Hence, the negative log-likelihood of the demixing matrices $\mathcal{W}$ and source model $\mathcal{V}$ given the observed mixture signals $\mathcal{X}$ is given by

$$\mathcal{L}_{\mathrm{MVAE}}(\mathcal{X}|\mathcal{W},\mathcal{V}) \overset{c}{=} \sum_{j,f,n} \Big(\log \mathsf{v}_j(f,n) + \frac{|y_j(f,n)|^2}{\mathsf{v}_j(f,n)} \Big) - 2N \sum_f \log |\det \mathbf{W}(f)|, \tag{3.5}$$

which is the same as the objective function of ILRMA (2.65). Similar to ILRMA, we need to make constraints or incorporate spectral structures into $\mathsf{v}_j(f,n)$ to elimi-

nate the permutation ambiguity during the estimation of $\mathcal{W}$. The difference is that instead of using the NMF model, which assumes sources are low-rank and estimates $\mathsf{v}_j(f,n)$ frame-wise, we train a DGM to model spectrograms of utterances so that no low-rank assumption is needed and both spectral and temporal structures of signals can be captured.

3.2.2 VAE and CVAE

VAEs [74, 75] are stochastic neural network models consisting of encoder and decoder networks. The encoder network generates a set of parameters for the conditional distribution $q_\phi(\boldsymbol{z}|\boldsymbol{s})$ of a latent space variable $\boldsymbol{z}$ given input data $\boldsymbol{s}$, whereas the decoder network generates a set of parameters for the conditional distribution $p_\theta(\boldsymbol{s}|\boldsymbol{z})$ of the data $\boldsymbol{s}$ given the latent space variable $\boldsymbol{z}$. Given a training data set $\mathcal{S} = \{\boldsymbol{s}_m\}_{m=1}^M$, VAEs learn the parameters of the entire network so that the encoder distribution $q_\phi(\boldsymbol{z}|\boldsymbol{s})$ becomes consistent with the posterior $p_\theta(\boldsymbol{z}|\boldsymbol{s}) \propto p_\theta(\boldsymbol{s}|\boldsymbol{z})p(\boldsymbol{z})$. By using Jensen's inequality, the log marginal distribution of the data $\boldsymbol{s}$ can be lower-bounded by

$$\log p_\theta(\boldsymbol{s}) = \log \int q_\phi(\boldsymbol{z}|\boldsymbol{s}) \frac{p_\theta(\boldsymbol{s}|\boldsymbol{z})p(\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{s})} \mathrm{d}\boldsymbol{z} \tag{3.6}$$

$$\geq \int q_\phi(\boldsymbol{z}|\boldsymbol{s}) \log \frac{p_\theta(\boldsymbol{s}|\boldsymbol{z})p(\boldsymbol{z})}{q_\phi(\boldsymbol{s}|\boldsymbol{s})} \mathrm{d}\boldsymbol{z} \tag{3.7}$$

$$= \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{s})}[\log p_\theta(\boldsymbol{s}|\boldsymbol{z})] - \mathrm{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{s})||p(\boldsymbol{z})], \tag{3.8}$$

where the difference between the left- and right-hand sides of (3.8) is given by

$$\log p_\theta(\boldsymbol{s}) - \int q_\phi(\boldsymbol{z}|\boldsymbol{s}) \log \frac{p_\theta(\boldsymbol{s}|\boldsymbol{z})p(\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{s})} \mathrm{d}\boldsymbol{z}$$

$$= \int q_\phi(\boldsymbol{z}|\boldsymbol{s}) \log \frac{p_\theta(\boldsymbol{s})q_\phi(\boldsymbol{z}|\boldsymbol{s})}{p_\theta(\boldsymbol{s},\boldsymbol{z})} \mathrm{d}\boldsymbol{z} \tag{3.9}$$

$$= \int q_\phi(\boldsymbol{z}|\boldsymbol{s}) \log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{s})}{p_\theta(\boldsymbol{z}|\boldsymbol{s})} \mathrm{d}\boldsymbol{z}, \tag{3.10}$$

which is equivalent to the KL divergence between $q_\phi(\boldsymbol{z}|\boldsymbol{s})$ and $p_\theta(\boldsymbol{z}|\boldsymbol{s})$. Obviously, this is minimized when

$$q_\phi(\boldsymbol{z}|\boldsymbol{s}) = p_\theta(\boldsymbol{z}|\boldsymbol{s}). \tag{3.11}$$

This means we can make $q_\phi(\boldsymbol{z}|\boldsymbol{s})$ and $p_\theta(\boldsymbol{z}|\boldsymbol{s}) \propto p_\theta(\boldsymbol{s}|\boldsymbol{z})p(\boldsymbol{z})$ consistent by maximizing the lower bound of (3.8). One typical way of modeling $q_\theta(\boldsymbol{z}|\boldsymbol{s})$, $p_\theta(\boldsymbol{s}|\boldsymbol{z})$, and $p(\boldsymbol{z})$ is to assume Gaussian districutions

$$q_\phi(\boldsymbol{z}|\boldsymbol{s}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_\phi(\boldsymbol{s}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{s}))), \tag{3.12}$$

$$p_\theta(\boldsymbol{s}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{s}|\boldsymbol{\mu}_\theta(\boldsymbol{z}), \mathrm{diag}(\boldsymbol{\sigma}^2_\theta(\boldsymbol{z}))), \tag{3.13}$$

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I}), \tag{3.14}$$

where $\boldsymbol{\mu}_\phi(\boldsymbol{s})$ and $\boldsymbol{\sigma}^2_\phi(\boldsymbol{s})$ are the outputs of an encoder network with parameter ϕ, and $\boldsymbol{\mu}_\theta(\boldsymbol{z})$ and $\boldsymbol{\sigma}^2_\theta(\boldsymbol{z})$ are the outputs of a decoder network with parameter θ. Here, it should be noted that to compute the first term of this objective function, we must compute the expectation with respect to $\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{s})$. Although this expectation cannot be expressed in an analytical form, we can compute it by using a Monte Carlo approximation. However, simply sampling $\boldsymbol{z}$ from $q_\phi(\boldsymbol{z}|\boldsymbol{s})$ does not work, since once $\boldsymbol{z}$ is sampled, it is no longer a function of ϕ, which makes it impossible to evaluate the gradient of the objective function with respect to ϕ. Fortunately, by using a reparameterization

$$\boldsymbol{z} = \boldsymbol{\mu}_\phi(\boldsymbol{s}) + \boldsymbol{\sigma}_\phi(\boldsymbol{s}) \circ \boldsymbol{\epsilon} \tag{3.15}$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{I})$ where $\circ$ indicates the element-wise product, sampling $\boldsymbol{z}$ from $q_\phi(\boldsymbol{z}|\boldsymbol{s})$ can be replaced by sampling $\boldsymbol{\epsilon}$ from the standard normal distribution, which is independent of ϕ. This allows us to compute the gradient of the first term of the objective function with respect to ϕ by using a Monto Carlo approximation of the expectation $\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\cdot]$. This technique is called a reparameterization trick. By using this reparameterization, the first term of the lower bound can be written as

$$\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{s})}[\log p_\theta(\boldsymbol{s}|\boldsymbol{z})]$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0},\boldsymbol{I})}\Big[-\frac{1}{2}\sum_{\mathrm{n}} \log 2\pi[\boldsymbol{\sigma}_\theta^2(\boldsymbol{\mu}_\phi(\boldsymbol{s}) + \boldsymbol{\sigma}_\phi(\boldsymbol{s}) \circ \boldsymbol{\epsilon})]_{\mathrm{n}} \\ - \sum_{\mathrm{n}} \frac{(\boldsymbol{s}_{\mathrm{n}} - [\boldsymbol{\mu}_\theta(\boldsymbol{\mu}_\phi(\boldsymbol{s}) + \boldsymbol{\sigma}_\phi(\boldsymbol{s}) \circ \boldsymbol{\epsilon})]_{\mathrm{n}})^2}{2[\boldsymbol{\sigma}_\theta^2(\boldsymbol{\mu}_\phi(\boldsymbol{s}) + \boldsymbol{\sigma}_\phi(\boldsymbol{s}) \circ \boldsymbol{\epsilon})]_{\mathrm{n}}}\Big], \tag{3.16}$$

where $[\cdot]_{\mathrm{n}}$ denotes the nth element of a vector. We can confirm from equation (3.16) that the second term reduces to a negative weighted squared error between $\boldsymbol{s}$ and $\boldsymbol{\mu}_\theta(\boldsymbol{\mu}_\phi(\boldsymbol{s}))$ when $\boldsymbol{\epsilon} = \mathbf{0}$, which can be interpreted as an autoencoder reconstruction error. On the other hand, the second term of (3.8) is given as the negative KL divergence between $q_\phi(\boldsymbol{z}|\boldsymbol{s})$ and $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \boldsymbol{I})$. This term can be interpreted as a regularization term that forces each element of the encoder output to be independent and normally distributed.

Conditional VAEs (CVAEs) are an extension version of VAEs where the only difference is that the encoder and decoder networks can take an auxiliary variable c as an additional input. With CVAEs, distribution (3.12) and (3.13) are replaced with

$$q_\phi(\boldsymbol{z}|\boldsymbol{s}, c) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_\phi(\boldsymbol{s}, c), \mathrm{diag}(\boldsymbol{\sigma}_\phi^2(\boldsymbol{s}, c))), \tag{3.17}$$

$$p_\theta(\boldsymbol{s}|\boldsymbol{z}, c) = \mathcal{N}(\boldsymbol{s}|\boldsymbol{\mu}_\theta(\boldsymbol{z}, c), \mathrm{diag}(\boldsymbol{\sigma}_\theta^2(\boldsymbol{z}, c))), \tag{3.18}$$

and the variational lower bound to be maximized becomes

$$\mathcal{J}(\phi, \theta) = \mathbb{E}_{(\boldsymbol{s},c)\sim p_{\mathrm{data}}(\boldsymbol{s},c)}\big[\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{s},c)}[\log p_\theta(\boldsymbol{s}|\boldsymbol{z}, c)] - \mathrm{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{s}, c)||p(\boldsymbol{z})]\big], \tag{3.19}$$

where $\mathbb{E}_{(\boldsymbol{s},c)\sim p_{\mathrm{data}}(\boldsymbol{s},c)}[\cdot]$ denotes the sample mean over the training examples $\mathcal{S} = \{\boldsymbol{s}_m, c_m\}_{m=1}^M$.

One notable feature of CVAEs is that they are able to learn a "disentangled" latent representation underlying the data of interest. For example, when a CVAE is trained using the MNIST data set of handwritten digists and c as the digit class label, $\boldsymbol{z}$ and c are disentangled so that $\boldsymbol{z}$ represents the factors of variation corresponding to handwriting styles. We can thus generate images of a desired digit with random handwriting styles from the trained decoder by specifying c and randomly sampling $\boldsymbol{z}$. Analogously, we would be able to obtain a generative model

that can represent the spectrograms of a variety of sound sources if we could train a CVAE using class-labeled training examples.

3.2.3 CVAE source model

Let $\boldsymbol{S} = \{s(f,n)\}_{f,n}$ be the entire complex spectrogram of an utterence and $\mathbf{c}$ be the class label of that source. Here, $\mathbf{c}$ is a one-hot vector consisting of C elements, indicating to which class the spectrogram $\boldsymbol{S}$ belongs. For example, if we consider speaker identities (IDs) as the class category, each element of $\mathbf{c}$ will be associated with a different speaker, and $\mathbf{c}$ will be filled with 1 at the index of a certain speaker and with 0 everywhere else.

We now model the generative model of $\boldsymbol{S}$ using a CVAE with an auxiliary input $\mathbf{c}$. So that the decoder distribution has the same form as the LGM (2.63), which is defined as a zero-mean complex Gaussian distribution,

$$p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c}) = \mathcal{N}_{\mathbb{C}}(\boldsymbol{S}|\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}^2_\theta(\mathbf{z},\mathbf{c}))), \tag{3.20}$$

$$= \prod_{f,n} \mathcal{N}_{\mathbb{C}}(s(f,n)|0, \sigma^2_\theta(f,n;\mathbf{z},\mathbf{c})). \tag{3.21}$$

Here, $\sigma^2_\theta(f,n;\mathbf{z},\mathbf{c})$ denotes the (f,n)th element of the decoder output. Once the parameter θ and ϕ of the encoder and decoder are trained by minimizing the negative variational lower bound

$$-\mathcal{J}(\phi,\theta) = -\mathbb{E}_{(\boldsymbol{S},\mathbf{c})\sim p_{\mathrm{data}}(\boldsymbol{S},\mathbf{c})}\left[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})}[\log p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})] + \mathrm{KL}[q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})||p(\mathbf{z})]\right] \tag{3.22}$$

using speaker-labeled training utterance $\{\boldsymbol{S}_m, \mathbf{c}_m\}_{m=1}^M$, the decoder with fixed θ can be used as a generative model of spectrograms for each speaker $p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j)$ at test time. Here, $p_{\mathrm{data}}(\boldsymbol{S},\mathbf{c})$ is approximated as the empirical distribution of $\{\boldsymbol{S}_m, \mathbf{c}_m\}_{m=1}^M$, and $q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})$ and $p(\mathbf{z})$ are assumed to be Gaussian distributions

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0},\boldsymbol{I}), \tag{3.23}$$

$$q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\boldsymbol{S},\mathbf{c}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{S},\mathbf{c}))), \tag{3.24}$$

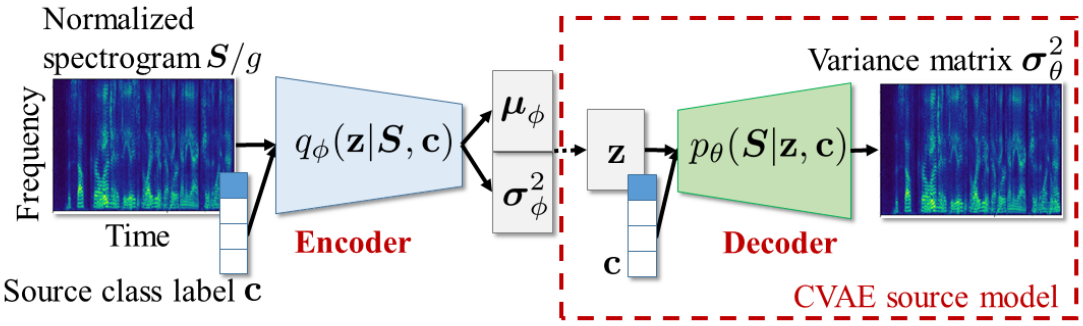

Figure 3.1: Illustration of CVAE source model used in MVAE.

$$= \prod_{\mathrm{n}} \mathcal{N}(z(\mathrm{n})|\mu_\phi(\mathrm{n};\boldsymbol{S},\mathbf{c}),\sigma^2_\phi(\mathrm{n};\boldsymbol{S},\mathbf{c})), \tag{3.25}$$

where $z(\mathrm{n})$, $\mu_\phi(\mathrm{n};\boldsymbol{S},\mathbf{c})$, and $\sigma^2_\phi(\mathrm{n};\boldsymbol{S},\mathbf{c}))$ denote the nth element of the latent space variable $\mathbf{z}$ and the encoder outputs $\boldsymbol{\mu}_\phi(\boldsymbol{S},\mathbf{c})$ and $\boldsymbol{\sigma}^2_\phi(\boldsymbol{S},\mathbf{c})$, repectively.

Normalizing the mean and variance of each training sample is one of the common practices in neural network training. Similarly, in the CVAE training in the MVAE method, the total energy of each training utterance is normalized to 1. However, of course, the total energy of the spectrogram of each source in a test mixture can vary from source to source and does not necessarily equal 1. So that the generative model can flexibly bridge this gap, a scale parameter g is additionally incorporated into (3.20) and treated as a free parameter to be estimated at test time. Namely, the generative model of the complex spectrograms $\boldsymbol{S}_j$ of utterances of speaker j can be expressed as

$$p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j,g_j) = \prod_{f,n} p_\theta(s_j(f,n)|\mathbf{z}_j,\mathbf{c}_j,g_j), \tag{3.26}$$

where

$$p_\theta(s_j(f,n)|\mathbf{z}_j,\mathbf{c}_j,g_j) = \mathcal{N}_{\mathbb{C}}(s_j(f,n)|0,g_j\sigma^2_\theta(f,n;\mathbf{z}_j,\mathbf{c}_j)), \tag{3.27}$$

and $\mathbf{z}_j$, $\mathbf{c}_j$, and g_j are the unknown parameters to be estimated. (3.26) is called the *CVAE source model*. We can immediately confirm that the decoder distribution in

(3.20) corresponds to a particular case of (3.26) where $g_j = 1$. Since the CVAE source model is given in the same form as the LGM in (2.63), where $\mathsf{v}_j(f,n)$ is given by $g_j\sigma_\theta^2(f,n;\mathbf{z}_j,\mathbf{c}_j)$. The trained decoder distribution $p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j,g_j)$ can be used as a universal generative model that is able to generate spectrograms of all the sources involved in the training examples where the latent space variable $\mathbf{z}_j$, the auxiliary input $\mathbf{c}_j$, and the global scale g_j can be interpreted as the source model parameters. Fig. 3.1 shows an illustration of the CVAE source model used in the MVAE method.

According to the properties of CVAEs, we consider that the CVAE training promotes disentaglement between $\mathbf{z}_j$ and $\mathbf{c}_j$, where $\mathbf{z}_j$ characterizes the factors of intraclass variation while $\mathbf{c}_j$ characterizes the factors of categorical variation that represent source identities. Instead of CVAE, one can also think of using a regular (unconditional) VAE, as in the VAE-NMF framework proposed for monaural speech enhancement [77, 78]. In this case, all the factors of variations in speech spectra, including the speaker identity factor, will be encoded into the latent variables. However, this can lead to an overparametrized representation since even though the speaker identity factor should be considered time-invariant (unlike phoneme- and F_0-related factors), the latent variables are allowed to vary over time. Hence, when estimating the latent variable sequence of each source in a given mixture, we would want to separate out only the speaker identity factor from the latent variable sequence and force it to be time-invariant so as not to allow it to change during the utterance. This is the motivation behind the idea of using a CVAE instead of a regular VAE.

Using the decoder distribution $p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j,g_j)$ as the generative model of each source leads to the same form of the log-likelihood as in ILRMA (2.65):

$$\log p(\mathcal{X}|\mathcal{W},\Psi,\mathcal{G}) \tag{3.28}$$

$$\begin{aligned}
=&2N\sum_f \log|\det\mathbf{W}(f)| + \sum_j \log p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j,g_j)\\
\overset{c}{=}&2N\sum_f \log|\det\mathbf{W}(f)| - \sum_{f,n,j}\left(\log g_j\sigma_\theta^2(f,n;\mathbf{z}_j,\mathbf{c}_j) + \frac{|\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)|^2}{g_j\sigma_\theta^2(f,n;\mathbf{z}_j,\mathbf{c}_j)}\right),
\end{aligned} \tag{3.29}$$

where $\mathcal{G} = \{g_j\}_j$ and $\Psi = \{\mathbf{z}_j,\mathbf{c}_j\}_j$. Since $\mathbf{z}$ is assumed to follow $\mathcal{N}(\mathbf{z}|\mathbf{0},\boldsymbol{I})$ when θ

Algorithm 1 MVAE Algorithm

Require: Network parameter θ trained using (3.22), observed mixture signal $\mathbf{x}(f,n)$, iteration number $\mathscr{L}$

randomly initialize $\mathcal{W}$, Ψ
optional: update $\mathcal{W}$ using a BSS method
for $\ell = 1$ to $\mathscr{L}$ **do**
  for each source j of J **do**
    $y_j(f,n) = \mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)$
    (updating source model parameters)
    initialize g_j using (3.34)
    normalization: $\bar{\boldsymbol{S}}_j = \{y_j(f,n)/g_j\}_{f,n}$
    for $k = 1$ to 100 **do**
      update $\mathbf{z}_j$ and $\mathbf{c}_j$ using backpropagation while keeping θ fixed
    end for
    calculate $\sigma_j^2(f,n;\mathbf{z}_j,\mathbf{c}_j,g_j=1,\theta)$
    update g_j using (3.34)
    compute $\mathsf{v}_j(f,n) = g_j \cdot \sigma_j^2(f,n;\mathbf{z}_j,\mathbf{c}_j,g_j=1,\theta)$
    (updating demixing matrices)
    update $\mathbf{w}_j(f)$ using the IP method $(3.31), (3.32),$ and (3.33)
  end for
end for

and ϕ are trained, it would be reasonable to assume it as a prior distribution for $\mathbf{z}$ also at test time. The prior $p(\mathbf{c})$ is the empirical distribution of the training examples $\{\mathbf{c}_m\}_{m=1}^M$, expressed as a multinomial distribution. Thus, the log-posterior

$$\log p(\mathcal{X}|\mathcal{W},\Psi,\mathcal{G};\theta) + \log p(\mathbf{z}) + \log p(\mathbf{c}) \tag{3.30}$$

is the objective function of the MVAE method to be maximized with respect to $\mathcal{W}$, Ψ, and $\mathcal{G}$.

3.2.4 Convergence-guaranteed optimization algorithm

A stationary point of (3.30) can be found by iteratively updating $\mathcal{W}$, Ψ, and $\mathcal{G}$ so that (3.30) is guaranteed to be non-decreasing. Since the differential of (3.30) with respect to $\mathbf{W}(f)$ is equivalent to that of the objective function of ILRMA when

Ψ and $\mathcal{G}$ are fixed, the update rules of $\mathbf{W}(f)$ are given as

$$\mathbf{Q}_j(f) = \mathbb{E}\Big[\frac{1}{\mathsf{v}_j(f)}\mathbf{x}(f)\mathbf{x}^{\mathsf{H}}(f)\Big], \tag{3.31}$$

$$\mathbf{w}_j \leftarrow (\mathbf{W}^{\mathsf{H}}(f)\mathbf{Q}_j(f))^{-1}\boldsymbol{e}_j, \tag{3.32}$$

$$\mathbf{w}_j \leftarrow \mathbf{w}_j(f)/\sqrt{\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{Q}_j(f)\mathbf{w}_j(f)}, \tag{3.33}$$

which are equivalent to the IP method. By setting the differential of (3.30) with respect to $\mathcal{G}$ at 0, we can obtain the update rule of $\mathcal{G}$ as the closed-form solution:

$$g_j \leftarrow \frac{1}{FN}\sum_{f,n}\frac{|\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)|^2}{\sigma_\theta^2(f,n;\mathbf{z}_j,\mathbf{c}_j)}. \tag{3.34}$$

Note that (3.34) maximizes (3.30) with respect to g_j when $\mathcal{W}$ and Ψ are fixed. While keeping $\mathcal{W}$ and $\mathcal{G}$ fixed, a gradient descent method, which is implemented as updating the inputs of decoder with the network parameter θ fixed using a back-propagation, can be used to search for the optimal $\mathbf{z}_j$ and $\mathbf{c}_j$ that maximize (3.30), or equivalently $\log p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j,g_j)+\log p(\mathbf{z}_j)+\log p(\mathbf{c}_j)$ for each j in parallel, where each element of $\boldsymbol{S}_j$ is given by $s_j(f,n) = \mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)$. Note that estimating $\mathbf{c}_j$ from a test mixture corresponds to identifying which source is present in the mixture. There are, however, certain cases where we know which sources are present prior to separation. Thank to the conditional modeling, we can also use our model in such cases by simply fixing $\mathbf{c}_j$ at a specified index. When updating $\mathbf{c}_j$, the sum-to-one constraint must be taken into account. This is easily implemented by inserting an appropriately designed softmax layer that outputs $\mathbf{c}_j$,

$$\mathbf{c}_j = \mathsf{softmax}(\mathbf{u}_j), \tag{3.35}$$

and treating $\mathbf{u}_j$ as the parameter to be estimated instead. The source separation algorithm of the MVAE method is summarized in *Algorithm 1*.

The proposed MVAE method is noteworthy in that it offers the advantages of the conventional methods concurrently: (1) it takes full advantage of the strong representation power of DNNs for source power spectrogram modeling, (2) the log-likelihood is guaranteed to be non-decreasing at each iteration of the source

separation algorithm by using a carefully chosen step size or applying a backtracking line search, and (3) the criteria for CVAE training and source separation are consistent, thanks to the consistency between the expressions of the CVAE source model and the LGM.

3.3 Learn source model with StarGAN

3.3.1 Motivation

Compared to the linear NMF model, the nonlinear CVAE source model not only increases the representation power but also makes it possible to capture the temporal structures of sources thanks to carefully designed network architectures for sequential modeling. However, one well-known problem as regards VAEs is that outputs from the decoder tend to be oversmoothed, which means the source spectrograms may leak spectral details. Besides VAEs, another promising approach to modeling spectrogram is GANs [76], where the generative distribution of spectrograms is optimized by playing a minimax game between a generator and a discriminator. Compared to VAEs, which explicitly assumes the prior distribution about the data, e.g., Gaussian distribution in a regular VAE or complex Gaussian distribution $\boldsymbol{S} \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{S}|\boldsymbol{\mu}_\theta(\mathbf{z},\mathbf{c}), \boldsymbol{\sigma}^2_\theta(\mathbf{z},\mathbf{c}))$ in the MVAE method, and learns data distribution by forcing an approximate posterior distributions to be consistent with the true one, GANs train a generator network to deceive a real/fake discriminator network so that the generator distribution is optimized to fit the target data distribution without explicit density assumption. This allows us to avoid the mismatch between the assumed and real distributions and the approximation error occuring in the posterior estimation. Thanks to the training strategy, it is expected that GAN can learn a data distribution more accurately than VAE. This motivates us to exploit GAN to model power spectrograms of sources.

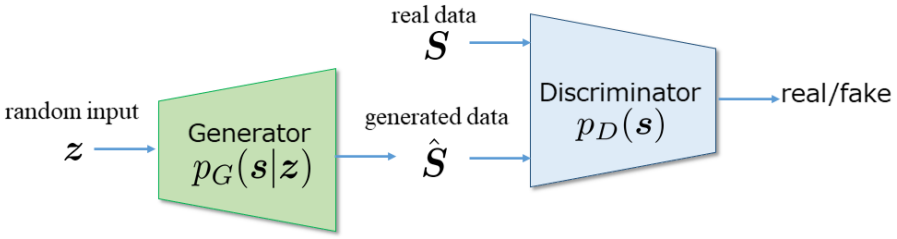

Figure 3.2: Illustration of regular GAN.

3.3.2 GAN and StarGAN

GAN is a training framework of neural network for estimating generative models, which consists of a generator G that learns the data distribution $p_G(\boldsymbol{s}|\boldsymbol{z};\theta)$ from input noise $\boldsymbol{z}$ randomly sampled from $p(\boldsymbol{z})$, which is usually defined as a Gaussian distribution $\mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\boldsymbol{I})$, and a discriminator D that estimates the probability of a sample to be real data $p_D(\boldsymbol{s};\phi)$. Here, θ and ϕ are network parameters. Fig. 3.2 shows an illustration of GAN. This framework corresponds to a minimax two-player game. The generator G is trained to maximize the probability estimated by the discriminator D to deceive it, while the discriminator D is trained to accurately classify the real and generated data as a binary classifier. By assigning the label for real data as 1 and that for fake data as 0, we can train the generator and discriminator using the following loss function:

$$\min_G \max_D \mathcal{J}(D,G) = \mathbb{E}_{\boldsymbol{s}\sim p_{\text{data}}(\boldsymbol{s})}[\log p_D(\boldsymbol{s})] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1 - p_D(p_G(\boldsymbol{s}|\boldsymbol{z})))]. \quad (3.36)$$

In practice, the generator G and discriminator D are updated iteratively during the training. By fixing the parameter θ, we can obtain the optimum of discriminator D, which is expressed as

$$p_D^{\#}(\boldsymbol{s};\phi) = \frac{p_{\text{data}}(\boldsymbol{s})}{p_{\text{data}}(\boldsymbol{s}) + p_G(\boldsymbol{s}|\boldsymbol{z};\theta)}. \quad (3.37)$$

Then, substituting the optimal discriminator distribution into (3.36), we can obtain

$$\begin{aligned}\mathcal{J}(G) &= \mathbb{E}_{\boldsymbol{s}\sim p_{\text{data}}(\boldsymbol{s})}[\log p_D^{\#}(\boldsymbol{s};\phi)] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1 - p_D^{\#}(p_G(\boldsymbol{s}|\boldsymbol{z})))] \\ &= \mathbb{E}_{\boldsymbol{s}\sim p_{\text{data}}(\boldsymbol{s})}\left[\log \frac{p_{\text{data}}(\boldsymbol{s})}{p_{\text{data}}(\boldsymbol{s}) + p_G(\boldsymbol{s}|\boldsymbol{z};\theta)}\right] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\log \frac{p_G(\boldsymbol{s}|\boldsymbol{z};\theta)}{p_{\text{data}}(\boldsymbol{s}) + p_G(\boldsymbol{s}|\boldsymbol{z};\theta)}\right]\end{aligned}$$

$$= -\log(4) + \mathrm{KL}\Big[p_{\mathrm{data}}(\boldsymbol{s})\Big|\Big|\frac{p_{\mathrm{data}}(\boldsymbol{s}) + p_G(\boldsymbol{s}|\boldsymbol{z})}{2}\Big] + \mathrm{KL}\Big[p_G(\boldsymbol{s}|\boldsymbol{z})\Big|\Big|\frac{p_{\mathrm{data}}(\boldsymbol{s}) + p_G(\boldsymbol{s}|\boldsymbol{z})}{2}\Big]$$
$$= -\log(4) + 2 \cdot \mathrm{JS}[p_{\mathrm{data}}(\boldsymbol{s})||p_G(\boldsymbol{s}|\boldsymbol{z})], \tag{3.38}$$

where $\mathrm{JS}[\cdot||\cdot]$ denote the Jensen-Shannon (JS) divergence [85]. Minimizing (3.36) with respect to G with fixed discriminator D is equivalent to force the generator distribution becomes as close as possible to the data distribution in terms of the JS divergence, where the optimal solution is achieved when the generator distribution becomes identity to the data distribution.

Although GAN has shown great success in many tasks, to stably train a GAN is difficult. One reason is the loss function defined using JS divergence, which causes gradient vanishing when two distributions are disjoint. To address this problem, extensions of GAN, such as least square GAN (LSGAN) [86] and Wasserstein GAN (WGAN) [87], have been proposed. Instead of using the cross-entropy in (3.36), LSGAN utilizes the least square loss to measure the classification accuracy of the discriminator. The loss functions for the generator and discriminator are given as

$$\min_D \mathcal{J}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{s}\sim p_{\mathrm{data}}(\boldsymbol{s})}[(p_D(\boldsymbol{s}) - b_1)^2] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[(p_D(p_G(\boldsymbol{s}|\boldsymbol{z})) - b_2)^2], \tag{3.39}$$
$$\min_G \mathcal{J}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[p_D(p_G(\boldsymbol{s}|\boldsymbol{z}) - b_3)^2]. \tag{3.40}$$

Here, b_1, b_2, and b_3 are constant values, which are usually defined as $(b_1, b_2, b_3) = (-1, 1, 0)$ or $(b_1, b_2, b_3) = (0, 1, 1)$. WGAN proposes using earth-mover distance, also named as Wasserstein-1, to measure the dissimilarity of the generator distribution $p_G(\boldsymbol{s}|\boldsymbol{z})$ and $p_{\mathrm{data}}(\boldsymbol{s})$ instead of the JS divergence, since loss function based on the JS divergence is discontinuous, which may cause the training unstable. The training loss function of WGAN is given as

$$\min_G \max_{D\in\mathbb{D}} \mathcal{J}(D, G) = \mathbb{E}_{\boldsymbol{s}\sim p_{\mathrm{data}}(\boldsymbol{s})}[p_D(\boldsymbol{s})] - \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[p_D(p_G(\boldsymbol{s}|\boldsymbol{z}))], \tag{3.41}$$

where $\mathbb{D}$ denotes a family of 1-Lipschitz continuous functions. In the WGAN, instead of classifying real and fake sample, the discriminator is trained to learn a 1-Lipschitz continuous function to help compute Wasseristein distance. Therefore,

the discriminator is called “critic”. As the loss function (3.41) decreases in the training, the Wasserstein distance gets smaller and the generator distribution grows closer to the real data distribution. To enforce the 1-Lipschitz continuity during the training, [88] proposes a practical trick called weight clipping, namely, clamping the weights ϕ to a small range $[-a, a]$ after every gradient update to keep the parameter space compact so that the function $p_D(\boldsymbol{s})$ preserves the Lipschitz continuity. Here, a is a small value usually set at 0.01. However, WGAN using weight clipping still suffers from unstable training, vanishing gradients, and slow convergence when an inappropriate clipping range is employed. To further improve the training process, [89] proposes an alternative way to enforce the Lipschitz continuity. Since a differentiable function satisfies 1-Lipschtiz if and only if it has gradients with norm at most 1 everywhere, the gradient norm of the critic's output with respect to its input can be used as a constraint. This constraint is called gradient penalty. Therefore, the training loss function of WGAN with gradient penalty (WGAN-GP) is expressed as

$$\min_D \max_G \mathcal{J}(D,G) = \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[p_D(p_G(\boldsymbol{s}|\boldsymbol{z}))] - \mathbb{E}_{\boldsymbol{s}\sim p_{\text{data}}(\boldsymbol{s})}[p_D(\boldsymbol{s})] + \lambda_{\text{grad}}\mathbb{E}_{\tilde{\boldsymbol{s}}\sim p(\tilde{\boldsymbol{s}})}[(||\nabla_{\tilde{\boldsymbol{s}}} p_D(\tilde{\boldsymbol{s}})||_2 - 1)^2], \quad (3.42)$$

where $\tilde{\boldsymbol{s}}$ is a data sampled uniformly along straight lines between pairs of points sampled from the real data distribution $p_{\text{data}}(\boldsymbol{s})$ and the generator distribution $p_G(\boldsymbol{s}|\boldsymbol{z})$. λ_{grad} is a nonnegative weight paramter and $||\cdot||_2$ denotes L_2 norm.

StarGAN [90] is a GAN variant, which consists of a generator G, a discriminator D, and a domain classifier O. The generator G is trained to translate input data $\boldsymbol{s}$ into an output data $\hat{s}$ conditioned on the target domain label c, $p_G(\hat{\boldsymbol{s}}|\boldsymbol{s}, c)$. The discriminator produces the probability of a data to be real $p_D(\boldsymbol{s})$. The domain classifier classifies to which domain the data belongs, $p_O(c|\boldsymbol{s})$. First, to make the generated data indistinguishable from real data, an adversarial loss is defined as

$$\mathcal{J}_{\text{adv}}(D,G) = \mathbb{E}_{\boldsymbol{s}\sim p_{\text{data}}(\boldsymbol{s})}[\log p_D(\boldsymbol{s})] + \mathbb{E}_{(\boldsymbol{s},c)\sim p_{\text{data}}(\boldsymbol{s},c)}[\log(1 - p_D(p_G(\hat{\boldsymbol{s}}|\boldsymbol{s}, c)))]. \quad (3.43)$$

The generator G aims to minimize this objective function, while the discriminator D

aims to maximize it. The aim of the generator is to translate $\boldsymbol{s}$ to $\hat{\boldsymbol{s}}$, which is properly classified to the target domain c. To achieve this condition, a domain classification loss is used to train the network, which is defined as

$$\mathcal{J}_{\mathrm{cls}}(O,G) = \mathbb{E}_{(\boldsymbol{s},c')\sim p_{\mathrm{data}}(\boldsymbol{s},c')}[-\log p_O(c^{'}|\boldsymbol{s})] + \mathbb{E}_{(\boldsymbol{s},c)\sim p_{\mathrm{data}}(\boldsymbol{s},c)}[-\log p_O(c|p_G(\boldsymbol{s},c))]. \tag{3.44}$$

By minimizing this objective function, the domain classifier O learns to classify a real data $\boldsymbol{s}$ to its corresponding original domain $c^{'}$ and a generated data $p_G(\boldsymbol{s},c)$ to the target domain c. The generator G is trained to minimizing this objective function to generate data that can be classified as the target domain c. Finally, a reconstruction loss is considered since minimizing the losses (3.43) and (3.44) does not guarantee that translated data preserve the content of its input while changing only the domain-related information. The reconstruction loss is expressed as

$$\mathcal{J}_{\mathrm{rec}}(G) = \mathbb{E}_{(\boldsymbol{s},c',c)\sim p_{\mathrm{data}}(\boldsymbol{s},c',c)}[||\boldsymbol{s} - p_G(p_G(\boldsymbol{s},c),c^{'})||_1], \tag{3.45}$$

where $||\cdot||_1$ denotes L_1 norm. This objective function is minimized when the generator G completely reconstructs the original data $\boldsymbol{s}$ taking the translated data $p_G(\boldsymbol{s},c)$ and the original domain label $c^{'}$ as input. The total objective functions for each network is given as

$$\mathcal{J}(D) = -\mathcal{J}_{\mathrm{adv}}, \tag{3.46}$$

$$\mathcal{J}(O) = \mathcal{J}_{\mathrm{cls}}, \tag{3.47}$$

$$\mathcal{J}(G) = \mathcal{J}_{\mathrm{adv}} + \lambda_{\mathrm{cls}}\mathcal{J}_{\mathrm{cls}} + \lambda_{\mathrm{rec}}\mathcal{J}_{\mathrm{rec}}, \tag{3.48}$$

where λ_{cls} and λ_{rec} are parameters weigh the importance of domain classification loss and reconstruction loss.

StarGAN was originally proposed for multi-domain translation [90], which has recently been adapted for use in many-to-many voice conversion [91–93] and shown to perform remarkably. This confirms the effectiveness of StarGAN for modeling audio signals.

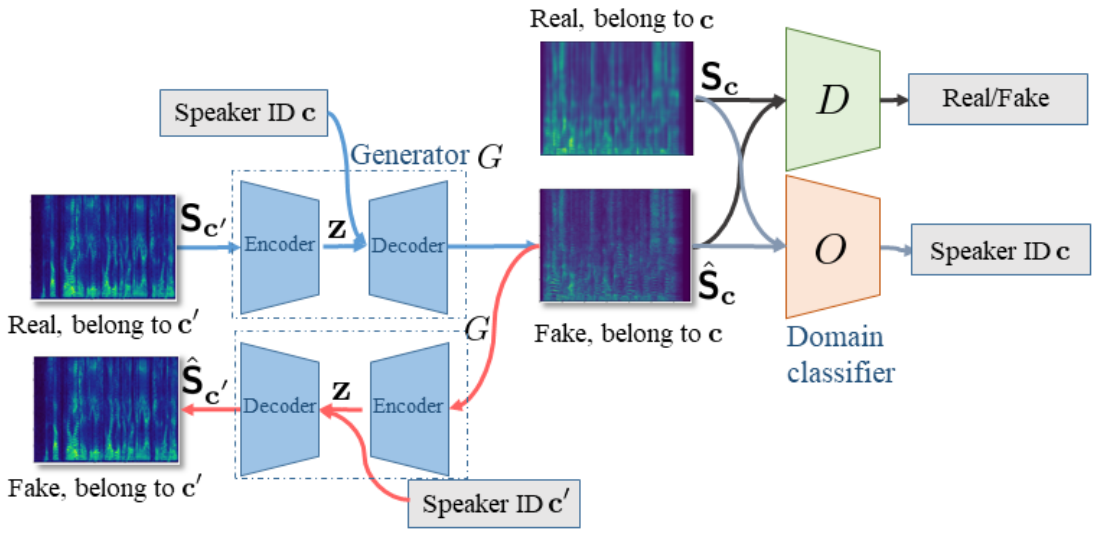

Figure 3.3: Concept of StarGAN training. Generator is designed as an encoder-decoder architecture, where trained decoder distribution is used as a source model, called StarGAN source model. Inputs of decoder, namely, $\mathbf{z}$ and $\mathbf{c}$, are parameters of source model.

3.3.3 StarGAN source model

Let us consider a power spectrogram $\mathbf{S} = \{|s(f,n)|^2\}_{f,n}$ as the data and a target speaker ID $\mathbf{c}$ as the class label, namely, we consider a translation among speaker domain, $p_G(\hat{\mathbf{S}}|\mathbf{S},\mathbf{c})$. If we use the power spectrogram $\hat{\mathbf{S}}_j$ as the variance of $\mathbf{v}_j(f,n)$, we can explain the generator as a source variance convertor, which converts variance matrix of speaker $\mathbf{c}'$ to speaker $\mathbf{c}$. One of the goals of StarGAN is to make $\hat{\mathbf{S}}$ as realistic as real spectrograms belonging to the speaker $\mathbf{c}$. To realize this we use a real/fake discriminator D to produce a probability $p_D(\mathbf{S})$ to measure how likely the power spectrogram $\mathbf{S}$ is a real spectrogram whereas we use a speaker classifier O to produce class probabilities $p_O(\mathbf{c}|\hat{\mathbf{S}})$ of $\hat{\mathbf{S}}$.

Fisrt, instead of the adversarial loss function of GAN (3.43) used in the original StarGAN, we define an adversarial loss using WGAN-GP [89], which can stablize the training procedure:

$$\mathcal{J}^D_{\mathrm{adv}}(D) = \mathbb{E}_{(\mathbf{S},\mathbf{c})\sim p_{\mathrm{data}}(\mathbf{S},\mathbf{c})}[p_D(p_G(\mathbf{S},\mathbf{c}))] - \mathbb{E}_{\mathbf{S}\sim p_{\mathrm{data}}(\mathbf{S})}[p_D(\mathbf{S})] + \lambda_{\mathrm{grad}}\mathbb{E}_{\tilde{\mathbf{S}}\sim p(\tilde{\mathbf{S}})}[(||\nabla_{\tilde{\mathbf{S}}} D(\tilde{\mathbf{S}})||_2 - 1)^2], \quad (3.49)$$

$$\mathcal{J}_{\text{adv}}^{G}(G) = -\mathbb{E}_{(\mathbf{S},\mathbf{c})\sim p_{\text{data}}(\mathbf{S},\mathbf{c})}[p_D(p_G(\mathbf{S},\mathbf{c}))]. \tag{3.50}$$

Here, $\mathbb{E}[\cdot]$ denotes sample mean, $||\cdot||_2$ denotes L_2 norm, and λ_{grad} is a nonnegative weight parameter. $\mathcal{J}_{\text{adv}}^{D}(D)$ takes a small value when D correctly classifies $p_G(\mathbf{S},\mathbf{c})$ and $\mathbf{S}$ as fake and real spectrograms whereas $\mathcal{J}_{\text{adv}}^{G}(G)$ takes a small value when G successfully deceives D so that $p_G(\mathbf{S},\mathbf{c})$ is misclassified as real spectrograms by D. Next, we consider domain classification losses for classifier O and generator G, which are defined as

$$\begin{aligned}\mathcal{J}_{\text{cls}}^{O}(O) &= -\mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}),\ \mathbf{S}\sim p(\mathbf{S}|\mathbf{c})}[\log p_O(\mathbf{c}|\mathbf{S})],\\ \mathcal{J}_{\text{cls}}^{G}(G) &= -\mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}),\ \mathbf{S}\sim p(\mathbf{S})}[\log p_O(\mathbf{c}|p_G(\mathbf{S},\mathbf{c}))].\end{aligned} \tag{3.51}$$

Both $\mathcal{J}_{\text{cls}}^{O}(O)$ and $\mathcal{J}_{\text{cls}}^{G}(G)$ take small values when O correctly classifies $\mathbf{S}\sim p(\mathbf{S}|\mathbf{c})$ and $p_G(\mathbf{S},\mathbf{c})$ as belonging to speaker $\mathbf{c}$. Training G, D, and O using only the above losses does not guarantee that G will preserve the linguistic information of the input spectrogram. To encourage $p_G(\mathbf{S},\mathbf{c})$ to be a bijection, a cycle consistency loss is also employed for training, which is expressed as

$$\mathcal{J}_{\text{cyc}}(G) = \mathbb{E}_{\mathbf{c}'\sim p(\mathbf{c}),\ \mathbf{S}\sim p(\mathbf{S}|\mathbf{c}'),\ \mathbf{c}\sim p(\mathbf{c})}[||p_G(p_G(\mathbf{S},\mathbf{c}),\mathbf{c}') - \mathbf{S}||_1], \tag{3.52}$$

where $||\cdot||_1$ denotes L_1 norm. We also consider an identity mapping loss

$$\mathcal{J}_{\text{id}}(G) = \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}),\ \mathbf{S}\sim p(\mathbf{S}|\mathbf{c})}[||G(\mathbf{S},\mathbf{c}) - \mathbf{S}||_1] \tag{3.53}$$

to ensure that an input spectrogram into G will remain unchanged when the input already belongs to the target speaker. To summarize, the full objectives of StarGAN to be minimized with respect to G, D, and C are given as

$$\mathcal{J}_G(G) = \mathcal{J}_{\text{adv}}^{G}(G) + \lambda_{\text{cls}}\mathcal{J}_{\text{cls}}^{G}(G) + \lambda_{\text{cyc}}\mathcal{J}_{\text{cyc}}(G) + \lambda_{\text{id}}\mathcal{J}_{\text{id}}(G), \tag{3.54}$$

$$\mathcal{J}_D(D) = \mathcal{J}_{\text{adv}}^{D}(D), \tag{3.55}$$

$$\mathcal{J}_O(O) = \mathcal{J}_{\text{cls}}^{O}(O), \tag{3.56}$$

respectively, where $\lambda_{\text{cls}} \geq 0,\ \lambda_{\text{cyc}} \geq 0,\ \lambda_{\text{id}} \geq 0$ are regularization parameters weigh-

ing the importance of the domain classification loss, the cycle consistency loss, and the identity mapping loss relative to the adversarial losses.

Fig. 3.3 shows an illustration of the StarGAN source model training. We design the generator as an encoder-decoder structure. The encoder aims to extract the low-dimensional latent representation $\mathbf{z}$ of the input spectrogram, whereas the decoder takes the latent variable $\mathbf{z}$ and a class index $\mathbf{c}$ as inputs and performs spectrogram conversion. The decoder distribution can be utilized as a generative model of power spectrograms, where $\mathbf{z}$ and $\mathbf{c}$ are parameters of the model. Furthermore, since the generator of StarGAN is trained as a speaker convertor for multiple speakers, the decoder can generate power spectrograms belonging to all the speakers included in the training dataset, which has the same property of that in the CVAE source model. We call the decoder distribution trained with the StarGAN the *StarGAN source model*. With the trained decoder distribution, we can employ the multichannel source separation algorithm proposed in the Subsec. 3.2.4 for determined situations, where we call the method MSGAN. Note that although the optimization algorithm of MSGAN is guaranteed to be non-decreasing at each iteration of log-likelihood as that in the MVAE method, the criteria used for training the source model and separation are different, where an adversarial loss is used for training the source model and a ML criterion is used for separation. This is different from the MVAE method.

3.4 A fast optimization algorithm for MVAE

3.4.1 Motivation and idea

It is worth noting that with the algorithm described in Subsec. 3.2.4, the model parameters can be updated so as not to decrease the log-likelihood at each iteration by using a carefully chosen step size or applying a backtracking line search. However, one downside is the high computational cost of the backpropagation process involved in each iteration, which is a major barrier to the practical application of the MVAE method. To address this drawback, in this section, we propose an accelerated version of the MVAE method called the “FastMVAE” (or fMVAE).

Since the process of updating the parameters of the CVAE source model is more computationally costly than that of updating the other parameters, our main focus is on how to accelerate this process. When $\mathcal{W}$ is fixed, each element of $\boldsymbol{S}_j$ will be fixed at $s_j(f,n) = \mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)$. Now, since the terms that depend on $\mathbf{z}_j$ and $\mathbf{c}_j$ in (3.30) are given as

$$\log p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j,g_j) + \log p(\mathbf{z}_j) + \log p(\mathbf{c}_j) \stackrel{c}{=} \log p_\theta(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j), \tag{3.57}$$

we would like to find $\mathbf{z}_j$ and $\mathbf{c}_j$ that maximize the posterior $p(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j)$ after updating $\mathcal{W}$. This posterior can be factorized as

$$p(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j) = p(\mathbf{z}_j|\boldsymbol{S}_j,\mathbf{c}_j,g_j)p(\mathbf{c}_j|\boldsymbol{S}_j,g_j). \tag{3.58}$$

Here, we notice that the first factor, $p(\mathbf{z}_j|\boldsymbol{S}_j,\mathbf{c}_j,g_j)$, resembles the encoder (or inference) distribution in the CVAE in (3.24), with the difference being that it is also conditioned on the scale parameter g_j. Since the total energy of each training utterance is assumed to be normalized to 1 in the CVAE training as mentioned earlier, g_j can be thought of as a parameter that plays the role of normalizing the total energy of an unnormalized input $\boldsymbol{S}_j$ to 1 at test time so that the scale of the encoder input is ensured to be consistent with the training utterances. Specifically, the encoder distribution that allows for unnormalized inputs is implicitly assumed to be given as the following expression:

$$q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c},g) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\boldsymbol{S}/g,\mathbf{c}),\mathrm{diag}(\boldsymbol{\sigma}_\phi^2(\boldsymbol{S}/g,\mathbf{c}))), \tag{3.59}$$

$$= \prod_{\mathrm{n}} \mathcal{N}(z(\mathrm{n})|\mu_\phi(\mathrm{n};\boldsymbol{S}/g,\mathbf{c}),\sigma_\phi^2(\mathrm{n};\boldsymbol{S}/g,\mathbf{c})), \tag{3.60}$$

which reduces to (3.24) when $g = 1$. Thus, we can use the trained encoder $q_\phi(\mathbf{z}_j|\boldsymbol{S}_j,\mathbf{c}_j,g_j)$ as an approximation of the first factor of the posterior $p(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j)$. This means that if we could obtain the true distribution $p(\mathbf{c}_j|\boldsymbol{S}_j,g_j)$ or its approximate distribution $r(\mathbf{c}_j|\boldsymbol{S}_j,g_j)$, we would be able to find an approximation of the maximum point of the posterior $p(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j)$ by finding the maximum point of the corresponding approximate distribution.

In this section, we review the concept of an auxiliary classifier VAE (ACVAE)

[94], present how this concept can be used to obtain $r(\mathbf{c}_j|\boldsymbol{S}_j, g_j)$, and introduce the details of the proposed optimization algorithm.

3.4.2 Auxiliary classifier VAE

ACVAE [94] is a CVAE variant, which incorporates an information-theoretic regularization [95] that assists in making the decoder outputs as correlated as possible with the class variable $\mathbf{c}$ by maximizing the mutual information between $\mathbf{c}$ and an output $\boldsymbol{S} \sim p_\theta(\boldsymbol{S}|\mathbf{z}, \mathbf{c})$ from the decoder, conditioned on $\mathbf{z}$. The mutual information is expressed as

$$\mathcal{I}(\mathbf{c}, \boldsymbol{S}|\mathbf{z}) = \mathbb{E}_{\mathbf{c}\sim p_{\mathrm{data}}(\mathbf{c}), \boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c}), \mathbf{c}'\sim p(\mathbf{c}|\boldsymbol{S})}[\log p(\mathbf{c}'|\boldsymbol{S})] + H(\mathbf{c}), \tag{3.61}$$

where $p_{\mathrm{data}}(\mathbf{c})$ is the empirical distribution of $\mathbf{c}$ in the training dataset, and $H(\mathbf{c})$ represents the entropy of $\mathbf{c}$, which can be considered as a constant term. Although it is difficult to optimize $\mathcal{I}(\mathbf{c}, \boldsymbol{S}|\mathbf{z})$ directly since it requires access to the posterior $p(\mathbf{c}|\boldsymbol{S})$, we can derive a variational lower bound of the first term of $\mathcal{I}(\mathbf{c}, \boldsymbol{S}|\mathbf{z})$ by using a variational distribution $r(\mathbf{c}|\boldsymbol{S})$ to approximate $p(\mathbf{c}|\boldsymbol{S})$:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{c}\sim p_{\mathrm{data}}(\mathbf{c}), \boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c}), \mathbf{c}'\sim p(\mathbf{c}|\boldsymbol{S})}[\log p(\mathbf{c}'|\boldsymbol{S})] \\
=&\mathbb{E}_{\mathbf{c}\sim p_{\mathrm{data}}(\mathbf{c}), \boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c}), \mathbf{c}'\sim p(\mathbf{c}|\boldsymbol{S})}\big[\log \frac{r(\mathbf{c}'|\boldsymbol{S})p(\mathbf{c}'|\boldsymbol{S})}{r(\mathbf{c}'|\boldsymbol{S})}\big] \\
=&\mathbb{E}_{\mathbf{c}\sim p_{\mathrm{data}}(\mathbf{c}), \boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})}\big[\mathrm{KL}[p(\mathbf{c}'|\boldsymbol{S})||r(\mathbf{c}'|\boldsymbol{S})] + \mathbb{E}_{\mathbf{c}'\sim p(\mathbf{c}|\boldsymbol{S})}[\log r(\mathbf{c}'|\boldsymbol{S})]\big] \\
\geq&\mathbb{E}_{\mathbf{c}\sim p_{\mathrm{data}}(\mathbf{c}), \boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c}), \mathbf{c}'\sim p(\mathbf{c}|\boldsymbol{S})}[\log r(\mathbf{c}'|\boldsymbol{S})] \\
=&\mathbb{E}_{\mathbf{c}\sim p_{\mathrm{data}}(\mathbf{c}), \boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})}[\log r(\mathbf{c}|\boldsymbol{S})],
\end{aligned} \tag{3.62}$$

where the equality holds if and only if $r(\mathbf{c}|\boldsymbol{S}) = p(\mathbf{c}|\boldsymbol{S})$. This technique of lower bounding mutual information is known as variational information maximization [96]. The last line of (3.62) follows the lemma presented in [95]. Therefore, we can indirectly maximize $\mathcal{I}(\mathbf{c}, \boldsymbol{S}|\mathbf{z})$ by increasing the lower bound with respect to $p_\theta(\boldsymbol{S}|\mathbf{z}, \mathbf{c})$ and $r(\mathbf{c}|\boldsymbol{S})$. One way to achieve this involves expressing the variational distribution $r(\mathbf{c}|\boldsymbol{S})$ as a neural network and training it along with $q_\phi(\mathbf{z}|\boldsymbol{S}, \mathbf{c})$ and $p_\theta(\boldsymbol{S}|\mathbf{z}, \mathbf{c})$.

Specifically, $r(\mathbf{c}|\boldsymbol{S})$ can be expressed as a multinomial distribution

$$r_\psi(\mathbf{c}|\boldsymbol{S}) = \mathrm{Mult}(\mathbf{c}|\boldsymbol{\rho}_\psi(\boldsymbol{S})). \tag{3.63}$$

Here, $\mathrm{Mult}(\mathbf{c}|\boldsymbol{\rho}) \propto \prod_{\mathrm{i}} \rho_{\mathrm{i}}^{c_{\mathrm{i}}}$ denotes a multinomial distribution, where $\mathbf{c} = [c_1, \ldots, c_{\mathrm{I}}]^{\mathsf{T}}$ and $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_{\mathrm{I}}]^{\mathsf{T}}$. $\boldsymbol{\rho}_\psi(\boldsymbol{S})$ denotes a neural network that takes $\boldsymbol{S}$ as an input and produces a probability vector consisting of C elements. (3.63) is called an auxiliary classifier. Therefore, the regularization term that we would like to maximize over the training samples with respect to ϕ, θ, and ψ becomes

$$\mathcal{J}_{\mathrm{ac1}}(\phi, \theta, \psi) = \mathbb{E}_{(\boldsymbol{S},\mathbf{c})\sim p_{\mathrm{data}}(\boldsymbol{S},\mathbf{c}), q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})}[\mathbb{E}_{\mathbf{c}\sim p_{\mathrm{data}}(\mathbf{c}), \boldsymbol{S}\sim p_\theta(\boldsymbol{S}|\mathbf{z},\mathbf{c})}[\log r_\psi(\mathbf{c}|\boldsymbol{S})]], \tag{3.64}$$

where $r_\psi(\mathbf{c}|\boldsymbol{S})$ must satisfy the sum-to-one constraint. With the regularization term (3.64), the auxiliary classifier is trained using only the reconstructed spectrograms. Since we can also use the spectrograms of real speech to train the auxiliary classifier, we can further use the cross-entropy

$$\mathcal{J}_{\mathrm{ac2}}(\psi) = \mathbb{E}_{(\boldsymbol{S},\boldsymbol{c})\sim p_{\mathrm{data}}(\boldsymbol{S},\mathbf{c})}[\log r_\psi(\boldsymbol{c}|\boldsymbol{S})] \tag{3.65}$$

as the training criterion. The entire training criterion is thus given by combining the loss function of CVAE (3.22) with regularization terms,

$$-\mathcal{J}(\phi, \theta) - \lambda_{\mathrm{ac1}}\mathcal{J}_{\mathrm{ac1}}(\phi, \theta, \psi) - \lambda_{\mathrm{ac2}}\mathcal{J}_{\mathrm{ac2}}(\psi), \tag{3.66}$$

where $\lambda_{\mathrm{ac1}} \geq 0$ and $\lambda_{\mathrm{ac2}} \geq 0$ are the parameters weighing the importance of the regularization terms. Fig. 3.4 shows an illustration of ACVAE.

3.4.3 FastMVAE algorithm

As mentioned above, the auxiliary classifier distribution $r_\psi(\mathbf{c}|\boldsymbol{S})$ trained using $\{\boldsymbol{S}_m, \mathbf{c}_m\}_{m=1}^M$ is expected to be a good approximation of the conditional distribution $p(\mathbf{c}|\boldsymbol{S})$. Now, in the same way that we considered the encoder that flexibly allows for an unnormalized input, here we also consider an auxiliary classifier $r_\psi(\mathbf{c}|\boldsymbol{S}, g)$

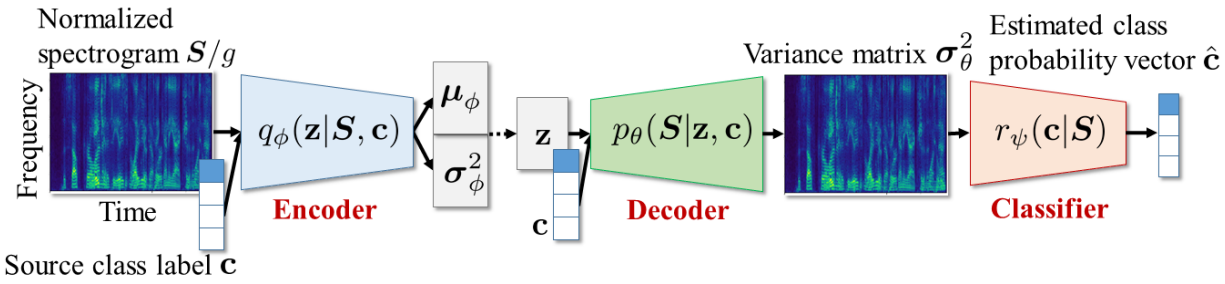

Figure 3.4: Illustration of ACVAE used in fMVAE method.

that incorporates the global scale parameter g such that

$$r_\psi(\mathbf{c}|\boldsymbol{S},g) = \mathrm{Mult}(\mathbf{c}|\boldsymbol{\rho}_\psi(\boldsymbol{S}/\mathrm{g})). \tag{3.67}$$

Using the trained auxiliary classifier and encoder, we can obtain an approximation

$$p(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j) \approx r_\psi(\mathbf{c}_j|\boldsymbol{S}_j,g_j)q_\phi(\mathbf{z}_j|\boldsymbol{S}_j,\mathbf{c}_j,g_j). \tag{3.68}$$

Since the maximum points of $r_\psi(\mathbf{c}_j|\boldsymbol{S}_j,g_j)$ and $q_\phi(\mathbf{z}_j|\boldsymbol{S}_j,\mathbf{c}_j,g_j)$ can be found immediately, we can use these approximate distributions to find an approximate solution to

$$(\mathbf{z}_j,\mathbf{c}_j) = \underset{\mathbf{z}_j,\mathbf{c}_j}{\mathrm{argmax}}\, p(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j) \tag{3.69}$$

instead of the gradient descent update for increasing $\log p_\theta(\boldsymbol{S}_j|\mathbf{z}_j,\mathbf{c}_j,g_j)+\log p(\mathbf{z}_j)+\log p(\mathbf{c}_j)$. Fig. 3.5 shows the flowchart of the proposed algorithm for the $I=2$ case. The algorithm is summarized in *Algorithm 2*. The main difference between the new algorithm from the original version is that the optimal $\mathbf{z}_j$ and $\mathbf{c}_j$ are estimated using the forward propagations of the two pretrained networks instead of using gradient descent updates. Specifically, $\mathbf{z}_j$ is given as the mean of the encoder distribution $\boldsymbol{\mu}_\phi(\boldsymbol{S}_j/g_j,\mathbf{c}_j)$. There are two possible ways to update the class variable $\mathbf{c}_j$. One is to directly use the probability vector produced by the auxiliary classifier network

$$\mathbf{c}_j \leftarrow \boldsymbol{\rho}_\psi(\boldsymbol{S}_j/g_j). \tag{3.70}$$

Algorithm 2 FastMVAE Algorithm

Require: Network parameter θ, ϕ, ψ trained using (3.66), observed mixture signal $\mathbf{x}(f,n)$, iteration number $\mathscr{L}$, weight parameter α

randomly initialize $\mathcal{W}$, Ψ
optional: update $\mathcal{W}$ using a BSS method
for $\ell = 1$ to $\mathscr{L}$ **do**
  for each source j of J **do**
    $y_j(f,n) = \mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)$
    (updating source model paremeters)
    initialize g_j using (3.34)
    normalization: $\bar{\boldsymbol{S}}_j = \{y_j(f,n)/g_j\}_{f,n}$
    update $\mathbf{c}_j$ using (3.70) or (3.71)
    update $\mathbf{z}_j$ using (3.75)
    compute $\sigma_j^2(f,n;\mathbf{z}_j,\mathbf{c}_j,g_j=1,\theta)$
    update g_j using (3.34)
    compute $\mathsf{v}_j(f,n) = g_j \cdot \sigma_j^2(f,n;\mathbf{z}_j,\mathbf{c}_j,g_j=1,\theta)$
    (updating demixing matrices)
    update $\mathbf{w}_j(f)$ by IP method with $(3.31),(3.32),(3.33)$
  end for
end for

We hereafter refer to the proposed algorithm using this update rule as *fMVAE_c*. The other is to use the one-hot vector closest to the output of the auxiliary classifier

$$[\mathbf{c}_j]_{\mathrm{n}} \leftarrow \begin{cases} 1 & (\mathrm{n} = \hat{\mathrm{n}}), \\ 0 & (\mathrm{n} \neq \hat{\mathrm{n}}), \end{cases} \tag{3.71}$$

$$\hat{\mathrm{n}} = \underset{\mathrm{n}}{\operatorname{argmax}}\, [\boldsymbol{\rho}_\psi(\boldsymbol{S}_j/g_j)]_{\mathrm{n}}, \tag{3.72}$$

where $[\cdot]_{\mathrm{n}}$ is used to denote the nth element of a vector. We hereafter refer to the algorithm using this update rule as *fMVAE_o*. Here, the subscripts are the first letters of "continuous" and "one-hot", respectively. $r_\psi(\mathbf{c}_j|\boldsymbol{S}_j,g_j)$ can be seen as a speaker recognizer trained with explicit supervision. Hence, the proposed algorithm is expected to perform better than the original version in terms of speaker identification accuracy. However, one downside would be that it does not guarantee a non-decrease in the objective function because of the approximation $p(\mathbf{z}_j,\mathbf{c}_j|\boldsymbol{S}_j,g_j) \approx r_\psi(\mathbf{c}_j|\boldsymbol{S}_j,g_j)q_\phi(\mathbf{z}_j|\boldsymbol{S}_j,\mathbf{c}_j,g_j)$. How this actually affects source separation performance will be discussed later. Note that the proposed fast algo-

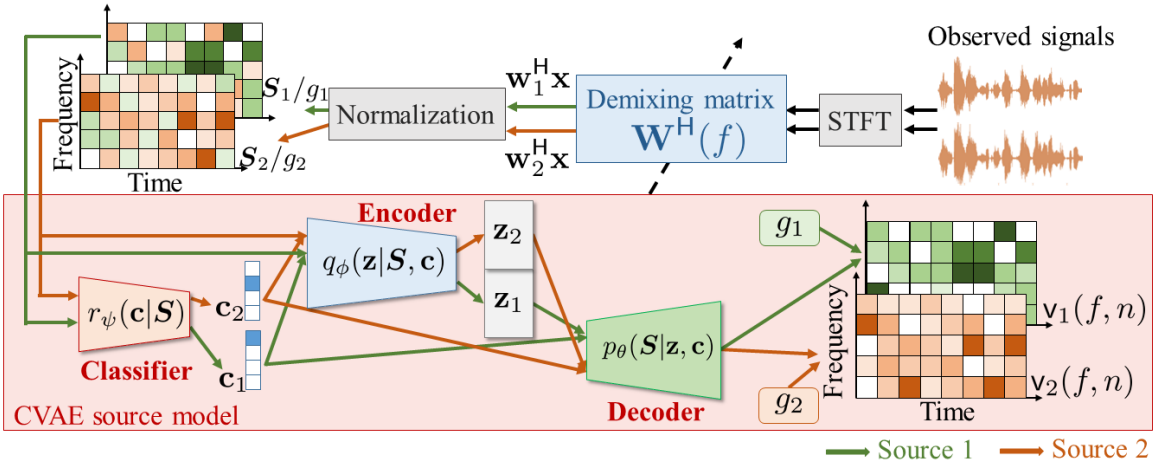

Figure 3.5: Flowchart of fMVAE for $I = 2$ case.

rithm can be applied to MSGAN, since the StarGAN source model is trained with domain classifier, and the generator is designed to have an encoder-decoder architecture, which can be used as the auxiliary classifier, encoder, and decoder in the FastMVAE method.

3.4.4 Prior-weighted inference

The encoder network is trained so that $q_\phi(\mathbf{z}|\boldsymbol{S},\mathbf{c})$ becomes as close as possible to $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0},\boldsymbol{I})$. However, through preliminary experiments, we found that at test time the trained encoder occasionally produced outliers that significantly deviated from the assumed distribution $\mathcal{N}(\mathbf{z}|\mathbf{0},\boldsymbol{I})$. This may be because the encoder did not generalize very well due to the limited amount of training data or the mismatch between the training and test conditions. Since the decoder network was trained under the assumption that its input follows $\mathcal{N}(\mathbf{z}|\mathbf{0},\boldsymbol{I})$, these outliers tended to negatively affect the resulting decoder outputs and eventually the estimate of $\mathbf{W}(f)$. One heuristic way to address this problem would be to reapply the prior distribution $p(\mathbf{z})$ during inference. In the following, we omit the source index j in this subsection for simplicity of notation.

As a way of reapplying the prior, we adopt the concept of product-of-experts

(PoE) [97] and define $\hat{\mathbf{z}}$ as

$$\begin{aligned}\hat{\mathbf{z}} &= \underset{\mathbf{z}}{\operatorname{argmax}}\, p(\mathbf{z}|\boldsymbol{S}, \mathbf{c}, g)p(\mathbf{z})^{\alpha} \\ &\approx \underset{\mathbf{z}}{\operatorname{argmax}}\, q_{\phi}(\mathbf{z}|\boldsymbol{S}, \mathbf{c}, g)p(\mathbf{z})^{\alpha} \\ &= \underset{\mathbf{z}}{\operatorname{argmax}} \log q_{\phi}(\mathbf{z}|\boldsymbol{S}, \mathbf{c}, g) + \alpha \log p(\mathbf{z}),\end{aligned} \tag{3.73}$$

where α weighs the importance of the prior in the inference. Since both $q_{\phi}(\mathbf{z}|\boldsymbol{S}, \mathbf{c}, g)$ and $p(\mathbf{z})$ are multivariate Gaussian distributions, (3.73) can be expressed as

$$\begin{aligned}&\log q_{\phi}(\mathbf{z}|\boldsymbol{S}, \mathbf{c}, g) + \alpha \log p(\mathbf{z}) \\ &\overset{c}{=} -\frac{1}{2}\big(\mathbf{z} - \boldsymbol{\mu}_{\phi}(\boldsymbol{S}/g, \mathbf{c})\big)^{\mathsf{T}}\boldsymbol{\Sigma}_{\phi}^{-1}\big(\mathbf{z} - \boldsymbol{\mu}_{\phi}(\boldsymbol{S}/g, \mathbf{c})\big) - \frac{\alpha}{2}\mathbf{z}^{\mathsf{T}}\mathbf{z} \\ &\overset{c}{=} -\frac{\boldsymbol{\Sigma}_{\phi}^{-1} + \alpha\boldsymbol{I}}{2}(\mathbf{z} - \boldsymbol{\mu})^{\mathsf{T}}(\mathbf{z} - \boldsymbol{\mu}),\end{aligned} \tag{3.74}$$

where $\boldsymbol{\Sigma}_{\phi} = \operatorname{diag}(\boldsymbol{\sigma}_{\phi}^2(\boldsymbol{S}/g, \mathbf{c}))$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma}_{\phi}^{-1}(\boldsymbol{\Sigma}_{\phi}^{-1} + \alpha\boldsymbol{I})^{-1}\boldsymbol{\mu}_{\phi}(\boldsymbol{S}/g, \mathbf{c})$. Therefore, the update rule for $\mathbf{z}$ can be easily derived as

$$\mathbf{z} \leftarrow \boldsymbol{\Sigma}_{\phi}^{-1}(\boldsymbol{\Sigma}_{\phi}^{-1} + \alpha\boldsymbol{I})^{-1}\boldsymbol{\mu}_{\phi}(\boldsymbol{S}/g, \mathbf{c}). \tag{3.75}$$

Note that (3.75) reduces to the mean of the encoder distribution when $\alpha = 0$.

3.5 Experimental evaluations

To evaluate the effectiveness of the proposed methods, we conducted several multi-speaker source separation experiments in which we considered both speaker-dependent and speaker-independent separation tasks. Specifically, the speaker-dependent and speaker-independent conditions indicate whether the test speaker is seen in the training dataset. It should be noted that even in the speaker-dependent condition, the training and test sets are disjoint at the sentence level.

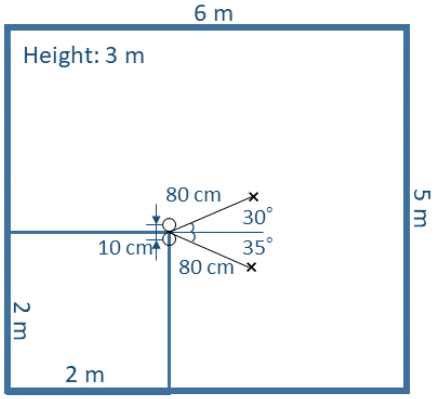

Figure 3.6: Configuration of room, where ∘ and × represent the positions of microphones and sources, respectively.

3.5.1 Dataset for speaker-dependent separation

We used speech utterances of two male speakers (SM1, SM2) and two female speakers (SF1, SF2) excerpted from the Voice Conversion Challenge (VCC) 2018 dataset [98] for the speaker-dependent source separation experiment. The audio files for each speaker were about seven minutes long and manually segmented into 116 short sentences, where 81 and 35 sentences (about five and two minutes long, respectively) served as training and test sets, respectively.

We used two-channel mixture signals of two sources as the test data, which were synthesized using simulated room impulse responses (RIRs) generated using the image method [99] and real RIRs measured in an anechoic room (ANE) and an echo room (E2A). Fig. 3.6 shows the configuration of the room used for simulating RIRs. To meet the instantaneous mixing model assumption, the reverberation times (RT_{60}) [100] of the simulated RIRs were set at 78 and 351 ms, which were controlled by setting the reflection coefficient of the walls at 0.20 and 0.80, respectively. For the measured RIRs, we used the data included in the RWCP Sound Scene Database in Real Acoustic Environments [101]. The RT_{60} of ANE and E2A were 173 and 225 ms, respectively. The test data included 4 pairs of speakers, i.e., SF1+SF2, SF1+SM1, SM1+SM2, and SF2+SM2. For each speaker pair, we generated ten mixture signals. Hence, there were a total of 40 test signals for each reverberation condition, each of which was about four to seven seconds long. All the speech signals were resampled at 16 kHz.

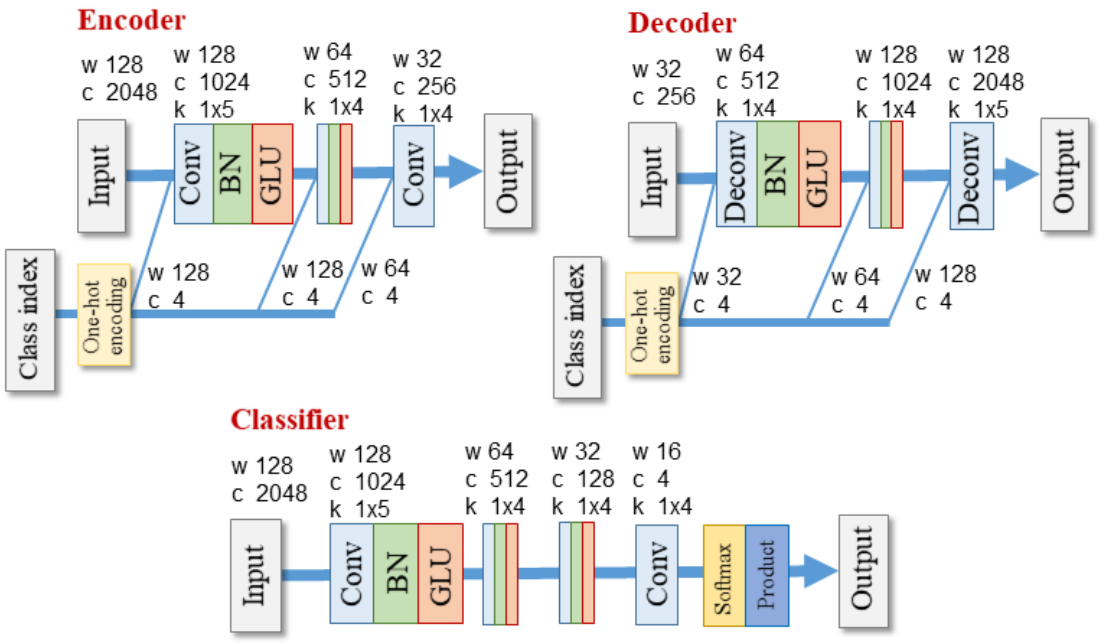

Figure 3.7: Network architectures of the encoder and decoder used for MVAE and fMVAE and the classifier used for fMVAE. The inputs and outputs are one-dimensional data, where the frequency dimension of the spectrograms is regarded as the channel dimension. The 'w', 'c', and 'k' denote the width, channel number, and kernel size, respectively. Conv and Deconv denote one-dimensional convolution and deconvolution; BN and GLU stand for batch normalization and gated linear unit.

3.5.2 Network architectures for proposed methods

Fig. 3.7 depicts the details of the network architectures employed in the MVAE and fMVAE methods. We used the same network architectures to train the CVAE and ACVAE. All the networks were designed to be fully convolutional to handle input spectrograms of signals with arbitrary lengths. We used one-dimensional gated convolutional neural networks (CNNs) [102] to model spectrograms, which allows the networks to capture time dependencies in spectral sequences. At each gated CNN layer in the encoder and decoder, a broadcast version of $\mathbf{c}$ is appended along the channel dimension to the output of the previous layer.

Gated CNNs were initially introduced to model word sequences for language modeling and shown to outperform long short-term memory (LSTM) [103] language models trained in a similar setting. By using $\mathbb{O}_{l-1}$ to denote the output of the $(l-1)$th layer, the output of the lth layer $\mathbb{O}_l$ of a gated CNN can be written

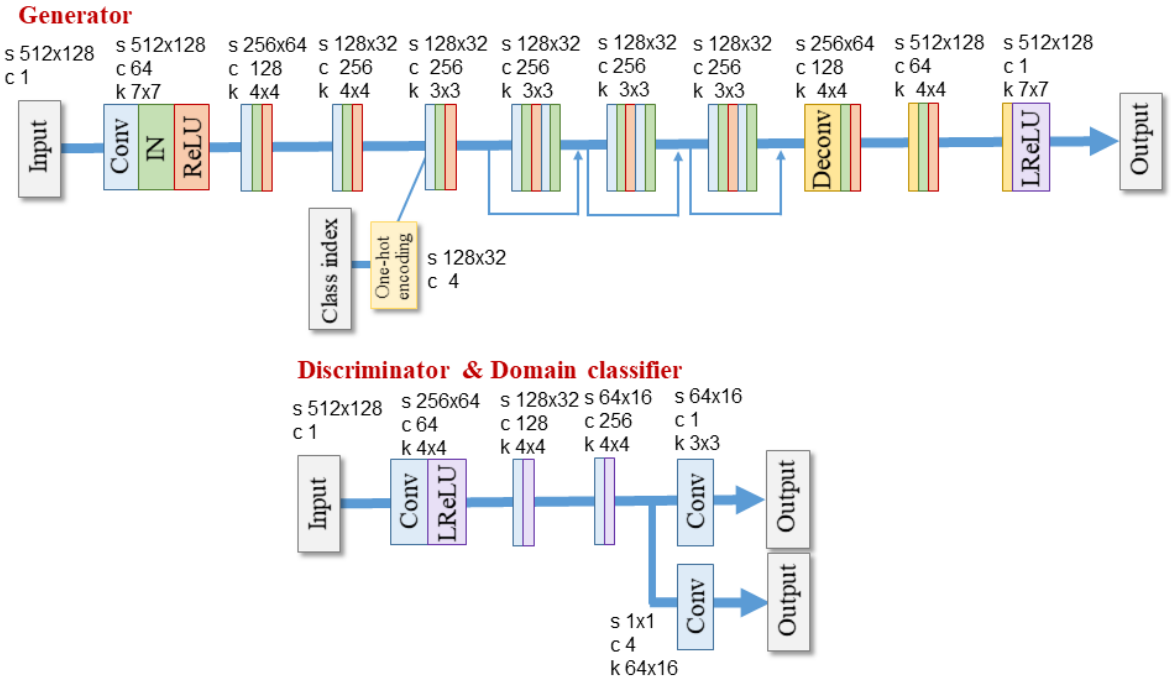

Figure 3.8: Network architectures of the generator, discriminator, and domain classifier used for MSGAN. The inputs and outputs are two-dimensional data. The 's', 'c', and 'k' denote data size, channel number, and kernel size, respectively. Conv and Deconv denote two-dimensional convolution and deconvolution; IN and LReLU stand for instance normalization and Leacky ReLU. Class index is concatenated along channel dimension.

as

$$\mathbb{O}_l = (\mathbb{O}_{l-1} * \mathbb{W}_l^{\mathrm{f}} + \mathbb{b}_l^{\mathrm{f}}) \otimes \sigma(\mathbb{O}_{l-1} * \mathbb{W}_l^{\mathrm{g}} + \mathbb{B}_l^{\mathrm{g}}), \tag{3.76}$$

where $\mathbb{W}_l^{\mathrm{f}}$, $\mathbb{W}_l^{\mathrm{g}}$, $\mathbb{B}_l^{\mathrm{f}}$, and $\mathbb{B}_l^{\mathrm{g}}$ are weight and bias parameters of the lth layer, $\otimes$ denotes element-wise multiplication, and σ is the sigmoid function. The main difference between a gated CNN and a regular CNN layer is that a gated linear unit (GLU), namely the second term of (3.76), is used as a nonlinear activation function. Like LSTMs, GLUs have data-driven gates, which control the information passed on in the hierarchy. Although it is indeed a natural choice for modeling long-term dependencies of time series data using recurrent netural networks (RNNs)-based architecutre, including LSTMs. CNNs also have excellent potential for capturing long-term structures and modeling spectrograms of audio signals. We have made

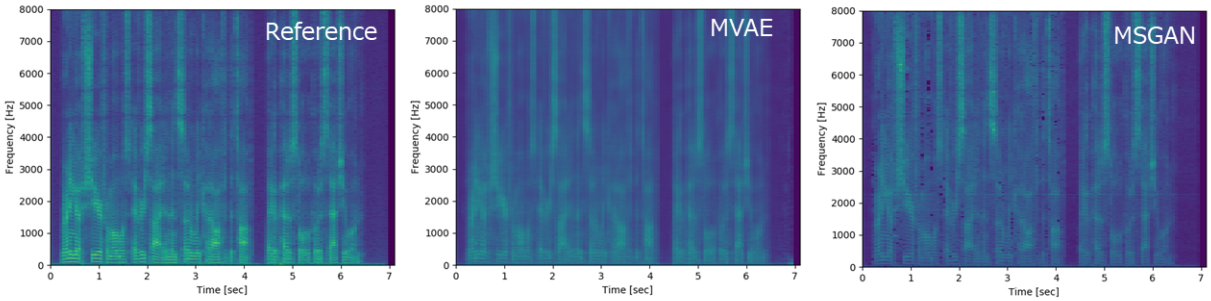

Figure 3.9: Example of CVAE and MSGAN source models obtained under 'ANE' condition.

an investigation of gated CNNs for spectrogram modeling in [I3], where we compared the source separation performance of DC [64], amounts of training data, and training time of gated CNNs with bidirectional LSTM (BiLSTM). We found that using gated CNNs could achieve comparable performance of using BiLSTMs with fewer parameters. Gated CNNs could be trained more quickly and stably. After our work, gated CNNs have been widely used for modeling spectrograms and shown remarkable performance [104, 105].

Fig. 3.8 depicts the details of the network architectures employed in the MSGAN method. We leverage the idea of Patch-GAN [106] to devise a real/fake discriminator D, the output of which is a sequence of probabilities that measures how likely each segment of the input is to be real. This forces the generator to generate more local details. Otherwise, it will fail to deceive the discriminator. The domain classifier O is designed to share the low-level features with the discriminator. We used two-dimensional convolution and transpose convolution networks for all the networks used in the MSGAN method. Adam [107] was used to train the networks.

3.5.3 Difference between VAE, CVAE, and StarGAN source models

We first made a comparsion between the CVAE and StarGAN source models. We run the MVAE and MSGAN methods for 30 iterations after initializing the demixing

Table 3.1: SDR, SIR, SAR, PESQ, and STOI achieved by MVAE and MSGAN under various reverberant conditions. The bold font indicates the beat scores.

	SDR [dB]		SIR [dB]		SIR [dB]	
Reverberant conditions	MVAE	MSGAN	MVAE	MSGAN	MVAE	MSGAN
$RT_{60} = 78$ ms	22.69	**24.08**	27.38	**28.85**	26.31	**27.21**
$RT_{60} = 351$ ms	**7.63**	6.09	**14.95**	12.34	**8.97**	8.11
ANE ($RT_{60} = 173$ ms)	19.44	**20.92**	23.73	**25.69**	23.41	**24.08**
E2A ($RT_{60} = 225$ ms)	**6.76**	6.36	**15.28**	13.98	7.94	**7.95**
average	14.13	**14.36**	**20.33**	20.21	16.66	**16.84**

	PESQ		STOI	
Reverberant conditions	MVAE	MSGAN	MVAE	MSGAN
$RT_{60} = 78$ ms	3.40	**3.50**	0.9375	**0.9480**
$RT_{60} = 351$ ms	**2.05**	1.96	**0.8221**	0.8074
ANE ($RT_{60} = 173$ ms)	3.18	**3.19**	0.9047	0.9047
E2A ($RT_{60} = 225$ ms)	**2.36**	2.31	**0.7666**	0.7585
average	**2.75**	2.74	**0.8577**	0.8547

matrix $\mathcal{W}$ by running ILRMA for 30 iterations. The spectral templates number K for ILRMA was set at 1. Adam [107] was used to estimate the source model parameter $\Psi = \{\mathbf{z}_j, \mathbf{c}_j\}_j$ in both algorithms.

Table 3.1 shows SDR, SIR, SAR, PESQ, and STOI scores obtained by MVAE and MSGAN methods. All the results were averaged over the 40 test signals under each reverberant condition. The results reveal that MSGAN slightly outperformed MVAE in terms of SDR and achieved comparative results in terms of other criteria. Comparing the results under each reverberant condition, we find that MSGAN performed better in low reverberant situations and the performance degraded with relatively heavy reverberantion. Fig. 3.9 depicts an example of the power spectrograms estimated by different methods. We found that the CVAE and StarGAN source models used in the MVAE and MSGAN methods can precisely capture spectro-temporal structures of sources. Moreover, the StarGAN source model could represent more details of harmonics than the CVAE source model, while it might lead to more unexpected distortions in local. Considering the training difficulty of StarGAN and the limited improvement of the StarGAN source model, we

Table 3.2: Average SDR, SIR, SAR, PESQ, and STOI scores achieved by MVAE with CVAE and VAE for source modeling. The bold font indicates the best scores.

Method	SDR [dB]	SIR [dB]	SAR [dB]	PESQ	STOI
MVAE(VAE)	15.35	20.30	17.91	**2.72**	0.8495
MVAE(CVAE)	**17.03**	**23.75**	**18.61**	2.24	**0.8717**

Table 3.3: Methods for comparison

Category	Method	Notation	Initialization
unsupervised uninformed	ILRMA	Baseline1: u.u.ILRMA	random
supervised uninformed	ILRMA	Baseline2: s.u.ILRMA	random
	MVAE	Baseline3: s.u.MVAE	random/IVA/u.u.ILRMA
	fMVAE_o	Proposed1: s.u.fMVAE_o	random/IVA/u.u.ILRMA
	fMVAE_c	Proposed2: s.u.fMVAE_c	random/IVA/u.u.ILRMA
supervised informed	ILRMA	Baseline4: s.i.ILRMA	random
	IDLMA	Baseline5: s.i.IDLMA	random
	MVAE	Baseline6: s.i.MVAE	random/IVA/u.u.ILRMA
	fMVAE	Proposed3: s.i.fMVAE	random/IVA/u.u.ILRMA

thought the CVAE source model was more preferable. Therefore, we compared source separation between the baseline methods with the MVAE mothed.

We also confirmed the effectiveness of conditional modeling by comparing the performance obtained with the CVAE source model and its unconditional counterpart under the MVAE framework. Table 3.2 shows SDR, SIR, SAR, PESQ, and STOI scores. As can be seen from the results, the CVAE source model obtained a 1.7-dB higher SDR than a source model based on a regular VAE.

3.5.4 Baseline methods for comparison

We chose ILRMA [9] and IDLMA [73] as the baseline methods for comparison. We tested several different versions of the proposed and baseline methods. We use the terms “supervised/unsupervised” and “informed/uninformed” to properly categorize each version of the methods. The terms “supervised” and “unsupervised” indicate whether a method requires training examples of source signals prior to

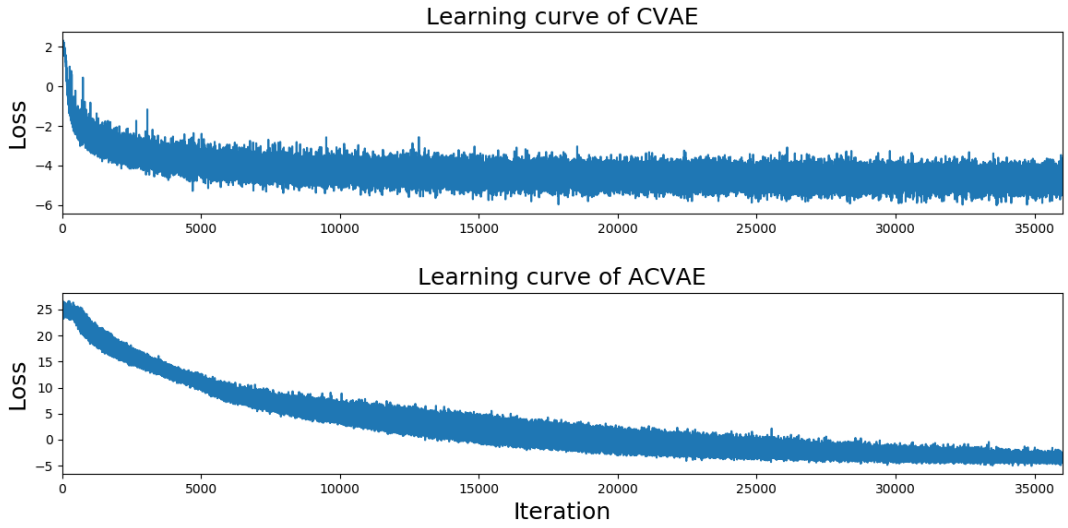

Figure 3.10: Learning curves of CVAE and ACVAE source models.

source separation, while the terms “informed” and “uninformed” indicate whether a method is informed about which sources are present in a test mixture signal. Categorization of each version is summarized in Table 3.3.

We set the basis number $K_j = 10$ for u.u.ILRMA and randomly initialized the basis spectra and activation matrix. For supervised ILRMA, basis spectra with $K = 10$ were pretrained for each speaker in the training dataset using the NMF algorithm. They were then concatenated and used as a unified model to represent all the sources in s.u.ILRMA, whereas the basis spectra corresponding to the specific speakers present in a mixture signal were provided to the method in s.i.ILRMA. Note that *Algorithm 1* and *Algorithm 2* correspond to s.u.MVAE and s.u.fMVAE_o/s.u.fMVAE_c, respectively. For s.i.MVAE and s.i.fMVAE, the correct class label $\mathbf{c}_j$ is given and fixed during the update. Fig. 3.10 shows the learning curves of the CVAE and ACVAE training processes. The curves demonstrate that the networks were trained stably with fast convergence.

For s.i.IDLMA, we used a fully connected neural network with four hidden layers. Each layer had 1024 units, and a rectified linear unit was used for the output of each layer, which was the same as the network architecture used in [73]. We implemented the training settings described in [73], namely using the Gaussian-IDLMA loss function and concatenation of the current, preceding, and succeeding

Table 3.4: Average SDR [dB] obtained with various STFT settings. The bold font shows the best scores.

Method	Window length [ms]			
	32	64	128	256
s.u.MVAE	10.91	13.38	**14.01**	12.27
s.i.MVAE	10.84	13.41	**13.76**	12.47
s.u.fMVAE_o	11.63	12.11	**14.67**	13.85
s.u.fMVAE_c	4.31	10.36	**13.51**	13.26
s.i.fMVAE	11.57	12.25	**14.76**	14.13

frames to capture the temporal dependency, data augmentation, and regularization. The only difference was the optimization algorithm, where we used Adam to train the network for 700 epochs instead of Adadelta [108] for 200 epochs. More training details are available in [73].

3.5.5 Experimental analysis of hyperparameters and source separation performance

In this subsection, we compare the separation performance across different STFT window lengths, different initialization methods for the MVAE and fMVAE algorithms, and different α settings.

Since all the methods are based on the instantaneous linear mixture model, the STFT window length may affect the separation performance of each of them, especially under reverberant conditions. We computed the STFT using a Hamming window with a length of $\{32, 64, 128, 256\}$ ms, and by shifting half of the length for each frame. In this experiment, all the MVAE and fMVAE methods were initialized by running u.u.ILRMA for 30 iterations. The MVAE or fMVAE algorithm was then run for 30 iterations, where Adam was used to update $\mathbf{z}_j$ and $\mathbf{c}_j$ in the MVAE methods with a step size set of 0.01. We used $\alpha = 0$ for fMVAE in this experiment. Table 3.4 shows the SDR scores obtained with each method. From these results, the optimal window length that gave the best overall performance was 128 ms for the current dataset. Therefore, we conducted all the following experiments using a window length of 128 ms.

Table 3.5: Average SDR [dB] obtained by MVAE and fMVAE methods adopting different initialization approaches. The bold font shows the best scores.

Method	Initialization		
	random	IVA	ILRMA
s.u.MVAE	**17.03**	12.58	14.01
s.i.MVAE	**16.58**	12.45	13.76
s.u.fMVAE_o	14.26	13.67	**14.67**
s.u.fMVAE_c	**13.78**	12.62	13.51
s.i.fMVAE	**14.93**	13.82	14.76

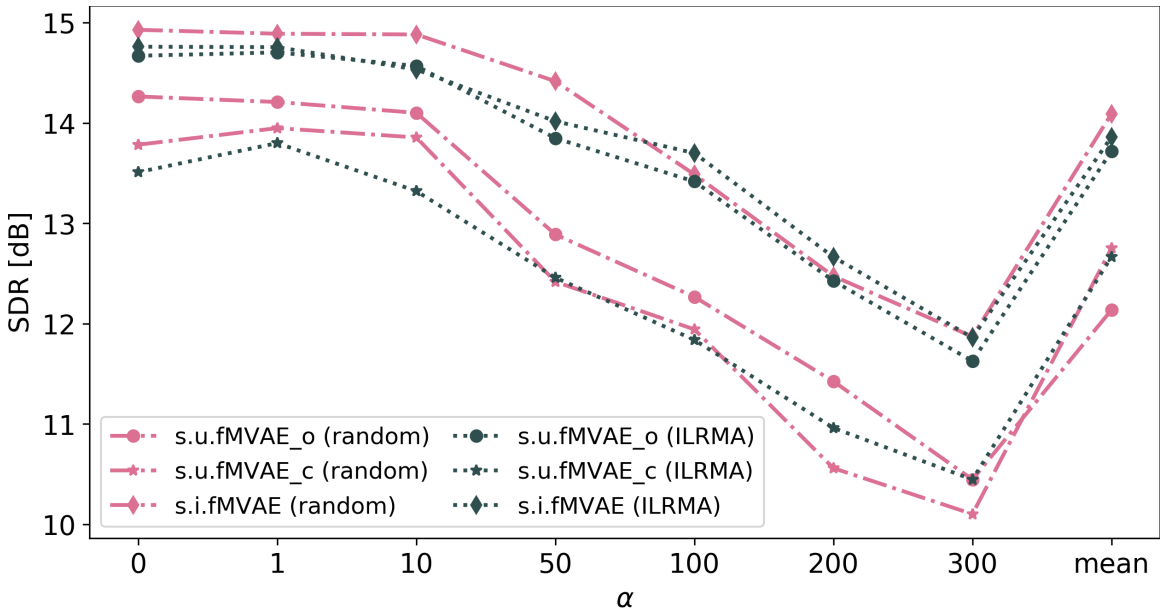

Figure 3.11: Average SDR achieved with various α in a speaker-dependent condition.

To confirm the impact of the initialization for the MVAE and fMVAE methods on the source separation performance, we compared the algorithms using the following three initialization methods: 1) random initialization with the demixing matrices initialized at identity matrices; 2) IVA; and 3) u.u.ILRMA. To keep the number of updates of the demixing matrices constant, each algorithm was run for 60 iterations for the random initialization case and 30 iterations after an initialization algorithm was run for 30 iterations for the other cases. Table 3.5 shows the SDR scores over the 160 test samples. From these results, we found that the methods adopting ILRMA for initialization achieved better performance than those using IVA for ini-

tialization. One possible reason could be that block permutation had occurred in IVA. It is worth noting that the MVAE methods with random initialization obtained more than 3 dB higher SDR improvements than when using IVA and ILRMA for initialization. Meanwhile, though random initialization slightly outperformed ILRMA in s.u.fMVAE_c and s.i.fMVAE, there were no noticeable differences. Therefore, we adopted random initialization in the following experiments.

Finally, we investigated how much the performance depends on the weight parameter α in the prior-weighted inference. We set α at $\{0, 1, 10, 50, 100, 200, 300, \text{mean}\}$, where "mean" indicates the data-dependent setting

$$\alpha = \frac{1}{\mathrm{N}} \sum_{\mathrm{n}}^{\mathrm{N}} \sigma_{\phi}^{2}(\mathrm{n}; \boldsymbol{S}, \mathbf{c}). \tag{3.77}$$

Fig. 3.11 shows the average SDR scores over 160 test signals. We found that the effectiveness of the prior distribution $p(\mathbf{z})$ in improving the source separation performance was modest in the speaker-dependent case and that the SDRs started to decrease at $\alpha > 10$, which indicates that a smaller value leads to better performance for the speaker-dependent case. Moreover, the curve of fMVAE_o was entirely above the curve of fMVAE_c without regard for the choice of the initialization methods, which indicates that fMVAE_o is more effective in speaker-dependent scenarios.

Table 3.6 shows scores obtained by each method with the optimal parameter setting. By comparing supervised methods to the blind method (u.u.ILRMA), we confirmed that an appropriately pretrained source model could lead to considerably improved source separation performance. The MVAE methods achieved the best scores in both the uninformed and informed categories, which significantly outperformed the other methods. The fMVAE method yielded an average SDR score that was 2.8 dB lower than the original MVAE method, but about 0.75 dB higher than the other baseline methods.

Table 3.6: Average SDR, SIR, SAR, PESQ, and STOI scores achieved by each method with the optimal parameter setting. The bold font indicates the best scores.

Method	SDR [dB]	SIR [dB]	SAR [dB]	PESQ	STOI
u.u.ILRMA	12.36	17.77	15.29	1.83	0.8345
s.u.ILRMA	13.50	19.01	16.60	1.92	0.8367
s.u.MVAE	**17.03**	**23.75**	**18.61**	2.24	**0.8717**
s.u.fMVAE_o	14.26	19.89	16.71	2.07	0.8454
s.u.fMVAE_c	13.95	19.54	16.33	2.66	0.8452
s.i.ILRMA	13.30	18.60	17.02	1.91	0.8355
s.i.IDLMA	14.15	21.11	15.59	1.77	0.8692
s.i.MVAE	16.58	22.87	18.40	**2.84**	0.8641
s.i.fMVAE	14.93	21.00	16.98	2.73	0.8548

Table 3.7: Computational times of MVAE and fMVAE methods with random initialization.

Processor	Method	rumtime/iteration [sec]	total [sec]
GPU	s.u.MVAE	2.8147	172.5241
	s.u.fMVAE_o	**0.0367**	**5.5661**
	s.u.fMVAE_c	**0.0365**	**5.5372**
CPU	s.u.fMVAE_o	0.0979	8.6823
	s.u.fMVAE_c	0.0969	8.7434

3.5.6 Computational time

The average computational times of the MVAE and fMVAE methods with random initialization are summarized in Table 3.7. All the programs were run using an Intel (R) Core i7-7800X CPU@3.50 GHz and a TITAN V GPU with 12-GB memory. Here, “runtime/iteration” means the computational time required to update the parameters once using the MVAE or fMVAE algorithm. The “total” time indicates the time taken by the entire process, including the time for constructing the system (e.g., loading the pretrained networks to a GPU), updating parameters, and performing the separation. Through the comparison of the runtime at each iteration, we found that the fMVAE algorithm was about 70 times faster than the MVAE algorithm. Moreover, fMVAE was found to reduce the computational time by more than 90% even when using a CPU. These results indicate a tradeoff between the source

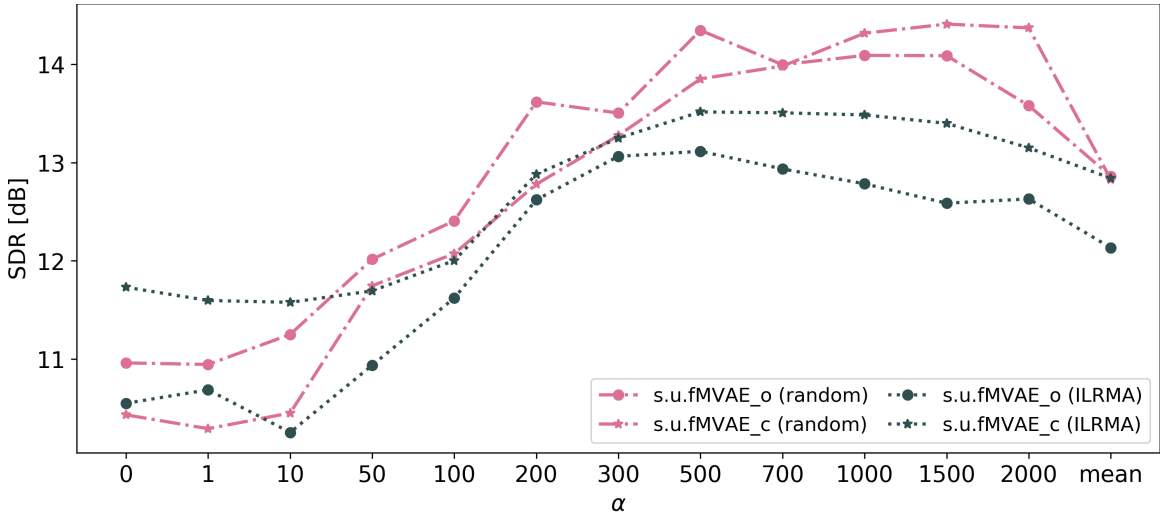

Figure 3.12: Average SDR over 200 test signals achieved with various α.

separation performance and computational time: the MVAE method provides better separation performance with high computational cost, whereas fMVAE significantly reduces computational cost but with performance degradation.

3.5.7 Speaker-independent separation

In practical applications, the speakers in a given mixture signal are not always included in the training dataset. In this subsection, we show the performance of the MVAE and fMVAE methods in speaker-independent tasks and compare them with u.u.ILRMA, which requires no prior information about the speakers.

We created datasets using utterances from the Wall Street Journal (WSJ0) corpus [109]. All the utterances in WSJ0 folder `si_tr_s` (around 25 hours) were used as the training set, which consists of 101 speakers in total. If there is a large number of utterances of a sufficiently wide variety of speakers in the training dataset, the trained model is expected to have an ability to express spectrograms of unseen speakers. When a test mixture contains unseen speakers, (3.70) can be interpreted as how similar speaker j is to the speakers in the training set, whereas (3.71) indicates the speaker in the training set most similar to speaker j. A test set was created by randomly mixing two different speakers selected from the WSJ0 folders `si_dt_05` and `si_et_05`, where the number of speakers was 18. We generated test data using simulated RIRs with $RT_{60} = 78$ ms and $RT_{60} = 351$ ms,

Table 3.8: Average SDR, SIR, SAR, PESQ, and STOI scores obtained with uninformed methods. The bold font shows the best scores.

Method	SDR [dB]	SIR [dB]	SAR [dB]	PESQ	STOI
u.u.ILRMA	13.76	19.94	17.09	**3.05**	0.8727
s.u.MVAE	**17.58**	**25.13**	**19.26**	2.65	**0.8934**
s.u.fMVAE_o	14.35	21.06	17.25	3.04	0.8746
s.u.fMVAE_c	14.41	21.21	17.35	3.04	0.8776

where 100 mixture signals were generated under each reverberation condition. The average SDRs of the datasets were about 0.60 dB and -0.78 dB, respectively. Other experimental conditions and network architectures were the same as those described in Subsec. 3.5.1.

As in the speaker-dependent case, we first investigated the dependence of the separation performance on the α setting. Fig. 3.12 shows the average SDR scores over the entire test dataset achieved with various α settings. Since the scores obtained with $\alpha = 200$ and $\alpha = 300$ increased continuously, we additionally evaluated the performance obtained when $\alpha = \{500, 700, 1000, 1500, 2000\}$. The optimal α settings were 500 for s.u.fMVAE_o and 2000 for s.u.fMVAE_c, respectively. This was considerably different from the speaker-dependent case, where a smaller α performed better. From these results, we can assume that the proposed prior-weighted update rule was more effective under open-set conditions than under closed-set conditions.

Table 3.8 summarizes the average SDR, SIR, SAR, PESQ, and STOI scores obtained with each method with random initialization. The results demonstrate the ability of the MVAE and fMVAE methods to handle speaker-independent scenarios with an increasing variety and amount of training data. Both the MVAE and fMVAE methods were superior to u.u.ILRMA, where s.u.MVAE achieved an improvement of more than 3.5 dB over u.u.ILRMA. As with the speaker-dependent case, the fMVAE methods provided less improvement than the MVAE method.

3.6 Summary of chapter 3

In this chapter, we proposed two determined BSS methods, namely, MVAE and MSGAN. The proposed methods incorporate DGM-based source models into the FDICA-based BSS framework to capture the spectro-temporal structures of sources so that the structures can be used as a clue to solve the permutation problem and improve the source separation performance. The MVAE method uses a CVAE to train the source model, whereas the MSGAN method uses a StarGAN to train the source model. We made a comparison between both models. The experimental results showed that the StarGAN source model could lead to a slight improvement in terms of SDR. By considering the training difficulty of the StarGAN and the limited improvement, we thought the CVAE source model was preferable. Both MVAE and MSGAN are noteworthy in that the log-likelihood of signals are guaranteed to be non-decreasing at each iteration. However, the computational cost and time are high. We proposed a fast parameter optimization algorithm for the MVAE method, called FastMVAE, which uses ACVAE for training the CVAE source model. With the trained auxiliary classifier and encoder, we are allowed to search for the parameters that approximately maximizes the posterior. FastMVAE has been shown to significantly reduce computational time by more than 90% compared with the original MVAE method. The experimental evaluation showed that the MVAE method and FastMVAE method could handle both speaker-dependent and speaker-independent scenarios, which outperformed conventional methods.

Chapter 4

Directional speech enhancement using geometry information

4.1 Introduction

In this chapter, we consider using the geometry of microphone arrays as prior information to guide the demixing matrices estimated by BSS methods. Although ILRMA has shown to outperform IVA in terms of source separation performance, IVA, especially AuxIVA, has still attracted much attention and been widely studied due to the fast and stable optimization algorithm and its online extension [110–112]. However, when considering practical applications of speech enhancement, an additional process is necessary for selecting the target speech after the separation, which is typically performed by utilizing the spatial information, i.e., DOA of the target. Moreover, it is reported that block permutation problem occurs between the low- and high-frequency bands in IVA, which results in the degradation of the performance [113] though IVA is theoretically able to solve the permutation problem. One promising approach to eliminate the block permutation is exploiting spatial information to guide the demixing matrices $\mathcal{W}$. For example, [114] derives IVA in a maximum a posterior (MAP) fashion so that a spatially informed prior of demixing matrices can be incorporated into the optimization. Another well-known framework is the geometrically constrained BSS [11–13, 115, 116]. In [13], a penalty term restricting the Euclidean angle between the separation filter and the far-field steering

vector calculated from the desired source DOA is combined with IVA to force the desired signal always being output at the corresponding channel. However, there are two drawbacks to prevent this method from a wide adoption to real applications. Firstly, a relatively large number of microphones are needed to meet the constraints of forming a sharp beam. Secondly, we must carefully tune the step-size parameter of the gradient-based algorithm to make the system work under different real use cases.

To address these problems, we propose a novel geometrically constrained IVA (GCIVA) method that combines linear constraints that restrict far-field responses of demixing filters with IVA. We derive a convergence-guaranteed optimization algorithm based on the auxiliary function approach, and vectorwise coordinate descent (VCD) [14], which we call "GCAV (geometrically constrained auxiliary function with VCD)-IVA", to preserve the advantages from the auxiliary function approach of fast convergence and no step-size tuning. Although the proposed GCAV-IVA is an extension of a determined method that improves the source separation performance for determined situations, since the geometric constraints can be well-designed as a BM [117], which works as a noise estimator, GCAV-IVA can be easily extended to handle underdetermined situations by applying noise suppression as done in the GSC [118]. Moreover, thanks to the constraints, GCAV-IVA works well even though diffuse noise exists, which relaxes the strict restriction of the determined case. From this point of view, we consider the proposed GCAV-IVA as an underdetermined method. We also extend the proposed GCAV-IVA to an online algorithm for those applications where real-time processing is necessary.

4.2 Geometrically constrained IVA using auxiliary function approach

4.2.1 Problem formulation

IVA assumes that sources follow a multivariate distribution and thus dependencies over frequency components can be exploited to avoid the permutation prob-

lem. The demixing matrices $\mathcal{W}$ are estimated by minimizing the following objective function

$$\mathcal{L}_{\mathrm{IVA}}(\mathcal{W}) = \sum_{j} \mathbb{E}[\mathcal{G}(\boldsymbol{y}_j)] - \sum_{f} \log|\det \mathbf{W}(f)|. \tag{4.1}$$

Here, $\boldsymbol{y}_j$ is the source-wise vector representation and $\mathcal{G}(\boldsymbol{y}_j)$ is the contrast function. Now, let us consider a geometric constraint [11] that restricts the far-field response of the jth demixing filter estimated by IVA at the direction ϑ, which is described as

$$\mathcal{L}_{\mathrm{gc}}(\mathcal{W}) = \sum_{j} \lambda_j \sum_{f} |\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{d}_j(f,\vartheta) - q_j|^2. \tag{4.2}$$

Here, $\mathbf{d}_j(f,\vartheta)$ is the steering vector pointing to the direction ϑ, q_j is the nonnegative-valued constant for all frequency bins, and $\lambda_j \geq 0$ is a parameter weighing the importance of the constraint. This concept is used in the linearly constrained minimum variance (LCMV) beamformer [119]. Note that (4.2) with $q_j = 1$ forces the spatial filter to form a conventional delay-and-sum beamformer steering at the direction ϑ to preserve the target source while a small value of q_j essentially creates a spatial null towards the target direction ϑ aiming at suppressing the target source and preserving all other sources. The null constraint on the target direction can serve as a BM [117], so that the corresponding channel can produce good estimate of interference and noise. Such estimate would have potential benefit of better handling under/overdetermined cases compared to traditional BSS methods. The objective function of the proposed GCIVA is summarized as

$$\mathcal{L}_{\mathrm{GCIVA}}(\mathcal{W}) = \mathcal{L}_{\mathrm{IVA}}(\mathcal{W}) + \mathcal{L}_{\mathrm{gc}}(\mathcal{W}). \tag{4.3}$$

4.2.2 Inference algorithm based on auxiliary function approach

In this section, we derive an iterative algorithm for parameter estimation of (4.3) with the auxiliary function approach [25]. Since the geometric constraints are linear, we can simply obtain the auxiliary function that upper-bounds (4.3) by com-

bining the original AuxIVA's auxiliary function (2.54) with these linear constraints:

$$\mathcal{L}^{+}_{\mathrm{GCIVA}}(\mathcal{W},\mathcal{Q}) \overset{c}{=} \sum_{j}\sum_{f}\left\{\frac{1}{2}\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{Q}_j(f)\mathbf{w}_j(f) - \log|\det\mathbf{W}(f)|\right\} + \mathcal{L}_{\mathrm{gc}}(\mathcal{W}), \quad (4.4)$$

where $\mathbf{Q}_j(f)$ is the weighted covariances expressed as

$$\mathbf{Q}_j(f) = \mathbb{E}\Big[\frac{G_R'(r_j)}{r_j}\mathbf{x}(f)\mathbf{x}^{\mathsf{H}}(f)\Big]. \quad (4.5)$$

The update rule for $\mathcal{Q} = \{\mathbf{Q}_j(f)\}_{j,f}$ is obtained straightforwardly by applying (4.5). Here, we consider the source model with

$$G_R(r_j) = r_j. \quad (4.6)$$

Then, we focus on deriving the update rule for $\mathcal{W}$. The indices of f and ϑ are omitted hereafter for the notation simplicity. Due to the linear constraint terms, the equation $\partial\mathcal{L}^{+}_{\mathrm{GCIVA}}(\mathcal{W},\mathcal{Q})/\partial\mathbf{w}_j^{*} = 0$ cannot be solved as the HEAD problem anymore. To obtain the optimal $\mathbf{w}_j$ of (4.4) with fixed $\mathcal{Q}$, inspired by the VCD method [14], we embrace the idea of arranging the term $\log|\det\mathbf{W}|$ by using the property of cofactor expansion

$$\boldsymbol{B} = [\boldsymbol{b}_1,\ldots,\boldsymbol{b}_J] \overset{\mathrm{def}}{=} (\det\mathbf{W})\mathbf{W}^{-1}, \quad (4.7)$$

where $\boldsymbol{b}_j$ is the jth column of the adjugate matrix of $\mathbf{W}$ defined as

$$\boldsymbol{B}_{pq} = (-1)^{p+q}\tilde{\mathbf{W}}_{qp}. \quad (4.8)$$

Here, the index pq denotes the (p,q) entry of $\boldsymbol{B}$ and $\tilde{\mathbf{W}}_{qp}$ is the (q,p) minor determinant of $\mathbf{W}$. We can then obtain

$$\det\mathbf{W} = \mathbf{w}_j^{\mathsf{H}}\boldsymbol{b}_j. \quad (4.9)$$

The partial derivative of (4.4) with respect to $\mathbf{w}_j^*$ is calculated as

$$\frac{\partial \mathcal{L}^+_{\mathrm{GCIVA}}(\mathcal{W}, \mathcal{Q})}{\partial \mathbf{w}_j^*} = \mathbf{D}_j \mathbf{w}_j - \frac{\boldsymbol{b}_j}{\mathbf{w}_j^{\mathsf{H}} \boldsymbol{b}_j} - \lambda_j q_j \mathbf{d}_j, \tag{4.10}$$

where

$$\mathbf{D}_j = \mathbf{Q}_j + \lambda_j \mathbf{d}_j \mathbf{d}_j^{\mathsf{H}}. \tag{4.11}$$

From $\partial \mathcal{L}^+_{\mathrm{GCIVA}} / \partial \mathbf{w}_j^* = 0$, we have

$$\mathbf{w}_j = \mathbf{D}_j^{-1}(\nu_j \boldsymbol{b}_j + \lambda_j q_j \mathbf{d}_j), \tag{4.12}$$

where

$$\nu_j = \frac{1}{\mathbf{w}_j^{\mathsf{H}} \boldsymbol{b}_j}. \tag{4.13}$$

From (4.13), we obtain

$$\nu_j \mathbf{w}_j^{\mathsf{H}} \boldsymbol{b}_j - 1 = 0. \tag{4.14}$$

By substituting (4.12) into (4.14), we obtain

$$\boldsymbol{b}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \boldsymbol{b}_j |\nu_j|^2 + \lambda_j q_j \mathbf{d}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \boldsymbol{b}_j \nu_j - 1 = 0. \tag{4.15}$$

Because the first and third terms in (4.15) are real numbers, the imaginary part of the second term must be 0 as

$$\Im[\lambda_j q_j \mathbf{d}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \boldsymbol{b}_j \nu_j] = 0. \tag{4.16}$$

Since $\nu_j \neq 0$, we can obtain

$$\nu_j = \gamma_j (\lambda_j q_j \mathbf{d}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \boldsymbol{b}_j)^* = \gamma_j \lambda_j q_j \boldsymbol{b}^{\mathsf{H}} \mathbf{D}_j^{-1} \mathbf{d}_j \tag{4.17}$$

or

$$\lambda_j q_j \mathbf{d}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \boldsymbol{b}_j = 0, \tag{4.18}$$

where $\gamma_j \in \mathbb{R}\backslash\{0\}$ is a scale parameter. If (4.17) holds, we obtain a quadratic equation with respect to γ_j from (4.15) as

$$\lambda_j^2 q_j^2 \boldsymbol{b}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \boldsymbol{b}_j |\boldsymbol{b}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \mathbf{d}_j|^2 \gamma_j^2 + \lambda_j^2 q_j^2 |\boldsymbol{b}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \mathbf{d}_j|^2 \gamma_j - 1 = 0. \tag{4.19}$$

By substituting the solution of γ_j of (4.19) into (4.17), we have the solution of ν_j as

$$\nu_j = \frac{\lambda_j q_j \boldsymbol{b}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \mathbf{d}_j}{2 \boldsymbol{b}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \boldsymbol{b}_j} \left(-1 \pm \sqrt{1 + \frac{4 \boldsymbol{b}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \boldsymbol{b}_j}{\lambda_j^2 q_j^2 |\boldsymbol{b}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \mathbf{d}_j|^2}} \right). \tag{4.20}$$

Here, we should take the positive solution based on the appendix in [14]. If (4.18) holds, the solution of (4.15) becomes

$$\nu_j = \frac{e^{i\phi_j}}{\sqrt{\boldsymbol{b}_j^{\mathsf{H}} \mathbf{D}_j^{-1} \boldsymbol{b}_j}}, \tag{4.21}$$

where i denotes the imaginary unit and $\phi_j \in (-\pi, \pi]$ denotes an arbitrary phase. Note that ϕ_j does not change the value of $\mathcal{L}_{\mathrm{GCIVA}}^{+}$. Therefore, we set ϕ_j at

$$\phi_j = \angle \frac{(\det \mathbf{W}_j)^*}{|\det \mathbf{W}_j|}. \tag{4.22}$$

From (4.12), (4.20), (4.21), and the relationship $\boldsymbol{b}_j = (\det \mathbf{W}_j) \mathbf{W}_j^{-1} \boldsymbol{e}_j$, the update rules of $\mathbf{w}_j$ are obtain as

$$\mathbf{u}_j = \mathbf{D}_j^{-1} \mathbf{W}^{-1} \boldsymbol{e}_j, \tag{4.23}$$

$$\hat{\mathbf{u}}_j = \lambda_j q_j \boldsymbol{D}_j^{-1} \mathbf{d}_j, \tag{4.24}$$

$$h_j = \mathbf{u}_j^{\mathsf{H}} \mathbf{D}_j \mathbf{u}_j, \tag{4.25}$$

$$\hat{h}_j = \mathbf{u}_j^{\mathsf{H}} \mathbf{D}_j \hat{\mathbf{u}}_j, \tag{4.26}$$

Algorithm 3 Offline GCAV-IVA Algorithm

Require: Observed mixture signal $\mathbf{x}(f,n)$, iteration number $\mathscr{L}$
  Initialize $\mathbf{W}(f)$ with identity marix.
  for $\ell = 1$ to $\mathscr{L}$ **do**
    for each source j of J **do**
      $y_j(f,n) = \mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)$
      for $f = 1$ to F **do**
        (updating auxiliary variables)
        update $\mathbf{Q}_j(f)$ using (4.5)
        (updating demixing matrices)
        calculate $\mathbf{D}_j(f)$ using (4.11)
        update $\mathbf{w}_j(f)$ using the IP method $(4.23) - -(4.27)$
      end for
    end for
  end for

$$\mathbf{w}_j = \begin{cases} \frac{1}{\sqrt{h_j}}\mathbf{u}_j + \hat{\mathbf{u}}_j \ \ (\text{if } \hat{h}_j = 0), \\ \frac{\hat{h}_j}{2h_j}\Big[-1 + \sqrt{1 + \frac{4h_j}{|\hat{h}_j|^2}}\Big]\mathbf{u}_j + \hat{\mathbf{u}}_j \ \ (\text{otherwise}). \end{cases} \tag{4.27}$$

Here, $\boldsymbol{e}_j$ is the jth column of the $J \times J$ identity matrix. Therefore, the parameter optimization algorithm of GCAV-IVA is summarized in *Algorithm 3*, which consists of updating the auxiliary variable $\mathbf{Q}_j(f)$ with (4.5), calculating $\mathbf{D}_j(f)$ with (4.11), and updating $\mathbf{W}(f)$ with (4.23)–(4.27). We can simply confirm that these update rules are equivalent to those employed in AuxIVA when $\lambda_j = 0$ for all j. Therefore, GCAV-IVA can be interpreted as an geometrically constrained extension of AuxIVA. It is noteworthy that the algorithm takes benefits of the auxiliary function approach, namely, no step-size tuning, fast convergence, and is guaranteed to decrease monotonically. Moreover, the algorithm having similar updating procedures with AuxIVA allows us to adopt autoregressive estimation [111] to develop online systems, which is indispensable in real-time and low-latency applications.

4.3 An extension for online applications

In the GCAV-IVA described above, which is an offline algorithm, only (4.5) requires all the observed samples over time $n = 1, \ldots, N$,

$$\mathbf{Q}_j(f) = \frac{1}{N} \sum_n \Big[\frac{G_R'(r_j(n))}{r_j(n)} \mathbf{x}(f,n)\mathbf{x}^{\mathsf{H}}(f,n) \Big]. \tag{4.28}$$

Hence, this equation is the point of formulation for the online algorithm. One simple way is to calculate $\mathbf{Q}_j(f)$ in a blockwise manner by using

$$\mathbf{Q}_j(f,n) = \frac{1}{L} \sum_{\tau=n-L+1}^{n} \Big[\frac{G_R'(r_j(\tau))}{r_j(\tau)} \mathbf{x}(f,\tau)\mathbf{x}^{\mathsf{H}}(f,\tau) \Big], \tag{4.29}$$

where $\mathbf{Q}_j(f,n)$ denotes the calculated $\mathbf{Q}_j(f)$ at frame n, L denotes the block size, and $r_j(\tau)$ is calculated by replacing $\mathbf{w}_j(f)$ with $\mathbf{w}_j(f,\tau)$ in (2.56)

$$r_j(\tau) = ||\boldsymbol{y}_j(\tau)||_2 = \sqrt{\sum_f |\mathbf{w}_j^{\mathsf{H}}(f,\tau)\mathbf{x}(f,\tau)|}. \tag{4.30}$$

If we directly employ (4.29) to obtain sufficient statistics $\mathbf{Q}_j(f,n)$, the past observation with relatively large L needs to be retained and the summation must be calculated at every new frame arrives, which is highly cost-consuming. On the other hand, if we set a small value to L for reducing the complexity, the insufficient statistics may lead to severe performance degradation.

To reduce computational cost and properly compute the statistics, we propose applying autoregressive calculation of $\mathbf{Q}_j(f,n)$ as done in the online AuxIVA [111] that uses the previously calculated $\mathbf{Q}_j(f,n-L)$ as follows:

$$\mathbf{Q}_j(f,n) = \varkappa \mathbf{Q}_j(f,n-L) + (1-\varkappa)\frac{1}{L} \sum_{\tau=n-L+1}^{n} \Big[\frac{G_R'(r_j(\tau))}{r_j(\tau)} \boldsymbol{x}(f,\tau)\boldsymbol{x}^{\mathsf{H}}(f,\tau) \Big]. \tag{4.31}$$

Here, $0 \leq \varkappa < 1$ is a forgetting factor, which controls how much statistics of past signals is considered. Sufficient statistics can then be computed with a small value of L. Note that (4.31) reduces to (4.29) when $\varkappa = 0$. Since a longer interval of past samples is considered through the recursion, it is expected that this approximation

Algorithm 4 Online GCAV-IVA Algorithm

Require: Observed mixture signal $\mathbf{x}(f,n)$, iteration number $\mathscr{L}$, forgetting factor $\varkappa$, block size L.
Initialize $\mathbf{W}(f,n)$ with identity marix.
Initialize $\mathbf{Q}_j(f,0)$.
for $n=1$ to N **do**
  for $\ell=1$ to $\mathscr{L}$ **do**
    for each source j of J **do**
      $y_j(f,n)=\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{x}(f,n)$
      for $f=1$ to F **do**
        (updating auxiliary variables)
        update $\mathbf{Q}_j(f,n)$ using (4.31)
        (updating demixing matrices)
        calculate $\mathbf{D}_j(f,n)$ using (4.11)
        update $\mathbf{w}_j(f,n)$ using the IP method $(4.23)--(4.27)$
      end for
    end for
  end for
end for

can improve separation performance in the fixed source situation with a large $\varkappa$. In contrast, separation performance in moving source situation is expected to improve with a small $\varkappa$, where any change in source positions can be reflected quickly via the blockwise term. The proposed online algorithm, called "online GCAV-IVA" (oGCAV-IVA), is a natural extension of the offline algorithm and the implementation can be very simple. However, note that the theoretical correctness of the approximation has not been guaranteed. *Algorithm 4* summarizes the algorithm.

4.4 Experimental evaluations

To evaluate the effectiveness of the proposed method, we conducted several speech enhancement experiments with a dual-microphone system.

4.4.1 Systems for a dual-microphone case

To develop a dual-mirophone system, we take the following conditions into consideration:

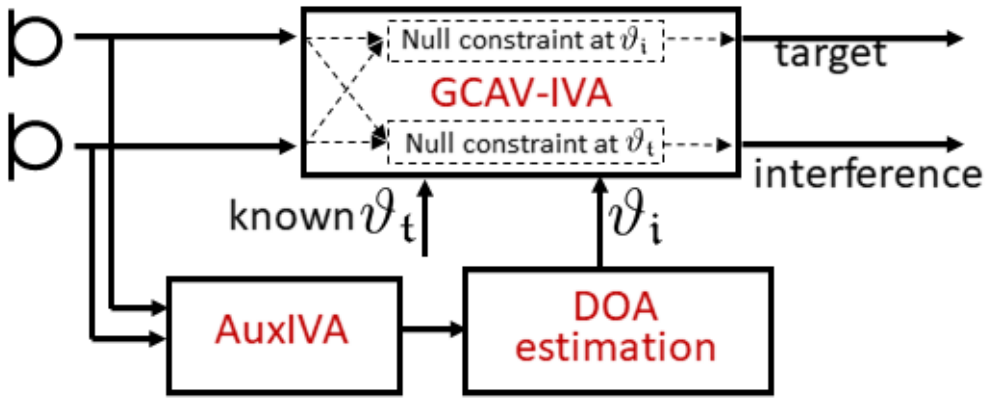

Figure 4.1: A dual-microphone system.

- The correct DOA of the target speaker ϑ_t is known;
- Null constraints are employed, i.e., $q_j = 0$ or close to zero. It is a practical choice since only two microphones are available.

Fig. 4.1 shows an overview of the dual-microphone system. Under the conditions above, we always apply a null constraint to the interference channel, where the null is formed toward the target speaker direction. For the target channel, we evaluate three options in the next section.

1. No constraint.
2. Null constraint at the interference direction from the oracle in 2-speaker case or at a dummy interference direction in 1-speaker case. This option is only for reference purpose.
3. Null constraint at the interference direction estimated by a separate AuxIVA system.

The motivation of third option is that, as demonstrated later, we find that the constraining both channels can lead to a higher enhancement performance. In this option, the interference DOA is obtained from a separate AuxIVA system. Since a BSS system can be interpreted as a set of adaptive null-beamformers [120], the directional nulls, which can be identified from the directivity patterns, usually point

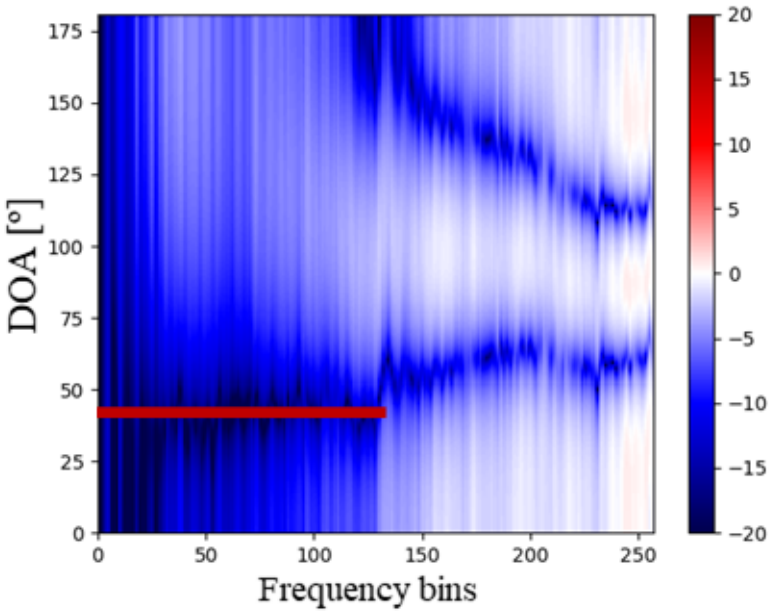

Figure 4.2: Example of directivity pattern of demixing filter estimated with AuxIVA, where a null steering to about 40° exists. This filter suppresses the signal coming from about 40°.

out the directions where the sources come from [12, 121, 122]. Fig. 4.2 shows an example of the directivity pattern of the demixing filter achieved by AuxIVA. We can see that there exists a null steering to about 40°, which suppresses the signal coming from that direction. Hence, we can consider that there exists a signal at about 40°.

In the system, the DOA of the jth output sources is given as

$$\hat{\vartheta}_j = \underset{\vartheta}{\operatorname{argmin}} \sum_{f}^{F/2} |\mathbf{w}_j^{\mathsf{H}}(f)\mathbf{d}(f,\vartheta)|. \tag{4.32}$$

The interference DOA $\hat{\vartheta}_{\mathrm{i}}$ can then be obtained by selecting the one far away from the target DOA ϑ_{t}:

$$\hat{\vartheta}_{\mathrm{i}} = \underset{\hat{\vartheta}_j}{\operatorname{argmax}} \left[|\hat{\vartheta}_j - \vartheta_{\mathrm{t}}|\right],\ j = 1, 2 \tag{4.33}$$

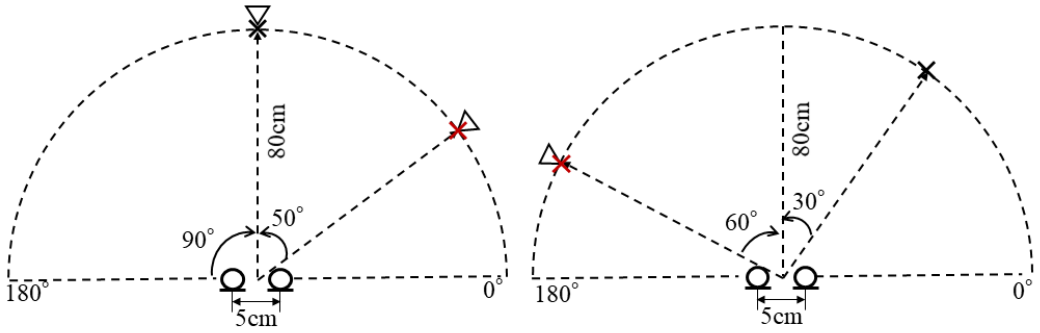

Figure 4.3: Configurations of sources and microphones, where "×" and "△" denote source positions used for 2-speaker and 1-speaker case, respectively. Red "×" denotes the target.

4.4.2 Dataset and settings for offline speech enhancement

We used speech samples of 4 speakers (2 females and 2 males) excerpted from VCC2018 database [98], which included 81 sentences for each speaker. The audio files were about 3-7 seconds long. The mixture signals were created by simulating two-channel recordings of two sources where the RIRs were synthesized using the image method [99]. Fig. 4.3 shows the positions of the sources and microphones. The interval of microphones was set at 5 cm. 2 DOA settings were investigated in the 2-speaker case, and 3 settings were investigated in the 1-speaker case. We tested two different reverberant conditions where RT_{60} was about 200 ms and 470 ms, which were controlled by setting the reflection coefficient of the walls at 0.4 and 0.8. To simulate the more realistic acoustic environment, 4 types of diffuse noise excerpted from DEMAND database [123], including park, office, cafeteria, and metro, were added to reverberant speech signals. We generated 1920 and 960 test samples for the 2-speaker and 1-speaker cases with various target-to-interference energy ratios and speech-to-noise energy ratios. The SNRs of the test samples in the 2-speaker case and 1-speaker case were between [-2, 6] dB and [0, 6] dB, respectively.

All the speech signals were sampled at 16 kHz. The STFT was computed using a Hanning window whose length was set at 32 ms, and the window shift

Table 4.1: Summary of tested GCAV-IVA systems.

System #	ϑ_{i}	q_1	q_2	λ_1	λ_2
(1)	No constraint	—			
(2)	Known	0	0	2	10
(3)		0.5	0.2		
(4)	Estimated by AuxIVA	0	0		
(5)		0.5	0.2		

was 16 ms. We compared the minimum power distortionless response (MPDR) beamformer [124] calculated with the far-field steering vectors, the AuxIVA using $G_R(r_j(n)) = r_j(n)$, and the GCAV-IVA method with various constraints. The specific settings of the tested systems are summarized in Table 4.1. For MPDR and GCAV-IVA, we evaluated the output from the target channel, whereas for AuxIVA, we evaluated outputs from all the channels and took the best score as the result.

4.4.3 Offline speech enhancement

First we investigated the potential of the standard AuxIVA as a DOA estimator. The AuxIVA had 3 update iterations and the DOA range was set at $[0°, 180°]$ with an interval of $5°$. Fig. 4.4 shows the estimation results in a histogram format, which were calculated from the 2-speaker dataset. It is revealed that more than 60% of the estimated directions is located in the range of $\pm 20°$ against the true DOA. In the next subsection, we will demonstrate the benefit of the DOA estimation in speech enhancement experiments.

Table 4.2 and Table 4.3 summarize the speech enhancement results. The proposed GCAV-IVA method exceeded the conventional MPDR in terms of all criteria and achieved higher scores than AuxIVA in terms of SDRs and SIRs, which confirmed the advantage of the geometric constraints. Comparing the results achieved by system (1) with other systems, we found that constraining two channels led to higher enhancement performances, even in the situation where any interference speaker doesn't exist, i.e., 1-speaker case. The results also indicate that carefully tuned q_j was able to produce slightly higher SDR and SIR scores. Interestingly, the system exploiting interference DOA estimation outperformed the one using true

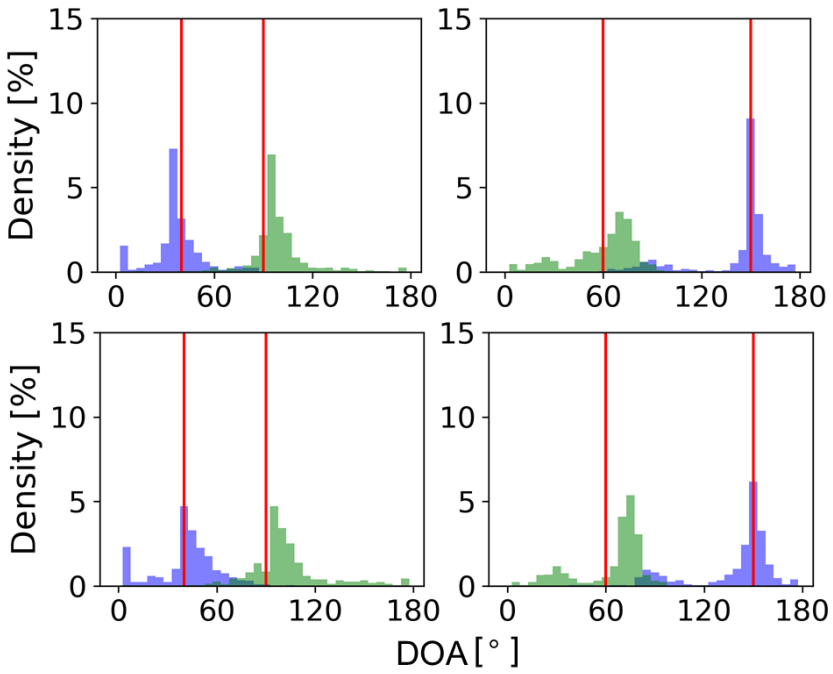

Figure 4.4: DOA estimation results achieved by performing AuxIVA update for 3 times under reverberant conditions where $RT_{60} = 200$ ms (upper) and $RT_{60} = 470$ ms (bottom). Red lines show true DOAs. Blue and green graphs are estimated DOA histograms for two directions.

Table 4.2: SDR, SIR, and SAR of 2-speaker case.

Method	$RT_{60} = 200$ ms			$RT_{60} = 470$ ms		
	SDR [dB]	SIR [dB]	SAR [dB]	SDR [dB]	SIR [dB]	SAR [dB]
unproc	1.46	1.61	23.02	0.78	1.47	12.11
MPDR	3.82	4.89	12.30	3.55	5.33	9.95
AuxIVA	7.12	8.98	**14.05**	4.96	7.42	**10.51**
GCAV-IVA(1)	8.42	11.19	13.33	6.47	10.33	9.86
GCAV-IVA(2)	8.71	11.50	13.53	6.51	10.34	9.89
GCAV-IVA(3)	8.75	11.62	13.49	6.55	**10.50**	9.84
GCAV-IVA(4)	8.72	11.52	13.52	6.53	10.36	9.93
GCAV-IVA(5)	**8.80**	**11.69**	13.51	**6.57**	**10.50**	9.88

DOAs. One possible reason is that, since the DOA estimate coming from the AuxIVA points out the direction including the most statistically independent components, suppressing that direction can result in a higher SIR.

Table 4.3: SDR, SIR, and SAR of 1-speaker case.

Method	$RT_{60} = 200$ ms			$RT_{60} = 470$ ms		
	SDR [dB]	SIR [dB]	SAR [dB]	SDR [dB]	SIR [dB]	SAR [dB]
unproc	3.03	3.37	21.61	2.14	3.06	12.48
MPDR	1.29	2.79	9.50	2.14	4.03	8.98
AuxIVA	6.04	8.00	**13.12**	4.07	6.65	**10.04**
GCAV-IVA(1)	7.00	10.20	11.73	5.47	10.20	8.76
GCAV-IVA(2)	7.37	10.33	12.23	5.60	10.30	8.90
GCAV-IVA(3)	7.32	10.40	12.20	5.55	10.36	8.75
GCAV-IVA(4)	7.39	10.27	12.37	5.71	10.41	9.03
GCAV-IVA(5)	**7.43**	**10.41**	12.31	**5.73**	**10.56**	8.93

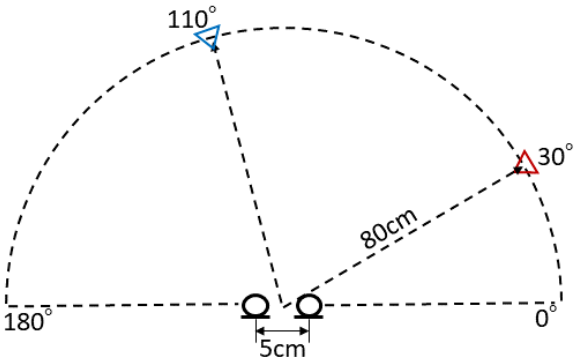

Figure 4.5: Configurations of microphones and a pair of fixed sources, where red and blue marks denote target and interference positions, respectively

4.4.4 Dataset and settings for online speech enhancement

To evaluate the effectiveness of the proposed online GCAV-IVA method in the dual-microphone system, we conducted speech enhancement experiments in two situations: 2 spatially fixed sources and 1 fixed target source with 1 moving interference source.

We used speech samples of 4 speakers (2 females and 2 males) excerpted from VCC2018 database [98]. Clean signals for the simulation were generated by concatenating utterances spoken by a single speaker in random order, whose length was about 30 seconds long. For 2 spatially fixed sources, the mixture sig-

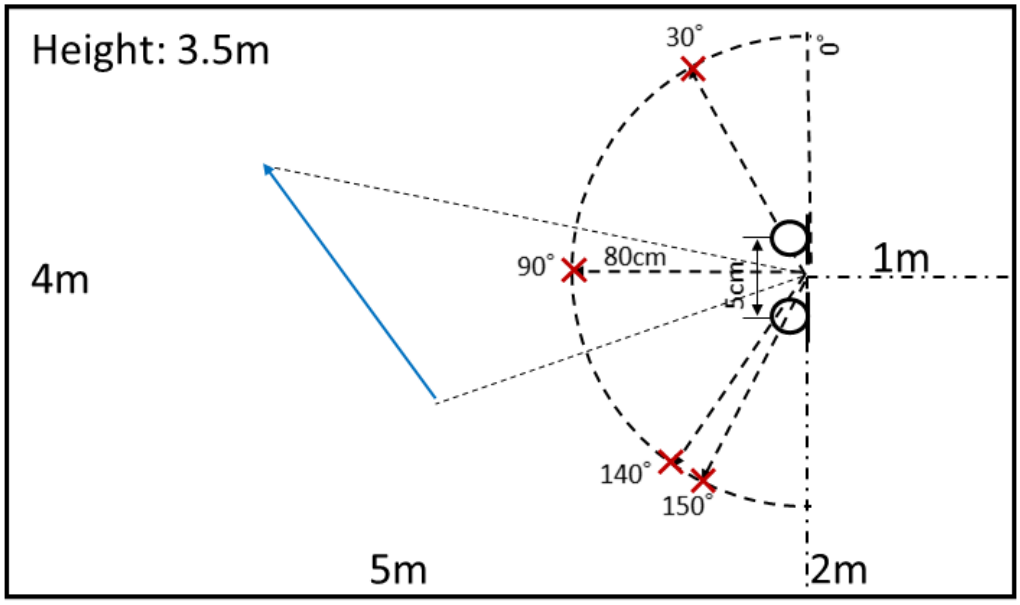

Figure 4.6: Configurations of sources and microphones. Red mark and blue line denote fixed target source and the trace of moving interference, respectively.

Table 4.4: Summary of tested online GCAV-IVA systems.

System #	ϑ_{i}	q_1	q_2	λ_1	λ_2
(a)	No constraint	—			
(b)	Known	0.5	0.2	1	1
(c)	Estimated by AuxIVA	0.5	0.2	1	1

nals were created by simulating two-channel recordings of two sources where RIRs were synthesized using the image method [99]. Fig. 4.5 shows the positions of microphones and a pair of sources. The interval of microphones was set at 5 cm. We tested 5 pairs of DOA settings involving (30°, 110°), (70°, 100°), (150°, 60°), (40°, 90°), (90°, 150°), where the former and latter angles are target and interference positions, respectively. For the spatially nonstationary situation, we first generated reverberant signals of moving interference sources using "signal generator" [1]. Then we mixed the generated signals with the reverberant target signals. 4 positions of the target signal were tested, namely, 30°, 90°, 140°, and 150°. More configuration details are available in Fig. 4.6. We tested two different reverberant

[1] https://www.audiolabs-erlangen.de/fau/professor/habets/software/signal-generator

conditions. To meet the instantaneous mixing model assumption, RT_{60} were set at 78 ms and 200 ms, which were controlled by setting the reflection coefficient of the walls at 0.2 and 0.4, respectively. To simulate the realistic background noise, 4 types of diffuse noise excerpted from DEMAND database [123], including park, office, cafeteria, and metro, were also added to reverberant speech signals to generate "noisy" datasets. We refer to the dataset without/with diffuse noise as "S+I" and "S+I+N", respectively. The energy ratio of target-to-interference was set at 0 dB and the input SDR of noisy speech was about [-3, 0] dB.

All the speech signals were sampled at 16 kHz. The STFT was computed using a Hanning window whose length was set at 32 ms, and the window shift was 16 ms. We compared the proposed online GCAV-IVA (oGCAV-IVA) method using $L = 1$ with online AuxIVA (oAuxIVA) that also adopts (4.31) with $L = 1$. We run these two algorithms for 5 iterations with the first 5 frames to initialize demixing matrices. To update demixing matrices every frame, we run the algorithms for 2 iterations. The forgetting factor $\varkappa$ was set at 0.96 for both oAuxIVA and oGCAV-IVA. Similarly, we considered three options for the target channel, where we refer them as to system (a), (b), and (c). Table 4.4 shows the experimantal settings of q_j and λ_j for each system. λ_j was set at 1 for both channels or only the interference channel in the system (a). We set q_j at 0.5 for the target channel and 0.2 for the interference channel. For DOA estimation, the range was set at [0°, 180°] with an interval of 5°. For each concatenated utterance, we evaluated signals every second, then computed the average scores over 30 seconds as the results. For GCAV-IVA, we evaluated the output from the target channel, whereas for AuxIVA, we evaluated outputs from all the channels and took the best score as a result.

4.4.5 Online speech enhancement

Table 4.5 shows speech enhancement results. The proposed algorithm significantly outperformed oAuxIVA without regard to diffuse noise. Comparing with the GCAV-IVA system using true DOA of the interference, system (c) that adopts DOA estimation achieved a further improvement of more than 4 dB, which was impressive. One possible reason is that, since the DOA estimate coming from the sep-

Table 4.5: SDR, SIR, SAR scores obtained in spatially stationary condition.

Method	S+I			S+I+N		
	SDR [dB]	SIR [dB]	SAR [dB]	SDR [dB]	SIR [dB]	SAR [dB]
oAuxIVA	8.37	12.57	12.06	1.70	4.06	8.81
oGCAV-IVA (a)	11.77	15.72	14.51	6.07	8.48	12.06
oGCAV-IVA (b)	10.03	12.50	14.96	4.29	5.81	12.86
oGCAV-IVA (c)	**14.19**	**18.40**	**16.73**	**6.86**	**9.18**	**13.60**

Table 4.6: SDR, SIR, SAR scores obtained in spatially non-stationary condition.

Method	S+I			S+I+N		
	SDR [dB]	SIR [dB]	SAR [dB]	SDR [dB]	SIR [dB]	SAR [dB]
oAuxIVA	3.77	6.51	9.34	0.12	1.96	8.13
oGCAV-IVA (a)	**6.83**	**9.21**	11.66	**3.51**	**5.33**	10.50
oGCAV-IVA (c)	5.36	6.90	**12.33**	3.05	4.42	**11.41**

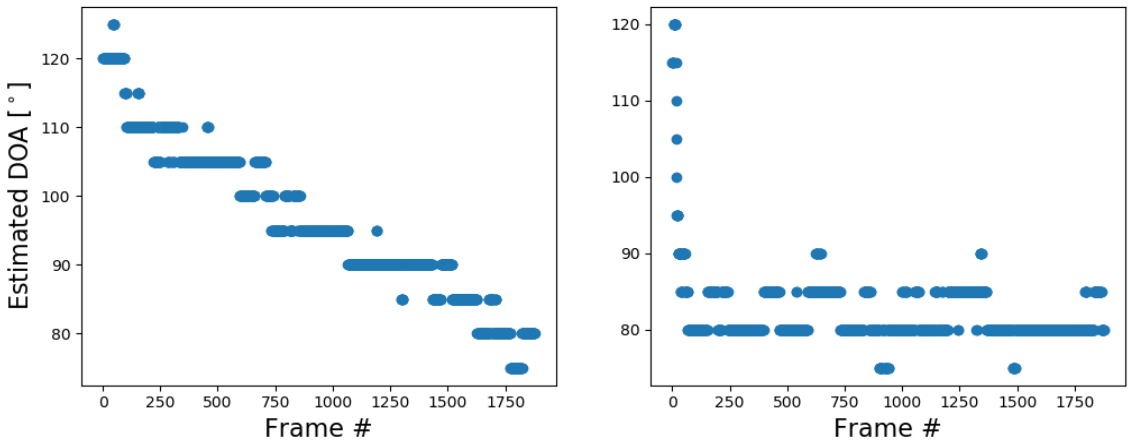

Figure 4.7: Examples of estimated DOA for moving source.

arate AuxIVA points out the direction involving the most statistically independent components, suppressing that direction can result in a higher SIR. Moreover, we found the proposed method was also able to improve the performance in the “noisy” situation, where the determined condition did not hold. oAuxIVA almost failed to enhance the speech with only achieving SDR score of 1.7 dB, whereas the proposed method exploiting geometric information still achieved SDR score of about 6.8 dB.

Table 4.6 shows the results of enhancing signals against moving sources. As

with the fixed source case, the proposed method outperformed oAuxIVA, where oGCAV-IVA achieved more than 1.5 dB and 2.9 dB improvement in the situation without/with diffuse noise, respectively. These results confirmed the effectiveness of geometric constraints in improving speech enhancement performance. The system adopting no constraint outperformed the one using DOA estimation in terms of SDR and SIR, which was different from the fixed source case. One possible reason is the accuracy of DOA estimation.

The trace of the moving source was designed to move with a uniform speed from 120° to about 80°, which was controlled by setting the positions of the start and endpoint, as shown in Fig. 4.6. Fig. 4.7 shows examples of the estimated interference DOA. The left figure shows an example of successful interference DOA estimation by oAuxIVA, while an example of failure cases can be seen in the right figure. In situations where oAuxIVA fails to estimate the interference DOA, the inappropriate constraint may degrade the performance.

All the experiments were run using an Intel (R) Core i7-7800X CPU@3.5 GHz. The measured average computational time was less than 16 ms, which was the length of window shift, namely, about 5 ms for the system (a) and (b), and about 15 ms for the system (c). These results indicated that the proposed algorithm could work in a real-time manner.

4.5 Summary of chapter 4

In this chapter, we proposed a GCIVA method, which combines IVA with a set of linear constraints restricting the far-field response of the demixing filter. We derived a convergence-guaranteed algorithm with the auxiliary function approach, which is called GCAV-IVA. We further extended the offline method to an online version by using an autoregressive estimation. We investigated the proposed offline and online algorithms using a dual-microphone system, where the DOA of target was known and that of interference was estimated using a separated AuxIVA. The experimental results revealed that the offline algorithm outperformed the conventional MPDR beamformer and AuxIVA, and the online algorithm outperformed online AuxIVA in both spatially static and dynamic conditions. Furthermore, the

online algorithm could perform in real-time, which confirmed the computational complexity of the proposed method was acceptable for online applications.

Chapter 5

Single-channel source separation based on discriminative nonnegative matrix factorization

5.1 Introduction

In this chapter, we consider supervised single-channel source separation, which is the most achievable condition in realistic environments since only one microphone is needed. With an NMF-based approach to supervised source separation, NMF is first applied to train the basis spectra of each source using training examples and is applied to the spectrogram of a mixture signal using the pretrained basis spectra at test time. The source signals are then separated out using a Wiener filter. A typical way to train the basis spectra is to minimize a dissimilarity measurement between the observed spectrogram and the NMF model. However, obtaining the basis spectra in this way does not ensure that the separated signal will be optimal at test time due to the inconsistency between the objective functions for training and separation, namely Wiener filtering.

To address this inconsistency, a framework called DNMF has been recently proposed [19]. While many methods called "discriminative NMF" [20, 35, 125–128], have been proposed with the aim of enhancing the discriminative power of the basis spectra, in this work, we use this term in relation to the work done in [19].

Note that the term “discriminative” is used in association with the discriminative models for classification and regression. The central idea of DNMF is that the basis spectra are trained in such a way that the output of the Wiener filter becomes as close to the spectrogram of each of the training examples as possible so that the separated signals become optimal at test time. This approach differs from the conventional supervised NMF framework in that it uses the training examples of all the sources to train the basis spectra for each of the sources. This is important since it helps to enhance the discriminative power of the basis spectra. However, the training criterion for DNMF becomes analytically more complex than the typical divergence measurements used in the standard NMF framework, which causes difficulty as regards optimization of the basis spectra. In the original work of DNMF, a multiplicative update algorithm was proposed, where the multiplicative factor is obtained by dividing the negative parts by the positive parts of the partial derivative of the objective function. Although this way of obtaining the update rules is indeed convenient in that it is applicable as long as an objective function is differentiable, one drawback is that the algorithm is generally not guaranteed to converge to a stationary point, which may limit the unleashing of the full potential of DNMF.

To overcome this weakness, in this chapter, we propose an auxiliary function-based algorithm for DNMF. We briefly review the formulation of DNMF in Sec. 5.2. Then, we derive the proposed algorithm in Sec. 5.3. We show in Sec. 5.4 that using the present basis training algorithm instead of the conventional MU algorithm leads to a notable improvement in speech enhancement performance. Sec. 5.5 conclude this chapter.

5.2 DNMF with multiplicative update algorithm

5.2.1 Formulation of DNMF

If we assume using the Wiener filter to obtain source signals, the training and test objectives become inconsistent. Namely, the basis spectra are not necessarily trained in such a way that the separated signals at test time will be optimal. With the standard supervised NMF approach, at test time, the basis matrix $\tilde{\mathbf{B}}$ is used not

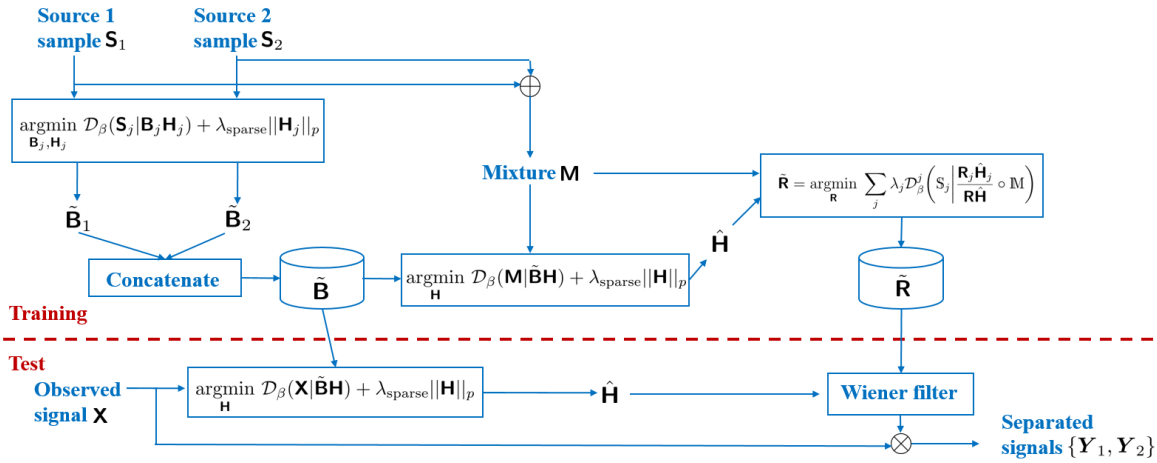

Figure 5.1: Flowchart of DNMF in 2 sources case.

only for estimating $\mathbf{H}$ from the mixture signal $\mathbf{X}$ but also for constructing the Wiener filter in (2.39). To make the training objective consistent with this test inference procedure, Weninger [19] proposed introducing two separate basis matrices for these different purposes, $\mathbf{B}$ and $\mathbf{R}$, and formulating a bilevel optimization problem

$$(\tilde{\mathbf{B}}_j,\tilde{\mathbf{H}}_j)=\underset{\mathbf{B}_j,\mathbf{H}_j}{\operatorname{argmin}}\ \mathcal{D}_\beta(\mathbf{S}_j|\mathbf{B}_j\mathbf{H}_j)+\lambda_{\text{sparse}}||\mathbf{H}_j||_p, \tag{5.1}$$

$$\hat{\mathbf{H}}=\underset{\mathbf{H}}{\operatorname{argmin}}\ \mathcal{D}_\beta(\mathbf{M}|\tilde{\mathbf{B}}\mathbf{H})+\lambda_{\text{sparse}}||\mathbf{H}||_p, \tag{5.2}$$

$$\tilde{\mathbf{R}}=\underset{\mathbf{R}}{\operatorname{argmin}}\ \sum_j \lambda_j\mathcal{D}_\beta^j\left(\mathbb{S}_j\middle|\frac{\mathbf{R}_j\hat{\mathbf{H}}_j}{\mathbf{R}\hat{\mathbf{H}}}\circ\mathbb{M}\right), \tag{5.3}$$

for training $\mathbf{B}$ and $\mathbf{R}$ so that $\mathbf{B}$ will be optimized for estimating $\mathbf{H}$ from $\mathbf{X}$ and $\mathbf{R}$ will be optimized for obtaining $\mathbb{Y}_1,\ldots,\mathbb{Y}_J$ based on the Wiener filtering. Here, $\lambda_j\geq 0$ is a constant that weighs the importance of source j. $\mathbf{M}=\{|m(f,n)|^2\}_{f,n}\in\mathbb{R}^{\geq 0,F\times N}$ denotes the power spectrogram of a mixture signal, which is simply constructed by mixing complex-valued spectrograms of multiple training samples $\boldsymbol{S}_1,\ldots,\boldsymbol{S}_J$. $\mathbb{M}$ and $\mathbb{S}_j$ denote the magnitude spectrograms $\sqrt{\mathbf{M}}$ and $\sqrt{\mathbf{S}_j}$, respectively. When our goal is to reconstruct a single source j only, we shall set λ_j at 1 and 0 for other sources $j'\neq j$. Fig. 5.1 illustrates the training and test processes of DNMF using 2 sources. Spectral templates in $\mathbf{B}$ are usually normalized to 1 as the standard NMF does to eliminate the scale arbitrary when estimating the activation matrix while

those in $\mathbf{R}$ do not need normalization. Note that the setting of β used in (5.2) and (5.3) are not necessary to be the same as long as those used in training and test time are the same. It has reported that best results were obtained by using KL divergence for (5.2) and EU distance for (5.3) [19]. This might be due to the KL divergence being better suited to decomposing mixtures.

The test inference algorithm for the DNMF approach consists of computing $\hat{\mathbf{H}}$ by solving (5.2) with the pretrained basis matrix $\tilde{\mathbf{B}}$ and observed mixture signal $\mathbf{X}$, constructing Wiener filter using $\tilde{\mathbf{R}}$ and $\hat{\mathbf{H}}$,

$$\hat{\mathbb{S}}_j = \frac{\tilde{\mathbf{R}}_j \hat{\mathbf{H}}_j}{\tilde{\mathbf{R}}\hat{\mathbf{H}}} \circ \mathbb{X}, \tag{5.4}$$

and performing iSTFT for each source j. Note that the test inference algorithm for the standard NMF approach corresponds to a special case where $\tilde{\mathbf{B}} = \tilde{\mathbf{R}}$.

5.2.2 MU algorithms for DNMF

It is obvious that the training criterion for DNMF is more analytically complex than the objective function of standard NMF. In [19], Weninger proposed a two-stage iterative algorithm for solving the above optimization problem: First, $\mathbf{B}$ and $\mathbf{H}$ are obtained by solving (5.1) and (5.2) using a standard NMF algorithm; Second, by using the obtained $\hat{\mathbf{H}}$, the basis matrix $\mathbf{R}$ is iteratively updated according to multiplicative update rules. Here, we set $\lambda_j = 1$ and $\lambda_{j':j' \neq j} = 0$ for speech enhancement tasks and define $\boldsymbol{\Upsilon} = \sum_j \mathbf{R}_j \hat{\mathbf{H}}_j$, $\boldsymbol{\Upsilon}_j = \mathbf{R}_j \hat{\mathbf{H}}_j$, $\boldsymbol{\Upsilon}_{\bar{j}} = \boldsymbol{\Upsilon} - \boldsymbol{\Upsilon}_j$, and $\hat{\mathbb{S}}_j = \frac{\boldsymbol{\Upsilon}_j}{\boldsymbol{\Upsilon}} \circ \mathbb{M}$.

For the KL divergence case, the objective function for each source j in (5.3) becomes

$$\mathcal{D}^j_{\mathsf{KL}}(\mathbb{S}_j | \hat{\mathbb{S}}_j) = \sum_{f,n} \mathtt{s}_j(f,n) \log \frac{\mathtt{s}_j(f,n)}{\mathtt{m}(f,n)\frac{\Upsilon_j(f,n)}{\Upsilon(f,n)}} + \mathtt{m}(f,n)\frac{\Upsilon_j(f,n)}{\Upsilon(f,n)} - \mathtt{s}_j(f,n). \tag{5.5}$$

Here, Gotham font denotes elements of magnitude spectrograms. Namely, $\mathtt{s}_j(f,n) = |s_j(f,n)|$ and $\mathtt{m}(f,n) = |m(f,n)|$. The partial derivative of (5.5) with respect to the

fth element of the kth basis function of the desired source, $r_{j,k}(f)$, is

$$\frac{\partial \mathcal{D}^j_{\mathsf{KL}}}{\partial r_{j,k}(f)} = \sum_n \mathtt{s}_j(f,n)\Big(\frac{\hat{\mathsf{h}}_{j,k}(n)}{\Upsilon(f,n)} - \frac{\hat{\mathsf{h}}_{j,k}(n)}{\Upsilon_j(f,n)}\Big) + \mathtt{m}(f,n)\frac{\hat{\mathsf{h}}_{j,k}(n)\Upsilon(f,n) - \Upsilon_j(f,n)\hat{\mathsf{h}}_{j,k}(n)}{\Upsilon^2(f,n)}$$
$$= \sum_n -\frac{\mathtt{s}_j(f,n)\Upsilon_{\bar{j}}(f,n)}{\Upsilon(f,n)\Upsilon_j(f,n)}\hat{\mathsf{h}}_{j,k}(n) + \frac{\mathtt{m}(f,n)\Upsilon_{\bar{j}}(f,n)}{\Upsilon^2(f,n)}\hat{\mathsf{h}}_{j,k}(n), \tag{5.6}$$

where the second quality is used by defining $\Upsilon_{\bar{j}}(f,n) = \Upsilon(f,n) - \Upsilon_j(f,n)$. Similarly, we obtain the partial derivative with respect to the $r_{j',k}(f)$ for any $j' \neq j$ as

$$\frac{\partial \mathcal{D}^j_{\mathsf{KL}}}{\partial r_{j',k}(f)} = \sum_n \frac{\mathtt{s}_j(f,n)}{\Upsilon(f,n)}\hat{\mathsf{h}}_{j',k}(n) - \frac{\mathtt{m}(f,n)\Upsilon_j(f,n)}{\Upsilon^2(f,n)}\hat{\mathsf{h}}_{j',k}(n). \tag{5.7}$$

Since all matrix elements are nonnegative, the multiplicative update rules can be derived by splitting (5.6) and (5.7) into positive and negative parts, as done in the standard NMF [23]:

$$\mathbf{R}_j \leftarrow \mathbf{R}_j \circ \frac{\frac{\mathbb{S}_j \circ \boldsymbol{\Upsilon}_{\bar{j}}}{\boldsymbol{\Upsilon}\circ\boldsymbol{\Upsilon}_j}\hat{\mathbf{H}}_j{}^{\mathsf{T}}}{\frac{\mathbb{M}\circ\boldsymbol{\Upsilon}_{\bar{j}}}{\boldsymbol{\Upsilon}^2}\hat{\mathbf{H}}_j{}^{\mathsf{T}}}, \tag{5.8}$$

$$\mathbf{R}_{\bar{j}} \leftarrow \mathbf{R}_{\bar{j}} \circ \frac{\frac{\mathbb{M}\circ\boldsymbol{\Upsilon}_j}{\boldsymbol{\Upsilon}^2}\hat{\mathbf{H}}_{\bar{j}}{}^{\mathsf{T}}}{\frac{\mathbb{S}_j}{\boldsymbol{\Upsilon}}\hat{\mathbf{H}}_{\bar{j}}{}^{\mathsf{T}}}, \tag{5.9}$$

where $\mathbf{R}_{\bar{j}} = [\mathbf{R}_1, \cdots, \mathbf{R}_{j-1}, \mathbf{R}_{j+1}, \cdots, \mathbf{R}_J]$, namely, the basis spectra of all sources excep j, and $\hat{\mathbf{H}}_{\bar{j}}$ is defined accordingly.

For the EU distance, the partial derivative of $\mathcal{D}^j_{\mathsf{EU}}$ leads to

$$\mathbf{R}_j \leftarrow \mathbf{R}_j \circ \frac{\frac{\mathbb{M}\circ\mathbb{S}_j\circ\boldsymbol{\Upsilon}_{\bar{j}}}{\boldsymbol{\Upsilon}^2}\hat{\mathbf{H}}_j{}^{\mathsf{T}}}{\frac{\mathbb{M}^2\circ\boldsymbol{\Upsilon}_j\circ\boldsymbol{\Upsilon}_{\bar{j}}}{\boldsymbol{\Upsilon}^3}\hat{\mathbf{H}}_j{}^{\mathsf{T}}} \tag{5.10}$$

$$\mathbf{R}_{\bar{j}} \leftarrow \mathbf{R}_{\bar{j}} \circ \frac{\frac{\mathbb{M}^2\circ\boldsymbol{\Upsilon}_j^2}{\boldsymbol{\Upsilon}^3}\hat{\mathbf{H}}_{\bar{j}}{}^{\mathsf{T}}}{\frac{\mathbb{M}\circ\mathbb{S}_j\circ\boldsymbol{\Upsilon}_j}{\boldsymbol{\Upsilon}^2}\hat{\mathbf{H}}_{\bar{j}}{}^{\mathsf{T}}} \tag{5.11}$$

The general case of $\lambda_j \geq 0$ for all j is a linear extension due to the linearity of the gredient. Although this way of obtaining update rules is convenient in that it is generally applicable as long as an objective function is differentiable, one downside is that the algorithm is not guaranteed to converge to a stationary point.

5.3 Auxiliary function approach for DNMF

To overcome the weakness of the conventional MU algorithm, we derive an algorithm for DNMF based on the auxiliary function approach, which is convergence-guaranteed. Here, we derive majorizers for the objective function where $\mathcal{D}_\beta$ is defined as the KL divergence and the IS divergence.

When using the KL divergence, the objective function in (5.3) is given by

$$\begin{aligned}\mathcal{F}_{\mathsf{KL}}(\mathbf{R}) &= \sum_j \lambda_j \mathcal{D}^j_{\mathsf{KL}}\Big(\mathbb{S}|\hat{\mathbb{S}}\Big) \\ &\overset{c}{=} \sum_j \lambda_j \sum_{f,n}\Big(-\mathtt{s}_{j,f,n}\log \Upsilon_{j,f,n} + \mathtt{s}_{j,f,n}\log \Upsilon_{f,n} + \frac{\Upsilon_{j,f,n}}{\Upsilon_{f,n}}\mathtt{m}_{f,n}\Big).\end{aligned} \tag{5.12}$$

Hereafter, we represent indices f and n as subscript for the notation simplicity. First, let us focus on the term $\Upsilon_{j,f,n}/\Upsilon_{f,n}$. To construct an upper bound for this term, we can use the following inequality:

Lemma 2. *For $a > 0$ and $b > 0$, we have*

$$\frac{a}{b} \le \frac{\zeta a^2}{2} + \frac{1}{2\zeta b^2}.$$

The equality holds if and only if

$$\zeta = \frac{1}{ab}.$$

Proof of Lemma 2. For $a, b, \zeta > 0$,

$$\begin{aligned}\zeta\left(a - \frac{1}{\zeta b}\right)^2 &= \zeta\left(a^2 - 2\frac{a}{\zeta b} + \frac{1}{\zeta^2 b^2}\right) \ge 0 \\ \Rightarrow \frac{a}{b} &\le \frac{\zeta a^2}{2} + \frac{1}{2\zeta b^2}.\end{aligned} \tag{5.13}$$

The equality holds if and only if $a - \frac{1}{\zeta b} = 0$. □

Since $\mathtt{m}$ is nonnegative, we can construct an upper bound for the third term of

(5.12) according to the above lemma,

$$\mathcal{F}_{\mathsf{KL}}(\mathbf{R}) \leq \sum_j \lambda_j \sum_{f,n} \Big(- \mathtt{s}_{j,f,n} \log \Upsilon_{j,f,n} + \mathtt{s}_{j,f,n} \log \Upsilon_{f,n} + \frac{\zeta_{j,f,n} \mathtt{m}_{f,n} \Upsilon_{j,f,n}^2}{2} + \frac{\mathtt{m}_{f,n}}{2\zeta_{j,f,n} \Upsilon_{f,n}^2} \Big). \tag{5.14}$$

The equality of (5.14) holds if and only if

$$\zeta_{j,f,n} = \frac{1}{\Upsilon_{j,f,n} \Upsilon_{f,n}}. \tag{5.15}$$

In the following, we construct upper bounds for each of the terms on the right-hand side of (5.14).

We notice that the function $-\log x$ is convex. Since $\mathtt{s}_{j,f,n}$ is positive, $-\mathtt{s}_{j,f,n} \log \Upsilon_{j,f,n}$ is convex in $\Upsilon_{j,f,n}$. Hence, we can use Jensen's inequality to obtain an upper bound for this term as

$$- \log \Upsilon_{j,f,n} \leq - \sum_k \gamma_{k,j,f,n} \log \frac{\mathsf{r}_{k,j,f} \hat{\mathsf{h}}_{k,j,n}}{\gamma_{k,j,f,n}}, \tag{5.16}$$

where $\gamma_{k,j,f,n}$ is a positive weight that sums to unity. The equality of (5.16) holds if and only if

$$\gamma_{k,j,f,n} = \frac{\mathsf{r}_{k,j,f} \hat{\mathsf{h}}_{k,j,n}}{\sum_{k'} \mathsf{r}_{k',j,f} \hat{\mathsf{h}}_{k',j,n}}. \tag{5.17}$$

The second term $\mathtt{s}_{j,f,n} \log \Upsilon_{f,n}$ is concave in $\Upsilon_{f,n}$. Hence, we can use the fact that a tangent line to the graph of a differentiable concave function liew entirely above the graph:

$$\log \Upsilon_{f,n} \leq \sum_k \frac{\mathsf{r}_{k,f} \hat{\mathsf{h}}_{k,n}}{\alpha_{f,n}} + \log \alpha_{f,n} - 1, \tag{5.18}$$

where $\alpha_{f,n}$ is an arbitrary positive number. The equality of this inequality holds if and only if

$$\alpha_{f,n} = \Upsilon_{f,n} = \sum_k \mathsf{r}_{k,f} \hat{\mathsf{h}}_{k,n}. \tag{5.19}$$

Since a quadratic function is convex, we can apply Jensen's inequality to the third term, which yields

$$\Upsilon_{j,f,n}^2 \leq \sum_k \frac{\mathsf{r}_{k,j,f}^2 \hat{\mathsf{h}}_{k,j,n}^2}{\xi_{k,j,f,n}}, \tag{5.20}$$

where $\xi_{k,j,f,n} > 0$ is also a positive number that satisfies $\sum_k \xi_{k,j,f,n} = 1$. The equality of (5.20) holds if and only if

$$\xi_{k,j,f,n} = \frac{\mathsf{r}_{k,j,f} \hat{\mathsf{h}}_{k,j,n}}{\sum_{k'} \mathsf{r}_{k',j,f} \hat{\mathsf{h}}_{k',j,n}}. \tag{5.21}$$

As regards the fourth term, we can use the fact that the function $1/x^2$ is convex in the first quadrant and then use Jensen's inequality to obtain an upper bound

$$\frac{1}{\Upsilon_{f,n}^2} \leq \sum_k \frac{\kappa_{k,f,n}^3}{\mathsf{r}_{k,j,f}^2 \hat{\mathsf{h}}_{k,j,n}^2}, \tag{5.22}$$

where $\kappa_{k,f,n}$ is a positive number that sums to unity. We can confirm that the equality of this inequality holds if and only if

$$\kappa_{k,f,n} = \frac{\mathsf{r}_{k,f} \hat{\mathsf{h}}_{k,n}}{\sum_{k'} \mathsf{r}_{k',f} \hat{\mathsf{h}}_{k',n}}. \tag{5.23}$$

From (5.16), (5.18), (5.20), and (5.22), we can construct a majorizer for the objective function with KL divergence as

$$\begin{aligned}
\mathcal{F}_{\mathsf{KL}}(\mathbf{R}) \leq \sum_j \lambda_j \sum_{k,f,n} \Bigg(& \frac{\mathtt{s}_{j,f,n} \mathsf{r}_{k,f} \hat{\mathsf{h}}_{k,n}}{\alpha_{f,n}} - \mathtt{s}_{j,f,n} \gamma_{k,j,f,n} \log \frac{\mathsf{r}_{k,j,f} \hat{\mathsf{h}}_{k,j,n}}{\gamma_{k,j,f,n}} \\
& + \frac{\zeta_{j,f,n} \mathtt{m}_{f,n}}{2\xi_{k,j,f,n}} \mathsf{r}_{k,j,f}^2 \hat{\mathsf{h}}_{k,j,n}^2 + \frac{\mathtt{m}_{f,n} \kappa_{k,f,n}^3}{2\zeta_{j,f,n} \mathsf{r}_{k,f}^2 \hat{\mathsf{h}}_{k,j,n}^2} \Bigg) + \text{const.} \\
=: \mathcal{F}_{\mathsf{KL}}^{+}(\mathbf{R}, \mathbf{\Gamma}),
\end{aligned} \tag{5.24}$$

where $\mathbf{\Gamma}$ denotes a set of all the auxiliary variables, $\{\zeta_{j,f,n}\}_{j,f,n}$, $\{\gamma_{k,j,f,n}\}_{k,j,f,n}$, $\{\alpha_{f,n}\}_{f,n}$, $\{\xi_{k,j,f,n}\}_{k,j,f,n}$, and $\{\kappa_{k,f,n}\}_{k,f,n}$.

By using Lemma 2, Jensen's inequality, and the concave inequality, we can also derive a majorizer for the case of the IS divergence in a similar manner (see

Appendix A). The majorizer is expressed as

$$\begin{aligned}\mathcal{F}_{\mathsf{IS}}(\mathbf{R}) &= \sum_j \lambda_j \mathcal{D}^j_{\mathsf{IS}}(\mathbb{S}|\hat{\mathbb{S}}) \qquad (5.25)\\ &\overset{c}{=} \sum_j \lambda_j \sum_{f,n} \Big(\frac{\mathtt{s}_{j,f,n}\Upsilon_{f,n}}{\mathtt{m}_{f,n}\Upsilon_{j,f,n}} - \log \Upsilon_{f,n} + \log \Upsilon_{j,f,n} \Big)\\ &\leq \sum_j \lambda_j \sum_{k,f,n} \Big(\frac{\mathtt{s}_{j,f,n}\zeta_{j,f,n}\mathsf{r}^2_{k,f}\hat{\mathsf{h}}^2_{k,n}}{2\mathtt{m}_{f,n}\xi_{j,f,n}} + \frac{\mathtt{s}_{j,f,n}\kappa^3_{k,j,f,n}}{2\mathtt{m}_{f,n}\zeta_{j,f,n}\mathsf{r}^2_{k,j,f}\hat{\mathsf{h}}_{k,j,n}}\\ &\quad - \gamma_{k,f,n}\log\frac{\mathsf{r}_{k,f}\hat{\mathsf{h}}_{k,n}}{\gamma_{k,f,n}} + \frac{\mathsf{r}_{k,j,f}\hat{\mathsf{h}}_{k,j,n}}{\alpha_{j,f,n}}\Big) + \text{const.}\\ &=: \mathcal{F}^+_{\mathsf{IS}}(\mathbf{R},\mathbf{\Gamma}), \qquad (5.26)\end{aligned}$$

where $\mathbf{\Gamma}$ denotes a set of all the auxiliary variables, $\{\zeta_{j,f,n}\}_{j,f,n}$, $\{\gamma_{k,f,n}\}_{k,f,n}$, $\{\alpha_{j,f,n}\}_{j,f,n}$, $\{\xi_{k,f,n}\}_{k,f,n}$, and $\{\kappa_{k,j,f,n}\}_{k,j,f,n}$.

These majorizers are particularly noteworthy in that they can be minimized analytically with respect to $\mathsf{r}_{k,j,f}$ since they are given as the sum of the reciprocal, logarithmic, first-order, and second-order functions. We can obtain the update rules for $\mathsf{r}_{k,j,f}$ by setting the partial derivatives of the above majorizers with respect to $\mathsf{r}_{k,j,f}$ at zeros. Thus, the optimal update of $\mathsf{r}_{k,j,f}$ is given by the positive solution of

$$\begin{aligned}&\lambda_j\Big(\sum_n \frac{\zeta_{j,f,n}\mathtt{m}_{f,n}}{\xi_{k,j,f,n}}\hat{\mathsf{h}}^2_{k,j,n}\Big)\mathsf{r}^4_{k,j,f} - \lambda_j\Big(\sum_n \mathtt{s}_{j,f,n}\gamma_{k,j,f,n}\Big)\mathsf{r}^2_{k,j,f}\\ &+ \Big(\lambda_j\sum_n \frac{\mathtt{s}_{j,f,n}\hat{\mathsf{h}}_{k,j,n}}{\alpha_{f,n}} + \sum_{j':j'\neq j}\lambda_{j'}\sum_n \frac{\mathtt{s}_{j',f,n}\hat{\mathsf{h}}_{k,j,n}}{\alpha_{f,n}}\Big)\mathsf{r}^3_{k,j,f}\\ &- \Big(\lambda_j\sum_n \frac{\mathtt{m}_{f,n}\kappa^3_{k,f,n}}{\zeta_{j,f,n}\hat{\mathsf{h}}^2_{k,j,n}} + \sum_{j':j'\neq j}\lambda_{j'}\sum_n \frac{\mathtt{m}_{f,n}\kappa^3_{k,f,n}}{\zeta_{j',f,n}\hat{\mathsf{h}}_{k,j,n}}\Big) = 0 \qquad (5.27)\end{aligned}$$

for the KL divergence case, and

$$\begin{aligned}&\Big(\lambda_j\sum_n \frac{\zeta_{j,f,n}\mathtt{s}_{j,f,n}\hat{\mathsf{h}}^2_{k,j,n}}{\mathtt{m}_{f,n}\xi_{k,f,n}} + \sum_{j':j'\neq j}\lambda_{j'}\sum_n \frac{\zeta_{j',f,n}\mathtt{s}_{j',f,n}\hat{\mathsf{h}}^2_{k,j,n}}{\mathtt{m}_{f,n}\xi_{k,f,n}}\Big)\mathsf{r}^4_{k,j,f}\\ &- \Big(\lambda_j\sum_n \gamma_{k,f,n} + \sum_{j'}\lambda_{j'}\sum_n \gamma_{k,f,n}\Big)\mathsf{r}^2_{k,j,f}\end{aligned}$$

$$+ \lambda_j \sum_n \frac{\hat{\mathsf{h}}_{k,j,n}}{\alpha_{j,f,n}} \mathsf{r}^3_{k,j,f} - \lambda_j \sum_n \frac{\mathsf{s}_{j,f,n} \kappa^3_{k,j,f,n}}{\zeta_{f,n} \mathtt{m}_{f,n} \hat{\mathsf{h}}^2_{k,j,n}} = 0 \tag{5.28}$$

for the IS divergence case. It is worth noting that since in $\mathcal{F}^+_{\mathsf{KL}}(\mathbf{R}, \mathbf{\Gamma})$ and $\mathcal{F}^+_{\mathsf{IS}}(\mathbf{R}, \mathbf{\Gamma})$, each element of $\mathbf{R}$ is isolated in a separate term, we can update each of the elements in parallel. Thus, this algorithm is well suited to parallel implementations. Furthermore, since each of the update rules consists of a negative 0th-order term and a negative 2nd-order term, it turns out that there is only one positive solution, implying that there is no need to solve a solution selection problem.

$\mathcal{F}^+_{\mathsf{KL}}(\mathbf{R}, \mathbf{\Gamma})$ is minimized with respect to the auxliary variables when the exact bounds of Eqs. (5.14), (5.16), (5.18), (5.20), and (5.22) are achieved, namely when Eqs. (5.15), (5.17), (5.19), (5.21), and (5.23). The proposed basis training algorithm with the KL divergence can therefore be summarized as *Algorithm 5*. The algorithm with the IS divergence can be developed in the same way. Since the proposed algorithm is derived based on the auxiliary function approach, we call the proposed method "AuxDNMF".

Algorithm 5 Proposed basis training algorithm with KL divergence

Require: $\mathbf{S}_1, \ldots, \mathbf{S}_J$, $\mathbf{M}$
  Compute $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{H}}$ using NMF to solve (5.1) for all j.
  Compute $\hat{\mathbf{H}}$ using NMF to solve (5.2).
  Initialize $\mathbf{R}$ by, for example, $\mathbf{R} \leftarrow \tilde{\mathbf{B}}$.
  Fix $\hat{\mathbf{H}}$.
  while not converged **do**
    Update $\mathbf{\Gamma}$ via Eqs. (5.15), (5.17), (5.19), (5.21), and (5.23).
    Update $\mathbf{R}$ by solving (5.27).
  end while
  return $\tilde{\mathbf{B}}$, $\mathbf{R}$

5.4 Experimental evaluations

5.4.1 Dataset and settings

To evaluate the effect of the proposed algorithm, we conducted speech enhancement experiments, namely $j = \{\mathrm{s}, \mathrm{n}\}$. For comparison, we tested (i) the stan-

dard supervised NMF method [129] with EU distance (SNMF_EU), KL divergence (SNMF_KL), and IS divergence (SNMF_IS), (ii) DNMF using the MU algorithm [19] with KL divergence (DNMF_KL) and EU distance (DNMF_EU), and (iii) proposed AuxDNMF with KL divergence (AuxDNMF_KL) and IS divergence (AuxDNMF_IS). Note that we have excluded DNMF_IS from the baselines since it has not been studied in [19]. Also note that the results for AuxDNMF_EU are not provided. This is because we have yet to come up with an auxiliary function with a tractable form for the EU distance case.

We constructed the training and test datasets using speech signals excerpted from the WSJ0 corpus [109] and noise signals excerpted from the CHiME4 background noise database [130], which includes 4 types of noise recorded in a bus, cafe, pedestrian area, and street, respectively. The training dataset consisted of 600 utterances, each of which was created by mixing randomly selected utterances from `si_tr_s` and noise signals with SNRs set at $\{-5, 0, 5\}$dB. We also created a validation dataset consisting of 90 utterances in the same way. Each of the four test datasets consisted of 100 utterances, half of which we created using speech signals in `si_tr_s` and the other half using speech signals of different speakers in `si_dt_05`. The SNRs for three of the four test datasets were set at $\{-5, 0, 5\}$ dB and those for the remaining dataset were randomly set between $[-10, 10]$ dB.

All the audio signals were monaural and downsampled to 16 kHz. STFT was computed using a Hanning window that was 32 ms long with a 16 ms overlap. We used the same number K of basis for speech and noise, i.e., $K^{\mathrm{s}} = K^{\mathrm{n}} = K$. In this task, we tested $K = \{25, 50, 100\}$. For $K = 100$, we evaluated the effectiveness of sparse regularization in the case of a large number of basis numbers by setting $\lambda_{\mathrm{sparse}} = \{0, 0.5, 1, 5, 10\}$. SNMF_KL was run for 100 iterations. For the DNMF algorithms, SNMF_KL was used for initialization. For the separation, the Wiener filter was constructed using the trained basis and activation matrices obtained using the standard NMF that was run for 100 iterations.

Table 5.1: Comparison of the computational times with basis number $K = 50$.

Method	Time / Iteration [sec]	Total time [sec]
SNMF_EU	0.1468	62.6800
SNMF_KL	0.4687	192.3910
SNMF_IS	0.2820	121.4615
DNMF_EU	1.3460	256.3109
DNMF_KL	0.6234	236.6248
AuxDNMF_KL	1.4947	434.0287
AuxDNMF_IS	1.5184	437.2275

5.4.2 Convergence behaviors and computational time

We compared the convergence behaviors of the proposed algorithms, DNMF_EU, and DNMF_KL within the first 500 iterations. For all the algorithms, we used the same initialization and evaluated the SDR improvements. Two examples are shown in Fig. 5.2. As can be seen from the example when tested on bus noise with $K = 100$, DNMF_EU and DNMF_KL did not decrease the objective functions monotonically. This indeed shows the fact that each update in the MU algorithms does not guarantee a decrease in the objective functions. It is also worth noting that the objective function value does not directly reflect the speech enhancement performance, as shown in the experimental results when tested on street noise with $K = 50$. According to the SDR results obtained with the validation dataset as well as the setting in [19], in the following experiments, we set the iteration number at 150 for the proposed algorithms and 25 for the MU algorithms.

We compared the computational times of all the algorithms with $K = 50$ using the training data of about 1 hour long. The algorithms were implemented using MATLAB and run on an Intel Xeon Gold 5120 @2.2GHz processor. Table 5.1 shows the average computational time of updating **B** or **R** at each iteration and that of the entire process. Note that the total time of DNMF includes the time of computing $\tilde{\mathbf{B}}$ for initialization and $\hat{\mathbf{H}}$. That the time complexity of the proposed algorithm is $O(FKNJ^2)$, whereas that of the standard NMF and DNMF algorithms with multiplicative update rules is $O(FKNJ)$. Since J was 2 in the speech enhancement task, it did not have a significant impact on the computation time. Rather, the

Table 5.2: SDR obtained with $K = \{25, 50, 100\}$ average over all the test datasets (4 types noise) with 5 random initializations. The average input SDR was about 0.063 dB.

Method	Basis number K		
	25	50	100
SNMF_EU	2.55	2.52	2.53
SNMF_KL	2.42	2.38	2.44
SNMF_IS	2.11	1.83	1.68
DNMF_EU	2.87	2.69	2.71
DNMF_KL	2.52	2.62	2.63
AuxDNMF_KL	**3.49**	**3.39**	**3.36**
AuxDNMF_IS	2.10	2.26	2.34

increase in the number of iterations in the proposed algorithm led to an increase in the total computation time.

5.4.3 Speech enhancement performance

The speech enhancement performances were numerically evaluated in terms of SDRs, SIRs, and SARs. Table 5.2 shows the average SDRs took over all the test data with basis number $K = \{25, 50, 100\}$. For each noise type with different K, we conducted 5 trials with different initializations. The average input SDR of the test data was about 0.063 dB. As Table 5.2 shows, increasing the bases did not always lead to an improvement in speech enhancement performance. Comparing the results of the standard NMF and DNMF algorithms, we found that the latter outperformed the former. This indicates the effectiveness of the ability to learn discriminative bases. Furthermore, the proposed algorithm performed best among all the algorithms based on the same divergence measure. In Table 5.3, the average SDRs, SIRs, and SARs evaluated using $K = 25$ with various input SNRs are shown. These results were averaged over 4 noise types. As the results show, AuxDNMF_KL performed best among all the algorithms in terms of SDR and SIR. Specifically, it achieved about 1.2 dB improvements over DNMF_EU and DNMF_KL, and about 1.7 dB improvements over SNMF_KL. This shows that the proposed algorithm with the KL divergence criterion had a better ability to learn dis-

Table 5.3: From top to bottom are the average SDRs, SIRs, SARs over 4 types noise with basis number $K = 25$.

	Method	Input SNR [dB]				
		-5	0	5	[-10,10]	Avg
SDR [dB]	unprocessed	-4.92	0.05	5.03	0.09	0.06
	SNMF_EU	-2.01	2.69	7.07	2.48	2.55
	SNMF_KL	-2.04	2.63	6.79	2.29	2.42
	SNMF_IS	-2.35	2.35	6.45	2.00	2.11
	DNMF_EU	-1.61	3.02	7.30	2.77	2.87
	DNMF_KL	-1.99	2.73	6.94	2.41	2.52
	AuxDNMF_KL	**-0.92**	**3.78**	**7.77**	**3.34**	**3.49**
	AuxDNMF_IS	-2.35	2.19	6.52	2.04	2.10
SIR [dB]	SNMF_EU	-1.18	3.73	8.75	3.87	3.79
	SNMF_KL	-1.04	3.94	8.99	4.05	3.94
	SNMF_IS	-0.87	4.22	9.22	4.26	4.21
	DNMF_EU	-0.41	4.49	9.51	4.61	4.55
	DNMF_KL	-1.01	3.94	8.74	3.91	3.90
	AuxDNMF_KL	**0.75**	**5.77**	**10.53**	**5.73**	**5.70**
	AuxDNMF_IS	-1.29	3.44	8.22	3.51	3.47
SAR [dB]	SNMF_EU	**10.04**	**11.62**	**13.06**	**11.62**	**11.58**
	SNMF_KL	8.90	10.33	11.56	10.24	10.26
	SNMF_IS	7.11	8.77	10.48	8.83	8.80
	DNMF_EU	8.85	10.60	12.35	10.64	10.61
	DNMF_KL	9.26	10.97	12.56	10.91	10.93
	AuxDNMF_KL	7.88	10.00	11.99	9.94	9.95
	AuxDNMF_IS	8.65	10.40	12.43	10.61	10.52

criminative bases than the baseline algorithms. However, the SARs obtained with the proposed algorithms tended to be lower than those obtained with the baseline algorithms.

We also evaluated the effectiveness of sparse regularization. The results are shown in Table 5.4. We found that $\lambda_{\text{sparse}} = 0.5$ achieved the best score for each method except for AuxDNMF_IS, where the best performance was obtained without sparse regularization. AuxDNMF_KL outperformed other methods regardless of the sparse regularization.

Table 5.4: SDR [dB] obtained with $\lambda_{\text{sparse}} = \{0, 0.5, 1, 5, 10\}$ and $K = 100$ average over all the test datasets with 5 random initializations. Bold font shows the highest score for each method.

Method	λ_{sparse}				
	0	0.5	1	5	10
SNMF_EU	2.53	**2.66**	2.64	2.37	2.14
SNMF_KL	2.44	**2.48**	2.41	2.40	2.40
SNMF_IS	1.68	**1.98**	1.96	1.80	1.78
DNMF_EU	2.71	**3.62**	3.52	3.12	2.88
DNMF_KL	2.63	**3.78**	3.77	3.77	3.77
AuxDNMF_KL	3.36	**3.88**	3.87	3.87	3.87
AuxDNMF_IS	**2.34**	1.99	1.93	1.71	1.65

5.5 Summary of chapter 5

DNMF is noteworthy in that it directly uses the reconstruction errors of the separated signals as the training criterion, which eliminates the inconsistency between the objective functions for training and separation in the conventional NMF method and is able to increase the discriminative power of the trained basis. However, such training criterion causes difficulty as regards optimization. In this chapter, we derived a novel majorizer for the objective function of DNMF and successfully developed an MM algorithm that is guaranteed to converge to a stationary point. Experimental results showed that the proposed algorithm with the KL divergence criterion achieved significant improvements in terms of the SDR and SIR over standard NMF and DNMF using the MU algorithm.

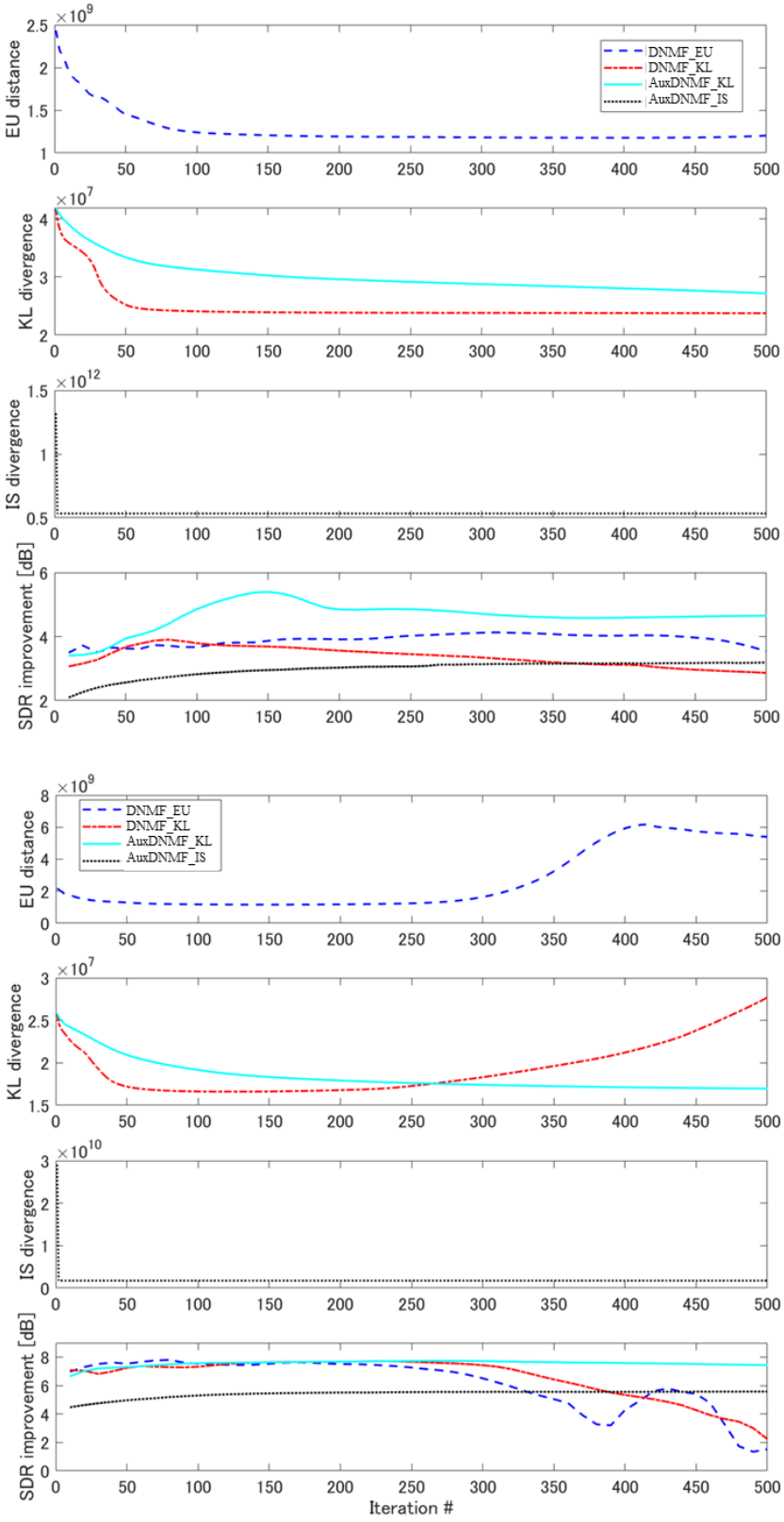

Figure 5.2: Convergence behavior and corresponding SDR improvements obtained with each method in street noise with $K = 50$ case (top) and bus noise with basis number $K = 100$ case (bottom).

Chapter 6

Conclusion

6.1 Summary of thesis

In this dissertation, we addressed speech enhancement problems by separating the target speech signal and non-target signals, which can be applied to many speech processing systems. This problem is divided into determined and underdetermined multichannel and single-channel cases, depending on the relationship between the numbers of microphones and sources. Moreover, depending on the different hardware configurations and application prerequisites, more conditions should be considered during the algorithm development. We proposed several source separation methods, which considered each of the following conditions:

- supervised determined multichannel with high performance;
- supervised determined multichannel with a low computational cost;
- guided underdetermined multichannel;
- real-time guided underdetermined multichannel;
- supervised single-channel.

We proposed novel effective optimization algorithms based on the auxiliary function approach for all the methods so that the objective functions are guaranteed to be non-increasing at each iteration.

In Chapter 3, we proposed two determined multichannel source separation methods called MVAE and MSGAN, which utilize CVAE or StarGAN to model spectrograms of utterances of sources. These methods can be considered as extending FDICA-based methods by incorporating DNNs into the source model so that the source model has stronger representation power to improve the accuracy of source estimation. Owing to the sequence modeling, the CVAE and StarGAN source models can capture the spectro-temporal structures of sources. Thanks to the conditional modeling, we are allowed to separate the speaker information, which is time-invariant, from other speech information. The advantage of the separation performance of these two methods was experimentally confirmed in multi-speaker source separation tasks. Moreover, to reduce the high computational cost of the MVAE and MSGAN method, we proposed a fast optimization algorithm called FastMVAE. Instead of estimating parameters that exactly maximize the log-posterior, FastMVAE utilizes an auxiliary classifier and encoder to estimate parameters that maximize the approximate log-posterior. Although the convergence is not guaranteed anymore in the fast algorithm with the approximation, which slightly decreased the separation performance, FastMVAE successfully reduced the computational time of more than 90% even when using a CPU, making it closer to practical applications.

In Chapter 4, a geometrically constrained IVA method, called GCAV-IVA, was described. As one of the powerful FDICA-based BSS methods, IVA can simultaneously solve permutation problem and source separation. However, it has been reported that block permutation occurs in IVA, and postprocessing is generally needed for applying IVA to speech enhancement. To overcome these problems, we proposed a GCIVA that combines LCVM-based geometric constraints with IVA, making it possible to control the demixing filters manually. By incorporating the constraints, GCIVA could work in a situation where there are a number of sources equals to the number of microphones and diffuse noise. This relaxes the strict restriction of determined conditions. Furthermore, since LCVM-based constraints can be designed to perform as BM, which is used in GSC to suppress the estimated interferences from the target channel, the proposed GCIVA has the potential to handle underdetermined situations. We further extended the algorithm to

an online algorithm, which could perform in real-time. To evaluate the proposed methods, we performed an experimental evaluation of speech enhancement. The results revealed that the proposed methods could significantly improve enhancement performance.

The main topic in Chapter 5 was supervised single-channel source separation, where we proposed a parameter optimization algorithm for DNMF based on the auxiliary function approach. Since the objective function of DNMF is analytically complex, which is formulated as a bi-level optimization problem, an MU algorithm has been proposed in a heuristic way, which limits the unleashing of the full potential of DNMF. To address this problem, we successfully found majorizers for the objective function of DNMF using KL divergence and IS divergence and derived the parameter update rules. Through simulation experiments, we showed that the proposed methods could converge fast and outperformed existing MU algorithms in the ability to speech enhancement.

6.2 Future perspectives

We have proposed several methods to improve the performance of source separation and reduce the computational cost and time to meet the prerequisites for practical use in real environments, but there is still much room for improvement.

- Although DGM-based source models have shown to outperform the conventional NMF model, it has been reported that the likelihood produced by a DGM does not always coincide with the speech quality [131]. Namely, speech with bad quality may be scored with a high likelihood. Therefore, it is necessary to improve the discriminative power of the source model so that the model not only scores the clean speech with high likelihood, but also scores other signals, such as noise or mixture signals, with low likelihood.

- Although FastMVAE has significantly reduced the computational time, it also leads to performance degradation, which is undesirable. One possible reason may be the inadequate training, where the log-likelihood of the reconstructed spectrograms trained with an ACVAE was lower than that trained

with a CVAE. This indicates that the source model trained with ACVAE has worse generative power, which subsequently decreases the log-posterior estimation accuracy. A more effective training approach is necessary to reduce the computational time of the MVAE method while maintains its impressive source separation performance.

- We have already confirmed the ability of GCAV-IVA to estimating interferences and noise. To perform it as an underdetermined situation method, extending GCAV-IVA to a GSC framework is necessary. Moreover, although we have investigated the performance of the offline algorithm of GCAV-IVA in an in-car environment [C11], we have to study the online algorithm in order to make it possible for practical applications in more realistic environments.

- In Chapter 3, we have made constraints on the source model to solve the permutation problem, whereas, in Chapter 4, we have made constraints on the demixing filters to eliminate the block permutation. We can expect that methods that combine the geometric constraints with the MVAE or MSGAN method are able to further improve the source separation performance by taking advantage of both approaches.

- As mentioned in the first point, it is necessary to improve the discriminative power of the source model. Another promising approach is to perform discriminative training for the source model as done in the DNMF. Namely, we can train a generative source model for estimating the underlying source signals from a mixture signal and a discriminative source model for separating.

Appendix A

Derivation of a majorizer for DNMF with IS divergence

For DNMF with the IS divergence case, a majorizer can be derived using Lemma 2 introduced in Chapter 5, Jensen's inequality, and the concave inequality. We express indices f and n as subscript for the notation simplicity. The objective function is given as

$$\begin{aligned}\mathcal{F}_{\mathsf{IS}}(\mathbf{R}) &= \sum_{j} \lambda_j \mathcal{D}^{j}_{\mathsf{IS}}(\mathbb{S}|\hat{\mathbb{S}}) \\ &\overset{c}{=} \sum_{j} \lambda_j \sum_{f,n} \Big(\frac{\mathtt{s}_{j,f,n} \Upsilon_{f,n}}{\mathtt{m}_{f,n} \Upsilon_{j,f,n}} - \log \Upsilon_{f,n} + \log \Upsilon_{j,f,n} \Big).\end{aligned} \tag{A.1}$$

First, we focus on the first term of (A.1). By using Lemma 2, we can obtain an upper bound

$$\mathcal{F}_{\mathsf{IS}}(\mathbf{R}) \leq \sum_{j} \lambda_j \sum_{f,n} \Big(\frac{\zeta_{j,f,n} \mathtt{s}_{j,f,n} \Upsilon^2_{f,n}}{2\mathtt{m}_{f,n}} + \frac{\mathtt{s}_{j,f,n}}{2\zeta_{j,f,n} \mathtt{m}_{f,n} \Upsilon^2_{j,f,n}} - \log \Upsilon_{f,n} + \log \Upsilon_{j,f,n} \Big), \tag{A.2}$$

the equality of which holds if and only if

$$\zeta_{j,f,n} = \frac{1}{\Upsilon_{j,f,n} \Upsilon_{f,n}}. \tag{A.3}$$

In the following, we construct upper bound for each of the terms on the right-hand side of (A.2).

Since a quadratic function with positive coefficient is convex, we can apply Jensen's inequality to the first term:

$$\Upsilon_{f,n}^2 \leq \sum_k \frac{\mathsf{r}_{k,f}^2 \hat{\mathsf{h}}_{k,n}^2}{\xi_{k,f,n}}, \tag{A.4}$$

where $\xi_{k,f,n}$ is a positive number that satisfies $\sum_j \xi_{k,f,n} = 1$. The equality of (A.4) holds if and only if

$$\xi_{k,f,n} = \frac{\mathsf{r}_{k,f}\hat{\mathsf{h}}_{k,n}}{\sum_{k'} \mathsf{r}_{k',f}\hat{\mathsf{h}}_{k',n}}. \tag{A.5}$$

For the second term, which is a function of $1/x^2$, we can utilize the fact that the function in the first quadrant is convex and use Jensen's inequality to obtain an upper bound

$$\frac{1}{\Upsilon_{j,f,n}^2} \leq \sum_k \frac{\kappa_{k,j,f,n}^3}{\mathsf{r}_{k,j,f}^2 \hat{\mathsf{h}}_{k,j,n}^2}, \tag{A.6}$$

where $\kappa_{k,j,f,n}$ is a positive number that sums to unity. We can confirm that the equality of this inequality holds if and only if

$$\kappa_{k,j,f,n} = \frac{\mathsf{r}_{k,j,f}\hat{\mathsf{h}}_{k,j,n}}{\sum_{k'} \mathsf{r}_{k',j,f}\hat{\mathsf{h}}_{k',j,n}}. \tag{A.7}$$

Since $-\log x$ is a convex function, we can apply Jensen's inequality to the third term of (A.2),

$$-\log \Upsilon_{f,n} \leq -\sum_k \gamma_{k,f,n} \log \frac{\mathsf{r}_{k,f}\hat{\mathsf{h}}_{k,n}}{\gamma_{k,f,n}}, . \tag{A.8}$$

Here, $\gamma_{k,f,n}$ is a positive weight that sums to unity. The equality of (A.8) holds if and only if

$$\gamma_{k,f,n} = \frac{\mathsf{r}_{k,f}\hat{\mathsf{h}}_{k,n}}{\sum_{k'} \mathsf{r}_{k',f}\hat{\mathsf{h}}_{k',n}}. \tag{A.9}$$

For the fourth term $\log x$, we utilize a tengent line to obtain an upper bound

$$\log \Upsilon_{j,f,n} \leq \sum_k \frac{\mathsf{r}_{k,j,f}\hat{\mathsf{h}}_{k,j,n}}{\alpha_{j,f,n}} + \log \alpha_{j,f,n} - 1, \tag{A.10}$$

where $\alpha_{j,f,n}$ is an arbitrary positive number. The equality holds if and only if

$$\alpha_{j,f,n} = \Upsilon_{j,f,n} = \sum_k \mathsf{r}_{k,j,f}\hat{\mathsf{h}}_{k,j,n}. \tag{A.11}$$

From (A.4), (A.6), (A.8), and (A.10), we can construct a majorizer for the objective function with IS divergence as

$$\begin{aligned}\mathcal{F}_{\mathsf{IS}}(\mathbf{R}) \leq \sum_j \lambda_j \sum_{k,f,n} \Big(& \frac{\mathsf{s}_{j,f,n}\zeta_{j,f,n}\mathsf{r}_{k,f}^2\hat{\mathsf{h}}_{k,n}^2}{2\mathtt{m}_{f,n}\xi_{k,f,n}} + \frac{\mathsf{s}_{j,f,n}\kappa_{k,j,f,n}^3}{2\mathtt{m}_{f,n}\zeta_{l,f,n}\mathsf{r}_{k,j,f}^2\hat{\mathsf{h}}_{k,j,n}} \\ & - \gamma_{k,f,n}\log\frac{\mathsf{r}_{k,f}\hat{\mathsf{h}}_{k,n}}{\gamma_{k,f,n}} + \frac{\mathsf{r}_{k,j,f}\hat{\mathsf{h}}_{k,j,n}}{\alpha_{j,f,n}}\Big) + \text{const.} \\ =: \mathcal{F}_{\mathsf{IS}}^{+}(\mathbf{R}, \boldsymbol{\Gamma}),\end{aligned} \tag{A.12}$$

where $\boldsymbol{\Gamma}$ denotes a set of all the auxiliary variables, $\{\zeta_{j,f,n}\}_{j,f,n}$, $\{\gamma_{k,f,n}\}_{k,f,n}$, $\{\alpha_{j,f,n}\}_{j,f,n}$, $\{\xi_{k,f,n}\}_{k,f,n}$, and $\{\kappa_{k,j,f,n}\}_{k,j,f,n}$.

Appendix B

List of Publications

B.1 Journal Papers

[J1] **L. Li**, H. Kameoka, S. Inoue, and S. Makino, "FastMVAE: A fast optimization algorithm for the multichannel variational autoencoder method," *IEEE Access*, vol. 8, pp. 228740–228753, Dec. 2020. (Chapter 3)

[J2] H. Kameoka, **L. Li**, S. Inoue, and S. Makino, "Supervised determined source separation with multichannel variational autoencoder," *Neural Computation*, vol. 31, no. 9, pp. 1891–1914, Sep. 2019. (Chapter 3)

[J3] **L. Li**, H. Kameoka, and S. Makino, "Majorization-minimization algorithm for discriminative non-negative matrix factorization," *IEEE Access*, vol. 8, pp. 227399–227408, Dec. 2020. (Chapter 5)

B.2 Peer-Reviewed International Conferences

[I1] **L. Li**, H. Kameoka, and S. Makino, "Fast MVAE: Joint separation and classification of mixed sources based on multichannel variational autoencoder with auxiliary classifier," in *Proc. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2019)*, pp. 546–550, May 2019. (Chapter 3)

[I2] **L. Li**, H. Kameoka, and S. Makino, "Determined audio source separation

with multichannel star generative adversarial network," in *Proc. The 30th IEEE International Workshop on Machine Learning for Signal Processing (MLSP2020)*, Sep. 2020. (Chapter 3)

[I3] **L. Li**, and H. Kameoka, "Deep clustering with gated convolutional networks," in *Proc. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2018)*, pp. 16–20, Apr. 2018. (Chapter 3)

[I4] **L. Li**, and K. Koishida, "Geometrically constrained independent vector analysis for directional speech enhancement," in *Proc. 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2020)*, pp. 846–850, May 2020. (Chapter 4)

[I5] **L. Li**, K. Koishida, and S. Makino, "Online directional speech enhancement using geometrially constrained independent vector analysis," in *Proc. The 21th Annual Conference of the International Speech Communication Association (Interspeech2020)*, pp. 61–65, Oct. 2020. (Chapter 4)

[I6] **L. Li**, H. Kameoka, and S. Makino, "Discriminative non-negative matrix factorization with majorization-minimization," in *Proc. The 5th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA2017)*, pp. 141–145, Mar. 2017. (Chapter 5)

B.3 Other Journal Papers

[O1] H. Kameoka, T. Higuchi, M. Tanaka, and **L. Li**, "Non-negative matrix factorization with basis clustering using cepstral distance regularization," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, Vol. 26, No. 6, pp. 1025–1036, Jun. 2018.

[O2] S. Seki, H. Kameoka, **L. Li**, T. Toda, and K. Takeda, "Underdetermined source separation based on generalized multichannel variational autoencoder," *IEEE Access*, vol. 7, No. 1, pp. 168104–168115, Nov. 2019.

B.4 Other International Conferences

[C1] **L. Li**, H. Kameoka, T. Higuchi, and H. Saruwatari, "Semi-supervised joint enhancement of spectral and cepstral sequences of noisy speech," in *Proc. The 17th Annual Conference of the International Speech Communication Association (Interspeech 2016)*, pp. 3753–3757, Sep. 2016.

[C2] **L. Li**, H. Kameoka, T. Toda, and S. Makino, "Speech enhancement using non-negative spectrogram models with mel-generalized cepstral regularization," in *Proc. The 18th Annual Conference of the International Speech Communication Association (Interspeech2017)*, pp. 1998–2002, Aug. 2017.

[C3] **L. Li**, H. Kameoka, and S. Makino, "Mel-generalized cepstral regularization with discriminative non-negative matrix factorization," in *Proc. The 27th IEEE International Workshop on Machine Learning for Signal Processing (MLSP2017)*, Sep. 2017.

[C4] **L. Li**, K. Yamaoka, Y. Koshino, M. Matsumoto, and S. Makino, "Voice activity detection under high levels of noise using gated convolutional neural networks," in *Proc. International Congress on Acoustics (ICA2019)*, pp. 2862–2869, Sep. 2019.

[C5] **L. Li**, T. Toda, K. Morikawa, K. Kobayashi, and S. Makino, "Improving singing aid system for laryngectomees with statistical voice conversion and VAE-SPACE," in *Proc. 20th International Society for Music Information Retrieval Conference (ISMIR2019)*, pp. 784–790, Nov. 2019.

[C6] S. Inoue, H. Kameoka, **L. Li**, S. Seki, and S. Makino, "Joint separation and dereverberation of reverberant mixtures with multichannel variational autoencoder," in *Proc. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2019)*, pp. 56–60, May 2019.

[C7] S. Inoue, **L. Li**, H. Kameoka, and S. Makino, "Joint separation, dereverberation and classification of mixed sources using multichannel variational autoencoder with auxiliary classifier," in *Proc. International Congress on Acoustics (ICA2019)*, pp. 6988–6995, Sep. 2019.

[C8] K. Yamaoka, **L. Li**, N. Ono, S. Makino, and T. Yamada, "CNN-based virtual microphone signal estimation for MPDR Beamforming in underdetermined situations," in *Proc. The 2019 European Signal Processing Conference (EUSIPCO2019)*, pp. 1049–1053, Sep. 2019.

[C9] S. Seki, H. Kameoka, **L. Li**, T. Toda, and K. Takeda, "Generalized multichannel variational autoencoder for underdetermined source separation," in *Proc. The 2019 European Signal Processing Conference (EUSIPCO2019)*, pp. 1973–1977, Sep. 2019.

[C10] R. Takahashi, K. Yamaoka, **L. Li**, S. Makino, T. Yamada, and M. Matsumoto, "Underdetermined multichannel speech enhancement using time-frequency-bin-wise switching beamformer and gated CNN-based time-frequency mask for reverberant environments," in *Proc. RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP2020)*, Feb. 2020.

[C11] K. Goto, **L. Li**, R. Takahashi, S. Makino, and T. Yamada, "A study on geometrically constrained IVA with auxiliary function approach and VCD for in-car communication," in *Proc. The 12th annual conference of Asia-Pacific Signal and Information Processing Association (APSIPA2020)*, pp. 858–862, Dec. 2020.

[C12] R. Takahashi, **L. Li**, S. Makino, and T. Yamada, "VMInNet: Interpolation of virtual microphones in optimal latent space explored by autoencoder," in *Proc. The 2021 RISP Internaonal Workshop on Nonlinear Circuits, Communicaions and Signal Processing (NCSP2021)*, Mar. 2021. (accepted)

[C13] N. Murashima, H. Kameoka, **L. Li**, S. Seki, and S. Makino, "Single-channel multi-speaker separation via discriminative training of variational autoencoder spectrogram model," in *Proc. The 2021 RISP Internaonal Workshop on Nonlinear Circuits, Communicaions and Signal Processing (NCSP2021)*, Mar. 2021. (accepted)

[C14] S. Nakaoka, **L. Li**, S. Inoue, and S. Makino, "Teacher-student learning for low-latency online speech enhancement using wave-U-net," in *Proc. 2021*

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2021), Jun. 2021. (accepted)

[C15] S. Inoue, H. Kameoka, **L. Li**, and S. Makino, "SepNet: A deep separation matrix prediction network for multichannel audio source separation," in *Proc. 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2021)*, Jun. 2021. (accepted)

B.5 Non-Reviewed Domestic (Japanese) Conferences and Workshops

[D1] **L. Li**, H. Kameoka, T. Higuchi, and H. Saruwatari, "Speech enhancement based on semi-supervised non-negative matrix factorization with cepstral distance regularization," *2016 Spring Meeting of Acoustical Society of Japan (ASJ)*, 1-P-27, pp. 721–724, Mar. 2016 (in Japanese).

[D2] **L. Li**, H. Kameoka, T. Higuchi, H. Saruwatari, and S. Makino, "Joint enhancement of spectral cand cepstral sequences of noisy speech," *IEICE Technical Report*, SP2016-32, vol. 116, no. 189, pp. 29–32, Aug. 2016 (in Japanese).

[D3] Y. Zou, **L. Li** and H. Kameoka, "Vocal tract spectrogram estimation with formant frequency contour factorization," *2017 Spring Meeting of Acoustical Society of Japan (ASJ)*, 1-Q-41, pp. 323–326, Mar. 2017.

[D4] **L. Li**, H. Kameoka, and S. Makino, "Auxiliary function approach to discriminative non-negative matrix factorization," *2017 Spring Meeting of Acoustical Society of Japan (ASJ)*, 1-P-4, pp. 519–522, Mar. 2017 (in Japanese).

[D5] **L. Li** and H. Kameoka, "Multi-speaker separation using deep clustering with gated CNN," *2018 Spring Meeting of Acoustical Society of Japan (ASJ)*, 1-4-17, pp. 453–456, Mar. 2018 (in Japanese).

[D6] S. Seki, H. Kameoka, **L. Li**, T. Toda, and K. Takeda, "Underdetermined source separation using multichannel variational autoencoder," *2019 Spring Meeting*

of Acoustical Society of Japan (ASJ), 1-6-20, pp. 229–230, Mar. 2019 (in Japanese).

[D7] S. Inoue, H. Kameoka, **L. Li**, S. Seki, and S. Makino, "Unified approach for determined BSS and dereverberation using multichannel variational autoencoder," *2019 Spring Meeting of Acoustical Society of Japan (ASJ)*, 2-Q-32, pp. 399–402, Mar. 2019 (in Japanese).

[D8] R. Takahashi, K. Yamaoka, **L. Li**, S. Makino, and T. Yamada, "Underdetermined speech enhancement with time-frequency-bin-wise switching beamformer and gated convolutional network-based time-frequency mask," *2019 Spring Meeting of Acoustical Society of Japan (ASJ)*, 1-6-5, pp. 181–184, Mar. 2019 (in Japanese).

[D9] **L. Li**, H. Kameoka, and S. Makino, "Fast algorithm for semi-blind source separation using multichannel variational autoencoder with auxiliary source label classifier," *2019 Spring Meeting of Acoustical Society of Japan (ASJ)*, 1-6-10, pp. 201–204, Mar. 2019 (in Japanese).

[D10] S. Seki, H. Kameoka, **L. Li**, T. Toda, and K. Takeda, "An evaluation of underdetermined source separation based on multichannel variational autoencoder," *IEICE Technical Report*, EA2018-154, vol. 118, no. 495, pp. 323–328, Mar.2019 (in Japanese).

[D11] **L. Li**, Y. Koshino, M. Matsumoto, and S. Makino, "Voice activity detection under high levels of noise using gated convolutional neural networks," *IEICE Technical Report*, EA2018-102, vol. 118 no. 495, pp. 19–24, Mar. 2019 (in Japanese).

[D12] **L. Li**, H. Kameoka, S. Inoue, and S. Makino, "Speaker-independent source separation with multichannel variational autoencoder," *IEICE Technical Report*, EA2019-77, vol. 119, no. 334, pp. 79–84, Dec. 2019 (in Japanese).

[D13] H. Taga, S. Seki, **L. Li**, K. Takeda, and T. Toda, "Fundamental frequency contour generation of singing voice based on variational autoencoder incorporating generalized command response model", it 2020 Autumn Meeting

of Acoustical Society of Japan (ASJ), 1-2-16, pp. 731–732, Sep. 2020 (in Japanese).

Appendix C

Awards Received

1. The 14th Student Conference Paper Award from IEEE Signal Processing Society (SPS) Japan Chapter, December 2020.
2. The 2nd Student Award from IEEE Signal Processing Society (SPS) Tokyo Joint Chapter, November 2018.
3. The 13th Best Student Presentation Award from Acoustical Society of Japan (ASJ), March 2016.
4. The Best Student Presentation Award from IEICE Electroacoustics Symposium, December 2019.
5. Chair Award of the Department of Computer Science from Graduate School of Systems and Information Engineering, University of Tsukuba, March 2018.

Bibliography

[1] P. C. Loizou, *Speech enhancement: Theory and practice*. CRC press, 2013.

[2] S. Makino, *Audio source separation*. Springer, 2018, vol. 433.

[3] S. Makino, T.-W. Lee, and H. Sawada, *Blind speech separation*. Springer, 2007, vol. 615.

[4] P. Comon, "Independent component analysis, a new concept?" *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.

[5] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, no. 1-3, pp. 21–34, 1998.

[6] T. Kim, T. Eltoft, and T.-W. Lee, "Independent vector analysis: An extension of ica to multivariate components," in *International conference on independent component analysis and signal separation*. Springer, 2006, pp. 165–172.

[7] A. Hiroe, "Solution of permutation problem in frequency domain ica, using multivariate probability density functions," in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2006, pp. 601–608.

[8] H. Kameoka, T. Yoshioka, M. Hamamura, J. Le Roux, and K. Kashino, "Statistical model of speech signals based on composite autoregressive system with application to blind source separation," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2010, pp. 245–253.

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

[10] N. Makishima, S. Mogami, N. Takamune, D. Kitamura, H. Sumino, S. Takamichi, H. Saruwatari, and N. Ono, "Independent deeply learned matrix analysis for determined audio source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 10, pp. 1601–1615, 2019.

[11] L. C. Parra and C. V. Alvino, "Geometric source separation: Merging convolutive source separation with geometric beamforming," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 6, pp. 352–362, 2002.

[12] H. Saruwatari, S. Kurita, and K. Takeda, "Blind source separation combining frequency-domain ica and beamforming," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 5. IEEE, 2001, pp. 2733–2736.

[13] A. H. Khan, M. Taseska, and E. A. Habets, "A geometrically constrained independent vector analysis algorithm for online source extraction," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2015, pp. 396–403.

[14] Y. Mitsui, N. Takamune, D. Kitamura, H. Saruwatari, Y. Takahashi, and K. Kondo, "Vectorwise coordinate descent algorithm for spatially regularized independent low-rank matrix analysis," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 746–750.

[15] O. Hoshuyama, A. Sugiyama, and A. Hirano, "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters," *IEEE Transactions on signal processing*, vol. 47, no. 10, pp. 2677–2684, 1999.

[16] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Transactions on antennas and propagation*, vol. 30, no. 1, pp. 27–34, 1982.

[17] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[18] P. Smaragdis, B. Raj, and M. Shashanka, "Supervised and semi-supervised separation of sounds from single-channel mixtures," in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2007, pp. 414–421.

[19] F. Weninger, J. L. Roux, J. R. Hershey, and S. Watanabe, "Discriminative nmf and its application to single-channel source separation," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[20] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Supervised non-euclidean sparse nmf via bilevel optimization with applications to speech enhancement," in *2014 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*. IEEE, 2014, pp. 11–15.

[21] H. Nakajima, D. Kitamura, N. Takamune, H. Saruwatari, and N. Ono, "Bilevel optimization using stationary point of lower-level objective function for discriminative basis learning in nonnegative matrix factorization," *IEEE Signal Processing Letters*, vol. 26, no. 6, pp. 818–822, 2019.

[22] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2000.

[23] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis," *Neural computation*, vol. 21, no. 3, pp. 793–830, 2009.

[24] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[25] D. R. Hunter and K. Lange, "A tutorial on mm algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[26] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[27] R. Badeau, N. Bertin, and E. Vincent, "Stability analysis of multiplicative update algorithms and application to nonnegative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 21, no. 12, pp. 1869–1881, 2010.

[28] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of machine learning research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[29] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE transactions on audio, speech, and language processing*, vol. 15, no. 3, pp. 1066–1074, 2007.

[30] K. W. Wilson, B. Raj, and P. Smaragdis, "Regularized non-negative matrix factorization with temporal dependencies for speech denoising," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[31] G. J. Mysore and P. Smaragdis, "A non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 17–20.

[32] K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 4029–4032.

[33] J. Eggert and E. Korner, "Sparse coding and nmf," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, vol. 4. IEEE, 2004, pp. 2529–2533.

[34] E. Vincent, N. Bertin, and R. Badeau, "Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 109–112.

[35] Z. Wang and F. Sha, "Discriminative non-negative matrix factorization for single-channel speech separation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3749–3753.

[36] L. Molgedey and H. G. Schuster, "Separation of a mixture of independent signals using time delayed correlations," *Physical review letters*, vol. 72, no. 23, p. 3634, 1994.

[37] L. Tong, R.-W. Liu, V. C. Soon, and Y.-F. Huang, "Indeterminacy and identifiability of blind identification," *IEEE Transactions on circuits and systems*, vol. 38, no. 5, pp. 499–509, 1991.

[38] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[39] S.-i. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," in *Advances in neural information processing systems*, 1996, pp. 757–763.

[40] H. Buchner, R. Aichner, and W. Kellermann, "A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics," *IEEE transactions on speech and audio processing*, vol. 13, no. 1, pp. 120–134, 2004.

[41] K. Torkkola, "Blind separation of convolved sources based on information maximization," in *Neural Networks for Signal Processing VI. Proceedings of the 1996 IEEE Signal Processing Society Workshop*. IEEE, 1996, pp. 423–432.

[42] S. Makino, H. Sawada, and S. Araki, "Frequency-domain blind source separation," in *Blind Speech Separation*. Springer, 2007, pp. 47–78.

[43] N. Murata, S. Ikeda, and A. Ziehe, "An approach to blind source separation based on temporal structure of speech signals," *Neurocomputing*, vol. 41, no. 1-4, pp. 1–24, 2001.

[44] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of frequency-domain blind source separation," *IEEE transactions on speech and audio processing*, vol. 12, no. 5, pp. 530–538, 2004.

[45] H. Sawada, S. Araki, and S. Makino, "Measuring dependence of bin-wise separated signals for permutation alignment in frequency-domain bss," in *2007 IEEE International Symposium on Circuits and Systems*. IEEE, 2007, pp. 3247–3250.

[46] J. Anemüller and B. Kollmeier, "Amplitude modulation decorrelation for convolutive blind source separation," in *Proc. ICA*, 2000, pp. 215–220.

[47] N. Mitianoudis and M. E. Davies, "Audio source separation of convolutive mixtures," *IEEE transactions on Speech and Audio processing*, vol. 11, no. 5, pp. 489–497, 2003.

[48] R. Mukai, H. Sawada, S. Araki, and S. Makino, "Frequency-domain blind source separation of many speech signals using near-field and far-field models," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, no. 1, p. 083683, 2006.

[49] H. Saruwatari, S. Kurita, K. Takeda, F. Itakura, T. Nishikawa, and K. Shikano, "Blind source separation combining independent component analysis and beamforming," *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 11, p. 569270, 2003.

[50] M. Z. Ikram and D. R. Morgan, "Permutation inconsistency in blind speech separation: Investigation and solutions," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 1, pp. 1–13, 2004.

[51] H. Sawada, S. Araki, R. Mukai, and S. Makino, "Blind extraction of dominant target sources using ica and time-frequency masking," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 2165–2173, 2006.

[52] J.-F. Cardoso, "Infomax and maximum likelihood for blind source separation," *IEEE Signal processing letters*, vol. 4, no. 4, pp. 112–114, 1997.

[53] T. Kim, H. T. Attias, S.-Y. Lee, and T.-W. Lee, "Blind source separation exploiting higher-order frequency dependencies," *IEEE transactions on audio, speech, and language processing*, vol. 15, no. 1, pp. 70–79, 2006.

[54] N. Ono, "Stable and fast update rules for independent vector analysis based on auxiliary function technique," in *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2011, pp. 189–192.

[55] N. Ono and S. Miyabe, "Auxiliary-function-based independent component analysis for super-gaussian sources," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2010, pp. 165–172.

[56] T. Ono, N. Ono, and S. Sagayama, "User-guided independent vector analysis with source activity tuning," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 2417–2420.

[57] N. Ono, "Auxiliary-function-based independent vector analysis with power of vector-norm type weighting functions," in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, 2012, pp. 1–4.

[58] C. Févotte and J.-F. Cardoso, "Maximum likelihood approach for blind audio source separation using time-frequency gaussian source models," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.* IEEE, 2005, pp. 78–81.

[59] E. Vincent, S. Arberet, and R. Gribonval, "Underdetermined instantaneous audio source separation via local gaussian modeling," in *International*

Conference on Independent Component Analysis and Signal Separation. Springer, 2009, pp. 775–782.

[60] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[61] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[62] M. Kolbæk, Z.-H. Tan, and J. Jensen, "On the relationship between short-time objective intelligibility and short-time spectral-amplitude mean-square error for speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 2, pp. 283–295, 2018.

[63] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 2. IEEE, 2001, pp. 749–752.

[64] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 31–35.

[65] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," *arXiv preprint arXiv:1607.02173*, 2016.

[66] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 241–245.

[67] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[68] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[69] Y. Liu and D. Wang, "Divide and conquer: A deep casa approach to talker-independent monaural speaker separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2092–2102, 2019.

[70] J. Le Roux, G. Wichern, S. Watanabe, A. Sarroff, and J. R. Hershey, "Phasebook and friends: Leveraging discrete representations for source separation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 370–382, 2019.

[71] M. Delfarah and D. Wang, "Deep learning for talker-dependent reverberant speaker separation: An empirical study," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1839–1848, 2019.

[72] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1652–1664, 2016.

[73] S. Mogami, H. Sumino, D. Kitamura, N. Takamune, S. Takamichi, H. Saruwatari, and N. Ono, "Independent deeply learned matrix analysis for multichannel audio source separation," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1557–1561.

[74] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[75] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," *Advances in neural information processing systems*, vol. 27, pp. 3581–3589, 2014.

[76] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, pp. 2672–2680, 2014.

[77] Y. Bando, M. Mimura, K. Itoyama, K. Yoshii, and T. Kawahara, "Statistical speech enhancement based on probabilistic integration of variational autoencoder and non-negative matrix factorization," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 716–720.

[78] S. Leglaive, L. Girin, and R. Horaud, "A variance modeling framework based on variational autoencoders for speech enhancement," in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2018, pp. 1–6.

[79] S. Leglaive, U. Şimşekli, A. Liutkus, L. Girin, and R. Horaud, "Speech enhancement with variational autoencoders and alpha-stable distributions," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 541–545.

[80] M. Pariente, A. Deleforge, and E. Vincent, "A statistically principled and computationally efficient approach to speech enhancement using variational autoencoders," *arXiv preprint arXiv:1905.01209*, 2019.

[81] S. Pascual, A. Bonafonte, and J. Serra, "Segan: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[82] K. Sekiguchi, Y. Bando, A. A. Nugraha, K. Yoshii, and T. Kawahara, "Semi-supervised multichannel speech enhancement with a deep speech prior," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2197–2212, 2019.

[83] S. Leglaive, L. Girin, and R. Horaud, "Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 101–105.

[84] K. Sekiguchi, Y. Bando, K. Yoshii, and T. Kawahara, "Bayesian multichannel speech enhancement with a deep speech prior," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 1233–1239.

[85] B. Fuglede and F. Topsoe, "Jensen-shannon divergence and hilbert space embedding," in *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.* IEEE, 2004, p. 31.

[86] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[87] S. Martin Arjovsky and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34 th International Conference on Machine Learning, Sydney, Australia*, 2017.

[88] C. Villani, "Optimal transport, old and new, grundlehren der matematischen wissenschaften, 338 (2008)."

[89] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[90] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[91] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks,"

in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 266–273.

[92] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," *arXiv preprint arXiv:1907.12279*, 2019.

[93] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Nonparallel voice conversion with augmented classifier star generative adversarial networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2982–2995, 2020.

[94] ——, "Acvae-vc: Non-parallel voice conversion with auxiliary classifier variational autoencoder," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1432–1443, 2019.

[95] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," *Advances in neural information processing systems*, vol. 29, pp. 2172–2180, 2016.

[96] D. Barber and F. V. Agakov, "The im algorithm: A variational approach to information maximization," in *Advances in neural information processing systems*, 2003, p. None.

[97] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[98] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," *arXiv preprint arXiv:1804.04262*, 2018.

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

[100] M. R. Schroeder, "New method of measuring reverberation time," *The Journal of the Acoustical Society of America*, vol. 37, no. 6, pp. 1187–1188, 1965.

[101] S. Nakamura, K. Hiyane, F. Asano, and T. Endo, "Sound scene data collection in real acoustical environments," *Journal of the Acoustical Society of Japan (E)*, vol. 20, no. 3, pp. 225–231, 1999.

[102] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*. PMLR, 2017, pp. 933–941.

[103] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[104] Z. Shi, H. Lin, L. Liu, R. Liu, S. Hayakawa, S. Harada, and J. Han, "Furcanet: An end-to-end deep gated convolutional, long short-term memory, deep neural networks for single channel speech separation," *arXiv preprint arXiv:1902.00651*, 2019.

[105] Q. Yanmin and D. Yu, "Monaural multi-talker speech recognition with attention mechanism and gated convolutional networks," Jun. 30 2020, uS Patent 10,699,700.

[106] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[107] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[108] M. D. Zeiler, "Adadelta: An adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[109] J. S. Garofolo et al., "Csr-i (wsj0) complete ldc93s6a. web download," *URL: https://catalog.ldc.upenn.edu/LDC93S6A*, 1993.

[110] N. Ono, "Blind source separation on iphone in real environment," in *21st European Signal Processing Conference (EUSIPCO 2013)*. IEEE, 2013, pp. 1–5.

[111] T. Taniguchi, N. Ono, A. Kawamura, and S. Sagayama, "An auxiliary-function approach to online independent vector analysis for real-time blind source separation," in *2014 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*. IEEE, 2014, pp. 107–111.

[112] M. Sunohara, C. Haruta, and N. Ono, "Low-latency real-time blind source separation for hearing aids based on time-domain implementation of online independent vector analysis with truncation of non-causal components," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 216–220.

[113] Y. Liang, S. Naqvi, and J. Chambers, "Overcoming block permutation problem in frequency domain blind source separation when using auxiva algorithm," *Electronics letters*, vol. 48, no. 8, pp. 460–462, 2012.

[114] A. Brendel, T. Haubner, and W. Kellermann, "Spatially guided independent vector analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 596–600.

[115] M. Knaak, S. Araki, and S. Makino, "Geometrically constrained independent component analysis," *IEEE transactions on audio, speech, and language processing*, vol. 15, no. 2, pp. 715–726, 2007.

[116] Y. Zheng, K. Reindl, and W. Kellermann, "Analysis of dual-channel ica-based blocking matrix for improved noise estimation," *EURASIP journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 26, 2014.

[117] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 692–730, 2017.

[118] I. Cohen, “Analysis of two-channel generalized sidelobe canceller (gsc) with post-filtering,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 684–699, 2003.

[119] J. Bourgeois and W. Minker, “Linearly constrained minimum variance beamforming,” *Time-Domain Beamforming and Blind Source Separation: Speech Input in the Car Environment*, pp. 27–38, 2009.

[120] S. Araki, S. Makino, Y. Hinamoto, R. Mukai, T. Nishikawa, and H. Saruwatari, “Equivalence between frequency-domain blind source separation and frequency-domain adaptive beamforming for convolutive mixtures,” *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 11, p. 198923, 2003.

[121] A. Lombard, T. Rosenkranz, H. Buchner, and W. Kellermann, “Multidimensional localization of multiple sound sources using averaged directivity patterns of blind source separation systems,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 233–236.

[122] Y. Zheng, A. Lombard, and W. Kellermann, “An improved combination of directional bss and a source localizer for robust source separation in rapidly time-varying acoustic scenarios,” in *2011 Joint workshop on Hands-Free speech communication and microphone arrays*. IEEE, 2011, pp. 58–63.

[123] J. Thiemann, N. Ito, and E. Vincent, “Demand: a collection of multi-channel recordings of acoustic noise in diverse environments,” in *Proc. Meetings Acoust*, 2013.

[124] H. L. Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons, 2004.

[125] E. M. Grais and H. Erdogan, “Discriminative nonnegative dictionary learning using cross-coherence penalties for single channel source separation.” in *Interspeech*, 2013, pp. 808–812.

[126] G. Bao, Y. Xu, and Z. Ye, "Learning a discriminative dictionary for single-channel speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 7, pp. 1130–1138, 2014.

[127] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Discriminative non-negative matrix factorization for multiple pitch estimation." in *ISMIR*. Citeseer, 2012, pp. 205–210.

[128] K. Kwon, J. W. Shin, and N. S. Kim, "Target source separation based on discriminative nonnegative matrix factorization incorporating cross-reconstruction error," *IEICE TRANSACTIONS on Information and Systems*, vol. 98, no. 11, pp. 2017–2020, 2015.

[129] H. Kameoka, "Non-negative matrix factorization and its variants for audio signal processing," in *Applied Matrix and Tensor Variate Data Analysis*. Springer, 2016, pp. 23–50.

[130] E. Vincent, S. Watanabe, J. Barker, and R. Marxer, "The 4th chime speech separation and recognition challenge," *URL: http://spandh. dcs. shef. ac. uk/chime challenge {Last Accessed on 1 August, 2018}*, 2016.

[131] Y. Shi, H. Chen, Z. Tang, L. Li, D. Wang, and J. Han, "Can we trust deep speech prior?" *arXiv preprint arXiv:2011.02110*, 2020.